A Simulated Annealing Approach for Multistage Portfolio Optimization

Maria A. Osorio, Erika C. Jimenez, A. Sánchez L. and Miguel A. Gómez*

Facultad de Ciencias de la Computación, BUAP
14 Sur y San Claudio, 72570 Puebla, Pue. México
*Universidad de las Américas Puebla, Cholula, Pue. Mexico

Abstract. This paper describes the application of a simulated annealing approach to find an optimal investment strategy by maximizing expected wealth at the end of a multistage horizon, considering wealth, return, cash balance, withdrawals and upper bounds. The discretization of the random return values and its probability was represented in a scenario tree generated with simulation and randomized clustering. The performance of the linear optimization model on different scenario trees is illustrated using test examples. The model is stochastic, and exact optimization algorithms may have difficulties with large or complex instances, motivating the research of heuristic techniques. The computational results indicate that the approach is promising for this sort of problems because easily allow the introduction of more specific and real conditions, as constraints in the model.

1 Introduction

In financial portfolio management, multistage stochastic programming is used to find an optimal investment strategy by maximizing expected wealth at the end of the planning horizon taking in account the possible fluctuation of the assets return in the future (Trippy et al. [9]). The uncertainty on return values of instruments is accurately described by a continuous distribution represented by a discrete approximation. Given history up to the commencement of the investment period, the determination of the finitely many outcomes of the random return variables is called scenario tree generation. The discretization of the random values and the occurrence probability constitute a scenario tree (see Gülpinar et al [3]).

A Linear Programming (LP) model to maximize the expected wealth at the end of the investment horizon can be easily build. Expected wealth is calculated as the total net redemption value at time period T. The model is multistage because it uses the wealth generated in the previous period in order to represent the constraint in the next period. It takes into account the uncertainty of the assets return, based in the history of each asset, representing them in a scenario tree (as in Osorio et al [8]).

In spite of its theoretical interest, the basic portfolio optimization 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 that involve withdrawals, diversification constraints and left open the option to include specific conditions for different applications. This is the context

© S. Torres, I. López, H. Calvo. (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 27, 2007, pp. 65-76

Received 23/02/07
Accepted 08/04/07
Final version 21/04/07

where good metaheuristic techniques become important. In particular, Simulated Annealing has demonstrated to be an efficient and promising technique that can candle the complex models in an adequate way (Crama and Schyns [1]).

The rest of the paper is organized as follows. The Multistage Optimization in Portfolio Management theory and models is described in section 2. Section 3 includes a complete description of the Simulated Annealing procedure used. Computational examples are presented in section 4, and conclusions in section 5.

2 Multistage Optimization in Portfolio Management

In financial portfolio management, multistage stochastic programming is used to find an optimal investment strategy by maximizing expected wealth subject to constraints specified by the investor [8]. The uncertainty on return values of instruments is represented by a discrete approximation. Given history up to the commencement of the investment period, the determination of the finitely many outcomes of the random return variables is called scenario tree generation. Generating scenario trees is important for the performance of the multistage stochastic programming. The root node of the scenario tree represents the decision "today" and the nodes further on represent conditional decisions at later stages. The arcs linking the nodes represent various realizations of the uncertain variables. The dynamics of decision making is thus captured as decisions are adjusted according to realizations of uncertainty.

We use a multistage approach to the portfolio management problem to obtain a return-efficient multistage portfolio. The main concern of this paper is to find an optimal investment strategy using different asset allocations over a given finite investment horizon. Uncertainty on asset performances (or returns) is represented with a scenario tree generated by simulation. The performance of the linear optimization model on different scenario trees is illustrated using test examples.

2.1 Uncertainty Representation and Scenario Trees

Coherent uncertainty representation is a requirement for this type of models. The uncertainty is usually expressed in terms of multivariate continuous distributions. In order to represent the continuous distributions, the decision model is generated with internal sampling or a discrete approximation of the underlying continuous distribution. The random variables are the uncertain return values of each asset on an investment. The discretization of the random values and the probability space leads to a framework in which a random variable takes finitely many values. At each time period, new scenarios branch from the old, creating a scenario tree. Scenario trees can be generated based on different probabilistic approaches as simulation or optimization as presented in Gulpinar et al. [3].

Scenario trees can have different structures as shown in Fig. 1. For this research we took the last option of every parent having two branches.

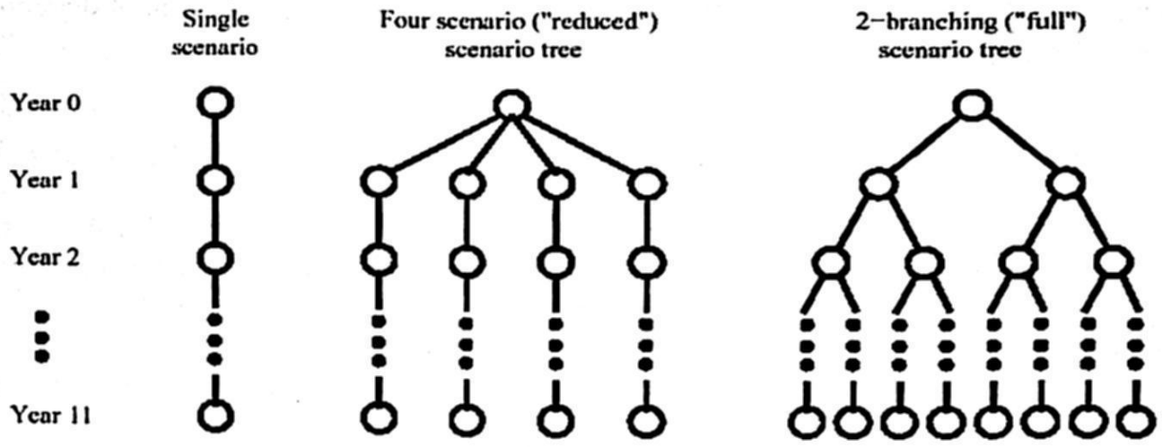

Fig. 1 Scenario Trees for Multiperiod Optimization

We assumed a portfolio of n risky assets and consider its optimal restructuring over a period in terms of expected return. After the initial investment (t=0), the portfolio may be restructured at discrete times t = 1, ..., T-1, and redeemed at the end of the period , (t=T).

Let the increasing σ-field F_t ($F_1 \subseteq \ldots \subseteq F_T$) be generated by stochastic events $\boldsymbol{\rho}^t \equiv \{\rho_1 \ldots \rho_t\}$; t = *1*, ..., *T*. Let the random variables $\mathbf{r}_t(\boldsymbol{\rho}^t)$ and $\mathbf{g}_t(\boldsymbol{\rho}^t)$ denote the uncertain dividend (or income) and capital gain returns on investment. Random variables and some specified coefficients of constraints are assumed to be F_t measurable functions ($\mathbf{r}_t$, $\mathbf{g}_t$: $\Omega_t \rightarrow \Re^n$) on some probability space (Ω_t, F_t, P_t). Due to the recourse nature of the multistage problem, decision variables $\mathbf{w}_t$, $\mathbf{b}_t$, and $\mathbf{s}_t$ are influenced by previous stochastic events $\boldsymbol{\rho}^t$, and hence $\mathbf{w}_t = \mathbf{w}_t(\boldsymbol{\rho}^t)$, $\mathbf{b}_t = \mathbf{b}_t(\boldsymbol{\rho}^t)$ and $\mathbf{s}_t = \mathbf{s}_t(\boldsymbol{\rho}^t)$. However, for simplicity, we shall use the terms $\mathbf{w}_t$, $\mathbf{b}_t$, and $\mathbf{s}_t$, and assume their implicit dependence on $\boldsymbol{\rho}^t$. We assume that ρ_t can take only finitely many values. Thus, the factors driving the risky events are approximated by a discrete set of scenarios or a sequence of events. Given the event history up to a time t, $\boldsymbol{\rho}^t$, the uncertainty in the next period is characterized by finitely many possible outcomes for the next observation ρ_{t+1}. This branching process is represented using a scenario tree.

A scenario is defined as a possible realization of the stochastic variables $\{\rho_1, \ldots, \rho_T\}$. Hence, the set of scenarios corresponds to the set of leaves of the scenario tree, N_T, and nodes of the tree at level $t \geq 1$ (the set N_t) correspond to possible realization of $\boldsymbol{\rho}^t$. We denote a node of the tree (or **event**) by $\mathbf{e} = (s,t)$, where s is a scenario (path from root to leaf), and time period t specifies a particular node on that path. The root of the tree is $\mathbf{0} = (s,0)$ (where s can be any scenario, since the root node is common to all scenarios). The ancestor (parent) of event $\mathbf{e} = (s,t)$ is denoted $a(\mathbf{e}) = (s, t-1)$, and the branching probability $p_\mathbf{e}$ is the conditional probability of event $\mathbf{e}$, given its parent event $a(\mathbf{e})$. The path to event $\mathbf{e}$ is a partial scenario with probability $P_\mathbf{e} = \prod p_\mathbf{e}$ along that path. Since probabilities $p_\mathbf{e}$ must sum to unity at each individual branching, probabilities $P_\mathbf{e}$ will sum up to unity across each layer of tree-nodes N_t for t = 0,1, ...,T.

Each node $\mathbf{e} \in N_t$ at a level t = *1*, ..., T corresponds to a decision $\{\mathbf{w}_\mathbf{e}, \mathbf{b}_\mathbf{e}, \mathbf{s}_\mathbf{e}\}$ which must be determined at time t, and depends in general on $\boldsymbol{\rho}^t$ and the past decisions $\{\mathbf{w}_j,$

$\mathbf{b}_j$, $\mathbf{s}_j\}$, for $j = 1, \ldots, t-1$. This process is adapted to ρ^t as $\mathbf{w}_t$, $\mathbf{b}_t$, $\mathbf{s}_t$ cannot depend on future events $\rho_{t+1} \ldots \rho_T$ which are not yet realized.

2.2 Scenario Trees Generated by a Simulation and Randomized Clustering

The scenario tree is the input to the financial optimization problem. The basic data structure is the scenario tree node, which contains a cluster of scenarios (vectors in Rn), one of which is designated as the centroid. The final tree consists of the centroids of each node and their branching probabilities.

We used the main steps used to generate the scenario tree according to Gülpinar et al. [3]. These steps are:

Step 1: (Initialization) Create a root node, with N scenarios. Initialize all the scenarios (including the centroid) with the desired starting point ("today's" prices). For a job queue consisting of the root node.

Step 2: (Simulation) Remove a node from the job queue. Simulate one time period of growth (from "today" to "tomorrow") in each scenario.

Step 3: (Randomized seeds) Randomly choose a number of distinct scenarios around which to cluster the rest: one per desired branch in the scenario tree.

Step 4: (Clustering) Group each scenario with the seed point to which it is the closest. If the resulting clustering is unacceptable, return to step 3.

Step 5: (Centroid selection) For each cluster, find the scenario which is the closest to its center, and designate it as centroid.

Step 6: (Queuing) Create a child scenario tree node for each cluster (with probability proportional to the number of scenarios in the cluster), and install its scenarios and centroid. If the child nodes are not leaves, append to the job queue. If the queue is nonempty, return to step 2. Otherwise, terminate the algorithm.

2.3 Definitions and Notation

Portfolio: A set of assets available for the investor.

Assets: The assets considered are Equities in the Mexican Bursaries Market (BMV), available for the constitution of a portfolio distribution.

Returns: Percentage of returns in the form of dividends for equities.

Net Redemption Value: Total amount of money received at the end of the horizon, when a the investment is encashed.

The notation used in the following definitions is described in Table 1. All quantities in boldface represent vectors in $\Re^n$. The transpose of a vector is denoted with the symbol $'$. In Table 1, subscript * indicates that vectors have two indices. The first index represents assets $i = 1,2, \ldots, n$. The second one denotes each event $\mathbf{e} \in N_t$ at time $t = 1, \ldots, T$ of the scenario tree.

Table 1. Notation

Symbols and Input Data	
$\mathbf{1}$	$\equiv (1,1,1, \ldots, 1)'$
$\mathbf{p} \circ \mathbf{q}$	$\equiv (p_1q_1, p_2q_2, \ldots, p_nq_n)'$ (Hadamard product)
$\mathbf{p}'\mathbf{q}$	$\equiv p_1q_1 + p_2q_2 + \ldots + p_nq_n$ (Inner product)
$\mathbf{e} \equiv (s,t)$	index denoting an event (a node of the scenario tree)
$a(\mathbf{e})$	ancestor of event **e** (parent in the scenario tree)
N_t	set of nodes of the scenario tree at time t
$p_\mathbf{e}$	branching probability of event **e**: $p_\mathbf{e} = \text{Prob}[\mathbf{e} \mid a(\mathbf{e})]$
$P_\mathbf{e}$	probability of event **e**: **if** $\mathbf{e} = (s,t)$, then $P_\mathbf{e} = \prod_{i=1 \ldots t} p_{(s,i)}$
n	number of investment assets
M	amount of initial investment
T	investment planning horizon
TW_t	total withdrawal at time t
ic_i	percentage paid in initial cost for asset i
ac_i	percentage paid in annual cost for asset i
$\mathbf{r}_{i\mathbf{e}}$	dividends or income returns for asset i at node **e**
tc	transaction cost
$\mathbf{w}^u_{i\mathbf{e}}$	upper bound for asset i
Decision Variables	
NR	net redemption value
$\mathbf{w}_\bullet$	amount of money held in each asset
$\mathbf{h}_\bullet$	withdrawal
$\mathbf{b}_\bullet$	amount bought of each asset
$\mathbf{s}_\bullet$	amount sold of each asset

2.4 Multistage LP Problem

The Linear Programming (LP) model maximizes the expected wealth at the end of the investment horizon. Expected wealth is calculated as the total net redemption value at time period T.

The redemption value is basically defined as the amount of money received at time T when the investment is encashed. The basic LP model only includes constraints to express the wealth return and cash balance. We added annual bank fees, transaction costs for purchase operations, the withdrawal variable in the wealth return equation, the total withdrawal (TW_t) equation in the model and the upper bounds on the assets amount in a diversification constraint in order to obtain a more complete and descriptive model. The constraints in the LP model are:

Net Redemption Value of every asset. **(1)**
Initial Allocation. **(2)**
Cash Balance Equations. **(3)**
Wealth for asset i in node **e**. **(4)**
Total Withdrawal at time t. **(5)**

Diversification constraints (6)

The model is multistage because it uses explicitly the wealth generated in the previous period in order to obtain the wealth in the next period. It takes into account the uncertainty of the assets return, based in the history of each asset and represent it in a scenario tree. The objective function is the sum of the net redemption values of every asset at the end of the complete horizon, i.e. the net amount of money that the investor can obtain when the total investment is encashed. The general expression for the multistage portfolio optimization model is:

$$\max \Sigma_{i=1,n} NR_i .$$

Subject to

$$NR_i = \Sigma_{e \in NT} Pe\ [1'\ \mathbf{w}_{ie}] \qquad i=1,\ldots,n \qquad (1)$$

$$\Sigma_{i=1,n} 1'\mathbf{w}_{i0} = M \qquad (2)$$

$$1'\mathbf{b}_{ie} - 1'\mathbf{s}_{ie} = 0 \qquad e \in N_t, t=1,\ldots,T,\ i=1,\ldots,n \qquad (3)$$

$$\mathbf{w}_{ie} = (1 - ac_i)\ [(1+\mathbf{r}_{ie})\mathbf{w}_{ia(e)}] - \mathbf{h}_{ie} + (1 - tc)\mathbf{b}_{ie} - \mathbf{s}_{ie} \qquad e \in N_t, t=1,\ldots,T, i=1, \ldots, n \qquad (4)$$

$$TW_t = \Sigma_{e \in Nt} Pe\ \Sigma_{i=1,n}\ 1'\mathbf{h}_{ie} \qquad t=1,\ldots,T \qquad (5)$$

$$\Sigma_{i=1,n}\ \mathbf{w}_{ie} \leq \mathbf{w}^u_{ie}\ \Sigma_{i=1,n}\ (1'\mathbf{w}_{ie}) \qquad e \in N_t, t=1,\ldots,T \qquad (6)$$

$$NR_i \geq 0 \qquad i=1,\ldots,n$$

$$\mathbf{w}_{ie},\ \mathbf{b}_{ie},\ \mathbf{s}_{ie} \geq 0 \qquad e \in N_t, t=1,\ldots,T, i=1, \ldots, n$$

Notice that the annual bank fees deducted by term $(1 - ac_i)$ for $i = 1, \ldots, n$ must be augmented by the bank's initial setup fees in the first year. For children of the root scenario node, $\mathbf{e} \in N_1$, the term becomes $(1 - ic_i - ac_i)$, and is imposed on all constraints. The wealth in every period t for asset i, is $\Sigma_{e \in Nt} Pe\ (1'\mathbf{w}_{ie})$, for $i=1,\ldots,n$, and $t=1,\ldots,T$. The total wealth in for every period can be evaluated as $\Sigma_{e \in Nt} Pe\ \Sigma_{i=1,n}\ 1'\mathbf{w}_{ie}$, for $t=1,\ldots,T$.

The number of variables and constraint in the LP model is increased by the number of assets and the topology of the scenario tree. The size of the scenario tree depends on the depth and branching at each time period. Our computational results show that even for large scenario trees it is possible to find solutions near to the optimal in a reasonable amount of time.

3 Simulated Annealing for Financial Investments

Simulated annealing is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems. The concept is based on the manner in which liquids freeze or metals recrystalize in the process of annealing. In an annealing process a melt, initially at high temperature and disordered, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state at T=0. The original Metropolis scheme was that an initial state of a thermodynamic system was chosen at energy E and temperature T, holding T constant the initial configuration is perturbed and the change in energy dE is computed. If

the change in energy is negative the new configuration is accepted. If the change in energy is positive it is accepted with a probability given by the Boltzmann factor *exp -(dE/T)*. This processes is then repeated sufficient times to give good sampling statistics for the current temperature, and then the temperature is decremented and the entire process repeated until a frozen state is achieved at T=0.

By analogy the generalization of this Monte Carlo approach to combinatorial problems is straight forward (Kirkpatrick et al. [5]). The current state of the thermodynamic system is analogous to the current solution to the combinatorial problem, the energy equation for the thermodynamic system is analogous to at the objective function, and ground state is analogous to the global minimum. The major difficulty (art) in implementation of the algorithm is that there is no obvious analogy for the temperature T with respect to a free parameter in the combinatorial problem. Furthermore, avoidance of entrainment in local minima (quenching) is dependent on the "annealing schedule", the choice of initial temperature, how many iterations are performed at each temperature, and how much the temperature is decremented at each step as cooling proceeds.

3.1 Algorithm

The general algorithm implemented includes a population instead of only one individual solution and a final condition of reaching an expected value (see Holland [4] and Michalewicz [7]).

```
Start
Define parameters (initial_investment, initial_temperature,
        long, cooling_factor, population_size, prodi-
        gies_percentage, elitist_percentage, move-
        ment_amplitud, change_percentage, replace-
        ment_percentage, cloning_probability)
Read the Scenario Tree
Generate polulation_size initial solutions
temperature = initial_temperature
Repeat
      For each solution of the population Do
          For 0 to long Do
                Select the nodes to modify
                !Depends on change_percentage
                Modify nodes
                !Depends on movement_amplitude
                Accept or refuse the modification
      Sort the solutions according to their means
      Select prodigy solutions
      !Depends on prodigies_percentage
      Assign an amplifying factor to each to each prodigy
      Select elitists solutions !as many as population_size
      Assign an amplifying factor to each elitist
      Select the poorest solutions to replace
      !Dependes on replacement_percentage
      For each poorest solution Do
```

```
                Generate a random z value
                If z < cloning_probability Then
                    Replace the solution with a clone
                Else
                    Replace the solution by "Average Idol"
            temperature = temperature*cooling_factor
        Until best_solution.objc_value = expected_mean
        Display solution
        End
```

We knew in advance the exact solution in the examples tested, and finished the algorithm when the optimal solution was found, in order to determine the parameters combination that worked better for this kind of problems. We show these parameters in Table 6.

Parameters utilized in the main algorithm presented are described in Table 2.

Table 2. Dictionary

Parameters	Explanation
initial_temperature	It is determined by the rule that the probability of accepting a movement is near 1 for each element of the neighborhood at the beginning of the algorithm.
long	Number of times that a modified solution is generated at the same temperature, i.e., the time that the system remains in each temperature to reach a stable state.
cooling_factor	Speed to which the temperature is reduced, diminishing the probability that "bad" solutions are accepted.
population_size	Number of solutions that will be conserved in the population, this number will remain constant in each generation.
prodigies_percentage	Number of solutions, that according to their quality, will be the parents of the following generation.
elitist_percentage	Percentage of best solutions found while they are preserved.
movement_amplitud	Percentage that determines the neighborhood around some value of the solution within which this value can be moved when doing modifications.
change_percentage	Percentage of nodes of the solution that will undergo modifications.
replacement_percentage	Percentage of solutions that due to its low quality will be replaced by others of better quality, or by means of clonation or by another method available called "Average Idol".
cloning_probability	Probability of replacing a solution by another one by means of clonation.
amplifying_factor	Assigned in linearly decreasing form, of the prodigies percentage.

4 Computational Examples

The procedure was tested with two examples. In both cases, 50 monthly periods (2002-2006) were used to build a scenario tree with four future stages. The scenario tree has two branches in each node. We considered five assets in the first example and ten assets in the second one. The data correspond to real assets in the BMV (Mexican bursaries market) and were obtained from Econom@tica (financial database). The examples were tested in a Pentium IV with 1.7 GHz and 256 Mb.

The initial amount M was of 100 money units for both examples and we considered a withdrawal of TW_t=0, for t=1,...,T. The scenario trees used for the example with 5 assets and the example with 10 assets are showed in Tables 3 and 4.

Table 3. Scenario Tree for 5 Assets

Id node	Asset1	Asset2	Asset3	Asset4	Asset5	Probability	Id Father node	Stage
0	0.747	0.684	0.769	0.673	0.696	1.000	-1	0
1	0.856	0.703	1.104	0.691	0.741	0.346	0	1
2	0.655	0.667	0.485	0.657	0.658	0.654	0	1
3	0.897	0.710	1.229	0.698	0.757	0.290	1	2
4	0.774	0.689	0.853	0.677	0.707	0.710	1	2
5	0.914	0.713	1.284	0.701	0.764	0.595	2	2
6	0.964	0.722	1.438	0.709	0.785	0.405	2	2
7	0.687	0.673	0.584	0.663	0.672	0.237	3	3
8	0.797	0.791	0.692	0.904	0.680	0.763	3	3
9	0.736	0.682	0.736	0.671	0.692	0.559	4	3
10	0.504	0.641	0.020	0.632	0.597	0.441	4	3
11	0.797	0.693	0.924	0.681	0.717	0.805	5	3
12	0.695	0.675	0.610	0.664	0.675	0.195	5	3
13	0.716	0.678	0.673	0.668	0.683	0.499	6	3
14	0.757	0.685	0.798	0.674	0.700	0.501	6	3

Table 4. Scenario Tree for 10 Assets

id_node	Benchmarks_182D	Benchmarks_28D	Benchmarks_364D	Benchmarks_7D	Benchmarks_91D	America_Movil_A	America_Movil_L	Ara_Con_A31sorcio	Arca_Embotelladora	Asureste_B	probability	id_father_node	Stage
0	0.67	0.60	0.69	0.59	0.61	3.34	3.39	1.94	1.00	2.62	1.00	-1	0
1	1.08	0.14	0.98	1.12	0.96	3.18	3.41	0.93	0.50	1.82	0.23	0	1
2	1.14	1.17	0.91	0.72	0.46	1.53	4.21	1.07	1.53	1.56	0.77	0	1
3	1.24	0.02	0.37	1.79	1.35	1.87	4.24	0.43	0.83	3.61	0.44	1	2
4	0.05	0.21	0.40	2.18	0.43	4.97	0.60	0.72	0.71	2.69	0.56	1	2
5	0.29	2.16	0.78	1.44	0.31	0.82	4.54	0.80	0.17	2.11	0.77	2	2
6	1.17	0.83	0.28	1.36	0.89	0.73	5.24	0.55	2.85	2.78	0.23	2	2
7	1.87	0.00	0.48	3.46	1.17	1.99	6.57	0.33	0.81	0.00	0.96	3	3
8	0.09	0.01	0.70	3.58	2.56	0.13	2.19	0.82	0.93	5.34	0.04	3	3
9	0.01	0.27	0.29	3.92	0.76	7.86	1.11	0.23	1.40	4.94	0.42	4	3
10	0.06	0.38	0.31	3.11	0.21	6.88	0.52	0.25	0.75	5.33	0.58	4	3
11	0.14	3.28	0.51	2.37	0.07	0.89	3.44	1.26	0.17	0.71	0.01	5	3
12	0.19	2.39	0.29	1.39	0.61	0.83	7.25	1.05	0.26	4.08	0.99	5	3
13	0.09	0.77	0.47	0.18	1.33	1.43	8.54	0.49	3.06	1.12	0.35	6	3
14	1.67	1.66	0.47	2.65	1.19	0.90	3.83	0.25	3.57	1.47	0.65	6	3

The amount of money obtained and reinvested (because annual withdrawals are 0 for every period, for both examples) in every asset, can be seen in tables 5 and 6.

Table 5. Results for example with 5 assets

Asset	Stage 0	Stage 1	Stage 2	Stage 3	Stage 4
Asset 1	0	0	0	189.22	328.54
Asset 2	0	176.90	0	149.42	248.58
Asset 3	100.00	0	214.81	260.01	484.31
Asset 4	0	0	0	73.36	135.50

Table 6. Results for example with 10 assets

Asset	Stage 0	Stage 1	Stage 2	Stage 3	Stage 4
America_Movil_A	0	0	450.79	1,498.64	12,428.14
America_Movil_L	100.00	439.44	1,755.93	10,594.33	81,967.64

The same models were solved using CPLEX V 9.0, in order to adjust the simulated annealing procedure parameters for obtaining the optimal solutions. The number of iterations was the iterations needed to reach the optimal value. Figures 2 and 3 show the optimal value convergence (the net redemption value encashed at the end of the horizon) with the simulated annealing procedure presented in this paper.

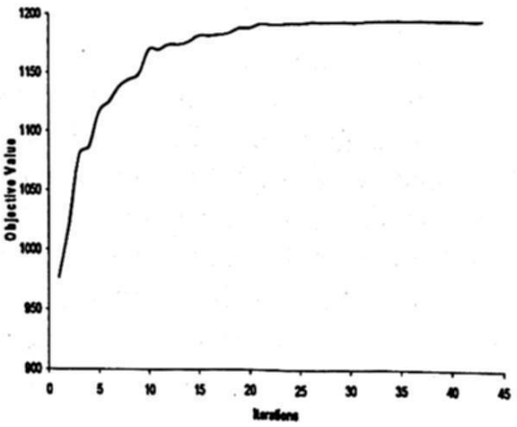

Fig. 2 Optimal value convergence for example with 5 assets

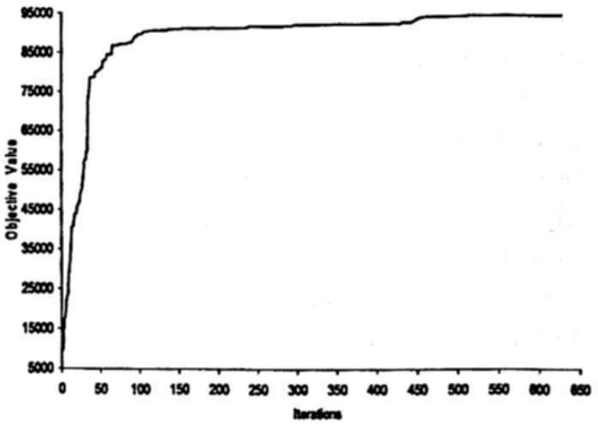

Fig. 3 Optimal value convergence for example with 10 assets

Because SA is a metaheuristic, there are many parameters to fix in order to turn it into an efficient algorithm. We have tested several parameters values to find more appropriate choices for this type of problems and the best parameters are presented in Table 7.

Table 7. Best parameter values

Parameters	5 Assets	10 Assets
initial temperature	450	1000
Cooling_factor	0.1	0.1
population_size	5	5
Long	5	5
prodigies_percentage	40%	40%
elitist_percentage	45%	45%
movement_amplitud	30	30
change_percentage	10%	10%
replacement_percentage	40%	20%
clonation_percentage	60%	40%

5 Conclusions

Portfolio selection gives rise to difficult optimization problems when realistic side constraints and variables are added to the basic model. Exact optimization algorithms cannot always deal efficiently with such complex models. It seems reasonable, therefore, to investigate the performance of heuristic approaches in this framework (Maringer et al. [6]).

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. The main objective of this paper was therefore to investigate the adequacy of simulated annealing for the solution of more realistic portfolio optimization models. The resulting algorithm allowed us to get the optimal net redemption value for the examples tested. The algorithm is able to handle more classes of constraints than many 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 (Green et al. [2]). 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. Nevertheless, the tailoring work required to fine-tune the parameters of the algorithm was rather delicate. Besides, introducing additional classes of constraints of new features in the model would certainly prove quite difficult again.

References

1. Crama, Y., Schyns, M.: Simulated annealing for complex portfolio selection problems. European Journal of Operational Research, Vol. 150, No. 3, November 2003, (2003) 546-571.
2. Green, R., Burton, H. (1992). When Will Mean-Variance Efficient Portfolios be Well Diversified? Journal of Finance, Vol. 5 (47), (1992) 1785-1809.
3. Gülpinar, N., Rustem, B., Settergren, R.: Optimization and Simulation Approaches to Scenario Tree Generation. Journal of Economics Dynamics and Control, Vol. 28, Issue 7 (2004) 1291-1315.
4. Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: The University of Michigan Press.
5. Kirkpatrick, S., Gelatt, C. D., Vechhi, P.M.: Optimization by simulated annealing. Science, Vol. 220, (1983) 671-680.
6. Maringer, D., Kellerer, H. Optimization of Cardinality Constrained Portfolios With an Hybrid Local Search Algorithm. Or Spectrum, Vol. 25(4), (2003) 481-495.
7. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996).
8. Osorio, M.A., Gulpinar, Settergren, R., Rustem, B.: Post-Tax Optimization w8. Osorio, M.A., Gulpinar, Settergren, R., Rustem, B.: Post-Tax Optimization with Stochastic Programming. European Journal of Operational Research, Vol. 157 (2004) 152-168.
9. Trippy, R., Lee, J.: Artificial Intelligence in Finance and Investing: state-of-the-art technologies for securities selection and portfolio management. IRWIN Professional Publishing, USA (1996).