

Time and nodal decomposition with implicit non-anticipativity constraints in dynamic portfolio optimization

Diana Barro^{1,2} and Elio Canestrelli¹

¹ Department of Applied Mathematics, University Ca' Foscari of Venice,
Dorsoduro 3825/E, 30123 Venice, Italy
{d.barro, canestre}@unive.it
<http://www.dma.unive.it>

² SSAV – University Ca' Foscari of Venice

Received: January, 2006 – Accepted: November 2006

Abstract. We propose a decomposition method for the solution of a dynamic portfolio optimization problem formulated as a multistage stochastic program. The method leads to the problem time and nodal decomposition in its arborescent formulation applying a discrete version of the Pontryagin Maximum Principle. The solution of the decomposed problems is coordinated through a weighted fixed-point iterative scheme. The introduction of an optimization step in the weights choice at each iteration leads to a very efficient solution algorithm.

Keywords. Portfolio optimization, stochastic programming, decomposition, fixed-point, maximum principle.

M.S.C. classification. 65K05, 90C15, 90C90.

J.E.L. classification. C61, C69, G11.

1 Introduction

The stochastic programming approach to financial optimization, and dynamic portfolio management in particular, is well documented in the literature, see, for example, [3], [13], [14], [15], [16], [23], [24], and [46]. For a collection of stochastic programming applications in many different fields and a survey of publicly available stochastic programming codes see [45].

In this contribution we propose a solution approach for a dynamic portfolio optimization problem formulated as a sequential decision problem under uncertainty in discrete time, in the framework of multistage stochastic programming.

We assume a discrete probability distribution for the vector stochastic process underlying the decision problem, whose dynamics can then be conveniently described using a scenario tree structure. These assumptions together with the

specification of the non-anticipativity constraints – in explicit or implicit form – allow us to recast the problem into a large-scale deterministic equivalent optimization problem. As further discussed in the sequel the large scale problem needs not be linear and is indeed well suited for the application of a decomposition solution algorithm applied to general nonlinear problems.

The scheme exploits the main features of the stochastic programming formulation and the discrete time version of the Pontryagin Maximum Principle, and overcomes the difficulties associated with rapidly increasing problem dimensions. The approach allows a solution method which applies to the deterministic equivalent problem written in arborescent form, based on a time decomposition and a further nodal decomposition of the problem.

In section 2 we briefly discuss the non-anticipativity property of the decision process and the formulation of the non-anticipativity constraints either in implicit or explicit form. Section 3 reviews the main solution approaches proposed in the literature. In section 4 we present the dynamic portfolio problem. In section 5 we describe our solution method. In section 6 we provide some computational results comparing the proposed approach with the time decomposition in the case of explicit non-anticipativity constraints and with the direct solution of the deterministic equivalent problem. Moreover we present some results on dynamic investment experiments using data from the Italian market and applying the proposed solution method. Section 7 concludes.

2 Implicit versus explicit non-anticipativity constraints

Scenario trees provide a common method to characterize uncertainty in a stochastic optimization problem. This can be done assuming that the probability distribution P of the random quantities affecting the problem solution can be described or approximated by a discrete distribution with a finite number of possible outcomes.

The discrete distribution can be represented through an event tree with nodes associated with the realizations of the stochastic quantities. The structure of the event tree is related to information arrival process. In a multistage framework information is revealed through time and at each stage the decision process can depend on decisions made at previous stages and realizations of the stochastic quantities up to that point, but cannot anticipate future outcomes. We say that the decision process is *measurable* with respect to the current *sigma-algebra* and all decisions are made in the face of residual uncertainty.

The introduction of an event tree to describe uncertainty, that is a finite number of possible outcomes at each stage t , allows us to formulate the deterministic equivalent problem which can have implicit or explicit non-anticipativity constraints (see, for example, [7]). The resulting problem is characterized by large dimensions and a block diagonal structure of the constraint matrix. We obtain a sparse large-scale optimization problem ideal for the application of a decomposition solution method.

The definition of the non-anticipativity constraint is central to the formulation of multistage stochastic programming problems and the choice of the most suitable solution approach. Non-anticipativity can be forced in explicit or implicit form.

In the case of explicit constraints we split the event tree path-wise and the decision process – every node has an associated local decision problem – follows the scenario evolution. The decision hierarchy is in this case forced along every scenario consistently with the original tree structure. The procedure leads to S dynamic problems, where S represents the number of scenarios, characterized by the same time structure where each scenario describes a unique path from the root of the tree to the leaf node. Non-anticipativity constraints are thus added explicitly to ensure feasibility of the decisions with respect to the set of information constraints.

In the case of implicit constraints the property of non-anticipativity is automatically fulfilled by introducing a unique vector of decision variables for each node of the tree making sure that the random coefficients of the problem are properly associated.

Both in the cases of explicit and implicit non-anticipativity constraints, a deterministic optimization problem is obtained from our multistage portfolio optimization model that can be tackled in the framework of discrete time optimal control problem with mixed constraints.

In the first case, already analyzed in [1], we relaxed the non-anticipativity constraints obtaining separability with respect to scenarios and then solved each scenario problem as a deterministic discrete time optimal control problem, see [11] and [12].

In this contribution we propose a new decomposition method, based on discrete time optimal control applied to the nodal formulation of the problem with implicit non-anticipativity constraints, which leads to time and nodal decomposition.

3 Solution approaches and decomposition methods

As widely pointed out in the literature (see, for example, [27], [38], [42], and [44]), the dimension of the deterministic equivalent problem, obtained from real applications, is frequently too large to be tractable by direct solution.

These problems present special structures which can be approached with solution methods based on decomposition. According to the literature see, for example, the review in [5], solution approaches for multistage stochastic programming problems can be broadly classified into two main groups.

In the first group we include general purpose algorithms specialized to improve the data structures and the solution strategies according to the problem features, i.e. sparsity or block diagonal structure of the coefficient matrix. Among these approaches we may cite [6], [8], [20], and [25].

In the second group we consider decomposition approaches which take advantage of the stochastic program structure aiming at reducing the original problem into a collection of small and easy-to-solve sub-problems.

Within decomposition methods we can also distinguish between methods that result in a nodal decomposition and methods that produce a scenario decomposition of the original problem.

In the first case the original problem is decomposed into a collection of subproblems each related to a node in the event tree see, for example, [4], [9], [17], [21], [39], [41], and [43].

In the second case each subproblem corresponds to a scenario and the original problem is decomposed according to the stochastic component, see, for example, [28], [29], [31], [36], and [40].

In [38] the authors propose an augmented lagrangian decomposition method that can be applied to obtain either a decomposition according to stages or a decomposition according to scenarios.

For decomposability features in the framework of large-scale linear-quadratic programming and discrete time optimal control problems see [34], [35], and [37].

For a review of decomposition methods and for more extensive references on solution approaches see [5] and [42].

We propose here a decomposition method for the solution of a dynamic portfolio optimization problem formulated as a multistage stochastic programming problem. The proposed method, that combines the main features of the stochastic programming formulation and a discrete version of the Pontryagin Maximum Principle, to obtain a time and nodal decomposition of the original problem.

The method allows nonlinear objective functions which arise in portfolio theory due to risk-averse investors and to exploit the time-decomposability feature provided by discrete time optimal control problems.

The motivation for the development of this method arises from the portfolio management problem but the formulation is quite general and can be adapted to a broader class of problems dealing with planning under uncertainty where the dynamics are linear and the objective function is additive in time.

4 The portfolio model

We consider a dynamic portfolio optimization problem over a finite horizon $[0, T]$. Key features of the model from a financial viewpoint are the explicit inclusion the transaction costs and a risk averse utility function for the investor. The model includes also restrictions on short-selling and borrowing.

For a review of discrete time dynamic portfolio management models see [14], [22], [30], and [47].

Uncertainty is modeled by a discrete time, discrete state stochastic process represented by an event tree. We assume a general structure for the tree. At time $t = 1, \dots, T$ there are $K_t - K_{t-1}$ nodes denoted by $k_t = K_{t-1} + 1, \dots, K_t$, while at time $t = 0$ there is a root node denoted by $k_0 \equiv K_0$ from which the tree

originates. Let $b(k_t)$ be the ancestor of node k_t in the previous period, where $b(k_1) = k_0$, and let $d(k_t) = 1, \dots, D(k_t)$ be the descendants from node k_t in the subsequent period. At the planning horizon there are $S = K_T - K_{T-1}$ leaf nodes. Each path connecting the root node with a leaf node is a *scenario*, i.e. a sequence of possible realizations. Therefore S is the number of scenarios which corresponds to the number of leaves of the tree. The probability of each scenario is denoted by π_{k_T} , with $\pi_{k_T} > 0$ and $\sum_{k_T=K_{T-1}+1}^{K_T} \pi_{k_T} = 1$.

At the initial time $t = 0$ the prices of the risky assets are known while prices and returns at future dates are described by a stochastic vector process. At each trading date, conditionally on previous information, the distribution of prices and returns of risky assets is described by a finite number of realizations which correspond to the time- t , $K_t - K_{t-1}$ nodes.

The model includes purchase and sale variables for each risky asset and a liquidity component (see, for example, [10]). We denote with $I = \{1, \dots, n+1\}$ the set of assets among which we can choose the composition of the portfolio; there are n risky assets and a liquidity component, denoted by $n+1$. We denote with $x_{k_t} = (x_{1k_t}, \dots, x_{n+1k_t}) \in \mathbb{R}^{n+1}$ the vector of the amounts of each asset held in node k_t , with $a_{k_t} = (a_{1k_t}, \dots, a_{nk_t}) \in \mathbb{R}^n$ the vector of the amounts of each risky asset purchased in node k_t , and with $v_{k_t} = (v_{1k_t}, \dots, v_{nk_t}) \in \mathbb{R}^n$ the vector of the amounts of each risky asset sold in node k_t . The transaction costs, cta and ctv , are expressed as a percentage of the amount of purchased and sold assets; moreover we set $d^+ = (1 + cta)$ and $d^- = (1 - ctv)$. r_{k_t} denotes the return rate on the liquidity component of the portfolio in the period $[t-1, t]$; $\rho_{k_t} = (\rho_{1k_t}, \dots, \rho_{nk_t})$ is the vector of return rates on the risky assets in the period $[t-1, t]$ moving from node $b(k_t)$ to node k_t . $U(\cdot)$ denotes a risk averse utility function.

We denote with $\hat{K} = \{k_t : k_t = K_{t-1} + 1, \dots, K_t, t = 1, \dots, T\}$ the set of nodes in the tree from time $t = 1$ to $t = T$ and with $\tilde{K} = \{K_0\} \cup \{k_t : k_t = K_{t-1} + 1, \dots, K_t, t = 1, \dots, T-1\}$ the set of nodes from time $t = 0$ to time $t = T-1$.

The deterministic equivalent model with implicit non-anticipativity constraints in arborescent form is

$$\max_{\substack{x_{k_t}, k_t \in \hat{K} \\ a_{k_t}, k_t \in \tilde{K} \\ v_{k_t}, k_t \in \tilde{K}}} \sum_{k_T=K_{T-1}+1}^{K_T} \pi_{k_T} U(1'x_{k_T}) \quad (1)$$

s.t.

$$x_{i k_t} = (1 + \rho_{i k_t})[x_{i b(k_t)} + a_{i b(k_t)} - v_{i b(k_t)}] \quad k_t \in \hat{K}, i = 1, \dots, n \quad (2)$$

$$x_{n+1 k_t} = (1 + r_{k_t})[x_{n+1 b(k_t)} - d^+ 1' a_{b(k_t)} + d^- 1' v_{b(k_t)}] \quad k_t \in \hat{K} \quad (3)$$

$$x_{k_0} = \bar{x} \quad (4)$$

$$a_{k_t} \geq 0 \quad k_t \in \tilde{K} \quad (5)$$

$$v_{k_t} \geq 0 \quad k_t \in \tilde{K} \quad (6)$$

$$x_{k_t} \geq 0 \quad k_t \in \hat{K}. \quad (7)$$

where $1'$ denotes a row vector of proper dimension with all components equal to one. The wealth in each node is given by the value of the portfolio and x_{k_0} denotes the initial endowment. The objective (1) is to maximize the expected utility of final wealth. Constraints (2) and (3) represent the asset inventory constraints and the cash balance equations, respectively. Borrowing and short selling are not allowed.

5 Decomposition of the problem

We follow the convention that variables a_{k_t} , v_{k_t} and x_{k_t} are determined in node k_t at time $t = 1, \dots, T - 1$ according to the following scheme.

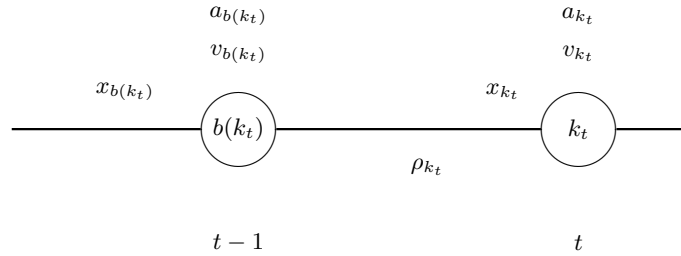


Fig. 1. Decision variables for $t = 1, \dots, T - 1$.

where $b(k_1) = k_0$ denotes the root node and $x_{b(k_1)} = x_{k_0} = \bar{x}$ is the initial endowment. The variables x_{k_t} , $k_t \in \hat{K}$ denote amounts available at the end of the period associated with returns that mature in the period $[t - 1, t]$. For $t = T$, i.e., the leaf nodes in the tree, no investments and dis-investments are allowed, and only the variables x_{k_T} appear to denote the final composition of the portfolio:

In equation (2) and (3) we can recognize an implicit dynamics from time $(t - 1)$ to time t . Exploiting this feature we can write problem (1)-(7) as a discrete time optimal control problem where x_{k_t} represent the state variables and a_{k_t} and v_{k_t} the controls. To this aim we reformulate the problem in order to avoid the non negativity constraints on the state variables. Taking into account equations (2) and (3), the non negativity constraints (7) become

$$x_{i b(k_t)} + a_{i b(k_t)} - v_{i b(k_t)} \geq 0 \quad i = 1, \dots, n \quad (8)$$

$$x_{n+1 b(k)} - d^+ 1' a_{b(k_t)} + d^- 1' v_{b(k_t)} \geq 0. \quad (9)$$

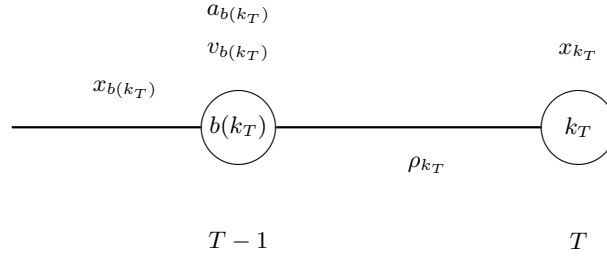


Fig. 2. Decision variables for $t = T$.

5.1 Time and nodal decomposition

Problem (1)-(6) together with (8)-(9) can be written as a discrete time optimal control problem, with mixed constraints, where the dimensions of the state and control variables vary with time (see [35]).

Let $x(t) = (x_{K_{t-1}+1}, \dots, x_{K_t})$ be the vector of state variables at time t , with $t = 1, \dots, T$ and $u(t) = (u_{K_{t-1}+1}, \dots, u_{K_t})$ be the vector of control variables at time t , with $t = 1, \dots, T-1$ and for $t = 0$ $u(0) = (u_{k_0})$. The discrete time optimal control problem is

$$\max_{u(0), \dots, u(T-1)} \{L_T(x(T))\} \quad (10)$$

$$x(t+1) = A(t)x(t) + B(t)u(t) + q(t) \quad (11)$$

$$x(0) = x_0 \quad (12)$$

$$G(t)x(t) + H(t)u(t) + r(t) \leq 0 \quad (13)$$

$$u(t) \geq 0 \quad (14)$$

$$t = 0, \dots, T-1$$

where the matrices in (11) and (13) are characterized by time varying dimensions and have block structure as follows

$$A(t) = \begin{pmatrix} \begin{pmatrix} A_1 \\ \vdots \\ A_{D(K_{t-1}+1)} \end{pmatrix}_{K_{t-1}+1} & \dots & 0 \\ \vdots & & \\ 0 & \dots & \begin{pmatrix} A_1 \\ \vdots \\ A_{D(K_t)} \end{pmatrix}_{K_t} \end{pmatrix} \quad (15)$$

$$B(t) = \begin{pmatrix} \begin{pmatrix} B_1 \\ \vdots \\ B_{D(K_{t-1}+1)} \end{pmatrix}_{K_{t-1}+1} & \dots & 0 \\ \vdots & & \\ 0 & \dots & \begin{pmatrix} B_1 \\ \vdots \\ B_{D(K_t)} \end{pmatrix}_{K_t} \end{pmatrix} \quad (16)$$

$$q(t) = 0 \quad r(t) = 0 \quad (17)$$

$$G(t) = \begin{pmatrix} G_{K_{t-1}+1} & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & G_{K_t} \end{pmatrix} \quad H(t) = \begin{pmatrix} H_{K_{t-1}+1} & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & H_{K_t} \end{pmatrix}. \quad (18)$$

The sub-matrices are defined as

$$A_{k_t} = \begin{pmatrix} \text{diag}(1 + \rho_{i k_t}) & 0 \\ 0 & (1 + r_{k_t}) \end{pmatrix} \quad (19)$$

$$B_{k_t} = \begin{pmatrix} \text{diag}(1 + \rho_{i k_t}) & -\text{diag}(1 + \rho_{i k_t}) \\ -(1 + r_{k_t})d^+1' & (1 + r_{i k_t})d^-1' \end{pmatrix} \quad (20)$$

$$G_{k_t} = -I_{n+1} \quad H_{k_t} = \begin{pmatrix} 0 & I_n \\ d^+1' & -d^-1' \end{pmatrix}. \quad (21)$$

Let $\psi(t+1)$ be the lagrangian multipliers associated with the dynamics of the state variables at each time t and with $\lambda(t)$ the multipliers associated with the mixed constraints. The generalized Hamiltonian is then

$$\begin{aligned} \tilde{H}(x(t), u(t), \psi(t+1), \lambda(t)) &= \psi(t+1)'[A(t)x(t) + B(t)u(t)] + \\ &\lambda(t)'[G(t)x(t) + H(t)u(t)] \end{aligned} \quad (22)$$

Applying a discrete version of the Pontryagin Maximum Principle [32], we obtain for any time t the necessary, and in this case also sufficient, optimality conditions which can be written as the optimality conditions for a saddle point problem of the generalized Hamiltonian. These optimality conditions can be separated in time and reorganized into four different subproblems.

The resulting decomposed problems are

$$x(t+1) = A(t)x(t) + B(t)u(t) \quad (23)$$

$$x(0) = \bar{x} \quad (24)$$

$$\psi(t) = A(t)'\psi(t+1) - G(t)'\lambda(t) \quad (25)$$

$$\psi(T) = \frac{\partial L_T(x(T))}{\partial x(T)} \quad (26)$$

$$\begin{cases} \max_{u(t)} \{ \psi(t+1)'B(t)u(t) \} \\ H(t)u(t) \leq -G(t)x(t) \\ u(t) \geq 0 \end{cases} \quad (27)$$

$$\begin{cases} \min_{\lambda(t)} \{ -[G(t)x(t)]'\lambda(t) \} \\ H(t)'\lambda(t) \geq B(t)'\psi(t+1) \\ \lambda(t) \geq 0 \end{cases} \quad (28)$$

$$t = 0, \dots, T-1.$$

Given the block-diagonal structure of the matrices involved in problems (23)-(28) a further decomposition arises. Let x_{k_t} be the vector of decision variables at time t in node k_t ; and u_{k_t} the vector of controls at time t in node k_t . For each t conditions (23)-(28) can be decomposed with respect to the nodes of the event tree at time t .

The resulting sub-problems are

$$x_{k_t} = A_{k_t}x_{b(k_t)} + B_{k_t}u_{b(k_t)} \quad k_t \in \hat{K} \quad (29)$$

$$x_{k_t}(0) = \bar{x} \quad (30)$$

$$\psi_{k_t} = \sum_{j=1}^{D(k_t)} A'_j \psi_j - G'_{k_t} \lambda_{k_t} \quad k_t \in \tilde{K} \quad (31)$$

$$\psi_{k_T} = \frac{\partial L_T(x_{k_T})}{\partial x_{k_T}} \quad (32)$$

$$\begin{cases} \max_{u_{k_t}} \left\{ \left[\sum_{j=1}^{D(k_t)} \psi'_j B_j \right] u_{k_t} \right\} \\ H_{k_t} u_{k_t} \leq -G_{k_t} x_{k_t} \\ u_{k_t} \geq 0 \end{cases} \quad (33)$$

$$k_t \in \tilde{K}.$$

$$\begin{cases} \min_{\lambda_{k_t}} \{ -[G_{k_t} x_{k_t}]' \lambda_{k_t} \} \\ H'_{k_t} \lambda_{k_t} \geq \sum_{j=1}^{D(k_t)} B'_j \psi_j \\ \lambda_{k_t} \geq 0 \end{cases} \quad (34)$$

$$k_t \in \tilde{K}.$$

Conditions (29)-(34) can be solved separately in the framework of an iterative scheme which is presented in the next subsection.

Applying a discrete version of the Maximum Principle to the arborescent formulation of the multistage stochastic programming problem we have obtained a time and nodal decomposition of the dynamic portfolio problem.

The main advantage is that the deterministic equivalent problem can be tackled solving a number of smaller and easier subproblems linked together.

The proposed time decomposition applies both to the problem in the case of *implicit* non-anticipativity constraints and to the problem with *explicit* non-anticipativity constraints. In the first case it is self contained and allows to obtain a further nodal decomposition of the problem while, in the second case, it must be jointly applied with a solution approach that supplies scenario decomposition. We refer to [1] for the development of this time decomposition in the case of explicit non-anticipativity constraints in conjunction with the Progressive Hedging Algorithm [36], and to [2] for an application of the method to a dynamic tracking error portfolio problem.

5.2 The iterative solution scheme

To obtain the optimal solution of the global problem (10)-(14) we apply an iterative scheme in which, at each iteration, we first solve conditions (29)-(34) separately for each time t and node k_t , and then adopt an iterative fixed-point update

$$y^{\nu+1} = F(y^\nu) \quad (35)$$

where F is the transformation defined by conditions (29)-(34). We set $y^\nu = \{x_{k_t}, k_t \in \tilde{K}\}$; and for each ν the next value $y^{\nu+1}$ is obtained solving the four subproblems for each t and each k_t . At the first step an initial admissible solution y^0 is obtained substituting an initial admissible sequence for the controls in (29)-(30). The admissible sequence in the case of the portfolio problem is $u_{k_t} \equiv 0$ for all $k_t \in \tilde{K}$. The values obtained from the first subproblem for all k_t and the initial admissible sequence for the controls are used as initial conditions to obtain initial values for λ_{k_t} and ψ_{k_t} for all k_t through conditions (34) and (31)-(32), respectively.

The iterative scheme is applied to (35) according to the mean value iterative method introduced by Mann (see [26], [19], and [33]) which at each step of the algorithm considers a weighted average of the admissible solutions found

in previous steps. Let z^ν be the weighted average of optimal solutions up to iteration ν and $z^0 = y^0$. The mean value iteration scheme is defined as

$$y^{\nu+1} = F(z^\nu) \quad (36)$$

$$z^\nu = \sum_{i=1}^{\nu} \delta_i^\nu y^i \quad (37)$$

where δ_i^ν denotes the elements of the ν -th row of an infinite triangular matrix Δ with the following properties

$$\delta_i^\nu \geq 0 \quad \forall \nu, i \quad (38)$$

$$\delta_i^\nu = 0 \quad \forall i > \nu \quad (39)$$

$$\sum_{j=1}^i \delta_j^\nu = 1 \quad \forall i. \quad (40)$$

Different matrices Δ can be applied, among them the Cesàro matrix (see [26]). To improve the speed of convergence we introduce an optimization step which allows us to choose the weights in an optimal way with respect to the objective function of the original problem.

At each step ν of the iterative scheme we do not fix a priori the weights, $\delta^\nu = (\delta_1^\nu, \dots, \delta_\nu^\nu)$, as in the Cesàro matrix, but we look for the coefficients best choice, as the solution of:

$$\max_{\delta^\nu} f(\delta^\nu) \quad (41)$$

$$\sum_{i=1}^{\nu} \delta_i^\nu = 1 \quad (42)$$

$$\delta_i^\nu \geq 0 \quad \forall i = 1, \dots, \nu. \quad (43)$$

where f is the objective function (1) of the portfolio problem in which the decision variables x_{kT} are substituted with the linear combination of the values obtained at previous iterations expressed as functions of δ^ν .

At each step of the iterative scheme the feasibility of the proposed solution, z^ν , is guaranteed by the constraints imposed on δ , since the feasible region of the original problem is convex.

For example, if f is a quadratic utility function of the form $f(w) = w - aw^2$, where w denotes the wealth at the end of the horizon, problem (41)-(43) is a quadratic optimization problem which is rather easy to solve. In general, if f is nonlinear it is possible either to directly solve the resulting nonlinear optimization problem or to consider a linear-quadratic approximation to $f(\delta)$ in (41) and solve the resulting quadratic optimization problem. The weights must satisfy constraints (42) and (43) and that choosing the weights in an optimal way improve considerably the convergence speed of the iterative scheme.

The number of variables of the optimization problem (41)-(43) increases linearly with the number of iterations of the fixed-point scheme, which ultimately depends on the precision required for the optimal solutions.

The convergence of the iterative scheme is monitored through two different stopping criteria. The first relates the objective function while the second applies to the sequence of the proposed solutions. If we denote with ϵ_1 and ϵ_2 the parameters for the precisions, we require that $\|f^\nu - f^{\nu-1}\| \leq \epsilon_1$ and/or $\|z^\nu - z^{\nu-1}\|_\infty \leq \epsilon_2$.

This improvement in the iterative scheme has a minor impact in the case of PMPTD (Portfolio Maximum Principle Time Decomposition, see [1]). In that case, however, the major concern for the convergence is the outer iterative procedure governed by the Progressive Hedging Algorithm. The main drawback of PHA, which is widely documented in the literature, is the sensitivity of the convergence speed and solution accuracy to the penalty parameter involved in the augmented objective function.

6 Computational results

We test our solution method against two other methods to solve problem (1)-(7). The first method, referred to as PMPTD, applies the Progressive Hedging Algorithm obtaining a scenario decomposition and solves each scenario problem applying the Maximum Principle that brought a time decomposition (see [1]). The second method, referred to as ICGLOBAL (Implicit Constraints Global), solves the global deterministic-equivalent optimization problem with a general purpose routine without exploiting the structure. Our method, that applies the Maximum Principle to the deterministic equivalent problem written in the arborescent form, is denoted by ICMP (Implicit Constraints Maximum Principle).

In our tests we consider a quadratic utility function in (1), as a consequence the objective function (41) is quadratic in the vector of weights δ . Let w be the vector whose elements are the final wealths in each scenario, that is the values of the portfolio, corresponding to the vectors of the amounts y . W is the matrix whose columns are given by the vectors w^i , $i = 1, \dots, \nu$ obtained in the first ν iterations of the fixed-point scheme. Moreover using the vector of probabilities assigned to each scenario, π , and denoting $\Pi = \text{diag}(\pi)$ a diagonal matrix that has the elements of π as diagonal elements we obtain for (41) the expression

$$f(\delta) = \pi'W\delta + a\delta'W'\Pi W\delta \quad (44)$$

We consider a set of test problems with increasing number of scenarios and risky assets. To generate the scenario trees we apply an historical simulation approach using data from the Milan Comit Indexes quoted in the Italian stock market. The dataset ranges from November 28, 1996 to September 14, 2006. The Indexes data history is presented in Figure 3. In Table 1 we present a set of descriptive statistics on the log-returns obtained from the time series of the indexes. We apply a simultaneous bootstrapping techniques across all time series

to generate return scenarios over different horizons. We consider regular 3-period scenario trees and for each node we randomly choose from historical returns the realization for the descendant nodes. The resulting random return dynamics are thus model independent and eliminate, if present, serial correlation. The procedure is repeated for each node in the tree from $t = 0$ to $t = T - 1$.

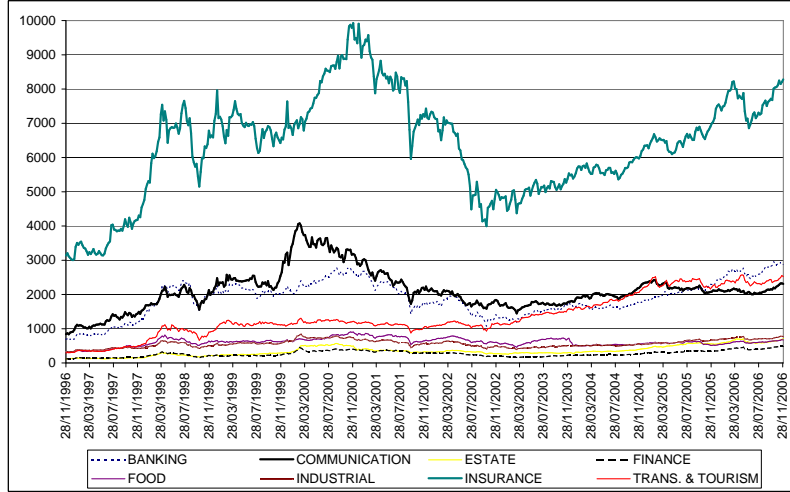


Fig. 3. Values of the Comit Sector Indexes: Banking, Communication, Estate, , Food, Industrial, Insurance, Transportation & Tourism, for the period November 28, 1996 to September 14, 2006.

Table 1. Statistics on the logarithmic returns for the Comit Sector Indexes in the period December 5, 1996 to September 14, 2006.

	mean	variance	asymmetry	kurtosis
Banking	0.00274	0.00114	-0.27837	3.37530
Communication	0.00193	0.00113	0.11400	1.11916
Estate	0.00335	0.00083	0.06336	2.46251
Finance	0.00269	0.00124	-0.10303	1.71502
Food	0.00141	0.00091	-0.95065	4.48844
Industrial	0.00163	0.00083	-0.45302	2.69041
Insurance	0.00186	0.00112	-0.30728	2.73643
Transp.& Tourism	0.00413	0.00087	0.24116	4.27671

We present in Table 2 the number of iterations and the time (in seconds) required by each method. The computational experiments have been carried

out on a personal computer with Pentium 4, 3.2 Mhz CPU and 1 GB RAM. The algorithm was coded using Gauss (Aptech Systems, Inc.) and its quadratic optimization routine.

For PMPTD and ICPM we set the following tolerance parameters $\epsilon_1 = 0.5 \cdot 10^{-5}$ and $\epsilon_2 = 10^{-3}$, while in the case of ICGLOBAL we accept the default tolerance parameter of Gauss optimization routine. Moreover in the case of PMPTD we need to choose a penalty parameter ρ which is crucial in the trade-off between solution accuracy and convergence speed. A good range of values, in the analyzed cases where we set the initial wealth $w_0 = 100$ and the utility parameter $a \in [-0.3/w_0, -0.1/w_0]$, proved to be $\rho \in [0.01, 0.1]$.

We observe that in the case of ICGLOBAL the algorithm reaches the insufficient memory limit (i.m.) very soon, while the PMPTD requires a great amount of iterations which results in a time limit exceeded (t.l.), this means that the computational time exceeded the 240 000 seconds.

Table 2. Comparison among PMPTD, ICGLOBAL and ICPM solution approaches for a set of problems with increasing number of scenarios (S), and risky assets (n); i.m. = insufficient memory and t.l. = time limit exceeded.

n	S	PMPTD		ICGLOBAL		ICMP	
		iter.	time(sec.)	iter.	time(sec.)	iter.	time(sec.)
4	64	1196	141	392	0.3	10	0.2
6	512	837	379	2186	78	9	0.5
7	2744	1266	2416	9218	7707	40	3
8	2744	657	1395	i.m.	-	38	3
10	2197	959	2130	9357	7853	51	4
11	2197	1451	3666	i.m.	-	40	3
13	1728	2951	4429	10407	8611	41	3
14	1728	7612	11363	i.m.	-	28	2
17	1331	6042	8389	10401	9714	31	3
18	1331	9931	18261	i.m.	-	37	4
22	1000	9389	16370	9799	7678	30	3
23	1000	13790	17313	i.m.	-	29	3
24	1000	-	t.l.	i.m.	-	24	3
25	1000	-	t.l.	i.m.	-	61	7
25	8000	-	t.l.	i.m.	-	232	162
25	27000	-	t.l.	i.m.	-	268	276
27	8000	-	t.l.	i.m.	-	210	147
27	27000	-	t.l.	i.m.	-	370	301
30	27000	-	t.l.	i.m.	-	279	319
30	91125	-	t.l.	i.m.	-	453	2218
30	140608	-	t.l.	i.m.	-	551	3774

6.1 Dynamic portfolio management: an example

We consider a dynamic portfolio management problem and apply the proposed decomposition method to solve the sequence of 2-period portfolio problems in the framework of a rolling horizon simulation.

Scenario generation and portfolio evaluation are based on the same dataset of the Comit indexes described at the beginning of section 6. Besides the eight risky assets we include a liquidity component for the portfolio for which we assume a weekly constant interest rate of 0.03%.

The scenario trees are generated using historical simulation from the time series of the Comit indexes as already described in this section. We consider regular 2-stage scenario trees with a constant number of branches from each node and a rolling simulation over a 10-week period with weekly re-balancing. The simulation experiments has been conducted according to the following steps. At time $t = 0$ we start our simulation experiment with an initial endowment in cash, normalized to 100 for sake of simplicity. We generate a 2-period scenario tree applying historical bootstrap and solve the corresponding 2-stage stochastic programming problem using the approach proposed in section 5. The first period optimal decision is implemented and the corresponding portfolio is composed. At the end of the first period we observe the realized returns in the market and compute the values of each component of the portfolio. These values represent the initial endowment for the next period. For each $t = 1, \dots, 9$ we generate a 2-period scenario tree and repeat the optimization and evaluation procedure described in the first step. At time $t = 10$ we evaluate the portfolio at the current realized returns.

The optimal policy over the 10-week simulation period is thus obtained as the sequence of first time optimal decisions for each problem, where the initial endowment depends on the value of the portfolio obtained from the previous period evaluated using the current market conditions.

The values obtained from the sequence of optimal problems are compared with those obtained from a static equally-weighted buy and hold strategy, in which the portfolio is composed at the beginning of the investment period and there is no re-balancing. We consider two equally-weighted portfolios composed using only the risky assets. In the first we do not include transaction costs while in the second we consider the same proportional transaction cost, equal to 0.2%, used in the optimized portfolio.

We carried out different experiments over different periods, in each experiment we consider a 10-week management period and in order to generate the scenario trees we consider all the observations in the dataset up to the beginning of the simulation period in such a way that there is no overlap between the data used to generate the scenarios and the data used to evaluate the portfolio.

Different examples can be considered, using scenario trees with more periods or considering different methods for turning the optimal solutions into a dynamic trading strategy, for a brief discussion see, for example, [18].

In the sequel we present the results obtained with increasing number of scenarios. The 10-week simulation period from April 15, 2004 to July 1, 2004, considered in the example, corresponds to a relatively stable period with respect to the values of the indexes considered. In Figure 4 we compare the values of the optimal portfolio strategies obtained solving the problem with an increasing number of scenarios, ranging from 25 to 500, with the values of the equally-weighted static portfolio composed with and without transaction cost, and denoted with ew_8_tc and ew_8 , respectively. We can observe that increasing the number of scenarios the solution of the problem becomes more stable and outperforms the static strategy. The portfolio values obtained optimizing over scenario trees with few branches are more volatile and the corresponding optimal solution are heavily dependent on the associated scenario tree while increasing the number of scenarios the optimal solution becomes more stable across different scenario trees and with respect to a larger number of branches. A sufficient number of scenarios can thus be implicitly identified and an associated robust portfolio strategy suggested to the decision maker. The optimal portfolio values and the optimal solutions obtained with 200 and 500 branches are not distinguishable. These results are confirmed independently of the adopted simulation periods.

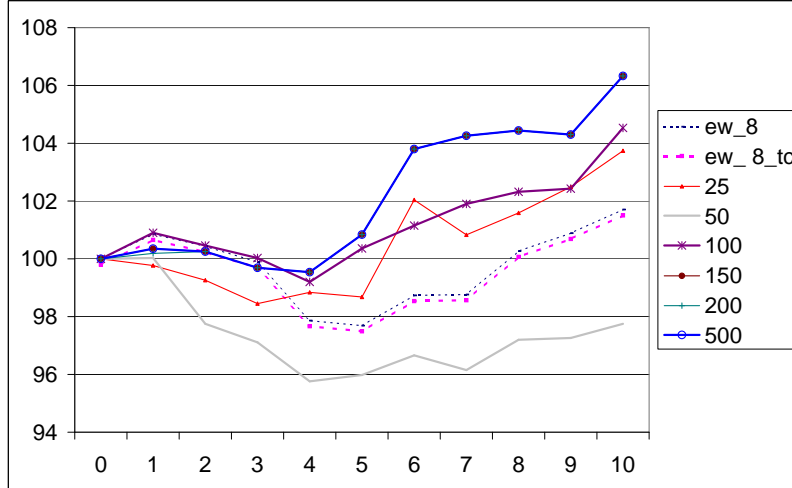


Fig. 4. Comparison among the values of the equally-weighted portfolios, with and without transaction costs, and the optimized portfolios with an increasing number of scenarios for the 10-week period April 15, 2004 - July 01, 2004.

7 Concluding remarks

We did present in this work a solution approach for a dynamic portfolio problem written as a multistage stochastic program in arborescent form, with implicit non-anticipativity constraints.

The problem can be rewritten as a discrete time optimal control problem, in which the state variables dynamic equations in the primal space connect a node with its descendants (forward from time t to time $t + 1$), while the adjoint variables dynamics connect in the dual space a node with its (unique) ancestor (backward from time $t + 1$ to time t). The mixed constraints represent the feasibility conditions with respect to the information structure: they relate the optimal decision in a specific node with the endowment received from previous period.

Applying a discrete version of the Maximum Principle to the arborescent formulation of the problem we have shown how to obtain a time and then a nodal decomposition of the original problem into smaller subproblems.

To reach a global solution we apply an iterative scheme in which, at each iteration we solve for each time step and for each node in the event tree four subproblems. The solution obtained at each iteration is certainly feasible but not necessarily optimal. For optimality we apply an iterative mean value method with an optimization step which allows us to optimally choose the weights. This method allows both an efficient decomposition of the deterministic equivalent problem and the convergence towards the optimal solution for the global problem with a limited number of iterations.

The comparison with two other solution approaches, such as the direct solution of the global deterministic equivalent problem and a decomposition according to scenarios, shows that the proposed method efficiently solves higher dimensional problems with reduced iterations and very competitive computational times.

In the final section relying on the proposed solution method the suitability of a dynamic approach against other commonly adopted portfolio strategies has also been documented with data from the Italian stock market.

References

1. Barro, D., Canestrelli, E.: Dynamic portfolio optimization: time decomposition using the maximum principle with a scenario approach. *European Journal of Operational Research* **163** (2005) 217–229
2. Barro, D., Canestrelli, E.: Tracking error: a multistage portfolio model. *Annals of Operations Research* (forthcoming)
3. Berger, A.J., Mulvey, J.M.: The Home Account Advisor, asset and liability management for private investors. In Ziemba, W.T., Mulvey J.M. (eds.): *World wide asset and liability modeling*, Cambridge University Press (1998) 634–665
4. Birge, J.R.: Decomposition and partitioning methods for multistage stochastic programs. *Operation Research* **33** (1985) 989–1007
5. Birge, J.R.: Stochastic programming computation and applications. *INFORMS Journal on Computing* **9** (1997) 111–133

6. Birge, J.R., Holmes, D.F.: Efficient solution of two-stage stochastic linear programs using interior point methods. *Computational Optimization and Applications* **1** (1992) 245–276
7. Birge, J.R., Louveaux, F.: *Introduction to stochastic programming*. Springer (1997)
8. Birge, J.R., Qi, L.Q.: Computing block-angular Karmarkar projections with applications to stochastic programming. *Management Science* **34** (1988) 1472–1479
9. Birge, J.R., Rosa, C.H.: Parallel decomposition of large-scale stochastic nonlinear programs. *Annals of Operations Research* **64** (1996) 39–65
10. Bradley, S.P., Crane, D.B.: A dynamic model for bond portfolio management. *Management Science* **19**(2) (1972) 139–151
11. Canestrelli, E.: Sulla ricerca di soluzioni ottimali per una classe di problemi di controllo ottimo nel discreto. *Atti del IV Convegno A.M.A.S.E.S. Grottaferrata, Roma* (1981)
12. Canestrelli, E.: Soluzione numerica di problemi di controllo ottimo nel discreto a vincoli lineari. *Atti del IX Convegno A.M.A.S.E.S. Levico Terme, Trento* (1985)
13. Cariño, D.R., Kent, T., Myers, D.H., Stacy, C., Sylvanus, M., Turner, A.L., Watanabe, K., Ziemba, W.T.: The Russell-Yasuda Kasai model: an asset/liability model for a Japanese insurance company using multistage stochastic programming. *Interfaces* **24**(1) (1994) 29–49
14. Consigli, G.: Asset-Liability management for individual investors. In: Ziemba, W.T., Zenios, S.A. (eds): *North-Holland Handbook on Asset and Liability Management*, vol. B, North-Holland (2006) in print
15. Consigli, G., Dempster, M.A.H.: Dynamic stochastic programming for asset-liability management. *Annals of Operations Research* **81** (1998) 131–161
16. Dantzig, G.B., Infanger, G.: Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research* **45** (1993) 59–76
17. Dantzig, G.B., Glynn, P.: Parallel processors for planning under uncertainty. *Annals of Operations Research* **22** (1990) 1–21
18. Dempster, M.A.H., Thompson G.W.P.: Dynamic portfolio replication using stochastic programming. In: Dempster, M.A.H. (ed.): *Risk management: value at risk and beyond*, Cambridge University Press, Cambridge (2002) 100–128
19. Franks, R.L., Marzec, R.P.: A theorem on mean-value iterations. *Proceedings of the American Mathematical Society* **30** (1971) 324–326
20. Gondzio, J., Ruszczyński, A.: A sensitivity method for basis inverse representation in multistage stochastic linear programming problems. *Journal of Optimization Theory and Applications* **74** (1992) 221–242
21. Higle, J., Sen, S.: Stochastic decomposition: an algorithm for two-stage stochastic programs with recourse. *Mathematics of Operations Research* **16** (1991) 650–669
22. Jarrow, R.A., Maksimovic, V., Ziemba, W.T. (eds.): *Finance. Handbooks in operations research and management science*, vol.9, North-Holland (1995)
23. Klaassen, P.: Solving stochastic programming models for asset/liability management using iterative disaggregation. In: Ziemba, W.T., Mulvey J.M. (eds.): *World wide asset and liability modeling*, Cambridge University Press (1998) 427–463
24. Kusy, M.I., Ziemba, W.T.: A bank asset and liability management model. *Operations Research* **34**(3) (1986) 356–376
25. Lustig, I.J., Mulvey, J.M., Carpenter, T.J.: Formulating stochastic programs for interior point methods. *Operations Research* **39** (1991) 757–770
26. Mann, W.R.: Mean value methods in iteration. *Proceedings of the American Mathematical Society* **4** (1953) 506–510

27. Mulvey, J.M., Shetty, B.: Financial planning via multi-stage stochastic optimization. *Computer & Operations Research* **31** (2004) 1–20
28. Mulvey, J.M., Ruszczyński, A.: A digonal quadratic approximation method for large scale linear programming. *Operations Research Letters* **12** (1992) 205–215
29. Mulvey, J.M., Ruszczyński, A.: A new scenario decomposition method for large scale stochastic optimization. *Operations Research* **43** (1995) 477–490
30. Mulvey, J.M., Ziemba, W.T.: Asset and liability allocation in a global environment. In: Jarrow, R.A., Maksimovic, V., Ziemba, W.T. (eds.): *Finance. Handbooks in operations research and management science*, vol.9, North-Holland (1995) 435–464
31. Nielsen, S.S., Zenios, S.A.: A massively parallel algorithm for nonlinear stochastic network problems. *Operations Research* **41** (1993) 319–337
32. Ortega, A.J., Leake, R.S.: Discrete maximum principle with state constrained control. *SIAM Journal on Control and Optimization* **15** (1977) 119–147
33. Rhoades, B.E.: Fixed point iterations using infinite matrices. *Transactions of the American Mathematical Society* **196** (1974) 161–176
34. Rockafellar, R.T.: Linear quadratic programming and optimal control. *SIAM Journal on Control and Optimization* **25** (1987) 781–814
35. Rockafellar, R.T., Wets, R.B.J.: Generalized linear quadratic problems of deterministic and stochastic optimal control in discrete time. *SIAM Journal on Control and Optimization* **28** (1990) 810–822
36. Rockafellar, R.T., Wets, R.B.J.: Scenario and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research* **16** (1991) 119–147
37. Rockafellar, R.T., Zhu, C.Y.: Primal-dual projected gradient algorithms for extended linear quadratic programming. *SIAM Journal on Optimization* **3** (1993) 751–783
38. Rosa, C., Ruszczyński, A.: On augmented lagrangian decomposition methods for multistage stochastic programs *Annals of Operations Research* **64** (1996) 289–309
39. Ruszczyński, A.: A regularized decomposition for minimizing a sum of polyhedral functions. *Mathematical programming* **35** (1986) 309–333
40. Ruszczyński, A.: An augmented lagrangian decomposition method for block diagonal linear programming problems. *Operations Research Letters* **8** (1989) 287–294
41. Ruszczyński, A.: Parallel decomposition of multistage stochastic programming problems. *Mathematical Programming* **58** (1993) 201–228
42. Ruszczyński, A.: Some advances in decomposition methods for stochastic linear programming. *Annals of Operations Research* **85** (1999) 153–172
43. Van Slyke, R., Wets, R.B.J.: L-Shaped linear programs with applications to optimal control and stochastic linear programs. *SIAM Journal of Applied Mathematics* **17** (1969) 638–663
44. Vladimirov, H.: Computational assessment of distributed decomposition methods for stochastic linear programs. *European Journal of Operational Research* **108** (1998) 653–670
45. Wallace, S.W., Ziemba, W.T. (eds.): *Applications of stochastic programming*. SIAM – Mathematical Programming Society Series on Optimization (2005)
46. Ziemba, W.T., Mulvey, J.M. (eds.): *World wide asset and liability modeling*. Cambridge University Press (1998)
47. Ziemba, W.T., Vickson, R.G.: *Stochastic optimization models in finance*. World Scientific, 2nd edition (2006)

