Faculté d'Economie, de Gestion et de Sciences Sociales
Ecole d'Administration des Affaires

Modelling Financial Data and Portfolio Optimization Problems

Dissertation présentée par
Michaël Schyns
pour l'obtention du grade de
Docteur en Sciences de Gestion

Promoteur:
Pr. Y. Crama
Université de Liège

Membres du jury:
Pr. A. Corhay (Université de Liège)
Pr. Y. Crama (Université de Liège)
Pr. W.G. Hallerbach (Erasmus University)
Pr. G. Hübner (Université de Liège)
Pr. A.W.J. Kolen (Maastricht University)
Pr. M. Roubens (Université de Liège)

Année Académique
2000-2001

Acknowledgment

I would like to thank all the people who helped me in one way or another to complete my dissertation.
Since the beginning, people have accepted to share their knowledge and time with me:

- Pr. Y. Crama has been a dynamic and motivating scientific supervisor. Despite his many occupations, his time has never mattered to answer my queries.
- Pr. A. Kolen, who initiated the second part of the thesis, has agreed to be a member of my thesis comittee. As such, we often met to coordinate the different parts of the work.
- Before and after jumping on board of my thesis comittee, Pr. G. Hübner managed to spare time to discuss with me about the financial part of my thesis each time I asked.
- Pr. M. Roubens has invited me several times to present my work and to discuss it with his team.
- Pr. A. Corhay's door has always stayed open for me. Either for my thesis or for many other reasons, he has listened to me and has helped me throughout.
- At home, my wife and my family have endured my ups and downs and have never stopped supporting me.
- Last but not least, Pr. R. Moors who has been more than my boss. Even if he thinks that helping me was part of his job ('I'm paid for it' as he puts it), I do not dare estimating the number of hours he spent listening to me, talking to me, cheering me up,...

I thank also Professor W.G. Hallerbach who, together with Professors A. Corhay, Y. Crama, G. Hübner, A. Kolen and M. Roubens, have accepted to be on my Jury.
I should not forget to mention Alexandre and Axelle who pleasantly agreed to do without me and play with their mother and grandparents to let me finish my work.

Contents

Chapter 1

General introduction

This doctoral dissertation in *management science*, entitled "Modelling Financial Data and Portfolio Optimization Problems", consists of two independent parts, whose unifying theme is the construction and solution of mathematical programming models motivated by portfolio selection problems. As such, this work is located at the interface of operations research and of finance. It draws heavily on techniques and theoretical results originating in both disciplines.

The first part of the dissertation *(Chapter 2)* deals with an extension of Markowitz model and takes into account some of the side-constraints faced by a decision-maker when composing an investment portfolio, viz. lower and upper bounds on the quantities traded, and upper bounds on the number of assets included in the portfolio. We focus on the algorithmic difficulties raised by this model and *we* describe an original simulated annealing heuristic for its solution.

The second (and largest) part of the thesis deals with a new multiperiod model for the optimization of a portfolio of options linked to a single index *(Chapters 4-10).* The objective of the model is to maximize the expected return of the portfolio under constraints limiting its value-at-risk. The model contains several interesting features, like the possibility to rebalance the portfolio with options introduced at the start of each period, explicit consideration of transaction costs, realistic pricing of options, consideration of advanced probability models to represent the future, etc. Some deep theoretical results from the financial literature *are exploited* in order to enrich the model and to extend its applicability. *In particular, several available schemes for the generation of scenarios and for option pricing have been critically examined, and the most appropriate ones have been implemented.* Furthermore, several optimization approaches (heuristic or exact procedures) have also been *developed,* implemented and tested.

The models investigated in the dissertation bear on very different portfolio problems, draw *on separate streams of scientific literature, and are handled by distinct algorithmic techniques. Therefore, the corresponding parts of the dissertation are fully independent, and each part contains its own specific introduction and literature review.*

PART ONE:

Simulated Annealing for a generalized mean-variance model

Chapter 2

Simulated Annealing for a generalized mean-variance model

2.1 Introduction

Markowitz' mean-variance model of portfolio selection is one of the best known models in finance. In its basic form, this model requires to determine the composition of a portfolio of assets which minimizes risk while achieving a predetermined level of expected return. The pioneering role played by this model in the development of modern portfolio theory is unanimously recognized (see e.g. [11] for a brief historical account).

From a practical point of view, however, the Markowitz model may often be considered too basic, as it ignores many of the constraints faced by real-world investors: trading limitations, size of the portfolio, etc. Including such constraints in the formulation results in a nonlinear mixed integer programming problem which is considerably more difficult to solve than the original model. Several researchers have attempted to attack this problem by a variety of techniques (decomposition, cutting planes, interior point methods, ...), but there appears to be room for much improvement on this front. In particular, exact solution methods fail to solve large-scale instances of the problem. Therefore, in this chapter, we investigate the ability of the simulated annealing metaheuristic (SA) to deliver high-quality solutions for the mean-variance model enriched by additional constraints.

The remainder of this chapter is organized in six sections. Section 2 introduces the portfolio selection model that we want to solve. Section 3 sums up the basic structure of simulated annealing algorithms. Section 2.4 contains a detailed description of our algorithm. Here, we make an attempt to underline the difficulties encountered when tailoring the SA metaheuristic to the problem at hand. Notice, in particular, that our model involves continuous as well as discrete variables, contrary to most applications of simulated annealing. Also, the constraints are of various types and cannot be handled in a uniform way. In Section 5, we discuss some details of the implementation. Section 6 reports on computational experiments carried out on a sample of 151 US stocks. Finally, the last section contains a summary of our work and some conclusions.

2.2 Portfolio selection issues

2.2.1 Generalities

In order to handle portfolio selection problems in a formal framework, three types of questions (at least) must be explicitly addressed:

1. data modelling, in particular the behavior of asset returns;
2. the choice of the optimization model, including:
   - the nature of the objective function;
   - the constraints faced by the investor;
3. the choice of the optimization technique.

Although our work focuses mostly on the third step, we briefly discuss the whole approach since all the steps are interconnected to some extent.

The first requirement is to understand the nature of the data and to be able to correctly represent them. Markowitz' model (described in the next section) assumes for instance that the asset returns follow a multivariate normal distribution. In particular, the first two moments of the distribution suffice to describe completely the distribution of the asset returns and the characteristics of the different portfolios. Real markets often exhibit more intricacies, with distributions of returns depending on moments of higher-order (skewness, kurtosis, etc.), and distribution parameters varying over time. Analyzing and modelling such complex financial data is a whole subject in itself, which we do not tackle here explicitly. We rather adopt the classical assumptions of the mean-variance approach, where (pointwise estimates of) the expected returns and the variance-covariance matrix are supposed to provide a satisfactory description of the asset returns. Also, we do not address the origin of the numerical data. Note that some authors rely for instance on factorial models of the asset returns, and take advantage of the properties of such models to improve the efficiency of the optimization techniques (see e.g. [2, 58]). By contrast, the techniques that we develop here do not depend on any specific properties of the data, so that some changes of the model (especially of the objective function) can be performed while preserving our main conclusions.

When building an optimization model of portfolio selection, a second requirement consists in identifying the objective of the investor and the constraints that he is facing. As far as the objective goes, the quality of the portfolio could be measured using a wide variety of utility functions. Following again Markowitz' model, we assume here that the investor is risk averse and wants to minimize the variance of the investment portfolio subject to the expected level of final wealth. It should be noted, however, that this assumption does not play a crucial role in our algorithmic developments, and that the objective could be replaced by a more general utility function without much impact on the optimization techniques that we propose.

As far as the constraints of the model go, we are especially interested in two types of complex constraints limiting the number of assets included in the portfolio (thus reflecting some behavioral or institutional restrictions faced by the investor), and the minimal quantities which can be traded when rebalancing an existing portfolio (thus reflecting individual or market restrictions). This topic is covered in more detail in Section 2.2.2.

The final ingredient of a portfolio selection method is an algorithmic technique for the optimization of the chosen model. This is the main topic of the present chapter. In view of the complexity of our model (due, to a large extent, to the constraints mentioned in the previous paragraph), and to the large size of realistic problem instances, we have chosen to work with a simulated annealing metaheuristic. An in-depth study has been performed to optimize the speed and the quality of the algorithmic process, and to analyze the impact of various parameter choices.

In the remainder of this section, we return in more detail to the description of the model, and we briefly survey previous work on this and related models.

2.2.2 The optimization model

The Markowitz mean-variance model

The problem of optimally selecting a portfolio among n assets was formulated by Markowitz in 1952 as a constrained quadratic minimization problem (see [50], [26], [48]). In this model, each asset is characterized by a return varying randomly with time. The risk of each asset is measured by the variance of its return. If each component x_i of the n-vector x represents the proportion of an investor's wealth allocated to asset i, then the total return of the portfolio is given by the scalar product of x by the vector of individual asset returns. Therefore, if $R = (R_1, \ldots, R_n)$ denotes the n-vector of expected returns of the assets and C the $n \times n$ covariance matrix of the returns, we obtain the mean portfolio return by the expression $\sum_{i=1}^{n} R_i x_i$ and its level of risk by $\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_i x_j$.

Markowitz assumes that the aim of the investor is to design a portfolio which minimizes risk while achieving a predetermined expected return, say *Rexp*. Mathematically, the problem can be formulated as follows for any value of *Rexp*:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_i x_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} R_i x_i = Rexp \\ & \sum_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \ \text{ for } i = 1, \ldots, n. \end{aligned} \tag{2.1}$$

The first constraint expresses the requirement placed on expected return. The second constraint, called *budget constraint*, requires that 100% of the budget be invested in the portfolio. The nonnegativity constraints express that no short sales are allowed.

The set of optimal solutions of the Markowitz model, parametrized over all possible values of *Rexp*, constitutes the *mean-variance frontier* of the portfolio selection problem. It is usually displayed as a curve in the plane where the ordinate is the expected portfolio return and the abscissa is its standard deviation.

If the goal is to draw the whole frontier, an alternative form of the model can also be used where the constraint defining the required expected return is removed and a new weighted term representing the portfolio return is included in the objective function. Hereafter, we shall use the initial formulation involving only the variance of the portfolio in the objective function.

Extensions of the basic model

In spite of its theoretical interest, the basic mean-variance model is often too simplistic to represent the complexity of real-world portfolio selection problems in an adequate fashion. In order to enrich the model, we need to introduce more realistic constraints. The present section discusses some of them.

Consider the following portfolio selection model (similar to a model described by Perold [58]).

Model (PS):

min	$\sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_ix_j$	Objective function
s.t.	$\sum_{i=1}^{n} R_ix_i = Rexp$	Return contraint
	$\sum_{i=1}^{n} x_i = 1$	Budget constraint
	$\underline{x}_i \leq x_i \leq \overline{x}_i \ (1 \leq i \leq n)$	Floor and ceiling constraints
	$\max(x_i - x_i^{(0)}, 0) \leq \overline{B}_i \ (1 \leq i \leq n)$	Turnover (purchase) constraints
	$\max(x_i^{(0)} - x_i, 0) \leq \overline{S}_i \ (1 \leq i \leq n)$	Turnover (sale) constraints
	$x_i = x_i^{(0)}$ or $x_i \geq (x_i^{(0)} + \underline{B}_i)$	
	or $x_i \leq (x_i^{(0)} - \underline{S}_i) \ (1 \leq i \leq n)$	Trading constraints
	$\lvert\{i \in \{1,\ldots,n\} : x_i \neq 0\}\rvert \leq N$	Maximum number of assets

- *Return and budget constraints:* These two constraints have already been encountered in the basic model.

- *Floor and ceiling constraints:* These constraints define lower and upper limits on the proportion of each asset which can be held in the portfolio. They may model institutional restrictions on the composition of the portfolio. They may also rule out negligible holdings of asset in the portfolio, thus making its control easier. Notice that the floor contraints generalize the nonnegativity constraints imposed in the original model.

- *Turnover constraints:* These constraints impose upper bounds on the variation of the holdings from one period to the next. Here, $x_i^{(0)}$ denotes the weight of asset i in the initial portfolio , $\overline{B}_i$ denotes the maximum purchase and $\overline{S}_i$ denotes the maximum sale of asset i during the current period ($1 \leq i \leq n$). Notice that such limitations could also be modelled, indirectly, by incorporating transaction costs (taxes, commissions, unliquidity effects, ...) in the objective function or in the constraints.

- *Trading constraints:* Lower limits on the variations of the holdings can also be imposed in order to reflect the fact that, typically, an investor may not be able, or may not want, to modify the portfolio by buying or selling tiny quantities of assets. A first reason may be that the contracts must bear on significant volumes. Another reason may be the existence of relatively high fixed costs linked to the transactions. These constraints are disjunctive in nature: for each asset i, either the holdings are not changed, or a minimal quantity $\underline{B}_i$ must be bought, or a minimal quantity $\underline{S}_i$ must be sold.

- *Maximum number of assets:* This constraint limits to N the number of assets included in the portfolio, e.g. in order to facilitate its management.

2.2.3 Solution approaches

The complexity of solving portfolio selection problems is very much related to the type of constraints that they involve.

The simplest situation is obtained when the nonnegativity constraints are omitted from the basic model (2.1) (thus allowing short sales; see e.g. [26], [45], [48]). In this case, a closed-form solution is easily obtained by classical Lagrangian methods and various approaches have been proposed to increase the speed of resolution for the computation of the whole mean-variance frontier or the computation of a specific portfolio combined with an investment at the risk-free interest rate [45].

The problem becomes more complex when nonnegativity constraints are added to the formulation, as in the Markowitz model (2.1). The resulting *quadratic programming* problem, however, can still be solved efficiently by specialized algorithms such as Wolfe's adaptation of the simplex method [70]. The same technique allows to handle arbitrary linear constraints, like the floor and ceiling constraints or the turnover constraints. Notice, however, that even in this framework, the problem becomes increasingly hard to manage and to solve as the number of assets increases. As a consequence, ad hoc methods have been developed to take advantage of the sparsity or of the special structure of the covariance matrix, (e.g., when factor models of returns are postulated; see [58], [2]).

When the model involves constraints on minimal trading quantities or on the maximum number of assets in the portfolio, as in model (PS), then we enter the field of mixed integer nonlinear programming and classical algorithms are typically unable to deliver the optimal value of the problem. (Actually, very few commercial packages are even able to handle this class of problems.)

Several researchers took up this challenge, for various versions of the problem. Perold [58], whose work is most often cited in this context, included a broad class of constraints in his model, but did not place any limitation on the number of assets in the portfolio. His optimization approach is explicitly restricted to the consideration of factorial models, which, while reducing the number of decision variables, lead to other numerical and statistical difficulties. Moreover, some authors criticize the results obtained when his model is applied to certain types of markets.

Several other researchers have investigated variants of model (PS) involving only a subset of the constraints. This is the case for instance of Dembo, Mulvey and Zenios [18] (with network flow models), Konno and Yamazaki [40] (with an absolute deviation approach to the measure of risk, embedded in linear programming models), Takehara [65] (with an interior point algorithm), and Bienstock [2] (with a 'branch and cut' approach). Dahl, Meeraus and Zenios [14], Takehara [65] and Hamza and Janssen [31] discuss some of this work.

Few authors seem to have investigated the application of local search metaheuristics for the solution of portfolio selection problems. Catanas [6] has investigated some of the theoretical properties of a neighborhood structure in this framework. Loraschi, Tettamanzi, Tomassini and Verda [46] proposed a genetic algorithm approach. Chang, Meade, Beasley and Sharaiha's work [8] is closest to ours (and was carried out concurrently). These authors have experimented with a variety of metaheuristics, including simulated annealing, on model (PS) *without* trading and turnover constraints. As we shall see in Section 2.6, the trading constraints actually turned out to be the hardest to handle in our experiments, and they motivated much of the sophisticated machinery described in Section 2.4. We also work directly with the return constraint in equality form, rather than incorporating it as a Lagrangian term in the objective function. This allows us to avoid some of the difficulties linked to the fact that, as explained in [8], the efficient frontier cannot possibly be mapped entirely in the Lagrangian approach, due to its discontinuity. In this sense, our work can be viewed as complementary to [8].

We propose to investigate the solution of the complete model (PS) presented in Section 2.2.2 by a simulated annealing algorithm. Our goal is to develop an approach which, while giving up claims to optimality, would display some *robustness* with respect to various criteria, including:

- quality of solutions;
- speed;
- ease of addition of new constraints;
- ease of modification of the objective function (e.g. when incorporating higher moments than the variance, or when considering alternative risk criteria like the semi-variance).

In the next section, we review the basic principles and terminology of the simulated annealing metaheuristic.

2.3 Simulated annealing

Detailed discussions of simulated annealing can be found in van Laarhoven and Aarts [67], Aarts and Lenstra [1] or in the survey by Pirlot [59]. We only give here a very brief presentation of the method.

Simulated annealing is a generic name for a class of optimization heuristics that perform a stochastic neighborhood search of the solution space. The major advantage of SA over classical local search methods is its ability to avoid getting trapped in local minima while searching for a global minimum. The underlying idea of the heuristic arises from an analogy with certain thermodynamical processes (cooling of a melted solid). Kirkpatrick, Gelatt and Vecchi [37] and Černý [7] pioneered its use for combinatorial problems. For a generic problem of the form

$$\min \; F(x) \;\; \text{s.t. } x \in X,$$

the basic principle of the SA heuristic can be described as follows. Starting from a current solution x, another solution y is generated by taking a stochastic step in some neighborhood of x. If this new proposal improves the value of the objective function, then y replaces x as the new current solution. Otherwise, the new solution y is accepted with a probability that decreases with the magnitude of the deterioration and in the course of iterations. (Notice the difference with classical descent approaches, where only improving moves are allowed and the algorithm may end up quickly in a local optimum.)

More precisely, the generic simulated annealing algorithm performs the following steps:

- Choose an initial solution $x^{(0)}$ and compute the value of the objective function $F(x^{(0)})$. Initialize the incumbent solution (i.e. the best available solution), denoted by (x^*, F^*), as: $(x^*, F^*) \leftarrow (x^{(0)}, F(x^{(0)}))$.
- Until a stopping criterion is fulfilled and for n starting from 0, do:
  - Draw a solution x at random in the neighborhood $V(x^{(n)})$ of $x^{(n)}$.
  - If $F(x) \leq F(x^{(n)})$ then $x^{(n+1)} \leftarrow x$ and
    if $F(x) \leq F^*$ then $(x^*, F^*) \leftarrow (x, F(x))$.
    If $F(x) > F(x^{(n)})$ then draw a number p at random in $[0, 1]$ and
    if $p \leq p(n, x, x^{(n)})$ then $x^{(n+1)} \leftarrow x$
    else $x^{(n+1)} \leftarrow x^{(n)}$.

The function $p(n, x, x^{(n)})$ is often taken to be a Boltzmann function inspired from thermodynamics models:

$$p(n, x, x^{(n)}) = \exp(-\frac{1}{T_n}\Delta F_n) \tag{2.2}$$

where $\Delta F_n = F(x) - F(x^{(n)})$ and T_n is the *temperature* at step n, that is a nonincreasing function of the iteration counter n. In so-called *geometric cooling schedules*, the temperature is kept unchanged during each successive *stage*, where a stage consists of a constant number L of consecutive iterations. After each stage, the temperature is multiplied by a constant factor $\alpha \in (0, 1)$.

Due to the generality of the concepts that it involves, SA can be applied to a wide range of optimization problems. In particular, no specific requirements need to be imposed on the objective function (derivability, convexity, ...) nor on the solution space. Moreover, it can be shown that the metaheuristic converges asymptotically to a global minimum [67].

From a practical point of view, the approach often yields excellent solutions to hard optimization problems. Surveys and descriptions of applications can be found in van Laarhoven and Aarts [67], Osman and Laporte [57] or Aarts and Lenstra [1].

Most of the original applications of simulated annealing have been made to problems of a combinatorial nature, where the notions of 'step' or 'neighbor' usually find a natural interpretation. Due to the success of simulated annealing in this framework, several researchers have attempted to extend the approach to continuous minimization problems (see van Laarhoven and Aarts [67], Dekkers and Aarts [15], CSEP [10], Zabinsky et al. [71]). However, few practical applications appear in the literature. A short list can be found in the previous references, in particular in Osman and Laporte [57].

We are especially interested in these extensions, since portfolio selection typically involves a mix of continuous and discrete variables (see Section 2). One of the aims of our work, therefore, is to gain a better understanding of the difficulties encountered when applying simulated annealing to mixed integer nonlinear optimization problems and to carry out an exploratory investigation of the potentialities offered by SA in this framework.

2.4 Simulated annealing for portfolio selection

2.4.1 Generalities: How to handle constraints ...

In order to apply the SA algorithm to problem (PS), we have to undertake an important tailoring work. Two notions have to be defined in priority, i.e. those of *solution* (or encoding thereof) and *neighborhood*.

We simply encode a solution of (PS) as an n-dimensional vector x, where each variable x_i represents the holdings of asset i in the portfolio. The quality of a solution is measured by the variance of the portfolio, that is $x^t C x$.

Now, how do we handle the constraints, that is, how do we make sure that the final solution produced by the SA algorithm satisfies all the constraints of (PS) ?

The first and most obvious approach enforces feasibility throughout *all* iterations of the SA algorithm and forbids the consideration of any solution violating the constraints. This implies that the neighborhood of a current solution must entirely consist of feasible solutions. A second approach, by contrast, allows the consideration of infeasible solutions but adds a penalty term to the objective function for each violated constraint: the larger the violation of the constraint, the larger the increase in the value of the objective function. A portfolio which is unacceptable for the investor must be penalized enough to be rejected by the minimization process.

The "all-feasible" vs. "penalty" debate is classical in the optimization literature. In the context of the simplex algorithm, for instance, infeasible solutions are temporarily allowed in the initial phase of the *big-M* method, while feasibility is enforced thereafter by an adequate choice of the variable which is to leave the basis at each iteration. For a discussion of this topic in the framework of local search heuristics, see e.g. the references in Pirlot [59].

Both approaches, however, are not equally convenient in all situations and much of the discussion in the next subsections will center around the "right" choice to make for each class of constraints. Before we get to this discussion, let us first line up the respective advantages and inconvenients of each approach.

When penalties are used, the magnitude of each penalty should depend on the magnitude of the violation of the corresponding constraint, but must also be scaled relatively to the variance of the portfolio. A possible expression for the penalties is

$$a \times |violation|^p \tag{2.3}$$

where a and p are scaling factors. For example, the violation of the return constraint can be represented by the difference between the required portfolio return ($Rexp$) and the current solution return ($R^t x$). The violation of the floor constraint for asset i can be expressed as the difference between the minimum admissible level $\underline{x}_i$ and the current holdings x_i, when this difference is positive.

The first inconvenient of this method is that it searches a solution space whose size may be considerably larger than the size of the feasible region. This process may require many iterations and prohibitive computation time.

The second inconvenient stems from the scaling factors: it may be difficult to define adequate values for a and p. If these values are too small, then the penalties do not play their expected role and the final solution may be infeasible. On the other hand, if a and p are too large, then the term $x^t Cx$ becomes negligible with respect to the penalty; thus, small variations of x can lead to large variations of the penalty term, which mask the effect of the variance term.

Clearly, the correct choice of a and p depends on the scale of the data, i.e. on the particular instance at hand! It appears very difficult to automate this choice. For this reason, we use penalties for "soft" constraints only, and when nothing else works.

In our implementations, we have selected values for a and p as follows. First, we let $a = V/\epsilon^p$, where V is the variance of the most risky asset, that is $V = \max_{1 \le i \le n} C_{ii}$, and ϵ is a measure of numerical accuracy. Since the variance of any portfolio lies in the interval $[0, V]$, this choice of a guarantees that every feasible portfolio yields a better value of the objective function than any portfolio which violates a constraint by ϵ or more (but notice that smaller violations are penalized as well). The value of p can now be used to finetune the magnitude of the penalty as a function of the violation: in our experiments, we have set $p = 2$.

Let us now discuss the alternative, all-feasible approach, in which the neighborhood of the current solution may only contain solutions that satisfy the given subset of constraints. The idea that we implemented here (following some of the proposals made in the literature on stochastic global optimization) is to draw a direction at random and to take a small step in this direction away from the current solution. The important features of such a move is that both its *direction* and *length* are computed so as to respect the constraints. Moreover, the holdings of only a few assets are changed during the move, meaning that the feasible direction is chosen in a low-dimensional subspace. This simplifies computations and provides an immediate translation of the concept of "neighbor".

The main advantage of this approach is that no time is lost investigating infeasible solutions. The main disadvantage is that it is not always easy to select a neighbor in this way, so that the resulting moves may be quite contrived, their computation may be expensive and the search process may become inflexible. On the other hand, this approach seems to be the only reasonable one for certain constraints, like for example the trading constraints.

For each class of constraints, we had to ponder the advantages and disadvantages of each approach. When a constraint must be strictly satisfied or when it is possible to enforce it efficiently without penalties in the objective function, then we do so. This is the case for the constraints on budget, return and maximum number of assets. A mixed approach is used for the trading, floor, ceiling and turnover constraints.

In the next sections, we successively consider each class of constraints, starting with those that are enforced without penalties.

2.4.2 Budget and return constraints

Basic principle

The budget constraint must be strictly satisfied, since its unique goal is to norm the solution. Therefore, it is difficult to implement this constraint through penalties.

The same conclusion applies to the return constraint, albeit for different reasons. Indeed, our aim is to compute the whole mean-variance frontier. To achieve this aim, we want to let the expected portfolio return vary uniformly in its feasible range and to determine the optimal risk associated with each return. In order to obtain meaningful results, the optimal portfolio computed by the procedure should have the exact required return. In our experience, the approach relying on penalties was completely inadequate for this purpose.

In view of these comments, we decided to restrict our algorithm to the consideration of solutions that strictly satisfy the return and the budget constraints. More precisely, given a portfolio x, the neighborhood of x contains all solutions x' with the following property: there exist three assets, labeled 1, 2 and 3 without loss of generality, such that

$$\begin{cases} x'_1 = x_1 - step \\ x'_2 = x_2 + step * (R_1 - R_3)/(R_2 - R_3) \\ x'_3 = x_3 + step * (R_2 - R_1)/(R_2 - R_3) \\ x'_i = x_i \qquad \text{for all } i > 3, \end{cases} \tag{2.4}$$

where *step* is a (small) number to be further specified below. It is straightforward to check that x' satisfies the return and budget constraints when x does so. Geometrically, all neighbors x' of the form (2.4) lie on a line passing through x and whose direction is defined by the intersection of the 3-dimensional subspace associated to assets 1, 2, 3 with the two hyperplanes associated to the budget constraint and the return constraint, respectively. Thus, the choice of three assets determines the direction of the move, while the value of *step* determines its amplitude.

Observe that, in order to start the local search procedure, it is easy to compute an initial solution which satisfies the budget and return constraints. Indeed, if x denotes an arbitrary portfolio and *min* (resp. *max*) is the subscript of the asset with minimum (resp. maximum) expected return, then a feasible solution is obtained upon replacing x_{min} and x_{max} by x'_{min} and x'_{max}, where:

$$\begin{cases} x'_{min} = [Rexp - \sum_{i \neq min,max}^{n} x_i R_i - (x_{min} + x_{max}) R_{max}]/(R_{min} - R_{max}) \\ x'_{max} = x_{min} + x_{max} - x'_{min}. \end{cases} \tag{2.5}$$

The resulting solution may violate some of the additional constraints of the problem (trading, turnover, etc.) and penalties will need to be introduced in order to cope with this difficulty. This point will be discussed in sections to come.

Direction of moves

Choosing a neighbor of x, as described by (2.4), involves choosing the direction of the move, i.e. choosing three assets whose holdings are to be modified. In our initial attempts, we simply drew the indices of these assets randomly and uniformly over $\{1, \ldots, n\}$. Many of the corresponding moves, however, were nonimproving, thus resulting in slow convergence of the algorithm.

We have been able to improve this situation by guiding the choice of the three assets to be modified. Observe that the assets whose return is closest to the required portfolio return have (intuitively) a higher probability to appear in the optimal portfolio than the remaining ones. (This is most obvious for portfolios with "extreme" returns: consider for example the case where we impose nonnegative holdings and we want to achieve the highest possible return, i.e. R_{max}.)

To account for this phenomenon, we initially sort all the assets by nondecreasing return. For each required portfolio return *Rexp*, we determine the asset whose return is closest to *Rexp* and we store its position, say q, in the sorted list. At each iteration of the SA algorithm, we choose the first asset to be modified by computing a random number normally distributed with mean q and with standard deviation large enough to cover all the list: this random number points to the position of the first asset in the ordered list. The second and third assets are then chosen uniformly at random.

Amplitude of moves

Let us now turn to the choice of the *step* parameter in (2.4). In our early attempts, *step* was fixed at a small constant value (so as to explore the solution space with high precision). The results appeared reasonably good but required extensive computation time (as compared to later implementations and to the quadratic simplex method, when this method was applicable).

In order to improve the behavior the algorithm, it is useful to realize that, even if a small value of *step* necessarily produces a small modification of the holdings of the first asset, it is more difficult to predict its effect on the other assets (see (2.4)). This may result in poorly controlled moves, whose amplitude may vary erratically from one iteration to the next.

As a remedy, we chose to construct a ball around each current solution and to restrict all neighbors to lie on the surface of this ball (this is inspired by several techniques for random sampling and global optimization; see e.g. Lovász and Simonovits [47] or Zabinsky et al. [71]). The euclidean length of each move is now simply determined by the radius of the ball. Furthermore, in view of equations (2.4), *step* is connected to the radius by the relation:

$$step = \pm \frac{radius * (R_2 - R_3)}{\sqrt{(R_2 - R_3)^2 + (R_1 - R_3)^2 + (R_2 - R_1)^2}}, \tag{2.6}$$

where the $\pm$ sign can be picked arbitrarily (we fix it randomly).

Now, how should we choose the radius of the ball? On the one hand, we want this value to be relatively small, so as to achieve sufficient precision. On the other hand, we can play with this parameter in order to enforce some of the constraints which have not been explicitly considered yet (floor, ceiling, trading, etc.). Therefore, we will come back to a discussion of this point in subsequent sections.

2.4.3 Maximum number of assets constraint

This cardinality constraint is combinatorial in nature. Moreover, a "natural" penalty approach based on measuring the extent of the violation:

$$violation = |\{i \in \{1, \ldots, n\} \, : \, x_i \neq 0\}| - N$$

(see model (PS)) does not seem appropriate to handle this constraint: indeed, all the neighbors of a solution are likely to yield the same penalty, except when an asset exceptionally appears in or disappears from the portfolio. Other types of penalties could conceivably be considered in order to circumvent the difficulty caused by this "flat landscape" (see e.g. [35, 59] for a discussion of similar issues arising in graph coloring or partitioning problems). We rather elected to rely on an all-feasible approach, whereby we restrict the choice of the three assets whose holdings are to be modified, in such a way as to maintain feasibility at every iteration. Let us now proceed with a case-by-case discussion of this approach.

First, observe that the initial portfolio only involves two assets (see Section 2.4.2) and hence is always feasible with respect to the cardinality constraint (we disregard the trivial case where $N = 1$).

Now, if the current portfolio involves $N - k$ assets, with $k \geq 1$, then we simply make sure, as we draw the three assets to be modified, that at most k of them are not already in the current portfolio. This ensures that the new portfolio involves at most N assets.

The same logic, however, cannot be used innocuously when the current portfolio involves exactly N assets: indeed, this would lead to a rigid procedure whereby no new asset would ever be allowed into the portfolio, unless one of the current N assets disappears from the portfolio by pure chance (that is, as the result of numerical cancelations in equations (2.4)). Therefore, in this case, we proceed as follows. We draw three assets at random, say assets 1, 2 and 3, in such a way that at most one of them is not in the current portfolio. If all three assets already are in the portfolio, then we simply determine the new neighbor as usual. Otherwise, assume for instance that assets 1 and 2 are in the current portfolio, but asset 3 is not. Then, we set the parameter *step* equal to x_1 in equations (2.4). In this way, $x'_1 = 0$ in the neighbor solution: the move from x to x' can be viewed as substituting asset 1 by asset 3 and rebalancing the portfolio through appropriate choices of the holdings x'_2 and x'_3.

2.4.4 Floor, ceiling and turnover constraints

The floor, ceiling and turnover constraints are similar to each other, since each of them simply defines a minimum or maximum bound on holdings. Therefore, our program automatically converts all turnover purchase constraints into ceiling constraints and all turnover sales constraints into floor constraints.

Suppose now that we know which three assets (say, 1, 2 and 3) must be modified at the current move from solution x to solution x', and suppose that the amplitude of the move has not been determined by the cardinality constraint (see previous subsection). Then, it is easy to determine conditions on the value of *step* such that x' satisfies the floor and ceiling constraints. Indeed, combining the latter constraints with equations (2.4) leads to the following conditions:

$$\begin{cases} x_1 - \overline{x}_1 \leq step \leq x_1 - \underline{x}_1 \\ \underline{x}_2 - x_2 \leq step * (R_1 - R_3)/(R_2 - R_3) \leq \overline{x}_2 - x_2 \\ \underline{x}_3 - x_3 \leq step * (R_2 - R_1)/(R_2 - R_3) \leq \overline{x}_3 - x_3. \end{cases} \tag{2.7}$$

These conditions yield a feasible interval of variation for *step* and hence (via equation (2.6)) for the radius of the ball limiting the move from x to x'. We denote by $[lb, ub]$ the feasible interval for $radius$.

Let us first assume that the interval $[lb, ub]$ is non empty (and, for practical purposes, not too "small"). Then, different strategies are applicable. We could start a linesearch optimization process to find the optimal value of the radius in $[lb, ub]$ (i.e., the value of the radius leading to the best neighbor x'). We have not experimented with this approach and have rather implemented a simpler option. We initially pick a small positive constant ρ. If, at any iteration, ρ is an admissible value for the radius, i.e. if $\rho \in [lb, ub]$, then we set the radius of the ball equal to ρ. Otherwise, if ρ is larger than ub (resp. smaller than lb), then we set the radius equal to ub (resp. lb).

Actually, in practice, we do not work with a single value of ρ but with two values, say $\rho_1 > \rho_2$. The larger value ρ_1 is used at the beginning of the algorithm, so as to accelerate the exploration of the solution space. In a latter phase, i.e. when improving moves can no longer be found, the radius is decreased to the smaller value ρ_2 in order to facilitate convergence to a local optimum.

Let us now consider the case where the feasibility interval $[lb, ub]$ is either empty or very narrow, meaning that x is either infeasible or close to the infeasible region. In this case, we disregard conditions (2.7) and simply set the radius of the ball equal to ρ, thus generating an infeasible solution x'. In order to handle this and other situations where infeasible solutions arise (see e.g. the end of Section 2.4.2), we introduce a penalty term of the form $a \times |violation|^p$ in the objective function for each ceiling or floor constraint, as discussed in Section 2.4.1 and further specified in Table 1. Notice that the penalty approach appears to be suitable here, since limited violations of the floor, ceiling or turnover constraints can usually be tolerated in practice.

Ceiling:	if $x_i > \overline{x}_i$, then penalty $= a(x_i - \overline{x}_i)^p$
Floor:	if $x_i < \underline{x}_i$, then penalty $= a(\underline{x}_i - x_i)^p$

Table 2.1: Penalties for floor and ceiling constraints

2.4.5 Trading constraints

The trading constraints are disjunctive: either the holdings of each asset remain at their current value $x^{(0)}$ or they are modified by a minimum admissible amount. These constraints are difficult to handle, as they disconnect the solution space into 3^n feasible subregions separated by forbidden subsets.

We use a similar approach as for the previous class of constraints. Denote by $x' = x + r\,d$ the neighbor of x obtained as explained in Section 2.4.4, where d is the direction of the move and r is the radius of the ball. If x' satisfies the trading constraints, then there is nothing to be done. Otherwise, we temporarily disregard the floor/ceiling constraints (which are anyway easier to enforce than the trading constraints) and we compute the smallest value t in the interval $[r, \infty)$ such that $x'' = x + t\,d$ satisfies the trading constraints. If t is not too large (i.e., if t does not exceed a predetermined threshold), then we retain x'' as neighbor of x. On the other hand, if t is larger than the threshold, then we reject the current move and we draw three new assets to be modified.

Observe, however, that solutions which violate the trading constraints still arise in some iterations of the algorithm. For instance, the initial solution is usually infeasible, and so are the solutions which are generated when the portfolio contains exactly N assets (see the last paragraph of Section 2.4.3). Such infeasibilities are penalized as described in Table 2 (the parameters a and p are fixed as in Section 2.4.1). Observe that penalties are high at the center of the forbidden zones and decrease in the direction of admissible boundaries (associated with no trading or with minimum sales/purchases). Therefore, starting from a forbidden portfolio, the process tends to favor moves toward feasible regions.

Purchase:	$Qpurchase = x_i - x_i^{(0)}$ if $Qpurchase \in]0, \underline{B}_i[$, then if $Qpurchase \leq \underline{B}_i/2$ then penalty $= a\ Qpurchase^p$ else penalty $= a(\underline{B}_i - Qpurchase)^p$
Sale:	$Qsale = x_i^{(0)} - x_i$ if $Qsale \in]0, \underline{S}_i[$, then if $Qsale \leq \underline{S}_i/2$ then penalty $= a\ Qsale^p$ else penalty $= a(\underline{S}_i - Qsale)^p$

Table 2.2: Penalties for trading constraints

2.4.6 Summary: Neighbor selection

We can summarize as follows the neighbor selection procedure.

Move direction

- If the current portfolio involves $N - k$ assets, with $k \geq 1$, then
  - select three assets, say 1, 2 and 3, at random as explained in Section 2.4.2, while ensuring that at most k of them are outside the current portfolio (see Section 2.4.3);
  - go to Case a.
- If the current portfolio involves N assets, then

  - select three assets, say 1, 2 and 3, at random as explained in Section 2.4.2, while ensuring that at most one of them is outside the current portfolio (see Section 2.4.3);
  - if all three selected assets are in the current portfolio, then go to Case a; else go to Case b.

Step length

Case a.

- Let d be the direction of the move as defined by equations (2.4) (with the sign of *step* fixed at random). Compute the feasible interval for the radius of the move, say $[lb, ub]$, and compute the radius r as explained in Section 2.4.4.
- If $x + r\,d$ satisfies the trading constraint, then
  - $x + r\,d$ is the selected neighbor; if necessary, compute penalties for the violation of the floor and ceiling constraints as in Table 1;
  - else, try to extend the move to $x + t\,d$, as explained in Section 2.4.5; if t is not too large, then
    - $x+t\,d$ is the selected neighbor; if necessary, compute penalties for the violation of the floor and ceiling constraints as in Table 1;
    - else, discard direction d and select a new direction.

Case b.

- Let assets 1 and 2 be in the current portfolio and asset 3 be outside. In equations (2.4), set the parameter *step* equal to x_1, set $x'_1 = 0$ and compute the corresponding values of x'_2 and x'_3.
- If necessary, compute penalties for the violation of the floor, ceiling and trading constraints as in Tables 1 and 2.

2.5 Cooling schedule, stopping criterion and intensification

2.5.1 Cooling schedule and stopping criterion

In our implementation of simulated annealing, we have adopted the geometric cooling schedule defined in Section 2.3. In order to describe more completely this cooling schedule, we need to specify the value of the parameters T_0 (the *initial temperature*), L and α.

Following the recommendations of many authors (see e.g. [1, 35, 59]), we set the initial temperature T_0 in such a way that, during the first cooling stage (first L steps), the probability of acceptation of a move is roughly equal to a predetermined, relatively high value χ_0 (in our numerical tests, $\chi_0 = 0.8$). In order to achieve this goal, we proceed as follows. In a preliminary phase, the SA algorithm is run for L steps without rejecting any moves. The average increase of the objective function over this phase, say Δ, is computed and T_0 is set equal to:

$$T_0 = \frac{-\Delta}{\ln \chi_0} \tag{2.8}$$

(see equation (2.2)).

After L moves, the temperature is decreased according to the scheme $T_{k+1} = \alpha T_k$. We use here the standard value $\alpha = 0.95$.

The fundamental trade-offs involved in the determination of the stage length L are well-known, but difficult to quantify precisely. A large value of L allows to explore the solution space thoroughly, but results in long execution times. Some studies (see [35, 59]) suggest to select a value of L roughly equal to the neighborhood size. In our algorithm, this rule leads to the value $L \approx \begin{pmatrix} n \\ 3 \end{pmatrix}$, which appeared excessively large in our computational tests. Therefore, we eventually settled for values of L of the same order of magnitude as n (e.g., we let $L = 300$ when $n = 150$). We elaborate on this topic in the next section.

In this first, basic version, the algorithm terminates if no moves are accepted during a given number S of successive stages. In our experiments, we used $S = 5$.

2.5.2 Intensification

We have experimented with several ways of improving the quality of the solutions computed by the SA algorithm (at the cost of its running time). In all these attempts, the underlying strategy simply consists in running several times the algorithm described above; in this framework, we call *cycle* each execution of the basic algorithm. The main difference between the various strategies is found in the initialization process of each cycle. Namely, we have tried to favor the exploration of certain regions of the solution space by re-starting different cycles from "promising" solutions encountered in previous cycles. Such *intensification* strategies have proved successful in earlier implementations of local search metaheuristics.

Strategy 1. In this naive strategy, we run several (say M) cycles successively and independently of each other, always from the same initial solution. The random nature of each cycle implies that this strategy may perform better than the single-start version.

Strategy 2. This second strategy is very similar to strategy 1, except that each cycle uses, as initial solution, the best solution found within the previous cycle. For all but the first cycle, the initial temperature is defined by formula (2.8) with $\chi_0 = 0.3$ (the idea is to set the initial temperature relatively low, so as to preserve the features of the start solution). Also, for all but the last cycle, the stopping criterion is slightly relaxed: namely, each intermediate cycle terminates after S' stages without accepted moves, where $S' < S$. In our experiments, $S' = 2$.

Strategy 3. In the first cycle, we constitute a list P of promising solutions, where a promising solution may be

a. either the best solution found during each stage,

b. or the best solution found in any stage where the objective function has dropped significantly (a drop is significant if it exceeds the average decrease of the objective function during the previous stage).

Next, we perform $|P|$ additional cycles, where each cycle starts from one of the solutions in P. For these additional cycles, the initial temperature is computed with $\chi_0 = 0.3$ and the cycle terminates after $S' = 2$ stages without accepted moves.

In the next section, we will compare the results produced by the basic strategy (strategy 0), strategy 1, strategy 2 and strategy 3b. In order to allow meaningful comparisons between the three multi-start strategies, we restrict the number of cycles performed by strategy 3b by setting an upper-bound $(M - 1)$ on the size of the list P. That is, after completion of the first cycle, we discard solutions from P by applying the following two rules in succession:

- if several solutions in P imply the same trades (i.e., all these solutions recommend to buy or to sell *exactly* the same securities), then we only keep one of these solutions; the rationale is here that our algorithm is rather good at finding the best solution complying with any given trading rules, so that all of these solutions yield 'equivalent' starting points;
- we only keep the best $(M - 1)$ solutions in P.

2.6 Computational experiments

2.6.1 Environment and data

The algorithms described above have been implemented in standard C (ANSI C) and run on a PC Pentium 450MHz under Windows 98. A graphical interface was developed with Borland C++ Builder. All computation times mentioned in coming sections are approximate real times, not CPU times. Unless otherwise stated, the parameter settings for the basic SA algorithm are defined as follows:

- stage size: $L = 2n$;
- stopping criterion: terminate when no moves are accepted for $S = 5$ consecutive stages;
- ball radius: $\rho_1 = 0.005$, decreased to $\rho_2 = 0.001$ as soon as fewer than 10 steps are accepted during a whole stage.

For the sake of constructing realistic problem instances, we have used financial data extracted from the DataStream database. We have retrieved the weekly prices of $n = 151$ US stocks covering different traditional sectors for 484 weeks, from January 6, 1988 to April 9, 1997, in order to estimate their mean returns and covariance matrix. The stocks were drawn at random from a subpopulation involving mostly major stocks. (Note that our goal was not to draw any conclusions regarding the firms, or the stock market, or even the composition of optimal portfolios, but only to test the computational performance of the algorithms.) These data have been used to generate several instances of model (PS) involving different subsets of constraints.

For each instance, we have approximately computed the mean-variance frontier by letting the expected portfolio return ($Rexp$) vary from -0.3688% to 0.737% by steps of 0.01% (110 portfolios). Linear interpolation is used to graph intermediate values. In each graph, the ordinate represents the expected portfolio return (expressed in basis points) and the abscissa represents the standard deviation of return.

We now discuss different instances in increasing order of complexity.

2.6.2 The Markowitz mean-variance model

As a first base case, we have used the simulated annealing (SA) algorithm to solve instances of the Markowitz mean-variance model (see Section 2.2.2) without nonnegativity constraints. Since these instances can easily be solved to optimality by Lagrangian techniques, we are

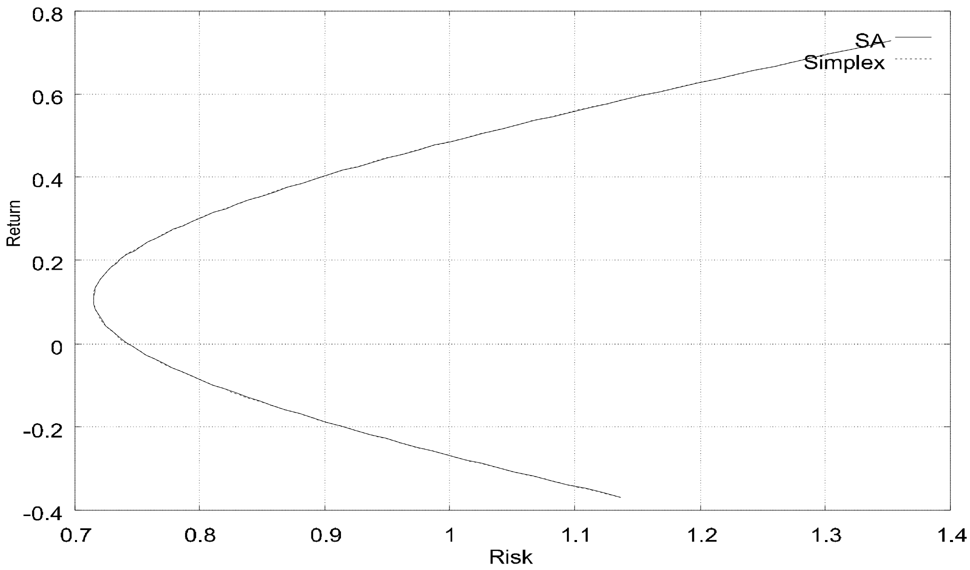


Figure 2.1: Mean-variance frontier with short sales

able to check the quality of the solutions obtained by the SA algorithm. Our algorithm finds the exact optimal risk for all values of the expected return. The SA algorithm requires 2 or 3 seconds per portfolio of 151 securities with the standard parameter settings.

The mean-risk frontier for this instance is plotted in Figure 2.1. It will also be displayed in all subsequent figures, in order to provide a comparison with the frontiers obtained for constrained problems.

For a particular value of the target return, Figure 2 illustrates the evolution of the portfolio variance in the course of iterations.

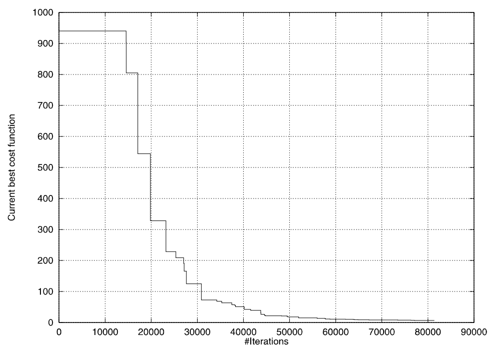


Figure 2: Evolution of the portfolio variance

2.6.3 Floor, ceiling and turnover constraints

We solved several instances involving floor, ceiling and turnover constraints. The first instance (Figure 3) imposes nonnegativity constraints on all assets (no short sales). The second

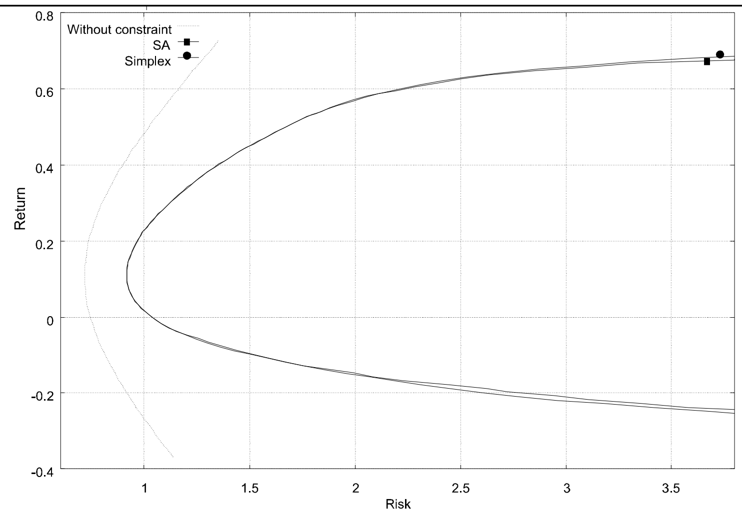


Figure 3

Stop if no accepted move for $5L$ iterations

#equities n: 151

#portfolios: 110

$L = 2n$

$\underline{x}_i = 0 \quad \forall i$

$\overline{x}_i = 1 \quad \forall i$

$\rho_1 = 0.005 \qquad \rho_2 = 0.001$

$\rho_1 \to \rho_2$: ≤ 10 moves

Time < 4"/portfolio

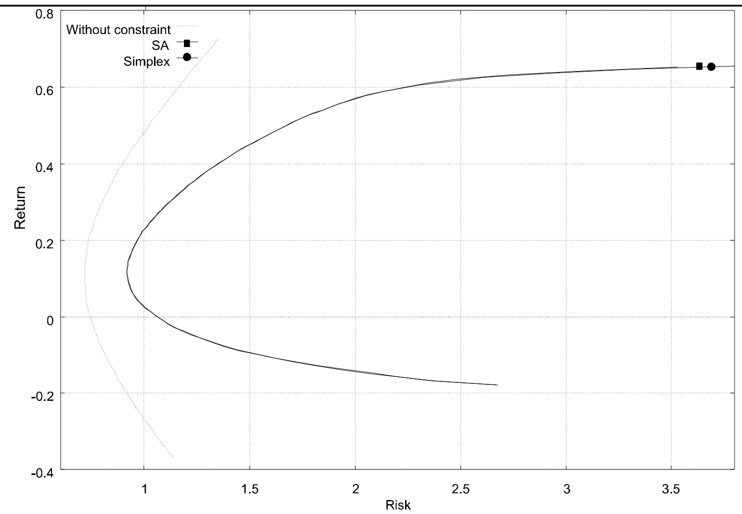


Figure 4

Stop if no accepted move for $5L$ iterations

#equities n: 151

#portfolios: 110

$L = 2n$

$\underline{x}_i = 0 \quad \forall i$

$\overline{x}_i = 0.2 \quad \forall i$

$\rho_1 = 0.005 \qquad \rho_2 = 0.001$

$\rho_1 \to \rho_2$: ≤ 10 moves

Time < 3"/portfolio

one (Figure 4) adds more restrictions on minimal and maximal holdings allowed: $\underline{x}_i = 0$ and $\overline{x}_i = 0.2$ for each security. The hypotheses and results are more completely described next to each figure.

Here again, the exact optimal solution can be computed efficiently (e.g., using Wolfe's quadratic simplex algorithm [70]) and can be used to validate the results delivered by the SA algorithm. The quality of the heuristic solutions is usually extremely good. Slight deviations from optimality are only observed for extreme portfolio returns. Moreover, the solutions always satisfy all the constraints (penalties vanish). Run times are short and competitive with those of the quadratic simplex method (less than 4 seconds per portfolio).

2.6.4 Trading constraints

When the model only involves floor, ceiling and turnover constraints, the mean-variance frontiers are smooth curves. When we introduce trading constraints, however, sharp discontinuities may arise. This is vividly illustrated by Figure 5: here, we have selected three securities and we have plotted all (mean return,risk)-pairs corresponding to feasible portfolios of these three securities. Observe that disconnected regions appear. (Similar observations are made by Chang et al. in [8].) Figure 6 shows the outcome provided by the simulated annealing algorithm: notice that the algorithm perfectly computes the mean-variance frontier for this small example.

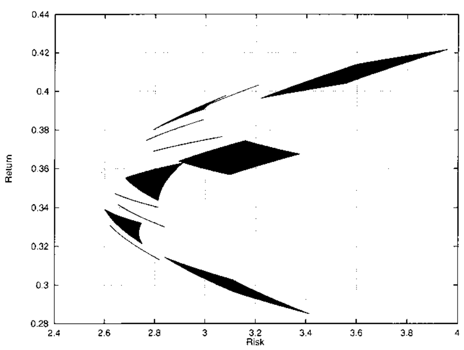


Figure 5: all portfolios

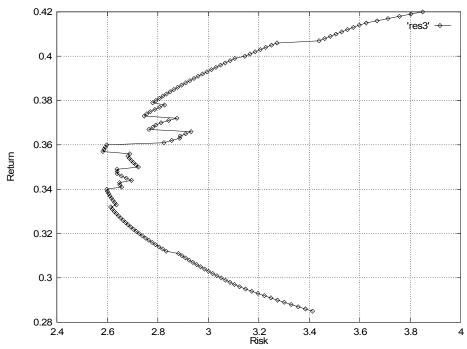


Figure 6: SA frontier

#equities: 3
$x_i^{(0)} = 1/n$
$\underline{B}_i = 0.1$
$\underline{S}_i = 0.2$
$\underline{x}_i = 0$

When the number of securities increases, the optimization problem becomes extremely difficult to solve. Figure 7 illustrate the results produced by the basic SA algorithm for the whole set of 151 securities, with trading constraints defined as follows:

- $\underline{B}_i = \underline{S}_i = 0.05$ $(i = 1, \ldots, n)$;
- the initial portfolio $x^{(0)}$ is the best portfolio of 20 stocks with an expected return of 0.24% (see Section 2.6.5 hereunder).

The computation times remain reasonable (about 10 seconds per portfolio). However, as expected, the frontier is not as smooth as in the simpler cases. The question is to know whether we succeeded in computing the actual frontier or whether the SA algorithm erred in this complex case. The simplex method cannot be used anymore to compute the optimal solutions, because of the mixed integer constraints. Therefore, we have carried out some additional experiments in order to better assess the performance of our algorithm.

First, we have used the commercial package LINGO in order to model and to solve a small instance of the problem. Indeed, LINGO allows to handle nonlinear programming problems involving both continuous and binary variables and to solve such problems to optimality by

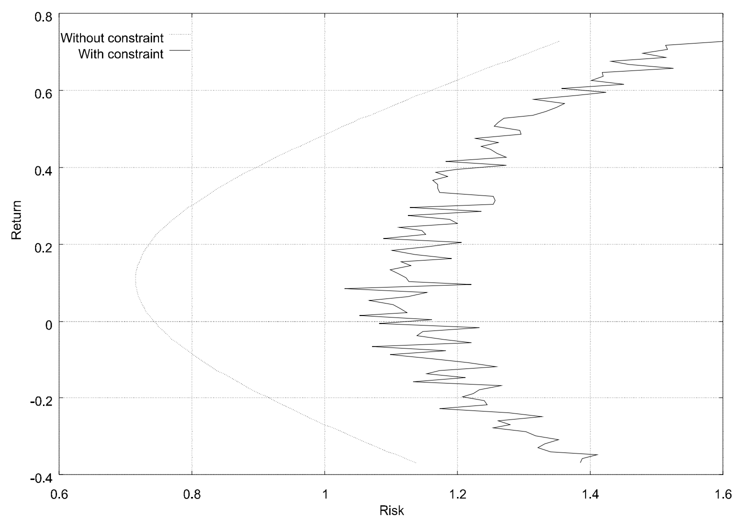


Figure 7
Stop if no accepted move for $5L$ iterations
#equities n: 151
#portfolios: 110
$L = 2n$
$x_i^{(0)}|N = 20$
$\underline{B}_i = 0.05 \ \ \forall i$
$\underline{S}_i = 0.05 \ \ \forall i$
$\rho_1 = 0.005 \quad \rho_2 = 0.001$
$\rho_1 \to \rho_2$: ≤ 10 moves
Time: 10"/portfolio

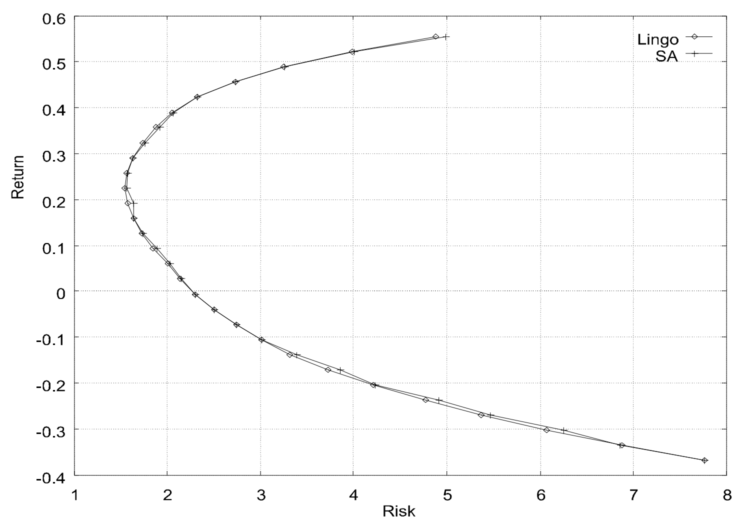


Figure 8
Stop if no accepted move for $5L$ iterations
#equities n: 30
#portfolios: 30
$L = 3n$
$\underline{x}_i = 0 \ \ \forall i$
$\overline{x}_i = 1 \ \ \forall i$
$x_i^{(0)}|N = 5$
$\underline{B}_i = 0.1 \ \ \forall i$
$\rho_1 = 0.005 \quad \rho_2 = 0.001$
$\rho_1 \to \rho_2$: ≤ 10 moves
Time < 0.5"/portfolio

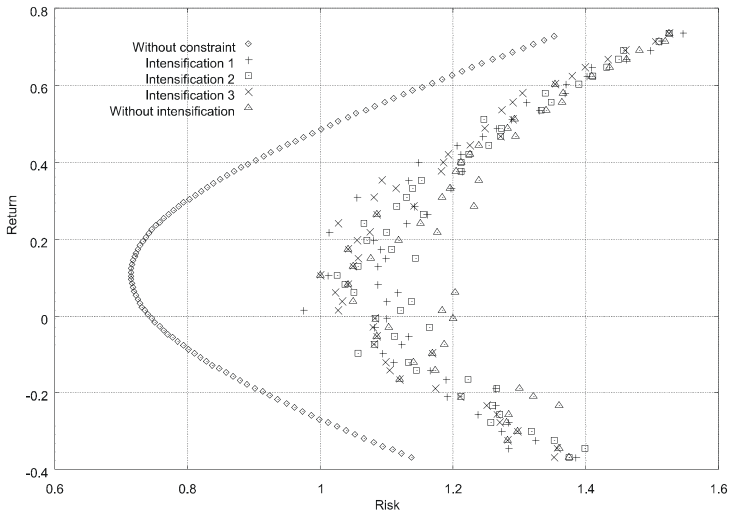


Figure 9
Stop if no accepted move for $5L$ iterations
#equities n: 151
#portfolios: 50
$L = 2n$
$x_i^{(0)} | N = 20$
$\underline{B}_i = 0.05 \quad \forall i$
$\underline{S}_i = 0.05 \quad \forall i$
$\rho_1 = 0.005 \qquad \rho_2 = 0.001$
$\rho_1 \to \rho_2$: ≤ 10 moves
Time
Intens1: 35"/portfolio
Intens2: 34"/portfolio
Intens3: 49"/portfolio

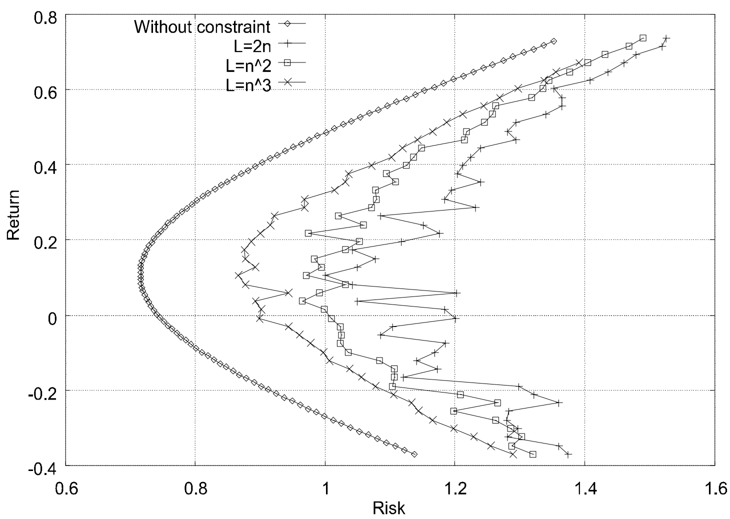


Figure 10
$L = 2n$ or n^2 or n^3
Stop if no accepted move:
$5L$ iterat. when $L = 2n$,
$3nL$ iterat. when $L = n^2$,
$2L$ iterat. when $L = n^3$.
#equities n: 151
#portfolios: 50
$x_i^{(0)} | N = 20$
$\underline{B}_i = 0.05 \quad \forall i$
$\underline{S}_i = 0.05 \quad \forall i$
$\rho_1 = 0.005 \qquad \rho_2 = 0.001$
$\rho_1 \to \rho_2$: ≤ 10 moves
$L = 2n$: 10"/portfolio
$L = n^2$: 23'42"/portfolio
$L = n^3$: 7h58'/portfolio

branch-and-bound. Computation times, however, increase sharply with the size of problem instances. We have therefore restricted the set of underlying assets to 30 equities, with $N = 5$. A visual comparison between the results obtained by the SA algorithm and by LINGO is provided in Figure 8. We observe that the SA algorithm obtains near optimal solutions for all target returns, within short computational times.

For another test, we have run new experiments on the full data set of 151 equities, using now the three intensification strategies described in the previous section. Globally, intensification tends to improve the results obtained by the basic algorithm (see Figure 9). However, there is no clear dominance between the three strategies tested. This is rather disappointing, if one remembers that strategy 1 simply consists in running several times the SA algorithm from the same initial solution. This observation suggests that increased running time, allowing for more exploration of the solution space, may be the key element in improving the performance of the SA algorithm. (Notice that similar conclusions have been drawn by other authors working with simulated annealing algorithms; see e.g. [1] or [35]). On the other hand, restarting the process from promising solutions does not appear to help much (probably because the features of these solutions are lost in the high-temperature phase of the SA algorithm).

In order to confirm these tentative conclusions, we have run again the basic SA algorithm on the same instances without intensification, but with much larger values of the stage length L, i.e. with $L = n^2$ and $L = n^3$ (see Section 2.5.1). The stopping criterion is adapted for $L = n^2$ to make sure that is it at least as strict as for $L = n^3$. In this way, we ensure that the improvement obtained for $L = n^3$ is due to the increase of L and not to the stopping criterion. The results of this experiment are displayed in Figure 10. On the average, over the whole range of target values, the standard deviation of the portfolio improves by 7% when $L = n^2$ and by 13% when $L = n^3$. The largest improvements are attained for intermediate values of the target return. It should be mentioned, however, that such improvements come at the expense of extremely long running times (about 8 hours per portfolio when $L = n^3$).

2.6.5 Maximum number of securities

Let us now consider a cardinality constraint limiting the number of assets to be included in the portfolio. Figure 11 displays the results obtained with the basic SA algorithm when we only allow $N = 20$ assets in the portfolio (without any other constraints in the model, besides the return and budget constraints).

In spite of the combinatorial nature of the cardinality restriction, the computation of the

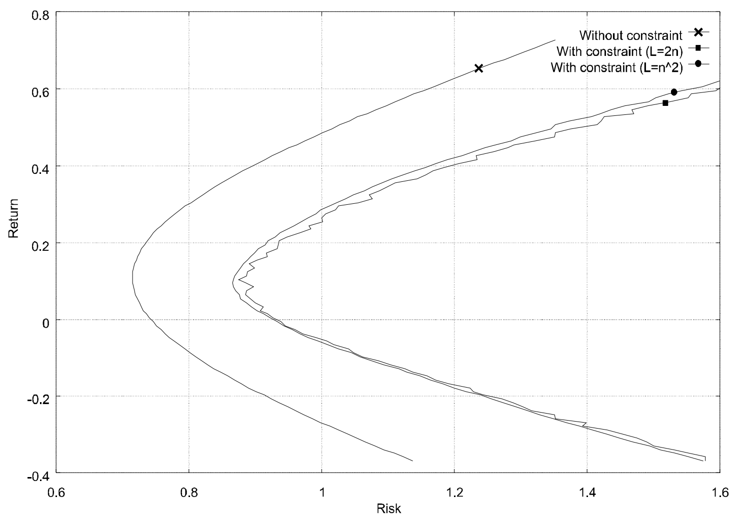


Figure 11
Stop if no accepted move for $5L$ iterations
#equities n: 151
#portfolios: 110
$L = 2n$ or n^2
$N = 20$
$\rho_1 = 0.005 \quad \rho_2 = 0.001$
$\rho_1 \rightarrow \rho_2$: ≤ 10 moves
Time:
$L = 2n$: $<$ 4"/portfolio
$L = n^2$: 4'44"/portfolio

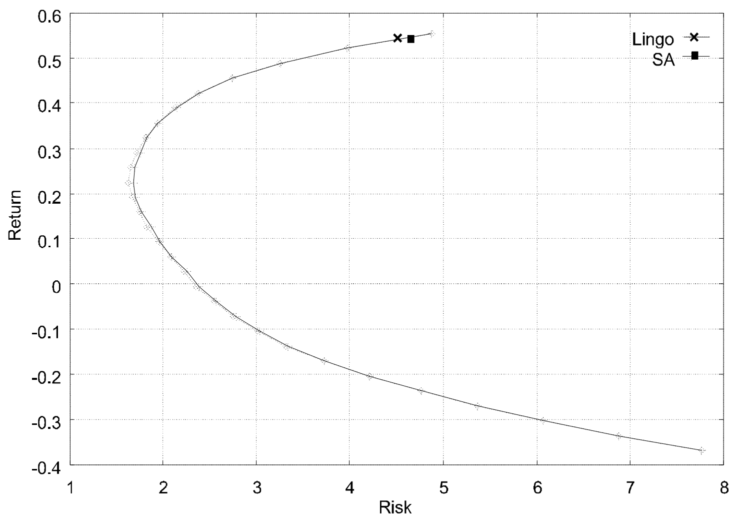


Figure 12
Stop if no accepted move for $5L$ iterations
#equities n: 30
#portfolios: 30
$L = 2n$
$\underline{x}_i = 0 \quad \forall i$
$\overline{x}_i = 1 \quad \forall i$
$N = 20$
$\rho_1 = 0.005 \quad \rho_2 = 0.001$
$\rho_1 \rightarrow \rho_2$: ≤ 10 moves
Time: $<$ 0.15"/portfolio

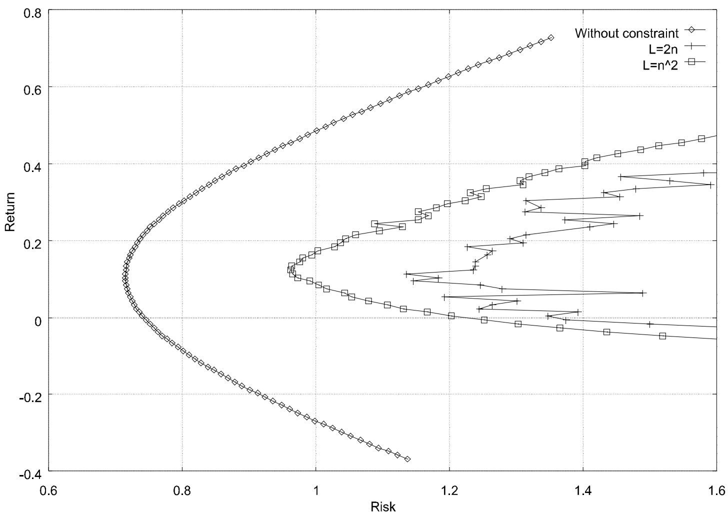


Figure 13
Stop if no accepted move for $5L$ iterations
#equities n: 151
#portfolios: 110
$L = 2n$ or n^2
$\underline{x}_i = 0 \quad \forall i$
$\overline{x}_i = 1 \quad \forall i$
$x_i^{(0)} | N = 20$
$\underline{B}_i = 0.05 \quad \forall i$
$\underline{S}_i = 0.05 \quad \forall i$
$N = 20$
$\rho_1 = 0.005 \qquad \rho_2 = 0.001$
$\rho_1 \to \rho_2$: ≤ 10 moves
Time:
$L = 2n$: $<$ 2"/portfolio
$L = n^2$: 3'33"/portfolio

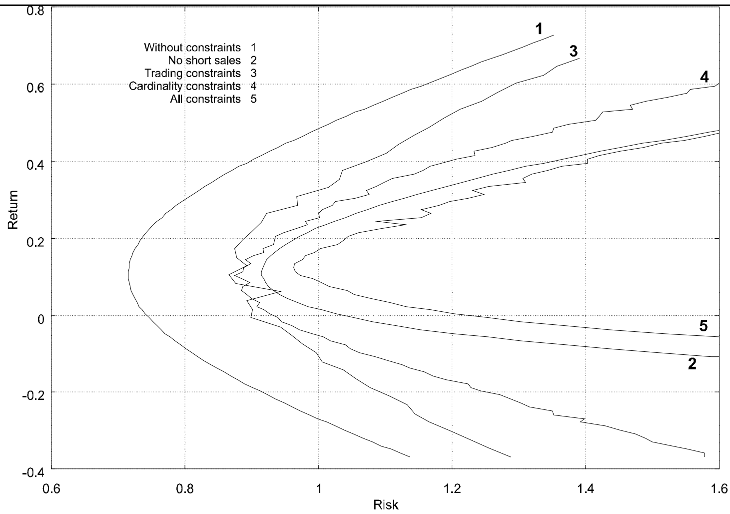


Figure 14:
Figure 2.1 (Without constraint)
+ Figure 3 (Ceiling)
+ Figure 10 $L = n^3$ (Trading)
+ Figure 11 $L = n^2$ (Cardinality)
+ Figure 13 $L = n^2$ (All)

mean-variance frontier is rather efficient for this problem. The solutions obtained by the SA algorithm are always feasible (i.e., no penalties remain when the algorithm terminates). Moreover, the graph in Figure 11 for $L = 2n$ is very smooth: this suggests that the SA algorithm may have achieved near-optimal solutions for all values of the target returns. In order to validate this hypothesis, we performed some experiments with larger stage lengths ($L = n^2$), and were only able to record minor improvements. We also ran LINGO on a subsample of 30 assets, with $N = 5$; for this smaller instance, the SA algorithm perfectly computed the whole mean-variance frontier (Figure 12).

2.6.6 Complete model

Investigating each class of constraint separately was important in order to understand the behavior of the algorithm, but our final aim was to develop an approach that could handle more realistic situations where all the constraints are simultaneously imposed. Figure 13 illustrates the results obtained by the basic SA algorithm with $L = 2n$ or $L = n^2$ for such a complex instance. Observe that, here again, the results obtained when $L = n^2$ are significantly better than when $L = 2n$. Even with the higher value of L, however, the computation time remains reasonably low.

Figure 14 sums up all the previous results. It illustrates the effect of each class of constraints on the problem and allows some comparison of the mean-variance frontiers computed in each case.

2.7 Conclusions

Portfolio selection gives rise to difficult optimization problems when realistic side-constraints are added to the fundamental Markowitz model. Exact optimization algorithms cannot deal efficiently with such complex models. It seems reasonable, therefore, to investigate the performance of heuristic approaches in this framework.

Simulated annealing is a powerful tool for the solution of many optimization problems. Its main advantages over other local search methods are its flexibility and its ability to approach global optimality. Most applications of the SA metaheuristic, however, are to combinatorial optimization problems. In particular, its applicability to portfolio selection problems is not fully understood, yet. The main objective of this work was therefore to investigate the adequacy of simulated annealing for the solution of a difficult portfolio optimization model.

As SA is a metaheuristic, there are quite a lot of choices to make in order to turn it into an actual algorithm. We have developed an original way to generate neighbors of a current solution. We have also proposed specific approaches to deal with each specific class of constraint, either by explicitly restricting the portfolios to remain in the feasible region or by penalizing infeasible portfolios.

Let us now try to draw some conclusions from this research. On the positive side, we can say that the research was successful, in the sense that the resulting algorithm allowed us to approximate the mean-variance frontier for medium-size problems within acceptable computing times. The algorithm is able to handle more classes of constraints than most other approaches found in the literature. Although there is a clear trade-off between the quality of the solutions and the time required to compute them, the algorithm can be said to be quite versatile since it does not rely on any restrictive properties of the model. For instance, the algorithm does not assume any underlying factor model for the generation of the covariance matrix. Also, the objective function could conceivably be replaced by any other measure of risk (semi-variance or functions of higher moments) without requiring any modification of the algorithm. This is to be contrasted with the algorithms of Perold [58] or Bienstock [2], which explicitly exploit the fact that the objective function is quadratic and that the covariance matrix is of low rank.

On the negative side, it must be noticed that the tailoring work required to account for different classes of constraints and to fine-tune the parameters of the algorithm was rather delicate. The trading constraints, in particular, are especially difficult to handle because of the discontinuities they introduce in the space of feasible portfolios. Introducing additional classes of constraints or new features in the model (e.g., transaction costs) would certainly prove quite difficult again.

PART TWO:

Optimization of a portfolio of options under Value-At-Risk constraints: a scenario approach

Chapter 3

Introduction to Part Two

The starting point of the work presented in this second part of the thesis is a portfolio optimization model proposed by Gielen [29] in 1998. This author summarized the objective of her work as developing "a systematic method for composing portfolios that best meet the investor's specific risk-return preferences. The portfolio can include stock on the AEX, associated put and call options and cash".

Similarly, our aim is to develop a systematic framework, based on operations research models and methods, for helping an investor to construct a portfolio of options. With this aim, we introduce a new multiperiod model for the optimization of a portfolio of options linked to a single financial index. The objective of the model is to maximize the expected return of the portfolio under constraints limiting its Value-at-Risk. The future is flexibly modelled through a multiperiod scenario approach.

It is a very common approach, in Operations Research, to concentrate on the mathematical structure of an optimization model (characterization of the set of feasible solutions, properties of the optimal solutions, tailoring of local search metaheuristics, ...), without paying much attention to the numerical values assumed by the parameters which define a specific instance of the problem. As an extreme example of this trend, many optimization algorithms are tested on randomly generated problem instances. In such cases, it is implicitly assumed that all values of the parameters, within a loosely defined domain, give rise to a meaningful instance of the problem (for instance, the coefficients appearing in the objective function and in the constraints of a generic linear programming problem are essentially unrestricted; the distances defining an instance of the travelling salesman problems are only required to be nonnegative; etc.). Even when the instances are not random but arise from some real-world application, it is usually the case that the models under consideration are sufficiently robust to remain meaningful if small perturbations are applied to their numerical parameters. This is why, in Chapter 2, we pretty much disregarded the issue of estimating the financial parameters (expected returns, covariance matrix, etc.) in the generalized Markowitz' model: if an independent financial analyst gives us the values of these parameters, then the SA algorithm that we have developed will usually deliver a heuristic solution of the portfolio selection problem. (This is not to say that the efficiency of the algorithm, or the quality of the solution, are not affected by the value of the parameters, but only that the algorithm will yield meaningful results on most problem instances.)

When working on Gielen' problem [29], however, it soon became apparent that solving the original optimization model would be utterly meaningless, unless we could guarantee that the data sets were financially realistic and internally consistent. For instance, carelessly generated option prices would lead, almost inevitably, to arbitrage opportunities and/or to unrealistic (infinite) expected returns. Therefore, we moved rapidly away from our initial, pure optimization perspective, to focus on a broader modelling challenge: the goal was no longer restricted to solving the VaR portfolio optimization model, but also to model realistically the financial data required by this model (and, as a matter of fact, by other models requiring the same type of numerical data). This ultimately lead us to an enriched model containing several interesting features, like the possibility to rebalance the portfolio with options introduced at any intermediate period, explicit consideration of transaction costs and of option bid-ask spreads, advanced schemes to model future index return distributions, realistic pricing and construction of options, etc.

As we will see in subsequent chapters, developing such a model requires to master a number of advanced financial concepts and to translate these theoretical concepts into operational ones. This raises a variety of problems of a financial, statistical and numerical nature, which will be described in subsequent chapters.

Finally, the financial perspective was also helpful in analyzing the theoretical properties of the model and in developing optimization approaches based on these properties.

The remainder of the thesis is organized as follows. Chapter 4 introduces some of the basic financial concepts that will be used throughout the dissertation. In Chapter 5, we introduce models and methods to represent the future, and more specifically the future index returns. Based on these models, we construct new option pricing models in Chapter 6. We then consider the VaR portfolio optimization problem itself in Chapters 7-10. Chapter 7 states the model. In Chapter 8, we examine some results from the financial literature that appear relevant for the solution of the VaR problem. In Chapter 9, we develop exact and heuristic optimization methods, based on operations research techniques and on financial theory, to solve the VaR problem. Finally, in Chapter 10, we present computational results obtained with various optimization algorithms on real data sets and under different hypotheses. Let us now take a deeper look at each of the chapters.

Chapter 4 briefly presents some basic financial concepts. Its aim is to sketch the elementary theoretical background, for those unfamiliar with these concepts. The material in this chapter comes essentially from textbooks like Hull [32], Lynch [49], McMillan [51], Gillet and Minguet [52], or Duffie [24]. This list is far from exhaustive, and lots of other references can be found in the literature (e.g. see [4, 9, 12, 21, 26, 48, 55, 68]). In particular, in this chapter, we first define the main characteristics of the financial securities we want to consider, i.e. stocks, portfolio of stocks, indices, options and risk-free investment (Section 4.2). We explain how options are typically priced (Section 4.5), and, more generally, we state a no-arbitrage relation that the security prices should fulfill. (Section 4.4). This no-arbitrage relation is a key concept that will be used several times in subsequent chapters.

The objective of the portfolio model is to maximize the expected value of the portfolio over a given horizon, under budget, guarantee and Value-at-Risk constraints. In particular, as the value of the portfolio depends on the value of the securities it contains, this implies that we need to be able to predict the possible values of each security at the end of the horizon. Moreover, we want to consider a dynamic portfolio problem in which the investor can adjust his portfolio at an intermediate date. Therefore, we also need to model the security prices at this time. In order to achieve these goals, we first present a two-period scenario tree model in Section 5.1 of Chapter 5. Each node of a scenario tree corresponds to a possible state of the world at the corresponding date. Such trees provide very generic models often used to represent the future in stochastic optimization problems, although we will only consider them here in a simple form (see e.g. Birge and Louveaux [3] or Prekopa [60] for a broad introduction to stochastic programming). In finance, especially, trees of scenarios have been used in numerous applied and theoretical models; see e.g. Dembo [16, 18], Dert [19], Dybvig [22, 23], Koskosidis and Duarte [41], Mulvey [53] or Prekopa [60]. Note also that the binomial methods, which are intensively used in finance (e.g. to price the options), rely on special types of scenario trees. Numerous presentations of binomial trees can be found in the references cited above; let us also add here a reference to the famous implied binomial tree method proposed by Rubinstein [63].

Obviously, a tree of scenarios becomes useful only when we are able to characterize the states of the world at each node. A classical simplifying hypothesis in finance consists in assuming that stock and index returns follow a Normal probability distribution. We could make this assumption to define the index prices at each leaf of the tree. However, we observed in our numerical experiments that the normality hypothesis leads to abnormal portoflio returns when we use it in conjunction with the option prices observed on the market. Moreover, the optimal portfolio returns are also very sensitive to slight perturbations in the value of some key parameters like the risk-free rate or the index dividend yield. Therefore, in Sections 5.2-5.3 of Chapter 5, we propose several methods to compute representative probability distributions of index returns and parameters. The models developed here are based on statistical distributions proposed by Theodossiou [66], Fernandez and Steel [28], Lambert and Laurent [42], Breeden and Litzenberger [5], Shimko [64] and Rubinstein [63]. In Section 5.4, the continuous probability density functions are sampled to obtain a set of discrete values that can be associated to the leaves of the tree of scenarios. Different methods to perform the sampling are considered [13, 32, 38, 39, 61]. In particular, the stratification approach allows to construct samples that represent faithfully the corresponding continuous distribution even for small sample sizes. Finally, in Section 5.5, we return to a discussion of probability distributions. Indeed, in finance, we are interested in two families of distributions: the consensus distributions and the risk-neutral distributions. The first ones represent the index returns in the "real" world as viewed by the investors. The latter ones correspond to a virtual world where the investors are risk neutral. This is a key financial concept described in Chapter 4. The consensus distributions are required in order to construct the tree of scenarios which underlies our model of the VaR problem, and the risk-neutral distributions are useful in order to price the options and to develop optimization heuristics. As the probability density functions defined in Section 5.3 belong either to the first world or to the latter one, we need tools which convert each distribution into a distribution of the other type. In order to perform the conversion, we develop an operational version of some of Rubinstein's results [63].

In Chapter 5, we have defined a tree of scenarios model, together with methods that allow to instantiate (i.e., to label) the tree with index values. However, we also want to work with options. Therefore, we need to define the value of each option at each node of the tree of scenarios. Besides these values, we also would like to model some of the market rules used to create options. Indeed, we consider explicitly a portfolio problem in which the investor will be able, at some date in the future, to adjust his portfolio by including some of the options that will become available at that time. This set of options varies according to time and scenario, and cannot, by definition, be observed initially. Therefore, in Section 6.2 of Chapter 6, we first review some of the characteristics of the options traded on real markets. Then, in Section 6.3, we propose new models which can be used to price the options within our framework. Classical approaches, like binomial methods or the Black and Scholes formulae, cannot be used here since the hypotheses supporting these models are not satisfied within a multinomial tree of scenarios. Therefore, the prices obtained by those methods usually lead to arbitrage opportunities, which are not acceptable when setting up a portfolio optimization model. So, we resort to a new model based on the no-arbitrage system of equations, as stated by Duffie [24]. Moreover, some improvements to this new model are considered. First, we try to minimize the deviation between the observed prices (or any other target prices) and the arbitrage-free prices computed from our model. In this process, we explicitly take into account the bid and ask prices of each option instead of a unique central price as in typical pricing formulae. Second, the model is modified so as to obtain, after optimization, valid risk-neutral probabilities for each leaf of the scenario tree. As mentioned above, these state-prices will be used in heuristics developed to solve the portfolio problem.

In Section 6.4, we develop a simulated annealing heuristic to solve this (nonlinear) option pricing model. Finally, in Section 6.5, we propose several models to define, for each scenario at the rebalancing date, a prior guess of the option prices. Indeed, such a prior guess is required by the option pricing model. We could simply use the Black and Scholes value as estimate, but we have also defined more advanced methods. The first approach is based on Shimko's version [64] of the Black and Scholes formula, which is modified to take into account the observed volatility smile. Alternative approaches use the risk-neutral probabilities (or more precisely the state-prices) computed from a subset of options.

In Chapter 7, we turn to the Value-at-Risk portfolio optimization problem itself. The model proposed in this chapter is inspired from Gielen's model [29]. It is also related to a model described by Dert and Oldenkamp [20], with the difference that the latter model is not based on a scenario approach and considers only one period.

The model imposes a minimal guaranteed return on investment at the end of the horizon. We describe two possible formulations of this guarantee constraint. The first formulation is based on a scenario approach, as in Gielen [29], and the second one on a strike-prices approach, as in Dert and Oldenkamp [20]. In order to be as realistic as possible, the model also integrates all the features mentioned in previous sections: a two-period tree of scenarios model allowing for dynamic portfolio rebalancing strategies, various probability density distributions to model the index returns, option pricing models based on the scenario tree, bid and ask option prices, and transaction costs.

But of course, the core of the model is the Value-at-Risk constraint. This constraint expresses that the return on the initial investment must reach a predefined level (say, at least 5%) with a predefined probability (say, at least 95%). In finance, Morgan popularized the VaR concept as a relevant measure of risk when he introduced it in the RiskMetrics^{TM} system [62], but similar constraints have also been used in the stochastic programming literature for several decades (see e.g. Prekopa [60]). Technical documents and working papers about VaR can be found on the RiskMetrics^{TM} Internet site. Another presentation of this measure can be found in Esch, Kieffer and Lopez [27]. From the Operations Research point of view, modelling VaR constraints within the scenario tree model is challenging, as it requires to introduce binary variables in the optimization model. Therefore, we end up with a mixed integer programming (MIP) problem, which is typically much more difficult to solve than a linear continuous problem of comparable size.

As observed by Dert and Oldenkamp [20] in their one-period continuous model, portfolios subject to a VaR constraint often have a typical structure. Therefore, we expect that, using some intrinsic financial properties of the VaR model, we could improve the efficiency of the optimization process. In Chapter 8, we have explored the possibility to detect a priori (that is, without solving the optimization model) for which scenarios the VaR lower bound will be satisfied in the optimal portfolio, i.e. which scenarios will achieve the highest payoffs. If we could efficiently identify these scenarios, then the formulation of the portfolio optimization model would be greatly simplified, as the MIP model would actually boil down to a linear programming problem. This idea is developed in Section 8.3, after we have described some properties of the optimal portfolio structure in Section 8.2. Next, two financial approaches are examined to identify the largest portfolio values.

In the strategy approach (Section 8.3), we consider four possible investor's behaviors: bullish, bearish, volatility, stability, and we study their impact on the portfolio distibution. Second, in Dybvig's approach (Section 8.5), we attemtp to exploit a relation established by Dybvig [22, 23] between state-prices and optimal consumption patterns (or portfolio distribution in our model). As the hypotheses underlying Dybvig's theorem are not always satisfied in our model, we examine the consequences of the violations and we propose some possible adjustments on the inputs of the VaR problem in order to reduce their effects (note that theoretical extensions of Dybvig's framework have also been investigated in the literature; see e.g. Jouini and Kallal [36]). Algorithmic implementations of these ideas will be presented in Chapter 9.

In Chapter 9, we propose an array of algorithmic approaches for the solution of the VaR portfolio optimization problem. Section 9.2 briefly describes the branch and bound (BB) approach, a classical method used in Operations Research for solving MIP problems (see e.g. Nemhauser and Wolsey [56], Winston [69]). Branch and bound can be used in particular to attack the initial VaR model presented in Chapter 7, but requires very much computing time to obtain the optimal solution. Therefore, in Section 9.3, we develop some new heuristics, with the aim to compute rapidly a good feasible solution of the VaR problem, i.e. a "near optimal" portfolio. In order to define these heuristics, we cast the two financial approaches presented in Chapter 8 into equivalent mathematical programming formulations. We also develop two additional approaches based on the continuous relaxation of the MIP model and on simple rounding techniques. Finally, Section 9.4 proposes some automated procedures to construct sets of options which are sufficiently realistic with respect to typical market rules. One such method attempts to "preselect" a subset of promising options, that is options which are likely to appear in an optimal portfolio. Applying this method allows to reduce the size of the portfolio optimization problem, and hence to speed up the optimization process.

Chapter 10 presents our experimental results. We have developed a C++ software which handles all the models mentioned above. The software contains implementations of several original procedures, and of some procedures from the textbook *Numerical Recipes* [61]. It also calls the simplex and the BB procedures implemented in the CPlex optimization library [33]. We have made numerous tests and numerical experiments using this software. The experiments considered the construction of a portfolio of options linked to the S&P500 index. They allowed us to examine the relevance of our models, to compare the efficiency and the effectiveness of the solution algorithms presented in Chapter 9, and to analyze the impact of various parameter settings.

Chapter 4

Financial concepts

4.1 Introduction

This chapter introduces some financial topics used in the rest of this work for readers unfamiliar with them. The specialists in finance can skip this part without remorse. This chapter does not claim to be a complete course of finance, and only those matters required in what follows will be quickly and simply covered. More will be said in the following chapters when necessary or can be found in the specialised literature ([32, 51, 49, 52]).

4.2 Financial securities

4.2.1 Stocks

Common stocks, which represent the equity of a company, are the basic corporate securities traded on financial markets. The price of a stock reflects the value of the company as estimated by the market. Investing in a stock is risky. We cannot predict with certainty how will evoluate the price in the future and some stocks are more risky than others. Usually, the larger the expected return in the future, the larger the risk because most investors are risk-averse; there is a positive relationship between risk and expected returns. A classical approach to measure the risk is to compute time volatility of the returns and to associate it with mean value.

4.2.2 Portfolio of stocks

As prices of the different stocks are not perfectly correlated, composing a stock porfolio by diversifying investments leads to a reduction of total risk. This means that shifts in price of some stocks can compensate shifts in the other direction of other stocks. The investor has to decide what stocks to include in his portfolio and in what proportions to maximize expected return and to minimize risk. This is called a portfolio selection problem. The expected return of a portfolio is obtained by weighing the sum of expected returns of the components by the proportion of each. The portfolio risk is not simply the weighted sum of the underlying risks. It also takes into account the correlation between all the stocks and is obtained by weighing the covariance matrix of the stock returns.

4.2.3 Indices

Stocks trade on different markets and in different activity sectors. To assess the quality of a portfolio and to try to optimize the stock selection, the investor generally compares his portfolio with a benchmark that has a similar risk. A typical objective is to try to beat this benchmark. Indices are defined over the different markets and sectors as benchmarks. An index is a virtual portfolio of stocks. Its value is given by the weighted value of its components, like for a classical portfolio of stocks. However it is only a virtual financial tool; it is not possible to buy or sell it. In order to exactly obtain the index return, one needs to replicate the portfolio by buying all the underlying stocks. Some mini-indices, called "trackers", are also traded with the purpose of tracking index with fewer stocks and less transaction costs. The S&P500 is a major index defined by 500 stocks traded on the New York Stock Exchange (NYSE), the largest market in the world, and is a good measure of American market wealth.

The value of an index is a weighted mean of the prices of the underlying stocks, but an adjustment is required to take dividends into account. The day a dividend of a stock is payable, its price falls by the same amount. The owner of the security maintains his wealth because the price reduction is compensated by the dividend income. If this stock happens to be in the index, the mean of the underlying prices usually takes into account the price reduction, but not the dividend income! That is why an adjustement is needed in principle to incorporate dividends. However, it is not possible to correct the index price each time a dividend is paid. For example, the S&P500 index is composed of 500 underlying stocks with dividends paid several times per year. Instead we use a continuous dividend yield to model discrete dividends incomes.

4.2.4 Options

Options are pure derivative instruments. An option is a contract that gives the owner the right to buy (call option) or sell (put option) an underlying asset, e.g. a stock or an index, at a pre-defined price (strike price) at or before a given date in the future (maturity), whatever the price of the underlying asset is at this time. This is a right and not an obligation. The owner of a call will only exercise the option if the strike price is lower than the price of the underlying asset. The option is then said to be in-the-money. In this case, ignoring the transactions costs and assuming the underlying asset is sold immediately, a positive payoff equal to the difference between the asset price and the strike price is obtained. Otherwise, the option is out-of-the-money and payoff is null. The same reasoning can be done for puts. Figures 4.1 illustrates the payoff pattern at maturity. For index options, as the index is a virtual tool usually based on a large number of assets whose delivey is difficult, the settlement is always in cash and corresponds to the payoff.

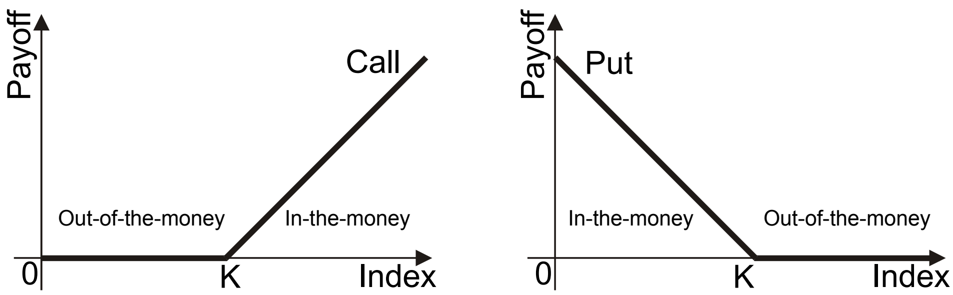


Figure 4.1: Option payoff

The price to pay to purchase an option, called a premium, depends on how the price of the underlying asset will fluctuate in the future. This is a complex subject that will be studied in more detail later. The option price is usually lower than the underlying asset price. Moreover, the payoff is a linear function of the underlying asset price at the exercise date. As the price is low and the payoff large, the investor can, with a small investment, obtain huge (positive or negative) returns. This phenomenon is known as financial leverage. Another advantage of the options is due to their typical piecewise affine payoff pattern. By appropriately combining options, it is possible to shape a portfolio payoff as one wants, manage the risk and even to completely insure a portfolio. These last two reasons already explain the reasons for option success.

4.2.5 Risk-free investment

We will assume that an investor can lend (or borrow) money at a given rate without risk of default. As there is no risk in this operation, the rate is lower than the expected stock returns. To find the level of the risk-free rate, a typical approach is to consider treasury bills. A treasury bill is as safe an investment as one can find. The risk of default is almost absent. Its return can be used as risk-free rate.

4.3 Continuous compounding

Indices and stocks are characterized by (expected) return rates. Risk-free investment is defined by the risk-free rate. The index dividend yield is also a rate of return. How can one handle all these rates?

Continuous compounding is generally used when working with options; this method will be used here. In this case, the interests of an investment are instantaneously and continuously reinvested and also produce interests. This is opposed to the simple interest method. The value of an investment S at a given rate R compounded m times per period at the end of a given horizon of T periods is

$$S(1 + \frac{R}{m})^{mT} \tag{4.1}$$

Continuous compounding is the limit of this expression (4.1) as m tends to infinity is the c and can be reformulated as

$$Se^{RT} \tag{4.2}$$

As interests produce interests, the final value is larger with continuous compounding than with any other compounding frequency. This is illustrated in Table 4.1 for an investment of 100USD at a rate of 10% during one year. As can been seen, after rounding, continuous compounding is close to daily compounding.

4.4 Arbitrage

4.4.1 Example

"An arbitrage is a portfolio offering something for nothing", Duffie [24]. "Arbitrage involves locking in a riskless profit by entering simultaneously into transactions in two or more mar-

Frequency	m	final value
Annually	1	110.00
Quarterly	4	110.38
Monthly	12	110.47
Weekly	52	110.51
Daily	365	110.52
Continuously	$\to \infty$	110.52

Table 4.1: Compounded interest

kets", Hull [32]. "In its simplest form, arbitrage means taking simultaneous positions in different assets so that one guarantees a riskless profit higher than the riskless return given by U.S. Treasury bills. If such profits exist, we say that there is an arbitrage opportunity", Neftci [55].

One rough definition of arbitrage could also be the possibility, without initial budget, to make a profit in the future whatever happens. When we consider a portfolio composed of a risk-free investment, stocks (or indices) and options, such opportunities could then appear if the price of the options is not carefully fixed. Let's consider the following example where an arbitrage opportunity exists due to underevaluation of a call price (thus it is quite interesting for the investor to buy it):

- Data:

Initial stock price (S) : \$20
Strike price (K) of the call : \$18
Risk free rate (r) : 10%/period
Call price : \$3

- Initially, the investor buys one call ($-\$3$), short sells one stock he doesn't possess($+\$20$) and lends the difference ($-\$17$). Initial cash flow is null.

- At maturity, the risk-free investment value is equal to $\$17e^{0.1} = +\$18.8$ and the value of the option depends on the stock:

  - Stock value $\geq \$18$:
    $-\$18$ (use call and close position) $+\$18.8$ (risk-free investment)
    $= \$0.8$ (payoff)

– Stock value (S_1) < \$18:
  $-S_1$ (buy stock and close position) +\$18.8 (risk-free investment)
  = \$18.8 − S_1 > \$0.8 (payoff)

Thus, without requiring an initial budget, it is possible, whatever happens in the future, to make a riskless gain of at least \$0.8. If there is no adjustment of the call price, nor other constraints, e.g. on the number of such portfolios that can be created, theoretically the investor will constitute an infinity of such portfolios to obtain an infinite profit. Of course, in reality, this is not possible; supply and demand on the market will adjust prices to remove such opportunities (or at least limit the consequences, as other constraints exist).

Optimization methods are very sensitive to the existence of arbitrage opportunities. We have seen that if the option price is not well defined and if no limiting constraints are imposed on traded quantities, then the optimization problem is unbounded and no solution can be returned. Option pricing is at once a financial problem and an operational one.

4.4.2 State-prices and arbitrage

The concept of risk-neutral valuation allows to characterize those security prices which exclude arbitrage possibilities.

Theorem. Let $S \in \mathbb{R}_+^N$ be the vector of current prices for a set of N securities, and let $\Pi \in \mathbb{R}_+^{N \times K}$ be the $N \times K$ matrix of future payoffs for the N securities under K possible scenarios. Then, there is no arbitrage opportunity if and only if there exists a positive vector $\psi \in \mathbb{R}_+^K$ such that:

$$S = \Pi\psi \tag{4.3}$$

We refer to Duffie [24] for a proof.

The vector ψ is called a state-price vector or Arrow-Debreu price vector. Its i-th component ψ_i is the marginal cost of obtaining an additional unit of account in state i. If the value of the stock becomes 1 in state 1 and 0 in state 2, the current value of the stock is given by ψ_1. Similarly, ψ_2 indicates how much investors would be willing to pay if the stock is worth 1 in state 2 and nothing in state 1. So by spending $\psi_1 + \psi_2$, the investor is sure to receive 1 unit of account in the future, whatever happens. The vector ψ can also be seen as the discounted risk-neutral probability for each scenario as we will show in the next section.

4.5 Option pricing

4.5.1 Classical methods

The two most famous methods to evaluate the value of an option are the Black-Scholes formula and the binomial tree model. The former relies on the continuity of the price of the underlying security, and the latter uses a discrete model where only two prices are considered at each given time period. In fact, these may be viewed as two extreme cases of a same model. It can be shown that letting the time period tend to zero in the binomial method yields the Black-Scholes model.

We will not develop these theories here, but only present the formulae and hypotheses. Complete explanations can be found in [32, 51, 49, 52].

4.5.2 Binomial trees

Principles

In a one-period binomial tree, we consider that only two states of the world can happen at the end of the period considered by the investor. It is possible to compute the initial option price by constructing a portfolio composed of the option, the underlying asset and the risk-free investment under the no-arbitrage assumption.

Consider the case of a stock initially priced at \$10, whose price at the end of the period becomes either \$12 or \$8, a risk-free rate r of 10% per period and a call with a strike price of \$11 and maturing at the end of the period. The value of the call at maturity is immediately obtained from the stock price at this time. This is illustrated in figure 4.2.

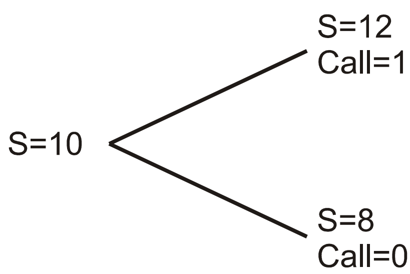


Figure 4.2: Binomial tree

We construct a portfolio including the stock and the option such that there is no uncertainty about the value of the portfolio at the end of the period; i.e. the portfolio value should be the same for the two possible states of the world. Suppose we buy Δ shares of stocks and sell one call. We obtain a single linear equation by equating the two portfolio values:

$$12\Delta - 1 = 8\Delta - 0$$

or

$$\Delta = 0.25$$

Whatever state happens in the future, if the portfolio is composed of one quarter of stock and one call in short position, then its value will always be \$2.

Since this portfolio value is without risk, its return must be equal to the risk-free rate; otherwise an arbitrage opportunity will appear. We can now compute the initial value of the portfolio by discounting its final value by the risk-free rate: $\$2e^{-0.1} = \$1.81$. As we know the composition of the portfolio, its initial value and the initial stock price, we then obtain the following linear equation depending on the unknown call option price :

$$\begin{aligned} & S\Delta - \text{call} = \text{portfolio value} \\ \Leftrightarrow\ & 10 \times 0.25 - \text{call} = 1.81 \\ \Leftrightarrow\ & \text{call} = 0.69 \end{aligned} \tag{4.4}$$

and the initial call price must be \$0.69 to avoid arbitrage opportunities.

Formulae

If u and d are respectively coefficients of increase and decrease of the stock price at the end of T periods, and f_u and f_d are respectively the final values of the option at the end of the T periods in the two corresponding cases, then the initial option price f must satisfy the equation:

$$\begin{aligned} & S\Delta - f = (Su\Delta - f_u)e^{-rT} = (Sd\Delta - f_d)e^{-rT} \\ & \text{where } \Delta = \tfrac{f_u - f_d}{Su - Sd} \end{aligned} \tag{4.5}$$

These conditions yield the value of f as:

$$\begin{aligned} & f = e^{-rT}(pf_u + (1-p)f_d) \\ & \text{where } p = \tfrac{e^{rT} - d}{u - d} \end{aligned} \tag{4.6}$$

This is the sole possible price to avoid arbitrage opportunities in a one-period binomial tree model.

The main assumptions formulated here are:

1. Only two states of the world are possible in the future;

2. No arbitrage opportunity exists.

Multi-period binomial trees

Representing the future by only two leaves of a tree is very restrictive. To increase the number of states, we can start a new binomial tree at each leaf of the previous one. Usually, but not necessarily, the lower node of a tree recombines with the upper node of the tree below it; so that one more state is added each time we add one layer to the binomial tree.

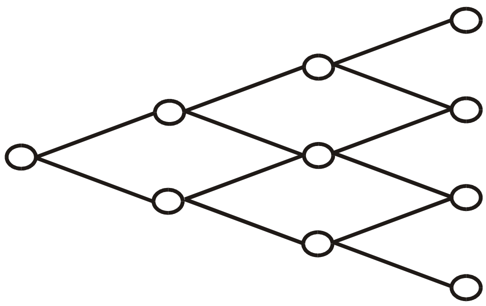


Figure 4.3: Multi-period binomial tree

If we know the risk-free rate for all sub-periods, the stock price at each node of the tree and the price of the option at the end of the tree (at maturity), then we can compute the price of the option at each node by backward propagation from the end to the beginning of the tree.

Construction of the tree

Assume that over the investment horizon, the stock returns are modelled by a normal probability distribution of parameters μ and σ. The the parameters u and d can be selected to approximate this distribution. Namely, if the number of sub-periods is large enough (in practice 30 or more layers) for the whole horizon, then the binomial tree yields a good representation of the distribution. It can be shown that a way to obtain this result is to set for each sub-period of length Δt:

$$\begin{aligned} u &= e^{\sigma\sqrt{\Delta t}} \\ d &= \tfrac{1}{u} \end{aligned} \tag{4.7}$$

Arbitrage

We can link the binomial approach with the arbitrage equations. As stated before, no arbitrage opportunities exist if and only if it is possible to find the risk-neutral probabilities. Indeed, the risk-neutral probabilities $(p, 1-p)$ can be identified with the variables $\psi_1, \psi_2)$, up to a constant factor. To see this, set:

$S_1 = 1$ (the price of \$1, representing the risk-free asset).
$S_2 =$ the initial price of the stock underlying the option.
$S_3 =$ the unknown price of the option.
$K = 2$ scenarios.
u, d as the coefficients of increase and decrease for the stock price (2 scenarios).
We get:

$$\begin{pmatrix} 1 \\ S \\ option \end{pmatrix} = \begin{pmatrix} e^r & e^r \\ Sd & Su \\ option(Sd) & option(Su) \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} \tag{4.8}$$

It follows that the risk-neutral probability p of the binomial tree is given by $e^r\psi_2$. The vector ψ can also be seen as the discounted risk-neutral probability for each scenario.

4.5.3 Black-Scholes formula

Formulae and hypotheses

The Black & Scholes formula assumes that stock prices follow a geometric Brownian motion. For this model and under some additional assumptions, Black and Scholes (**??**) derived the price of derivatives such as options by relying on arguments similar to the ones used for binomial trees. To define the prices, they construct an instantaneous portfolio composed of a fraction of the stock and of the option, so as to obtain a riskless portfolio. Full explanations can be found in [32, 51, 49, 52].

The well known option pricing formulae derived by Black and Scholes for the European calls an puts are:

$$\text{call} = SN(d_1) - Xe^{-r(T-t)}N(d_2) \tag{4.9}$$

$$\text{put} = Xe^{-r(T-t)}N(-d_2) - SN(-d_1) \tag{4.10}$$

where S is the current price of the underlying asset, X is the strike price, $N(x)$ is the cumulative probability distribution function of a standardized Normal variable, T is the time of maturity, and d_1 and d_2 are defined by

$$d_1 = \frac{\ln(\frac{S}{X}) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 = d1 - \sigma\sqrt{T-t}.$$

The assumptions made are:

1. Stock returns follow a Normal distribution defined by a mean μ and a constant standard deviation σ.
2. Short selling (the sale of something one does not possess) is allowed.
3. There are no transaction costs or taxes. All securities are perfectly divisible.
4. There are no dividends during the life of the derivative.
5. There are no arbitrage opportunities.
6. Security trading is continuous.
7. The risk-free rate r is constant and the same for all maturities.

Some of these assumptions can easily be relaxed. In particular, if the underlying asset is an index characterized by a continuous dividend yield q, then option prices become:

$$\text{call} = Se^{-q(T-t)}N(d_1) - Xe^{-r(T-t)}N(d_2) \tag{4.11}$$

$$\text{put} = Xe^{-r(T-t)}N(-d_2) - Se^{-q(T-t)}N(-d_1) \tag{4.12}$$

4.6 Risk-neutral valuation

4.6.1 Concept

The risk-neutral approach is one of the most important concepts in finance. In fact, the binomial option evaluation method and the Black and Scholes formula are two applications of this approach. We could have first this idea presented, but we preferred, as is usually done in finance books, to start with examples in order to help understand this central financial concept.

It is important to notice that, in the computation of option prices for binomial trees, we have never defined the probabilities attached to the possible states of the world. The option price is independant of these probabilities! The stock prices defined in the tree, together with the risk-free rate, contain all the information required to evaluate the option, even if we need more information to compute the (real market) expected stock return. Equivalently, this means we don't need or use the expected stock price. To construct the tree, this parameter is not even taken into account.

However, the parameter p in (4.6) can be seen as a probability associated with an upward movement and $(1-p)$ as a probability associated with a downward movement. For this probability distribution, the expression $pf_u + (1-p)f_d$ in (4.6) then becomes the expected value of the option which must be discounted by the risk-free rate in order to obtain the current option price. Moreover, with the same distribution, the expected stock price is given as :

$$\text{Expected stock price} = pSu + (1-p)Sd = Se^{rT}$$

Thus, when we use probability p, it appears that the expected return of the risky asset is the risk-free rate. This means that the investor doesn't require a compensation for investing in a risky asset, as if he were indifferent to risk. Such a constructed environment is called a risk-neutral world. This is a key result in finance. Knowledge of the risk-neutral probabilities leads to several simplifications of finance work. Here in particular, they allow to directly obtain the option price, by weighting the final option price and discounting the result by the risk-free rate. We don't even need information about the stock. Inversely, we can directly compute the risk-neutral probabilities from the risk-free rate and the stock prices in the binomial tree using the second equation in (4.6).

The same statement can be made for Black & Scholes model. It appears from the Black-Scholes equations that the option value doesn't depend on the risk preferences of investors. Indeed, the level of risk the investor can tolerate determines the expected return μ he requires for the stock. As μ doesn't appear anymore in equation (4.10), the value of the option remains the same whatever the risk preferences of investors.

4.7 Complete market

According to Dothan's definition [21], a market is complete if and only if every consumption process (portfolio values) is attainable. Mathematically, the market is complete if and only if $rank(\Pi) = K$ where Π is the payoff matrix defined previously. Indeed, in this case, all the columns of the payoff matrix Π are linearly independent. Therefore, for every consumption vector b selected by the investor, there always exists a solution of the system $\Pi^t x = b$, where x represents the quantities to invest in each security. This is of course a very valuable property.

Chapter 5

Modelling the future

5.1 A multi-period scenario approach

5.1.1 One-period multinomial model

A natural method to model the future is to use a scenario-tree approach. The root of the tree represents the current state of the world. The leaves, connected to the root, represent possible scenarios, or states of the world, or outcomes at the end of the period. More precisely, each leaf is associated with the values of each of the securities considered in the corresponding state of the world, and with the probability that this state occurs. We call such a tree a one-period multinomial tree of scenarios.

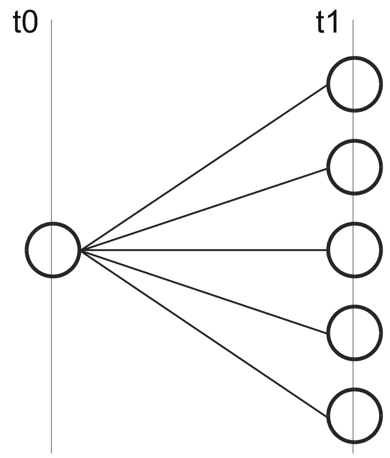


Figure 5.1: One-period multinomial tree of scenarios

A tree of scenarios is a flexible model where no constraint are set on how to define the states of the world. To do it, a possible approach is to instantiate the tree by sampling from probability density functions; e.g. the possible returns of a stock in the future are often approximated by a Normal probability density function.

Nevertheless, we are not restricted to this classical Normality assumption to represent securities in the future. In particular, if the investor has accurate or specific information about the market and can construct a corresponding probability density function, he will be able to profit of it by introducing his knowledge into the tree. This topic is covered in more details in section 5.3.

The uncertainty in this model is represented by the fact that we don't know which scenario will materialize.

5.1.2 Two-period multinomial model

In most of the subsequent developments, we will not restrict the representation of the future to one period, but we will add degrees of freedom by introducing a second period. So, there are now three distinguished instants in time, say $t0$, $t1$, $t2$. The initial instant is $t0$. The end of the first period is $t1$ and the end of the second period is $t2$. For each state of the world at time $t1$, say S,we consider another set of scenarios which defines all possible states of the world at time $t2$ given that state S has occured; i.e. we add one one-period tree to each leaf of the first period tree. The lengths of the first and second periods do not have to be equal in this setting.

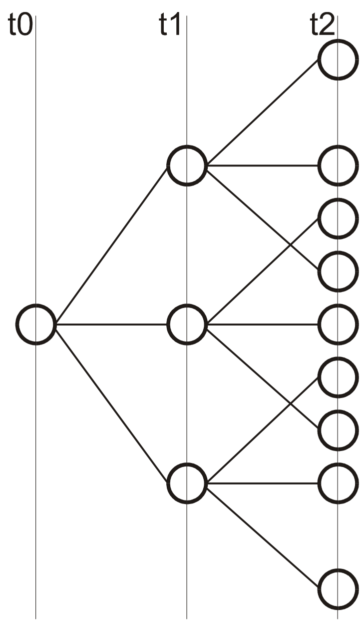


Figure 5.2: Two-period scenario-tree

As for the one-period tree, the investor can instantiate each node of the tree according to a probability density function which takes or not into account his specific knowledge of the market. Note that all the second period subtrees could be interpenetrated or not, with or without recombining leaves, with equiprobable or non-equiprobable nodes. Such multinomial trees of scenarios constitute a larger family than the classical family of binomial trees. As we will show in section 5.1.4, every (constant) multi-period binomial tree could be replaced by a multinomial one, but not the converse.

The two-period tree structure models the principle of information being revealed as time passes. Indeed, the main goal of introducing the second period is to model the fact that, when an investor is active over a long horizon, he adjusts his portfolio as time goes according to new available information an to changing conditions; i.e. the investment process is dynamic over time. The purpose of the second period is typically to model this ability to adjust a portfolio after some time.

Note that even if a two-period model is used for portfolio optimization, the adjustment to perform on the portfolio at the beginning of the second period (time $t1$) will normally not be implemented according to the optimal solution computed at time $t0$. Rather, a new two-period instance of the problem will be formulated and optimized at time $t1$; i.e. a suitable roll-over strategy will be adopted. Note also that the length of each period can change with each roll-over shift. In view of these comments, it is perhaps not optimal to add more time layers to the tree, as the resulting increase in the problem size and complexity may not outweigh the improvement in the quality of the solution. But we have not tested this hypothesis.

5.1.3 Interesting properties

Multiperiod multinomial tree of scenarios have some nice properties. First, the model is flexible as the investor is not constrained to a specific probability density function to instantiate the tree. Few assumptions are made. Second, the multiperiod construction allows to handle dynamic multi-period problems. Moreover, each period can be represented by only one subtree. This is an advantage with respect to binomial trees as a one-period multinomial tree of scenarios can be substituted to any multi-period binomial tree. Thus, this reduces the complexity. Third, a tree of scenarios is easily described and handled in contrast with other stochastic models. Those other models often require more complex set of equations to model the future and are only valid under simplifying hypotheses. Finally, it is a natural and easily understandable tool. All investors, even those lacking deep mathematical knowledge, are able to understand the principles of a tree of scenarios.

The reader might wonder whether the representation by a finite discrete set of scenarios does not omit too much information in comparison with a continuous "complete" representation. We think that if the sampling is performed carefully from an adequate probability density function, it is useless to consider large sample sizes. In particular, depending on the problem to solve and especially in the case of the VaR problem considered in the last chapters, it appears that increasing the number of scenarios in the tree does not modify significantly the results. The quality of the sampling process and the selection of the density function are more important than the sample size. These two topics are respectively presented in sections 5.4 and 5.3.

5.1.4 Binomial tree vs multinomial tree

Introduction

A multinomial tree could be seen as a multiperiod binomial tree where the intermediate nodes are pruned to keep only the final leaves. If we can construct a multiperiod binomial tree from the leaves of a multinomial tree, we can directly apply all the results obtained in finance for binomial trees to multinomial ones. Unfortunately, this is generally not possible as this is shown by the following two propositions. From these propositions, it is clear that the binomial family of trees is only a subset of the multinomial one. A third proposition is given to show that it is possible to create similar trees.

First proposition

It is generally impossible to construct a multiperiod binomial tree with the same leaves as a one-period multinomial tree.

Proof:

If we are able to construct a multiperiod binomial tree with the same leaves as a one-period multinomial tree, then the set of possible returns for the two trees are the same. For a binomial tree, Luenberger [48] defines the following system of equations to characterize the mean μ and the standard deviation σ of the returns:

$$\begin{aligned} p_{up} \ln u + (1 - p_{up}) \ln d &= \mu \Delta t \\ p_{up}(1 - p_{up})(\ln u - \ln d)^2 &= \sigma^2 \Delta t \end{aligned} \tag{5.1}$$

where p_{up} is the probability of the up branch and Δt is the size of one period (typically each period has the same length and $\Delta t = 1/k$).

To replicate the first two moments observed in the multinomial tree, we have to solve a set of two equations with three unknowns; i.e. we have one degree of freedom.

We can set u to replicate the largest value of the tree. If the largest value of the multinomial tree is S_{max} then:

$$\begin{aligned} & S_0 u^k = S_{max} \\ \Leftrightarrow \quad & u = \sqrt[k]{\tfrac{S_{max}}{S_0}} \end{aligned} \tag{5.2}$$

By setting u, we also determine a unique solution to the system (5.1); i.e. a unique possible value for the probabilities attached to the leaves and a unique value for the coefficient of decrease. This means that there exists only one multi-period binomial tree characterized by a given mean, volatility, number of final leaves and upper node even if there exists an infinity of multinomial trees, equiprobable and not equiprobable, that are characterized by the same properties. $\square$

Second proposition

If the number of states is larger than two then it is impossible to construct a multiperiod (constant) binomial tree with equiprobable leaves, then it is impossible to match an equiprobable multinomial tree with a multiperiod binomial tree. This is only possible with an unequiprobable multinomial tree.

Proof:

Consider the structure of a multiperiod binomial tree. After k periods the number of final states equals 2^k (we suppose here that the increase coefficient u and the decrease coefficient d along each of the two paths are kept constant through the periods), but we can observe only $(k+1)$ different values. Effectively, the final values are given by $S_0 u^i d^{(k-i)}$ where i is the number of up branches encountered to obtain the value. As i is an integer value and can vary from 0 to k, we get $k+1$ different possible values. Moreover, the probability of each of these values is given by $C_k^i p_{up}^i p_{down}^{(k-i)}$.

If the final leaves must be equiprobable, then the value $C_k^i p_{up}^i p_{down}^{(k-i)}$ must be the same for each i. If we consider the two extreme values 0 and k, then:

$$\begin{aligned} & C_k^0 p_{up}^0 p_{down}^k = C_k^k p_{up}^k p_{down}^0 \\ \Leftrightarrow \quad & p_{down}^k = p_{up}^k \\ \Leftrightarrow \quad & p_{down} = p_{up} = 0.5 \end{aligned} \tag{5.3}$$

If k equals one, we face a one-period binomial tree with equiprobable states. However, as soon as we add just one period, i.e. k is larger than one, the probabilities cannot stay equal. We already know the result $p_{down} = p_{up} = 0.5$ (independant of k) for i equals to 0 and k. If we consider i equals to 0 and 1, we find another value:

$$\begin{aligned} & C_k^0 p_{up}^0 p_{down}^k = C_k^1 p_{up}^1 p_{down}^{(k-1)} \\ \Leftrightarrow\ & p_{down}^k = k p_{up} p_{down}^{(k-1)} \\ \Leftrightarrow\ & p_{down} = k p_{up} \\ \Leftrightarrow\ & p_{down} = \tfrac{k}{k+1} \neq 0.5 \end{aligned} \tag{5.4}$$

[]

The following picture summarizes the two previous results:

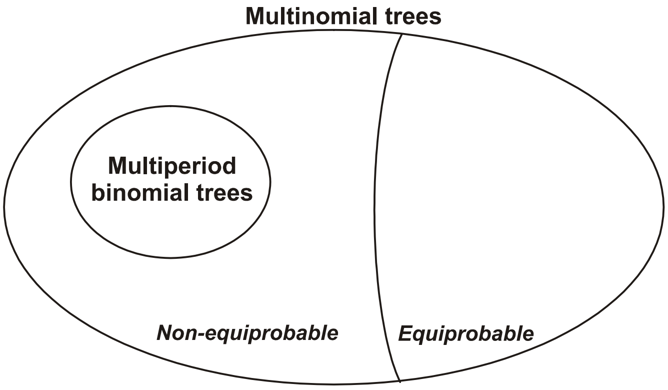


Figure 5.3: Binomial and multinomial sets

Third proposition

It is always possible to construct a multi-period binomial tree with the same number of final leaves, the same mean and the same standard deviation of returns as in any multinomial tree.

Proof:

By the first proposition, a $(n-1)$-period binomial tree with u,d and p_{up} given by (5.1) satisfies the required properties. []

We know that if the number of periods is large ($n > 30$ is often considered enough in finance), the distribution of returns converges to the normal case. The normal distribution being fully characterized by its mean and its standard deviation, the previous proposition implies that for large n it is always possible to construct a multi-period binomial tree similar to any multinomial tree. This considers only the normal distribution function.

For example, consider the following serie of 40 equiprobable returns

{ -1.13,-0.75,-0.58,-0.45,-0.35,-0.26,-0.19,-0.12,-0.06,0.00,0.05,0.11,0.16,0.21, 0.26, 0.30,0.35,0.39,0.44,0.48,0.52,0.56,0.61,0.65,0.70,0.75,0.79,0.84,0.89,0.95, 1.00,1.06,1.12,1.19,1.26,1.35,1.45,1.58,1.75,2.13}. The mean and the standard deviation are respectively equal to 0.5 and 0.7. We obtain by (5.1) the 39-period binomial tree $(u, d, p_{up}) = (1.1326, 0.9059, 0.5)$. By choosing the probability p_{up} equals to one half, we obtained a symmetric representation where most of the generated returns are around the mean and the others split equally around the mean. This is illustrated in Figure 5.4 where the corresponding cumulative distribution functions are very close.

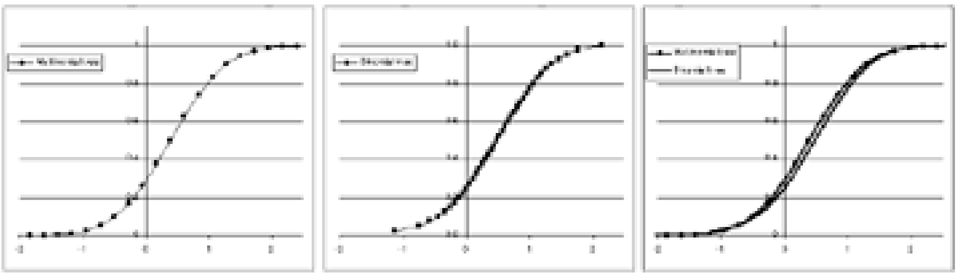


Figure 5.4: Similar binomial and multinomial trees

We are now able to obtain two similar trees. Both have the same distribution of returns and the same scheme of final leaves can be observed by sorting the final values by decreasing order.

5.2 Empirical data and implied parameters

5.2.1 Introduction

In order to define a node of the tree, for portfolio problems including options and an index, we need to define three sets of values: risk-free rate, index value and option values. These values, especially index and option values, depend on exogenous parameters such as dividend yield, length of each period and time to maturity. When setting up the tree, we need to obtain "true" values or extremely good approximations of these parameters. Our initial numerical experiments showed us that slight perturbations of some parameters lead to large variations of the results or to model incoherences (e.g. a large volatility spread between puts and calls if the dividend yield is not sharply adjusted).

The parameters must not only be set to their "true" values, they also need to constitute a coherent set. In particular, even when using the right exogeneous parameters, the definition of option prices must match the model used to define the underlying future index value. For example, we face serious problems if we try to use the observed S&P500 option prices, which usually do not perfectly match the Black and Scholes formula based on a Normal distribution, and model at the same time the S&P500 index return by a Normal probability function. In this case, in our initial numerical experiments, the expected return of the optimal portfolio is anormally huge.

Two methods can be considered to set the parameters: either an empirical study of the past or a prediction of the future based on instantaneous available data. When relying on past data, we implicitly assume that the past provides a good representation of the future or at least, that it contains enough information to develop accurate forecasts. That is we make the hypothesis that what happened in mean or in trend over several years in the past somehow allows to forecast what will happen in the short term future! E.g, first analyses for the S&P500 index show that the assumption of stationarity of returns is not satified and therefore we should be carefull when computing the past moments of the return distribution if we want to use them to model the future.

The second possibility is to use current available information. In particular, we will see in section 5.3.5 that it is possible to extract many interesting implied parameters from the option prices observed at time $t0$. The market prices of options can be interpreted as reflecting the investors' expectations about future market moves. It is interesting to use market prices because they contain all the relevant information and because they can be observed directly at the precise time when the investment decision is to be made.

We concentrate here on two sets of important implied parameters: first, the volatility and the smile effect in section 5.2.2; second, the risk-free rate and the dividend yield in section 5.2.3.

5.2.2 The smile effect

Definition

One of the most important parameters to define is the dispersion of returns. Volatility is a parameter used in the index return generation process, but is also used for the option pricing process as prices are usually defined according to index distribution (this can be viewed directly in the Black-Scholes formula).

As mentioned in the introduction, an adequate method to derive good current estimates of the parameters, in particular volatility, is to use the option prices observed on the market at time $t0$. Black and Scholes have defined option price as a function depending on the index volatility and it is possible (using dichotomic search) to inverse the BS formula to obtain volatility as a function of option price. So, if the BS formula applies (in particular if the Normal distribution assumption holds), then we can use one observed market option price to obtain the true instantaneous volatility. When the index return distribution is empirically close to the normal distribution, we can expect the inversion method to provide a good approximation of its volatility (better than the historical volatility?).

There is however a well known problem called the smile effect. Black & Scholes made the assumption that volatility is unique and constant. Therefore, in this model, all the options, even with different strike prices, are defined according to one same value of volatility : the index volatility. If the model applies we should obtain the same value for volatility for all the options observed on the market when we inverse the function. However, in fact, the volatilities implied by options are not constant, but are a function of option strike prices.

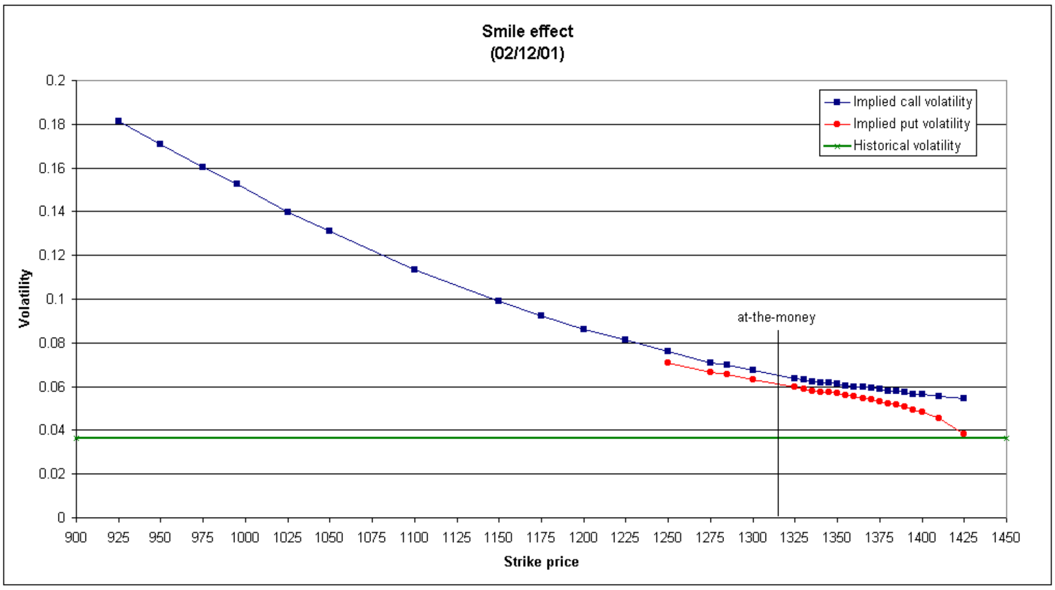


Figure 5.5: Smile effect

The figure 5.5 shows the monthly implied volatilities for calls and puts on the 2nd of February 2001 (10 o'clock) with maturity in March. 51 calls and 51 puts were observed on the market. 33 calls and 40 puts were used to compute the smile. The others were rejected due to too low a price and numerical instabilities. Knowing the value of the volatility as a function of the strike price is interesting in its own right (e.g., we will use this result in the next chapter to define some option prices), but our first goal is to extract one specific value of the volatiliy to model future index prices. Common practice is to use the mean of the volatilities implied by the two calls which are nearest-to-the-money, because, by definition of the BS formula, prices are more sensititive to volatility for at-the-money options than for deep-out-of-the-money options. However, other weigthing schemes are also possible (see Hull [32]).

In practice

To compute the smile, we usually use only call options and not puts. Indeed, calls give a better representation of volatility because they are more traded on the market than puts. Figure 5.5, based on almost all available calls and puts, clearly illustrates this fact.

The first task is to collect data. Due to the recent development of communication medias, we are no longer restrained to one source. We can use commercial databases or consult Internet websites. In Europe, DataStream is one of the most used database. The advantage of using a database is that lots of information, both current and past, is concentrated in one place. Usual difficulties are that we cannot obtain other data than those defined in the database (e.g. we obtain only option closing prices and not bid and ask prices at different times) and that we cannot check their validity. Practice shows that some errors and inconsistencies usually occur. Therefore, we prefer to use as often as possible current observed information available directly on websites of the markets where the securities are traded.

This also allows us to use prices at different times. When should we collect the data? There seems to be no consensus in the finance research world. Rubinstein [63] and Shimko [64] use SP500 option prices at 10 o'clock. DataStream gives us prices at closure of the markets. However, this appears to have little effects on our results.

Computing the smile is always presented as an easy task as it is a smooth and well-defined function. In reality it is a little bit more complex for three reasons. First, when the option is well out-of-the-money, the index price and the option strike price dominate the sigma parameter in the Black and Scholes formula. In this case, the option price is not very sensitive to a variation of the volatility; conversely for a given option price, it is difficult to determine exactly the volatility. Second, the Black and Scholes (BS) formula is an increasing function of volatility. As volatility is by definition positive, the minimum of this function is reached when the volatility goes to zero. It appears that for several options, observed prices are lower than minimal BS prices. This means that the implied volatility does not exist for these options. Finally, implied volatilities also depend on exogenous parameters: dividend yield and risk-free rate. As mentioned in the introduction 5.2.1, the value computed for the volatility strongly depends on the estimates of these parameters. For example, the volatility spread between the puts and calls observed in the previous figure is due to the use of the historical dividend yield. If we replace it by an implied value as defined in the next section, the spread disappears. At the very least, coherence is required. Using the historical dividend yield with implied instantaneous volatilities leads to troubles. More generally, we cannot assume stationarity of the volatility. Historical volatility computed over the last 10 years and represented on the Figure 5.5 is far below the current volatility smile. Solutions to handle the first and second problems when modelling option prices are presented in section 6.6. The third problem is considered in the next section.

5.2.3 Risk free rate, index price and dividend yield

Definition

In our numerical experiments, we could use the risk-free rate and the dividend yield found in the databases; but for a non stationarity reason, this should be avoided. We could also use T-Bills to compute a current risk-free rate. However, slight differences appear depending on the T-Bill chosen. Moreover, the use of a T-Bill doesn't guarantee consistency with option values.

Option prices depend on the distribution of index returns. For this reason, we can extract from them not only the implied volatility but even the whole index distribution. Shimko [64] explains the following method to obtain the index price discounted by the dividend yield and the risk free rate (discount factor). All the options must satisfy the put-call parity relation. For a given index price (S), strike price (X), time-to-maturity ($T-t$), dividend yield (q) and risk free rate (r), the price of a call (c) is a function of the price (p) of the equivalent put:

$$c - p = Se^{-q(T-t)} - Xe^{-r(T-t)} \tag{5.5}$$

In other words, the difference in prices between two options is a linear function of the strike price. This function intercepts the vertical axis at the index price discounted by the dividend yield and the angular coefficient is the risk-free discount factor. We can observe prices on the market (51 sets of call and put for our last dataset). By a linear least-square regression we obtain values for the intercept (or the index value discounted by the dividend yield) and angular coefficient (or the risk-free rate), from which we can compute values for q and r.

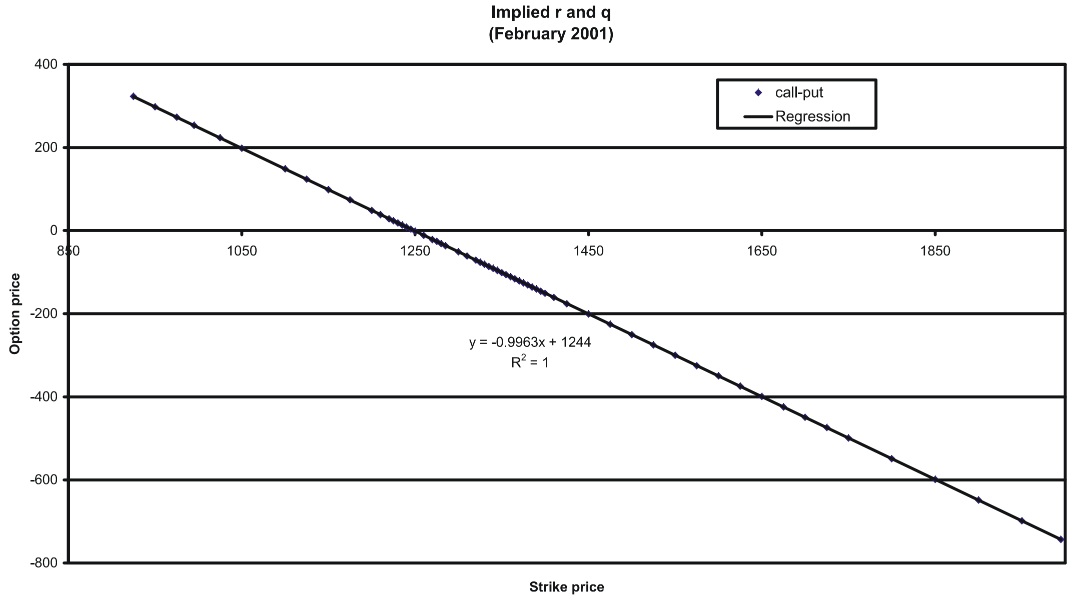


Figure 5.6: Call-put parity regression

In practice

First numerical experiments show that the S&P500 option prices nearly perfectly follow this linear relation. This is illustrated in Figure 5.6 for options with maturity one month later. The horizontal axis corresponds to the strike prices and the vertical axis measures the difference between the call price and the put price;i.e. the left term of equation (5.5). Each dot corresponds to a couple of call-put options with the same strike price observed on the market. The linear regression is represented by the line. Residuals are small (The coefficient R^2 in the previous example is close to one).

Obtaining the intercept and the angular coefficient is easy to perform. The stability of these two values seems good even when considering subset of options. This is clear when considering Figure 5.6. This is important because we could be worried that the deep-out-of-the-money options would have a negative impact. However, computing the implied parameters remains troublesome. There are nevertheless small variations. This appears more clearly when we perform the transformation to obtain the implied rates q and r. This logarithmic transformation implies that the variation of these rates is larger. The effect on the montlhy return for the previous example is too large to be accepted. For options observed in February 2001, the corresponding annual risk-free rate varies from 1 to 6 percent. For options observed in March 2000, the corresponding annual risk-free rate varies from 4 to 18 percent. Numerical variations are given in Table 5.1 for different subsets of options with strike prices equally reparted around the index price.

nbO	angular coef.	intercept	q	r	R^2
2	-1.000000	1 248.50	-0.18198%	0.00000%	1
4	-0.994857	1 242.10	0.33172%	0.51561%	0.999992082
20	-0.999011	1 247.33	-0.08854%	0.09892%	0.999989087
46	-0.997966	1 246.09	0.01122%	0.20359%	0.999998825
63	-0.996260	1 244.00	0.17924%	0.37473%	0.999998755

Table 5.1: Implied dividend yield and risk-free rate

Note that this is not a numerical rounding problem in the logarithmic transformation, but a problem in the regression process. Indeed, in financial problems, the rates are rarely used directly but instead almost always in the exponential form. We could then directly use the intercept and the angular coefficient in the models without performing the logarithmic transformation. However there is no significative loss of information when we first convert those two coefficients into the implied rates and then, when required, we compute the exponential values.

At this point, we cannot automatically use the results obtained by the regression over the whole set of observed options. To obtain valid parameters, we have first to select carefully the options that will be taken into account during the regression process. The nearest-to-the-money options are the most representative and are considered in priority. We then construct the largest possible set of options, but we reject it if the regression result is inconsistent, e.g. a negative dividend yield, and if the implied parameters q and r are too far from reasonable values, e.g. with respect to T-Bill rates and historical rates.

5.3 Probability density functions for index returns

5.3.1 Subjective and risk-neutral probabilities

The investor can freely define the tree of scenarios based on his knowledge of the markets, but it is a complex task to precisely define all the possible future states of the world. In a more general framework or as a starting point for the investor, we will consider that it is possible to model the future index return distribution by a probability density function.

Using density functions, we will try to represent what really happen on the market for index returns. We want to work with consensus subjective (i.e. that depends on the investor's behaviour and risk aversion) probabilities as the final goal is to model and solve real problems. Besides, in order to define financial tools, financiers work, implicitly or not, with a (certified) equivalent risk-neutral world, as explained in Chapter 4. This is implied in the option pricing formulas and also used for example to define state-prices (close to risk-neutral probabilities). Of course, density functions of index returns in the risky world and in the risk-neutral world are not the same. However, as Rubinstein observes [63] about his numerical results : "This tempts me to suggest that, despite warnings to the contrary, we can justifiably suppose a rough similarity between the risk-neutral probabilities implied in option prices and subjective beliefs".

The first three density functions discussed in the next sections (Normal, Theodossiou, Fernandez and Steel's density functions) are consensus subjective density functions. The following two (Shimko and Rubinstein) are certified equivalent risk-neutral density functions.

The subjective density function cannot be directly inferred from the risk-neutral one. However, by using utility functions it is possible to convert the first one to the latter one. The converse transformation is more complex, but also possible. We propose a method to do this. It will turn out to be a useful tool when optimizing the option pricing model (OP2) to be presented in Section 6.3.7.

5.3.2 Normal distribution

This is the basic density function presented until now. Its expression is given by:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{5.6}$$

where μ and σ are respectively the mean and the standard deviation of the distribution. These two parameters fully describe the shape of the function.

This function is essentially used for theoretical reasons as a benchmark. The analytical definition of the density function is well known. We work with the standardized version ($\mu = 0$ and $\sigma = 1$) and use the historical mean and deviation to adjust it afterwards. We have not defined here a likelihood procedure to find the best value for the two parameters. The main criticisms of this function are that it does not take into account a possible skewness and kurtosis usually observed on a real market. In particular, financial distribution are often characterized by fat tails. For the construction of the tree, due to the nature of the normal density function, we have to use either the analytical expression and numerical procedures or an interpolation technique over a pre-computed table.

5.3.3 Theodossiou's skewed distributions

The Student t function is a first improvement to represent the consensus distribution. It is close to the Normal distribution but uses one more parameter to allow to model a kurtosis effect (thickness of the tails). It is still unable to represent a skewness effect. Some authors have proposed an adaptation of the Student function with one more parameter to add this skewness property. We present here Theodossiou's version [66] and in the next section, Fernandez and Steel's version ([28]).

The idea is to split the density function in two areas around the null mode. We can rewrite Theodossiou's equations as:

$$f(x|k,n,\lambda,\sigma^2) = \begin{cases} f_1 = C(1 + \frac{k}{n-2}(\frac{|x|}{(1-\lambda)\theta|\sigma|})^k)^{\frac{-(n+1)}{k}} & \text{for } x < 0 \\ f_2 = C(1 + \frac{k}{n-2}(\frac{|x|}{(1+\lambda)\theta|\sigma|})^k)^{\frac{-(n+1)}{k}} & \text{for } x \geq 0 \end{cases} \tag{5.7}$$

with:

$$C = \frac{S(\lambda)}{2\sigma}\sqrt{\frac{B(\frac{3}{k},\frac{n-2}{k})}{B(\frac{1}{k},\frac{n}{k})^3}} \tag{5.8}$$

$$\theta = S(\lambda)\sqrt[k]{\frac{k}{n-2}}\sqrt{\frac{B(\frac{1}{k},\frac{n}{k})}{B(\frac{3}{k},\frac{n-2}{k})}} \tag{5.9}$$

$$S(\lambda) = \sqrt{1 + 3\lambda^2 - 4\lambda^2\frac{B(\frac{2}{k},\frac{n-1}{k})^2}{B(\frac{1}{k},\frac{n}{k})B(\frac{3}{k},\frac{n-2}{k})}} \tag{5.10}$$

where $B(\cdot)$ is the beta function, C and θ are normalizing constants ensuring that $f(\cdot)$ is a proper density function, k and n control the height and tails, λ measures the skewness (f_1 and f_2 have identical expressions except for the coefficient $(1 \pm \lambda)$) and σ^2 is the volatility.

The parameters are defined only over specific domains. Depending on their values, we obtain other classical density functions: Normal pdf ($\lambda = 0, k = 2$ and $n = \infty$), Student's t distribution ($\lambda = 0, k = 2$), Cauchy distribution... The four parameters don't correspond to the first four moments as we could hope. σ is the sole parameter corresponding to the theoretical standard deviation of the underlying random variable. Theodossiou gives (complex) analytical expressions to obtain the three other corresponding moments (mean, skewness and kurtosis) from its four parameters, but not the converse. It implies that it is not possible to perform an empirical study over the historical data to completely define Theodossiou's pdf as we cannot convert historical moments into Theodossiou's parameters. Therefore, a likelihood estimation must be performed to obtain the adequate values. However, as Theodossiou himself said, "the maximization of the log-likelihood function is troublesome". On many occasions, the iterative algorithm "overshoots" and assigns impossible values (negative variance, $|\lambda|$ larger than one...), resulting in a breakdown of the algorithm. Theodossiou suggests a slight modification of the likelihood function and the use of Berndt's algorithm to avoid these problems. Even using this approach or other methods, we were unable to develop a general and always succesful method to perform likelihood maximization for our datasets.

We also have other problems with the domain of definition of some parameters as defined by Theodossiou, as they don't always seem to be valid for some analytical expressions where it should be the case (even taking into account that some moments could not always exist and so some analytical expressions should not be used in some specific cases). So we set aside Theodossiou's method to concentrate on Fernandez and Steel's [28] approach.

5.3.4 Fernandez and Steel's skewed distributions

Fernandez and Steel ([28]) have proposed a general method to add a skewness parameter to every unimodal and symmetric density functions. In particular, they illustrate the approach for the Student's t distribution. The general formula used for a univariate pdf $f(\cdot)$, where $f(\cdot)$ is unimodal and symmetric around zero, is given by:

$$p(\varepsilon|\gamma) = \begin{cases} \frac{2}{\gamma+\frac{1}{\gamma}} f(\frac{\varepsilon}{\gamma}) \text{ for } \varepsilon \geq 0 \\ \frac{2}{\gamma+\frac{1}{\gamma}} f(\varepsilon\gamma) \text{ for } \varepsilon < 0 \end{cases} \tag{5.11}$$

where $\gamma \in (0, \infty]$ is the added skewness parameter. The basic idea is simply to introduce a scaling factor in the negative orthant and its inverse in the positive orthant. The coefficient ensures that the result is a proper pdf. The mode is preserved, but the skewness is modified when γ is not equal to one.

It is also possible to obtain E_r, the moment of order r of this new pdf, from the corresponding "absolute" moments M_r of $f(\cdot)$:

$$E_r = E(\varepsilon^r|\gamma) = M_r \frac{\gamma^{r+1} + \frac{(-1)^r}{\gamma^{r+1}}}{\gamma + \frac{1}{\gamma}} \tag{5.12}$$

where

$$M_r = 2\int_0^{\infty} s^r f(s) ds$$

Fernandez and Steel develop the case where $f(\cdot)$ is a general Student's t function. The expression becomes :

$$p(\varepsilon|m,s,\nu,\gamma) = \begin{cases} \frac{2}{\gamma+\frac{1}{\gamma}} \frac{\Gamma(\frac{\nu+1}{2})}{s\sqrt{\Pi\nu}\Gamma(\frac{\nu}{2})}(1+\frac{(\varepsilon-m)^2}{\nu s^2\gamma^2})^{\frac{-(\nu+1)}{2}} & \text{for } \varepsilon - m \geq 0 \\ \frac{2}{\gamma+\frac{1}{\gamma}} \frac{\Gamma(\frac{\nu+1}{2})}{s\sqrt{\Pi\nu}\Gamma(\frac{\nu}{2})}(1+\frac{\gamma^2(\varepsilon-m)^2}{\nu s^2})^{\frac{-(\nu+1)}{2}} & \text{for } \varepsilon - m < 0 \end{cases} \tag{5.13}$$

where:

- m, as the mode, models only the location. It is the mean of the underlying student's t distribution, but no longer the mean of the skewed pdf;
- s^2 models only the dispersion. It is the variance of the underlying student's distribution, but not of the skewed version;
- $\gamma > 0$ models only the skewness;
- $\nu > 0$ models only the kurtosis.

We can see here that each of the four parameters models only one aspect of distribution. This pdf is easier to interpret than Theodossiou's. However, the four parameters still don't correspond to the first four moments of the distribution. In particular, if there is skewness, then the mean shifts away from the mode m.

For simplicity, we use here the "classical" standardized Student's t function by setting m to 0 and s to 1 in the previous formula. We can use the standardized version because centering and reducing the data do not affect skewness and kurtosis (especially, changing the mean shifts the function but does not modify the shape). When we find these last two parameters, we can adjust the first two to match the real data. Instead of looking for four values, we now have only two to compute. A log-likelihood optimization by Powell's method ([61]) is performed to obtain the two adequate values for the parameters based on historical data. We also tried using a conjugate gradient maximization algorithm to do it where the gradient is computed numerically by Ridder's method, but the use of numerical derivatives in the process is not suitable here as it is time consuming and also leads to new approximations. To avoid a breakdown of the algorithm due to out-of-domain errors (negative values of the parameters), we substitute ν by $exp(\nu)$ and γ by $exp(\gamma)$; i.e. by two monotonic functions defined on $I\!R$ with values in $I\!R_+$.

When we know the two parameters of interest we can obtain the first four centered moments of the final distribution as follows:

$$
\begin{array}{|l|l|}
\hline
m_1 = E_1 = M_1(\gamma - \frac{1}{\gamma}) & \mu = m_1 \\
m_2 = E_2 - \mu^2 & \sigma^2 = m_2 \\
m_3 = E_3 - 3\mu\sigma^2 - \mu^3 & \text{skewness} = m_3/\sigma^3 \neq \gamma \\
m_4 = E_4 - 4\mu_3 - 6\mu^2\sigma^2 - \mu^4 & \text{kurtosis} = m_4/\sigma^4 \neq \nu \\
\hline
\end{array}
\tag{5.14}
$$

An alternative to the likelihood maximization method would be to use Lambert and Laurent's approach. Lambert and Laurent [42] use Fernandez and Steel's function with the standardized Student's t function (with an adjustement of ν). Their paper shows how to find the two parameters of Fernandez and Steel's function thanks to a GARCH (predictive) model. However, we did not use this method until now as it would haven taken us in a completely new direction.

5.3.5 Breeden, Litzenberger and Shimko's implied distributions

Introduction

Using Fernandez and Steel's function is already a great improvement over the Normal distribution approach, though this approach still has two drawbacks. First, in order to fit the distribution to the real world, we have to rely on historical data. So we use past information to model the current market and to help to make a decision today. A improvement could use a GARCH model to make a prediction of the future, but this would remain an estimation of a future period based on old information. In all cases, we do not use (essentially) the information describing the current market. Second, it is not enough for our purpose to perfectly model the index distribution. We need to ensure coherence between option prices and index distribution. From our first numerical experiments, it appears that we obtain coherent results if we use a normal option pricing process when index returns are normally distributed. However, this is no longer the case if we use non normally distributed market option prices in conjunction with the normally distributed index returns. The expected return of the optimal solution becomes unrealistically large.

If we observe the whole set of options on the market, then it is possible to retrieve lots of parameters of the index distribution. The current prices of the options reflect what the investors expect todays about the future index returns. If we can obtain implied index distribution parameters from option prices, then we solve our two previous problems: time adequation and coherency.

Breeden and Litzenberger [5] elicited the relation between option prices and the pdf of the underlying index. They illustrated it by using the Black and Scholes option pricing formula.

Shimko [64] improved their method by using observed option prices. His approach consists of three steps. First, define a continous function of the strike price to model (observed) option prices. Second, use Breeden and Litzenberger's theorem to define the density function between the lowest and largest observed strike prices. Third, adjust the tails of the pdf. We now describe these steps in more detail.

Option pricing function

Before applying Breeden and Litzenberger's theorem, we first need to define a continous function of the strike price which models (observed) option prices; i.e. a function of the form $C(X)$ where X is the strike price and $C(X)$ is the corresponding option price. We could use the discrete set of prices observed on the market and use a linear interpolation technique to obtain intermediate values. Shimko proposed a tighter model based on the option smile. He first computes the implied volatilities of the observed calls using the inverse BS formula and then fits the observations with a second degree equation by the least-square method:

$$\sigma(X) = A_0 + A_1 X + A_2 X^2 \tag{5.15}$$

By reapplying the BS formula in the opposite sense adapted using this definition of $\sigma(X)$ instead of the constant σ, we obtain a continuous option pricing function over the strike prices which takes into account the observed smile effect. The result is a smoother function than the one obtained by direct interpolation. Note that the BS formula is only used as a conversion tool from option prices to volatilities (and conversely). It is just a mathematical translation formula. We could define other tools. The properties and assumptions behind the BS formula don't affect or limit the final results. From our numerical experiments on the S&P500 and the AEX, the fitting curve closely mimics the observed option prices. The result is illustrated in Figure 5.7.

The pdf around the mode

Second, Breeden and Litzenberger have demonstrated that partial derivatives of the option price $C(X)$ with respect to strike price (X) are related to the distribution function $F(S)$ and the density function $f(S)$ of the index prices in the following manner:

$$\begin{array}{l} C(X) = e^{-r(T-t)} \int_X^\infty (S-X) f(S) dS \;\; \text{Cox, Ross and Rubinstein's definition} \\ \frac{\partial C(X)}{\partial X} = -e^{-r(T-t)}(1 - F(S|S=X)) \\ \frac{\partial^2 C(X)}{\partial X} = e^{-r(T-t)} f(S|S=X) \end{array} \tag{5.16}$$

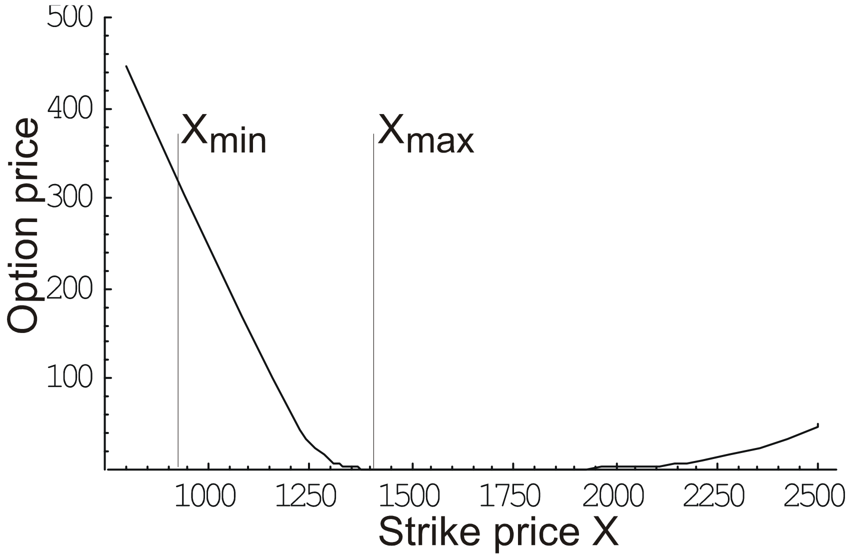


Figure 5.7: Option pricing function

No assumptions are made as to the stochastic processing governing the movement of the underlying security price or the option price. The only requirement here is that the option pricing formula be twice differentiable.

If we consider now the BS option pricing formula adjusted by Shimko to take into account the volatility, we can express $C(X)$ as follows:

$$\begin{array}{ll} C(X) = Se^{-q(T-t)}N(d_1) - Xe^{-r(T-t)}N(d_2) & \text{Black and Scholes' formula} \\ d_1 = \frac{\ln(\frac{S}{X}+(r-q+\frac{\sigma^2(X)}{2})(T-t)}{\sigma(X)\sqrt{T-t}} & \text{Shimko's } \sigma(X) \text{ defined in (5.15)} \\ d_2 = d_1 - \sigma(X)\sqrt{T-t} & \end{array} \tag{5.17}$$

where $N(\cdot)$ is the cumulative normal distribution and $n(\cdot)$ is the normal density function.

When we take the second partial derivative of (5.17 and adjust it by the risk-free rate as stated in (5.16), we obtain the probability density function of the index prices $f(S)$. The cumulative distribution is obtained by adjusting the first derivative.

$$\begin{array}{l} f(S|S=X) = -n(d_2)(d_{2X} - (A1 + 2A_2X)(1 - d_2d_{2X}) - 2A_2X) \\ F(S|S=X) = 1 + Xn(d_2)(A_1 + 2A_2X) - N(d_2) \end{array} \tag{5.18}$$

with:

$$\begin{array}{l} d_{1X} = \frac{-1}{Xv} + (1 - \frac{d_1}{v})(A_1 + 2A_2X) \\ d_{2X} = d_{1X} - (A_1 + 2A_2X) \\ v = \sigma(X)\sqrt{T-t} \end{array} \tag{5.19}$$

where $N(\cdot)$ is the cumulative normal distribution, $n(\cdot)$ is the normal density function and A_0, A_1, A_2 are the coefficients used in the volatility model (5.15).

The result is a well defined analytical expression because it is possible to analytically derive the BS option pricing formula. Of course, working with an analytical expression is always easier than working with a numerical derivative. However, if we want to consider another pricing function, the latter approach is also valid and could be the sole solution if no analytical (or simple) expression exists for the derivatives.

We face some problems because it appears that there is at least one typing error in Shimko's paper (the pdf is negative). Moreover, even after changing the sign, the final expressions mentioned by Shimko do not match the analytical (or numerical) derivatives we have computed. Our analytical developments were checked using Wolfram Mathematica software package and also by implementing numerical procedures in C language. The resulting equations are long and complex and so we prefer to illustrate, for our previous numerical example, in Figure 5.8 the differences between the two 2nd-order derivatives . In the rest of this work, we use the derivatives we have computed and not Shimko's ones.

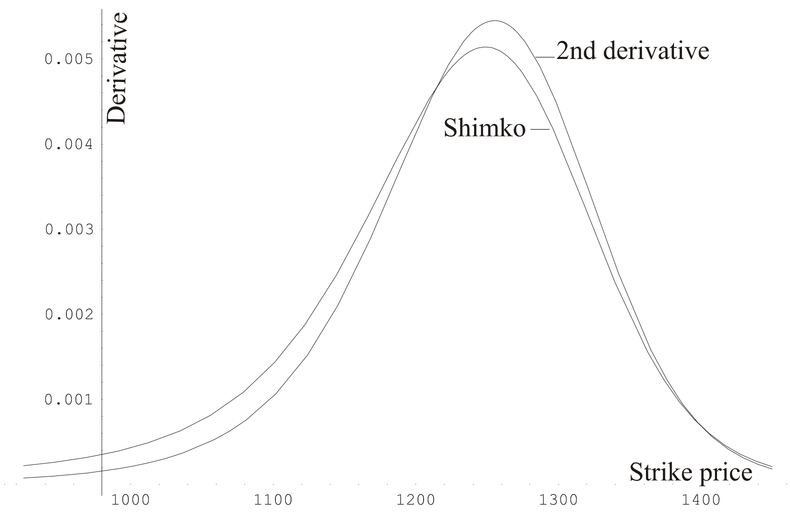


Figure 5.8: Differences between second partial derivatives

Adjusting the tails

Shimko performs a third step. He has defined the function $\sigma(X)$ between the minimal and maximal strike prices observed on the market. Outside this range, he assumes that the implied volatility is constant. Indeed, outside this range of values, we cannot guarantee the quality of the regression. Especially for large strike prices, the option pricing function is increasing due to the fact that the sigma function is quadratic (see figure 5.7). This leaves the technical problem: what should we do with probabilities beyond the range of strike prices associated with traded options?

We cannot arbitrarily construct the tails, but must face several constraints:

- The sum of densities must equal one.
- The cdf is a smooth function without breakpoints at the extreme strike prices.
- The expected return of the whole implied pdf should be equal to the risk-free return as the construction is made under a risk-neutral world hypothesis.

To obtain the lower tail, we could view the index price as an option with null strike price and use it in the regression. In this way, the whole lowest tail is defined. However, the lowest strike price for the options is typically much larger than zero; thus the index price is far from the option data cloud. This "outlier" value is likely to influence strongly the regression.

Instead, Shimko [64] assumes that the tail distributions of index prices are lognormal (returns normally distributed) and suggest to match the frequency and cumulative frequency of the distribution with a lognormal distribution in each tail. The idea is attractive, but Shimko [64] provides no further information about the procedure to follow. Therefore, we had to fill in the details by ourselves. Here is how we proceed. For each tail, we have to find the two parameters defining the lognormal pdf subject to the cdf and pdf constraints. We proceed in two steps. First we define two functions that, for a given mean, return the volatility satisfying respectively the cdf constraint and the pdf constraint. Second, we search the value of the mean such that the two values of sigma returned by the two functions are the same.

In order to define the two functions, we suppose that there are enough options to cover a large part of the pdf and that only the tails have to be computed. In this case, for a given value of the mean μ, the following two properties hold. Note that these properties are not true around the mode.

- The normal pdf is an increasing function of σ;
- The normal cumulative density function (cdf) increases (decreases) in the lower (upper) tail when σ increases .

For a given mean, using these two properties and a dichotomic search, we can easily find the two values of sigma satisfying respectively the pdf and cdf constraints. The difficulty is now to find the mean for which these two values are identical. We execute a bracketing procedure. We set a lower bound and an upper bound around the mean. The lowest and highest strike prices are good candidates if the pdf is largely represented by Breeden and Litzenberger's method (we only have to compute the tails). At these two points, the differences of the implied values of sigma (computed as described just before) have an opposite sign. By successively reducing the size of the interval preserving the opposite signs at the bounds, we quickly obtain the mean and the corresponding unique volatility.

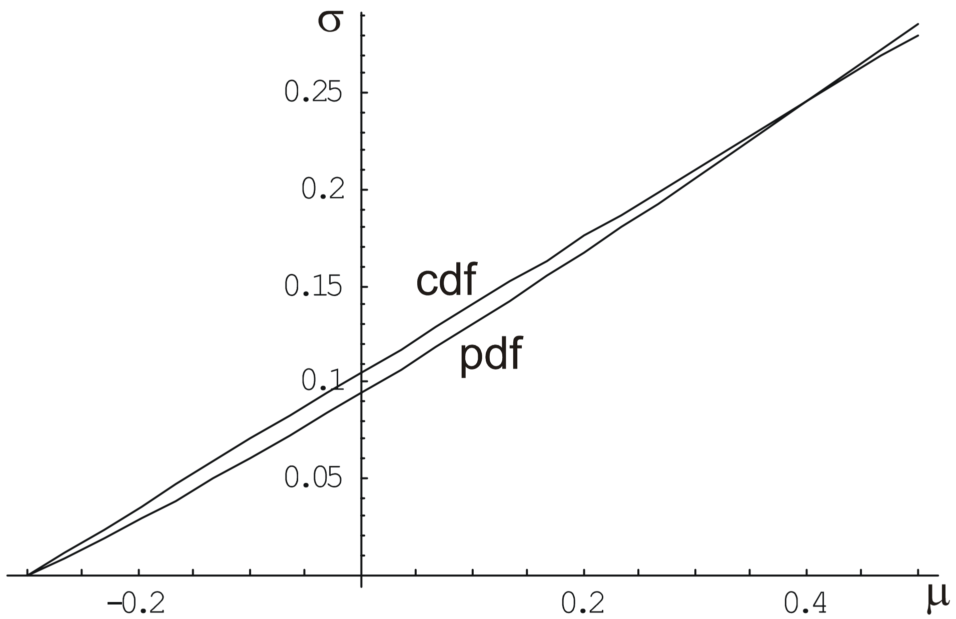


Figure 5.9: Volatilities for a given mean

Some difficulties appear in practice. Some of the numerical results in Shimko's paper are actually bizarre. The sum of densities under the pdf found by Shimko is larger than one, so that he needs to apply a normalizing coefficient to compute the moments. This indicates that the tails he has constructed don't match the cumulative frequencies of the normal distribution as they should. Moreover the first moment, which should be equal to the expected value of the index in a risk-free world as demonstrated by Shimko, doesn't match it. This could be due to an error in the formulae he used to construct the pdf but also to problems to define the tails.

We face such difficulties in our numerical experiments, because matching the frequency and cumulative frequency of the distribution is not an easy task. These two constraints are nonlinear lognormal functions of the volatility for a given mean. There is no proof of the existence and unicity of a solution! Graphically, when index prices follow a perfect lognormal distribution, a solution is easy to find. However, we observe in practice skewness and kurtosis effects. For the S&P500, the tails are fatter than the tails of a normal distribution with the same mean and volatility. Therefore, to obtain the same fat lower tail, we have to shift the mean of the lognormal pdf far from the largest strike price. This is illustrated in Figure 5.10 for the lower tail. In this figure, the horizontal axis represents the index returns and not the prices. Therefore the adjustement is made with a Normal distribution. The conversion from prices to returns is done as explained in section 5.3.5. This is done for consistency with the other distributions presented in the previous sections.

This means that the extreme strike prices are not good bounds as we said before. Therefore we have modified the procedure for computing μ as follows. We start from the extreme strike prices and increase them by steps until the sign of the difference between sigmas changes. It is very important to well tune the bounds over the mean as there exists up to three values of sigma satisfying the pdf constraint for a same mean. The first one is obtained when the index price is the lowest strike price. In this case, sigma goes to zero as it is illustrated in Figure 5.9. This value is clearly not valid. The second one is found when sigma is very large. In this case, the pdf becomes flat and we cannot anymore say that we adjust with the tail of a distribution. The third one is obtained between the two previous one and corresponds to a well shaped distribution.

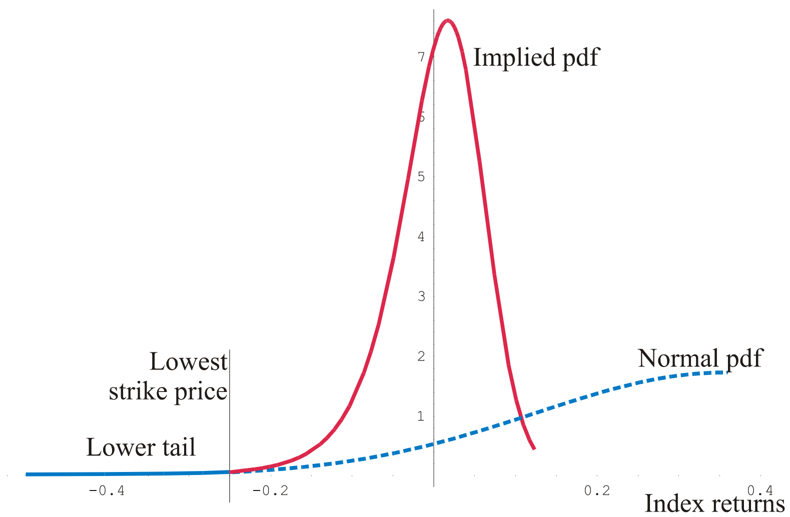


Figure 5.10: Adjusting the lower tail

The pdf as a function of the index returns

We perform a fourth step. As for the distributions presented in the previous sections, to preserve consistency, we would like to work with the distribution of the index returns instead of the distribution of index prices. So, we convert the pdf computed in the first three steps, which is a function of the index prices, into a pdf which is a function of the index returns. For this purpose, we use the general formulas:

$$\begin{aligned} &y = g(x) \\ &h(y) = f(g^{-1}(y))|\tfrac{dx}{dy}| \end{aligned} \tag{5.20}$$

where $f(x)$ and $h(y)$ are respectively the pdfs for arbitrary random variables x and y.

$$\begin{aligned} &\text{Since } return_T = return(S_T) = \tfrac{\ln \frac{S_T}{S_t}}{T-t} \\ &\text{this leads to } h(return_T) = f(S_T)(T-t)S_t e^{return_T(T-t)} \end{aligned} \tag{5.21}$$

where t is the initial time, T is the end of the period, S_t is the initial index price, S_T is the considered value of the index at time T and $return_T$ the corresponding return, $f(\cdot)$ is the pdf for index prices and $h(\cdot)$ is the pdf for index returns,

Risk-neutral world vs consensus world

A fifth step is required. Breeden and Litzenberger's demonstration is based on risk-neutral valuation and the pdf constructed by the implied approach is a certified equivalent risk-neutral world pdf. This pdf is different (even if Rubinstein thinks that the shape should be close and the location just shifted) from the consensus subjective real distribution. The mean is located at the risk-free rate and not at the mean index return and the two volatilities should converge. To construct the tree of scenarios, we still need to introduce a conversion function from risk-neutral world to risky world. This is presented in the next section.

Remarks

We formulate several more remarks. First, consistent with the implied approach, Shimko does not use the historical risk free rate and dividend yield, but instead uses the implied values described in the first section.

Second, it is interesting to notice that the pdfs computed by Shimko (and Rubinstein; see next section) are not always unimodal. A small hump sometimes appears for low returns in their results. We don't observe it in our numerical results.

Third, we can wonder if it is always meaningful to use a normal distribution to construct the tails. The normal pdf used just before to represent the lower tail is difficult to interpret because it is far from the main pdf. If the sum of densities for the pdf obtained by Shimko's method is close to one (99% in our numerical experiment), then the method we use should make little difference. Otherwise, we could face serious troubles as the tails influence directly the moments of the whole distribution. Indeed, a bad representation of the tails leads to a degradation of the implied mean and volatility. Theoretically, the mean should be equal to the risk-free rate. So, alternatively, we would suggest using a Student distribution or interpolate directly with a cubic spline (interpolation of the cdf points with a matching of the derivative given by the pdf) to give more freedom when adjusting the tails. Note however than in our case, even with a good representation (99%) of the central pdf, the expected return can be different from the risk-free rate depending on the option set and the implied rates.

Fourth, another advantage of this method is that we know the analytical form of both the cdf and the pdf. As we will explain further on, the cdf is required to construct future index values and the pdf is only used to compute statistics about the moments. For the distributions presented before, we numerically integrate the pdf to obtain the cdf. However, this numerical conversion is not necessary here.

5.3.6 Rubinstein's implied distribution

Rubinstein has proposed another approach to construct a risk-neutral pdf that matches the observed option prices; i.e. that rejects arbitrage opportunities. We will not apply this method to construct a pdf. But, we present it here because it is a well known method that could effectively be used, and in order to show analogies with the option pricing process we will present further on.

Let S^b (S^a) be the current bid (ask) price of the underlying asset and C_i^b (C_i^a) the bid (ask) price simultaneously observed at time t on a European call i maturing at T. If we know for each scenario j a prior guess P_j' of the risk-neutral probability, then Rubinstein suggests to obtain a discrete representation $\{P_j\}$ of the risk-neutral pdf by optimizing the following model:

$$
\begin{array}{ll}
\min & \sum_j (P_j - P'_j)^2 \\
\text{subject to} & \sum_j P_j = 1 \\
& Se^{r(T-t)} = e^{-q(T-t)} \sum_j P_j S_j \\
& S^b \leq S \leq S^a \\
& C_i e^{r(T-t)} = \sum_j P_j \max[0, S_j - K_i] \\
& C_i^b \leq C_i \leq C_i^a \\
& P_j \geq 0
\end{array}
\tag{5.22}
$$

This model is close to our option pricing model which will be presented in detail in section 6.3.7. The constraints are essentially the no-arbitrage equations. If a solution exists, then there is no arbitrage opportunity on the market due to the observed option prices. These equations involve a unique option price, not two (observed bid and ask prices). Rubinstein allows any price between the bid and ask prices. Indeed, by adding a spread on each side of the optimal price, we can obtain the bid and ask prices without losing the free-arbitrage property. The objective function can be interpreted directly: it simply minimizes the difference between the optimized pdf and the target pdf.

In principle, we can use any prior distribution $\{P'_j\}$. This could be for example Shimko's pdf. Rubinstein suggests to construct an n-step standard binomial tree (under the normality assumption) using the average of the BS implied volatilities of the two call options nearest-to-the-money and then calculate the risk-neutral probability for each of the final node .

Two issues are sensitive. First, what it the best objective function? Rubinstein uses a sum of squares function, but lists other possibilities. Second, what is the best prior guess? For Rubinstein, if a solution exists and other things being equal, then the denser the set of options, the less sensitive P_j will be to the prior guess (more constrained problems, fewer feasible solutions). A relatively large set of scenarios is required to have a fine representation of the future. If we want to reduce the impact of the prior guess, we will also proportionaly need a large set of calls. However, it is difficult to collect and to handle large set of options and so we consider that the prior guess is a relevant problem. This appears clearly in our initial numerical experiments.

We do not directly use this method to construct the pdf due to the two problems noted in the previous paragraph and especially the second one. Indeed, as we expect that the pdf obtained by this approach is closely related to the prior guess, i.e. to a target pdf, we have to construct carefully this target pdf; i.e. we should ideally initially know the optimal pdf we are looking for. There is also a third problem with Rubinstein's approach: he supposes that a solution exists; i.e. that the observed prices are without arbitrage!

Despite these comments, we use in the next chapter a similar approach based on the no-arbitrage equations. The main goal is not to construct a risk-neutral density but to price the options. In this case, we first construct a pdf to model the future and then we use it as a prior guess in the objective function of the option pricing process. The first and third problems are also considered.

5.3.7 Numerical results

Figure 5.11 illustrates the Normal, skewed Student and Shimko's pdf's for the S&P500. For the Normal and skewed Student pdfs, the parameters were computed from monthly returns over the last 10 years. Shimko's pdf's was computed from a set of options observed on the 18th of March 2000 and on the 23rd of February 2001 (with maturity one month later). The horizontal axe corresponds to the monthly returns.

The skewed Student pdf is close to the Normal one, but is negatively skewed and slightly leptokurtic. There is little difference using the first or the second one. At the converse, the shape of the implied "instantaneous" pdfs is very different from the "10 years" pdfs and has even significantly changed over a period of one year. However, we cannot draw many conclusions for the moment as the implied pdfs are defined in the context of a risk-neutral world and the two first pdfs in the context of a consensus world. More comments are done in section 5.6.

5.4 Sampling

5.4.1 Introduction

In this section, our goal is to define the index return (or equivalently the index price) for each node of a multinomial tree. In the previous section we used a continuous pdf to model the index return. We need a method to sample from such a continuous function in order to produce a representative discrete set of returns.

5.4.2 Monte-Carlo generator

The most straighforward method is to use a random Monte-Carlo generator. We have implemented the generator described in Numerical Recipes [61]. The principle is first to use a uniform random Monter-Carlo generator and then to convert the uniform deviates into deviates following the target distribution. In the case of normal distribution, we first draw

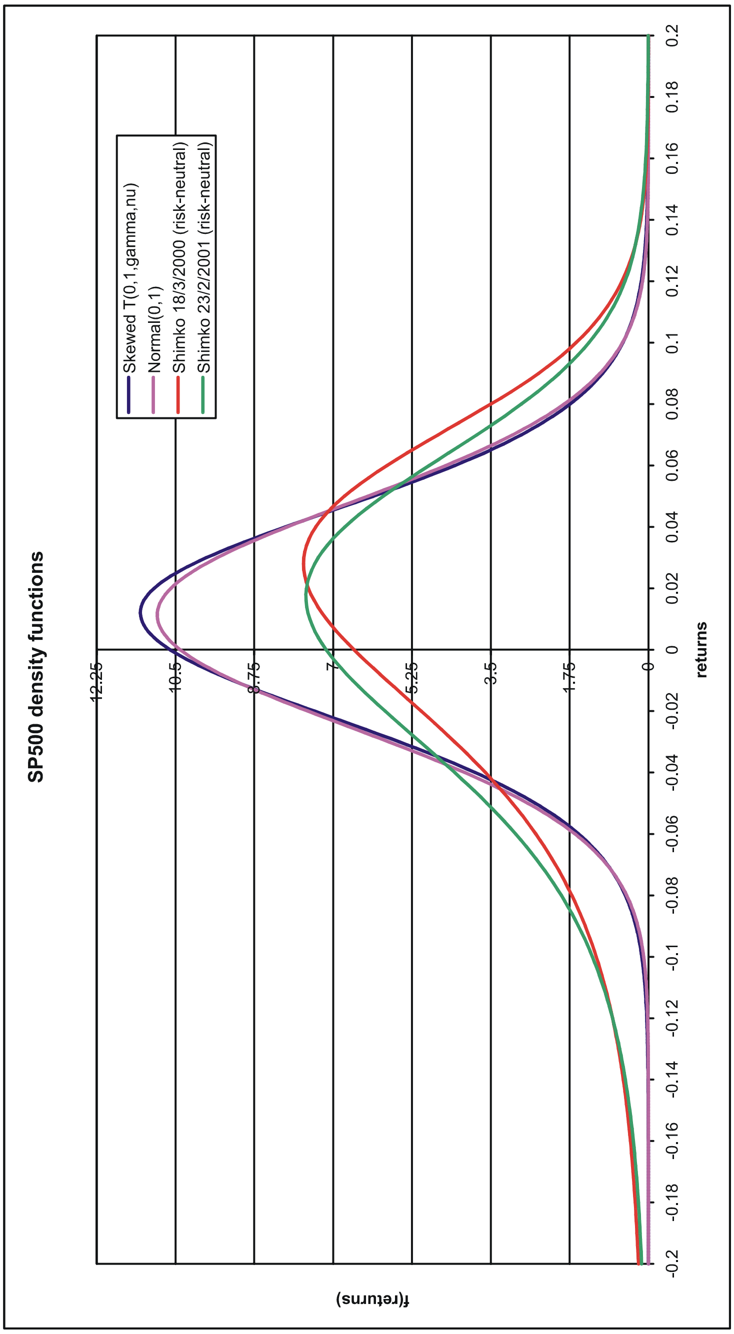


Figure 5.11: Probability density functions

at random two uniform deviates x_1 and x_2 between zero and one and we convert them into two normal deviates y_1 and y_2 by the trigonometric transformation:

$$
\begin{aligned}
y_1 &= \sqrt{-2\ln x_1}\cos 2\pi x_2 \\
y_2 &= \sqrt{-2\ln x_1}\sin 2\pi x_2
\end{aligned}
\tag{5.23}
$$

A seed is needed to initialize the pseudo-random sequence and a different sample of the Normal distribution is obtained each time a different seed is used. For someone who does not know the initial seed, the sequence looks as random.

Note that when generating n return estimates, according to a distribution with mean μ and standard deviation σ, the standard error of the estimate is $\frac{\sigma}{\sqrt{n}}$. Therefore, to double the accuracy, we must quadruple the sample size.

5.4.3 A grid generator

We propose here a small improvement over the previous generator. The Monte-Carlo generator needs two uniform deviates and so resorts to a uniform generator. Our goal is to obtain the sample of a given size (the number of scenarios) for which the moments are as close as possible to the moments of the continuous Normal distribution to represent. We are not interested in the random property of this sample. The idea is to replace the random uniform generator of the previous method by another (deterministic) uniform generator of better quality.

This type of method is described in Kleijnen [38, 39]. If we know in advance the sample size n, the best possible representation of the uniform distribution is obtained by dividing the range into n equally spaced points. Other methods have been developed for unknown sample sizes (Antonov&Salev [38]).

The situation is a little bit more complex in our case because we need two uniform deviates at the same time in order to construct two normal deviates. Therefore, we work in dimension two instead of dimension one. A direct adaptation is to lay a grid (equally spaced) on the $(0,1) \times (0,1)$ area, and to interpret the coordinates of the center of each subsquare as defining the two uniform deviates. We face the problem that there is a square number of subsquares, which is not necessarily the case of the sample size. We use a grid of size slightly in excess of $\sqrt{n} \times \sqrt{n}$, and remove afterwards superflous deviates.

5.4.4 Stratification

A method to convert a continuous pdf into a discrete set of values is presented in Hull[32] under the name "stratified sampling". Hampel *et al.* [31] used the name "stylized sampling" for a similar method. Curran [13] developped this approach for option valuation.

In order to stratify the continuous distribution, we partition the area under the density curve. Each zone will correspond to one leaf of the tree of scenarios. A natural idea is to define equiprobable zones (equal areas). In this case, all the leaves are equiprobable, each future state of the world has the same weight and information is fairly distributed. In each zone, one value is chosen to represent the whole zone. Thus, to obtain a set of nbS leaves, we have to define nbS zones, each with the probability given by the area of the corresponding zone ($1/nbS$ for an equiprobable tree).

To define the partition, we need to be able to compute the areas below the density function; i.e. to compute the cumulative distribution function. The density functions we will use are very complex and it is not always possible to analytically describe the cumulative function. When we don't know the cdf analytically, we use Romberg's numerical method [61] to compute the integral of the given density function.

When we know the cdf, we still need to find the lower and upper bounds of each zone. Of course, the upper bound of a zone is the lower bound of the following one. So, we just need to find the upper bound of each one, as well as the lower bound of the very first zone. The distribution of returns range from minus infinity to infinity. However, numerically, the value minus infinity is not defined and a huge negative number cannot be used due to numerical problems (domain of definition) with the logarithmic functions used to define the density functions. This problem is easily solved by performing a quick search in the negative domain to find an adequate value for which the density function can be considered as null. The same is performed for the largest upper bound. When computing the upper bound of each zone, it is not possible to invert the numerical cumulative function in order to obtain a function of the required probability giving as result the corresponding return. Therefore we perform a dichotomic search over the returns.

When the zones are defined, we have to choose a representative return for each one. Several choices are possible: the mean of the area, the median of the area, the mean of the bounds, the upper bound, the lower bound... Each is a possible representative, but some are perhaps better than others. For example, in the case of the symmetric normal distribution, using the mean for each zone preserves the global mean. By contrast, using a bound shifts the global mean. However computing the mean of a zone could be troublesome because if we don't know the analytical expression of the mean, we again need to compute a numerical integral over a small area. The median is easier to compute but also needs a dichotomic search to divide the original zone into two equiprobable areas. The easiest method is to use the mean of the bounds. In the rest of the work, we prefer to use the mean of each zone.

The stratification process is illustrated in Figure 5.12. The vertical plain lines represent the partition in equiprobable zones under the density curve and the dashed lines represent the mean of each zone.

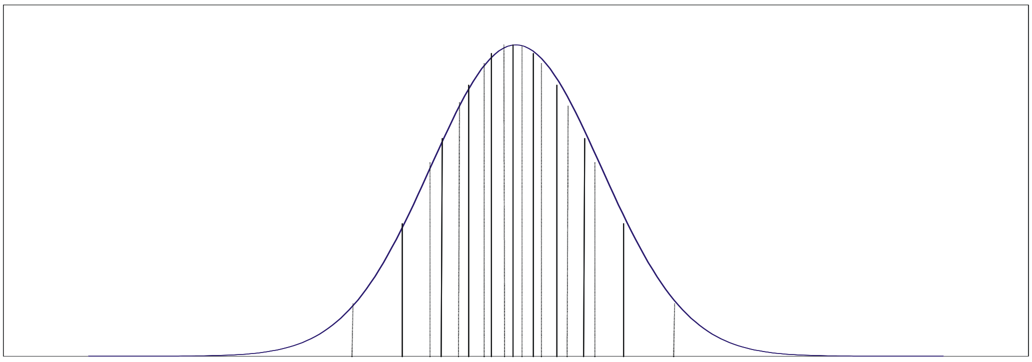


Figure 5.12: Stratification

5.4.5 Quality of the sample

To check the quality of the stratification, we have computed the moments of several density functions, both from their continuous representation, and from their discretization. We applied the Wilk-Shapiro test to compare the samples with the theoretical Normal distribution. We have not here defined other tests to compare the sample with other possible distributions.

The stratification method gives an excellent representation of the distribution. The mean of the sample exactly equals the theoretical mean by construction. The standard deviation quickly tends to its theoretical value. The disadvantage of the method is the increased difficulty to compute it with respect to a random generation, but numerically it remains an easy problem that is far much easier to solve than a portfolio optimization problem like the one described later.

This method also has the major advantage to give extremely good representations of a distribution even for a small number of sampled points. A random Monte-Carlo generator requires larger sample size to reach the same quality. Therefore, the stratification method allows to reduce the size of the problems to optimize.

5.4.6 Numerical results

Figure 5.13 illustrates Shimko's pdf for the S&P500. Thirty equiprobable areas are defined and the mean of each plotted.

Numerically, for each continuous probability density function described previously, table 5.24 describes the moments for a stratification in 30 equiprobable zones. The results of the Wilk-Shapiro tests are also included. Those give the probability that the sample corresponds to a Normal distribution. The first line indicates the historical moments extracted from the database. The risk-free rate is equal to 0.42% and the dividend yield to 0.096% for all models. For the two implied pdf's, the tails constructed thanks to the normal density function represent about 10% of the probability distribution.

Consensus world					
	μ	σ	skewness	kurtosis	WS
10 years historical data	1.131%	3.658%	−0.214	3.816	
Normal pdf	1.131%	3.658%	0	3	
Normal Monte-Carlo[1]	1.156%	3.616%	0.015	2.546	100%
extreme values	−0.422%	2.629%	1.095	4.533	100%
Normal stratification	1.131%	3.679%	0	2.490	100%
Skewed T pdf	1.131%	3.658%	−0.025	3.612	
Skewed T stratification	1.131%	3.694%	−0.025	2.710	100%
[1] mean values over 100 simulations					
Risk-neutral world					
	μ	σ	skewness	kurtosis	WS
implied pdf 18/2/2000	0.448%	6.49%	−1.183	5.640	
implied strat. 18/2/2000	0.448%	6.43%	−1.027	4.414	12%
implied pdf 23/2/2001	0.165%	5.927%	−0.721	4.129	
implied strat. 23/2/2001	0.167%	5.885%	−0.622	3.396	32%

(5.24)

The stratification process preserves quite well the first four moments. The fourth moment is not perfectly estimated, but still close to the moment of the continuous distribution. Better estimation could be obtained by increasing the sampling size (only 30 scenarios for this example).

In the case of the Normal distribution, we can compare the moments of the stratified sample with the moments of the random sample. We have generated 100 random samples and computed the mean of the four moments. In mean, the fourth moments are well preserved

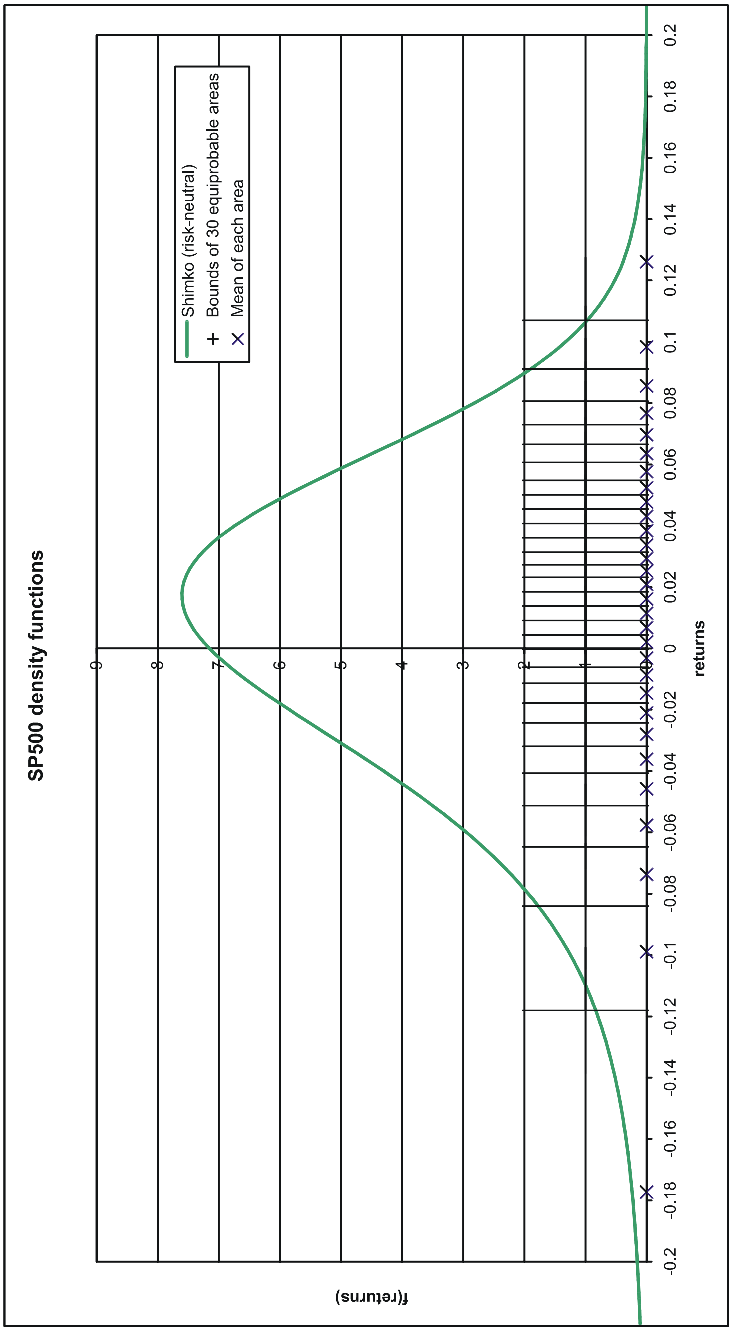


Figure 5.13: Stratification

and close to the moments of the continuous distribution. However, the moments vary greatly from one random sample to another. When we look at the extreme values encountered, the fourth moments can be very different from the moments of the continuous distribution. At the converse, the stratified sample, obtained by the deterministic process presented before, matches the moments.

Note also that according to the Wilk-Shaphiro test, all the samples constructed from the Normal distribution and from the skewed t distribution are considered normally distributed. This shows that we should be very carefull when interpretating such statistical probability tests. Indeed, the sample size is probably too small to draw rigorous conclusions. Note however that for the implied pdfs, which have a very different shape, the Wilk-Shapiro test rejects the normality assumption.

5.5 Probability conversions

5.5.1 Introduction

We want to model the future with a multinomial tree of scenarios representing the consensus world. Indeed, our final goal is to model a portfolio problem subject to consensus constraints. So, in order to define the future index returns in each leaf, we will stratify a consensus pdf.

If we consider that the index returns in the world follows a Normal distribution or a skewed t distribution, we can use directly the pdfs described before. Otherwise, if we construct an implied risk-neutral pdf from the options, then we need to define a translation method in order to convert this risk-neutral pdf into a consensus pdf.

Conversely, it is also usefull to define a conversion method from consensus pdfs to risk-neutral pdfs. Indeed, as explained in chapter 4, the risk-neutral probabilities, or their close cousins the state-prices, are intensively used in finance. In particular, we will use them in this work to improve the quality of the option pricing process (section 6.3.7) and to define heuristics to solve portfolio problems (sections 8.5 and 9.3).

5.5.2 From risk-neutral to consensus probabilities

Model

We present here some of Rubinstein's results [63] to convert risk-neutral probabilities into consensus ones. There is no such direct tool. However, as Rubinstein notes about his numerical results and the implied tree, "(...) despite warnings to the contrary, we can justifiably suppose a rough similarity between the risk-neutral probabilities implied in option prices and subjective belief. In the diffusion continuous-time limit, the move volatility calculated from risk-neutral probabilities and the move volatility calculated from the "true" market-wide (consensus) subjective probabilities converge to the same number as the move size approaches zero. However, this is not true for the mean. The risk-neutral mean for a single move is obviously the risk-free rate". According to these results, the subjective probabilities could be obtained by shifting the risk-neutral ones by the difference between the risk-neutral rate and the expected subjective return.

More formally, again according to Rubinstein's conclusions, it is possible to do the conversion in a fully specified utility theory framework. Rubinstein gives a classic example ([63]). He considers a complete market economy with a representative investor with constant relative risk aversion who maximizes the expected utility of his terminal wealth subject to the usual budget constraint that he invests all his wealth. Assume that if the investor has a initial wealth of one, and a utility function $U(\cdot)$. Let R_i be the return associated to the scenario i and let q be the continuous dividend yield. Then the investor chooses R_i by solving the following Lagrangian problem:

$$\max \sum_i Q_i U(e^{qT} R_i) - \lambda(\sum_i \frac{P_i}{e^{rT}} e^{qT} R_i - 1)$$

where $Q_i > 0$ is the subjective probability and $P_i > 0$ is the risk-neutral probability ($\frac{P_i}{e^{rT}} = \psi_i$) associated to scenario i and T is time to the end of the period.

Solving the first order conditions that arise after differentiating with respect to R_i:

$$Q_i = \lambda \frac{P_i}{e^{rT} U'(e^{qT} R_i)} \tag{5.25}$$

where

$$\lambda = \frac{1}{\sum_i \frac{P_i}{e^{rT} U'(e^{qT} R_i)}}$$

Assuming a given expression of the utility function (e.g. a logarithmic utility function $U(x) = \ln x$ or a power function $U(x) = -(x)^{-0.65}$) and knowing the returns R_i and risk-neutral probabilities P_i for each scenario i, it is now possible to obtain the subjective probabilities Q_i.

Equiprobable trees

Thanks to the set of equations (5.25), we can convert the real probabilities associated to a set of returns into risk-neutral probabilities. Risk exists on real markets and so the resulting probabilities are not equal to the original ones. This also implies that if the returns have equiprobable risk-neutral probabilities, the subjective ones are no longer equiprobable. So, when we stratify the implied risk-neutral pdf into equiprobable zones, this results in a non-equiprobable consensus sample and we cannot use it to construct an equiprobable tree. Four possibilites are considered now to handle this case.

First, we could do nothing. We decided to use an equiprobable tree to simplify the explanations, but in fact, this is not a must. We could work without any problem with arbitrary set of probabilities. By this way, without any other data manipulations, we can use directly the consensus sample to model the future and the risk-neutral pdf to price the options.

Second, we could approximate the consensus continuous pdf, not using equations in (5.25), but by shifting the risk-neutral distribution to the real historical mean. The equiprobable returns would then be obtained by stratification. This is not a careful approach as the validity of the shift translation it not proved but is only an assumption (Rubinstein [63]).

Third, we could approximate the consensus cumulative distribution function using equations (5.25). The first step is to convert the risk-neutral continuous pdf into a discrete set of (equiprobable) returns as well as possible. The larger the set, the better the representation. The second step is to convert the probabilities into subjective probabilities using equations (5.25). The third step is to perform partial sums of the equiprobable probabilities for each of the consensus returns of the sorted list so as to obtain an approximated cdf. A linear interpolation can be used to define the intermediate returns. When the size of the initial sample tends to infinity, we get the perfect continuous distribution. Finally, a stratified equiprobable sample can be drawn from this cdf.

Fourth, we could try to obtain an exact solution by inverting equations (5.25). If we set each Q_i to the equiprobable probability $1/nbS$, then we obtain a set of nbS nonlinear equations with $2\,nbS$ unknowns. Moreover, each R_i is a function of P_i by stratification of the risk-neutral cdf considered. We can add nbS nonlinear equations to fully define the system. It is extremely difficult to solve systems of more than one nonlinear equation. We could try using Newton-Raphson's method starting from the approximation presented just above. The difficulty to work with the equations (especially the last ones based on stratification) and the solving method itself (requiring derivatives) make this approach very time consuming.

Moreover, we have no guarantee that a global optimum will be found. For these reasons, we prefer to first concentrate on the previous approximation scheme.

5.5.3 From subjective to risk-neutral probabilities

The approach

The subjective pdf is required to construct a real tree, but the risk-neutral one is also useful in some contexts, e.g. in the option pricing process. If we have constructed the consensus pdf from the implied risk-neutral pdf, we have the advantage to possess both. Otherwise, when we work with a Normal distribution or with a skewed Student distribution, we have to define a translation algorithm if we need to work with the risk-neutral world. We will not consider here a translation tool between the continuous functions, but between discrete representations. This is not a restriction because our goal is to work with discrete multinomial trees of scenarios.

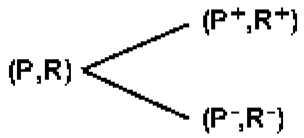
To obtain (prior guess) risk-neutral probabilities starting from subjective probabilities, Rubinstein constructs a standard binomial tree and computes the state-prices (or discounted risk-neutral probabilities) associated with each leaf. We cannot construct such a multiperiod binomial tree because, as demonstrated in section 5.1.4, it is not possible to match the index prices at the leaves of a multinomial equiprobable tree with the index values at the leaves of a standard multiperiod binomial one. In fact, such a matching does not exist if all parameters of the binomial tree are required to remain constant at each period, but it exists if we allow a variation of the parameters (increase u and decrease d) for each possible binomial subtree. Rubinstein [63] developed a related approach in another context and with other inputs.

For a set of possible future index prices and corresponding risk-neutral probabilities, Rubinstein [63] shows how to construct a variable multiperiod binomial tree in which each leaf matches one of the elements of the set. From this tree, he can compute several interesting option parameters. Note however that he needs as input the risk-neutral probabilities we are looking for, whereas our goal is to compute them. Our idea is to develop a similar algorithm but where we replace the risk-neutral probabilities by the consensus subjective ones as obtained from our pdf. Rubinstein's process requires probabilities to work on, but is not able to distinguish between real and subjective ones.

Once we have obtained a multi-period binomial tree with leaves corresponding to the future index prices for given real probabilities, we can use it to compute the corresponding required risk-neutral probabilities by following each path to the leaves and applying at each node the binomial formula for risk-neutral probabilities. Under Rubinstein's assumptions, there is only one implied tree and so only one risk-neutral probability distribution. However, some of these assumptions were artificially fixed to allow the unicity of the solution.

Rubinstein's implied tree

Rubinstein constructs a recombining binomial tree where the ending nodal values are ordered from lowest to highest. For each binomial subtree, by working backwards, we have to find the adequate increase and decrease subperiod returns u and d and the corresponding probabilities in agreement with the final returns and the final risk-neutral probabilities. To define completely the process, we arbitrarily decide that the risk-free rate is constant per unit of time. In short, the solution is a "One-Two-Three" procedure, as Rubinstein says :



1. $P = P^{-} + P^{+}$
2. $p = P^{+}/P$
3. $R = ((1-p)R^{-} + pR^{+})/r_b$

where P^{+}, P^{-} are the path probabilities, R^{+}, R^{-} are the node returns and r_b is the risk-free rate over the period.

We quote from Rubinstein: "To start everything rolling, go to the end of the tree and attach to each node its nodal value R_j and nodal probability P_j. Now take each ending nodal probability and divide it by the number of paths to that node to get the path probability, which is in general:

$$P_j/C_n^j$$

Also, define the interest return r_b as the nth root of the sum of P_jR_j, so that:

$$r_b^n = \sum_j P_j R_j$$

"

The first step shows that the probabilities at the leaves of the binomial subtree sum to the probability at the root. The second step defines the upward path probability. The last step defines the return at the root, in a risk-neutral world, as the discounted return expected value.

Target state-prices ψ'_j

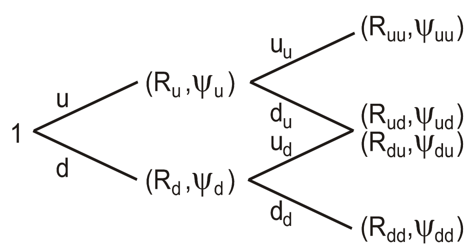
As mentioned above in our application of Rubinstein's method, we replace risk-neutral probabilities by real ones. The first step still splits the probabilities between the two leaves. At the second step, p is the subjective probability of an up move. Finally, the weighted sum is the expected return at the end of the subtree in a real world and no longer in a risk-neutral one. The discount factor r_b is no longer the risk-free rate, but has to be interpreted as the expected return required for the period in this real world. R is the interior nodal value of the return in the true and risk-neutral worlds. In this way, we obtain a binomial tree where we know the return value for each node.

Let us now see how we can compute the state-prices associated to the tree. By definition of the state-prices, we have:

$$\begin{pmatrix} 1 \\ R \end{pmatrix} = \begin{pmatrix} e^{r_b} & e^{r_b} \\ R^- & R^+ \end{pmatrix} \begin{pmatrix} \psi^- \\ \psi^+ \end{pmatrix} \tag{5.26}$$

In this way, we can compute the state-prices at each node. For each possible path to a leave, we multiply the nodal state-prices encountered. By summing all these products along paths leading to a same leaf, we obtain the final state-price of this leaf. It directly gives the risk-neutral probability associated to the leaf return.

It may be tempting to think that, as the binomial tree is recombining, we just have to compute the final-state price along one complete path and to multiply it by the number of paths leading to the node. But this is false! Rubinstein has formulated an assumption that all paths that lead to the same ending node have the same risk-neutral probability. In our approach, risk-neutral probabilities are replaced by real ones and the risk-neutral probabilities that we reconstruct do no longer satisfy this property. Consider for instance the following two-period binomial tree:



Rubinstein's implied tree is recombining ($R_{ud} = R_{du}$). It was possible to construct it because the increase and decrease factor are not constant ($u \neq u_u \neq d_u \neq d \neq u_d \neq d_d$) for each period as for a classical binomial tree. As these coefficients define the nodal returns and the nodal returns define the state-prices, there is no reason to obtain constant state-prices over time. So, for the computation of the final state-prices, an up move followed by a down move is not equivalent to a down move followed by an up move:

$$\psi_u \psi_{ud} \neq \psi_d \psi_{du}$$

We have shown that already for a 2-step binomial tree, depending on the path followed to a same leaf, we obtain products of nodal state-prices that are different. Of course, when the number of stages increases, we obtain more and more different paths. So, we cannot compute the product along one path and multiply it by the number of paths.

The sole solution is to consider every possible path and to perform the products along each of them. Of course, some shorcuts can be implemented. The partial product obtained at a given intermediate node can be used for all paths sharing the same initial path to this node, splitting only afterwards. A recursive procedure is then used to compute the partial product only once at each node of the tree. This is easy to do, but it gives rise to an intensive numerical process. If we consider only 30 scenarios at the end of the tree, the implied binomial tree consists of $2^{30+1} - 1$ nodes; more than two thousand million products to perform! Moreover, we want to model two periods; i.e. 31 subtrees of 30 scenarios. Fortunately, as the same process with the same parameters is used to create all the subtrees of the second period, we only have one implied tree to consider to represent the 30 subtrees of the second period. It is even better when the length of the first and second periods are identical then only one implied tree is required. In this ideal case, the process takes between 5 and 10 minutes on a PIII600 computer. There is an explosion of the required CPU time if the sample size is increased as the size of the implied multiperiod binomial tree grows very fast.

Results and remarks

By this procedure we were able to convert a subjective probability distribution into a risk-neutral one. This was done by permuting the risk-neutral input with the real one. Could we do the inverse using the same method : convert a risk-neutral pdf into a real one? The answer is clearly not. The first conversion was possible because we know a relation between the risk-neutral probabilities and the returns: the free-arbitrage equations. But there is no such relation between the real probabilities and the returns.

5.5.4 Numerical results

Risk-neutral to consensus

Figure 5.14 and Table 5.28 show how the implied risk-neutral pdf measured on the 18th of March 2000 was converted into a consensus distribution. The three real pdf's were obtained respectively by shifting the risk-neutral one, using the logarithmic utility function and the power utility function. The moments are computed over a sample of 400 returns obtained by stratification (third method discusses in section 5.5.2).

	μ	σ	skewness	kurtosis
risk-neutral	0.449%	6.500%	−1.176	5.576
consensus shift	1.131%	6.501%	−1.171	5.548
consensus ln	0.856%	6.265%	−1.139	5.587
consensus power	1.105%	6.124%	−1.112	5.571

(5.27)

We obtain results similar to Rubinstein [63]. There are slight differences between the three consensus pdfs, but the shapes are very close. Moreover, the shapes of the consensus pdfs are close to the risk-neutral one. As already mentioned by Rubinstein, the main difference between a risk-neutral pdf and the corresponding consensus pdf is a modification of the first moment.

In this sense and as the index return is equal to 1.131%, the logarithmic utility function gives the worst results. Also, we prefer not to use in the rest of this work the shifted consensus pdf. Indeed, we prefer not to base the validity of this approach on only one numerical experiment and on beliefs. So, the last consensus pdf and transformation method have our preference. We have now all required information to compare all the pdfs presented in this chapter. This is done in the final section 5.6.

By the way, note also that the stratification of the shifted consensus pdf in 400 zones has preserved perfectly the last three moments of the distribution (and also the first one but it cannot be seen in Table 5.28 due to the shift of the mean). This illustrates that an increase of the sample size leads to improvement of the representation of the moments. Indeed, we could have wondered if the fourth moment could be exactly represented by a finite sample.

Real to risk-neutral

Table 5.28 gives the results of the conversion of consensus pdf's into risk-neutral ones. Figure 5.15 illustrates the case of normal distribution. The moments are computed over a sample

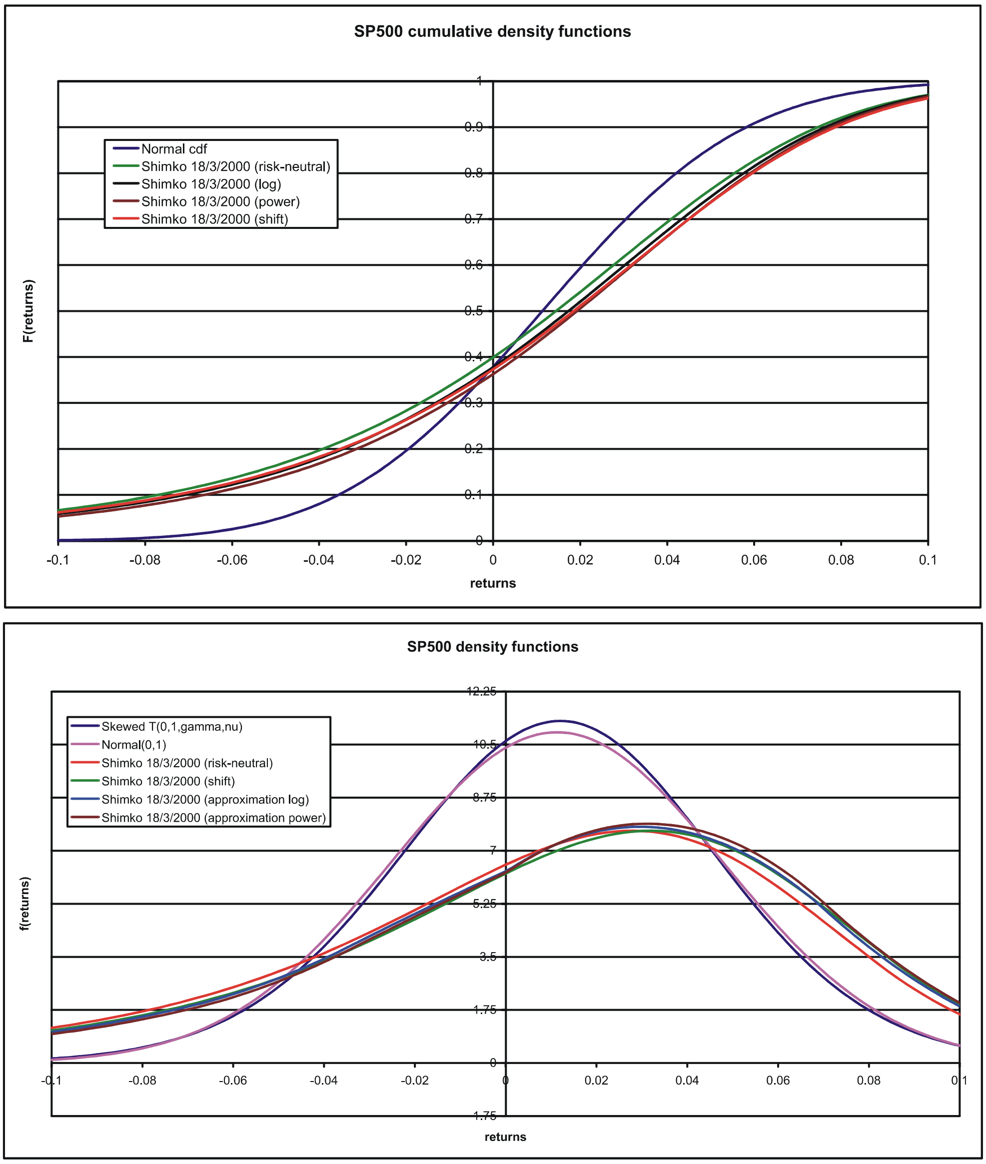


Figure 5.14: Risk-neutral to consensus world

of 30 returns obtained by stratification. Larger sets of returns require too much time to construct the adapted Rubinstein's implied tree.

	μ	σ	skewness	kurtosis
consensus normal pdf	1.131%	3.679%	0.000	2.487
risk-neutral normal pdf	0.355%	3.606%	0.104	2.668
consensus skewed T pdf	1.130%	3.694%	−0.025	2.712
risk-neutral skewed T pdf	0.354%	3.622%	−0.075	2.901
consensus power implied pdf	1.048%	6.224%	−0.961	4.160
risk-neutral power implied pdf	0.236%	6.128%	−0.947	4.242

(5.28)

Once again, we observe here that the shape of the distributions is preserved, but that the mean is reduced to come closer to the risk-free rate. However, there is not a perfect match between the expected return and the historical risk-free rate (0.42%). This is even more visible for the implied distribution for which the double conversion (first risk-neutral to consensus then consensus to risk-neutral) returns approximately to the original shape shifted to the left (lower mean). Small differences in the last moments are essentially due to the reduced size of the stratification. This is probably due to an underestimation of the risk-free rate by the conversion process, but could also be partially due to an inadequate computation of the risk-free rate. As the difference is not huge, that this conversion process is only used later to illustrate some cases with only the Normal distribution and that this risk-neutral pdf is only used as a prior guess in the option pricing process, we do not care too much with the difference.

5.6 Numerical results and conclusions

In this section, in order to compare the different pdfs we can define to construct the tree of scenarios, we put together the partial results obtained in the previous parts of this chapter. In section 5.3, we have shown how to construct continuous pdfs to model future index returns. The two first pdfs, the Normal pdf and the skewed Student t pdf, are constructed using historical data and are correspond to a consensus world. At the opposite, the implied pdf is based on option prices observed at time $t0$. This last pdf is defined in a risk-neutral world. In section 5.4, we have proposed methods to sample those continuous pdf into discrete sets of values in order to instantiate the tree of scenarios. Finally, in section 5.5, we have presented methods to convert the risk-neutral pdfs into consensus ones and the converse. In particular,

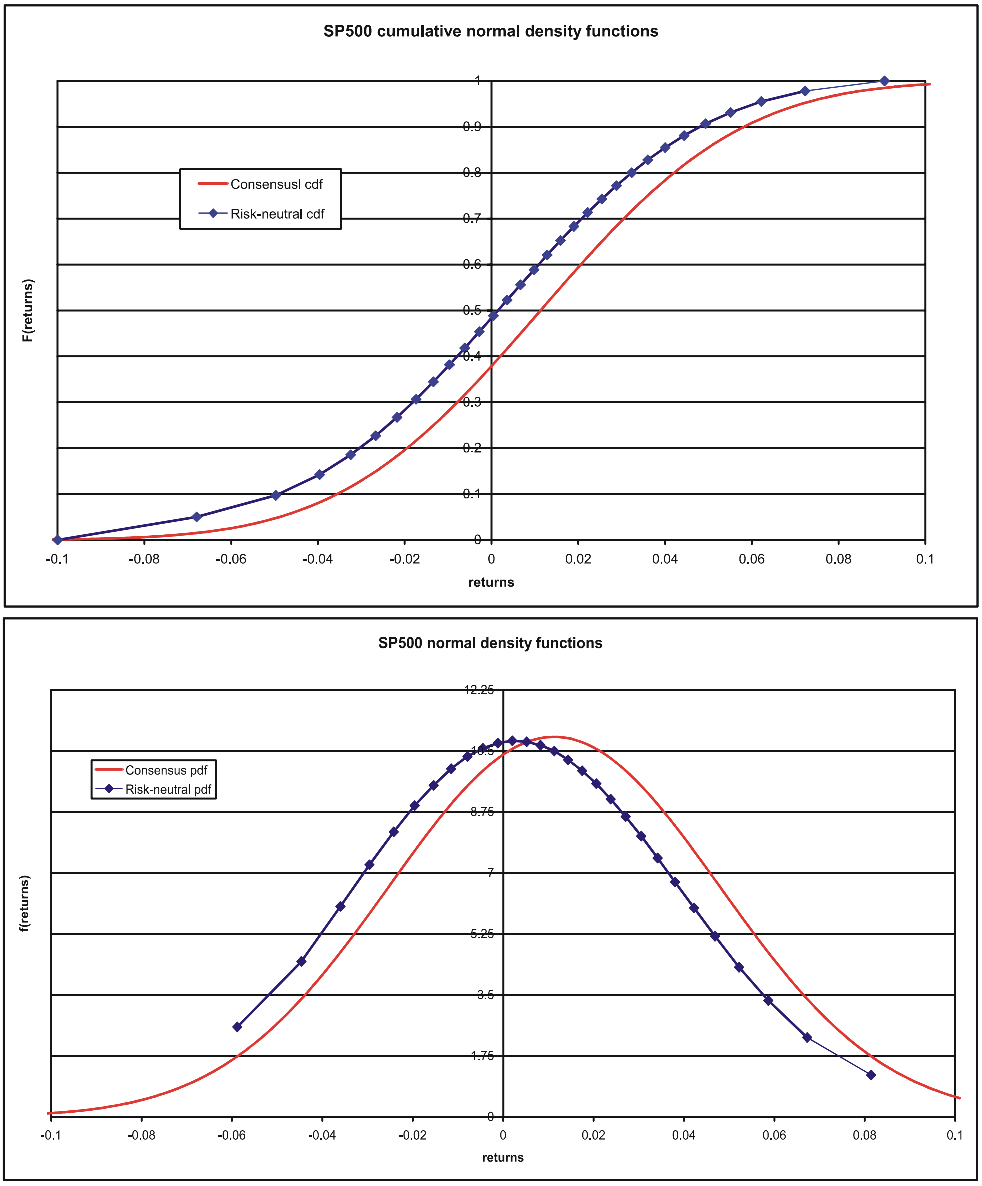


Figure 5.15: Consensus to risk-neutral world

if we want to use the risk-neutral implied pdf to construct the consensus tree of scenarios, we first need to convert this pdf into a consensus one. Numerical results corresponding to these topics were presented at the end of each of the corresponding sections.

For our S&P500 example, Figure 5.14 represents, in the consensus world, all the pdfs we could use to model the future. As explained in section 5.3.7 and from numerical values in Table 5.24 of section 5.4.6, there is little difference between the Normal probability distribution and the skewed Student t distribution. As the skewed Student t distribution family includes the Normal probability distribution, we should prefer to use the skewed Student t pdf. However, in the context of a theoretical study based on normality assumptions, we nevertheless should use the Normal pdf in order to be consistent.

Also, there is little difference between the consensus pdfs obtained from the risk-neutral implied pdf. As stated by Rubinstein, the conversion consists essentially in a shift of the shape to let change the first moment from the risk-neutral rate to the consensus expected return. This is illustrated graphically in Figure 5.14 and numerically in Table 5.28 of section 5.5.4. As explained in this section 5.5.4, we have a preference for the power utility function between the three methods proposed to convert the risk-neutral pdf.

Should we use one of the first two pdfs or the implied one? It is important to remember that the first two pdfs are computed from past data. For the S&P500, the moments, which define the two probability distributions, are computed over a 10-year period. This approach cannot model all the variations in the returns during subperiods but only tracks the trend over the 10 years. Therefore, when we consider short term periods in the tree of scenarios and when we use these parameters, we make first an approximation and second a strong hypothese. First, we probably miss the short term variations and second, we also consider that the long term past trend represents well the short term future. Of course, there exists some econometric approaches, e.g. the GARCH model proposed by Lambert and Laurent [42], to better predict the parameters of these pdfs. By contrast, the implied pdf is based on instantaneous information: the option prices at $t0$. As mentioned before, the current option prices reflect what the investor expects today about the future index returns. Figure 5.11 shows clearly the difference between the two "historical" pdfs and implied ones. We see also in this figure, that the shape of the implied pdfs change depending on the period. So, this approach already seems more appealing.

Moreover, the implied pdf is not a parametric function defined by only a few moments. It can take any shape according to the option prices. Rubinstein [63] and Shimko [64] even obtained implied pdfs with two modes. This cannot be represented by the first two pdfs.

Finally, there is one important more reason to use the implied pdf, but it does not yet appear in these numerical results. The implied pdf is defined according to the options observed on the market. Therefore, there is a strong relation between both of them and so the pdf and the options define a coherent set. At the opposite, we will show in the numerical results about portfolio optimization, that using at once a Normal pdf and market option prices lead to incoherencies and abnormal returns.

For these reasons, we prefer to use an implied pdf. However, we make the same comment as in the second paragraph: in the context of a theoretical study based on normality assumptions, we nevertheless should use the Normal pdf in order to be consistent

Chapter 6

Modelling option prices

6.1 Introduction

We would like to consider not only indices and stocks in the tree of scenarios but options as well. Large investment societies intensively use options because options are very powerful tools. Thanks to their low prices, options provide financial leverages and also allow to precisely shape portfolio values and insure portfolios, due to their typical piecewise linear payoff pattern.

Options are complex tools. They are defined by several parameters which depend on the underlying asset. We have already presented the main characteristics of options in the introductory chapter on finance. To go further in the modelling process, we will now develop this subject in the next section.

One of the main modelling issues when working with options is to define their price. If we are interested in the option price today, we can simply oberve it on the market. We can also use the Black and Scholes formula or construct a binomial tree; which link the option price to the price and the distribution of the underlying asset as mentioned in Chapter 4. It is likely that the three corresponding values will be different. This is due to the hypotheses behind each model : continuity, normal return distribution, market... It is also likely that these three values will not be coherent with a tree of scenarios model, because the hypotheses underlying the construction of the tree and the computation of the option prices are not the same. Section 6.3 will specify the conditions of no-arbitrage to satisfy in order to obtain valid prices for the scenario model.

The conditions stated to price options in the case of a multinomial tree of scenarios are not enough to uniquely define the option prices, as is the case for the binomial tree approach. We will present some additional constraints and objective functions to improve the quality of the prices and to fit closer to real market observations. This is explained in the next subsections of Section 6.3.

We have discussed until now a one-period option pricing model. However in order to obtain more flexibility, we would like to work with multi-period multinomial trees of scenarios. This leads to several difficulties because the last periods depend on the previous ones; so available information is partitioned over the whole tree. In Section 6.4, we present how to compute prices for a multi-period tree. More generally this section is devoted to numerical optimization issues. Exact methods and heuristics are proposed there.

So Sections 6.3-6.4 explain how to compute option prices in a multi-period tree of scenarios. The quality of the results depends on our ability to model the future. Especially, it is more difficult to model option prices for periods far in the future than for the initial one, since we have less information. The quality of the representation of the risk-neutral world for each period is determinant to solve this problem. Section 6.5 explains how to define the risk-neutral probabilities as well as possible in agreement with the observed option prices and the distribution of the underlying asset. As a result, the optimization process returns information about the risk-neutral world that will be useful to model real problems, especially the Value-at-Risk model presented in the next chapter.

We have considered several models to price options and represent the future. We need to check the quality of the solutions. Also, to improve the results or simply to help the investor, it could be useful to perform some option cleaning processes. This is developed in Section 6.6.

Finally, in Section 6.7, some numerical results illustrate the theoretical developments explained in this chapter.

6.2 Definitions

6.2.1 Introduction

All options cannot be handled the same way. To be able to fully define the model, we have to review some characteristics of options not presented in Chapter 4. In the following subsections, we will present the most interesting contract specifications of the exchange-traded options and consequences for the scenario model. As stocks and indices do not have the same characteristics and behaviors, some distinctions must be made. More details can be found in [32, 51, 49, 52].

6.2.2 European and American options

There exist two main families of options : European and American ones, distinghished by the time at which the owner can exercise his option. “American exercise” means that the option may be exercised at any time between the day of the purchase and the strike date. “European exercise” means that the option may only be exercised on its expiration day. Many of the cash-based index options (the most popular type) have the European style (cash-based options are options for which the settlement is made in cash rather than by providing all the equities of the index, which is often impossible due to the number of equities in the index). Some index options have the American exercise feature. For example, options on the S&P500 and on the NYSE index are cash-based European options. However, the option on the S&P100 is a cash-based American option.

It can be shown that it is not optimal to exercise an American call before maturity. Therefore American and European calls can be handled uniformly. Only the American put has to be treated differently.

In this work, we will consider models where the options can be exercised at one predefined date; i.e. we will focus on European options.

6.2.3 Strike price

It is important to know the strike price scheme to be able to model options that cannot be observed today (because they don’t yet exist), but will appear sometime in the future (in the second and subsequent periods of the tree of scenarios). Based on this scheme, an automatic procedure to construct options over each period can be developped. We now present some of the features of the scheme:

- When creating options, the usual rules followed by markets is to space strike prices depending on the underlying stock price S as follows:
  - $S \leq \$25 : \$2\frac{1}{2}$ spacing;
  - $S \in\ ]\$25, \$200] : \$5$ spacing;
  - $S > \$200 : \$10$ spacing;

  For indices, the strike prices vary depending on the underlying instrument. Typically, index option contracts will have strike price intervals of 2 1/2 points or 5 points (e.g. S&P100 and S&P500) when time to maturity is close by and 10 points (i.e. S&P100) or 25 points (i.e. S&P500) for later expiration months.

- When an option with a new expiration date is introduced, the two strike prices closest to the current stock price are usually selected for this option. A third one may also be selected if one of the first two strike prices is very close to the existing stock price.

  For indices, in-the-money, at-the-money and several out-of-the-money strike prices will be proposed;

- If the stock price rises above the highest current strike price, a new, higher strike price is usually introduced. The same is true if the stock price falls below the lowest current strike price.

6.2.4 Expiration date

To construct the multi-period tree of scenarios we have to define the beginning and the end of each period. An optimal choice takes into account expiration dates of the options. Again, it is useful to know the rules used in practice:

- For a given expiration month, there is only one expiration date: the Saturday immediately following the third Friday of the month;

- Usually, options last less than one year. For a given stock, options follow a specific cycle of expiration months:

  - January cycle: January, April, July, October;
  - February cycle: February, May, August, November;
  - March cycle: March, June, September, December.

  On a given day, the stock options traded are those with the next two coming expiration dates (on Saturday of each month of the year) and the next two months coming afterwards in the cycle. This implies that at a given day, there exist only four expiration dates.

  For example, IBM is on a January cycle. At the beginning of February, options are traded with expiration dates in February, March, April and July.

  For indices, it is more complex, and depends on the underlying instrument. For example, the S&P100 is characterized by four consecutive expiration months. The S&P500 trades with the three near-term expiration months plus three months of the March cycle (or equivalently expires each of the months of the March cycle plus the two near-term months not in the cycle). At the beginning of February, S&P500 options are traded with expiration dates in February, March, April, June, September and December.

- When an option expires on a Saturday, a new one is created the following Monday.
- There exist longer-dated stock options: LEAPS (long-term equity anticipation securities). For stocks, these can have one expiration date in January up to two years into the future. For indices, they can have one expiration date in December up to three years into the future.

In our models, problems appear when expiration dates fall before the end of a period. These problems could be solved by using discounted cash-flows. However, we prefer to avoid them altogether by imposing that the creation and expiration of options should be possible only before the initial period of the model, at the end of each period, or after the last period. There is only one key date per month for each option and, except for the special case of LEAPS, about 10 key dates during the life of an option (investment horizon less than one year for an option).

6.2.5 Settlement price and date

- As explained above, for an option on an index, the settlement is in cash rather than by providing the underlying portfolio of stocks. The settlement procedure is different for European and American style options.
- For American index options:
  - Last trading date: third Friday of the expiration month;
  - Settlement price computed each day with the last (closure) trading prices reported for each of the underlying stocks. Last possible "settlement" date: third Friday of the expiration month;
  - Expiration date: the Saturday (not a business day) following the third Friday.
- For European index options:
  - Last trading date: the Thursday (if it's a business day) preceding the third Friday of the expiration month;
  - Settlement price computed on the basis of the opening prices of each of the underlying stocks (even if the underlying markets do not open at the same time) the third Friday of the expiration month;

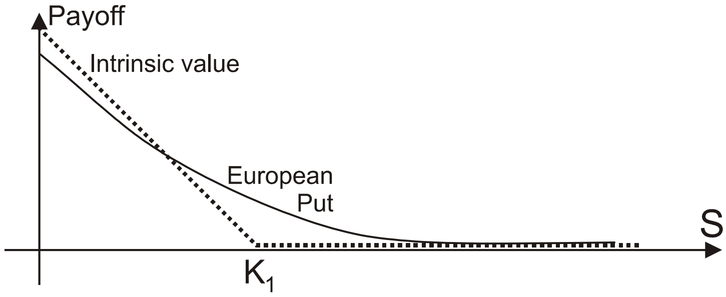


Figure 6.1: Intrinsic value and option value

– Expiration date: the Saturday (not a business day) following the third Friday.

It is important to note that the amount received by the investor at the expiration of the option is not the difference between the index value and the strike price, but the difference between the settlement price and the strike price. Of course, we can expect difference to be very small. We only model the index price in a tree of scenarios so as to avoid complexity of computation of the settlement price.

6.2.6 Intrinsic, time and volatility values

The intrinsic value of an option is its payoff at maturity. Options out-of-the-money have value zero. The payoff of options in-the-money is the difference between the price of the underlying asset and the strike price. Thus, the intrinsic value is a piecewise linear function. To obtain the value of an option at some instant before maturity, we have to add two other components: time value and volatility value. The time value is a positive (negative) constant for calls (puts) in-the-money and zero otherwise. The volatility value is a nonlinear function of the underlying value. This value can be negative or positive.

As a result, the option value is either a piecewise affine function of the stock price or a more general non linear function. This is illustrated in Figure 6.1. The affine function has nicer properties and could lead to simplifications when solving real problems. In particular, these properties will be used to handle the guarantee constraint in the next chapter.

6.2.7 Ask and bid prices

Each option is defined by two prices. The ask price is the price the investor has to pay to purchase the option. When he sells the option, he receives the bid price. The bid price is always smaller than the ask price. The difference between the two prices is the bid-ask spread.

Option price	**Maximal spread**
$\leq$ \$0.5	\$0.25
]\$0.50,\$10]	\$0.5
]\$10,\$20]	\$0.75
$>$ \$20	\$1.00

Table 6.1: Bid-ask spread limits

Usual option pricing methods define only one price for each option, even if both prices are required to match reality. We will show in Section 6.3.5 how to construct such prices to be as close as possible to real values observed on markets.

6.2.8 Size of a contract

One option contract gives the holder the right to buy or sell 100 shares at the specified strike price. This is convenient since the shares themselves are normally traded in lots of 100. Note that the quoted option price is the price of an option for one share; i.e. one contract costs 100 times the quoted price.

For LEAPS on indices, one contract is an option on 100 times one-tenth of the index; i.e. the index is divided by 10 for the purposes of quoting the strike price and the option price (otherwise a contract would be too expensive). Our model makes no difference between short-dated and long term contracts.

The fact that we work with indivisible lots of options must be taken into account when defining a model for real problems.

6.2.9 Market limits

Each exchange specifies a position limit for each stock. This defines the maximum number of option contracts that an investor can hold on one side of the market (e.g. 8000 contracts). Long calls (a positive quantity) and short puts (a negative quantity) are considered to be on the same side. Short calls and long puts are on the other side.

The exercise limit defines the maximum number of contracts that can be exercised by any individual in any period of five consecutive business days.

The exchange also sets upper limits for the bid-ask spread with respect to the option price. Typical limits are given in Table 6.1.

These real constraints are easy to integrate in models based on a scenario tree.

6.2.10 Commissions and taxes

Commissions

For the retail investor, commissions vary significantly from broker to broker. Discount brokers generally charge lower commissions. The actual amount charged is usually calculated as a fixed cost plus a proportion of the dollar amount on the trade. For example:

- Trade $\leq \$2500 : \$20 + 0.02$ of the dollar amount;
- Trade$\in]\$2500, \$10000] : \$45 + 0.01$ of the dollar amount;
- Trade $> \$10000 : \$120 + 0.0025$ of the dollar amount.

The typical maximum commission is \$30 per contract for the first five contracts plus \$20 per contract for each additional contract. The minimum commission is \$30 for the first contract plus \$2 per contract for each additional contract.

The commission must be paid each time the investor sells or buys a contract. When the option is exercised, the investor pays the same commission that he pay would when placing an order to buy or sell the underlying stock. In general the commission system tends to push investors in the direction of selling options rather than exercising them.

Taxes

The taxes are complex and impossible to describe in full generality. All these characteristics are not useful to price options and construct the tree of scenarios, but are important to model before trying to solve a realistic problem.

6.3 Option pricing models

6.3.1 Introduction

To define an option price at the root of the tree of scenarios, we can observe its price on the market. At each possible date and node of the tree, we can use either the Black-Scholes formula or a binomial tree approach. These classical methods are briefly described in Chapter 4. Unfortunately, when used in our multinomial representation of the future, these prices usually lead to numerical, artificial arbitrage opportunities, which in turn lead to unbounded solutions for the optimization model. This difficulty disappears if we set bounds on the variables involved in the optimization model, but then the optimal solution becomes an artificial one, where some or all quantities reach their bounds. A more satisfactory solution consisits in finding prices that satisfy the arbitrage equations. This will be discussed in subsections 6.3.2-6.3.7.

The market represented by a multinomial tree (involving more than two scenarios) is not complete, as there are more unknowns than equations in the no-arbitrage conditions and so, in general, an infinity of solutions satisfying the no-arbitrage equations. We will develop here several models to select the solution corresponding best to reality.

In this section, we will only consider one-period trees of scenarios. The generalization is left for Section 6.4.

6.3.2 Arbitrage equations

Definition

The arbitrage equations were presented in section (4.4). We reformulate them here for the case of a one-period tree of scenarios. To exclude arbitrage opportunities in a one-period tree of scenarios, there must exist a positive vector ψ such that:

$$\begin{pmatrix} 1 \\ S_t \\ \tilde{popt}_1 \\ \vdots \\ \tilde{popt}_{nbOpt} \end{pmatrix} = \begin{pmatrix} e^{r(T-t)} & \dots & e^{r(T-t)} \\ S_{T1}e^{q(T-t)} & \dots & S_{TnbS}e^{q(T-t)} \\ popt_{1,T1} & \dots & popt_{1,TnbS} \\ \vdots & \vdots & \vdots \\ popt_{nbOpt,T1} & \dots & popt_{nbOpt,TnbS} \end{pmatrix} \begin{pmatrix} \tilde{\psi}_1 \\ \tilde{\psi}_2 \\ \tilde{\psi}_3 \\ \vdots \\ \tilde{\psi}_{nbS} \end{pmatrix} \tag{6.1}$$

where t is the beginning of the period, T is the end of the period, r is the risk-free rate, q is the dividend yield, S_t is the initial price of the underlying asset, S_{Tj} is the final price of the asset in scenario j, $\tilde{popt}_k$ is the initial price of option k, $popt_{k,Tj}$ is the final value of option k in scenario j, $nbOpt$ is the number of options considered, nbS is the number of scenarios in the tree.

The notation $\tilde{x}$ means that x is an unknown. We suppose that all the option prices are known at time T either because they are at maturity or otherwise by using a suitable option pricing method (as will be shown further on). This implies that (6.1) is a set of linear constraints where the unknowns are only the initial option prices and the quantities ψ.

The vector ψ is the state-price vector. Its elements are the risk-neutral probabilities discounted by the risk-free rate.

Note that, if $nbS > 2$, then there are more unknowns than equations in (6.1). So that, in general, the system is under-determined. We are now going to see how we can compute "reasonable" option prices which satisfy (6.1).

6.3.3 The model

If we consider that Black-Scholes gives a good approximation of what the price should be, we can try to minimize the deviation between the Black-Scholes prices and the ones satisfying equations (6.1). This is achieved by the following model:

$$\begin{array}{ll} \min & \sum_k (popt_k - BS_k)^2 \\ \text{subject to} & (6.1) \\ & \psi_j \geq 0 \\ & popt_k \geq 0 \end{array} \tag{6.2}$$

Of course, instead of the Black-Scholes estimates, we could use the observed option price when available. More generally, we will assume we know an ideal target price for each option. In Section 6.5, we shall discuss how to obtain such target prices. Then the option pricing model becomes:

$$\begin{array}{ll} \min & \sum_k (popt_k - target_k)^2 \\ \text{subject to} & (6.1) \\ & \psi_j \geq 0 \\ & popt_k \geq 0 \end{array} \tag{6.3}$$

6.3.4 First improvement: absolute relative objective function

The objective function in (6.3) is the square of the difference between the target price and the unknown price. We used a squared difference function because it is classic and also easily optimized by commercial software. However, the errors are amplified by this objective function. When using market prices as target, the largest differences are observed for options far in-the-money (liquidity problem), i.e. for options that are very costly and "mispriced" (overpriced) on the market. By using a squared function, we give more weight to these options. This implies that we try to adjust all the option prices to match as well as possible the amplified abnormal extreme prices. It is precisely to avoid similar difficulties that the implied volatility is usually not computed over all the call options, but only with the ones nearly at-the-money, or by using some more advanced weighting schemes.

For these reasons, a linear objective function seems more suitable (and this also allows some pre and post-processing as presented in Section 6.6). As the differences can be positive or negative, we replace the square deviations by absolute deviations. An sum of absolute deviations cannot be handled directly by linear programming methods, but after adding *nbOpt* variables which measure the absolute value of each difference, and 2*nbOpt* constraints which model these absolute values, we obtain a linear model solvable by the simplex method. The resulting model is larger than the previous one but still easily and quickly solvable.

Another improvement is to adjust the weight of each option by using a relative difference instead of an absolute one. Indeed, the target option prices vary from zero to large values. The absolute differences tend to be larger when the target price is large, and so the highly priced options are heavily taken into account by the model. This inequity can be removed by using relative differences; i.e. by expressing each error as a percentage of the target price.

Altogether, we obtain the following linear model:

$$\begin{array}{lll} \min & \sum_k \frac{AbsDifOpt_k}{target_k} & = \sum_k \frac{|popt_k - target_k|}{target_k} \\ \text{subject to} & (6.1) & \\ & AbsDifOpt_k \geq popt_k - target_k & \\ & AbsDifOpt_k \geq target_k - popt_k & \\ & \psi_j \geq 0 & \\ & popt_k \geq 0 & \end{array} \tag{6.4}$$

Another way to weigh the options was suggested to us by Oldenkamp. The idea is to introduce the option *vega* (the rate of change of the value with respect to the volatility of the underlying asset: $\partial price/\partial\sigma$) to weigh the differences. The goal is to reduce the impact of some difficult to handle options (deep in-the money), which are typically overpriced on markets due to liquidity problems.

6.3.5 Second improvement: bid and ask prices

Construction of the prices

Classical option pricing methods give only one price for each option. Also, databases like DataStream store only one daily price per option. However, on the real market, there is a difference between the purchasing price (ask price) and selling price (bid price) of each option. The difference between the two prices is called the bid-ask spread. The typical market approach is to add (remove) a small sum to the option value to obtain the ask (bid) price.

In the previous model, we only used one target price for each option and not the corresponding bid and ask prices. It is easy to adapt the optimal prices obtained from the optimization process to define the two prices without losing the no-arbitrage property. Arbitrage opportunities can only come from two reasons, and to each corresponds only one investor reaction: either the option price is too low and the investor should buy the option, or the option price is too high and the investor should sell the option. From this observation, it easy to show that if $\{popt_1, popt_2, \ldots, popt_{nbOpt}\}$ is an arbitrage-free system of prices, and the ask price (resp. bid price) for option k is higher (resp. lower) than $popt_k$, then the system of ask and bid prices offer no arbitrage opportunities. So, for instance, adding (resp. removing) a typical market half-spread to the optimal prices derived from (6.4) yields valid ask-bid prices.

We use this approach as it is not possible to adapt the free-arbitrage equations to take the bid and ask prices into account. If we look at the arbitrage equations (6.1) we see that there is no variable to measure the quantities purchased or sold. That is, the arbitrage equations don't take into account the operation performed by the investor, and there is no way to split the option values between the purchase case and the sale case.

Improvement

It is important to understand that the target price is only used in the option pricing process and will not be used afterwards to model real cases. Instead, the only prices of interest will be the computed bid and ask prices. This suggests that, setting a precise target price for each option in the model is not required, but that we can, as in Rubinstein's implied tree model [63], relax the target price between the observed ask and bid prices. In this improved model, the target price becomes an unknown variable constrained between an upper bound (the target bid price) and a lower bound (the target ask price).

Rubinstein's constraints are strict. He requires that the optimal option prices fall between the two corresponding observed bid and ask prices. If such prices do not exist, then the problem has no solution, meaning that the observed prices allow arbitrage opportunities. We expect arbitrage opportunities to exist on real markets, even if they are limited!

By contrast, since we cannot tolerate arbitrage opportunities in our optimization models, our philosophy is to find the closest optimal prices without these opportunities. The idea is to define the objective function as a sum of penalty terms. If it possible to find an optimal option price which falls between the bid and ask prices, then there is no penalty. Otherwise, a penalty term corresponding to the distance between the optimal price and either the bid or ask price is added to the objective function. Note that at most one penalty term is incurred per option. As the bid price is always smaller than the ask price, and the constraints impose that the optimal price should be smaller than the ask price and larger than the bid price, there is only one possible violation for each option: either the optimal price is smaller than the bid price (and there is no problem with the ask price) or the optimal price is larger than the ask price (and there is no problem with the bid price).

After optimization, we adjust the prices depending on the value. When the optimal prices fall in the target bid-ask range, we set the (arbitrage-free) bid and ask prices equal to the target bid and ask prices, respectively. For other options, we increase the total spread. When the optimal price is less than the target bid price, then we set the (arbitrage-free) bid price equal to the optimal price, and the (arbitrage-free) ask price to the target ask price. Therefore the resulting spread is larger than the target spread. A similar procedure is applied for the opposite case. In other words, for most options, the bid and ask prices correspond to the observed bid and ask prices. For the others, at least one of the bid or the ask is set to the target value, and the other is set as close as possible without arbitrage opportunities. This approach leads to smaller deviations between the optimal and target prices than in the previous models.

A second advantage of this approach is to allow compensations between options. Indeed, in this approach and for each option p, there are an infinity of target prices between the target bid and ask prices by comparison to the unique target price in the previous models. For each value which satisfies the no-arbitrage equations in this range, there is no penalty term and each value is an optimal solution for the option pricing problem (at a local option optimization level). Therefore, the optimization process can select indifferently one or another value. As however this value influates on the other option values by the no-arbitrage system of equations, this set of optimal values for option p gives more freedom to define the other option prices. Therefore, the optimal price of option p can be selected in the range of the target bid and ask prices by the optimization process such as to reduce the deviations for other options.

In this improved model we maintain the relative measure of the errors. This new formulation requires $2nbOpt$ more constraints than in model (6.3) to define the absolute values (as for model (6.4)) and $2nbOpt$ more variables (more than for model (6.4)). This gives:

$$
\begin{array}{ll}
\textbf{Model(OP1)}: & \\
\text{min} & \min \sum_k \frac{err_{bid,k}+err_{ask,k}}{target_k} \\
\text{subject to} & (6.1) \\
 & err_{bid,k} \geq bid_k - popt_k \\
 & err_{bid,k} \geq 0 \\
 & err_{ask,k} \geq popt_k - ask_k \\
 & err_{ask,k} \geq 0 \\
 & \psi_j \geq 0 \\
 & popt_k \geq 0
\end{array}
\tag{6.5}
$$

where bid_k and ask_k are respectively the target bid price and the target ask price of the k^{th} option, $err_{bid,k}$ and $err_{ask,k}$ are respectively the errors measured with respect to those bid and ask prices. The first two constraints after the arbitrage equations state that the bid error is null if the price is larger than the bid price, and equal to the positive difference otherwise. The next two constraints define the error with respect to the ask price. By definition of the ask and the bid prices, at least one of the two errors is always null for each option.

6.3.6 Third improvement: parity equations

By the arbitrage equations it is possible to express the price of a call as a function of price of the corresponding put.

$$c - p = Se^{-q(T-t)} - Xe^{-r(T-t)} \tag{6.6}$$

Since these two sets of prices are not independent, it is redundant to use both of them in the model. This is true only if we know with certainty all the parameters. In particular, we know from the previous chapter that it is difficult to compute the dividend yield q and the risk-free rate r. This appears clearly in the numerical results for the smile effect presented in the last section of this chapter. Note also that if we know with certainty all the parameters, then the put equations are redundant with the call ones but have no impact on the quality of final solution. For these reasons, we did not modify the software to take the parity equations into account.

6.3.7 Fourth improvement: state-prices

The model

Finally, we propose a last modification which is not really aimed at improving the quality of the option pricing process, but rather at improving the quality of the corresponding state-prices.

Indeed, a side result of the computation of the arbitrage-free option prices using (6.1) is the computation of positive state-prices for each scenario. We have defined an optimization model because we know that if a solution exists, then typically an infinite number of solutions exist, since our representation of the market is usually not complete. We have chosen to minimize the deviation of the option prices with respect to target prices, but we have specified nothing about the other unknowns: i.e. the state-prices.

However, when we look at numerical results obtained with this model (6.5), we observe that the risk-neutral probabilities are chaotic! They absolutely do not coincide with the risk-neutral pdf defined in the previous chapter. Since the value of the state-prices will be used in our models, it is natural to require that the values obtained through different methods be coherent with each others.

The idea that we have implemented to achieve this purpose, is to add a penalty term to the objective function, so as to minimize the difference between the risk-neutral probabilities associated with our tree of scenarios and the "true" target probabilities obtained as described in the previous chapter. This new term corresponds to the objective function defined by Rubinstein in the construction of an implied binomial tree (see model (5.22) in Section 5.3.6). However, the main objective must remain the computation of the most suitable option prices, so that more weight should be attached to the corresponding terms. In fact, as state-prices are adjusted probabilities, the sum of their values is always smaller than unity and the sum of the errors tends to be even smaller. If the error on the bid and ask prices are measured in relative terms, their order of magnitude is similar but the range of values tends to be larger. Hence, we have not increased the weight of the target penalty term, as when a significative price on the prices appears, it usually dominates in the objective function.

We maintain a linear model by adding a linear absolute term to reflect the penalty on the state-price errors. This adds nbS variables and $2nbS$ constraints and leads to the following model:

$$
\begin{array}{ll}
\textbf{Model(OP2)}: & \\
\text{min} & \sum_k \frac{err_{bid,k}+err_{ask,k}}{target_k} + \sum_j AbsDifProba_j \\
\text{subject to} & (6.1) \\
& err_{bid,k} \geq bid_k - popt_k \\
& err_{bid,k} \geq 0 \\
& err_{ask,k} \geq popt_k - ask_k \\
& err_{ask,k} \geq 0 \\
& AbsDifProba_j \geq \psi_j - \psi'_j \\
& AbsDifProba_j \geq \psi'_j - \psi_j \\
& \psi_j \geq 0 \\
& popt_k \geq 0
\end{array}
\tag{6.7}
$$

6.4 Option pricing optimization

6.4.1 Introduction

In this section, we are going to extend the above option pricing models to a multiperiod framework. We describe appropriate optimization procedures (Sections 6.4.3-6.4.4).

6.4.2 Optimization over one period

The arbitrage equations (6.1) used in the previous sections are adequate for one-period models. They take into account the values of the securities at the beginning and at the end of a period, and this corresponds perfectly with a one-period multinomial tree of scenarios. By using the values defined in the tree, we can easily instantiate the equations.

Moreover all the constraints are linear, and the objective function is either quadratic or linear. Classical optimization methods, such as the barrier method or the simplex algorithm, can easily and efficiently compute optimal solutions. In our implementations, we have used the professional CPlex optimization library. In this way, we benefit from a large set of optimized and reliable procedures.

6.4.3 Optimization over two periods

Tree of trees

Difficulties appear when we use more than one period to model the future. We consider here especially the case of a two-period tree of scenarios. The results will remain valid for more periods.

As stated in the previous subsection, the arbitrage equations (6.1) don't take into account intermediate dates. Only the beginning and the end of a same period are defined. In order to handle two-period models, we can consider the multi-period tree of scenarios as a tree of one-period trees. Indeed, in our construction, the subtrees are not recombining and the tree is constructed from the root to the leaves by adding a one-period subtree to each leaf of the previous one-period subtree. This may suggest that the pricing process can be performed locally for each subtree.

However, in a multi-period tree, the price of an option at each node of the tree depends on the prices at previous nodes and has an impact on the prices at the nodes of the next subtrees. Therefore, pricing cannot be done independently at each node or within each subtree. Let us see this in more detail.

Backward procedure

In the case of an option maturing at the end of the second period, its value at the end of the first period depends on its value at maturity, i.e. at the leaves of the second period subtrees. This means that in the arbitrage equations (6.1) for the first period, the final option values $popt_{i,Tj}$ are also unknowns. The problem is non linear and cannot be solved due to lack of information.

However, if we first consider the trees of the second period, then all the information is available and the option pricing problem can be solved. We then obtain the option prices at the beginning of the second period, for each subtree. As the root of each second period tree corresponds to a leaf of the first period tree, these options prices are also the final option prices at the end of the first period, so that the option pricing problem for the first subtree can now be solved. In conclusion, a multi-period tree of scenarios can be handled by a backward procedure.

Note that the set of options can be divided into three parts: options covering only the first period, options covering at least the two periods and also options that do no exist initially and that appear only at the beginning of the second period. The backward procedure is only required if options are defined over several periods. If we consider options covering only the first period or/and options covering only the second period, then the problems can be solved separately.

Moreover the set of options covering only the second period is particularly interesting because it allows to model the future according to realistic market rules (some options will exist only if some future states occur and so only in some subtrees). This also implies that, in a multi-period tree, we can model options that do not initially exist on the market but that will appear later!

6.4.4 Simulated Annealing algorithm

The backward procedure (first optimize for each of the subtrees of the second period and next, use the optimal prices found to optimize for the first period) is a local optimization method. More precisely, as option prices are computed from the end to the beginning of the tree, initial ones are strongly constrained by final ones. Minimization of errors is still performed locally at each subtree and there is no global minimization. This implies that a sub-optimal solution for a second period subtree could lead to an improved first period solution, which in turn could improve the global solution. In other words, perturbing the option prices at the end of the second period, while still satisfying the arbitrage equations for this period, could lead to an improvement of the global solution. A simulated annealing heuristic has been developed for this purpose.

The idea is to first compute an initial solution by the backward procedure described in the previous section and then try to improve it by a simulated annealing process (SA). The outcome of this procedure are option prices at the root and at the leaves of the first-period subtree.

At each step, SA tries to improve the current solution by changing one option price for one scenario at the end of the first period. The dimension of the space is *nbS* x *nbOpt*. A list of all these possibilities is constructed and randomized at each stage of the SA process, so that all the *nbS* x *nbOpt* possibilities are considered during a stage.

For the option selected, the price is modified by a tiny value (10 basis points) in the best direction; i.e. it is reduced (increased) if the current price is larger (smaller) than the target price. This price is then corrected if necessary to the nearest value in the same direction leading to no arbitrage. This is done by constructing and solving a new linear minimization problem in which the objective function measures the difference between the unknown arbitrage-free price and the SA price, and the constraints are the no-arbitrage equations (6.1) for the corresponding second-period subtree and a bound equal to the SA price on the option price to restrain the direction. If such a value exists (i.e. if the problem is feasible), a new initial option price is computed by locally reoptimizing the first subtree with the new value of the option.

When a new solution is obtained, its quality is analyzed. There is an improvement if the current option price has moved closer to the target option price and if the option prices at the intermediate date remain close to the target prices (if the second-period penalty increases, then the increase must be less than a fraction of the improvement of the initial price).

The process stops if the best solution is achieved (penalty for the first period less than a user-given precision), if no more improvements take place or if a user-given delay is elapsed.

6.4.5 Arbitrage and numerical instabilities

To avoid the arbitrage problem, we should find option prices and state prices which exactly satisfy the arbitrage equations. These equations divide the solution space in two parts: with and without arbitrage opportunities.

This could lead to a numerical problem. Indeed, the option prices computed by the above-mentioned procedure, and declared to be "arbitrage-free", but however close to the arbitrage frontier, could possibly lead to arbitrage opportunities when used in subsequent models and procedures only due to numerical approximations (as the precision of a computer is by definition limited).

In fact, this problem generally does not appear because we never use directly the optimal option price, but rather the bid and ask prices obtained by applying a spread around this optimal value. As the spread is typically far larger than computer precision, this removes the numerical rounding error and the arbitrage opportunities. The same conclusion can also be obtained when modelling a transaction cost for each option.

If we want however to model a problem without spread or option transaction cost, then we could simply fix an artificial spread equal to a tiny value larger than computer precision. This implies nothing for the investor, who in practice sees no difference between the original value and the original value plus epsilon, but this removes the numerical differences.

6.5 Target option prices and probabilities

6.5.1 First and second periods

Using more than one period adds some new difficulties. In order to price options using model (6.7), we need to know the bid and ask prices for each option, and the risk-neutral target probabilities of the underlying asset. Initially, prices and distribution of probability can be observed or derived from market data, but this is no longer the case for periods in the future.

Defining target prices for the first period is easily done if the root of the tree corresponds to the present time. The most natural targets are then the prices observed on the market. It is also relatively easy to define target risk-neutral probabilities. If we compute the implied risk-neutral pdf to construct the tree of scenarios (or use a consensus pdf converted into a risk-neutral one), then we have more information than required. The stratification of the risk-neutral pdf directly gives the index value and the state-price corresponding to each leaf.

For the second and subsequent periods, things are more complex, as we cannot observe future prices on the market. In a theoretical context, under the usual normality assumption, the Black and Scholes (BS) formula is a good choice to define target prices because most of its assumptions are satisfied. Even if the true return distribution is not Normal, this already gives a good estimate. Improved approaches are presented in Sections 6.5.2-6.5.3. For the target probabilities, there is a solution similar to the one used for the first period. To construct subtrees of the second period, we need to define the distribution of the underlying asset, and so we can use the corresponding risk-neutral pdf to define target state-prices.

6.5.2 Improved Black and Scholes formula for the target option prices

Even if we know that the assumptions behind the BS formula are usually violated on real markets and in our model, the BS formula has also proved, in practice, to generally give good approximations of market value. Its main drawback comes from constant volatility assumption and the inability to match the observed volatility smile.

As we have seen for construction of the index distribution, most improved approaches do not totally reject the BS formula, but rather modify it to better correspond to reality. In particular, Shimko [64] has proposed a quadratic approximation of the volatility as a function of the index price (equation (5.15)) and used this function in the BS formula instead of the constant volatility σ (equations (5.17)). In single-period models, this improved BS pricing process yields prices very close to observed prices (nearly exactly equal in our numerical experiments).

For the second period, we do not have enough information to estimate the volatility of the tree model. However, we typically expect a similar structure for each subtree of the second and first periods (we use the same rates, returns...) and so we can expect a similar volatility smile in each subtree. Therefore, instead of using the classical BS formula to construct a target price at the beginning of the second period, we should use the improved BS formula defined with respect to the initial observed smile. In this way, we also introduce a smile property in the prices of the second period. The smile used is probably not the one we will observe at this time, but is a good approximation (far better than the constant initial volatility).

In conclusion, we use as target prices at the beginning of the first period, the observed option prices, and at the beginning of the second period, a first possibility is to use the improved BS prices, as adapted by Shimko, but using the volatility smile estimated at time $t0$. A second approach is proposed in Section 6.5.3.

6.5.3 Target option prices from state-prices

Principle

Another approach to define target option prices at the root of a subtree is to use the state-prices corresponding to this subtree. By definition, to avoid arbitrage opportunities, the option price at the beginning of a period is given by the sum, over all possible scenarios for this subtree, of the option price multiplied by the corresponding state-price:

$$\tilde{popt}_i = \sum_{j=1}^{nbS} \psi_j popt_{i,j} \tag{6.8}$$

If we know the state-prices for all periods, then it is easy to compute target prices. Moreover, we already have a prior guess of these state-prices by sampling the risk-neutral version of the pdf used to model the period. We will propose in the rest of this section two methods to adjust the prior guess in order to compute target prices. If these methods cannot be applied, e.g. if a different pdf is used to model the future in each subtree, then we can only use the prior guess.

It is important to notice that this principle is a similar approach to the one used until now to price the options. Indeed, the option pricing model uses this system of equations to define prices since (6.8) is a subsystem of (6.1) and therefore a subsystem of the constraints of the pricing model (6.7). Therefore if the state-prices we compute by using one of the methods presented in this section, match the state-prices which are solution of the pricing process (6.7) performed afterwards, then the option prices obtained by (6.8) and the pricing model (6.7) are the same. Moreover, as the state-prices obtained from the optimization pricing process are the optimal state-prices that match the assumptions and constraints of the model with respect to the prior guess (sampled from the risk-neutral pdf modelling the period), the methods presented in this section should try to estimate a priori these state-prices.

This leads to two conclusions. First, the target option prices and target risk-neutral probabilities are highly correlated. Second, the prediction of second period (target) option prices could be improved if we were able to tune the discrete set of state-prices. Two algorithms are now presented for this purpose.

State-prices replication

Typically, the same probability distribution is used at each period to construct the tree of scenarios. We expect that in the short term the same distribution remains valid. This implies that we expect to be in the same (consensus and risk-neutral) world for each subtree. This also implies that we can learn from the results of the first initial and observable period and draw useful conclusions for other periods.

Especially, if we compute the optimal option prices for the first period by considering only options covering this period (so we don't require the full backard procedure), we obtain adjusted state-prices that could be used for the other periods. These new state-prices satisfy the assumptions behind the model and take into account the properties of the options initially observed on the market, such as the implied smile effect. These properties are then replicated by the state-prices for the second period.

Clearly, this set of state-prices is compatible with the second-period subtrees. Indeed, let $\{ret_i\}$ be the set of returns sampled from the consensus pdf used to model all the periods, then the set of index prices $\{S_{Ti}\}$ at the end of one period is given by:

$$S_{Ti} = S_t e^{ret_i(T-t)} \tag{6.9}$$

where as before t is the beginning of the period, T is the end of the period and S_t is the initial index price. From (6.9), we can reformulate the arbitrage equations (eqarbitmod2) as follows:

$$\begin{pmatrix} 1 \\ 1 \\ \tilde{popt}_1 \\ \vdots \\ \tilde{popt}_{nbOpt} \end{pmatrix} = \begin{pmatrix} e^{r(T-t)} & \dots & e^{r(T-t)} \\ e^{(ret_1-q)(T-t)} & \dots & e^{(ret_{TnbS}-q)(T-t)} \\ popt_{1,T1} & \dots & popt_{1,TnbS} \\ \vdots & \vdots & \vdots \\ popt_{nbOpt,T1} & \dots & popt_{nbOpt,TnbS} \end{pmatrix} \begin{pmatrix} \tilde{\psi}_1 \\ \tilde{\psi}_2 \\ \tilde{\psi}_3 \\ \vdots \\ \tilde{\psi}_{nbS} \end{pmatrix} \tag{6.10}$$

Note that the first two lines of the payoff matrix, representing respectively the risk-free asset and the index, are the same for all the subtrees. Therefore, the state-prices solving the first-period problem, satisfy, for all the periods and all the subtrees, the subsystem composed by these first two lines. Moreover, the other lines of the system (6.10) define the prices of the options considered in each specific subtree. All the remaining components of the initial price vector are the unknown option prices $\tilde{popt}_i$ and they are directly defined by the state-prices using the relation (6.8) of the system (6.10). So, the set of state-prices of the first-period subtree is a solution satisfying the arbitrage equations (6.10) for all the subtrees.

The complete replication procedure requires two steps. First, we compute the option prices and state-prices for the first period by considering only the options covering the first period. Second, we start the backward procedure. We first run the optimization process for all the subtrees of the second period and for all options (covering only the first period, both periods and only the second period) using the state-prices obtained at the first step to define the target option prices. Finally, if there exists options covering the two periods, we restart the optimization for the first period, but this time with all the options covering the first period; oterwhise, if there exists no options covering the two periods, the optimal solution found at the first step remains valid.

State-prices from options with different maturities

In the previous procedure, we have not used all available information. To construct state-prices for the first and second period, we have only used options with a maturity at the end of the first period. We do not use options initially available with a maturity at the end of the second period.

The final state-price for a given leaf of the two-period multinomial tree is given by the product of the state-price for the first period with the state-price for the second period, along the unique path from the root to the leaf. Thanks to the options maturing at the end of the first period and the option pricing model, we are able to obtain state-prices for the first period. We can now perform the same kind of optimization over the two periods by hiding the intermediate layer and using only options available today with maturity at the end of the second period. This will give a final state-price for each leaf. For each node at the end of the second period, dividing the latter by the corresponding state-price of the first period yields the target state-price to use for the second period.

The quality of this solution partially depends on the target state-prices for the two-period model, but also on the representativity (quantity and quality) of the options initially observed.

This last approach seems better than all the previous ones because it takes into account both the index distribution and the observed option prices over two periods (with the initial smile effect). However, one difficult problem remains to be solved: how can we compute the state-prices over the two periods for each leaf of the two-period tree? We cannot simply hide the intermediate layer and construct an equivalent one-period tree. Indeed, remember that the tree is not recombining, and so that several final leaves can be defined with a same index price. As when we construct a one-period tree, we can only attach one index price to each leaf, it means that we have to associate several leaves, characterized by a unique index price but different state-prices, of the two-period tree to one leaf, characterized by one index price and one state-price, of the one-period tree. If we solve the pricing problem corresponding to the one-period tree, we have no way to know afterwards how to split the values of the state-prices between the leaves of the two-period tree.

Therefore, in order to obtain the two-period state-prices, we only know the two-period tree and, without using other system of equations than the arbitrage equations (6.1), the two-period state-prices can only be obtained by the product of the first-period state-prices by the second-period state-prices that we are looking for. This approach is temporarily let aside.

6.6 Pre and post-processing

6.6.1 Statistics

We have defined models to compute arbitrage-free option prices. A priori, we cannot predict if these prices are good estimations of the option prices observed if the corresponding scenario materializes. Before the pricing process, we can even not predict if the arbitrage-free option prices are close to the target option prices. We only have the guarantee that the optimal solution minimizes the bias, first between target prices and free-arbitrage ones, and secondly between risk-neutral probabilities. Here we suggest performing post-processing to study the quality of the solution with respect to the target values.

The first set of statistics concerns the bias between target and optimal option prices. We consider target prices as the best piece of information available to represent the market. For each option, we measure the absolute difference between the prices. This is consistent with the linear absolute objective function used in the option pricing model. Moreover, as we will use bid and ask prices in portofolio optimization problems instead of the target and optimal prices defined in the option pricing model, and as we have decided to adapt the spread to match as much as possible the bid and ask target prices, we compute the bias between the target and optimal bid prices and the same for the ask prices. Numerical results are provided in the next section.

The second piece of available information consists of the risk-neutral pdf. We have constructed a pdf to model the future market as we imagine it. We should obtain a close representation from the optimal state-prices. We have decided to graphically compare the target and optimal pdf's. The risk-neutral optimal pdf is only a rough approximation. There is no way to convert a discrete set of probabilities into a continuous pdf.

6.6.2 Option cleaning

Unwanted options

We use option prices to model the future and to compute optimal portfolios, but working with options could be tricky. As option prices greatly influence the results and that some options can be mispriced on the market, we need to take great care regarding the set of options to use. Problems will essentially come from options not at-the-money. We consider here three sources of problems.

First, with respect to what was said in Section 6.6.1, we could face large differences between target prices and arbitrage-free prices. This is a problem if the target prices are the market prices, as we will not be able to use the arbitrage-free solution to perform real investments. In particular, if the optimal solution for a portfolio optimization problem contains at least one option for which the free-arbitrage price is very different from the observed market one, then the optimal portfolio value could become suboptimal on the real market when using the market price. The problems come from the fact that the prices are defined by complex market operations which cannot all be modelled. However, this does not mean that the model is not reach enough. Often, such options should not be considered in real investment problems, e.g. for options deeply in-the-money as explained in the next paragraph, and therefore should simply be rejected before solving a portfolio problem. To detect them, we first need to optimize the option pricing problem to be able to check differences between the prices.

Second, we know that options deeply in-the-money are less traded on the market. This affects the option price. Numerical computation of the price should take into account this problem of liquidity, but this is not the case in our model nor in usual methods. It implies that larger biases between target and free-arbitrage prices will generally appear for these options. Conversely, we must be careful before using such options, as the prices are only valid for small exchanged quantities. If the optimal solution of a portfolio problem implies a large investment in deeply in-the-money option, we could expect that the price on the real market will be adjusted, and will then modify the inputs of the problem and so change the optimal investment solution. Once again, if these options are not fully traded on the market, we should not take them into account.

The difficulty is to define when an option is not enough at-the-money. We suggest as a first rule to consider the implied volatility. By inverting the Black and Scholes formula, we can compute an implied volatility from the option price. Everything else kept constant, volatility increases with the price. The minimal price is obtained for the limit to zero of the volatility. If the market price is smaller than the minimal BS price, then it is impossible to compute an implied volatility. This means that the market price depends on other factors than the ones modelled in the BS formula and that we must be suspicious with this option. Three solutions are considered here:

1. Do nothing. We consider that the target price is valid and we use it.
2. We do not consider the option anymore. This is the recommended approach.
3. We adjust the target price. If we are only interested in a theoretical study, we could replace the target price by either the BS one, or better, by using the adjusted BS formula taking the smile effect into account.

Third, a problem arises when the optimal or target prices are null or numerically close to null. This could happen as prices are only constrained to positive values. This will be the case for example when the option is out-of-the-money in the future in nearly all scenarios, but slightly in-the-money for one scenario. A null price doesn't mean that we can buy or sell an option for zero monetary unit, but that the option does not exist on the market and so that we cannot invest in this option. Therefore, we here have two possible interpretations of the option price depending on its value. If we do not take this second meaning into account, then an optimal portfolio solution could contain a quantity of non-existing options, or worse could be influated by numerical errors arising from investing huge quantities in nearly free options. Either we take this into account when defining the model for problems handling options, or, better, we simply remove these options beforehand.

Pre and post-processing with two optimization steps

The three problems presented just before can be detected at different times. The bias between prices can only be measured after option pricing optimization. The implied volatily is measured for the target prices before optimization. The null target price can be detected before optimization and the null optimal prices afterwards. The option cleaning process is then divided in two steps: one before optimization and one afterwards.

Moreover, the options removed after the option pricing optimization were used during this optimization and influate on all the other securities by the arbitrage equations (6.1).

Thus, these options have a large impact on the optimal solution as the bias is typically large for them and as the objective function is the sum of the biases. Therefore these options can occult the others. If we had not considered them initially, we would probably have obtained a lower bias for the other options.

For this reason, we improve the option pricing algorithm by starting a second option pricing optimization without these options after the first optimization. Ideally, we should do it several times until no more changes are observed. In practice, this is not optimal as it is time consuming and as one pass is enough to detect the "mispriced" options (as they are not mispriced by the optimization process, but by the market).

One more remark can be made. Post-processing results give an indication about the adequation between index and options prices. If the index probability distribution used to model the future is not coherent with the final option prices, it is more difficult to minimize the bias between the optimal prices and the target prices. Indeed, the state-prices, which represent the risk-neutral pdf, are used at once to price the initial index price and the initial option prices. In this case, more options will be removed by the cleaning post-process. Conversely, few or no options are removed in the post-process when the index distribution is computed implicitly, and so coherently, from the option prices. This will clearly appear in the numerical results of Section 6.7.

6.6.3 Selection of options

Initially, we can observe the set of available options on the market. Therefore, it is easy to define the inputs of the problem for the first period. It is however more complex when we have to define the second-period set of options. Moreover, this set depends on the scenarios.

The options that will appear on the market depend on the underlying asset. Thanks to the tree of scenarios, we know for each node all we need about the index. As explained in Sections 6.2.3, the strike prices can then be deduced automatically. It is important to notice that the set of options is defined independently for each scenario, so that each set is probably different. Option prices are obtained by the optimization process defined previously.

Another improvement could be made. The automatic procedure described above could lead to large set of options, to large sizes of problems and so to long computing times. This is because all the options are considered, even if a specialist would be able to immediately say that several options are probably not interesting for the portfolio optimization problem to be solved. With respect to the VaR portfolio problem discussed in the next chapter, we propose a pre-processing approach to initially detect the most promising options. As this is defined according to the portfolio problem, it will be dealt with in Chapter 9.

6.7 Numerical results

6.7.1 Introduction

In this section we present some preliminary numerical results. The aim is to give a feeling about the methods presented until now. In Sections 6.7.2-6.7.3, we present numerical results for the pre and post-processing described in Section 6.6. In Section 6.7.4, we show that the volatility smile exists on the S&P500 market and so that we will need the developments made in Section 6.5. Finally, in Section 6.7.5, we compare the option pricing models (6.5) and (6.7) with respect to the state-prices. Throughout all this numerical section, links are made with several parts of this work. At the end of each section, we draw some more conclusions on which models we should use and with wich inputs.

6.7.2 Option cleaning

We consider here the S&P500 index, the same set of 48 options observed on the market during one month as before, the different density functions defined in Section 5.3 and a stratification into 30 scenarios. We use the advanced option pricing model (6.7).

In the next table we give the number of options removed during the pre and post-processes for the different densities. We have decided not to remove options due to large biases between target and free-arbitrage prices and to study this problem in the subsection 6.7.3.

Density functions	Pre-processing		Post-processing
	Null price	No volatility	Null price
Normal pdf	0	6	13
Skewed T pdf	0	6	11
Implied pdf	0	6	0

(6.11)

The pre-processing requires as inputs only the target option prices and, to compute the implied volatilities, the parameters used in the BS formula. As it does not take into account the density functions used to model the index returns, the results are the same for each pdf. It is also interesting to say that the removed options are essentially put options (11 over 25 options) and that the options without implied volatilities are the deepest in-the-money options. This one pass pre-processing filter worked well as no more options are removed when we restart a new optimization process.

In the post-process, we observe that the more complex the density function, the fewer the options removed. This is an indication that the implied pdf seems to better match the observed option prices. This is not a proof that this implied pdf is the best model to use to represent the market, but only an indication of adequation between the index distribution and the option prices! Indeed, we could artificially create "mispriced" target option prices, construct from them the implied pdf and still obtain a good adequation between them.

These first results tends to suggest that we effectively should not use all the options observed on the market to model the future and to price the options according to the tree of scenarios.

6.7.3 Mean deviation

Still using the S&P500 set of data and the same different density probability distributions as before, Tables 6.12-6.13 give the mean percentage of absolute error between target prices (observed market prices or Black and Scholes prices) and free-arbitrage optimal prices obtained after considering either the model (6.7) (with risk-neutral probabilities in the objective function) or the model (6.5) (without risk-neutral probabilities in the objective function). The differences are expressed in a relative percentage form instead of an absolute dollar bias.

To be able to make valid comparisons, we have decided to work with exactly the same set of options for all the pdfs and so we have removed the 13 options rejected by the normal pdf post-processing. Therefore, we work with a clean set of options where the largest biases between target and optimal prices have already been removed.

With risk-neutral targets	Market/ Optimum	BS/ Optimum
Normal pdf	4.20%	0.14%
Skewed T pdf	3.63%	0.00%
Implied pdf	1.10%	–

(6.12)

Without risk-neutral targets	Market/ Optimum	BS/ Optimum
Normal pdf	4.13%	0.14%
Skewed T pdf	3.45%	0.00%
Implied pdf	0.96%	–

(6.13)

The mean difference between the observed market price and the BS price is about 153%. This explains why we do not obtain the same results in the last two columns of Tables

6.12-6.13.

The smallest biases are obtained when the option prices are generated by the BS formula. In this case, it is not surprising that the index representation by the Normal probability distribution gives very small residues (because of the coherency between the index pdf and the option prices). Moreover, for both market and BS targets, we again observe a decrease in error when we use more complex density distributions. Finally, it is important to notice that even if we add constraints when we set also risk-neutral targets, we do not observe a significant decrease in quality.

Figure 6.2 illustrates the complete results for option pricing model (6.7) and the implied pdf. The option prices are given on the vertical axis with respect to the strike prices on the horizontal axis. The spread around the prices is about 1.25USD; so you cannot distinguish on the figure between ask, middle and bid prices. For each option, three prices (three marks) are considered: the market price, the arbitrage-free price obtained when the target price is the market price, and the BS price. The prices are linearly interpolated between the strike prices. We observe the typical decreasing pattern for the calls and increasing pattern for the puts. In this numerical example, the BS formula always underestimates the prices with respect to the target and free-arbitrage prices. As described in Table 6.12, the free-arbitrage curves are close to the target curves.

These numerical experiments confirm our previous conclusions. First, the implied distribution allows to better match the target option prices than any other considered pdf. It should be used excepted in the case of a theoretical study based on normality assumptions. Second, the option pricing model (6.7) allows us to construct arbitrage-free option prices close to the target prices. Moreover, there are few variations of the option prices when using the model (6.7) instead of the model (6.5). We will however see in Section 6.7.5 that there is a large increase in the quality of the state-prices. In consequence, we will in priority consider the option pricing model (6.7) in the portfolio optimization problems discussed later.

6.7.4 Smile effect

Let's also have a look at the implied volatilities. The call and put market implied volatilities are represented in the Figure 6.3 as well as the free-arbitrage ones, first for the historical values of the dividend yield and risk-free rate and then for the implied values. The plot is very instructive for different reasons.

First, we know that under the BS hypothesis, volatility should be the same for all options. This is not the case, and the smile effect appears clearly in the market implied volatility

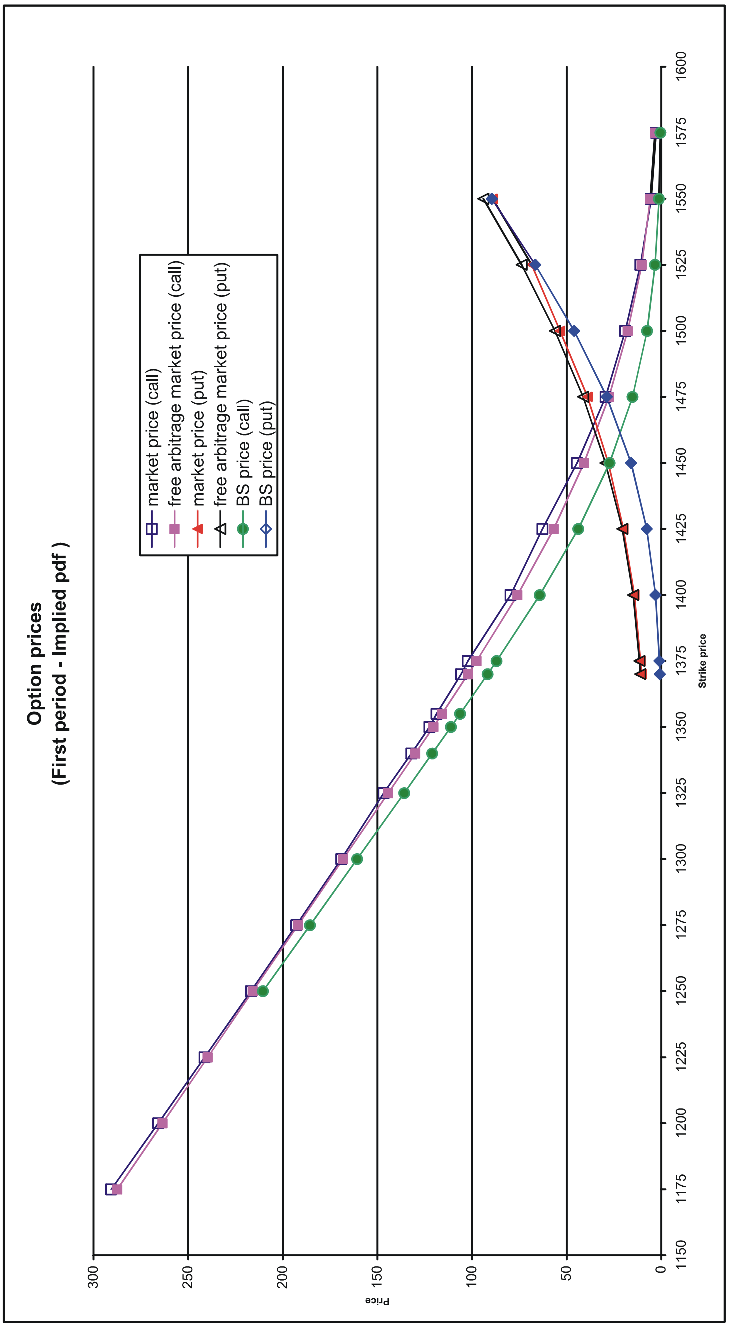


Figure 6.2: Option prices

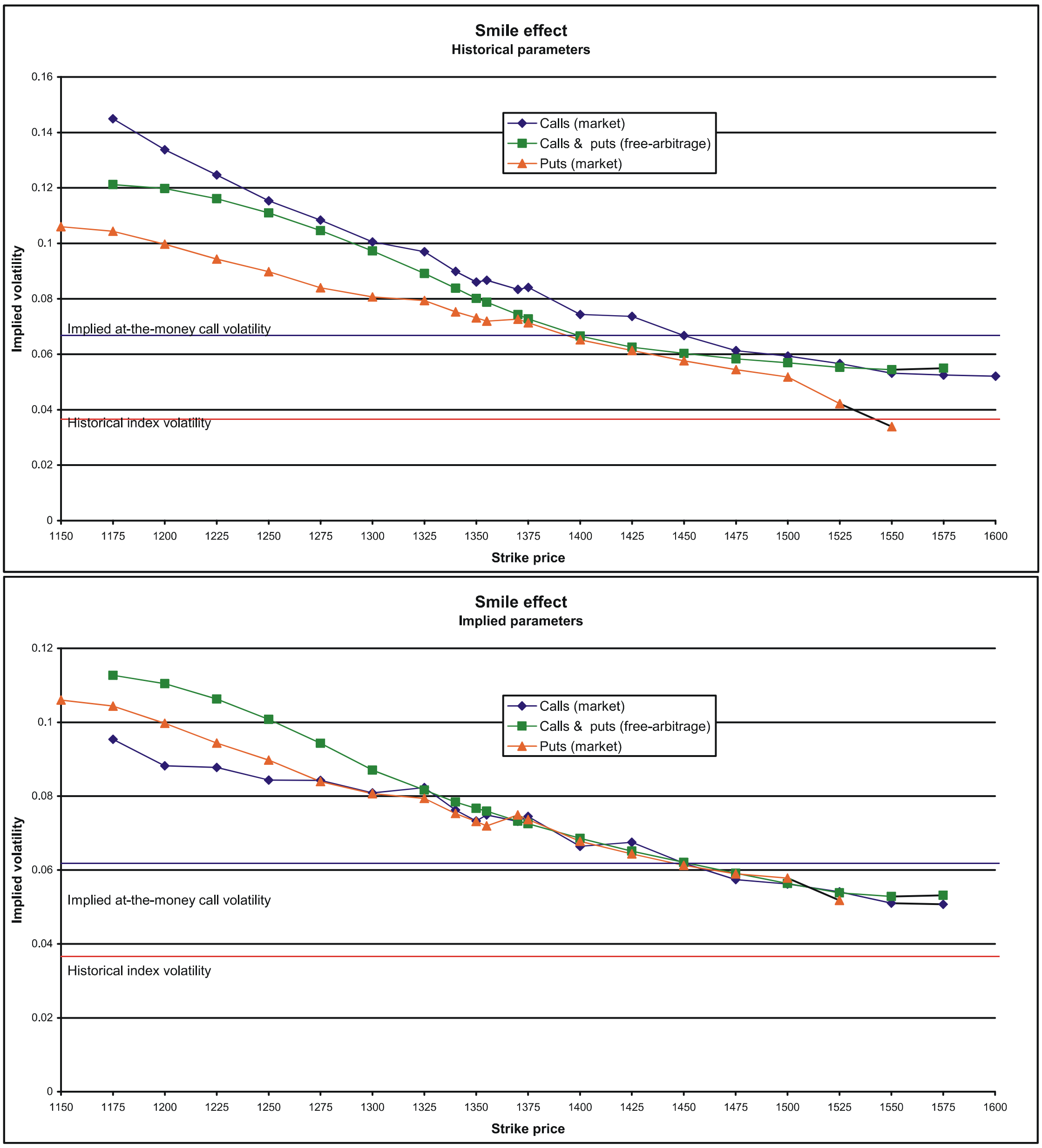


Figure 6.3: Volatility smile

curves. More important is to notice that the optimal arbitrage-free prices we have computed are characterized by a similar smile. The option pricing process has preserved this property.

Second, due to the smile convex shape, we already know that it is difficult to represent volatility by a unique value. But even if we use the most representative one, the at-the-money implied call volatility, there is still a large difference between the mean historical index volatility and this instantaneous one. This is a first example to show how difficult it is to define the parameters to use to instantiate a model.

Third, we observe a spread between the put and call implied volatilities. We know however that prices of calls and puts are linked by the put-call (arbitrage-free) parity equations. Everything else kept constant, if the parameters in the BS formula are well defined, we should then obtain the same implied volatility for the calls and puts defined with the same strike price. This property is verified for the optimal arbitrage-free prices, which by definition satisfy the free-arbitrage equations and so the put-call parity relation. Indeed, these two curves are merged in the plot. We conclude that the parameters used are not perfectly defined. Effectively, if we use the dividend yield and the risk-free rate implied by the put-call parity equations instead of the historical rates, we obtain the second plot in Figure 6.3 where the spread disappears for nearly all the strike prices. However, we still face the difficulty to compute these implied parameters. In this case, the implied risk-free rate is close to the historical one but the implied dividend yield is slightly negative which is impossible.

Note also that the free-arbitrage smile lies always between the market put and call volatility smiles. Moreover, if we consider only calls or only puts during the option pricing process, then, whatever the dividend yield and risk-free rate, the market and free-arbitrage implied volatilities are close to each other (we no longer face a put-call parity problem).

This also has an effect on the quality of the prices. For the historical yield and rate, the observed option prices and the index prices cannot satisfy the arbitrage equations as the parity equations are not satisfied. Therefore, the option pricing process cannot construct arbitrage-free prices equal to the target values at the same time for the calls and the corresponding puts; i.e. the process cannot suppress all deviations. This shows once again how important tuning of the parameters is.

As conclusion, we observe that market prices are effectively subject to the volatility smile effect and so that we have to use the advanced schemes presented in Section 6.5 to construct target option prices for the subsequent periods. We observe also that all the results depends greatly on the quality of the estimation of the inputs. Especially, the dividend yield and the risk-free rate are two difficult parameters to tune. As explained in Section 5.2.3, we do not have a perfect solution to this problem.

6.7.5 Density functions

Figure 6.4 illustrates the pdfs corresponding to the state-prices obtained using option pricing model (6.7) (with risk-neutral probability targets) and model (6.5) (without risk-neutral probability targets). The risk-neutral pdfs are only rough approximations. Indeed, in order to obtain a continuous representation from the discrete set of probabilities given by the option pricing process, we have linearly interpolated the cumulative distribution between the points of the discrete risk-neutral set of probabilities, then used the angular coefficient of each of the lines at the middle return to construct a density value. A complete pdf is obtained by a second linear interpolation between the density values. This rough procedure could partially explain some bad shapes in the figures. Moreover, we work with only ($nbS = 30$) scenarios; i.e. discrete probabilities.

The pdf constructed from the state-prices obtained by model (6.5) looks chaotic. At the opposite, the pdf constructed from the state-prices obtained by model (6.7) is closer to the target risk-neutral pdf. Clearly, we cannot use these state-prices in the subsequent models if we do not specify targets in the option pricing model. It could also be useful to improve the weighting scheme between the price term and the probability term in the objective function to reduce remaining errors.

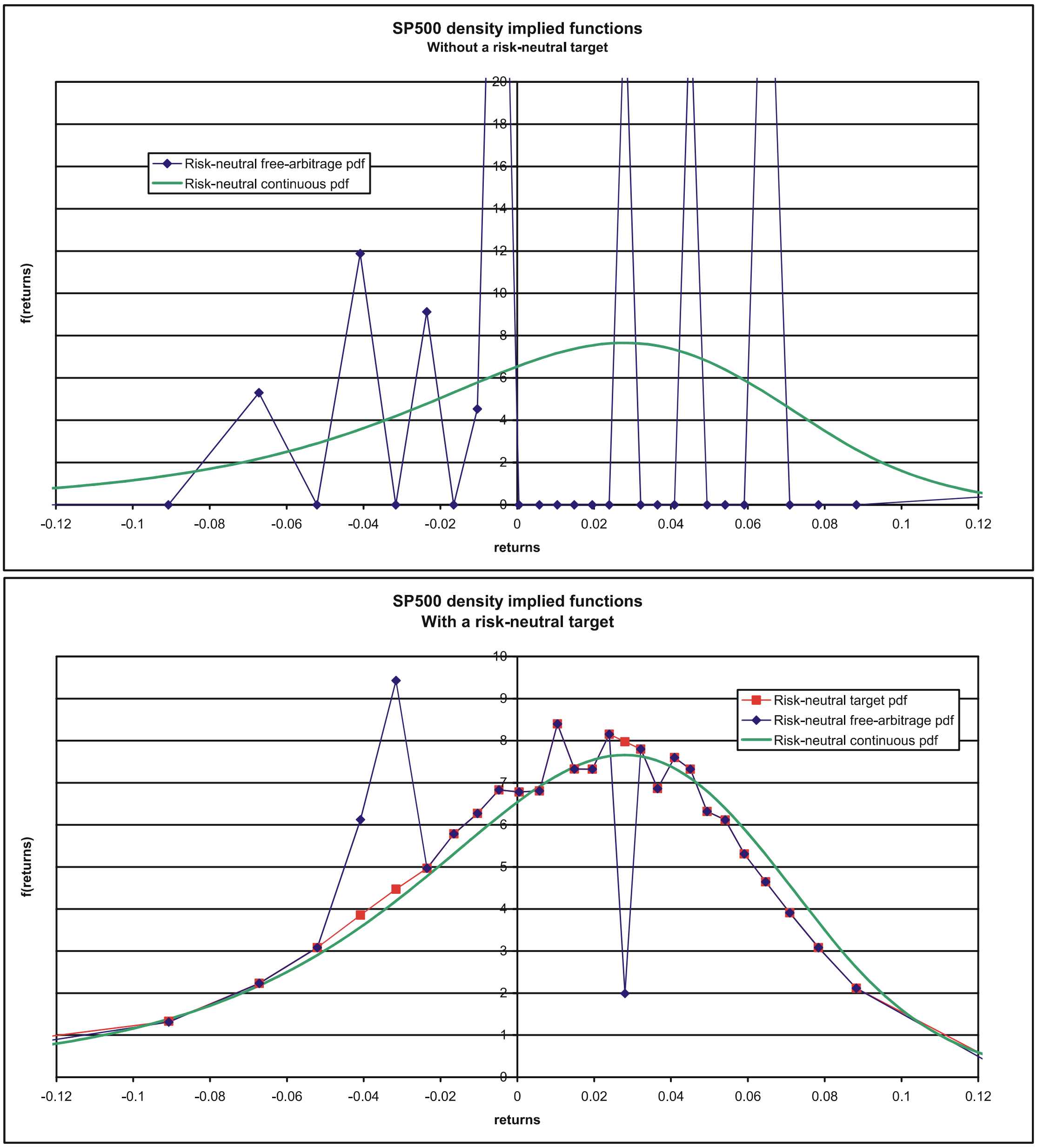


Figure 6.4: Risk-neutral pdfs

Chapter 7

Modelling Value-at-Risk constraints

7.1 Introduction

In this chapter, we introduce a new multiperiod model for the optimization of a portfolio of options linked to a single index. This model is a variant of two models due respectively to Dert and Oldenkamp [20], and to Gielen [29]. The objective of the model is to maximize the expected return of the portfolio under constraints limiting its Value-At-Risk. The future is flexibly modeled through a multiperiod scenario approach. The current model contains several interesting features, like the possibility to rebalance the portfolio with options introduced at the start of each period, explicit consideration of transaction costs and of option bid-ask spreads, realistic pricing and construction of options, etc. Section 7.2 presents the goals and constraints. The resulting mathematical model is described in Section 7.3.

7.2 The portfolio optimization problem

7.2.1 Introduction

Before presenting its mathematical programming formulation (see Section 7.3), we first propose a verbal description of the portfolio optimization model, and we discuss some of its subtleties.

7.2.2 Framework

Modelling the future and the securities

On a financial market, an investor has to take several decisions. First, he has to select the family of securities in which he will invest. He is not limited to only bonds or only stocks; e.g. he could also consider options. Second, the securities considered, but also the amounts invested in each, will depend on his risk aversion. The investor can develop specific strategies to precisely model the shape of the future payoff of his portfolio investment (insurance strategies) or to obtain some guarantees; e.g. he could want to limit the expected volatility of the future returns of his portfolio, or require, with certainty, at least a given return in the future. To do it, due to their typical payoff pattern (see Section 4.2.4), options are intensively used.

In this work, we want to give the possibility to the investor to consider all these tools and strategies and to help him to construct the optimal portfolio. We are interested in complex portfolios composed with a risk-free investment (one bond or a cash investment), a diversified portfolio of stocks (represented by an index) and options (on the index). Each of these securities has been described in Chapter 4.

We will assume, as is often the case, that the investor's main goal is to maximize the expected value of the portfolio in the future. The first task is to model the possible future events in a natural way depending on what seems judicious to the investor, in agreement with reality. So we resort to the scenario model and all the methods presented in Chapter 5.

We do not want to limit our study to a one-shot investment. Indeed, in many practical situations, it is not reasonable to assume that an investor will construct a portfolio and then leave it aside until a given date in the future. It is more likely that the investor will regularly check the performance of his investment and rebalance his portfolio to take into account new available information. However, due to various reasons (transaction costs, opening hours of markets ...), it is usually impractical to do so continuously. A discrete multiperiod model seems therefore suitable. We restrict our attention in this thesis to a two-period model, but the model could easily be generalized.

To be able to achieve a given objective at the end of the second period, the investor has to take some decisions at the beginning of the first period. Clearly, the initial and rebalancing decisions are not independent. Therefore, our model tries to optimize the decisions over the whole portfolio life time with respect to the possible rebalancing dates. As mentioned in Section 5.1.2, the first aim of the two-period approach is to model the degree of freedom given to the investor when he can dynamicaly rebalance his portfolio. Of course, in practice, the investor would not rebalance his portfolio at the end of the first period according to the optimal solution found initially, but he would restart a new optimization by taking into account all available information and for the next two periods. In other words, he would adopt a roll-over strategy. For this reason, we feel that considering more than two periods at a time adds complexity to the model without necessarily improving significantly the quality of the initial investment. This hypothesis will not be tested in our work, but could become the subject of further investigations.

As we want to model the future by a two-period tree of scenarios and to consider options, all the material presented in Chapter 6 will be clearly of interest in this portfolio optimization model.

Modelling the risk and the investment limits

We want to consider investor strategies currently used in practice. Until now we have defined the global goal: maximizing the expected value of the portfolio. Of course, the investor faces a variety of constraints which restricts his possibilities. The main ones are first, to be limited by a given budget, and second, to limit the exposure to risk, especially when the investor works with client's budgets. To reduce the risk, the classical Markowitz' approach is to impose an upper bound (or is to minimize) the volatility of the portfolio, i.e. the standard deviation of the returns. This method has several drawbacks. In particular, the standard deviation makes no difference between returns which are higher or lower than the expected return, even if only the lower returns are actually bad news for the investor. In other words, the standard deviation is a symmetric tool used to measure a disymmetric financial concept. Also, the standard deviation is by definition the expected deviation of the returns from the mean value. Therefore, even if the investor sets an upper bound on the standard deviation, i.e. on the mean deviation, the final portfolio return could be any value. In worst cases, the deviation between the final return and the expected one can be far larger than the standard deviation. In other words, the investor cannot limit his loss using only the standard deviation as measure of risk.

For the previous reasons, lots of investors prefer to use Value-at-Risk (VaR) measures of the risk. Value-at-Risk is defined as the maximal loss of the portfolio value with a given probability and over a specific horizon. This concepts, already presented by Edgeworth [25] in 1888, becomes very popular when introduced by JP Morgan in RiskMetrics^{TM} [62] in October 1994 (RiskMetrics^{TM} is a set of tools allowing the users to estimate their exposure to market risk under the "Value-at-Risk framework"). This measure of risk has the advantages to only measure the negative deviation of the portfolio values, and to be defined according to a probability and a time period, which is not the case of typical measures of risk. However, VaR has also some "drawbacks". First, it is an incomplete measure of risk. It is possible to obtain two optimal portfolios with the same expected value, and which both satisfy the VaR constraint, but however have very different payoffs. In particular, the investor could face huge losses for some index prices if they are compensated by large enough profits for other index prices. It is why we will also add a guarantee constraint. Secondly, as we will show in Chapter 8 and Chapter 10, the porfolios subject to VaR constraints have typical flat payoff structures coinciding with the lower bounds defined by the constraints.

In short, the main constraints we set in the model are:

- Budget:	the initial cost of settlement is less than a given initial budget. Moreover, at the beginning of the second period, the investor cannot invest in a new portfolio more than the value of the current one. There is no new cash-flow.
- Guarantee:	the final value of the portfolio cannot be less than a predefined fraction of the initial budget under any circumstances.
- Value-at-Risk:	with a predefined probability, the final value of the portfolio cannot be less than a predefined fraction of the initial budget.

Note already that the guarantee constraint is a special case of the Value-at-Risk constraint where the probability is set to 100%. We could actually impose several VaR constraints in the model. The methodology would be about the same for each of them, even if the process could be optimized (several VaR constraints would imply overlapping lower bounds).

Note also that these VaR constraints are only applied at the end of the second period. We are ready to accept more risk at the end of the first period if we are certain that we can finally control it. However, obviously, the VaR constraints have consequences for all the periods, since the decisions taken for the first period must ensure that all the constraints can be satisfied at the end of the second period. This is a kind of backpropagation of the VaR.

The budget constraint is applied initially and at each leaf of the first period tree of scenarios. Initially, it can be viewed as a normalizing constraint. At the beginning of the second period, we have not allowed any new extra cash-flow wich would increase the budget. Allowing this possibility could be done easily but is without interest if the extra cash-flow is the same for each second-period subtree. Indeed, in this case, the corresponding amount can be discounted at the risk-free rate and added to the initial budget. Otherwise, if the extra cash-flow depends on the scenario at the end of the first period, extensions of the model can be considered, but difficulties arise to define the extra cash-flow as a function of the scenario.

In some situations, we add one more constraint on the index. Indeed, we are essentially interested by handling portfolios of options. The index, which is not traded on markets, has been presented because the options we consider are defined according to it. Therefore, we will construct models where the index can appear or not in the portfolio.

Excepted in the previous particular case, we did not add bounds on the quantities invested although this is often applied on financial markets. These constraints could be easily modelled.

Modelling additional features of the market

In order to model some additional features of real markets, we take into account the transaction costs for the index and the options when necessary. We also define a bid and a ask price for the options.

7.2.3 Time of creation and maturity of the options

Initially, we can observe the options available on the market and introduce their definition in the optimization model to be solved. However, we would also like to determine initially what options will be available on the market at the end of the first period. Indeed, in Section 7.2.2, we have decided that the investor can adjust his portfolio at the beginning of the second period. Therefore, we need to know, for each subtree of the second period, the set of securities that are available at this time on the market. This will lead to an extension of the model where new options can be purchased or sale at the beginning of the second period, and where different options could be handled for each subtree of the second period.

Also, we do not want to restrict the choice of options to those with maturity at the end of the first or of the second period: the model can handle options with a maturity after the end of the second period ($t2$). All we need to know is a value for these options at $t2$ which we obtain by applying for example the BS method.

Thus, we consider three sets of options in the model : those covering only the first period, those covering (at least) the two periods, and those which do not appear until the beginning of the second period. We make no difference here between options with maturity exactly at $t2$ or those with maturity after $t2$.

Working with options which cover only the second period has several advantages:

- It is realistic to model the possibility to adapt the portfolio with securities which are not available initially but will be available at the rebalancing date.
- This allows to consider a different set of options for each tree of scenarios of the second period. This is realistic since the set of options found on real markets depends on the price of the underlying stock or index.

- If we consider options only for the second period (or/and only for the first one), the option pricing processes, performed locally at each subtree, returns optimal solutions over the two-periods. In this case, we do not need to resort to the SA heuristic presented in Section 6.4.4. It is not true anymore when we consider options covering both periods.

- Some specific financial tools and theorems can be used when we deal with complete market to improve the optimization methods. As, each subtree of the second period corresponds to a market problem, it is easier to complete all or some subtrees thanks to options covering only the second period than with options covering both ones.

7.2.4 The costs of transaction

Basically, the portfolio value is obtained by adding up the values of each of its components. But things become in fact a little bit more complex in the presence of transaction costs and bid-ask price spreads.

As explained in Section 6.2.10, commissions consist of a fixed value and a variable rate. They vary between predetermined minimum and maximum values. Moreover, the commission value depends on the absolute dollar amount exchanged, and not on the quantities of securities traded. Our model is specially intended for large investments and we will use this property to simplify the reality. If the initial budget is big, then the fixed commission cost is small with respect to the whole amount invested and it can be neglected. For the same reason, as the amount invested in each option can be considered as large, we can use only one variable rate of taxation; i.e. the rate which applies to the largest trading amounts. Such a percentage is applied on the options and on the index (which can be viewed as an option with a null strike price) at each of the three dates of interest in the model ($t0$, $t1$, $t2$).

However, a first complication occurs because the costs of transaction must not always be applied to the options at maturity. When the investor has a long position on an option for which, due to the minimal absolute cost of transaction, the price to receive is smaller than the cost of transaction to pay, then this investor will not exercise the option but trash it. So, for long positions, we have to model the option value at maturity not only with respect to the index price, but also with respect to the cost of transaction. In the case of a short position, we consider that the investor will pay the due amount to the counterpart and the transaction costs only when the option is in-the-money, i.e. has a positive value. We do not consider here that the investor on the other side of the transaction, and so with a long position, could trash the option, and so not required the amount due, even when the option is slightly in-the-money, due to transaction costs larger than the option value. Indeed, this would imply increasing the complexity of the model to include the possible behaviours of the different investors on the market and to handle very small amounts.

The minima and maxima on commissions are applied on the basis of a one hundred size contract. But in our models, we consider that we can divide the contracts and that we can buy or sell individuals options. Indeed, if, as we suppose, the quantities of options traded in the optimal portfolio are large, then the one hundred size of the contract is small with respect to these quantities and the divisibility assumption is a small approximation. So, we simply divide the minima and maxima by one hundred to obtain the values per option for our model.

The costs of transaction are obviously linked to the dollar amounts traded, and not to the resulting positions. This is important for options covering the two periods. For instance, if we modify the position in one option at the beginning of the second period (sell at $t0$ and purchase at $t1$ or the converse), then we have to compute the transaction cost on the basis of the modification and not on the basis of the new position. This implies that, in the mathematical formulations, we have to distinguish between the (absolute) quantities traded initially and those traded at the beginning of the second period.

Bid and ask prices

In order to be realistic, we need to define different prices for each option depending on whether the investor wants to sell the option (bid price) or buy it (ask price). The bid price is always smaller than the ask price. We have explained in Chapter 6 how these prices are set in our models.

The existence of bid and ask prices has of course direct consequences on the portfolio value, but also on the model itself. In particular, we need to be able to make a distinction between sales and purchases of options. This is most difficult for options covering both periods, because we can purchase them and sell them at two different dates. Thus, we have to take into account the absolute positions, and exclude to hold simultaneously a short and a long position on a same option. In the mathematical formulations, we have to work with one sum of the quantities traded initially and at the beginning of the second period, and we must be able to distinguish whether the absolute final position is short or long. Remember also that in order to handle the transaction costs, we must distinguish between the the quantities traded at $t0$ and $t1$ and not only the absolute positions.

We also have to model the fact that the bid and ask prices are not used to evaluate the value of the options at maturity. Instead, the value of each option at this time is given directly by the difference between its strike price and the index price (see Section 4.2.4).

We model the spread as a percentage of the option value, but we also impose that its absolute value cannot become smaller than a given minimal constant. This is financially realistic and also numerically attractive, as it allows us to avoid numerical arbitrage opportunities as explained in Section 6.4.5.

7.2.5 Other option features

As explained in the previous chapter, the other characteristics of the options, e.g. the strike-price and the maturity, can be determined by the market rules. Especially, we can automaticaly define the options and the corresponding strike prices that should appear on the market for each of the subtrees of the second period.

We will also describe in Section 9.4.3, a heuristic which allows to preselect promising options to consider in each specific problem instance and so to reduce the problem size.

7.2.6 The guarantee constraint

Basic approach

The guarantee constraint requires that whatever happens in the future, the portfolio value cannot become smaller than a given guaranteed level. The easiest way to model this constraint is to impose it for every possible scenario at the end of the second period ($t2$), thus creating a number of constraints equal to the number of scenarios at $t2$. A subtle, but bothering, drawback of this approach is that it does not ensure that the constraint will be satisfied if the future state of the world turns out to be different from those explicitly considered in the tree, even if this actual outcome only represents a slight modification of one of the explicit scenarios. This is illustrated in Figure 7.1.

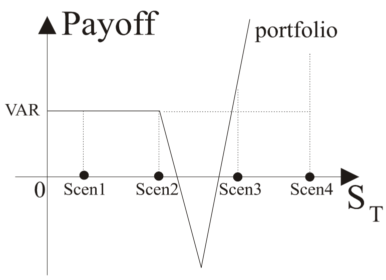


Figure 7.1: Leak in the guarantee constraint

This is especially true if the representation of the future is poor because the number of scenarios is too small, or because the sample distribution is bad. When the sample size increases, the probability that the guarantee constraint is always satisfied also increases, but the size of the problem and the numerical difficulties explode. In our preliminary experiments, sample sizes smaller than 50 proved insufficient and gave rise to the phenomenon displayed in Figure 7.1, while sample sizes larger than 100 were numerically intractable. Therefore, we had to turn to alternative modelling approaches.

Improved approach

Dert and Oldenkamp [20] proposed another model. If none of the options comes to maturity after the end of the second period, then the portfolio value is a linear combination of the values of its components. Indeed, the components are the risk-free asset (which is independent of the index), the index and the options. A call (put) option has no value if the index price is lower (higher) than the strike price, and its value increases (decreases) linearly if the index price is higher (lower) than the strike price; i.e. the option value is a piecewise linear function of the index. This implies that the portfolio value also is a piecewise linear function of the index, with break points at each option strike price (see Figure 7.2).

Consequently, if the guarantee constraint is satisfied at two consecutive strike prices, then it is also satisfied for all intermediate prices. This only leaves us with the two extreme possible values of the index (zero and infinity) to worry about. For index prices lower than the smallest strike price, we need to make sure that the portfolio value remains above the guarantee level. As the lowest possible index value is zero, we can simply add a virtual strike price of zero. For index prices higher than the largest strike price, we must make sure that the portfolio value is not a decreasing function of the index value, since otherwise, the portfolio value will tend to minus infinity when the index value tends to infinity. Note that for this last case, all the put options are out-of-the-money (valueless) and all the call options are in-the-money (valuable). This is illustrated in Figure 7.2.

This model has several advantages over the previous one. First, it ensures that the guarantee constraint is always satisfied even for small trees of scenarios, which allows to reduce the numerical complexity of theproblem. The number of constraints required in this model is also typically (when the market is uncomplete) smaller than in the previous case, since the number of strike prices (linked to the number of options) is typically smaller than the number of scenarios.

Unfortunately, Dert and Oldenkamp [20] made the strong assumption that all the options

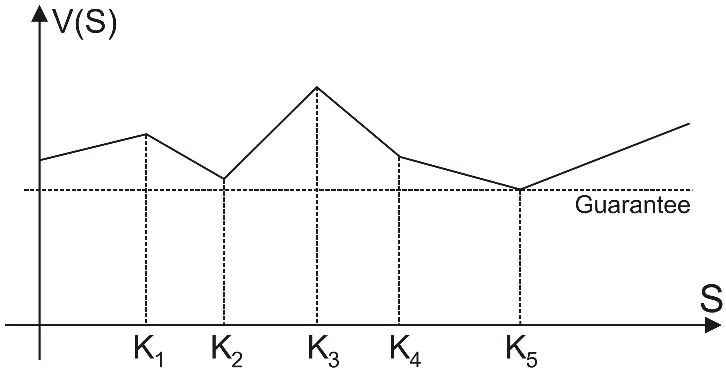


Figure 7.2: Portfolio value as a piecewise linear function of the index

expire at the investment horizon. Indeed, the payoff function an option is a piecewise linear function of the underlying asset at maturity only. Before this time, the value of an option is the sum of its intrisic value (the piecewise linear function), its time value (a constant) and its volatility value (a nonlinear function) as explained in Section 6.2.6. The result is a nonlinear function. If we consider options with maturities after the end of the investment horizon, Dert and Oldenkamp's model cannot be used. Figure 7.3 illustrates the problem for a portfolio composed of one call and one put. The constraints at the strike prices are satisfied, but however the portfolio value is under the guarantee level for infinitely many index values between two strike prices.

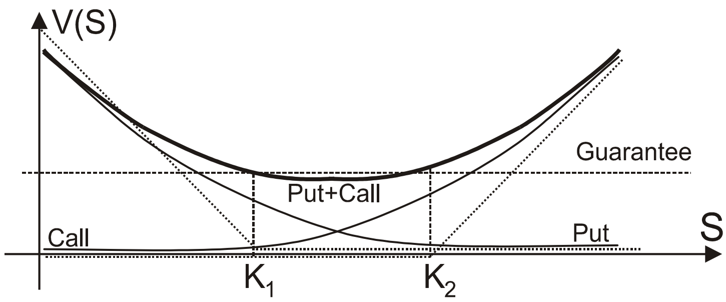


Figure 7.3: Non linear leak

If all the options expire at time $t2$, Dert and Oldenkamp's model is clearly better than the first model. Otherwise, if some options come to maturity after $t2$, both approaches do not perfectly model the guarantee constraint. We have not performed in-depth analyses to decide which one leads to the smallest leaks, but some comments can be made. In Figure 7.3 representing Dert and Oldenkamp's model, the leak is not as sharp as in Figure 7.1 representing the scenario approach. Therefore, we could expect that the leaks are smaller with Dert and Oldenkamp's model. However, the depth of the leak first comes from the quantities of options in the portfolio. If we keep the same portfolio as the one described to construct Figure 7.1, but that we increase the quantities of the two options by a same factor, then the violation increases. Second, the importance of the violation also depends on the distance between the prices where the guarantee level is imposed. In Dert and Oldenkamp's model, the prices considered correspond to the strike prices of the options, and, in the other approach, to the prices of the index at the leaves of the tree. As stated previously, when the market is uncomplete, the number of options is smaller than the number of scenarios. Therefore, the strike prices leave more space to the violations. For these reasons, in the sequel, we use Dert and Oldenkamp's model if all the options expire at time $t2$, and we otherwise prefer the other model, with a large number of scenarios.

A second problem appears with the cost of transactions. These costs reduce the payoffs and this should be taken into account in order to satisfy strictly the guarantee constraint. Moreover, due to the minimum and maximum limits on the transaction costs, the reduction depends on the option value. Modelling this effect induces an increase in complexity. As this remains a side effect, we propose two simplified models in Section 7.3.10.

7.2.7 The Value-at-Risk constraints

In our model, we will consider that the VaR constraint is satisfied when the sum of the probabilities of the leaves of the tree where the portfolio value is larger than the VaR lower bound, is larger than the VaR probability. As in the basic approach developed for the guarantee constraint, the lower bound associated with the VaR constraint is only checked at the leaves of the tree.

As the lower bound must no longer be satisfied at all the leaves as in the guarantee constraint, but only at some such as to obtain the minimal VaR probability, the optimization problem is to decide at which leaf of the tree to apply the lower bound on the portfolio value. Indeed, it is not optimal to apply it at all the nodes, or at more nodes than required, as this results in a more constrained problem. This allocation problem can be reformulated as to decide, for each scenario, whether or not to apply the lower bound. Due to the binary choice to perform at each leaf, this becomes a Mixed Integer Problem (MIP).

We face the same problem of leaks than for the guarantee constraint, especially if the number of scenarios is small. However this problem is reduced by the fact that the constraint must only be satisfied for a portion of the probability distribution of the portfolio values (according to the VaR probability), and as the guarantee constraint defines a second lower bound under the VaR constraint (which is a strict lower bound when Dert and Oldenkamp's model is used). In our preliminary experiments, the results are very promising when, after the portfolio optimization process, we construct a new tree with a larger set of leaves to only check if the VaR constraint is still satisfied when considering more index values than the ones used during the optimization. Indeed, in conjunction with Dert and Oldenkamp's guarantee model, the VaR probability is always close to the required probability.

Note that a model similar to Dert and Oldenkamp's guarantee model cannot be developed for the VaR constraint. Indeed, this approach was defined to ensure a strict (100% probability) lower bound on the portfolio values and cannot be directly relaxed to a given VaR probability.

7.3 The mathematical programming model

7.3.1 Introduction

We are now ready for a mathematical programming formulation of the portfolio optimization model which we went to tackle. This model is inspired by a similar model due to Gielen [29], but presents some added features as the possibility to consider options covering only the second period, the definition of trees with unequiprobable leaves, the use of the improved model for the guarantee constraint, and the construction of more realistic transaction costs and bid-ask spreads. Also, all the models presented in Chapter 5 and Chapter 6 were developed to construct valid and realistic inputs for the model presented in this chapter. Finally, in Chapter 8 and Chapter 9, we propose additional constraints to this model to construct optimization heuristics.

7.3.2 Notations

Parameters:

Modelling the future	
$t0$:	beginning of the first period: time of the initial investment
$t1$:	end of the first period and beginning of the second one: rebalancing of the portfolio
$t2$:	end of the second period: horizon of investment
nbS:	number of scenarios per subtree in each period
Pr_i :	probability of scenario i

r:	risk-free rate
$s0$:	index prices at time $t0$ in scenario i as defined in Chapter 5
$s1_i$:	index prices at time $t1$ in scenario i as defined in Chapter 5
$s2_{ij}$:	index prices at time $t2$ in scenario j of subtree j as defined in Chapter 5
q_1, q_2:	index dividend yield for the first and the second period
$tstock$:	transaction cost for the index (rate)

Modelling the options

$O1$:	set of options covering only the first period.
$O2_i$:	set of options covering only the second period in subtree i.
$O12$:	set of options covering both periods.
$nbO1, nbO2_i$, $nbO12$:	number of options in $O1$, $O2_i$ and $O12$.
$popt$:	option price without spread nor transaction cost.
$cost$:	option costs (transaction and spread).
$vopt1\{ab\}_{tki}$:	ask or bid price of option k ($k \in O1$) in scenario i at time t with transaction costs (as defined in Chapter 6).
$vopt2\{ab\}_{tki}$:	ask or bid price of option k ($k \in O2$) in scenario i at time t with transactioncosts (as defined in Chapter 6).
$vopt12\{ab\}_{tki}$:	ask or bid price of option k ($k \in O12$) in scenario i at time t with transaction costs (as defined in Chapter 6).
$popt12_{tki}$:	price of option k ($k \in O12$) without spread nor transaction cost. in scenario i at time t
$cost12\{ab\}_{tki}$:	costs (transaction and spread) of option k ($k \in O12$) in scenario i at time t in the case of a purchase (a) or a sale (b) .
$spread\{ab\}_k$:	spread to be applied to the option value in order to obtain the ask and the bid price; expressed as a percentage of the option value.
$spreadmin$:	absolute minimal bid-ask spread price
$topt$:	transaction cost for the options (rate)
$tmin$:	minimal transaction cost (abolute value per option)
$tmax$:	maximal transaction cost (abolute value per option)

Modelling the constraints	
B:	initial budget
θ:	percentage of the initial budget to be guaranteed at $t2$
u:	minimal probability defining the VaR constraint
λ:	percentage of the initial budget to be guaranteed at $t2$ with probability u

Variables:

Modelling the index	
$y0a$:	quantity of the index purchased (ask) at $t0$.
$y0b$:	quantity of the index sold (bid) at $t0$.
$y1a_i$:	quantity of the index to be purchased at $t1$ if scenario i materializes.
$y1b_i$:	quantity of the index to be sold at $t1$ if scenario i materializes.
$absy_i$:	absolute index quantity owned during the second period $= \lvert y0a + y1a_i - y0b - y1b_i \rvert$.

Modelling the risk-free asset	
$z0$:	amount invested in the risk-free asset at $t0$.
$z1_i$:	amount invested in the risk-free asset at $t1$ if scenario i materializes.

Modelling the options	
$x1\{ab\}_k$:	quantity of the option k ($k \in O1$) purchased (ask) or sold (bid) at $t0$.
$x12\{ab\}_{0k}$:	quantity of the option k ($k \in O12$) at $t0$.
$x12\{ab\}_{1ki}$:	quantity of the option k ($k \in O12$) purchased or sold at $t1$ in scenario i.
$absx12a_{ki}$:	long position of option k ($k \in O12$) in scenario i. $= \max(0, x12a_{0k} + x12a_{1ki} - x12b_{0k} - x12b_{1ki})$.
$absx12b_{ki}$:	short position of option k ($k \in O12$) in scenario i. $= \max(0, x12b_{0k} + x12b_{1ki} - x12a_{0k} - x12a_{1ki})$.
$x2\{ab\}_{ki}$:	quantity of the option k ($k \in O2$) purchased or sold at $t1$ in scenario i.

Modelling the VaR constraint	
α_i:	equals one if the VaR lower bound is applied in scenario i and equals zero otherwise

Note that all decision variables, with the exceptions of $z0$ and $z1_i$, will be constrained to assume nonnegative values only.

Ask price:	
If at maturity **then** vspread:=0;	//No spread at maturity
Else if $popt * (spreada + spreadb) < spreadmin$ **then** {	//Minimal spread
dif=$(spreadmin - popt * (spreada + spreadb))/2$;	(see Section 6.4.5)
vspread=$popt * spreada$ + dif;	
} **Else** vspread=$popt * spreada$;	//Standard spread
price := $popt$ + vspread;	//Ask price
ctransac := price $*topt$;	//Costs of transaction
If ctransac < $tmin$ **then** ctransac := $tmin$;	//Lower bound on costs
Else if ctransac > $tmax$ **then** ctransac := $tmax$;	//Upper bound on costs
If at maturity **and** price <= 0 **then** $costa$:= -$popt$;	//Trash option
Else $costa$:= vspread + ctransac;	//Option costs
$vopta := popt + costa$;	//Option value

Table 7.1: Ask price

7.3.3 Computation of the bid-ask spread and costs of transaction

The value of an option takes into account the bid-ask spread and the commissions. We have to define a procedure to compute all these extra costs in every possible case. This includes the suppression of the spread at maturity, the consideration of bounds on the transaction costs and the rejection of options with negligeable value. In the case of the purchase of an option or the settlement of a short position at maturity, the computation of the option value and of the sum of the costs is performed as described in Table 7.1. Note that to trash a valueless option, we simply set its value to zero by adjusting the costs. Comments are added on the right.

The same is done in Table 7.2 for the computation of the bid price in the case of the sale of an option or for the exercice of a long position at maturity. The main differences is that we remove the costs of transaction to the price we receive when selling the option and that we trash options for which the cost of transaction to pay is larger than the price to receive.

7.3.4 The budget constraints

Initially, the price that the investor pays to acquire his portfolio must be smaller than or equal to his budget.

To compute the cost of the investment in each security, we need to make a distinction between the purchases and the sales. This is because the bid and ask prices are different and because we need to identify the kind of position, short or long, in order to apply correctly

Bid price:	
If at maturity **then** vspread:=0;	//No spread at maturity
Else if $popt * (spreada + spreadb) < spreadmin$ **then** {	//Minimal spread
dif=$(spreadmin - popt * (spreada + spreadb))/2$;	(see Section 6.4.5)
vspread=$popt * spreadb$ + dif;	
} **Else** vspread=$popt * spreadb$;	//Standard spread
price := $popt$ - vspread;	//Bid price
ctransac := price $*topt$;	//Costs of transaction
If ctransac < $tmin$ **then** ctransac := $tmin$;	//Lower bound on costs
Else if ctransac > $tmax$ **then** ctransac := $tmax$;	//Upper bound on costs
If at maturity **and** price < ctransac **then** $costb := popt$;	//Trash option
Else $costb$:= vspread + ctransac;	//Option costs
$voptb := popt$ - $costb$;	//Option value

Table 7.2: Bid price

the costs of transaction. For the index and for the options, we define two sets of positive variables which model the bid and ask transactions. This could be done also for the risk-free investement if we want to define two risk-free rates (lending or borrowing). Of course, the optimal solution will be such that, for each particular asset, short and long positions will not be simultaneously taken.

The initial budget constraint is modelled hereafter:

budget0 (1) :

$$\begin{aligned}&(y0a - y0b)s0 + (y0a + y0b)s0\ tstock + z0 + \sum_{k}^{nbO1} (x1a_k\ vopt1a_{0k} - x1b_k\ vopt1b_{0k})\\ &+ \sum_{k \in O12} (x12a_{0k}\ vopt12a_{0k} - x12b_{0k}\ vopt12b_{0k}) \leq B\end{aligned} \tag{7.1}$$

Note that as explained in the previous subsection, the parameters $vopt$ include the bid-ask spread and the transaction costs, with respect to the bid or ask move. No option covering only the second period, nor dividend yield appear in $t0$.

Let us now turn to the budget constraint at time $t1$, when the situation is a little bit more complex. There is no new external budget to invest at $t1$, but some funds are provided by the initial portfolio. First, there is an extra amount available provided by the options at maturity at the end of the first period. Second, the index provides dividends. Finally, as there is only one risk-free rate and no cost of transaction on the risk-free investment, we artificially decide to close the risk-free position (with a payoff of $z0e^{r(t1-t0)}$) and to open a new one ($z1$) for the second period.

For each of the scenarios at the end of the first period, we can rebalance differently the portfolio while preserving the budget. The left hand term in each of the equations 7.2 models the new investment decisions. The right hand term models the new available budget provided by the initial portfolio.

The investor can rebalance the portfolio by buying or selling the index, by buying or selling options, and by lending or borrowing. The transaction costs must be applied only to the new quantities exchanged on the market and not to the global quantities in the portfolio. Indeed, transaction costs have already been paid for the first period. Moreover, the investments can be different for each scenario and so one set of variables is not enough. For these reasons, we need to define new vectors of variables to model the new quantities purchased or sold at $t1$ for each of the subtrees of the second period. It is also why the initial vectors of quantities $y0\{ab\}$ and $x12\{ab\}_{0k}$ do not appear to measure the index and option contributions to the portfolio value in these budget equations, as they are "maintained" in the portfolio without any change (and so without any effect on the budget). Note however, that $y0\{ab\}$ still appears to measure the dividend yield perceived at the end of the first period. For each security, the new global quantities are expressed by the sum of the four variables: bid and ask quantities exchanged at $t0$ and at $t1$. Note also that at $t1$, we close the positions for the options at maturity. This implies that the $vopt1b$ values are used to close the long positions ($x1a$) and the $vopt1a$ values for the short positions ($x1b$).

$$
\begin{aligned}
&\textbf{budget1 (nbS)}: \forall i \in [1, nbS] \\
&\quad (y1a_i - y1b_i)s1_i + (y1a_i + y1b_i)s1_i \; tstock + z1 \\
&\quad + \sum_{k \in O2_i} (x2a_{1ki} \; vopt2a_{1ki} - x2b_{1ki} \; vopt2b_{1ki}) \\
&\quad + \sum_{k \in O12} (x12a_{1ki} \; vopt12a_{1ki} - x12b_{1ki} \; vopt12b_{1ki}) \\
&\quad \le (y0a - y0b)s1_i(e^{q1(t1-t0)} - 1) + z0e^{r(t1-t0)} + \sum_{k \in O1} (x1a_k \; vopt1b_{1ki} - x1b_k \; vopt1a_{1ki})
\end{aligned}
\tag{7.2}
$$

7.3.5 The guarantee constraint: first approach

In this first approach, in order to satisfy the constraint, the portfolio value must be larger than the guarantee level θB for each of the nbS scenarios of each of the nbS subtrees of the second period. To compute the portfolio value, we consider that the investor sells it at $t2$.

If the investor has purchased some amount of the index at $t0$ and sold some amount at $t1$, his goal must have been to reduce the position and not to have two opposite positions in the same portfolio. Similar statements hold for the opposite transactions and for the options covering the two periods. This implies that at $t2$, we must not close separately the two positionsat $t0$ and $t1$ by two separate transactions, but only by one. That is, we must apply the new costs of transaction to only one closing transaction for a quantity given by the absolute value of the sum of the previous quantities exchanged. The absolute position for the index is modelled by the variables $absy_i$, the $nbOpt12$ options are modelled by variables $absx12a_{ki}$ and $absx12b_{ki}$. Model 7.3 is correct. Indeed, when optimizing the objective function 7.5, each of the variables $absy_i$, $absx12a_{ki}$ and $absx12b_{ki}$ will be minimized, and therefore, in conjunction with their defining constraints in 7.3, will take the correct absolute values.

$$
\begin{aligned}
&\textbf{final index position } (\mathbf{nbS}) : \forall i \in [1, nbS] : \\
&\quad y0a + y1a_i - y0b - y1b_i \leq absy_i \\
&\quad -(y0a + y1a_i - y0b - y1b_i) \leq absy_i \\
&\textbf{final option position } (\mathbf{nbS.nbO12}) : \forall i \in [1, nbS], \forall k \in O12 : \\
&\quad x12a_{0k} + x12a_{1ki} - x12b_{0k} - x12b_{1ki} \leq absx12a_{ki} \\
&\quad -(x12a_{0k} + x12a_{1ki} - x12b_{0k} - x12b_{1ki}) \leq absx12b_{ki} \\
&\textbf{guarantee } (\mathbf{nbS^2}) : \forall i \in [1, nbS],\ \forall j \in [1, nbS] \\
&\quad (y0a + y1a_i - y0b - y1b_i) s2_{ij}\ e^{q2(t2-t1)} - absy_i\ s2_{ij}\ tstock + z1_i\ e^{r(t2-t1)} \\
&\quad + \sum_{k \in O12} (absx12a_{ki}\ vopt12b_{2kij} - absx12b_{ki}\ vopt12a_{2kij}) \\
&\quad + \sum_{k \in O2_i} (x2a_{1ki}\ vopt2b_{2kij} - x2b_{1ki}\ vopt2a_{2kij}) \geq \theta B
\end{aligned}
\tag{7.3}
$$

7.3.6 The VaR constraint

The VaR constraint is similar to a guarantee constraint, but must hold in probability only. At each of the final leaves, we check if the portfolio value is larger than the given VaR level λB. If all the leaves are equiprobable, the constraint must be satisfied at least for a fraction u of scenarios. Conversely, the VaR lower bound can be violated at most a fraction $(1 - u)$ of all scenarios. If each leaf has a distinct probability, then we have to adjust the number of scenarios according to these probabilities.

We model this constraint by associating a binary decision variable α_i to each scenario i. If α_i equals zero, then the lower bound holds for scenario i. Otherwise, the constraint is relaxed and the lower bound may hold or not. In order to relax the constraint when α_i equals one, we add a term $M\alpha_i$ to the portfolio value, where M is large enough to satisfy the VaR lower bound irrespective of the portfolio value. On the other hand, the value M is chosen as small as possible, so as to tighten the problem formulation (as in [**?**, Nem]. As the smallest possible value of the portfolio is given by the guarantee level θB, the minimal amount M required to satisfy the VaR bound is given by the difference between the two levels; i.e. $(\lambda - \theta)B$.

Mathematically, we obtain:

$$\begin{aligned}
&\mathbf{VaR}\ (\mathbf{nbS^2}) : \forall i \in [1, nbS],\ \forall j \in [1, nbS] \\
&\quad (y0a + y1a_i - y0b - y1b_i)s2_{ij}\ e^{q2(t2-t1)} - absy_i\ s2_{ij}\ tstock + z1_i\ e^{r(t2-t1)} \\
&\quad + \sum_{k \in O12} (absx12a_{ki}\ vopt12b_{2kij} - absx12b_{ki}\ vopt12a_{2kij}) \\
&\quad + \sum_{k \in O2_i} (x2a_{1ki}\ vopt2b_{2kij} - x2b_{1ki}\ vopt2a_{2kij}) + (\lambda - \theta)\ B\ \alpha_{ij} \geq \lambda B \qquad (7.4) \\
&\mathbf{VaR\ sum}\ (\mathbf{1}) : \\
&\quad \sum_{i=1}^{nbS} \sum_{j=1}^{nbS} Pr_{ij}\alpha_{ij} \leq 1 - u
\end{aligned}$$

7.3.7 The objective function

The goal is to maximize the expected value of the portfolio at $t2$; i.e. the weighted sum of the portfolio values for all scenarios at $t2$, where the weights are the probabilities of the scenarios.

$$\begin{aligned}
&\mathbf{objective} : \\
&\max \sum_{i}^{nbS} \sum_{j}^{nbS} Pr_{ij}(e^{q2(t2-t1)}(y0a - y0b + y1a_i - y1b_i)s2_{ij} - absy_i\ s2_{ij}\ tstock + e^{r(t2-t1)} z1_i \\
&\quad + \sum_{k \in O12} (x12a_{0k} + x12a_{1ki} - x12b_{0k} - x12b_{1ki}) popt12_{2kij} \\
&\quad - \sum_{k \in O12} (absx12a_{ki}\ cost12b_{2kij} + absx12b_{ki}\ cost12a_{2kij}) \\
&\quad + \sum_{k \in O2_i} (x2a_{ki}\ vopt2b_{2kij} - x2b_{ki}\ vopt2a_{2kij}))
\end{aligned} \qquad (7.5)$$

There is slight variation in the formulation used to compute the portfolio value in each scenario by contrast to the one used for the guarantee constraint. The parameters $vopt12$, i.e.

the values of options in $O12$, cannot be used in the objective function, and the parameters $popt12$ and $cost12$, that define $vopt12$, must be used instead. Indeed, as stated in Section 7.3.5, in order to ensure that the variables $absx12b_{ki}$ represent the final short positions of options in $O12$, the optimization process must minimize their values. This is only the case when the coefficients of $absx12b_{ki}$ are negative in the objective function, which is not the case when we use the fomulation with $vopt12$. At the converse, when we split this last parameter in its two components $popt12$ and $cost12$, the variables $absx12b_{ki}$ are only required in conjunction with the parameters $cost12$, to reduce the portfolio value by the corresponding costs. So, as required, this results in a negative coefficient for the variables $absx12b_{ki}$.

7.3.8 Mathematical programming model M1

Putting all the pieces together, we finally obtain the following mixed integer programming model. In order to formulate it in a standard form, we have inversed the sign of some of the constraints.

$$
\begin{aligned}
&\textbf{objective}:\\
&\max \sum_{i}^{nbS}\sum_{j}^{nbS} Pr_{ij}(e^{q2(t2-t1)}(y0a - y0b + y1a_i - y1b_i)s2_{ij} - absy_i\ s2_{ij}\ tstock + e^{r(t2-t1)}z1_i\\
&\quad + \sum_{k\in O12}(x12a_{0k} + x12a_{1ki} - x12b_{0k} - x12b_{1ki})popt12_{2kij}\\
&\quad - \sum_{k\in O12}(absx12a_{ki}\ cost12b_{2kij} + absx12b_{ki}\ cost12a_{2kij})\\
&\quad + \sum_{k\in O2_i}(x2a_{ki}\ vopt2b_{2kij} - x2b_{ki}\ vopt2a_{2kij}))\\
&\textbf{budget0 (1)}:\\
&\quad (y0a - y0b)s0 + (y0a + y0b)s0\ tstock + z0 + \sum_{k}^{nbO1}(x1a_k\ vopt1a_{0k} - x1b_k\ vopt1b_{0k})\\
&\quad + \sum_{k\in O12}(x12a_{0k}\ vopt12a_{0k} - x12b_{0k}\ vopt12b_{0k}) \le B
\end{aligned}
\tag{7.6}
$$

budget1 $(\mathbf{nbS})$ $:\forall i \in [1, nbS]$

$$(y1a_i - y1b_i)s1_i + (y1a_i + y1b_i)s1_i\ tstock + z1 + \sum_{k\in O2_i} (x2a_{1ki}\ vopt2a_{1ki} - x2b_{1ki}\ vopt2b_{1ki}) + \sum_{k\in O12} (x12a_{1ki}\ vopt12a_{1ki} - x12b_{1ki}\ vopt12b_{1ki}) \leq (y0a - y0b)s1_i(e^{q1(t1-t0)} - 1) + z0e^{r(t1-t0)} + \sum_{k\in O1} (x1a_k\ vopt1b_{1ki} - x1b_k\ vopt1a_{1ki})$$

guarantee $(\mathbf{nbS^2})$ $:\forall i \in [1, nbS],\ \forall j \in [1, nbS]$

$$-(y0a + y1a_i - y0b - y1b_i)s2_{ij}\ e^{q2(t2-t1)} + absy_i\ s2_{ij}\ tstock - z1_i\ e^{r(t2-t1)} - \sum_{k\in O12} (absx12a_{ki}\ vopt12b_{2kij} - absx12b_{ki}\ vopt12a_{2kij}) - \sum_{k\in O2_i} (x2a_{1ki}\ vopt2b_{2kij} - x2b_{1ki}\ vopt2a_{2kij}) \leq -\theta B$$

VaR $(\mathbf{nbS^2})$ $:\forall i \in [1, nbS],\ \forall j \in [1, nbS]$

$$-(y0a + y1a_i - y0b - y1b_i)s2_{ij}\ e^{q2(t2-t1)} + absy_i\ s2_{ij}\ tstock - z1_i\ e^{r(t2-t1)} - \sum_{k\in O12} (absx12a_{ki}\ vopt12b_{2kij} - absx12b_{ki}\ vopt12a_{2kij}) - \sum_{k\in O2_i} (x2a_{1ki}\ vopt2b_{2kij} - x2b_{1ki}\ vopt2a_{2kij}) + (\lambda - \theta)\ B\ \alpha_{ij} \leq -\lambda B$$

VaR sum $(\mathbf{1})$:

$$\sum_{i=1}^{nbS}\sum_{j=1}^{nbS} Pr_{ij}\alpha_{ij} \leq 1 - u$$

final index position $(\mathbf{nbS})$ $:\forall i \in [1, nbS]$:

$$y0a + y1a_i - y0b - y1b_i \leq absy_i$$

$$-(y0a + y1a_i - y0b - y1b_i) \leq absy_i$$

final option position $(\mathbf{nbS.nbO12})$ $:\forall i \in [1, nbS], \forall k \in O12$:

$$x12a_{0k} + x12a_{1ki} - x12b_{0k} - x12b_{1ki} \leq absx12a_{ki}$$

$$-(x12a_{0k} + x12a_{1ki} - x12b_{0k} - x12b_{1ki}) \leq absx12b_{ki}$$

$y0a, y0b \in \mathbb{R}_+$

$z0 \in \mathbb{R}$

$x1a_k, x1b_k \in \mathbb{R}_+\ \forall k \in O1$

$x12a_{0k}, x12b_{0k} \in \mathbb{R}_+\ \forall k \in O12$

$y1a_i, y1b_i, absy_i \in \mathbb{R}_+\ \forall i \in [1, nbS]$

$z1_i \in \mathbb{R}\ \forall i \in [1, nbS]$

$x12a_{1ki}, x12b_{1ki}, abs12a_{ki}, abs12b_{ki} \in \mathbb{R}_+\ \forall i \in [1, nbS], \forall k \in O12$

$x2a_{ki}, x2b_{ki} \in \mathbb{R}_+\ \forall i \in [1, nbS], \forall k \in O2$

$\alpha_{ij} \in \{0, 1\}\ \forall i, j \in [1, nbS]$

7.3.9 Mathematical programming model M2

A second mathematical programming model M2 is considered here. It only is a slight variation of model M1. In order to obtain M2, we simply replace the two VaR constraints (7.4) in M1 by the following ones:

$$
\begin{aligned}
&\textbf{VaR ij } (\mathbf{nbS^2}) : \forall s2_{ij} \ (\forall i \in [1, nbS], \ \forall j \in [1, nbS]) \\
&\quad - (y0a - y0b + y1a_i - y1b_i) s2_{ij} e^{q2(t2-t1)} + absy_i tstock - z1_i e^{r(t2-t1)} \\
&\quad - \sum_{k}^{nbO12} (absx12a_{ki} \ vopt12b_{2kij} + absx12b_{ki} \ vopt12a_{2kij}) \\
&\quad - \sum_{k}^{nbO2_i} (x2a_{1ki} vopt2b_{2kij} - x2b_{1ki} vopt2a_{2kij}) - M\beta_{ij} \leq -\lambda B \\
&\textbf{VaR i } (\mathbf{nbS}) : \forall i \in [1, nbS] \\
&\quad \sum_{j}^{nbS} Pr_{ij} \beta_{ij} \leq \gamma_i \\
&\textbf{VaR sum } (\mathbf{1}) : \\
&\quad \sum_{i}^{nbS} \gamma_i \leq 1 - u
\end{aligned}
\tag{7.7}
$$

$$
\begin{aligned}
&\beta_{ij} \in \mathbb{B} \ \forall i \in [1, nbS], \forall j \in [1, nbS] \\
&\gamma_i \in \mathbb{R}+ \ \ \forall i \in [1, nbS]
\end{aligned}
$$

This model is obtained by splitting into two parts the sum of binary variables appearing in (7.4). Thus, (7.7) is clearly equivalent to (7.4). As we have added a number of continuous variables, this new model may appear (uselessly) more complex. However, this new formulation splits in two distinct parts the optimization of the VaR variables: one part at the whole model level (variables γ_i) and another part at the level of the second period only (variables β_{ij}). This model will be useful to develop an optimization method based on Dybvig's theorem.

7.3.10 The guarantee constraint: improved approach

In model M1, we have formulated the guarantee constraint by checking that it is satisfied for each scenario. But, as discussed in Section 7.2.6, we have no proof that the guarantee remains satisfied for other states of the world.

If no option expires after $t2$, we have also seen that an exact formulation is obtained by only imposing two simplified constraints. First, the guaranteed lower bound must be satisfied when the index price is equal to zero or to one of the strike prices. Second, the portfolio value must be an increasing function of the index price for prices larger than the largest strike price; equivalently, the first derivative of the portfolio value with respect to the index price must be nonnegative in this interval. Some small adaptations are required to handle the costs of transaction.

If we simplify the notations by considering only one period, no transaction cost and no spread for the index and the options, the portfolio value is given by one expression of the form:

$$V(S) = yS + ze^{rT} + \sum_{i=1}^{nbOpt} x_i popt_i$$

where:

$$\begin{cases} popt_i = \max(0, S - K_i) & \text{for a call with strike price } K_i \\ popt_i = \max(0, K_i - S) & \text{for a put with strike price } K_i \end{cases} \tag{7.8}$$

The derivative is easy to compute as all the calls are in-the-money and all the puts are out-of-the-money when the index value is larger than the largest strike price. The guarantee constraint can be modelled as:

$$\begin{cases} V(0) \geq \theta B \\ V(K_i) \geq \theta B \quad i = 1, .., nbStrike \\ y + \sum_{j=1}^{nbCall} x_j \geq 0 \end{cases} \tag{7.9}$$

where *nbStrike* is the number of different option strike prices, which is typically smaller than the number of options when puts and calls are simultaneously considered.

Let us now see what happens when we introduce transaction costs in the model. There is no special difficulty when the costs of transaction are computed as a percentage of the amount invested: this just adds a term in the derivative. But if minimum and maximum levels are introduced, then the cost of transaction becomes a piecewise linear function. This is illustrated in Figure 7.4 for a long position in a call option.

Ideally, to ensure a strict constraint, we should consider the additional breakpoints for each strike price and for each position of the option. Note that the cost of transaction is removed from the payoff of an option for long positions as for short positions. This implies that extra breakpoints appear for each of the four cases: short and long positions in calls and puts. Considering all these breakpoints would enlarge considerably the size of the problem, and one can wonder if this added complexity is useful. Indeed, differences will appear only

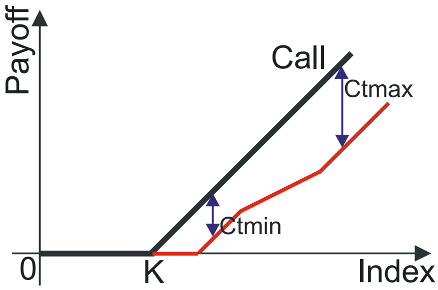


Figure 7.4: Minimal and maximal transaction costs

when the options are in-the-money, and for transaction costs that are typically small with respect to the payoff of the portfolio.

In view of these comments, we propose two intermediate, simpler approaches. Either we consider that the bounds on transaction costs have little effect in-between the strike prices, and we allow the guarantee constraint to be slightly violated. In this case, the transaction costs are always computed as a percentage of the amount invested. Or we strengthen the constraint by considering that, when a transaction cost is applied, it is always equal to the constant upper bound. By this trick, the option payoff is again linear between the strike prices. This also implies that the transaction cost is never underestimated, but rather, is, for each option, slightly overestimated for index values close to the corresponding strike price. Once again, as the final value of the portfolio is driven essentially by the payoff of the securities and not by the transaction costs, either method leads to similar quite realistic results. The first approach is no longer considered in the sequel.

So, if all the options expire at the horizon investment, using the second approach, the guarantee constraint of the model M1 (7.3.5) can be replaced by the following strong ones:

$$
\begin{aligned}
&\textbf{guarantee low } (\mathbf{nbS}.\textstyle\sum \mathbf{nbStrike_i} + \mathbf{nbS})) : \forall i \in [1, nbS],\ \forall j \in [0, nbStrike_i] \\
&\quad - (y0a - y0b + y1a_i - y1b_i) K_{ij}\ e^{q2(t2-t1)} + absy_i\ K_{ij}\ tstock - z1_i\ e^{r(t2-t1)} \\
&\quad - \sum_{k \in O12} ((absx12a_{ki} - absx12b_{ki})\ popt12K_{2kij} - (absx12a_{ki} + absx12b_{ki}) costG_{kij}) \\
&\quad - \sum_{k \in O2_i} ((x2a_{1ki} - x2b_{1ki}) popt2K_{2kij} - (x2a_{1ki} + x2b_{1ki}) costG_{kij}) \leq -\theta B \\
&\textbf{guarantee up } (\mathbf{nbS}) : \forall i \in [1, nbS] \\
&\quad - (y0a - y0b + y1a_i - y1b_i) e^{q2(t2-t1)} - \sum_{k}^{nbO2c_i} (x2a_{1ki} - x2b_{1ki}) \\
&\quad - \sum_{k \in O12c} (x12a_{0k} - x12b_{0k} + x12a_{1ki} - x12b_{1ki}) \leq 0.0
\end{aligned}
\tag{7.10}
$$

where:

$nbStrike_i$:	the number of different strike prices in scenario i.
K_{ij}:	option strike price, $\forall j \in [1, nbStrike_i]$.
K_0:	virtual strike price at zero.
$popt12K_{tkij}$:	as before but for the strike price K_{ij} instead of the index price.
$popt2K_{kij}$:	as before but for the strike price K_{ij} instead of the index price.
$costG_{kij}$:	equals $tmax$ if option k is in-the-money when the index value is K_{ij}. and zero otherwise.
$O12c$:	set of call options covering both periods.
$O2c_i$:	set of call options covering only the second period.

Chapter 8

Handling Value-at-Risk constraints

8.1 Introduction

In this chapter, we explore the possibility to detect for which scenarios the VaR lower bound will be satisfied in the optimal portfolio without performing the optimization of the problem as described in Chapter 7. Therefore, if we know a priori these scenarios, we do no longer need the binary variables α in model M1 to detect them, and the problem becomes easy to solve.

Section 8.2 analyzes the structure of VaR portfolio. First numerical experiments let expect that the optimal portfolios subject to VaR constraints have typical payoff patterns. Understanding the structure of the optimal portfolio could help to define improved optimization approaches.

In Section 8.3, we explain how we can simplify the optimization of model M1, if we know a priori how are distributed the optimal portfolio payoffs over the scenarios.

In Sections 8.4-8.5, we consider two approaches to detect a priori how are distributed these optimal portfolio payoffs. Both approaches are based on financial concepts. The first one, in Section 8.4, analyzes common trading strategies involving options. The second one, in Section 8.5, is based on a theorem due to Dybvig [22, 23].

8.2 Structure of the portfolio

8.2.1 Introduction

The investor wants to maximize the expected portfolio return without violating the guarantee constraint. Risk is measured here by a Value At Risk constraint. Either it is satisfied or not.

We do not try to minimize a risk measure like the variance in Markowitz's model.

This has important consequences on the structure of feasible portfolios and the optimal one. It is useful to well understand the portfolio structure before starting to study improvements of the maximization process.

8.2.2 Theoretical structure

Short position in the index

With a short position in the index, the payoff is negative and the loss becomes huge as the index price increases. To be sure to satisfy the guarantee constraint we must cover the position by a long position in a call (with a strike price low enough to satisfy the guarantee or 100% VaR constraint).

However, a short position in the index and a long position in a call has the same payoff as a put with the same strike price. By the put-call parity equations, if there are no transaction costs, the synthetic put has the same price as the real put. When there are transaction costs, as in our case, the synthetic put costs more than the real put. So if there exists a put with the same strike price as the call, we should never take a short position in the index, but rather take a long position in a put.

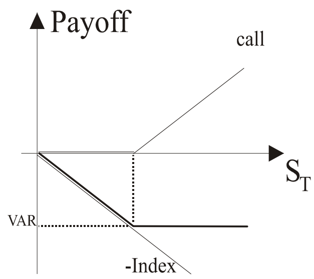


Figure 8.1: Short index

Long position in the index

With a long position in the index, the worst case happens when the index value becomes null. The loss is total but is limited. Two strategies are possible to satisfy the guarantee constraint, even for this worst case. Either we take a long position in a put to limit the loss for small index prices or we make a risk free investment to be sure to compensate the loss in the future. The first solution is not optimal for the same reason as in the case of a short position: a long position in the index and a long position in a put corresponds to a long position in a call. Because of transaction costs, it is better to purchase a call than purchase the index. The second solution is acceptable. However, we should remember that a long index is equivalent to a long call with a null strike price. Before choosing this possibility, we should consider the solution with no index and with calls. The leverage effect of the options could lead to a better alternative.

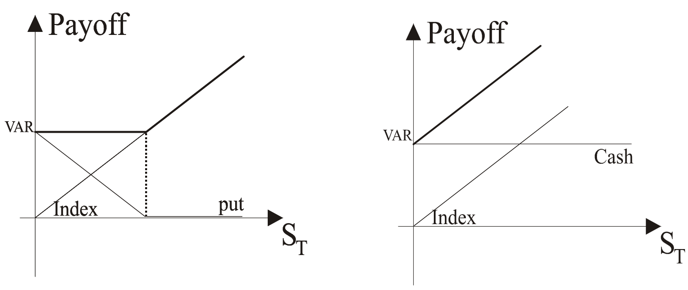


Figure 8.2: Long index

No index and long position in calls

If the index value in the future becomes smaller than the strike price, the call is valueless. A risk-free investment is the first solution to satisfy the guarantee constraint. The amount invested should at least correspond to the current value of the guarantee bound. The second possibility is to take a long position in a put with a large enough strike price. Note that this structure doesn't reject the possibility of short positions in calls at the same time.

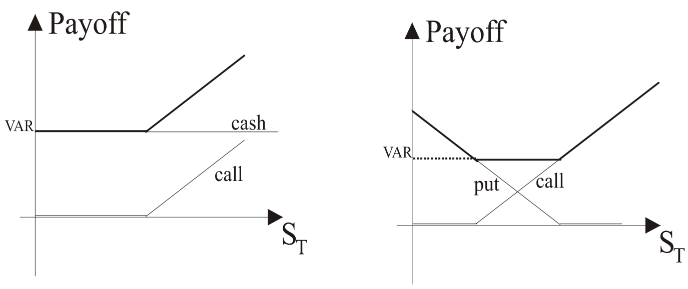


Figure 8.3: Long call

No index and long position in puts

If the index value in the future becomes larger than the strike price, the put is valueless. A risk-free investment is again the first possibility to satisfy the guarantee constraint. The amount invested should at least correspond at the current value of the guarantee bound. Note that in this case the expected portfolio return is bounded. A long position in a call with a small enough strike price is the alternative. Note that this structure doesn't reject the possibility of short positions in puts at the same time.

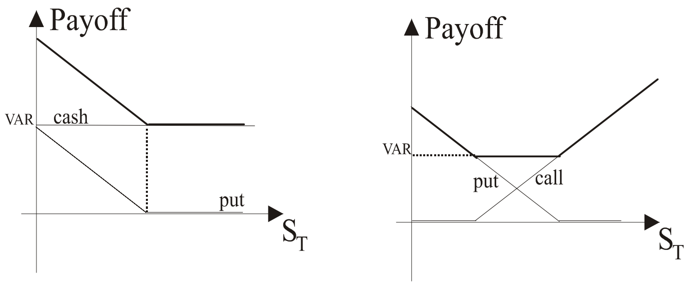


Figure 8.4: Long put

No index and short position in calls

This position leads to two problems: total loss if the index price becomes lower than the exercise price and huge loss (to infinity) if the index prices highly increase. We need to take a long position in the index or in a call to avoid the second problem. This means that a short position in calls never happens alone and is only a subcase of a long position on the index or in the calls.

No index and short position in puts

The risk of total loss can only be covered by a long position in a put or by a large risk-free investment.

Conclusions

A short position in the index can never happen if all the described options are available. A long position in the index could theoretically happen, but only in very special situations. The classical optimal value consists of long positions on options, with or without short positions, and a cash investment to satisfy the guarantee constraint.

The objective is to maximize the expected return under minimal guarantees. The first point is that the constraints can ever be satisfied if the portfolio consists of risk-free investments and long positions in options. Secondly, the highest return can only be achieved with long positions on options.

8.2.3 Empirical structure

The theoretical structure presented just above is based on the continuous range of index values. The whole distribution is considered. In our scenario model, we model the future by a discrete set of scenarios (prices). This means that the smallest value of the index price is known (and is generally larger than zero) as well as the largest value (large value, but not infinity).

This has some consequences on the previous conclusions in the cases of a short position on the index or in calls. It is now possible to cover a short position in the index by a large risk free investment because the maximal loss is not anymore the infinity but a large value. That is still unrealistic. The same is true for a short position in calls.

There are no fundamental changes in the other cases. Theoretically, the worst is to lose everything. Numerically, the worst is to lose nearly everything (but less). The strategies stay the same.

8.3 Optimal VaR allocation

8.3.1 Introduction

As already noticed in Dert and Oldenkamp's paper [19] under specific hypotheses, and also observed more generally in our preliminary computational tests, the limitation of the risk by a VaR constraint often results in typical payoff patterns for the optimal portfolio. For instance, when performing experiments with the S&P500 index, which is characterized by a large expected return, the optimal portfolio value typically coincides with the guarantee level when the future index value is low (due to the presence of a risk free investment in the portfolio), then increases to the VaR level (due to the presence of a few options), and eventually increases linearly for the largest index values. The content of the optimal portfolio is a function of the different scenarios, and differs accordingly for each of the portfolios adjusted at the end of the first period, but the general payoff pattern described here remains valid.

This pattern appears only because the expected return of the S&P500 is large and its volatility is reasonably small. By contrast, if a decrease of the index value is expected in the future (for example if we change the sign of the expected return before computing the probability density function), then the payoff pattern is reversed.

These observations suggest that, if we were able to understand *a priori* the structure of the optimal solution, then the corresponding solution could be simple enough to be computed directly without resorting to complex optimization procedures, or at least, the knowledge of its structure could be used to improve the optimization process. In particular, if we had prior knowledge about the distribution of the portfolio payoffs in the tree of scenarios, then we could decide immediately to which leaves of the tree we should apply the VaR lower bound, and this would result in a model without binary variables which could be solved easily and quickly by the simplex algorithm. Let us now develop more precisely this idea, which will play a central role in the remainder of the chapter.

8.3.2 Relaxation of the VaR constraint

Because of the guarantee constraint, every optimal solution of the portfolio optimization model M1 (see Section 7.3.8) is such that, in all scenarios, the final value of the portfolio is at least equal to θB. Moreover, because of the VaR constraint, the final value of the portfolio is at least equal to λB in a fraction u of all the scenarios (assuming that all scenarios are equiprobable). Note that, in every practical application, $\lambda \geq \theta$.

Consider now what would happen *if we knew a priori* the set of scenarios where the VaR requirement is satisfied, i.e. where the portfolio value is at least λB. For an arbitrary portfolio, and for every scenario $j = 1, 2, \ldots, nbS^2$, let us denote by $V(j)$ the final value of the portfolio under scenario j. Moreover, let $\mathcal{S}$ be a set of scenarios of size $|\mathcal{S}| \geq u \times nbS^2$, for which we know beforehand (i.e., before solving the optimization model) that they should satisfy the VaR requirement. Under this assumption, we could safely simplify model M1 as follows: we would impose the guarantee constraints

$$V(j) \geq \theta B \tag{8.1}$$

only for the scenarios *not in* $\mathcal{S}$ (compare with constraints 7.3 in model M1), we would impose the VaR constraints

$$V(j) \geq \lambda B \tag{8.2}$$

only for the scenarios *in* $\mathcal{S}$ (compare with constraints 7.4 in model M1), and we would remove all binary variables α_{ij} from model M1 (i.e., set them to 0). Let us denote by M1($\mathcal{S}$) this simplified model. We claim that the optimal value of M1 is equal to the optimal value of M1($\mathcal{S}$). Indeed, on one hand, the previous optimal solution of M1 satisfies all the constraints in M1($\mathcal{S}$), by construction. On the other hand, any feasible solution of M1($\mathcal{S}$) is also feasible for M1, since it necessarily satisfies the VaR requirement for all scenarios in $\mathcal{S}$ and it satisfies the guarantee constraint for all scenarios (due to the inequality $\lambda \geq \theta$). This establishes the claim.

So, M1 and M1($\mathcal{S}$) are equivalent. Note, however, that model M1($\mathcal{S}$) is more attractive than model M1, because it contains fewer contraints and, even more importantly, because it does not involve any binary variables. Therefore, we expect M1($\mathcal{S}$) to be much easier to solve than M1 (as a rule, linear programming problems are "easy", while 0-1 mixed integer programming problems may turn out to be extremely hard to solve optimally).

Of course, in general, we do not know how to set up model M1($\mathcal{S}$) a priori, since *we do not know beforehand* the set $\mathcal{S}$ of scenarios where the VaR requirement will be eventually satisfied (in case the model has several alternative optimal solutions, the set $\mathcal{S}$ may even vary from one solution to the next). Conceptually, however, the idea remains appealing, and can be cast in a broader framework: if we have prior information about the structure of the optimal portfolio, then this information could prove useful in simplifying the formulation and the solution of the mathematical programming model. Alternatively, the same idea can also be used to develop heuristic solution approaches for the solution of model M1. To see this, consider now *any set* $\mathcal{S}$ of scenarios such that $|\mathcal{S}| \geq u \times nbS^2$, and consider again the linear programming model M1($\mathcal{S}$) associated to $\mathcal{S}$ as explained above. Then, of course, it is no longer true that M1 and M1($\mathcal{S}$) are equivalent. However, every feasible solution of M1($\mathcal{S}$) remains a feasible solution of M1, and therefore, the optimal solution of M1($\mathcal{S}$) provides a heuristic solution of M1. The quality of this heuristic solution can be expected to be high if our "guess" concerning $\mathcal{S}$ is reasonably good.

In the following sections, we propose to pursue systematically these ideas by investigating the structure of optimal portfolios in connection with popular investment strategies used in practice (see Section 8.4), and with related results obtained by Dybvig in a theoretical framework (see Section 8.5).

8.4 Optimal VaR allocation vs. investment strategies

8.4.1 Trading strategies involving options

In this section, we examine to what extent the structure of an optimal portfolio is consistent with the structures which would emerge from the use of common, intuitive investment strategies involving options, as considered in the literature (see for example [32]). We concentrate on four such strategies which appear to be suitable in different market environments, characterized by different mean-variance combinations of the index distribution: bullish (in case of large positive expected returns), bearish (in case of large negative expected returns), butterfly (in case of high volatility of the returns), and stability (in case of low expected returns and low volatility).

Bullish portfolios

Ideally, the investor would like to constitute a portfolio which guarantees a very high return in every possible state of the world. But unfortunately, there is no free lunch on the financial marketplace, and every portfolio is usually characterized by a distribution of returns ranging from (relatively) low to (relatively) high, depending on the scenario that unfolds over the investment horizon. In practice, therefore, investors, try to match this distribution of returns to their expectations regarding the future.

To express this notion somewhat more formally, consider any solution of our portfolio optimization model. As before, let us denote by $V(j)$ the final value of the portfolio in scenario j, and let $s(j)$ be the price of the index in scenario j. We call a portfolio *bullish* if $V(j)$ is a non decreasing function of the value of the index in scenario j, i.e. if

$$s(j_1) \leq s(j_2) \Longrightarrow V(j_1) \leq V(j_2)$$

for all pairs of scenarios j_1, j_2.

Observe that, if an investor expects that the index price will be high in the future, then he is likely to construct an agressive portfolio which yields its highest returns when the index price is high and which, conversely, achieves the lowest returns when the index price decreases instead of increasing as expected. More precisely, we qualify an investment strategy as bullish if it produces a bullish portfolio. Such strategies are commonplace in practice. Implicitly, Dert and Oldenkamp [20] only consider bullish strategies in his work.

If we have reasons to believe that the optimal solution of the portfolio model M1 is bullish, then M1 can be simplified along the lines described in Section 8.3. Indeed, in this case, the VaR requirement only needs to be applied to the fraction u of leaves with the highest index values, since these correspond to the scenarios where the portfolio value will be highest and hence, where it is easiest to satisfy the VaR requirement (see constraint (8.2)). Also, as explained above, the guarantee constraint only needs to be applied to the remaining fraction $(1 - u)$ of leaves associated with the lowest values of the index.

Note that the index price at which the portfolio “jumps” from satisfying the guarantee constraint to satisfying the VaR constraint can be determined (if the distribution is well represented by the tree) by inverting the index distribution function. Indeed, it is simply the value for which the distribution function equals $(1 - u)$.

Bearish portfolios

Conversely, when an investor expects that the index price will be low in the future, then his natural strategy is to construct a protective portfolio to take advantages of low index prices. We say that a portfolio is *bearish* if its value $V(j)$ is a nonincreasing function of the price of the index in scenario j, i.e. if

$$s(j_1) \leq s(j_2) \Longrightarrow V(j_1) \geq V(j_2),$$

and we say that a strategy is bearish if it produces bearish portfolios.

Again, if we know beforehand that the optimal portfolio is bearish, then we can replace M1 by the simpler model M1($\mathcal{S}$), where $\mathcal{S}$ is the smallest set of scenarios of size at least $u \times nbS^2$ containing the lowest index prices.

Butterfly portfolio

We say that a portfolio is a *butterfly* when $V(j)$ is a nondecreasing (respectively nonincreasing) function of the index for values of the index lower (respectively larger) than a certain threshold, i.e. if there exists a price s^* such that

$$s(j_1) \leq s(j_2) \leq s^* \Longrightarrow V(j_1) \leq V(j_2)$$

and

$$s^* \leq s(j_1) \leq s(j_2) \Longrightarrow V(j_1) \geq V(j_2).$$

The investor adopts a butterfly strategy (that is, invests in a butterfly portfolio) when he believes that the future value of the index price should be close to the threshold value s^*, and is unlikely to assume extreme values (either very high or very low).

When this is the case, we can simplify model M1 by applying the VaR constraints only to a subset of scenarios around the threshold value, and by applying the guarantee constraint only for extreme index prices. Note, however, that the correct choice of the set $\mathcal{S}$ may depend on several parameters like the risk free rate, the expected return or the volatility of the prices. In particular, there is no reason to choose $\mathcal{S}$ symmetrically around s^*, nor to define it by excluding the same number of scenarios with low and with high index value. Therefore, even under the assumption that the optimal portfolio is a butterfly, selecting $\mathcal{S}$ and computing the optimal portfolio remains more complex than in the bullish or bearish cases.

Volatility

A *volatility* portfolio has the opposite shape of a butterfly: first decreasing, then increasing as a function of the index price (in financial parlance, this is the shape of inverse butterflies, straddles, strangles, strips and straps; see [32] for details). Buying this type of portfolio is justified when the investor thinks that the future value of the index price will deviate significantly from its expected value. As far as model M1 goes, this situation also is opposite to the previous one: here, the VaR requirement should be imposed on extreme scenarios, and the guarantee constraint on central ones. Again, the problem remains of correctly determining the corresponding set $\mathcal{S}$.

8.4.2 Selection of a strategy

If we can determine when to apply one of the above four strategies, and how to define the set $\mathcal{S}$ in case of the butterfly or of the volatility strategy, then we can transform the mixed integer problem M1 into a linear programming model which can be solved efficiently by the simplex algorithm.

If we only know that the optimal portfolio is a butterfly or has the volatility structure, then the problem of determining the correct choice of the set $\mathcal{S}$ remains. Of course, we could try all possible choices for $\mathcal{S}$. Note that choosing $\mathcal{S}$ amounts to choosing an interval of index prices, and therefore the number of possible choices is of the order of the number of final states of the world. Since this number is usually quite large, this enumerative approach is not feasible in practice. We will see in Chapter 9 how a compact and more efficient formulation of M1 can be derived in this case.

Conversely, if we do not know how to select the optimal strategy, but we know how to pick the set $\mathcal{S}$, then we could at worst apply the simplex algorithm once for each of the four strategies and keep the best solution found in the process.

In an attempt to choose, at least heuristically, between the four available strategies, we have developped two distinct approaches. The first (and most powerful one) is based on Dybvig's theorem and will be presented in Section 8.5. We now proceed to describe another, more primitive approach, which relies on partitioning the space of index returns into four zones, which we put in correspondence with the four investment strategies. Denote by μ and σ the expected value and the standard deviation of the index returns, and denote by r the risk free return. Then, our partitioning criterion goes like this:

(a) if μ is large with respect to r and σ is small with respect to $|\mu - r|$, construct a bullish portfolio;

(b) if μ is small with respect to r and σ is small with respect to $|\mu - r|$, construct a bearish portfolio;

(c) if μ is close to r and σ is small with respect to $|\mu - r|$, construct a butterfly portfolio;

(d) if σ is large with respect to $|\mu - r|$, construct a volatility portfolio.

The criterion can be further operationalized, for example, by specifying

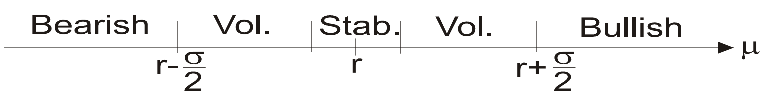
(a) if $\mu > r + \sigma/2$ then bullish;

(b) else if $\mu < r - \sigma/2$ then bearish;

(c) else if $\sigma < |\mu - r|/4$ then butterfly;

(d) in all other cases: volatility (case d)

(see Figure 8.4.2).



Note that these criteria are purely heuristic, as they simply translate the intuitive idea that the expected "level" of returns forms a reasonable basis for the choice an investment strategy. But in general, we do not know how to rigorously define a partition of the (μ, σ)-space in such a way that the resulting choice of strategy would necessarily be optimal (and actually, even the existence of a partition with this property is very doubtful).

The numerical experiments presented in Chapter 10 consider only the S&P500 index for a given period. For this dataset, the strategy is clearly bullish (large expected return). We have not test the approach of this section with other indices.

8.4.3 One-period strategy vs. two-period strategy

All the trading strategies presented above define a relation between the portfolios and the index values. Obviously, these approaches can be directly applied to the final leaves of the two-period tree of scenarios. Moreover, they also can be applied locally for each one-period subtree of the second period. Indeed, the investor can adjust the portfolio at $t1$, and so constitute a specific portfolio for each subtree.

In fact, as discussed more in detail in Section 9.3.2 of Chapter 9, applying locally the strategies leads to a more general approach, which includes the strategies globally applied at the final leaves. The drawback is an increase of the complexity.

8.5 Optimal VaR allocation vs. Dybvig's theorem

8.5.1 Introduction

Let us return once again to the ideas presented in Section 8.3. For simplicity, we will consider that the scenarios are equiprobable (the generalization would be immediate). So, our goal is to impose the VaR constraint on exactly $(u \times nbS^2)$ scenarios, in such a way as to minimize the effect of this simplification on model M1.

It follows from the discussion in Section 8.3 that, if we could sort the nbS^2 portfolio values of the optimal portfolio by increasing order, then we could simply apply the VaR constraint on the set of scenarios, say $\mathcal{S}$, corresponding to the highest portfolio values, and this would yield a model M1($\mathcal{S}$) perfectly equivalent to M1. Note that in order to implement this approach, we only need to be able to order the scenarios in agreement with the optimal portfolio values, but *we do not need to know the portfolio values* themselves. In some cases, a theorem due to Dybvig will allow us to perform this sorting without solving the optimization problem. We will discuss Dybvig's theorem in the remainder of this section.

8.5.2 Dybvig's theorem

Dybvig [22, 23] considers the pricing problem for a consumption bundle, which we can interpret as a tree of scenarios together with the value of a good (or asset, or portfolio) attached at each leaf of the tree. More precisely, he states the following hypotheses:

H1. Agents' preferences depend only on the probability distribution of consumption of a single good.

H2. Agents prefer more to less.

H3. The market is arbitrage-free, perfect (no taxes, no transaction costs, no information asymmetries) and is complete over finitely many equally probable terminal states or over an atomless continuum of states. Such a market allows short sales without penalty.

Let us introduce a new definition: if we denote the state price of a scenario j by $\psi(j)$, and its probability by $\Pr(j)$, then the *Arrow-Debreu density*, or *state-price density* of scenario j

is by definition the quantity

$$\varphi(j) = \psi(j)/Pr(j).$$

Under the above hypotheses, Dybvig [22, 23] proved that a consumption bundle is efficient if and only if it provides at least as much consumption in scenarios with lower Arrow-Debreu densities than in scenarios with higher Arrow-Debreu densities.

In our usual portfolio terminology, Dybvig's result can be rephrased as follows: if the portfolio values for all scenarios are not in reverse order of the Arrow-Debreu densities, then the investor pays too much for this portfolio. Indeed, it is always possible to construct a cheaper portfolio by switching the final scenario payoffs in such a way as to obtain the inverse relation with the Arrow-Debreu densities, without changing the probabilities of final payoffs. Conversely, if the investor is able to save money by buying this new portfolio, then he can subsequently increase the final expected value of his portfolio by investing the amount that he saved. Thus, the optimal portfolio is such that there is an inverse relation between portfolio values and state-price densities.

Of course, the property established by Dybvig is exactly what we need for our purpose, since we know the probability of each scenario (by construction of our model) and we also know the corresponding state-prices (we have shown how to compute the vector ψ in Chapter 6, when dealing with the option pricing problem). So, we can sort the Arrow-Debreu densities by increasing order, invert this sequence so as to rank the scenarios by increasing order of optimal portfolio values (without knowing these values), and deduce from this the adequate reformulation of model M1 into a linear programming model. One application of the simplex method is then enough to obtain the optimal portfolio.

In a sense, Dybvig's theorem is valid in a much more general settings than required for our specific portfolio model: if its hypotheses are satisfied, then it can be applied regardless of the presence of very special constraints like budget, guarantee and VaR constraints. Indeed, Dybvig's proof is based on permuting the final payoffs, i.e. the scenarios, and this has no consequence on the guarantee constraint (model (7.3)) and VaR constraint. The guarantee constraint defined in (7.10) is more complex and will be presented in Section 8.5.3. Also, since the proof works by reducing the initial cost of the portfolio, it certainly maintains the budget constraint.

Unfortunately, some of Dybvig's hypotheses are not necessarily satisfied in our framework. We now turn to a discussion of these hypotheses.

8.5.3 Validity of Dybvig's hypotheses

Dybvig's hypotheses H1 and H2 are clearly valid in our portfolio optmization model, but the hypothesis H3 is more troublesome. Moreover there is a problem with the formulation (7.10) of the guarantee constraint. Let us examine their ingredients in detail.

Hypothesis H3: Equiprobable states

The part of hypothesis H3 concerning "equally probable terminal states" does not raise any real difficulty. Indeed, in our model, we could sample the distribution of index returns by various methods, but we have decided, for simplicity and without loss of generality, to assign, in the explanations of this section, the same probability to each scenario. Moreover, Dybvig's theorem remains actually valid when the states have unequal probabilities, under the additional hypothesis that agents are risk averse; see Dybvig [23].

Hypothesis H3: Market completeness

In our model, the market is usually not complete. Remember that a market is complete if and only if every consumption process (portfolio values) is attainable on this market (see Dothan [21] and Chapter 4). Mathematically, the market is complete if and only if $rank(\Pi) = K$, where Π is the $N \times K$ (securities $\times$ scenarios) payoff matrix defined in Chapter 4, Section 4.4.2.

In order to obtain a good representation of the future, we need to choose a large value for K, namely a value which is typically much larger than the number of securities considered in the portfolio. Then, $rank(\Pi)$ is at most equal to the number of securities, and hence is smaller than K.

What does it imply? Dybvig's theorem is based on the observation that, everything else being kept constant, one can switch the final portfolio values of two scenarios because, due to the completeness of the market, it is always possible to reconstruct a new portfolio with this new distribution of final values. More formally, if we denote by $x \in I\!R^N$ the vector whose components represent the quantity of each security in an accessible portfolio, and by $V \in I\!R^K$ the vector whose components represent the final value of this portfolio in each scenario, then we have the relation:

$$\Pi^t x = V.$$

Under the assumption of market completeness, if we construct a new vector V' by permuting some elements of V, then there always exists a vector x' such that:

$$\Pi^t x' = V'.$$

This is obvious by definition of a complete market, but does not necessarily hold anymore when there are more scenarios than securities. In this case, there is no guarantee that the new portfolio is attainable.

When considering incomplete markets, another drawback is that the state-prices, and hence the state-price densities, are no longer uniquely determined. As a result, even just interpreting the statement of Dybvig's theorem becomes troublesome. Jouini and Kallal [36], for instance, have examined extensions of Dyvig's results to incomplete markets, and have proposed adequate generalizations of the concept of state prices. In our framework, however, this issue is slightly easier to resolve. Indeed, even though our *representation of the market* via scenario trees is incomplete, we may still assume that the *underlying market itself* is complete. This is in fact the assumption which justifies our computations of state prices, in Chapter 5, Section 5.3.1: we assume that the tree of scenarios provides a partial representation of the "real" complete market, where the state prices are uniquely defined – in the same sense as a sample from a statistical distribution gives a partial representation of the population.

Note that if we use a binomial tree, in which the investor could adjust the portfolio at each period, as we consider at least two independent securities (cash and index), the market is complete and Dybvig's theorem can be used (if the other hypotheses are satisfied).

Hypothesis H3: Frictions

The market that we consider in our portfolio model is not frictionless, since it involves transaction costs and bid-ask spreads. Due to these market frictions, there is no guarantee that we can still switch the porfolio values to construct a cheaper portfolio, and hence Dybvig's argument breaks down. Jouini and Kallal [36] (and other authors cited in their paper) have considered extensions of Dybvig's theorem for imperfect markets, but it is not clear at this point whether, and how, their findings could be integrated in our work.

Guarantee constraint

Dybvig's proof is based on permuting the final payoffs, i.e. the scenarios. In the first formulation (7.3) of the guarantee constraint, the portfolio value is only computed for the index values at the leaves of the tree. The sequence of scenarios does not matter, and therefore the guarantee constraint remains satisfied even for Dybvig's optimal portfolio obtained by permuting some leaves.

This is no longer true for the second formulation (7.10) of the guarantee constraint using the strike-prices method. This model is stronger than the previous one since the constraint is satisfied for any index values, rather than only the index values at the leaves. Therefore, the new portfolio constructed using Dybvig's theorem must also satisfy the guarantee constraint "between" the leaves and for larger index values than the largest one of the tree. However, Dybvig's approach considers only the portfolio values at the leaves, and therefore nothing can prevent us to obtain an inadequate portfolio; i.e. an infeasible solution.

8.5.4 Handling Dybvig's hypotheses

The previous section has established that Dybvig's theorem cannot be applied directly to our portfolio investment model. We are now going to describe a few possible approaches to resolve the difficulties that have been identified, and examine their pros and cons.

Hypothesis H3: Market completeness

The financial market in our model is typically incomplete. It is theoretically possible to overcome this difficulty by *completing the market*, that is by creating as many artificial options as there are scenarios in the model, with the objective to satisfy the condition that the rank of Π is equal to the total number of scenarios. For instance, we could create for each scenario an option (either a call or a put) with a strike price equal to the index value of the scenario.

This approach has obvious advantages: it allows to satisfy one of Dybvig's hypotheses and, as a side effect, it also simplifies the computation of the option prices (because the state-prices are now unique, we do not need anymore to solve the minimization pricing models described in Section 6.3).

On the other hand, the approach also has major inconvenients: it introduces options that the investor perhaps does not want to consider (although they can always be constructed over the market), and, much more importantly, it increases considerably the size of the model. To understand that the latter feature could translate into very serious numerical difficulties, let us pause a moment to contemplate the size of our optimization model.

Each time the investor can adjust his portfolio, i.e. at each of the nbS subtrees, the matrix Π has nbS columns and should possess at least the same number of rows to satisfy $rank(\Pi) = nbS$. To obtain a good representation of the future, we need to consider a large number of scenarios (at least $nbS = 30$ per period with a stratification process). With 30 subtrees of each 30 scenarios, we need to consider about 900 options to complete the market. This would imply the construction of an enormous amount of "synthetic" options and the solution of various questions linked with this operation (determination of transaction costs, pricing, ...).

Moreover, if we consider the simplex table at any one node of the branch and bound solution process, the number of elements to put in memory (in our simplest model) is approximately $8 \times nbS^4$. With $nbS = 30$, this gives at worst 6.5 millions elements or, in double precision, about 49MB of memory at each one node! For $nbS = 100$ scenarios, the memory required to input the constraint matrix, before even starting the optimization process, is about 6.1GB. Even with careful programming, problems of this size cannot be handled by traditional methods on ordinary computers!

From a practical point of view, and considering only this computer memory problem, we deem it very difficult, or even plainly impossible to complete the market in our model when the number of scenarios is large.

Hypothesis H3: Frictions

Even if we consider a complete market, we still face the problem caused by market frictions (transaction costs and spreads). These frictions exist on real markets and cannot be disregarded. It would be interesting, however, to know whether they alter significantly the optimal solutions. Jouini and Kallal [36] discuss, in general terms, the impact of frictions on market efficiency. They mention that market frictions usually modify, and typically shrink, the set of efficient investment strategies, shifting investors away from well-diversified strategies into low cost ones, and, in case of large frictions, into no trading at all. Hence, we can observe strategies that become inefficient in the presence of market frictions, as well as strategies that are rationalized by their presence.

More specifically, frictions translate into costs in the optimization model M1, and so decrease the budget available to buy or sell the securities. Consider the portfolio model M_D (with D for Dybvig) obtained by removing all transactions costs and spreads from model M1, denote by U_D its optimal value, and denote by X_D its optimal solution. Then, it is clear that U_D is an upper bound on the optimal value of M1. Moreover, X_D can be viewed as a feasible solution of M1 (if the guarantee constraint is expressed by (7.3)), although there is no guarantee that this solution is optimal for M1. If we compute the value of the objective function of M1 (taking all real costs into account) for the feasible solution X_D, then this value provides a lower bound on the optimal value of M1. Denote this lower bound by L_D.

Intuitively, the smaller the frictions, the closer M_D should be to the true model M1 and hence, the closer U_D and L_D should be to the optimal value of M1. It would be interesting to know how tight these bounds are in practice, and whether they could be useful to speed up a branch and bound process. In particular, we would like to know how they compare to the bound obtained by relaxing the integrality conditions $\alpha_{ij} \in \{0, 1\}$ in M1. Dybvig's solution is an integer solution, i.e. lower than the corresponding relaxation, but increased due to the supression of the frictions. Numerical results are given in Chapter 10.

In practice, we will not construct the pure model M_D, but rather use directly Dybvig's state-prices relation in model M1. The method is to define the set $\mathcal{S}$ such as to contain the scenarios with the lowest Arrow-Debreu densities, and then to apply the VaR constraint only to this set (as described in Section 8.3.2). This new model is presented in Section 9.3.3.

Guarantee constraint

The guarantee constraint expressed by (7.10) reduces the set of portfolio payoffs that can be considered by Dybvig's approaches. This is the same consequence as the problem of market incompleteness. However, we cannot simply use here a different or extended set of data to overcome the difficulty. In order to apply efficiently Dybvig's theorem, we must use the first formulation (7.3) of the guarantee constraint.

If we use the first formulation (7.3), then the number of scenarios nbS should be large enough to avoid the problems of "leaks" described in Section 7.2.6, and therefore the problem size is larger and the problem is harder to solve.

So, without relaxing the guarantee constraint by using (7.3) and a reasonable number of scenarios, there is no way to reduce the penalties due to the guarantee constraint. We have considered that it is more suitable to satisfy strictly the guarantee constraint than reducing the penalties. Therefore, the model (7.10) for the guarantee constraint is used in the sequel. In Section 10.4.4 of Chapter 10 however, the results obtained for the two models are compared.

Conclusion

In conclusion, there are several reasons why Dybvig's theorem cannot be used directly to find an optimal solution of model M1, as we may have hoped originally. If we want to obtain an exact optimal solution of M1, then we still have to rely on the mixed integer programming formulation of the model, and on a generic algorithm (like branch and bound) for the solution of this formulation. However, in Chapter 9, we will be able to propose heuristic approaches using Dybvig's theorem, in order to compute quickly lower bounds on the optimal solution of M1.

8.5.5 Index values and Dybvig's theorem

Introduction

It is commonly assumed, both in the financial literature and in investment practice, that the value of the optimal portfolio based on an index should be an increasing function of the value of the index. In our previous terminology, this translates into the assumption that an optimal portfolio is necessarily bullish. For instance, Dert and Oldenkamp [20] refer to Dybvig's theorem to derive that, for a continuous normal distribution of index returns, "payoff patterns that are not monotonically increasing in the index are suboptimal".

However, as we have already discussed in Section (8.4.2), bullish portfolios are not necessarily optimal under all possible circumstances, and a distinction between different cases could possibly be established on the basis of the index distribution. In this section, we would like to draw on Dybvig's theorem to analyze somewhat further the relation between index values and the final optimal portfolio values.

Binomial trees

First, we would like to prove that, if the distribution of returns is modelled by a one-period binomial tree and if the world is risk-averse, then the state-price densities are a nonincreasing function of the index values. Under the assumption of Dybvig's theorem, this immediately implies that the optimal portfolio value is an increasing function of the index values, i.e. that the optimal portfolio is bullish.

Proposition. Let S_{up} and S_{down} be the two possible future values of an index, and let φ_{up} and φ_{down} denote the corresponding state-price densities. Assume further that the expected return of the index is larger than the risk-free return. If $S_{up} > S_{down}$, then $\varphi_{up} < \varphi_{down}$. So, under these assumptions, the optimal portfolio is always bullish.

Proof.

Consider a market with only two possible investments: cash and index. We get the following relation between the Arrow-Debreu prices ψ_{up} and ψ_{down} and the future values of the index:

$$\begin{pmatrix} 1 \\ S \end{pmatrix} = \begin{pmatrix} 1+r & 1+r \\ S_{down} & S_{up} \end{pmatrix} \begin{pmatrix} \psi_{down} \\ \psi_{up} \end{pmatrix}. \tag{8.3}$$

The unique solution of this system is

$$\begin{cases} \psi_{down} = (S_{up} - S(1+r))/c \\ \psi_{up} = (-S_{down} + S(1+r))/c \\ c = (1+r)(S_{up} - S_{down}) \end{cases} \tag{8.4}$$

From $S_{up} > S_{down}$, we deduce $c > 0$.

Let now p_{up} and p_{down} denote the probabilities of the two possible states of the world. For an initial investment of S monetary units, the hypothesis that the expected return of the index exceeds the risk-free returns implies:

$$\begin{aligned} & E_{risky\ invest}(S) > E_{risk\ free\ invest}(S) \\ \Leftrightarrow\ & p_{up}S_{up} + p_{down}S_{down} > (p_{up} + p_{down})S(1+r) \\ \Leftrightarrow\ & p_{up}(S_{up} - S(1+r)) > p_{down}(-S_{down} + S(1+r)) \\ \Leftrightarrow\ & p_{up}(S_{up} - S(1+r))/c > p_{down}(-S_{down} + S(1+r))/c \quad (\text{since } \ c > 0) \\ \Leftrightarrow\ & p_{up}\psi_{down} > p_{down}\psi_{up} \\ \Leftrightarrow\ & \psi_{down}/p_{down} > \psi_{up}/p_{up} \\ \Leftrightarrow\ & \varphi_{down} > \varphi_{up}. \end{aligned} \tag{8.5}$$

Note that we can add other securities (such as options) in the model without modifying the relation between the state-prices and the index values. Indeed, adding securities amounts to adding rows in the payoff matrix of equation (8.3). If a state-price vector still exists, then it must be identical to the vector obtained from (8.3) without the additional securities. This completes the proof of the Proposition. []

Observe that, in a risk-averse world, the expected return of the index is certainly larger than the risk-free return. Therefore, this hypothesis is rather weak.

However, perhaps surprisingly, the above Proposition cannot be extended to multiperiod equiprobable binomial trees. Indeed, when we consider a n-period recombining binomial tree, the set of final index values is given by Su^id^{n-i}, for $i = 1..n$, where u (resp. d) is the coefficient of the increase (resp. decrease) of S in each period, and the set of final state-price densities is given by $C_n^i\ \varphi_{up}^i\varphi_{down}^{n-i}$, where C_n^i is the combinatorial number of paths leading to the corresponding final leaf. It can be shown that when u is larger than d, the mathematical expression Su^id^{n-i} is increasing with i, and the mathematical expression

$\varphi_{up}^{i}\varphi_{down}^{n-i}$ is decreasing, similarly as observed for the one-period case. However, the coefficient C_n^i, which is first increasing and then decreasing with respect to i, no longer guarantees that the relation always holds.

Multinomial trees and Arrow-Debreu prices

Ideally, we would like to extend the previous proposition to the multinomial case. Namely, we would like to prove that in a multinomial tree

$$S_i < S_{i+1} \Rightarrow \frac{\psi_i}{p_i} > \frac{\psi_{i+1}}{p_{i+1}} \text{ for every scenario } i, \tag{8.6}$$

where S_i, ψ_i and p_i respectively denote the price of the index, the state-price and the probability of scenario i.

However, we can provide numerical examples showing that the relations (8.6) do not hold for all multinomial trees. We will give two such counterexamples, involving scenarios which are equiprobable or not.

In order to describe these examples, we display the arbitrage equations defining the state-prices. As usual, the first equation is associated with the risk-free asset, the second one with the index, and the next $(n-1)$ equations correspond to $(n-1)$ options with strike prices equal to the $(n-1)$ lowest possible future values of the index (the largest future value is not interesting because its payoff is always null). So, the arbitrage equations have the following form:

$$\begin{pmatrix} 1 \\ S_0 \\ o_1 \\ o_2 \\ \vdots \\ o_{n-1} \end{pmatrix} = \begin{pmatrix} 1+r & 1+r & \dots & 1+r \\ S_1 & S_2 & \dots & S_n \\ 0 & S_2 - S_1 & \dots & S_n - S_1 \\ 0 & 0 & \dots & S_n - S_2 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S_n - S_{n-1} \end{pmatrix} \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_n \end{pmatrix} \tag{8.7}$$

where the payoff of option j for the scenario i is given by $\max(0, S_i - S_j)$ and the present value of this option is the constant o_j. (One of these equations is redundant, but we are not going to worry about it.)

As before, we also want to impose the condition that the expected return of the investment in any risky asset must be larger than the return of the risk-free asset.

Unequiprobable case

The following system satisfies all the previous conditions:

$$\begin{pmatrix} 1 \\ 10 \\ 0.202 \\ 0.089 \\ 0.013 \end{pmatrix} = \begin{pmatrix} 1.393 & 1.393 & 1.393 & 1.393 \\ 13.653 & 14.957 & 18.174 & 19.91 \\ 0 & 1.304 & 4.521 & 6.256 \\ 0 & 0 & 3.217 & 4.953 \\ 0 & 0 & 0 & 1.736 \end{pmatrix} \begin{pmatrix} 0.631 \\ 0.0632 \\ 0.016 \\ 0.008 \end{pmatrix}. \tag{8.8}$$

If the probabilities of each leaf are respectively given by $(9/10, 1/30, 1/30, 1/30)$ then the vector of state-price densities is given by

$$\varphi = \begin{pmatrix} \psi_1/p_1 \\ \psi_2/p_2 \\ \psi_3/p_3 \\ \psi_4/p_4 \end{pmatrix} = \begin{pmatrix} 0.701 \\ 1.895 \\ 0.474 \\ 0.229 \end{pmatrix},$$

and the relations (8.6) fail.

Equiprobable case

In the previous example, the state-prices $(\psi_1, \psi_2, \psi_3, \psi_4)$ were in reverse order of the index prices, and the probabilities had to be chosen carefully in order to obtain the required counterexample. However, even in the equiprobable case, it is possible to find an example in contradiction with (8.6). For instance, the following arbitrage equations define such an example:

$$\begin{pmatrix} 1 \\ 0.799 \\ 0.403 \\ 0.4 \end{pmatrix} = \begin{pmatrix} 2.5 & 2.5 & 2.5 \\ 0.99 & 1 & 5 \\ 0 & 0.01 & 4.01 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} 0.1 \\ 0.2 \\ 0.1 \end{pmatrix}. \tag{8.9}$$

Conclusions

We have shown here that there is no apparent monotone relation between the index value and the optimal portfolio value when we consider multinomial trees of scenarios. Clearly, bullish portfolios are not necessarily optimal.

In the next section, we show how we could select one of the possible strategies using Dybvig's theorem with respect to the index values.

8.5.6 Strategies and Dybvig's theorem

One-period subtrees

Some points are interesting if we look at the state-price as a function of the index price. To illustrate this, we plot in Figure 8.5 the Arrow-Debreu densities (and the set of probabilities used to define them) as a function of the index price. As above, we here use only the normal distribution and the BS formula to avoid the problems due to incoherencies between the data. We use the S&P500 set, the 29 options for the first period and no option for the second one, and we model the future with an equiprobable two-period tree.

For each of the two periods, the target risk-neutral curves are the same. They are perfectly smooth and decreasing. For the first period, the optimal risk-neutral curve is close to the target one with only deviations at the extremities. For the second period, as we consider no option, the problem is nearly not constrained and the optimal curve matches the target one. This is the reason why we have decided to not consider options for this period. This simplifies the analysis.

In the previous example and for each period, we obtain a smooth decreasing density function for increasing index values. According to Dybvig's theorem, the optimal portfolio values should be increasing when the index values increases. In other words, a bullish strategy is optimal for this subtree.

For each subtree of the second period, if the frictions are not too large or with few effects, we can decide what the optimal strategy to apply is (and the set of corresponding constraints) by only analyzing the state-price density curve. There is a strong link between the strategy and Dybvig's approaches.

Two-period tree

If the curve is smooth over the first period, it is no longer true over two periods. This is still a decreasing function but slightly oscillating. For each leaf of the tree of scenarios, the state-prices over two periods are obtained by the product of the state-prices of the first and the second periods and then are recombined for each distinct index return.

The oscillation comes from the errors between target probabilities and optimal ones at the first period. As the two-period risk-neutral probabilities are obtained by the product of the two one-period probabilities, errors are propagated.

Remember that Dybvig's theorem is defined according to the Arrow-Debreu densities, which are obtained by dividing the state-prices probabilities by the real probabilities of each index value. We could be tempted to use directly the state-prices for equiprobable trees,

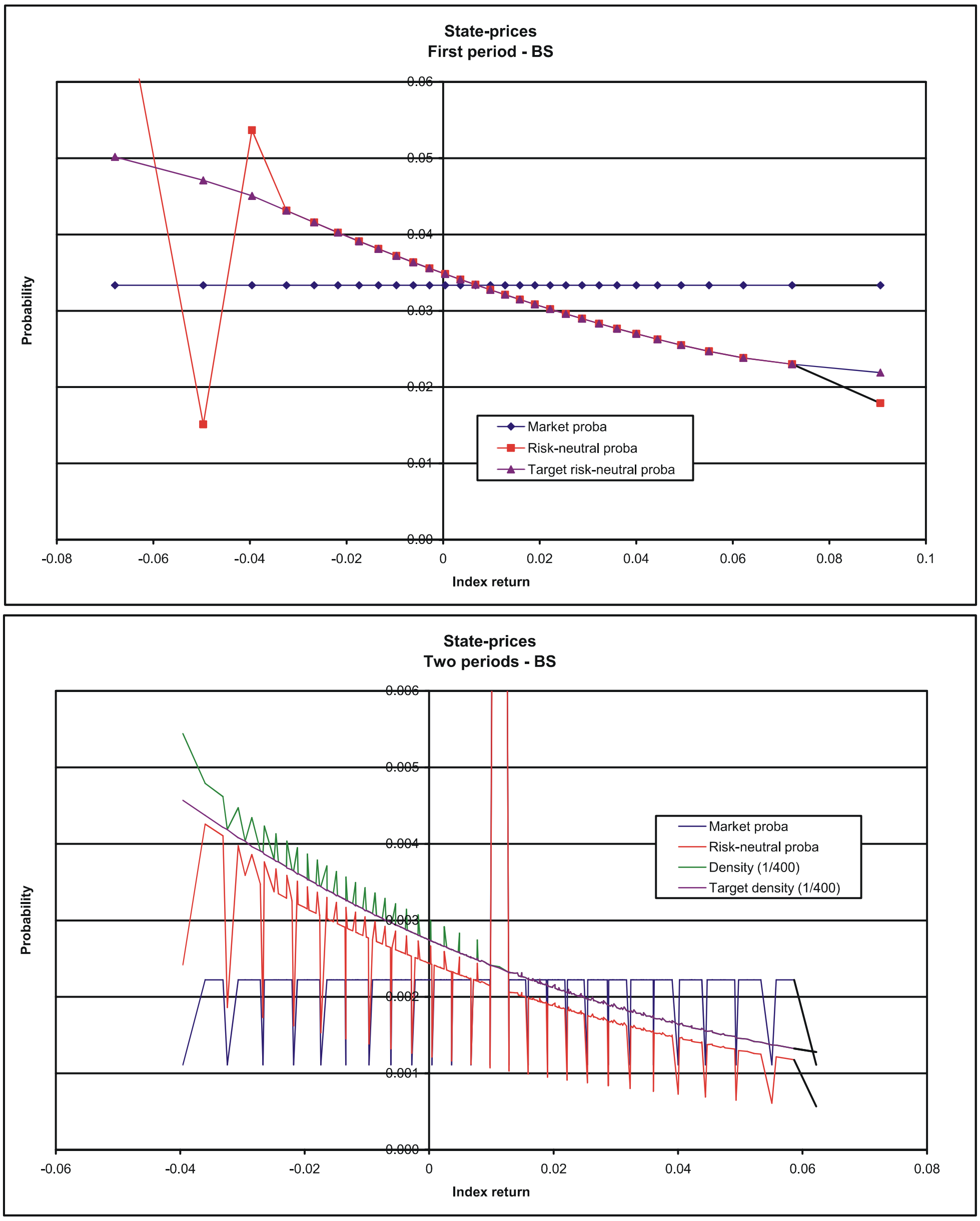


Figure 8.5: Arrow-Debreu densities vs. index prices

as done in Dybvig [22, 23], instead of the densities, but this is not correct for two-period trees. Even when the leaves of the tree are equiprobable, the density curves are usually not similar to the risk-neutral curves. Indeed, the second-period subtrees overlap each other, and therefore, typically, several leaves are defined with the same index value (a kind of implied recombination). So, equiprobable leaves do not imply equiprobable index values. This appears clearly in the second plot of Figure 8.5, in which the risk-neutral curves and real probabilities are highly oscillating. Of course, this effect does not appear in one-period tree, and therefore equiprobable leaves of one-period tree imply equiprobable index values.

Using density curves and Dybvig's theorem, we can now say how optimal portfolio values with respect to index returns are distributed over the two periods.

8.5.7 One-period trees vs. two-period tree

As trading strategies can be applied globally at the final leaves or locally in each subtree, Dybvig's theorem can also technically be applied to the two-period tree or in each one-period subtree. If Dybvig's hypotheses are satisfied, the global use is a specific case of the set of local optimizations. However, when we face some violations, the local approaches can lead to better results. The advantage of the global approach, as for the trading strategies, is that all binary variables α can be define a priori, and so the problem is easy to solve. At the opposite, the local approaches first require to split the VaR probability between the subtrees, which is difficult to do optimally a priori.

8.5.8 Remark

In order to use Dybvig's theorem, we need to obtain valid state-prices. This is not the case when using the option pricing model OP1 (6.5). We must use the second model OP2 (6.7).

Chapter 9

Solving Value-at-Risk problems

9.1 Introduction

In this chapter, we focus on the development of algorithmic procedures for the solution of the portfolio optimization models presented in Chapter 7.

Section 9.2 provides a brief sketch of the classical branch and bound scheme, which is classically used in operations research for the solution of mathematical programming models involving a mix of continuous and integer variables.

In Section 9.3, we return to the financial properties of the VaR model discussed in Chapter 8, and we show how they can be used in order to derive several (heuristic or exact procedures) optimization approaches. We also present two heuristics based on numerical rounding techniques.

Finally, in Section 9.4, we describe pre-processing methods which can be used to instantiate coherently the optimization model, as well as to preselect a promising subset of options for inclusion in the model.

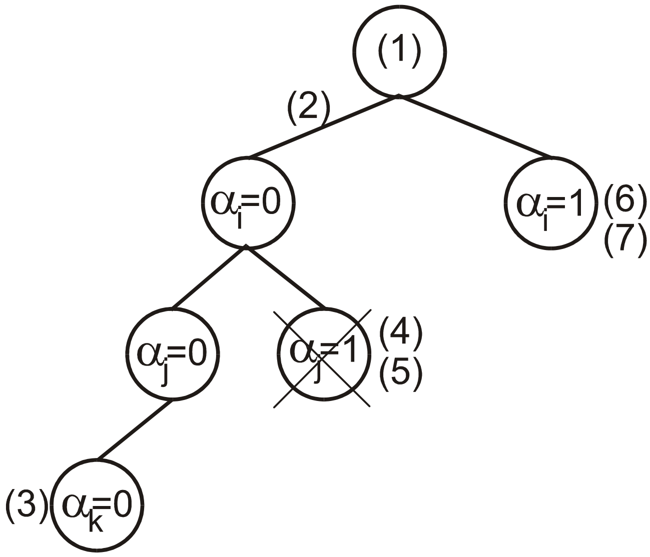
All the procedures described in this chapter have been implemented and tested on benchmark instances. The results of these experiments will be discussed in Chapter 10.

9.2 The branch and bound method

Mathematical programming problems involving both continuous and integer (in our case, binary) variables are usually solved by some version of the branch and bound (BB) algorithm. We rely on a state-of-the-art commercial implementation of the BB method for the solution of our portfolio optimization models. More precisely, we have used the CPlex optimization library, which is open and flexible enough to allow us to tune its parameters in various ways, and even to interfere quite extensively with the logical structure of the BB procedure itself. Therefore, for a good understanding of this and the next chapter, some familiarity with the basic ingredients of the BB method will be necessary. We are now going to review the method very briefly. More details can be found in numerous textbooks, for instance in [56], [69].

We restrict ourselves to the consideration of mixed 0-1 linear programming problems, that is problems requiring the maximization of a linear objective function subject to linear inequality constraints, where each variable is either continuous or restricted to take 0-1 values only.

The BB method considers implicitly all the possible values of the integer variables, but attempts to curtail the enumeration by applying various tests of optimality at intermediate stages. The process is illustrated in the following figure for our binary problem.



1. Relaxation:
   upper bound
2. Set one α
3. First integer solution:
   lower bound
4. Backtrack to an
   unexplored node
5. Obj < lower bound:
   fathomed
6. All the nodes of a
   level explored:
   new upper bound
7. (up-down) < precision:
   stop

The BB process constructs a tree by assigning one of its two possible value to each binary variable. The root of the tree represents the original optimization problem, say IP, and each other node of the tree represents a subproblem of the original problem, say SP, where SP is obtained by fixing a subset of binary variables to specific 0-1 values. At each node, one of three basic procedures can be applied (and typically, all three are applied in succession): we can compute an *upper bound* $U(SP)$ on the optimal value of the current subproblem, we can compute a *lower bound* $L(SP)$ on this value, and we can select a new binary variable to be fixed to either 0 or 1 (this is called *branching*). Let us see this in more detail for a generic node and for the corresponding optimization subproblem SP.

For this subproblem SP, all the binary variables (which have not yet been fixed) can be relaxed, i.e. considered as continuous ones. The resulting relaxed problem at this node is continuous and linear, so that it can be easily solved. This yields an upper bound $U(SP)$ on the optimal value of SP. If this upper bound is smaller than the best available lower bound on IP, i.e. $U(SP) \leq L(IP)$, then the node SP can be fathomed (closed), and the procedure can be repeated with another subproblem.

If, in the optimal solution of the relaxed subproblem, all the binary variables turn out (by chance) to assume a binary value, then this solution is optimal for SP and it is feasible for IP. Therefore, the node SP can again be fathomed, and its value provides a lower bound on the optimal value of IP. This will typically happen at the bottom of the tree, where the number of binary variables is strongly reduced. Otherwise, a lower bound $L(SP)$ on the optimal value of SP and of IP can be obtained by applying a heuristic solution procedure in order to compute a feasible, but typically suboptimal solution of SP. Again, if $L(SP) = U(SP)$, then SP is completely solved and the corresponding node can be fathomed (in practice, the node is fathomed as soon as the gap between these two bounds becomes small enough). If $L(SP) > L(IP)$, then we have improved the best available solution of IP, and the lower bound $L(IP)$ can be updated accordingly.

Finally, if the node SP cannot be fathomed on the basis of the bounds $L(SP)$ and $U(SP)$, then branching can be performed by selecting a binary variable α in SP, and by creating two new subproblems (or nodes) associated to the possible values of α. Then, a new node is selected and the whole process can be repeated.

Note that several branching strategies are possible. Each time a subproblem is selected, we have to decide if we first consider the previous nodes not yet optimized (jumptracking approach) or if we want to follow first the current branch down to its leaves before going back (backtracking approach). Also, each time we want to create two new subproblems, we have to decide on which variable to branch.

Note that, since the number of binary variables is finite, the BB process eventually terminates. Of course, we hope that it will terminate without enumerating all possible binary vectors. But it is important to realize that in practice, mixed integer programming problems involving a large number of 0-1 variables can turn out to be extremey difficult, or even impossible to solve exactly to optimality (remember the extension of Markowitz' model that we tackled in Chapter 2). Although a general statement is difficult to make, since the complexity of each problem is very much influenced by its structure and/or by the numerical value of the parameters in any specific instance, it is safe to say that the solution of problems containing a few hundred binary variables is frequently out of reach, even for the most efficient optimization algorithms.

9.3 Improvements of the BB process and heuristics

9.3.1 Introduction

From the description of the BB method, it should be intuitively clear that its efficiency is strongly determined by the tightness of the bounds $L(SP)$ and $U(SP)$ which are computed for every subproblem SP (we say that a bound is *tight* if it is close to the true optimal value of SP). We observed empirically that, when running on the initial formulation of our portfolio optimization models, the heuristic implemented in CPlex often finds quickly a nearly optimal solution of the problem, although CPlex spends a lot of time in the BB process in order to prove that this solution is actually optimal. We have little information about the exact nature of the CPlex heuristic.

In the following sections, we would like to propose some efficient heuristics for the solution of the portfolio optimization models M1-M2. If the solutions obtained are of high quality, they could be adopted by the investor and suppress the need for the whole branch and bound process. Alternatively, these initial feasible solutions could be used as lower bounds in the BB process, so as to fathom the nodes of the BB tree as soon as possible and to speed up the optimization process. In order to tailor some heuristics for the solution of our models, we will rely on the theoretical insights gained in Chapter 8 (strategies, Dybvig's theorem), as well as on some classical techniques from operations research (rounding and relaxation, BB tuning).

9.3.2 Investment strategies

Principle

In Chapter, Sections 8.3 and 8.4, we have already explained how prior information about the structure of the optimal portfolio could be used to simplify the formulation of model M1. When we know beforehand which strategy is optimal (bullish, bearish, butterfly or volatility), then new constraints can be introduced in order to model the strategy and to reduce the feasible solution space. We have also mentioned, however, that it is usually difficult to predict whether the optimal portfolio displays a bullish, bearish, butterfly or volatility structure (see also Section 8.5.5).

In case we do not know which strategy is best, we can still rely on the same idea to solve the problem heuristically, by picking one of the strategies and solving the corresponding simplified model, or successively considering all four strategies. Let us see how this can be implemented computationally.

Bullish strategies

If we assume that a bullish portfolio is optimal, then the portfolio value is an increasing function of the index values at $t2$. Then, as explained in Section 8.4, we can simplify the optimization problem by selecting a set $\mathcal{S}$ containing the fraction u of scenarios associated with the highest index prices, and by formulating an equivalent linear programming model M1($\mathcal{S}$), where the VaR constraint is only applied to the scenarios in $\mathcal{S}$. (It may be interesting to stress, at this point, that the optimal solution of M1($\mathcal{S}$) is *not* necessarily bullish. Simply, we can claim that every bullish portfolio remains feasible for M1($\mathcal{S}$).)

Now, even if we do not know anything about the structure of the optimal solution of M1, we can still formulate and solve the linear program M1($\mathcal{S}$): in this case, its optimal solution provides a heuristic solution of model M1. We will refer to this heuristic (and/or to the underlying model, when no confusion arises) by the name M1(bullish), and we will report on its performance in Chapter 10.

Let us also note that a slightly tighter formulation can be obtained by observing that, when a bullish portfolio is optimal, then the portfolio value should actually be an increasing function of the index value *for each of the nbS portfolios considered at* $t1$. Therefore, in each of the corresponding subtrees, we can sort the leaves by increasing index values, and the VaR requirement will partition each such list into (at most) two parts: for higher values of the index, the VaR lower bound λB must be satisfied, while it can be violated for smaller values. Globally, at $t2$, the VaR lower bound must be satisfied by a fraction u of the leaves, but this fraction can be different from u for each particular subtree. As a result, we do not know exactly for which leaves of each subtree the Var lower bound λB should hold. However, all we need to know is that, if the constraint is applied for one index value, then it must be applied for all the index values larger than this one in the subtree. This observation can be exploited as follows.

Remember that in model M1, the VaR lower bound must hold in scenario (i, j) (defined as the succession of scenario i in period 1 and scenario j in period 2) when the corresponding binary variable α_{ij} is null (see model M1 in Section 7.3.8). If the scenarios of each subtree are sorted so that the index value is at least as high in scenario (i, j) as in scenario $(i, j-1)$, then the following constraints can be added to model M1:

$$\begin{aligned}&\textbf{local bullish strategy } (\mathbf{nbS}.(\mathbf{nbS}-\mathbf{1})) : \forall i \in [1, nbS],\ \forall j \in [2, nbS],\\ &\quad \alpha_{ij} - \alpha_{i,j-1} \leq 0.\end{aligned} \tag{9.1}$$

Let us call the resulting model M1(local_bullish) (it is "local" in the sense that only the second-period portfolios are required to be bullish). This model has the following property: if one of the optimal solutions of model M1 is bullish at time $t2$, then this solution satisfies constraints (9.1), and hence M1(local_bullish) has the same optimal value as M1.

But even when we do not know anything about the optimal solution of M1, we can still impose (9.1) and solve M1(local_bullish) to obtain a heuristic solution of model M1. The value of this heuristic solution is at least as tight as the value provided by M1(bullish). However, M1(local_bullish) involves binary variables and hence, must be solved by BB. Our hope is that, because of the constraints (9.1), model M1(local_bullish) may be easier to solve than M1 (for instance, when a variable is fixed to 0 in the branching process, constraints (9.1) automatically force other variables to 0 as well). Actually, as the goal is to find (quickly) a good lower bound and not the optimal value, parameters of the BB method can be fixed to stop the enumeration process after a given time, or as soon as the best available feasible solution is "good enough" (i.e. as soon as its value comes within a predefined gap, or percentage, of the optimal solution value). Preliminary experiments indicate that it is not useful to spend too much time computing this lower bound, as this slows the main optimization process.

Again, this will be tested empirically in Chapter 10.

Bearish strategies

The case of the bearish strategy is similar to the previous one. A model M1(local_bearish) is obtained upon replacing the constraints (9.1) by the following ones:

$$\begin{aligned}&\textbf{local bearish strategy } (\mathbf{nbS}.(\mathbf{nbS}-\mathbf{1})) : \forall i \in [1, nbS],\ \forall j \in [1, nbS-1],\\ &\quad \alpha_{ij} - \alpha_{i,j+1} \leq 0.\end{aligned} \tag{9.2}$$

Volatility and butterfly strategies

When the optimal portfolio has a volatility structure, the VaR constraint can be applied to the scenarios associated with the smallest and the largest index prices, so that the appropriate set $\mathcal{S}$ consists of two separate blocks. So, when the scenarios are sorted by nondecreasing index price, the α-variables take successively value 0, then 1, then 0, as represented schematically in Figure 9.1.

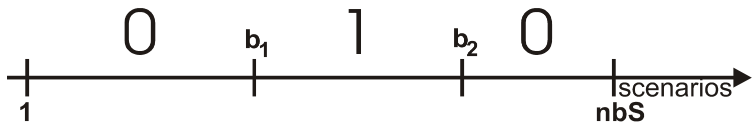


Figure 9.1: Volatility and butterfly strategies

As already mentioned in Section 8.4.2, a difficulty with this approach is that the set $\mathcal{S}$ is not as easily determined as in the bullish or in the bearish case. To obtain a workable model, let us first sort the scenarios by increasing index values, and let us view the thresholds b_1 and b_2 as two decision variables (see Figure 9.1). It is enough to impose that the α-variables should be zero to the left of b_1 and should be nonincreasing to the right of b_1 (this implicitly determines the value of b_2).

So, for each subtree i, let us introduce two variables b_{1i} and vol_i, where b_{1i} is the position in the sorted list of scenarios corresponding to the upper bound of the first block, and vol_i is a binary variable which is null for the scenarios on the right of b_{1i}. Then, we can formulate the following constraints for each scenario (i, j).

$$\begin{array}{ll}
\textbf{local volatility strategy } (\mathbf{3nbS^2 - nbS}) : \forall i \in [1, nbS], & \forall j \in [1, nbS] \\
\quad b_{1i} - j \leq nbS(1 - \alpha_{ij}) & //j < b_{1i} \Rightarrow \alpha_{ij} = 0 \\
\quad j - b_{1i} + 1 \leq (nbS + 1)(1 - vol_i) & //j \geq b_{1i} \Rightarrow vol_i = 0 \\
\quad \alpha_{ij} - \alpha_{i,j-1} \leq vol_i & //vol_i = 0 \Rightarrow \alpha_{ij} \leq \alpha_{i,j-1} \\
\quad vol_i \in \{0, 1\}. &
\end{array} \tag{9.3}$$

We will call M1(local_volatility) the model obtained by adding these constraints to M1.

Clearly, the same approach can be used for the butterfly strategy except that the VaR constraint should be applied between b_1 and b_2 (see Figure 9.1) rather than at the end of the interval. This gives rise to a model M1(local_butterfly), obtained by assing the following constraints to M1.

$$\begin{array}{ll}
\textbf{local butterfly strategy } (\mathbf{3nbS^2 - nbS}) : \forall i \in [1, nbS], & \forall j \in [1, nbS] \\
\quad b_{1i} - j \leq nbS\alpha_{ij} & //j < b_{1i} \Rightarrow \alpha_{ij} = 1 \\
\quad j - b_{1i} + 1 \leq (nbS + 1)(1 - vol_i) & //j \geq b_{1i} \Rightarrow vol_i = 0 \\
\quad \alpha_{ij} - \alpha_{i,j+1} \leq vol_i & //vol_i = 0 \Rightarrow \alpha_{ij} \leq \alpha_{i,j+1} \\
\quad vol_i \in \{0, 1\}. &
\end{array} \tag{9.4}$$

Remarks

Notice that the bullish and bearish strategies are extreme cases of the volatility strategy: the bullish strategy is obtained by setting b_1 to minus infinity, and the bearish strategy by setting b_2 to infinity in Figure 9.1. Therefore, model M1(local_volatility) subsumes both M1(local_bullish) and M1(local_bearish). Therefore, in principle, only the models M1(local_volatility) and M1(local_butterfly) need to be solved in a heuristic approach. These models cover all possible ways to apply the VaR lower bound to scenarios associated either with extreme values, or with an interval of intermediate values within the range of index prices.

9.3.3 Uses of Dybvig's theorem

Two-period approach

A heuristic approach to the solution of model M1 can be derived by "pretending" that Dybvig's theorem is valid for M1. Namely, we can sort the scenarios (i, j) according to their two-period state-price density φ_{ij}, determine the set $\mathcal{S}$ containing the fraction u of scenarios with the lowest values of φ_{ij}, set up the associated model M1($\mathcal{S}$), and solve it by the simplex method (or any other LP optimization technique). We will denote this heuristic approach as M1(Dybvig).

Note that we could bring the model closer to respecting Dybvig's hypothesis by completing the "market", as previously discussed in Section 8.5.4. But we have already mentioned that this would increase the size of the optimization model. As far as market imperfections go, we expect that, the smaller the frictions, the closer the optimal value of M1(Dybvig) will be to the optimal value of M1.

Local approach

Just as we did when working with investment strategies, we can also take as a starting point that the inverse relation between portfolio values and state-price densities should hold individually and independently for each of the nbS portfolios formed at time $t1$, but not necessarily for the whole range of portfolio values at $t2$. In this "local" approach, the state-prices should be computed over the second period only.

There are several ways to implement this basic idea.

A rather straightforward (though not very promising) approach consists in imposing that the VaR constraint should be satisfied with the same probability u by *each* of the nbS

second-period portfolios subtree. Then, for each subtree, it suffices to collect the smallest $u \times nbS$ state price densities φ_2 into a set $\mathcal{S}$, and to replace the model M1 by the linear programming model M1($\mathcal{S}$) in the usual way. We call M1(Dybvig:equipartition) the resulting model. Although we cannot expect this simplistic heuristic to be very tight, we will use it as simple benchmark in Chapter 10.

A less trivial approach consists in repeating what we did in Section 9.3.2 for the bullish and the bearish strategy. Namely, assume now that the scenarios are sorted by nonincreasing state price density: i.e., assume that the state price density of scenario (i, j) is at least as large as the state price density of scenario $(i, j+1)$. Then we can add the following set of constraints to model M1:

$$\begin{aligned} &\textbf{local Dybvig M1 } (\mathbf{nbS}.(\mathbf{nbS}-\mathbf{1})) : \forall i \in [1, nbS],\ \forall j \in [1, nbS-1], \\ &\quad \alpha_{ij} - \alpha_{i,j+1} \geq 0. \end{aligned} \tag{9.5}$$

We denote the resulting mixed 0-1 linear programming model as M1(local_Dybvig). The constraints of M1(local_Dybvig) ensure that, in each subtree, the values α_{ij} are a monotone function of the state price densities and hence, that the VaR lower bound λB is satisfied for the scenarios with the lowest state price densities. Therefore, the optimal value of M1(local_Dybvig) coincides with the optimal value of M1 when Dybvig's conclusions hold. Otherwise, the optimal solution of M1(local_Dybvig) provides a heuristic solution of M1.

Finally, let us mention another "local" approach based on model M2 rather than M1 (see Section 7.3.9). Model M2 splits the optimization process into two parts: the partition of the VaR probability u among second-period subtrees on one hand, and the satisfaction of the resulting VaR constraint within each subtree on the other hand. Accordingly, in model M2, the variables γ_i represent the maximal fraction of scenarios which may violate the VaR lower bound within subtree i ($i = 1, 2, \ldots, nbS$). If the values γ_i are available, then we can rely (heuristically) on Dybvig's theorem to fix the variables β_{ij} in M2. Formally, assume that the scenarios are sorted by nonincreasing state price density, as in equation (9.5), and denote by Pr_{ij} the probability of scenario (i, j). Then, we can formulate a new model M2(local_Dybvig) by adding the following constraints to model M2:

$$\begin{aligned} &\textbf{local Dybvig M2 } (\mathbf{nbS^2}) : \forall i \in [1, nbS],\ \forall j \in [1, nbS], \\ &\quad \gamma_i - \sum_{k=1}^{j} Pr_{ik} \leq \beta_{ij}. \end{aligned} \tag{9.6}$$

These constraints express that, as long as the cumulated probability of scenarios $(i, 1)$ to (i, j) does not exceed γ_i, the corresponding variable β_{ij} must be set equal to 1. Thus, the

VaR lower bound is again satisfied for those scenarios associated with the lowest values of the state price densities, in agreement with the conclusions of Dybvig's theorem.

Observe that both M1(local_Dybvig) and M2(local_Dybvig) are mixed integer programs and hence, must be solved by branch and bound. As was the case for the models based on intuitive investment strategies (M1(local_bullish), M1(local_bearish), etc.), it is sufficient to solve these models approximately in order to produce a heuristic solution of the portfolio optimization problem.

Note also that, both in M1(local_Dybvig) and in M2(local_Dybvig), the method is to apply Dybvig's theorem locally. Even if the formulations are different, the constraints have the same effect; i.e. they locally define the binary variables according to the sequence of state-price densities. Moreover, M2(local_Dybvig) requires one more set of control variables (γ) and more constraints. Therefore, in the sequel, we no longer consider M2(local_Dybvig), but we rather prefer to use M1(local_Dybvig).

9.3.4 Rounding approach

In this section, we turn to heuristic optimization procedures based on rounding techniques. Procedures of this nature are well-known in the integer programming literature, but the reader should note that they do not exploit explicitly the structure nor the economic interpretation of the optimization model.

Complete rounding

If, in model M1, we replace the integrality constraints

$$\alpha_{ij} \in \{0,1\} \;\; \forall i,j \in [1, nbS]$$

by the weaker contraints

$$0 \leq \alpha_{ij} \leq 1 \;\; \forall i,j \in [1, nbS],$$

we obtain a linear programming model which we call the *linear relaxation* of M1, and which we denote by M1Relaxed. Since M1Relaxed is a linear programming problem, it can be solved quite efficiently. Let U^* be its optimal value, and let α^* denote the values assumed by the variables α in the optimal solution of M1Relaxed.

We have already noted in Section 9.2 that U^* is an upper bound on the optimal value of M1. We have also observed that, if (by chance) $\alpha^*_{ij} \in \{0,1\}$ for all scenarios (i,j), then the optimal solution of M1Relaxed is also optimal for M1.

When some of the values α^*_{ij} are fractional, however, the optimal solution of M1Relaxed does not have an immediate economic interpretation. But even in this case, we can still hope that a good heuristic solution of M1 can be obtained by rounding each α^*_{ij} to the closest binary value. Intuitively, this solution should be especially tight if all values α^*_{ij} are already close to zero or one (although there is no theoretical guarantee that it should be so).

When implementing this approach, some care should be exercised in order to make sure that the rounding operation preserves the VaR constraints: we cannot simply set each variable to the nearest binary value, otherwise the VaR lower bound could be violated by too many few scenarios. To solve this difficulty, we have developed the following procedure M1(rounding):

1. Solve the linear programming problem M1Relaxed by the simplex method and record the optimal values α^*_{ij}.

2. Sort all scenarios (i, j) in nondecreasing order of the difference $\min\{\alpha^*_{ij}, 1 - \alpha^*_{ij}\}$, i.e. the difference between α^*_{ij} and the nearest binary integer.

3. Consider successively each scenario (i, j) in the previous list, and set the corresponding variable α_{ij} to the binary value α^{**}_{ij} nearest to α^*_{ij}: i.e., set $\alpha^{**}_{ij} = [\alpha^*_{ij}]$. As soon as the VaR probability is attained by this assignment (i.e., as soon as $\sum_{(i,j)} Pr_{ij}\,(1-\alpha^{**}_{ij}) \geq u$), set all remaining variables to 1, irrespective of their initial value.

4. If the VaR probability cannot be attained, then set to 0 some of the α_{ij}'s already fixed to 1 among those with the initial highest values $\min\{\alpha^*_{ij}, 1 - \alpha^*_{ij}\}$.

5. Solve M1 with all binary variables fixed at the value α^{**}_{ij} determined in Step 3.

A few comments are in order regarding the procedure M1(rounding). Note that by working with a sorted list in Step 3, we give priority to the values which are already very close to either one or zero, and not to the "fuzziest" ones. Moreover, as long as the VaR constraint

$$\sum_{i=1}^{nbS} \sum_{j=1}^{nbS} Pr_{ij}\,(1 - \alpha^{**}_{ij}) \geq u$$

is satisfied, the solution produced by the procedure is feasible for the original problem M1. In fact, if we denote by $\mathcal{S}$ the set of scenarios for which α^{**}_{ij} is set to 0 in Step 3, then Step 4 amounts exactly to solving the linear programming model M1($\mathcal{S}$), as we defined it in Section 8.3.

Partial rounding

In Step 3 of the previous approach, all the variables are successively set to a binary value. It could happen, however, that most of the values α^*_{ij} are close to 1/2, in which case it is less likely that the optimal assignment should necessarily be to the nearest binary value. Therefore, it makes sense to define an alternative heuristic, whereby we only round those values which are initially close to one or zero, and we use a reduce BB process to optimize the remaining values.

In order to be able to control the efficiency of the reduced BB process, we specify beforehand a parameter fd, which represents the percentage of variables to be fixed in Step 3 prior to entering the BB phase. Now, we need again some care if we want to preserve simultaneously the VaR requirement and the percentage f. We have implemented the following heuristic M1(Rounding & BB, f%):

0. Define the parameter $0 \leq f \leq 100$.

1. Solve the linear programming problem M1Relaxed by the simplex method and record the optimal values α^*_{ij}.

2. Sort all scenarios (i, j) in nondecreasing order of the difference $\min\{\alpha^*_{ij}, 1 - \alpha^*_{ij}\}$, i.e. the difference between α^*_{ij} and the nearest binary integer.

3. While at most f% of the variables have been assigned, and while the VaR probability is not attained, consider successively each scenario (i, j) in the previous list, and set the corresponding variable α_{ij} to the binary value α^{**}_{ij} nearest to α^*_{ij}.

4. If the VaR probability has been attained by setting less than f% of the variables, then set the free variables to the value 1, in decreasing order of α^*_{ij}, until the required percentage is reached.

5. If the VaR probability cannot be attained , even by setting all remaining free variables to 0 (i.e., if the current assignment is infeasible), then set to 0 some of the α_{ij}'s already fixed to 1 among those with the initial highest values $\min\{\alpha^*_{ij}, 1 - \alpha^*_{ij}\}$, and set all remaining free variables to 0.

6. Solve M1 by BB, with the binary variables fixed as determined in Steps 3-5.

Typically, the computational time required by this procedure decreases when f increases, but so does the quality of the solution that it produces.

Note that M1(Rounding & BB, 1) is equivalent to M1(rounding).

9.4 Preselection of options

9.4.1 Introduction

In this section, we would like to examine whether we can automatically determine a priori (i.e., before the portfolio model is optimized) a subset of "promising" options which are likely to appear in the optimal portfolio. Preselecting the options would allow to reduce the size of the optimization model, and hence to speed up the optimization process.

Before we get to this point, however, we first need to explain what type of options are included in the basic formulation of the model. We handle this issue in the next subsection.

9.4.2 Constructing the "universe" of options

As an underlying guiding principle, we would like our models to reproduce (automatically) many of the conditions faced by an investor on a real-world financial market. In particular, the models should involve options displaying realistic features.

In our computer implementations, we provide several ways to select the universe of options to be considered during the optimization process. The options are grouped in three sets: those covering only the first period, those covering both periods (and possibly extending beyond the horizon), and those created at the beginning of the second period. The options in each set can be defined by a variety of different methods.

The most straightforward method is to provide, in an input file, the list of all the options to be considered. As input, the user must encode the type (call or put), the strike price, the maturity time, and the bid and ask prices of each option. For the options covering only the second period, the input is more complex since the user must also specify for which subtree(s) of the second period these options exist. For such options, other input methods appear desirable.

A second way to generate options requires only to specify the total number of options, as well as their class (only calls, only puts, or calls and puts). Then, an automated procedure can be used to generate a given number of options displaying certain predefined features. In our implementation, the strike prices are uniformly distributed over a range $[l_S, u_S]$ of index values. The range of index values can be defined so as to avoid extreme strike prices, for which the option would be out-of-the-money with very high probability at the end of the period. We set the lower endpoint l_S equal to the expected index price minus one standard deviation. Similarly, the upper bound u_S is defined by adding one standard deviation to the expected price.

The third method, defined for options available at $t0$, creates a number of options equal to the number of scenarios nbS (this number is doubled if both calls and puts are required). For options covering only the first period, the strike prices coincide with the index values of the first-period subtree. For options covering both periods, the strike prices are given by the median index value of each second-period subtree. The investor can still decide if he wants only calls, only puts, or calls and puts.

The fourth method concerns the options covering only the second period: it creates nbS options in each of the nbS second-period subtrees (or a total of $2nbS^2$ options if the user wants both calls and puts) by introducing a strike price for each index value at the end of the subtrees. This approach is attractive to the extent that it completes the market (as required by Dybvig's theorem), but it creates a large number of options, and hence, it tends to slow down the optimization process.

Our last method is probably the most realistic and interesting one. Here, the user specifies the number of options required and their type (only calls, only puts or calls and puts). Then, a procedure automatically generates the given number of options, with strike prices computed according to the usual market rules (based on the time series of index values; see Section 6.2.3). For the options covering only the second period, the user can leave it to the procedure to decide how many options should be created for each set of scenarios. In this case, the procedure analyzes the past index values and constructs the strike prices based on each specific time series, according again to usual market rules.

9.4.3 Advanced selection of options

The above methods allow to construct a wide variety of options, but there is no way to know whether any of these options is interesting for the investor until the end of the optimization process. So, on one hand, the investor may be tempted to include a very large number of options in the optimization model, in order not to forego any profit opportunity. But on the other hand, considering too many options increases the complexity of the optimization problem and decreases the efficiency of solution algorithms. Hence, it would be very useful to be able to select a set of promising options before the main optimization takes place, and to consider only this reduced set in the portfolio model.

Since it is difficult to know before the optimization whether an option is promising or not, our idea is to make the decision on the basis of a reduced branch and bound optimization process. Initially we decide how many options will be considered, and how many will be eventually selected. As usual, we also fix the type of the options and the bid-ask spread.

The other characteristics are automatically defined by the procedure. In particular, the strike prices are fixed according to the usual market rules.

Then, the full portfolio optimization model is passed to the reduced BB procedure. Note that this model is exactly the model that we wish to optimize, but we limit the execution of the BB procedure, both in precision (gap) and in running time. The other CPlex parameters BB are set as usual.

Now, the idea is to track all feasible intermediate solutions generated by the reduced BB process. Each time an improved feasible solution is found at a node, the quantities of all options that appear in the portfolio are recorded. At the end of the reduced BB process, the options which have appeared frequently and in significant quantities, i.e. those with the largest total recorded quantities, are deemed to be most "promising" and are selected. The user is also allowed to add his own choice of options to this selection.

Finally, all selected options are used to set up again the portfolio model, and the main optimization procedure is run on this smaller model.

Chapter 10

Computational experiments

10.1 Introduction

In this chapter, several parameters of th VaR models are numerically considered and analyzed. The aim is essentially to illustrate and discuss the models presented in Chapters 7-9. Therefore, no more experiments are performed here to study how to represent the future, and how to price the options (see Sections 5.6 and 6.7 for the respective numerical results).

In Section 10.2, we first define the general framework; i.e. the market conditions faced by the investor, and the objectives of this investor. As, according to the results obtained in Sections 5.6 and 6.7, the future and option pricing models have to be considered beforehand to instantiate the VaR model, the construction of the tree of scenarios is also explained.

Section 10.3 briefly presents in which computer environment the numerical experiments were performed. In particular, we make a short description of the software that we have written in order to handle all the models described in the part two of this thesis.

The numerical experiments, and the corresponding results, are discussed in Section 10.4. In Section 10.4.1, we first look for the optimal number of scenarios to construct in order to obtain, in a reasonable computation time, coherent and stable portfolio returns. The computation time also depends on the precision required by the investor; i.e. the maximal difference that he accepts between the optimal solution of the problem and the best solution found by the optimization process. This is presented in Section 10.4.2. Then, in Section 10.4.3, the structure of the optimal portfolio is studied to illustrate this typical portfolio payoff pattern, which is often encountered when considering VaR constraints, and which justifies the development of specific heuristics in order to solve the problem. Those heuristics are presented and commented in Section 10.4.4. In Section 10.4.5, we show how the optimal expected portfolio return evolves when different sets of options are considered at $t0$ and

Parameter	**Value**
Market	CBOE
Options	SPX (S&P500)
Period 0: t0=0	19/2/2001
Period 1: t1=1	19/3/2001
Period 2: t2=2	23/4/2001
Number of observed options (first period)	126
Initial index price	1246.23 USD
Index dividend yield	0.1058%
Index transaction cost	0.30%
Risk free rate	0.42%
Historical index return	1.131%
Historical index volatility	3.658%
Options transaction cost	0.30%
minimal cost per option	2 USD
maximal cost per option	20 USD

Table 10.1: Market data

$t1$. This is only possible thanks to the dynamic property of the two-period VaR model. Similarly, when no adjustments of the portfolio are allowed at $t1$, i.e. when we consider a static "one-period" tree of scenarios, the portfolio performances are highly affected. This comparison of dynamic two-period trees vs. one-period trees, is described in Section 10.4.6. Throughout this work, we have also stressed that the option prices used in the problem must be coherent with the index return distribution. An example of what happens when this is not the case is given in Section 10.4.7. Finally, in Section 10.4.8, some modifications of the financial parameters are considered.

In the last Section 10.5, we draw some conclusions.

10.2 The financial problem

10.2.1 Market

On February 2001, on the CBOE, an investor would like to invest in a portfolio of options on S&P500. He considers only the options on this major US index, and the risk-free asset. The index is not integrated in the portfolio because the investor wants to avoid the additional work implied by the construction of a stock portfolio which mimics the index.

The future is modelled by a two-period tree of scenarios. The length of each period is equal to one month in order to allow the investor to adjust his portfolio when new options appear on the market. Two probability density functions are considered to model the future index returns: the implied pdf constructed from the observed option prices, and the Normal pdf. We expect to have a better representation of the future with the first pdf than with the second one. However, this will allow us to compare the classical Normal model with a more advanced one.

The investor can purchase options covering each of the two periods. For the first period, the bid and ask option prices as well as the strike prices are observed on the CBOE. In order to be coherent, the target option prices at $t0$ are either the observed prices if we use the implied distribution, or, otherwise, the Black and Scholes prices. For the second period, options are created artificially according to usual market rules. Therefore, the sets of options vary from one subtree to another. This time, the target option prices are either the improved BS prices, which take into account the smile effect of the first period, when the implied pdf is used, or simply the classical BS prices otherwise.

The target bid and ask option prices, the initial index price and the index dividend yield are taken from the CBOE internet site. The index expected return and volatility are computed over the last 10 years from prices extracted from DataStream.

The bid and ask prices, as well as the risk-neutral probabilities, which are required by the heuristics based on Dybvig's theorem, are computed using the option pricing model OP2.

Table 10.1 summarizes the market parameters. The parameters defined to model the future and the options are given in Table 10.2. In the next sections, we consider some perturbations of these parameters and the consequences on the optimization results.

10.2.2 Investor's decisions

Table 10.3 summarizes the investor's decisions for the VaR portfolio problem. He defines a minimal guaranteed level of 95% after two months; i.e. he accepts to face the risk to lose 5% of his initial investment after two months in the worst case. This corresponds to a maximal loss of 26% over one year. However, he requires at the same time, that, with a probability of 95%, the return on his investment will be positive and larger than 5.15% per year (0.84% over the two months); i.e. larger than the return of the safest investment (return of the risk-free asset). This is a protective portfolio strategy.

Parameter	**Value**
Modelling the future	
PDF	
Implied distribution	
utility function	Power
subsampling size	400
equiprobable leaves	No
Normal distribution	
equiprobable leaves	Yes
Number of scenarios	
one-period subtree	30
full two-period tree	900
Modelling option prices	
Model	OP2
2nd period	
number per subtree	30 (15 calls & 15 puts)
number for the 2nd period	900
target price	$\text{BS}_{smile}/\text{BS}$
strike price	Market rule
spread	2%
Cleaning	Applied
Modelling VaR constraints	
Model	M1
Guarantee	Strong
BB backtracking	Deep

Table 10.2: Parameters of the models

Parameter	**Value**
Initial budget	1 000 000 USD
Guarantee level	95%
VaR level	100.84%
VaR probability	95%
Adjustment of the portfolio at t1	Allowed
Investment in the index	Unallowed
Quality of the solution (gap)	0.01%

Table 10.3: Investor's decisions

10.3 Computer environment

10.3.1 The software

All the numerical experiments were made on a 600MHz notebook computer with 128Mb RAM. The operating system is MS Windows 98 and all the times given in the results are real time.

A software was written in language C++ to solve the problems described in the preceding chapters. The software is highly customizable, and can handle all the models and parameters presented above. In order to achieve this, the software is initialized by reading a file containing numerous parameters. This file is described in the second appendix.

The two optimization processes, i.e. option pricing and portfolio optimization, are performed by calls to routines of the professional CPlex 6.6 library. Therefore, we take advantage of improved versions of optimization methods. Moreover, these routines can also be customized according to our goals. The next section presents some of the CPlex parameters.

The instantiation of all the models presented in this work, and their conversion in CPlex format is performed by the software. Also, the software handles the construction of the probability density functions, the definition of the option strike prices and target prices, the cleaning and preselection of options, the simulated annealing process to improve the prices of options, and the treatment of the results.

10.3.2 CPlex parameters

The speed of the BB process, as well as the quality of the solution that it produces, can be improved by tuning several CPlex parameters. We consider especially the following ones:

1. **Branching.** We can impose a priority order for branching on the variables. Instead of letting CPlex fix this order (for instance, according to the magnitude of the coefficients), we can impose to branch first on the value 0 for those VaR variables associated with a low state-price density (cf. Dybvig's theorem).

2. **Jumptracking.** Two basic branching strategies are available: either backtracking or jumptracking. With backtracking, CPlex performs a depth-first traversal of the tree, in the hope to find as quickly as possible a valid lower bound. With jumptracking, the algorithm moves from one node to another node at the same level before going down the tree. Each time a level of the tree has been fully considered, we obtain a new smaller upper bound on the optimal value. Since the CPlex heuristic tends to provide

a good lower bound, we prefer to rely on a jumptracking strategy in order to reduce quickly the gap before the best available upper and lower bounds.

3. **Root heuristic.** We can force CPlex to use its root heuristic (a reduced BB process, probably similar to our partial rounding heuristic).

4. **Advanced starting solution.** If we know an initial integer feasible solution (as it is the case when we compute an initial heuristic solution), we can give it as input to CPlex before starting the BB process. This provides an initial lower bound which can be useful to trim down the enumeration tree.

5. **Storage management.** If the problem size is large, then the memory required during the BB process can become huge. In our experiments, using the Windows virtual memory resulted in dramatic performance improvements (use of the processor was brought down from 98% to 10% in some cases). Most of the computing time is actually spent in disk swaps. A way to control this phenomenon is to fix a CPlex limit to the physical memory that can be used to store the BB tree. If this limit is reached, then CPlex write the current tree (the part it does not need immediately) to disk and continues with the next nodes.

6. **Gap.** CPlex stops the BB enumeration when the best available lower bound comes within a predetermined percentage (called *gap*) of the best available upper bound. Reducing the gap translates usually into much longer computing times. Since, in our case, the objective function is the expected portfolio value, and since this value only provides a rough summary (point-estimate) of the quality of the future portfolio value, it does not seem necessary to impose a very tight gap.

7. **Other parameters.** CPlex provides other control parameters, such as the number and the nature of cuts used, a variety of heuristics, etc.

10.4 Numerical results

10.4.1 Number of scenarios and pdfs

The first task which has to be performed for handling the portfolio problem described in Section 10.2, is to construct the tree of scenarios. In order to do it, as described in Chapter 5, the implied and Normal probability density functions are stratified. The larger the sample size, i.e. the number of scenarios per subtree, the better the representation of the continuous distribution. Moreover, large sample sizes are safer to avoid "leaks" in the VaR and guarantee constraints. However, this also leads to larger problems and larger computation times.

When at least 30 scenarios are considered, the sample is a good representation of the continuous distribution. This was already observed in Section 5.4.6. Moreover, for all sample sizes equal to or larger than 30 scenarios, the VaR constraint and the guarantee constraints are nearly perfectly satisfied. In order to check this, we compute the value of the optimal portfolio at the leaves of a new two-period tree with 50 times more nodes at each second-period subtree. Obviously, as expected, both constraints are more violated when using the scenario approach (7.3) to model the guarantee constraint than when using the strike-price model (7.10).

nbS	**Leaves**	**Tree**	**BB**	**Expected return**	**First gap**	**Final gap**
Normal distribution						
20	400	2"	7'02"	0.58% ([7.16%,7.23%[)	0.44%	0.01%
30	900	2'05"	5h	0.54% ([6.69%,8.01%[)	0.43%	0.21%
40[1]	1 600	4"	5h	0.54% ([6.75%,8.49%[)	0.42%	0.27%
50[1]	2 500	6"	5h	0.52% ([6.44%,8.85%[)	0.47%	0.37%
100[1]	10 000	8"	10h	0.52% ([6.47%,9.40%[)	0.47%	0.45%
Implied distribution						
20	400	3"	22"	0.63% ([7.90%,7.97%[)	0.05%	0.01%
30	900	4"	3h30'51"	0.61% ([7.56%,7.62%[)	0.09%	0.01%
40	1 600	4"	5h	0.61% ([7.61%,7.81%[)	0.26%	0.03%
50	2 500	5"	5h	0.60% ([7.43%,7.81%[)	0.19%	0.06%
100	10 000	15"	10h	0.56% ([6.98%,7.88%[)	0.23%	0.14%
[1] model OP1 is used to price the options						

Table 10.4: Number of scenarios

Table 10.4 gives some effects of the sample size on the optimization process. The first two columns give respectively the sample size and the resulting number of leaves at $t2$. The third column indicates the time required to prepare the data for the VaR problem; i.e. time spent in the pdf construction, stratification, probability conversions, option pricing and pre-processes. The fourth column is the time spent in the branch and bound optimization of the VaR problem. The BB process was not allowed to work more than 5 hours (10 hours for the largest sample size). The best expected portfolio return found is indicated in the fifth column. This is a monthly return. The corresponding yearly return, and the upper bound on the optimal solution, are given between brackets. The gaps between the best feasible solution found and the best possible solution (the optimal solution of the relaxed problem) vary depending on the number of scenarios. The first and final gaps are given in the last two columns. Note that the gap is measured with respect to the objective function expressed as a portfolio payoff and not as a portfolio return. Thus, there is no linear relation between the expected returns in column five and the final gap.

The construction of the tree of scenarios only requires a few seconds for each sample size and both pdfs. Most time is spent in the conversion of the Normal consensus pdf into the risk-neutral counterpart. The computation time increases exponentially with the number of scenarios. Therefore, it was not possible to compute the converted pdf for sample sizes larger than 30 scenarios and the Normal distribution. Thus, the option pricing model (OP1) is used for large sample sizes.

The optimization of the VaR problems requires more time and blows up when the number of scenarios increases. For larger sample sizes, it was even impossible to precisely compute the optimal portfolio within the time limit of 5 hours. This appears also clearly when we look at the first and final gaps. The first gap is obtained at the first iteration by CPlex heuristic (note that the time required varies from a few second to several minutes according to the number of scenarios). Comparing the first and the final gap shows how quickly the optimization process progresses. This is fast for few scenarios and very slow for large sizes. Note also that the problem seems more difficult when the future is represented by the Normal distribution.

In order to represent correctly the continuous pdf and to avoid troubles with the VaR constraints, sample sizes sizes smaller than 30 are not adequate. Moreover, since we wanted small problems leading quickly to optimal solutions, 30 scenarios turned out to be suitable. We would be more confident if the optimal returns are stable when more scenarios are constructed. Indeed, it would mean that 30 scenarios already represent precisely enough the future for the VaR problem. We have to be careful before drawing conclusions from the figures in Table 10.4. First, one should not forget that each problem, i.e. for each number of scenarios, is different. Indeed, the options covering the second period are defined according to market rules. Each time we increase the number of scenarios, additional subtrees are defined in the second period; each with a new set of options. However, as each set of options is defined according to the same set of market rules, this should not have a large impact on the optimal objective value. Secondly, the increase of the number of scenarios implies that we were not able to compute precise optimal solutions in a reasonable laps of time. We could expect that the optimal solution obtained for trees with large numbers of scenarios can be improved when more time is provided. However, it appears in Table 10.4 that all

Gap	Time	Expected return
Normal distribution		
0.43%	* 23"	0.54% ([6.69%,9.47%[)
0.30%	17'18"	0.54% ([6.69%,8.59%[)
0.21%	5h	0.54% ([6.69%,8.01%[)
Implied distribution		
0.09%	56"	0.60% ([7.46%,8.02%[)
0.08%	1'26"	0.60% ([7.49%,7.97%[)
0.07%	1'36"	0.61% ([7.55%,7.96%[)
0.06%	2'50"	0.61% ([7.55%,7.91%[)
0.05%	6'31"	0.61% ([7.55%,7.84%[)
0.04%	15'11"	0.61% ([7.55%,7.78%[)
0.03+%	* 20'04"	0.61% ([7.56%,7.76%[)
0.03%	38'10"	0.61% ([7.56%,7.72%[)
0.02%	1h53'47"	0.61% ([7.56%,7.66%[)
0.01%	3h30'51"	0.61% ([7.56%,7.62%[)

Table 10.5: Computation time

the optimal results are close and that the range of optimal returns, for a given number of scenarios, is always nearly a subrange of the results obtained for more scenarios. Therefore all the results are compatible, and we consider that 30 scenarios is enough for all the subsequent experiments.

10.4.2 Computation time

Even for 30 scenarios, the computation times given in Table 10.4 are disappointing. The investor would certainly like to obtain more rapidly a solution. In fact, the computation time also depends on the precision required by the investor. It is well known that the branch and bound process requires more and more time to reduce the gap obtained for the successive solutions. This is due to the explosion of nodes in the bottom of the branch and bound tree. This is shown clearly in the first two columns of Table 10.5.

Even more frustrating for the investor, the best feasible solution is usually found well before the end of the optimization process. The branch and bound optimization takes time to check that the best feasible solution encountered is really the best one (within the required gap). The time spent before obtaining the best portfolios, is marked with a star in Table

10.5. For the Normal pdf, the best solution is the first one computed, 5 hours before the end of the process. For the implied pdf, the best solution is found 3 hours before the end of the optimization process.

In the numerical experiments presented in Table 10.4, the branch and bound process stops either when the time limit has expired, or when the gap between the best feasible solution and the upper bound, becomes smaller than 0.01%. This gap implies that the best feasible yearly return obtained is at most about 0.05% smaller than the best possible feasible solution of the problem. A feasible solution within this gap can be considered perfect by the investor. However, the investor could already be satisfied with a 0.05% gap. Indeed, this corresponds to an upper bound on the optimal yearly return of about 0.30% more. In the case of the implied distribution, the optimization process requires only 6 minutes instead of more than 3 hours. Therefore, in the sequel, we set the required gap to 0.05%.

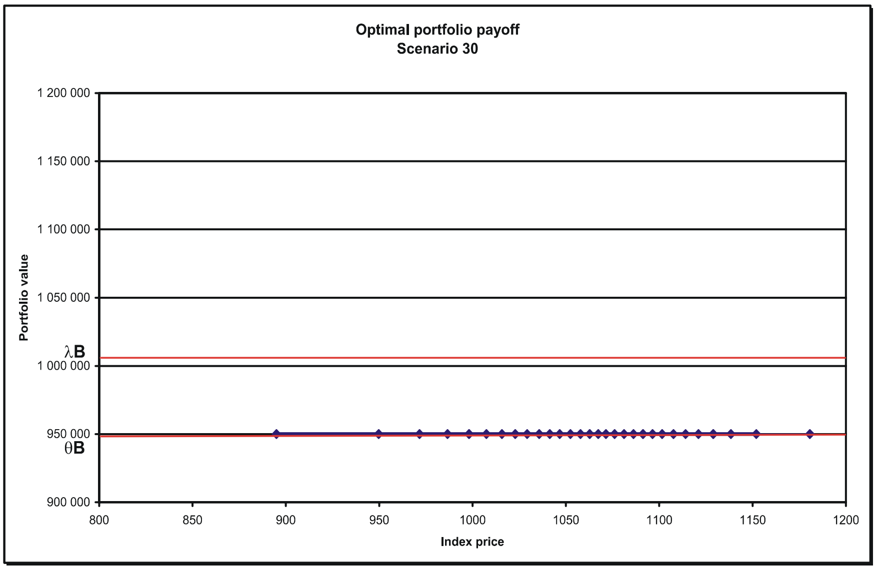
10.4.3 Structure of optimal portfolios

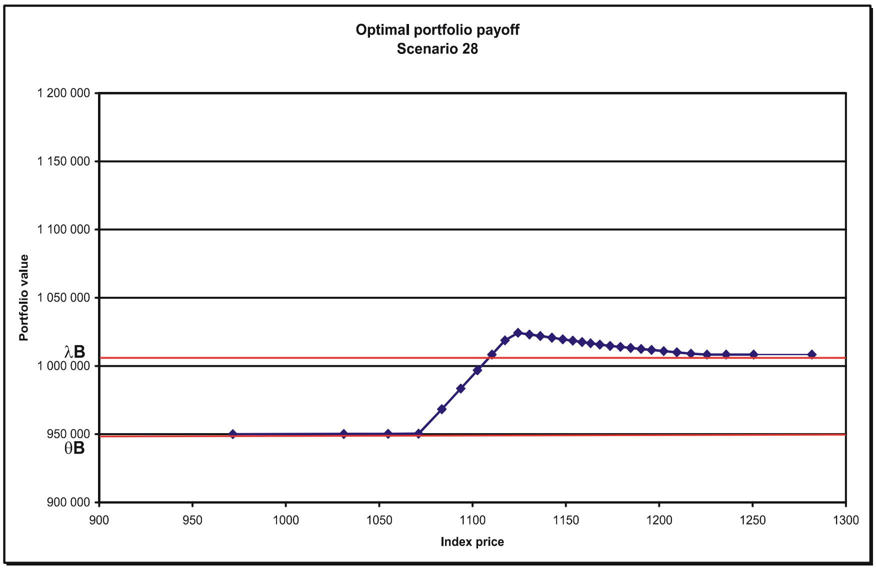
Dert and Oldenkamp [20] observed, for one-period problems with guarantee and VaR constraints, that the portfolio consists of few options. Therefore, we have investigated the structure of the portfolio and derived heuristics to solve the problem. For the problem presented in Section 10.2, typical results also are observed. In this section, the results are detailed for the implied distribution. The same conclusions also hold for the Normal pdf.

First, let us have a look at Figures 10.1-10.2. Each plot illustrates the portfolio payoff for a subtree of the second period. The horizontal axis represents the portfolio values at $t2$. The corresponding portfolio value is reported on the vertical axis. A mark is plotted on the portfolio payoff line for each index value corresponding to a leaf of the subtree. Also, the guaranteed level (θB) and the VaR level (λB) are indicated. Note that it is impossible to plot such a figure for the whole tree, since the initial portfolio is adjusted at the root of each subtree, and therefore it constitutes a new independent portfolio each time.

We observe that the payoff line matches the guaranteed level for the 30th subtree, i.e. for the smallest index values, then progressively shifts to the VaR level when upper subtrees (subtrees 28-27) are considered, to match perfectly the VaR level for upper subtrees (subtree 26). The pattern of subtrees 7 to 26 are identical to the pattern of subtree 26; i.e. the portfolio payoff is equal to the VaR lower bound. Finally, when the largest index values are considered, the portfolio payoff increases linearly (subtrees 5 to 1).

This increasing piecewise payoff can first be explained by the first moments of the index return distribution. As for the example given by Dert and Oldenkamp [20], the expected





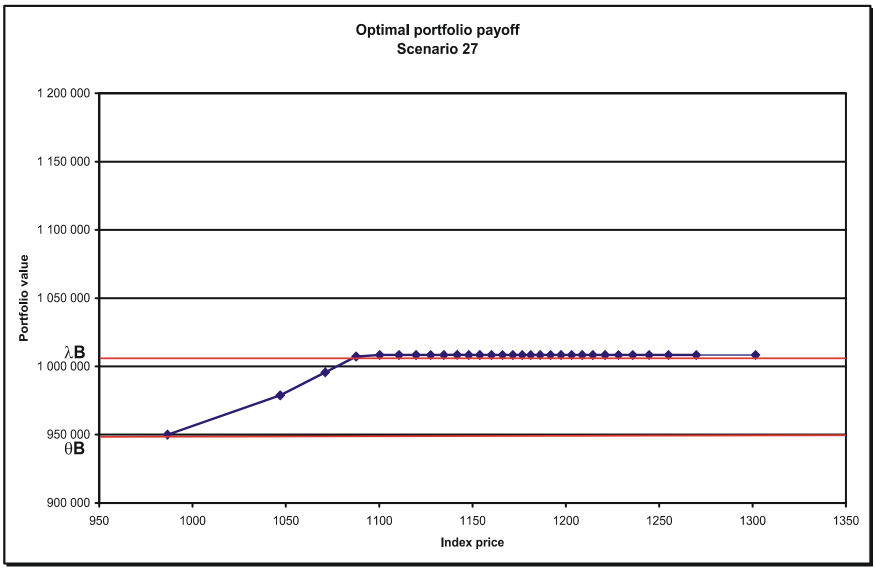


Figure 10.1: Optimal portfolio payoff: scenarios 27-38

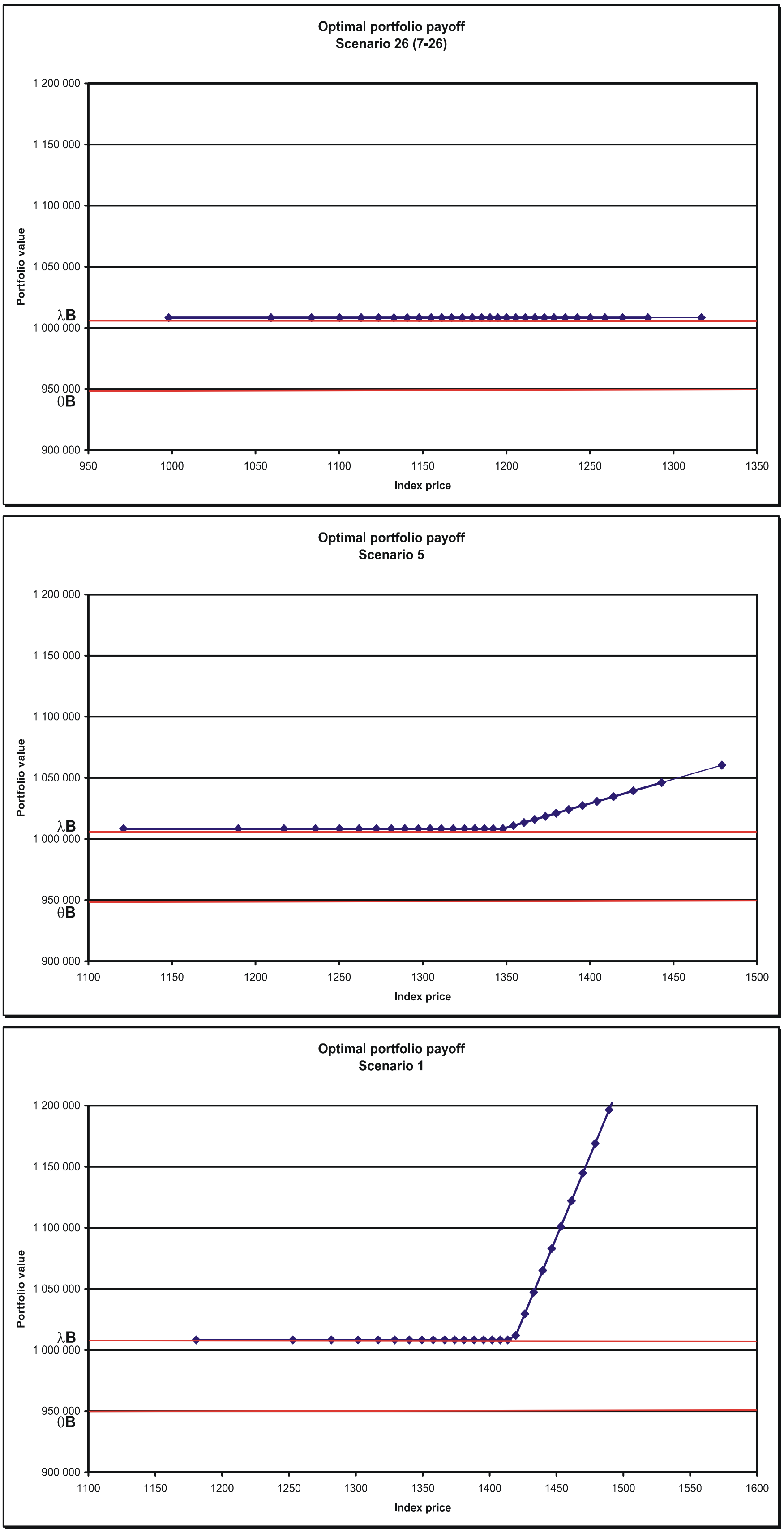


Figure 10.2: Optimal portfolio payoff: scenarios 1-26

mean of the S&P500 is large while the volatility is reasonable. As explained in Section 8.4.2, the investors expect an increase of the index value and are quite confident in the size of this increase. Therefore, they will invest such as to maximize the portfolio payoff when the index is large; this corresponds to a bullish strategy. As they do not expect small index values in the future, this also implies that they are indifferent to the portfolio payoff in those cases, and that they will not pay more to obtain larger payoffs. Therefore, the optimal portfolio values are close to the Var and guarantee lower bounds for small index values.

Dert and Oldenkamp [20] are particularly interested by the "step" pattern that we observe in subtrees 27 and 28. In the VaR problem that we consider here, the VaR probability is high, and therefore, few subtrees and index values are constrained by the guarantee constraint. So, the step appears in few subtrees. Note however that such steps can be created by using a trading strategy called in finance "bull spread". This strategy involves two calls or two puts (see Hull [32] for details).

Figures 10.1-10.2 also show that each payoff line is flat for most of the subtrees. It is piecewise in four parts in subtrees 27 and 28 to define the "VaR step", and is piecewise in two parts for the uppest subtrees. In Section 7.2.6, we have shown that the portfolio payoff is a piecewise linear function with breakpoints at the option strike prices. Therefore, Figures 10.1-10.2 indicate that at most 4 option strike-prices, and so 8 options (calls and puts), are required in each subtree to obtain the corresponding portfolio payoff. This is indeed the case when we look at the exact composition of the optimal portfolios. For subtrees 1-26, the optimal portfolio consists of a risk-free investment that provides the VaR level at $t2$. Options do not appear in the portfolio before scenario 6, when one long call is required to create the breakpoint in the piecewise pattern. For subtrees 27-29, the risk-free investment provides at least the guaranteed level. Up to 8 options are used in short and long positions to create the steps. In subtree 30, the portfolio is only a risk-free investment. Note also that this implies a block structure in most of the subtrees.

A similar analysis can be performed for the first period. The budget constraint is the sole constraint applied at $t1$. This is a normalizing constraint and so, it does not imply a particular portfolio payoff pattern at the end of the first period. The VaR and guarantee constraints, which shape the portfolio, are only applied at $t2$. Therefore, the portfolio values at $t1$ are computed such as to define the optimal budget in each second-period subtree in order to maximize the final expected portfolio payoff (with respect to the final constraints). As a bullish strategy appears optimal for this problem, it implies that larger budgets should be available in the upper subtrees than in the lower ones. This is actually observed in the numerical results. This is depicted in Figure 10.3. Note that we consider no index and no option covering both periods, and so the budget available at the root of each subtree is equal to the portfolio value.

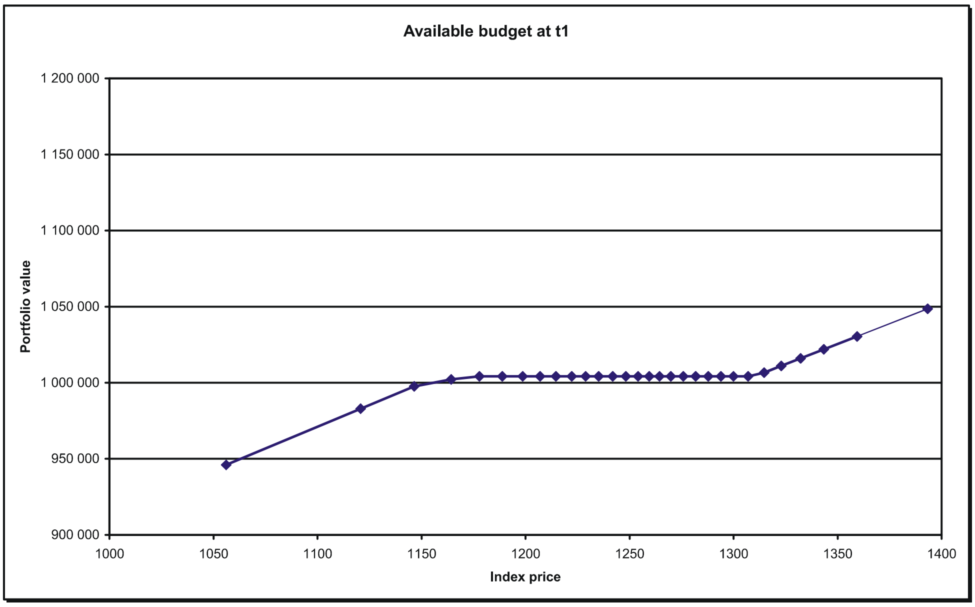


Figure 10.3: Budgets at t1

Again, since the portfolio payoff is a piecewise linear function with breakpoints at the option strike prices, and since the shape of the payoff in Figure 10.3 is simple, few options are required. Numerically, only 3 options, with maturity at $t1$, appear in the optimal portfolio. This structure is observed because all the options initially purchased mature at $t1$. We can expect more complex portfolio shapes when we consider options covering both periods (and also because options initially purchased and maturing at $t2$ are directly considered in the VaR and guarantee constraints (see Section 10.4.6)).

Of course, we cannot draw general conclusions only from this specific example. It would be interesting to consider an example in which the bullish strategy is not so efficient. However, even for other strategies, similar results in which the portfolio payoff is close to the Var and guaranteed levels in some subtrees are expected, to allow larger payoffs in the other subtrees. Therefore, we think it should be possible to speed up the optimization process by using specific heuristics.

10.4.4 Initial bounds

For the VaR problem described in Section 10.2, computation times and optimal expected returns are given in Sections 10.4.1-10.4.2. Even if the structure of the optimal solution, described in Section 10.4.3, is “simple”, the branch and bound process requires time to compute a satisfying solution. Therefore, we analyze in this section if the heuristics, presented in Chapters 8-9, speed up the process.

The heuristic results are given in Table 10.6 for the two pdfs. Each line corresponds to the results of the heuristic identified in the first column. The second column corresponds to the best expected yearly return of each heuristic. A symbol plus is added in front of those returns which are the lower bounds of the branch and bound process used by the heuristic. All the other returns are exact optimal returns since they are obtained by the simplex method. The third column gives the gap between the heuristic portfolio payoff and the upper bound initially obtained by the simplex method over the relaxed problem. The last column gives the time spent in the heuristic process. As the investor wants to obtain quickly a solution, the computation time for each heuristic is limited to 2 minutes. The heuristics are sorted by decreasing expected returns.

Nearly all the heuristics presented in Table 10.6 are discussed in Chapter 9. One additional heuristic named M1(Dybvig:bullish) is considered. In M1(Dybvig:bullish), the VaR probability is split between the second-period subtrees according to the final index values and a bullish strategy. First, all the leaves are sorted by decreasing index values. Then we split the list into two parts with probabilities u and $(1-u)$. Finally, the VaR probability associated to each subtree is given by the sum of the leaf probabilities corresponding to the subtree in the first part of the list.

We have also added to the table the upper bound obtained by the relaxation of the VaR problem, and the portfolio which consists only in the risk-free asset. This gives the two extreme possible portfolio values. Finally, we indicate the best solution obtained without heuristics, and the best upper bound that we know (returned by the branch and bound process at the end of the optimization of M1).

The three families of heuristics perform similarly for both the Normal and implied distributions. Indeed, when considering only the expected returns, the best solutions are obtained by the rounding heuristics, then by the strategy approaches, and finally by the heuristics using Dybvig's theorem. All the heuristics using the simplex method only require a few seconds. The rounding heuristics need less than one minute and the other branch and bound processes are stopped after two minutes.

The rounding heuristic with a reduced branch and bound process performs particularly well, since it provides nearly the optimal solution for the implied distribution in 26 seconds instead of more than 4 minutes. The solution, obtained in 9 seconds, is even better in the case of the normal distribution, since the branch and bound optimization over the VaR

Heuristic	**Bound**	**Gap upper bound**	**Time**
Normal distribution			
Upper bound (relaxation)	9.47%		1”
Upper bound (BB, gap 0.21%)	8.01%		5h00’00”
Upper bound (BB, gap 0.06%)	7.32%		One week
M1(Rounding & BB,800/900)	+6.96%	0.39%	47”
M1(Rounding & BB,850/900)	+6.96%	0.39%	9”
Optimal solution (gap 0.21%)	6.69%	%	5h00’00”
CPlex heuristic	6.69%	0.43%	23”
M1(local_bullish)	+6.57%	0.45%	2’00”
M1(bullish)	5.95%	0.55%	2”
M1(Dybvig:bullish)	5.95%	0.55%	2”
M1(local_Dybvig)	+5.95%	0.55%	2’00”
M1(Dybvig)	5.67%	0.59%	1”
M1(Dybvig:equirepartition)	5.19%	0.67%	2”
M1(Rounding & BB,900/900)	5.19%	0.67%	3”
Risk-free investment	5.169%	0.67%	0”
M1(local volatility)	-	-	2’00”
Implied distribution			
Upper bound (relaxation)	8.02%		2”
Upper bound (BB, gap 0.01%)	7.62%		3h30’51”
Optimal solution (gap 0.05%)	7.55%	0.07%	6’31”
M1(Rounding & BB,800/900)	+7.55%	0.07%	26”
M1(Rounding & BB,850/900)	+7.47%	0.08%	12”
CPlex heuristic	7.46%	0.09%	56”
M1(bullish)	6.19%	0.29%	1”
M1(Dybvig:bullish)	6.19%	0.29%	1”
M1(local_bullish)	+6.11%	0.30%	2’00”
M1(Dybvig)	5.45%	0.40%	2”
M1(local_Dybvig)	+5.38%	0.41%	2’00”
M1(Rounding & BB,900/900)	5.23%	0.44%	3”
M1(Dybvig:equirepartition)	5.20%	0.44%	1”
Risk-free investment	5.169%	0.45%	0”
Local strategy: volatility	-	-	2’00”

Table 10.6: Heuristics to compute a lower bound

problem was unable to provide a feasible solution in the required gap in less than 5 hours, nor even to improve this heuristic solution in one week! Note also that Cplex heuristic, which is probably based on the same approach, gives similar results, but in more time. The investor could be tempted to only use the rounding heuristic. However, the upper bound obtained by relaxing the problem is still very far from the heuristic solution, and the upper bound provided by the branch and bound performed by the heuristic is not applied to the original problem. Therefore, we have no guarantee that the heuristic solution is close to the optimal one without performing a complete optimization. This is true for all the heuristics presented here.

It would be very useful to define heuristics to compute low upper bounds on the optimal solution. A typical approach is to consider cuts during the branch and bound process. However, when CPlex is asked to use all the classical cuts available in its library, none is applied. This suggests that classical cuts are not efficient for the VaR problem and that specific financial ones have to be defined.

The heuristics based on strategies give poor results. This could be partially explained by the fact that the assumptions behind these heuristics (bullish, volatility, stability) are stronger than in the rounding heuristics. Especially, when we apply these approaches to the final leaves, only one set of values is possible for the binary variables α . However, in Section 10.4.3, we have observed that the optimal portfolio payoff is an increasing function of the index value in each subtree. This seems to indicate that the bullish strategy is optimal. However, this does not prove that the global bullish strategy is optimal, but this only provides indications locally at the subtrees. This would imply that the global approach cannot integrate the possibility to adjust the portfolio at $t1$. This freedom results clearly in a complication since the subtrees typically overlap each other, and thus there exist several leaves defined with the same index value, but with different portfolio values. The poor results obtained when using the bullish strategy locally at each subtree do not contradict this explanation. Indeed, the optimal solution found in this case is provided by a branch and bound process, whose execution has been stopped after 2 minutes. Therefore, this is only a rough lower bound on the optimal solution. When the process is not stopped before completion (gap 0.01%), the yearly return, 7.51%, is close to the optimum, but it requires 3'46".

Dybvig's results are even more disappointing. However, the reasons are evident in this case: Dybvig's hypotheses are not satisfied. Indeed, the market is uncomplete in each subtree (less than 16 independent securities for 30 scenarios) and not perfect (transaction costs and spreads are applied). Moreover, the guarantee constraint is modelled by the strike price

Heuristic	Bound	Gap upper bound	Time
Implied distribution			
Upper bound (relaxation)	8.26%		2”
Optimal solution (gap 0.05%)	7.68%	0.09%	22’25”
Dybvig: bullish	6.49%	0.28%	1”
Dybvig	5.86%	0.38%	1”
Dybvig: BB model 1	+5.78%	0.39%	2’00”
Dybvig: equirepartition	5.45%	0.44%	1”

Table 10.7: Dybvig’s approaches and the guarantee constraint

approach, and therefore the constraint is satisfied for all possible index values and not only for the values associated to each leaf of the tree. Therefore, we can expect that this heuristic will give better results for other problems with less violations of Dybvig’s hypotheses.

It would be interesting to determine, for each violation, the effect on the quality of the heuristic solution. However, this cannot generally be computed. Indeed, completing the market or removing the costs modify the problem, and lead to larger optimal expected payoffs. It becomes impossible to compare the gaps between the heuristic solutions and the optimal ones, since the latter are different for each new problem. Therefore, we only provide, in this section, results when the guarantee constraint is modelled by the alternative formulation (7.3). In this case, and for the 900 scenarios at $t2$, the portfolio payoff is not too much affected. Results are given in Table 10.7. Note first that the upper bound and the optimal expected return are slightly larger than with formulation (7.10) of the guarantee constraint. The difference is not significative and the new optimal value is in the gap of the old one. Also, as was expected, the heuristic’s results are improved. The increase in the yearly return is not large, but already significative. However, due to the violations of Dybvig’s hypotheses, these heuristic returns remain low even when using this alternative formulation of the guarantee constraint.

Finally, note that Dybvig’s heuristics based on a bullish strategy, returns exactly the same solution as the global bullish strategy. Indeed, as observed in Section 8.5.6, there is a perfect inverse relation between the one-period state price densities and the index prices. Therefore, in each subtree, it is equivalent to define the VaR variables according to the decreasing index values or the increasing state price densities. Since the VaR probability is also defined in each subtree according to the decreasing index values, the two heuristics are identical.

ID	**Index**	**Options $t0$**	**Options $t1$**	**Time**	**Expected return**
Normal distribution					
NE1	No	126	30x30subtrees	5h	0.54% ([6.69%,8.01%[)
NE2	No	126	60x30subtrees	5h	0.54% ([6.74%,7.83%[))
NE3	No	126	60	5h	0.61% ([7.65%,8.62%[))
NE4	No	0	30x30subtrees	4'22"	0.50% ([6.16%,6.48%[))
NE5	No	126	0	11'12"	0.42% ([5.18%,5.49%[))
NE6	Yes	126	30x30subtrees	5h	0.56% ([6.96%, 7.81%[)
Implied distribution					
NE7	No	126	30x30subtrees	6'31"	0.61% ([7.55%,7.84%[)
NE8	No	126	60x30subtrees	16'26"	0.64% ([8.02%,8.32%[)
NE9	No	126	60	10'26"	0.67% ([8.39%,8.72%[)
NE10	No	0	30x30subtrees	4'26"	0.61% ([7.55%,7.88%[)
NE11	No	126	0	57"	0.46% ([5.74%,6.05%[)
NE12	Yes	126	30x30subtrees	5'14"	0.61% ([7.55%,7.88%[)

Table 10.8: Number of options

10.4.5 Options and index

Until now, we have always considered the same number of options. We would like to know how the optimal portfolio payoff and the computation time vary when we change the number of options, or when we consider the index. This is especially interesting since we work with a two-period model in which we can change the number of options in each period. This is, to our point of view, a first advantage of a dynamic two-period model vs the static one-period approach.

The results are given in Table 10.8. The first column identifies each numerical experiment. The next three columns indicate which securities are considered. The fifth column gives the total time spent in the optimization process. Finally, the last column gives the optimal expected returns. In the numerical experiments NE2 and NE3 (and NE8-NE9), 60 options are considered during the second period. In NE2, the set is constructed according to the market rules, and therefore the sets of options are different in each subtree. In NE3, the same set of options, i.e. with the same strike prices, is considered for each subtree.

As we can expect, when the number of securities is increased, the portfolio payoff increases or remains constant. The same is observed about the computation time.

When we remove options during one of the two periods, the problem becomes easier since there is no more than one security, i.e. the risk-free asset, to consider in the corresponding period. The amount invested in this asset is directly given by the available budget at the beginning of the period. Therefore, the optimization process requires less time. Moreover, in experiments NE4 and NE10, the budget at $t1$ is the same for all subtrees. Each subtree is nearly independent; the sole link is the VaR constraint. These experiments clearly show that the options covering the second period contribute more to the optimal expected portfolio payoff than the options with maturity at $t1$.

The case of the index is interesting. The index does not appear in the optimal solutions neither in NE6 nor in NE12. In fact, the index can be considered as an option with a null strike price, but the optimization process prefers options with larger strike prices. However, surprisingly, considering the index in NE6 allows the optimization process to reduce more quickly the gap. This is another indication that the computation time may significatively vary depending on the data. Similarly, when we compare experiments NE2 and NE3, or NE8 and NE9, even if all these problems have the same size (the only differences are the values of the strike prices), the gap is reduced quicker in NE3 and NE9.

10.4.6 One-period vs two-period model

In the previous section, we have already noticed that considering new options during the second period increases the optimal objective value more than options covering only the first period. We would like here to show that the dynamic two-period model is superior to the one-period model.

Obviously, if the horizon is the same in both models, the dynamic two-period model cannot perform worst than the one-period model. Indeed, the two-period tree is a more general model which includes the one-period tree. In order to compare both models, we construct in this section a one-period tree with the same representation of the future; i.e. the same final index values and number of leaves. In order to do it, we construct the two-period model, but remove the possibility to adjust the portfolio at $t1$, and to consider other options that the ones covering both periods.

Table 10.9 gives the optimal results for which 56 options with maturity at $t2$ are considered in the one-period model; i.e. all the options observed on the CBOE. The previous results for the two-period model, with all the observed options or just options covering the second period, are given again.

The two-period model clearly dominates the one-period approach. This is not due to the selection of options considered at $t0$ since NE6 and N3, without option initially, perform

ID	Model	Time	Return	Final gap
Normal distribution				
NE1	1 period (56 options)	5h	0.42% ([5.18%,5.79%[)	0.10%
NE2	2 periods (126+30 options)	5h	0.54% ([6.69%,8.01%[)	0.21%
NE3	2 periods (0+30 options)	4'22"	0.50% ([6.16%,6.48%[))	0.05%
Implied distribution				
NE4	1 period (56 options)	3'21"	0.42% ([5.17%,5.37%[)	0.03%
NE5	2 periods (126+30 options)	6'31"	0.61% ([7.55%,7.84%[)	0.05%
NE6	2 periods (0+30 options)	4'26"	0.61% ([7.55%,7.88%[)	0.05%

Table 10.9: one-period model vs. two-period model

better than NE4 and NE1. Moreover, the optimal portfolios for NE2 and NE5 contain only three options. NE4/NE5 and NE3/NE2 are very similar.

The difference is due to the VaR constraint which is very restrictive. The optimal return for the one-period model is close to the risk-free return. Without the freedom to consider new options at $t1$ and the possibility to adjust the portfolio, it is not possible to satisfy the VaR constraint using other securities than the risk-free asset.

This also explains why NE3 and NE6 perform so well even if the initial portfolio is only composed of the risk-free asset. The VaR constraint can be optimally handled by only using options covering the second period, and by optimizing the second-period subtrees. Therefore, portfolios composed of only the risk-free asset or only few options with maturity at $t1$, can be seen as a side effect of a strong (high bound and large probability) VaR constraint and the possibility to use distinct set of options (as explained in Section 10.4.3). In Section 10.4.8, other VaR parameters are considered.

10.4.7 Consistency

Until now, we have carefully separated the Normal and Implied cases; i.e. what we could call the theoretical and real cases. In order to be consistent, the Black and Scholes option prices are used as target when using the Normal pdf, and the observed CBOE prices are used as target when using the implied pdf. We cannot model the future by a Normal distribution when we want to consider the real market. Indeed, the observed option prices are not close to the Black and Scholes prices. When we consider the 126 observed option prices, the mean difference is 37.12% with respect to the BS prices. Therefore, when we try to use the observed prices and a tree with normally distributed index returns, the optimal expected portfolio payoff becomes unrealistic (5.40% per period or [91.26%,91.84%[per year).

10.4.8 Financial variations

In Section 10.4.6, we have already noticed that the VaR constraints for the problem described in Sections 10.2.1-10.2.2 are very restrictive. This is due to a large probability u and to a high VaR lower bound. On the other hand, a large probability u implies that only a very small subset of the variables α must be set equal to zero, and therefore that the problem is probably easier to solve than one with a smaller probability u .

When we consider the same problem as before, but rather with a VaR probability of 80%, we obtain, for the implied distribution, the results in Table 10.10. As expected, the computation time explodes. Model M1 cannot be solved as fast as observed before in Section 10.4.2. Moreover, the problem is less constrained and so the expected return is larger. Note also that the heuristics based on strategies now give the best results, even better than the result provided by Cplex heuristic.

Model	**Expected return**	**Time**
M1 (final gap 0.37%)	0.77% ([9.74%,12.21%[)	5h00'00"
M1(local_bullish)	+0.75% (9.42%)	2'00"
M1(Rounding & BB,800/900)	+0.75% (9.42%)	2'00"
M1(bullish)	0.70% (8.79%)	1"
M1(Dybvig:bullish)	0.70% (8.79%)	1"
CPlex heuristic	0.68% (8.55%)	8"
M1(Dybvig)	0.56% (6.96%)	1"
M1(Rounding & BB,900/900)	0.46% (5.46%)	3"
M1(Dybvig:equirepartition)	0.44% (5.46%)	1"

Table 10.10: VaR probability u=80%

10.4.9 Normal pdf vs. implied pdf

It is interesting to compare the results obtained for the two probability density functions. We already observed in Table 10.4 of Section 10.4.1 that the time required to optimize the VaR problem not only is a function of the number of variables and constraints in the model, but also depends on the nature of the data. Table 10.11 gives the computation time required to obtain the best portfolio returns within nearly the same gap. Note that the optimal solution found for the Normal case was obtained by the rounding heuristic, and that the branch and bound process requires one week just to reduce the upper bound. This shows that the data used to create an instance of the problem influence strongly the computation time. The Normal model is clearly more difficult to handle than the implied model.

Pdf	*nbS*	**Time**	**Expected return**	**Final gap**
Normal pdf*	30	1 week	0.56% ([6.96%,7.32%[)	0.06%
Implied pdf	30	6'31"	0.61% ([7.55%,7.84%[)	0.05%
* Rounding heuristic applied				

Table 10.11: Best returns for both pdfs

It is clear from the results in Section 5.6, that the Normal distribution gives a different representation of the future than the implied pdf. Therefore, we can expect that the optimal returns vary with the pdf used to model the future. According to the results in Table 10.11, the Normal world underestimates the expected returns since the Normal upper bound is smaller than the implied lower bound.

In order to compare both distributions, it is also tempting to test the two optimal portfolios, i.e. one for each pdf, on historical market data. However, by definition, only one scenario can happen in reality. Therefore, such a test would compute, for a specific past scenario, the portfolio value which corresponds to the optimal portofolio structure which was computed to maximize the portfolio value *in mean* over a large set of possible scenarios. These are two different contexts. This comparison would only be valid if it was possible to perform this test several times and to take the mean result. But, it is impossible since only one historical set of market data corresponds to each two-period tree of scenarios.

A more interesting comparison would be to compare the structure of all the portfolios at $t0$ and $t1$ for both distributions. Indeed, the optimal portfolios could have the same structure, and the differences in the optimal portfolio returns could only come from the distributions of prices. Unfortunately, it is not possible to compare the adjusted portfolios at $t1$ for the Normal distribution and the implied distribution since the scenarios at this time correspond to two different representations of the world, characterized by completely different parameters. However, as we suggest a roll-over strategy, only the initial portfolio is important for the investor. As most of the initial parameters are observed on the market, this comparison should be possible. But, this is not the case for all the parameters. In order to avoid abnormal results and to insure coherency, the initial target option prices for the Normal case are the BS prices instead of the market prices. Neverthless, the initial optimal porfolio structures are provided in Table 10.12. The two portfolios are clearly different. This is even more obvious in Figure 10.4 when considering the portfolio payoffs at $t1$, independently of the index distribution.

Implied distribution		**Normal distribution**	
Asset	**Quantity**	**Asset**	**Quantity**
Risk-free	999 965	Risk-free	999 965
Call 1300	535	Call 1280	261
Call 1310	-387	Put 1240	112
Put 1150	-182	Call 1285	-534

Table 10.12: Optimal portfolio structures

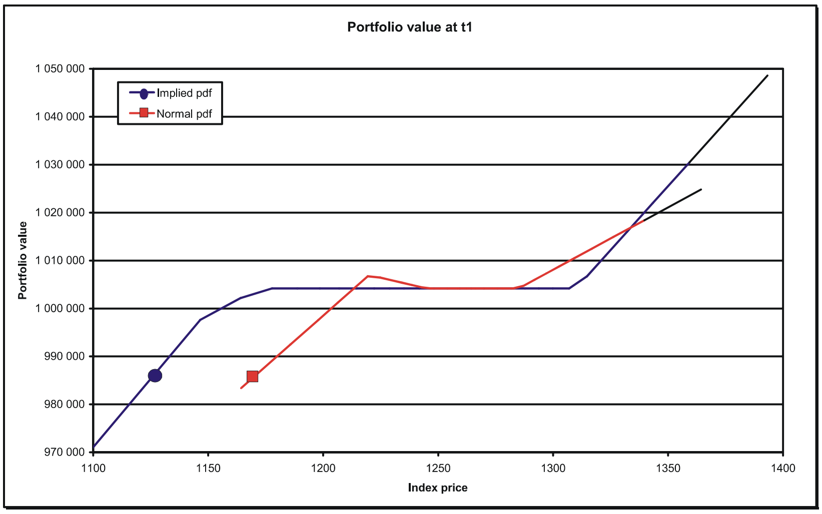


Figure 10.4: Initial portfolio payoffs

In order to check that both solutions would lead to different final returns, a last test could be to start the optimization of the VaR problem under one pdf, but by initially imposing the optimal initial portfolio obtained for the other pdf. Unfortunately, this approach is usually not possible. Indeed, two problems appear. First, due to the cleaning process, the same sets of options could not be available for both pdfs. Secondly, as the market option prices are not equal to the BS option prices, the initial investment to perform to construct the portfolio could not satisfy the budget constraint. This problem indeed appears for the problem considered in this chapter.

10.5 Conclusions

In Section 10.2 of this chapter, we define a specific VaR problem, and the underlying models. The index returns are modelled by the implied pdf. But, for theoretical reasons, we also consider the Normal case. In Section 10.3, we precise the computer environment, and especially we present the software written for the purpose. In Section 10.4, we discuss the results of several numerical experiments. For both the Normal distribution and the implied one, we have successively analyzed the required number of scenarios in the tree and the quality of the results, the computation time and the branch and bound gap, the specific structure of the optimal portfolio, the heuristic performances, the use of different number of options in each period and the index presence, the performances of the dynamic two-period tree versus the static one-period tree, the consequences of considering inconsistent option prices, and what happens when we consider another VaR probability.

Is it dangerous to draw general conclusions from a specific example. Moreover, as noticed in particular in Section 10.4.3, for the S&P500 in February, a bullish strategy seems clearly adequate. More numerical experiments should be made to consider all the possible cases (bullish, bearish, volatility, butterfly) with different parameters to describe the financial market and the investor's goals. Therefore, in the sequel, the general comments are made with respect to the VaR problem considered in this chapter. However, we expect that most of the results presented here remain valid even for other financial data.

It is difficult to compare the results obtained with the Normal pdf and with the implied pdf. Indeed, we observe, in Sections 10.4.1-10.4.2, that the time spent in the optimization process does not only depend on the size of the problem, but also strongly on the input values. For this VaR problem and the normal distribution, the optimization process cannot return results with high precision in a reasonable computation time. Moreover, as shown in Section 10.4.7, it is not possible to use the same option sets in both representations of the future. We must be coherent and use the Black and Scholes option prices when using the Normal distribution to model the future index returns. Note that, even if we could use identical data sets, the optimal results would probably be different. Indeed, in Section 5.5.4, the implied distribution, based on the observed option prices, and the Normal distribution have a very different shape. Finally, if we even assume that using different sets of options make few differences, the optimal return obtained under normality assumptions (6.96%) is lower than the optimal return obtained under the implied pdf (7.56%). Therefore, for all the reasons mentionned in this paragraph and the ones already formulated in Chapter 5, we consider that the implied pdf should be used to model the future.

Another interesting result is that 30 scenarios per subtree suffice to model the future. For this sample size and index returns obtained by the stratification of the continuous pdf, the tree of scenarios accurately represents the continuous index distribution. Moreover, as the sample size is small, the VaR problem size is also reduced. We can even afford to complete the market with nbS^2 options when it is useful. More important, the optimal portfolio returns appear stable, i.e. when we increase the number of scenarios, the returns remain in the same range, and the constraints are perfectly satisfied. More scenarios do not lead to improvements for this VaR problem, while has the drawback to increase, exponentially, time required to obtain an optimal solution. Note also that option pricing model (OP2) cannot be used for sample sizes larger under the Normal distribution.

In Section 10.4.3, the structure of the optimal portfolio is described. As for all feasible portfolios and options with maturity at $t2$, the payoff is a piecewise linear function of the index value with breakpoints at the option strike prices. This property was already stated, and used to define the improved version of the guarantee constraint in Section 7.3.10. Moreover, in each subtree, i.e. for each adjusted portfolios, the portfolio payoff is an increasing function of the index value. This is the definition we have given to a bullish portfolio in Section 8.4. In each subtree, the number of breakpoints in the portfolio payoff is small. It implies that few options are required in the optimal portfolio. We also observe that, starting from the guaranteed level, the portfolio payoff shifts to the VaR level, and finally increases linearly with the index value. The payoff pattern described in this paragraph is typical in several VaR problems (see also Dert and Oldenkamp [20]). Moreover, due to the possibility to adjust the portfolio at $t1$, the increasing portfolio payoff pattern is "split" into the subtrees. Therefore, in some substrees, the optimal portfolio has a constant payoff, and consists only of the risk-free rate. This is what we call a "block structure".

The typical portfolio payoff pattern observed made us expect the possibility to improve the optimization process by using heuristics. Models M1(Rounding & BB,f%), based on operations research techniques, work very well. They quickly find solutions close to the best solutions found by the complete optimization process. Heuristics based on trading strategies and on Dybvig's theorem are more disappointing. For this VaR portoflio problem, the bullish strategy should return very good results. However, when applied to the final leaves of the tree, this is not the case. This may be due to the possibility to adjust the portfolio at the root of each second-period subtree and to the overlapping of the subtrees. We need to use the general model with local optimization in each subtree to obtain the expected portfolio returns. Unfortunately, this is a MIP problem that requires more time to be solved. Heuristics based on Dybvig's theorem provide even worst results for this specific

VaR problem. The violations of Dybvig's hypotheses are too large.

Finally, when we compare the dynamic two-period tree model to a one-period tree model, we observe different behaviours. Firstly, as already stated for a two-period tree, the structure of the portfolio is typical, and especially the apparition of blocks. Secondly, introducing options at the beginning of the second period leads to higher returns. Moreover, considering such options is realistic and so very desirable. Thirdly, more than higher returns, this dynamic two-period model gives more freedom to construct the portfolio, especially, one more time, thanks to the different sets of options at $t1$. Therefore, the constraints seems more light, and portfolio payoffs, which are inaccessible in one-period tree models, can be achieved.

Chapter 11

Conclusions

11.1 Introduction

Let us now draw some conclusions. We will not repeat here in details the conclusions given in each of the four main parts of this work : Chapter 2 ("Simulated Annealing for a generalized mean-variance model"), Chapter 5 ("Modelling the future"), Chapter 6 ("Modelling option prices") and Chapters 7-10 ("Modelling VaR problems"). Instead, we will first briefly recall the complete method we suggest in order to handle portfolio selection problems. Then, we summarize the main results. And finally, we suggest some possible future developments.

11.2 Handling portfolio selection problems

Let us briefly recall the procedure we suggest to follow before solving a portfolio problem like M1. Note that the models described in Chapter 5, "Modelling the future", and in Chapter 6, "Modelling option prices" are not specific to the VaR model and could be used in a larger framework.

We propose to represent the future evolution of financial markets by constructing a two-period tree of scenarios. This model allows to represent at each of its node a possible future state of the world, that we call a scenario.

In order to instantiate such a tree to model the future stochastic returns of a given security, e.g. a stock or an index, we work with its consensus probability density function (pdf). Unfortunately, the pdf can only be approximated. A classical assumption in finance is to consider that the returns are normally distributed. We propose two alternative models: a skewed Student t pdf and an implied pdf. The latter is computed from the option prices observed at $t0$. This model is preferred since the option prices reflect, at $t0$, the future stock returns as expected by the market. Note that to define such a pdf, it is already required to tune the parameters used in its formulation (and which are also used in the VaR problem); e.g. the risk-free rate and the dividend yield. In Chapter 5, we also discuss how to convert a risk-neutral pdf into a consensus one, and conversely. Indeed, the implied pdf is defined in a risk-neutral world while the tree of scenarios models the consensus world. On the other hand, it is also helpful, in particular in our work, to know the risk-neutral pdf since it is used to price the options and to define optimization heuristics.

In order to convert the continuous pdf into a discrete set of values that can be associated to each leaf of the tree, we suggest to use a stratification method instead of a random generator. By this method, even for small sample sizes, the discrete sample represents accurately the continuous pdf.

Once the future index returns have been modelled, we still have to do the same for the option prices. We cannot resort to classical approaches, such as the Black and Scholes formula or the binomial trees, since the hypotheses underlying these models are not satisfied within the multinomial trees of scenarios, and hence the resulting option prices can lead to arbitrage opportunities. Therefore, we propose a new one-period model (OP2), which is based on the no-arbitrage equations. This model can be extended to multiperiod trees, by using a backpropagation technique, and if necessary, a simulated annealing algorithm. Note also that model (OP2) defines a bid price and an ask price for each option, and not a single price as with classical methods.

In order to use the option pricing model (OP2), we first have to define target option prices. Initially, these can be the prices observed on the market. At the beginning of the second period, we suggest to resort to an improved version of the Black and Scholes formula, which takes into account the volatility smile effect. An alternative is to price the options using the state-prices of the first period. Once again, the quality of the option pricing results depends on the quality of the inputs of the model. Realistic procedures to reject mispriced options have been developed.

Using the different models presented in Chapters 5-6, we are now in a position to formulate the portfolio problem. We can construct a two-period tree of scenarios, in which the index and option prices are defined at each node. In the VaR model M1 proposed in Chapter 7, the investor can invest in the index, in the risk-free asset, in options observed initially on the market, or even in options that will only be available at the end of the first period. Moreover, he can adjust his portfolio at $t1$. Transaction costs, bid-ask option spreads and dividend yield are modelled. The investor's objective is to maximize the expected payoff of his portfolio under guarantee and Value-at-Risk constraints.

M1 is a MIP problem which can be solved by the branch and bound optimization method. To speed up the process, several heuristics are proposed. The first ones are based on trading strategies involving options. The next ones use Dybvig's reults relating optimal portfolio payoffs to state price densities. Finally, rounding techniques are developed. These heuristics are first presented and then modelled in Chapters 8-9.

11.3 Main contributions of this thesis

We started this work from a pure operations research point of view, but our aim was to develop models and methods that can be used in practice on financial markets. However, it soon appeared that solving financial optimization models, without first obtaining realistic and consistent financial data, is meaningless. Indeed, the conclusions derived from financial models, like M1, are that results are more sensitive to the validity of their inputs than usual operations research models. Therefore, we have devoted a lot of attention to the application and the integration in our work of advanced financial concepts. This is especially true for the second part of the thesis dealing with VaR models.

As a result, we propose two new optimization models, Markowitz mean-variance model and a Value-at-Risk (VaR) model, which integrate realistic features of portfolio selection problems. Especially, the VaR model contains several interesting features. First, in cotnrast with most portfolio selection models considered in the operations research literature, it explicitely considers the possibility to invest in options. Also, it is based on a measure of risk frequently used by practitioners, namely Value-at-Risk. Moreover, as we know that VaR is an incomplete measure of risk, the approach is improved by considering a second VaR constraint with a total confidence probability: the guarantee constraint. Secondly, a strength of this model is to allow portfolio adjustments at an intermediate time, and to take this possibility into account when constructing the initial optimal portfolio. Thirdly, the model and the algorithms are able to consider all the options available initially on the market, but also to construct realistic sets of options that should appear in the future depending on each scenario at the end of the first period. Finally, realistic features, as transaction costs and bid-ask option spreads, are integrated in the model.

To solve mean-variance portfolio problems, we have developed a new approach based on a simulated annealing algorithm. The originality of this work consists in the application of the Simulated Annealing heuristic, typically used for pure combinatorial problems, to continuous problems. In particular, we have shown how to define effective neighborhoods in this framework.

Before solving financial problems based on future security values, we need to construct a good representation of this future. In order to do it, we have proposed a complete approach for which the result is coherent with information observed on the market. Four steps are required. First, we construct a multinomial multi-period tree of scenarios. This is a flexible tool, more general than classical binomial trees. Secondly, we show how to construct probability density functions. In particular, we suggest to use an implied pdf based on instantaneous market information, viz observed option prices. We have developed a complete and robust algorithm to construct such a pdf. Thirdly, we show how to convert pdfs from the risk-neutral world to the consensus world, and conversely. Representations in both worlds are useful. Finally, the continuous pdf is converted into a discrete set of returns by a stratification method. This results in small data sets which faithfully represent the continuous distribution.

We also propose a new option pricing method valid in a multinomial framework, and which is coherent with observed bid and ask prices. So, we have extended the binomial approach to a more general framework. Moreover, thanks to the models and methods developed to construct representative multinomial trees of scenarios, and the consideration of both the bid and the ask target option prices in the model, this method leads to results close to observed prices.

It is important to notice that the methods developed to model the future and the option prices are not only valid for the VaR model presented here, but for a larger class of financial problems.

Considering the VaR model, we have investigated some advanced methods to solve more quickly the problems than with the branch and bound method, or at least to obtain quickly good feasible solutions. We resort to three set of heuristics. The first two are based on financial properties of the model, and consider trading strategies or a theorem due to Dybvig. The last one is a pure operations research methods based on rounding the solutions of relaxed problems.

As a result, by putting all those models and methods together, we have developed a complete software that can be fully configured. Therefore, we are able to apply advanced theoretical methods, based on financial and operations research concepts, in order to solve realistic portfolio selection problems, and to obtain results that can be applied in practice, since they are coherent with observed market features.

11.4 Future developments

The idea of a research topic should not only yield answers to questions which initiate it, but should also open doors for future developments. This principle certainly applies when we try to model so complex entities such as financial markets. Here are some further improvements that could be considered in relation with the models and the technics presented in our thesis.

The first propositions are relative to the optimization process:

- Depending on the problems to solve, the heuristics already provide quickly good solutions. However, when we examine at the structure of the optimal portfolio, we think it should be possible to improve the quality of the lower bound by exploiting this structure more appropriately.

- It appears also that we do not know how to compute tight upper bounds on the optimal value of model M1. It would be useful to be able to improve the upper bound, so as to refine our estimate and to speed up the branch and bound process. Numerically, classical cuts used in operations research do not provide satisfying improvements. We could consider cuts based on financial properties of the model.

- The heuristics are applied only before the main optimization process. Some improvements could be achieved by considering specific heuristics during the branch and bound process.

A second set of improvements considers the model itself, i.e. the quality of the representation of the real market:

- We consider that applying both a guarantee and the VaR constraint in the model provides a great improvement over a unique VaR constraint. We can go further and add several VaR constraints, corresponding to different levels of returns and different probabilities. The model can be easily modified to take this into account.

- We are not restricted to VaR measures of risk. We could for example also add an upper limit on the portfolio volatility. Integrating a downside volatility measure of risk in the model should be possible whitout loosing its linearity. From a practical point of view, the real difficulty arises from the computation of a valid and realistic upper limit. It is not clear whether the classical mean-variance framework can be extended here.

- We numerically observed large differences between the historical and implied probability density functions. Clearly, the estimation of the parameters for the Normal and skewed Student t pdfs, based on a long period in the past, are too rough. Some weighing schemes should be considered, in particular to put additional wheight on the returns observed in the short past.

- Also, thanks to the flexibility of the tree of scenarios, we could increase the realism of the model by defining different risk-free rates and index volatilities depending on the period. In order to do so, and more generally to enrich the model, we could consider the possibility to include future contracts in the optimization models.

Appendix

Appendix A : list of stocks

List of the stocks considered in the computational experiments.

Code	Firm	Code	Firm
U:ABT	ABBOTT LABS.	U:CLF	CLEVELAND CLIFFS
U:ADM	ARCHER-DANLS.-MIDL.	U:CLX	CLOROX
U:AFL	AFLAC	U:CMB	CHASE MANHATTAN
U:AHM	AHMANSON (H.F.)	U:CMZ	CIN.MILACRON
U:ALK	ALASKA AIR GROUP	U:CQ	COMSAT SR.1
U:APA	APACHE	U:CSC	CMP.SCIENCES
U:AS	ARMCO INCO.	U:CTL	CENTURY TEL.
U:ASA	ASA	U:CTX	CENTEX
U:AVE	AVEMCO	U:CUM	CUMMINS ENGINE
U:AVP	AVON PRODUCTS	U:CVS	CVS
U:AVY	AVERY DENNISON CORP.	U:CYB	CYBEX INTL.
U:AXP	AMER.EXPRESS	U:DCN	DANA CORP.
U:AZ	ATLAS	U:DEC	DIGITAL EQUP.
U:BC	BRUNSWICK	U:DH	DAYTON-HUDSON
U:BDK	BLACK - DECKER	U:DIS	DISNEY (WALT)
U:BK	BANK OF NEW YORK	U:DLX	DELUXE
U:BS	BETHLEHEM STEEL	U:DOW	DOW CHEMICALS
U:CAG	CONAGRA	U:DUK	DUKE POWER
U:CAT	CATERPILLAR	U:DYA	DYNAMICS AMERICA
U:CBE	COOPER INDS.	U:EIX	EDISON INTL.
U:CC	CIRCUIT CITY STORES	U:EK	EASTMAN KODAK
U:CCK	CROWN CORK SEAL	U:ELK	ELCOR
U:CEN	CERIDIAN	U:EMR	EMERSON ELECTRIC
U:CG	COLUMBIA GAS SYS.	U:ESL	ESTERLINE
U:CHV	CHEVRON	U:EY	ETHYL

Code	Firm	Code	Firm
U:F	FORD MOTOR	U:LC	LIBERTY
U:FA	FAIRCHILD	U:LFB	LONGVIEW FIBRE
U:FES	1ST.EMPIRE STATE	U:LGL	LYNCH
U:FJC	FEDDERS	U:LPX	LNA.PACIFIC
U:FKL	FRANKLIN HDG.	U:MAT	MATTEL
U:FWC	FOSTER WHEELER	U:MCD	MCDONALDS
U:GFF	GRIFFON CORP.	U:MEA	MEAD
U:GIS	GEN.MILLS	U:MEL	MELLON BANK
U:GLW	CORNING	U:MER	MERRILL LYNCH
U:GM	GENERAL MOTORS	U:MHP	MCGRAW-HILL CO.
U:GTI	GTI	U:MMG	METROMEDIA INTL.GROUP
U:HAS	HASBRO	U:MO	PHILIP MORRIS
U:HON	HONEYWELL	U:MOB	MOBIL
U:HPC	HERCULES	U:MOT	MOTOROLA
U:HUG	HUGHES SUPPLY	U:MSN	EMERSON RADIO
U:ICI	IMP.CHM.INDS.ADR	U:MST	MERCANTILE STRS.
U:IDA	IDAHO POWER	U:MTS	MONTGOMERY STR.INC.SECS.
U:IIN	ITT INDUSTRIES	U:MUR	MURPHY OIL
U:IP	INTL.PAPER	U:MYE	MYERS INDS.
U:JET	JETRONIC	U:NAE	NORAM ENERGY CORP.
U:JII	JOHNSTON INDUSTRIES	U:NAV	NAVISTAR INTL.
U:JNJ	JOHNSON-JOHNSON	U:NBL	NOBLE AFFILIATES
U:JP	JEFFERSON PILOT	U:NCC	NAT.CITY
U:JPM	MORGAN (JP)	U:NL	NL INDUSTRIES
U:K	KELLOGG	U:NOB	NORWEST
U:KO	COCA COLA	U:NSH	NASHUA
U:KRI	KNIGHT-RIDDER	U:NU	NORTHEAST UTILITIES
U:KSF	QUAKER STATE	U:NUE	NUCOR
U:KUH	KUHLMAN	U:NYTA	NY.TIMES 'A'
U:KZ	KYSOR IND. DEAD	U:OG	OGDEN

Code	Firm	Code	Firm
U:OJ	ORANGE CO.	U:SNT	SONAT
U:OM	OUTBOARD MARINE	U:SO	SOUTHERN
U:OWC	OWENS-CORNING	U:SP	SPELLING ENTM.GP.
U:OXM	OXFORD INDS.	U:SPA	SPARTON
U:PEP	PEPSICO	U:SUN	SUN
U:PET	PACIFIC ENTS.	U:SVT	SERVOTRONICS
U:PG	PROCTER - GAMBLE	U:TA	TRANSAMERICA
U:PHM	PULTE	U:TAN	TANDY
U:PKD	PARKER DRILLING	U:TUR	TURNER
U:PNC	PNC BANK	U:TWX	TIME WARNER
U:PXR	PAXAR CORP.	U:TX	TEXACO
U:RAL	RALSTON PURINA RAL-PUR	U:TYC	TYCO INTERNATIONAL
U:RLM	REYNOLDS METALS	U:UL	UNILEVER ADR.
U:RML	RUSSELL	U:WEC	WISCONSIN ENERGY
U:ROK	ROCKWELL INTL.NEW	U:WTR	AQUARION
U:RTN	RAYTHEON	U:XON	EXXON
U:S	SEARS,ROEBUCK	U:XRX	XEROX
U:SGP	SCHERING-PLOUGH	U:Z	WOOLWORTH
U:SII	SMITH INTL.	U:ZAP	ZAPATA CORP.NEW
U:SKY	SKYLINE	U:ZCO	ZIEGLER CO.
U:SLE	SARA LEE CORP.	U:ZE	ZENITH ELEC.

Appendix B : software parameters

List of the software parameters with a short description.

Parameter	**Value**	**Description**
FINANCIAL DATA		
t1	1	End of the 1st period
t2	2	End of the 2nd period
B	1000000.0	Initial budget
theta	0.9	Guarantee level
lambda	1.01	VaR level
proba	0.95	VaR probability
TREE		
nbS	40	Number of scenarios per subtree
STOCK		
name	SP500	Name of the index to define the strike prices
allowed	off	Index can be in the portfolio
s0	1246.23	Initial index value
mean	0.01131035	Empirical mean index return
autostock	off	Compute index parameters thanks to the parity equations
paritypcmin	0.003	Minimal percentage to take the option into account
paritypcmax	0.08	Maximal percentage to take the option into account
r	0.0042	Risk-free rate
stdDev	0.03658057	Index volatility
q1	0.001058	Dividend yield for the 1st period
q2	0.001058	Dividend yield for the 2nd period
tstock	0.0030	Index cost of transaction (%)
samplemethod	1	pdf to model the future
		< 0=N from random generator,1=N stratification
		2=N from uniform generator,3=Skewed T stratification
		5=implied pdf stratification

Parameter	**Value**	**Description**
impliedconvert	3	Conversion from risk-neutral to consensus implied pdf
		1=shift,2=log,3=power
impliedsampling	400	Sample size for the implied conversion
equiproba	off	The leaves are equiprobable (implied distribution only)
minwilkp	0.00000	Reject the sample if Wilk's test is not satisfied
OPTIONS		
optload1	1	Option generation for the 1st period
		1=load from file "options1.txt"
		2=auto - nbopt1 options - strike prices uniformly distributed
		3=auto - nbS options - index prices at leaves as strike prices
		4=auto - nbopt1 options - market rules
nbopt1	126	Number of options to generate (1st period)
typ1	*	Generate c=call,p=puts,*=calls and puts
optload2	4	As optload1 but for the 2nd period
nbopt2	30	As nbopt1 but for the 2nd period
typ2	*	As typ1 but for the 2nd period
auto2	0	Preselect auto2 promising options in a set of auto2set
auto2set	0	Preselect auto2 promising options in a set of auto2set
target2	1	Option target price for the 2nd period
		0=read from file, 1=BS, 2=state-prices,3=BS(smile)
optload12	4	As optload1 but for both periods
nbopt12	0	As nbopt1 but for both periods
typ12	*	As typ1 but for both periods
auto12	0	As auto2 but for both periods
auto12set	0	As auto2set but for both periods
timeauto	1	Maximal time to spend in the preselection process
autobound	on	Compute heuristic bounds for the preselection BB
spread	0.02	Bid-ask percentage spread (if not read from file)
topt	0.0030	Option transaction cost (%)
spreadmin	0.0	Minimal spread (absolute value)
transmin	2.0	Minimal cost of transaction per 100 options
transmax	20.0	Maximal cost of transaction per 100 options
optvalmin	0.10	Option is valueless if price smaller than optvalmin
optregresmin	2.0	Minimal price of options used to construct the smile

optcompute	3	Option pricing method
		1=BS, 2=linear problem with squared objective function
		3=linear problem with absolute objective function,4=3+SA
optabsolu	off	Absolute or relative term in the objective function
optproba	2	1=OP1 (without state-prices), 2=OP2 (with state-prices)
optclean	1	If an option has no implied volatility
		0=do nothing, 1=delete,2=price:=BS,3=price:=BS(smile)
timeheuristic	60	Maximal time to spend in the SA (optcompute=3)
tolheuristic	0.1	Precision for the SA algo(optcompute=3)
autostrike	strikep.txt	File containing the market rules for different indices
BOUNDS		
bound	off	Compute one or several lower bounds
boundDybvigequi	on	Use Dybvig's theorem and equirepartition
boundDybvigsp	on	Use Dybvig's theorem and final state-prices
boundDybvigs2	on	Use Dybvig's theorem and final index values
boundtime	4	Maximal time to spend in each of the previous method
reducedBB	off	Use a BB process after relaxing some variables
reducedBBnb	100	Number of binary variables to set in the BB process
reducedBBgap	0.02000	Requested precision for this BB process
reduceBBtime	2	Maximal time to spend in this BB process
stratbull	off	Use a bullish heuristic
stratbear	off	Use a bearish heuristic
stratvol	off	Use a volatility heuristic
strat3gap	0.02	Requested precision for this BB process
strat3time	2	Maximal time to spend in this BB process
stratstab	off	Use a stability heuristic
stabgap	0.02	Requested precision for this BB process
stabtime	2	Maximal time to spend in this BB process
DybvigM2	off	Use Dybvig's theorem and BB over model 2
DybvigM1	off	Use Dybvig's theorem and BB over model 1
DybvigMxgap	0.020000	Requested precision for this BB process
DybvigMxtime	2	Maximal time to spend in this BB process

VaR OPTIMIZATION (CPLEX)		
portfolio	on	Compute the VaR portfolio and not only price options
model	1	VaR model 1 or 2
enableoptimt2	on	Improved guarantee constraints for maturity at t2
strongoptim	on	Strong guarantee constraints
cuts	off	Allow CPlex to try some cuts during the BB process
rootheuristic	-1	CPlex pre-heuristic (-1=off,0=auto,1=on)
heuristic	on	CPlex heuristic during the BB process
aggregator	off	CPlex aggregator
presolver	off	CPlex presolver
coeffreduce	on	CPlex coefficient reduction
gap	0.0200000	Stop if (best integer sol - best sol)/(best integer sol) < gap
treeRAM	64.0	Maximal physical memory CPlex can use
priorityDybvig	off	Branch according to Dybvig's theorem
time	10	Maximal time spent in the VaR BB process
backtrack	1.0	CPlex backtracking ([0,1],0=jump,1=deep)
OTHER PARAMETERS		
tofile	on	Save detailed results in separate files
rebalance	on	The investor can adjust the portfolio at t1
sizecheck	50	Post-processing:
		Check the constraints on a sample test of size nbS^2*sizecheck

Bibliography

[1] Aarts E., Lenstra J.K., *Local search in combinatorial optimization*, John Wiley & Sons, 1997.

[2] Bienstock D., *Computational study of a family of mixed-integer quadratic programming problems*, Mathematical Programming, 74 (1996), pp. 121-140.

[3] Birge J.R., Louveaux F., *Introduction to stochastic programming* (2nd edition), Springer Series in Operations Research, Springer-Verlag New York, 1999.

[4] Black F., Scholes M. *The pricing of options and corporate liabilities*, Journal of Political Economy, vol 81, May-June 1973, pp. 637-654.

[5] Breeden D.T., Litzenberger R.H. *Prices of state-contingent claims implicit in option prices*, Journal of Business, vol 51, No4, 1978.

[6] Catanas F., *On a neighbourhood structure for portfolio selection problems*, Working paper, Centre for Quantitative Finance, Imperial College, London, 1998.

[7] Černý V., *Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm*, Journal of Optimization Theory and Applications, 45 (1985) pp. 41-51.

[8] Chang T.J., Meade N., Beasley J.E. and Sharaiha Y.M., *Heuristics for cardinality constrained portfolio optimisation*, Computers & Operations Research, 27 (2000) pp. 1271-1302.

[9] Clewlow L., Strickland C., *Implementing derivatives models*, Wiley, 1998.

[10] Computational Science Education Project, *Mathematical optimization*, Electronic book at http://csep1.phy.ornl.gov/mo/mo.html, 1995.

[11] Constantinides G.M., Malliaris A.G.,*Portfolio theory*, in: *Finance*, R.A. Jarrow, V. Maksimovic and W.T. Ziemba (eds.), Elsevier Science, Amsterdam, 1995, pp. 1-30.

[12] Cox J.C., Rubinstein M., *Options markets*, Englewood Cliffs, Prentice Hall, 1985.

[13] Curran M., *Strata Gems*, Risk, March 1994, 1985, pp70-71.

[14] Dahl H., Meeraus A., Zenios S.A., *Some financial optimization models: I Risk management*, in: *Financial Optimization*, S.A. Zenios (ed.), Cambridge University Press, 1996, pp. 3-36.

[15] Dekkers A., Aarts E., *Global optimization and simulated annealing*, Mathematical Programming, 50 (1991) pp. 367-393.

[16] Dembo R.S., *Scenario optimization*, Annals of Operations Research, 30 (1991) pp. 63-80.

[17] Dembo R.S., Freeman A., *Seeing tomorrow: Rewriting the rules of risk*, Wiley, 1998.

[18] Dembo R.S., Mulvey J.M., Zenios S.A., *Large-scale nonlinear network models and their application*, Operations Research, 37 (1989) pp. 353-372.

[19] Dert C.L., *Asset liability management for pension funds: a multistage chance constrained programming approach*, PhD Thesis, Erasmus University, Rotterdam, The Netherlands.

[20] Dert C.L., Oldenkamp B., *Optimal guaranteed return portfolios and the casino effect*, Operations Research, 2000, September-October, Vol 48, pp. 768-775.

[21] Dothan M.U., *Prices in financial markets*, Oxford University Press, 1990.

[22] Dybvig P.H., *Inefficient dynamic portfolio strategies or how to throw away a million dollars in the stock market*, The Review of Financial Studies, Spring, 1988, Vol 1, pp67-88.

[23] Dybvig P.H., *Distributional analysis of portfolio choice*, Journal of Business, 1988, Vol 61, pp369-393.

[24] Duffie D., *Dynamic asset pricing theory*, Princeton University Press, Second Edition, 1996.

[25] Edgeworth, F.Y., *The mathematical theory of banking*, Journal of the Royal Statistical Society, March 1888, 51/1, pp113-127.

[26] Elton E.J., Gruber M.J., *Modern portfolio theory and investment analysis* (4th edition), John Wiley, New York-London, 1991.

[27] Esch L., Kieffer R., Lopez T., *Value at Risk: vers un Risk Management moderne*, De Boeck & Larcier, 1997.

[28] Fernandez C., Steel F.J. *On Bayesian modeling of fat tails and skewness*, Journal of the American Statistical Association, vol 93, No441, March, 1998.

[29] Gielen M.D., *A tailor-made portfolio: a two period optimization model for portfolio selection*, Thesis, University of Maastricht, 1998.

[30] Hampel F.R., Ronchetti E.M., Rousseeuw P.J., Werner A.S., *Robust statistics: the approach based on influence functions*, Wiley, 1986.

[31] Hamza F., Janssen J., *Extension réaliste pour la sélection de portefeuilles en présence des coûts de transaction*, Working paper, Université Libre de Bruxelles, 1997.

[32] Hull J.C., *Options, futures and other derivatives*, Prentice Hall, Third Edition, 1997.

[33] Ilog, *CPlex: Reference manual*, Ilog, March 1999.

[34] Ingber L., Wehner M.F., Jabbour G.M., Barnhill T.M., *Application of statistical mechanics methodology to term structure bond-pricing models*, Mathematical and Computer Modelling, 15 (1991) p. 77.

[35] Johnson D.S., Aragon C.R., McGeoch L.A., Scevon C., *Optimization by simulated annealing: An experimental evaluation. Part I, Graph partitioning*, Operations Research, 37 (1989) pp. 865-892.

[36] Jouini E., Kallal H., *Efficient trading strategies in the presence of market frictions*, Review of Financial Studies, 14 (2001), pp343-369.

[37] Kirkpatrick S., Gelatt C.D., Vecchi P.M., *Optimization by simulated annealing*, Science, 220 (1983) pp. 671-680.

[38] Kleijnen, *Statistical techniques in simulation: part I*, Marcel Dekker Inc., 1974.

[39] Kleijnen, *Statistical techniques in simulation: part II*, Marcel Dekker Inc., 1975.

[40] Konno H., Yamazaki H., *Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market*, Management Science, 37 (1991) pp. 519-531.

[41] Koskosidis Y.A., Duarte A.M., *A scenario-based approach to active asset allocation*, The Journal of Portfolio Management, Winter 1997.

[42] Lambert P., Laurent S., *Modelling financial time series using GARCH-type models and a Skewed Student density*, draft, January, 2001.

[43] Leibowitz M.L., Henriksson R.D., *Portfolio optimization with shortfall constraints: A confidence-limit approach to managing downside risk*, Financial Analysts Journal, March-April, 1989, pp34-41.

[44] Leibowitz M.L., Kogelman S., *Asset allocation under shortfall constraints*, The Journal of Portfolio Management, Winter, 1991, pp18-23.

[45] Levy H., Sarnat M., *Portfolio and investment selection: theory and practice*, Prentice-Hall International, 1984.

[46] Loraschi A., Tettamanzi A., Tomassini M., Verda P., *Distributed genetic algorithms with an application to portfolio selection problems*, in: *Proc. Int. Conf. on Artificial Neural Networks and Genetic Algorithms ICANNGA95*, D.W. Pearson, N.C. Steele and R.F. Albrecht (eds.), Springer-Verlag, 1995, pp. 384-387.

[47] Lovász L., Simonovits M., *Random walks in a convex body and an improved volume algorithm*, Random Structures and Algorithms, 4 (1993) pp. 359-412.

[48] Luenberger D.G., *Investment management*, Oxford University Press, 1998.

[49] Lynch M.R. and al., *The options toolbox v5.0*, Electronic book, DerivaTech, CBOE, http://www.cboe.com, 2001.

[50] Markowitz H.M., *Portfolio selection*, Journal of Finance, 7 (1952) pp. 77-91.

[51] McMillan L.G., *Options as a strategic investment*, New York Institute of Finance, 1993.

[52] Gillet R., Minguet A., *Micro-structure et rénovation des marchés financiers en Europe*, Presses Universitaires de France, 1995.

[53] Mulvey J.M., *Financial planning via multi-stage stochastic programs*, in: *Mathematical Programming:State of the Art 1994*, Edited by Birge J.R. and Murty K.G., The University of Michigan, 1994, pp151-171.

[54] Nakasato M., Furukawa K., *On the number of securities which constitute an efficient portfolio*, Annals of Operations Research, 45 (1993) pp. 333-347.

[55] Neftci S.N., *An introduction to the mathematics of financial derivatives*, Academic Press, 1996.

[56] Nemhauser G.L., Wolsey L.A., *Integer and combinatorial optimization*, Wiley, 1988.

[57] Osman I.H., Laporte G., *Metaheuristics: A bibliography*, Annals of Operations Research, 63 (1996) pp. 513-623.

[58] Perold A.F., *Large-scale portfolio optimization*, Management Science, 30 (1984) pp. 1143-1160.

[59] Pirlot M., *General local search heuristics in combinatorial optimization: A tutorial*, Belgian Journal of Operations Research, Statistics and Computer Science, 32 (1992) pp. 7-68.

[60] Prekopa A., *Stochastic programming*, Kluwer Academic Publishers, 1995.

[61] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., *Numerical recipes in C*, Cambridge University Press, Second Edition, 1992.

[62] Morgan J.P., *RiskMetrics - Technical document, 4th ed.*, Morgan Guaranty Trust Company, http://www.riskmetrics.com/research, New York, 1996.

[63] Rubinstein M., *Implied binomial trees*, The Journal of Finance, vol LXIX, n3, July, 1994.

[64] Shimko D., *Bounds of probability*, Risk, vol 6, n4, April, 1993.

[65] Takehara H., *An interior point algorithm for large scale portfolio optimization*, Annals of Operations Research, 45 (1993) pp. 373-386.

[66] Theodossiou, *Financial data and the skewed generalized T distribution*, Management Science, vol 44, n12, Part 1 of 2, December 1998.

[67] van Laarhoven P.J.M, Aarts E.H., *Simulated annealing: Theory and applications*, Kluwer Academic Publishers, 1988.

[68] Willmott P., Howison S., Dewynne J. *The mathematics of financial derivatives*, Cambridge University Press, 1996.

[69] Winston W.L. *Operations research: Applications and algorithms*, International Thomson Publishing, Duxbury Press, 3rd edition, 1993.

[70] Wolfe P., *The simplex method for quadratic programming*, Econometrica, 27 (1959) pp. 382-398.

[71] Zabinsky Z.B., Smith R.L., McDonald J.F., Romeijn H.E., Kaufman D.E., *Improving hit-and-run for global optimization*, Journal of Global Optimization, 3 (1993) 171-192.

Michaël Schyns
Business School, University of Liège
7, Boulevard du Rectorat, B-4000 Liège
Email: M.Schyns@ulg.ac.be

Members of the Jury

- Professor A. Corhay, University of Liège, president of the Jury
- Professor Y. Crama, University of Liège, member of the thesis committee and promoter
- Professor W.G. Hallerbach, Erasmus University, Rotterdam, The Netherlands
- Professor G. Hübner, University of Liège, member of the thesis committee
- Professor A. Kolen, University of Maastricht, The Netherlands,
  member of the thesis committee
- Professor M. Roubens, University of Liège

September 22, 2001
Liège.