

A Multiobjective Approach to the Portfolio Optimization Problem

Rubén Armañanzas, Jose A. Lozano

ISG - Department of Computer Science and Artificial Intelligence
University of the Basque Country - P.O. Box 649
20018 Donostia-San Sebastián, Spain
{ruben, lozano}@si.ehu.es

Abstract- The portfolio optimization problem uses mathematical approaches to model stock exchange investments. Its aim is to find an optimal set of assets to invest on, as well as the optimal investments for each asset. In the present work, the problem is treated as a multiobjective optimization problem. Three well-known optimization techniques greedy search, simulated annealing and ant colony optimization are adapted to this multiobjective context. Pareto fronts for five stock indexes are collected, showing the different behaviors of the algorithms adapted. Finally, the results are discussed.

1 Introduction

The big companies of monetary funds management are responsible for the investment of trillions of dollars annually. This money is invested on different products like pension funds, banking insurance policies, stock exchange assets, and other series of financial assets. The selection of an appropriate investment portfolio is a basic process for these financial companies. Although many of these decisions follow qualitative criteria, decisions based on quantitative approaches are appearing nowadays.

In the present problem, the financial assets are modeled by a probability distribution. This implies that the portfolio profit is completely described by the average individual profits of those assets; and the risk, by its total variance.

When investing money, companies are interested in obtaining the maximum profit of an investment set, which also minimizes the risk. However, the problem has several constraints. Usually, the number of assets a portfolio can contain is fixed at a constant. In the same way, when an asset is selected to invest in, there are minimum and maximum amounts of possible investments for that asset.

After fixing these parameters, a problem's solution is composed by a curve that compares risk and profit. For each risk level, it shows the maximum profit that can be obtained. This is a clear multiobjective design with two objective functions, one to be minimized and another to be maximized. The second problem's output is the asset set to invest on, and the adequate investment ratios for each one of them.

Here, we adapt three widely known optimization procedures to a multiobjective approach for the portfolio problem. Two of them belong to the local search paradigm: greedy search [1] and simulated annealing [2]. The third procedure is based on the ant colony heuristic paradigm [3]. Specific algorithms have also been designed for the common tasks of initialization and neighborhood computing.

Section 2 presents the complete mathematical modeliza-

tion of the portfolio problem. Section 3 covers the approaches of greedy search and simulated annealing for a multiobjective problem with two objective functions. Section 4 includes the ad-hoc implemented algorithms for initialization and neighborhood computing. The ant colony adaptation to this problem is shown in Section 5. The results obtained are shortly described and discussed in Section 6. Last, conclusions are exposed in Section 7.

2 Portfolio optimization

2.1 Multiobjective problems

Multiobjective problems [4, 5] are very common within the world of engineering optimization. In such contexts, not only must a unique objective function be optimized, but so must a set of functions belonging to the different problem's characteristics. Hence, an optimal solution does not exist, since it is not known how to compare two functions that come from completely different natures. Improving one of the objectives many times implies worsening some or all the other objectives. In this case, the idea is to search for a set of agreements between all the problem's objective functions. This way, the user of the final results will be the one that must select a certain solution or solution set of the ones found.

Let us introduce a briefly mathematical description of a multiobjective optimization problem:

Let

x_1, \dots, x_n the variables of the problem
 f_1, \dots, f_m the functions to optimize

We look for

$$\min f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n) \quad (1)$$

Subject to

$$g_1(x_1, \dots, x_n) \leq b_1 \quad (2)$$

\vdots

$$g_r(x_1, \dots, x_n) \leq b_r \quad (3)$$

A solution x is called a *nondominated solution* when there is no other solution that can improve the value of x for some objective function f_i without simultaneously degrading at least one of the other objective functions. On the basis of this concept, two solutions can be compared in a multiobjective problem, in such a way that:

given a multiobjective problem

$$\min_{x=(x_1, \dots, x_n)} f_1, \dots, f_m \quad (4)$$

a solution x' dominates another solution x'' if

$$\forall i \ f(x'_i) \leq f(x''_i) \quad \exists j \in 1, \dots, n \mid f(x'_j) < f(x''_j) \quad (5)$$

All the nondominated solution sets constitutes the Pareto optimal set. Grouping all the images of this Pareto optimal set generates a plot, often discontinuous, known as the *Pareto front*, or *Pareto border*. Its name refers to Vilfredo Pareto [6], who generalized these concepts in 1896.

All the techniques presented in this work try to find the different Pareto borders for each problem.

2.2 Mathematical modeling

Several different mathematical optimization approaches have been described for the portfolio optimization problem [7, 8]. In this study, a multinormal distribution modeling is chosen: the Markowitz model [9]. On the basis of this model, the restrictions and objective functions are defined.

Let

N	the available number of assets
μ_i	the expected mean of the i th asset
σ_{ij}	the covariance between the i th and the j th asset
K	the number of assets to invest ($K \leq N$)
ε_i	the minimum inversion ratio allowed in the i th asset
δ_i	the maximum inversion ratio allowed in the i th asset

We define the variable

$$z_i = \begin{cases} 1 & \text{if the } i\text{th } (i=1, \dots, N) \text{ asset is chosen} \\ 0 & \text{otherwise} \end{cases}$$

ω_i = money ratio ($0 \leq \omega_i \leq 1$) invested in the i th ($i=1, \dots, N$) asset

The optimization problem can be formulated in the following way:

$$\min \sum_{i=1}^N \sum_{j=1}^N \omega_i \omega_j \sigma_{ij} \quad (6)$$

$$\max \sum_{i=1}^N \sum_{j=1}^N \omega_i \mu_i \quad (7)$$

subject to

$$\sum_{i=1}^N \omega_i = 1 \quad (8)$$

$$\sum_{i=1}^N z_i = K \quad (9)$$

$$\varepsilon_i z_i \leq \omega_i \leq \delta_i z_i \text{ with } i = 1, 2, \dots, N \quad (10)$$

$$z_i \in \{0, 1\} \text{ with } i = 1, 2, \dots, N \quad (11)$$

Equation 6 minimizes the total variance (the risk) associated with the portfolio assets. Equation 7 maximizes the profit associated with the portfolio of assets. Equation 9 fixes the number of assets to invest in at K . Lastly, Equation 10 imposes the maximum and minimum inversions allowed for each asset.

3 Local search algorithms

3.1 Greedy search

The greedy local search [1] is one of the simplest technique used to solve an optimization problem. Its simplicity has a great advantage: it offers a solution very quickly. However, its main disadvantage is also due to its simplicity: when a local maximum or minimum value is reached, it is impossible to get out of it.

Starting with a random initial solution, its neighbor solutions are examined. The best neighbor becomes the new optimal solution. The process is repeated until no better solution is reachable.

Using the original design, the multiobjective modeling takes into account not only one evaluation function, but also the two objectives of the current problem.

Let

Ω	the feasible set of solutions, or search space
e	a problem's solution
$f(e)$	the score function of a solution e for Equation (6)
$g(e)$	the score function of a solution e for Equation (7)
$V(e)$	the neighbor solution set of e , or neighborhood of e

The algorithm implemented for the multiobjective greedy search is described in Figure 1.

```

choose an initial solution  $e_0 \in \Omega$ 
do
  for each  $e_i \in V(e_0)$ 
    if  $f(e_i) < f(e_0)$  and  $g(e_i) \geq g(e_0)$  or
        $f(e_i) \leq f(e_0)$  and  $g(e_i) > g(e_0)$  then
      replace  $e_0$  by  $e_i$ 
until no solution in  $V(e_0)$  dominates  $e_0$ 
 $e_0$  is returned as the solution

```

Figure 1: Greedy search biobjective pseudocode

When replacing the current search solution, the following policy has been implemented: once the neighborhood of the current solution is computed, the neighbors are visited, choosing as the new solution the first solution that dominates the current one. This is repeated until all the neighbors

have been visited. Therefore, for the same search iteration, several new solutions can be selected as the new one, only the last one of them being kept.

3.2 Simulated annealing

Simulated annealing [2] dates back to the statistical physics of the mid-1980s, between 1983 and 1985. Its physical foundation is the *annealing* of a solid, that is, the process of exposing a solid to high temperatures and leaving to cool down so slowly that, their particles look for the lowest energy positions.

Annealing can be considered an optimization problem as follows: a particle configuration is a solution to the problem; a minimum energy configuration corresponds to an optimal solution; the energy of a configuration is the objective function value; and, last of all, the temperature is the search control parameter.

Using this equivalence, an algorithm able to accept solutions that worsen the objective function is designed. It makes use of a probability function based on the Boltzman distribution. Initially, when the temperature is high, this probability is also high; as the algorithm advances in its iterations, this probability tends to zero. On the limit, the algorithm behaves like a local search, only accepting solutions that improve the current one.

In order to apply this technique, a series of annealing parameters must be defined, that is, an *annealing scheme* must be defined. Basically, an annealing scheme is composed of an initial and a final temperature, the temperature updating function, and the number of iterations that are made with the same fixed temperature (the chain's size).

In addition to the notation defined in Section 3.1, let:

$chain$	annealing control variable
c_k	current temperature value, for a given iteration k
α	the updating parameter for the temperature c_k
γ	the chain's size
$aleat$	a random number

The simulated annealing algorithm adapted to the present multiobjective problem is described in Figure 2.

The algorithm is a direct adaptation of the classic one. If a new solution dominates the current one, it is selected as the current solution. If it does not improve the current solution, it is possible to *anneal* both objective functions separately. Thus, some of the objectives can worsen, offering the possibility of reaching a better solution later. The stop criterion is reached when the present solution has not been modified at a fixed number Q of the main loop iterations.

4 Initialization and Neighborhood system

The search approaches presented in Section 3 make use of two common concepts: the initialization procedure and the neighborhood system. Both aspects are relevant to the

choose an initial solution $e_0 \in \Omega$

do

$chain = 0$

do

choose a solution $e \in V(e_0)$

$\delta = f(e_0) - f(e)$

$\epsilon = g(e_0) - g(e)$

if $\delta > 0$ **and** $\epsilon \leq 0$ **or**

$\delta \geq 0$ **and** $\epsilon < 0$ **then**

replace e_0 by e

else

choose $aleat \in [0, 1]$

if $\epsilon > 0$ **and** $e^{\frac{-\delta}{c_k}} > aleat$ **or**

$\delta < 0$ **and** $e^{\frac{\epsilon}{c_k}} > aleat$ **then**

replace e_0 by e

$chain = chain + 1$

until $chain = \gamma$

$c_{k+1} = \alpha \cdot c_k$

until e_0 does not change in Q loops

e_0 is returned as the solution

Figure 2: Simulated annealing biobjective pseudocode

search method's efficiency and, in this case, they are not trivial tasks. Thus, their implementation must be put forward in detail.

4.1 Portfolio initialization

Due to the nature of the portfolio problem and its formulation, the selection of an initial random solution is intrinsically an optimization problem. The investment restrictions of the problem (Equation 10), the possibility of varying the number of assets for each portfolio (Equation 9), as well as the fact of working with proportions in continuous domains (Equation 8), show that an initialization process is no a trivial task.

Moreover, an initialization process has to be as fast and random as possible, and it must also be a low computational cost procedure. Having in mind all these characteristics, a specific initialization algorithm has been implemented.

Let

\mathfrak{S}	an asset subset of the problem
a_i	the asset of i th index; $a_i \in \mathfrak{S}$
S	the initial solution bias
f_i	the deviation of the a_i asset investment ratio w.r.t. its high δ_i , or low ε_i restriction
m_i	the ω_i asset modifier for a_i

The algorithm randomly initializes K assets to invest

on, following the procedure defined in Figure 3.

```

choose  $K$  random assets,  $a_i$ , to initialize ( $\mathfrak{S}$  set)
for each  $a_i \in \mathfrak{S}$ 
    set a random inversion  $\omega_i$ , such that  $\varepsilon_i \leq \omega_i \leq \delta_i$ 
calculate the solution's leftover like

$$S = 1 - \sum_{i=1}^N \omega_i$$

if  $S \neq 0$  then
    for each  $a_i \in \mathfrak{S}$ 
        if  $S > 0$  then  $f_i = \delta_i - \omega_i$ 
        else  $f_i = \omega_i - \varepsilon_i$ 
    for each  $a_i \in \mathfrak{S}$ 

$$m_i = \frac{f_i}{\sum_{j=1}^K f_j}$$


$$\omega_i = \omega_i + S \cdot m_i$$


```

Figure 3: Initialization pseudocode

Initially, the algorithm randomly includes K assets in the initial portfolio. Next, and sampling a uniformly $[0,1]$ distributed random variable, random investment proportions ω_i are assigned to each previously selected asset. But the investment allowed for each asset must comply with the restriction of Equation 8. Because of this, the investments are normalized until they fulfill the restriction. This correction is made proportionally, so that the investment of each value increases or decreases in relation to the initial random distribution.

4.2 Neighborhood solution

One of the most important elements in a local search heuristic optimization approach is the neighborhood system defined. The neighbor system comprises the selection of a set of solutions, or neighborhood, *next to* a given one. From that given solution, a heuristic search procedure examines the neighborhood and makes the necessary search decisions.

For the portfolio problem, the distance between two portfolios does not have a clear definition. Trying to help the search algorithms, a neighborhood system that combines aspects of the two objective functions has been implemented.

Notice that the cardinality of the neighborhood system can slow down the search process. If the number of neighbors is too high, the objective function of a large set of solutions must be computed at each search iteration. Obviously, if the number of neighbors is low, the search process is not able to adequately explore the search space. Thus, the search process stops too early, reaching poor solutions. The system of neighbors proposed in Figure 4 has a size equal to the total number of assets, N , of the problem.

The algorithm first looks for the two assets that add the highest risk to the base solution. Next, it selects which one of them contributes less to the total portfolio profit. This

```

 $inc = 0$ ;  $neighbors = \emptyset$ 
given a solution  $e$ , calculate the asset  $a_i$  and  $a_j$  with the maximum risk associated
if  $\omega_{a_i} \cdot \mu_{a_i} < \omega_{a_j} \cdot \mu_{a_j}$  then  $worst = i$ 
else  $worst = j$ 
 $v = e$ 
 $c = \frac{\omega_{a_{worst}}}{N}$ 
do
     $inc = inc + c$ 
    if  $\omega_{a_{worst}} \neq 0$  then choose an asset  $a_k \neq a_{worst} \in v$ 
    else choose an asset  $a_k \neq a_{worst} \in (a_1, \dots, a_N)$ 
    if  $(\delta_{a_k} - \omega_{a_k}) \geq inc$  then
         $\omega_{a_k} = \omega_{a_k} + inc$ 
         $\omega_{a_{worst}} = \omega_{a_{worst}} - inc$ 
        add  $v$  to  $neighbors$ 
until  $|neighbors| = N$ 

```

Figure 4: Neighborhood pseudocode

asset will be the *pivot* asset from which all the neighborhood is generated.

Based on the total number of assets, N , the algorithm computes a coefficient c proportional to the pivot's investment. This coefficient is used to generate new solutions: randomly choosing assets, the pivot's investment is shared among them proportionally. In this sharing process, assets whose investment is zero can be selected. By this way, the neighborhood is a balance between exploitation and exploration.

5 Ant Colony approach

Using the artificial life paradigm known as *ant colony* metaheuristic (ACO), firstly proposed by Dorigo in the nineties [10], we tackle the *ad-hoc* approximation to the portfolio problem.

The ACO metaheuristic [3] is based on the collectivity characteristics of real ant colonies. Its main aim is to find a good-quality solution to a certain optimization problem, usually the minimum cost or shortest path that meets the problem's restrictions. This paradigm derives from a multiagent strategy: each ant individually builds a solution or a part of it, but always taking into account the solutions which the rest of the colony has previously reached. Communication is indirect: each ant modifies the global state of the problem, so every ant can notice these changes. This indirect communication occurs by means of *pheromones*. Pheromones are composed of a set of coefficients, one for each path or edge the ant can take. A pheromone has an evaporation ratio that makes it to decrease along the search iteration. When many ants choose the same path, this co-

efficient increases and constitutes the solution of one problem. All these briefly discussed components can be widely reviewed in [3].

As said, the ACO metaheuristic paradigm is mainly designed to detect a minimum cost path on a graph. This makes it unfeasible to compute a multiobjective solution directly by applying the ACO algorithm. A possible alternative to this limitation is to build different colonies, each one involved in the optimization of the functions that conform the multiobjective problem.

Thus, a complete connected graph is built: the n -th graph node corresponds to the n -th portfolio asset. The neighborhood of each node is composed of all the other nodes of the problem. At each main loop iteration, an ant is located at every node. This way, no asset has any advantage over the others. The ants behave synchronically, at each step choosing the next asset to visit. The number of steps allowed is equal to the number of assets to invest in, K . Hence, an ant's solution is comprised of the K steps made by its movements.

For each ant, the asset selection function has a fixed part a_{ij} and a variable part, that is, the pheromone and a random weight, $f(a_{ij} + \text{aleat} \cdot \tau_{ij})$. Similar to the *min-max ant systems* [11], a minimal (τ_{min}) and a maximal (τ_{max}) pheromone threshold is set out each edge in the graph. This technique tries to avoid the algorithm to quickly fall into local maxima, leaving the possibility of jumping to other solutions, obviously without behaving chaotically. Both values τ_{min} and τ_{max} are fixed experimentally after the initial runs of the model.

In order to compute the objective function for each colony, a prefixed investment value is arranged for those assets included in the portfolios, $w_i = \frac{\delta_i - \epsilon_i}{K}$.

Then, the problem is splitted into three different problems: three independent colonies with different costs for the node arcs are defined trying to cover three search space regions: high profit, low risk, and a trade-off between them. Therefore, the fixed component a_{ij} at each arc between the i -th and the j -th node includes:

1. The risk associated with both assets, σ_{ij}
2. The profit associated with the inclusion of the j asset, μ_j
3. A function that tries to balance both previous objectives, $\frac{\mu_i + \mu_j}{\sigma_{ij}}$

Let (12, 13, 14) be the evaluation functions for each colony. Each ACO algorithm has to minimize one of the

B_m ($m = 1, 2, 3$) objective functions defined as:

$$B_1 = \sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \quad (12)$$

$$B_2 = - \sum_{i=1}^N w_i \mu_i \quad (13)$$

$$B_3 = - \sum_{i=1}^N \sum_{j=1}^N w_i w_j \frac{\mu_i + \mu_j}{\sigma_{ij}} \quad (14)$$

Once the evaluation functions are computed, the best solution of each colony is chosen. If there is a draw among the evaluations of two different solutions, all of them are considered at that stage of the algorithm. The pheromone is updated on the basis of the following function:

$$\tau_{ij} = \frac{1}{B_m} + \tau_{ij} \text{ with } m \in \{1, 2, 3\} \quad (15)$$

Notice that the graph is visited in relation to the asset's position in the portfolio, from the lowest to the highest indexes. It is also necessary to set the pheromone evaporation parameter ρ , which is defined as:

$$\tau_{ij} = (1 - \rho) \tau_{ij} \quad (16)$$

After the convergence criterion is reached (for instance, a similar solution over a prefixed number of iterations), a secondary greedy search begins. Starting with the three asset sets selected, initialized with the prefixed investments proposed, three different multiobjective greedy searches (following the approach of Section 3.1) are performed. This is the stage that brings the pure multiobjective character to the whole modeling. The neighborhood algorithm presented in Section 4.2 can select neighbors not included on the ant's solution, so the search is not only limited to the asset sets found by the ants.

Summing up, the ACO approximation implemented brings a tuned set of assets to start with: three different starting points in the search space, one looking for high profit, another one looking for low risk, and the last one looking for a trade-off between them.

6 Experimental results

6.1 Data sets

The data sets used for the experimentation are available online and come from the OR-Library [12]. Five data sets compose the portfolio problem, named `port1` to `port5` respectively. Each data set corresponds to a different stock market of the world. The first index is the Hong Kong, *Hang Seng*, the second one is the German, *DAX 100*, the third one is the British *FTSE 100*, the fourth one is the U.S. *S&P 100*, and the fifth one is the Japanese *Nikkei 225*. The values included for each index were collected from March 1992 to September 1997. The data package contains the complete identification list of the assets included.

The data files are composed of 31, 85, 89, 98, and 225 assets, respectively. For each asset i , the average profit μ_i ,

and the individual risk σ_i is included. For each pair of assets i and j , the correlation ρ_{ij} between them is also included. The risk of investing in an asset having invested in another one simultaneously is modeled by the covariance between both. Then, the risk between two assets, i and j , is given by the expression:

$$\sigma_{ij} = \rho_{ij} \cdot \sigma_i \cdot \sigma_j \quad (17)$$

6.2 Experimental parameters

Among all the problem's restrictions presented in Section 2.2, there is one that calls our attention. We are interested in studying the behavior that the Pareto border shows when the number of assets to invest on increases. Usually, this number K is set at a fixed number. Here, we relax the condition and vary its value from the minimum, two assets, until the maximum, N . This way, we can see the evolution of the different Pareto borders, one for each K value.

In order to consider a search space as wide as possible, we set the values of the maximum and minimum investments at extreme values. For all the experiments, the values of ε_i and δ_i are 0.001 and 1.0, respectively.

For each different K value, the number of performed runs of the greedy search and the simulated annealing is 1,000. In the case of the ACO approach, and due to its three simultaneous colonies, this number is 333 for each colony.

Particularly for the annealing, the value of the initial temperature c_0 is set at 100, and the updating temperature parameter α is set at 0.95. The chain's size γ is also set at 100. The stop criterion is set at ten iterations ($Q=10$) without selecting any new solution.

Last, and concerning the ACO modelling, the stop criterion is fixed at 1,000 iterations with the same best solution. The colony parameters are set at 5,000 for τ_{max} , 100 for τ_{min} and 0.01 for ρ .

6.3 Results

A portfolio problem solution consists of the asset investment distribution, and the total amount of risk and profit that distribution achieves. The second component for all the runs forms the Pareto border of the multiobjective approach. The results computed for each run are filtered by deleting the dominated solutions. All the nondominated solutions for the three approaches shape the final Pareto borders. Figure 5 to 14 show, for each stock-exchange index, two of the Pareto borders obtained. Every approach contributes a different number of solutions to the borders, and, due to space limitations, no more plots can be included in this work. A selection of the most relevant Pareto borders of each data set is available on <http://www.sc.ehu.es/ccwbayes/members/ruben/cec2005/>.

In all data sets, from a certain K value onwards, the Pareto borders of each index take very similar forms. Moreover, when increasing the K parameter to its maximum values, the Pareto borders tend to 'roll-up' towards special points that we call *attract points*. These points fix an average profit and risk from which the algorithms are not able to quit. In Table 1, these points, shown in thousandths, are

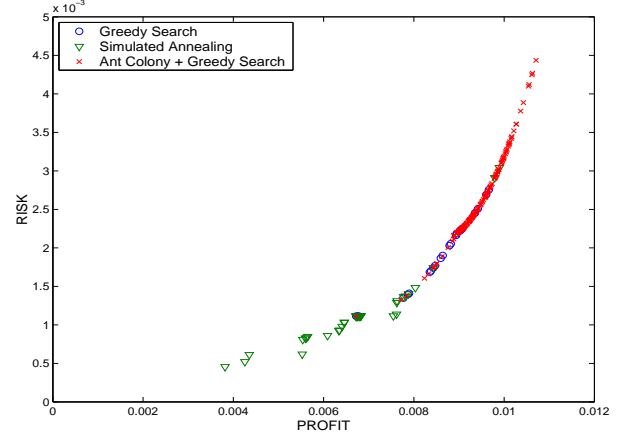


Figure 5: *Hang Seng* pareto border for $K = 2$

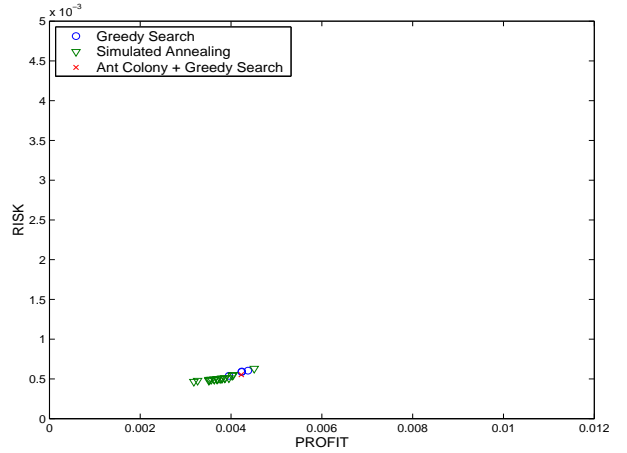


Figure 6: *Hang Seng* pareto border for $K = 31$

depicted. This is the confirmation of a natural behaviour: the more diverse the investment, the smaller the risk and the profit.

	Hang Seng	DAX100	FTSE100	S&P100
Profit	4	2	2.9	3
Risk	0.5	0.1	0.15	0.1

Table 1: *Attract points* for each index

Notice that for the *Nikkei 225* index, from a K value of 50 until its maximum, there are no positive profit solutions. As a consequence, the approaches detect no attract point. This is due to the expected mean values for these index asset profits. There are only 49, from the 225 assets included, with a positive expected profit.

The highest profits for all cases are obtained when the portfolio only includes two assets. The algorithm easily selects the two assets with the highest expected profit and, afterwards, it tries to fit the risks in the best way. Consequently, the risk associated with these solutions is also the greatest one of all the portfolios found. From a financial point of view this is also a natural tendency: when the investment is little diversified, the risk that an active falls down highly contributes to the profit uncertainty. The maximum values found for $K=2$ are included in Table 2. Values

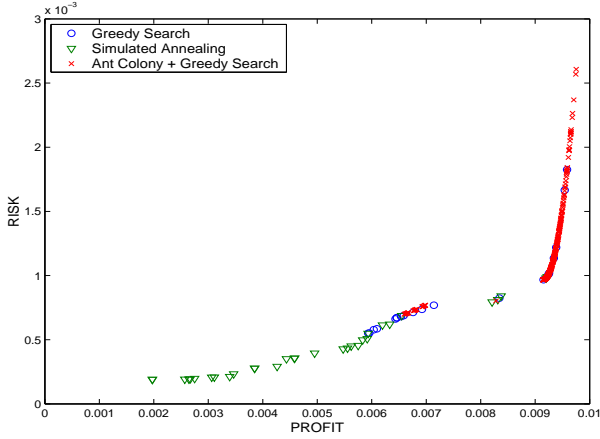


Figure 7: DAX 100 pareto border for $K = 2$

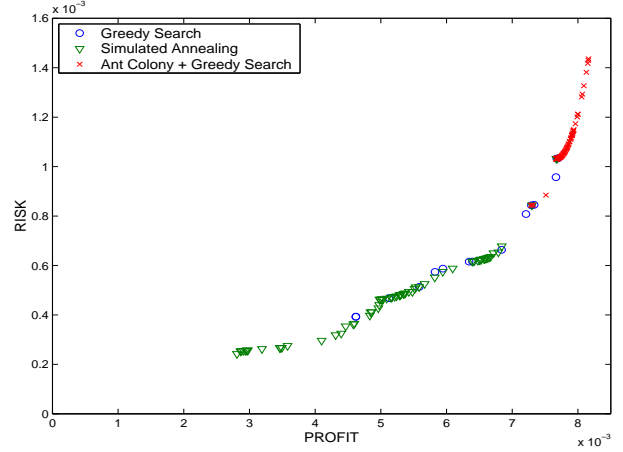


Figure 9: FTSE 100 pareto border for $K = 2$

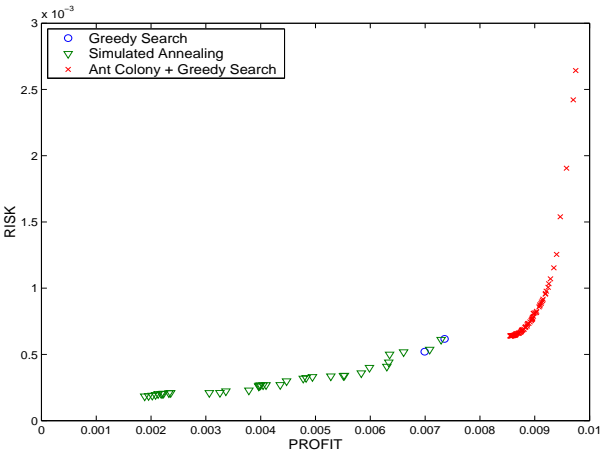


Figure 8: DAX 100 pareto border for $K = 3$

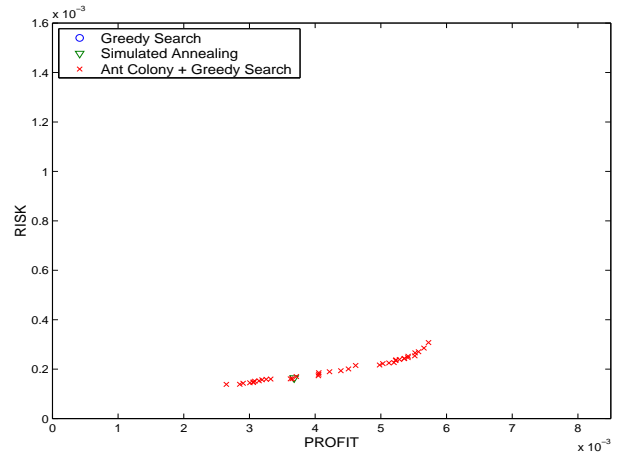


Figure 10: FTSE 100 pareto border for $K = 16$

are also shown in thousandths.

	Profit	Risk	Algorithm
Hang Seng	10.7	4.4	ACO
DAX 100	9.8	2.6	ACO
FTSE 100	8.2	1.4	ACO
S&P 100	9.2	2.7	ACO
Nikkei	0.4	0.15	ACO

Table 2: Maximum values for $K=2$ in each index

As can be seen in the figures, the nondominated results of ACO approach and the simulated annealing are outstanding. It is easy to check that the ACO technique fits Pareto's front parts better when risk and profit are high. On the other hand, the simulated annealing finds better solutions when both are lower. It is not possible to state which technique is better than the other. There are cases in which the simulated annealing almost found all the nondominated solutions (as in Figure 9) and others in which the ACO approach is the only technique capable of finding nondominated portfolios (see Figure 14).

The randomly initialized greedy search reaches a second place in this discussion. Its results are quite poor compared to the robustness of the two other approaches. It is capable of finding some complementary nondominated solutions

(see Figure 11), but its real power comes when combined with the ACO.

7 Conclusions

The multiobjective approach to the portfolio optimization problem proves itself to be very interesting. Usually, a solution for this problem is tackled by fixing a risk level and finding the best profit reachable. In our multiobjective model, no *a priori* constraint is given, neither for the risk nor for the profit. Consequently, an external decider has to fix the acceptance levels for each objective, and select non-dominated solutions that satisfy the criteria.

From an optimization point of view, we want to briefly add three other issues to the work tackled:

- For low K values, the number of solutions conforming the possible Pareto borders is quite high. Therefore, the decider has many options to choose from.
- Usually, the number of assets to invest is prefixed by hand, and these attract points are not identified. Covering all the possible assets to invest in has disclosed the attract points of each index. The K value indicates when the multiobjective makes more or less sense.

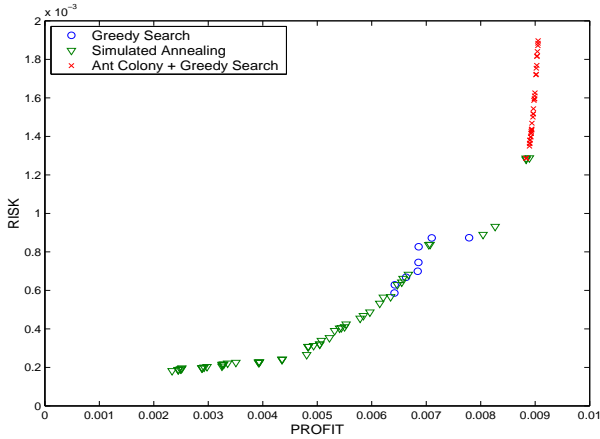


Figure 11: S&P 100 pareto border for $K = 3$

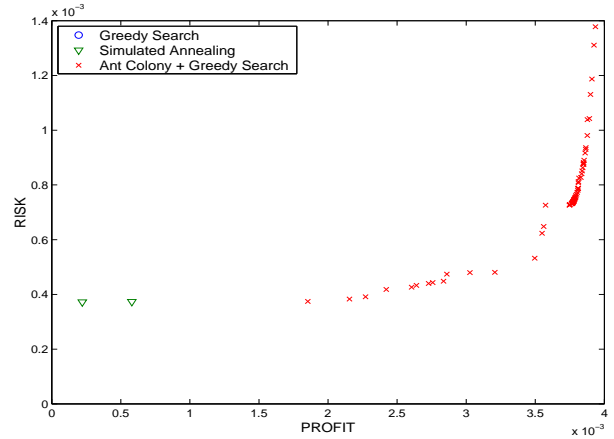


Figure 13: Nikkei 225 pareto border for $K = 3$

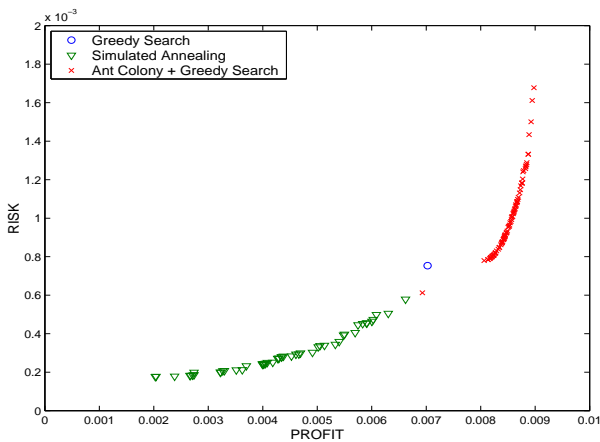


Figure 12: S&P 100 pareto border for $K = 4$

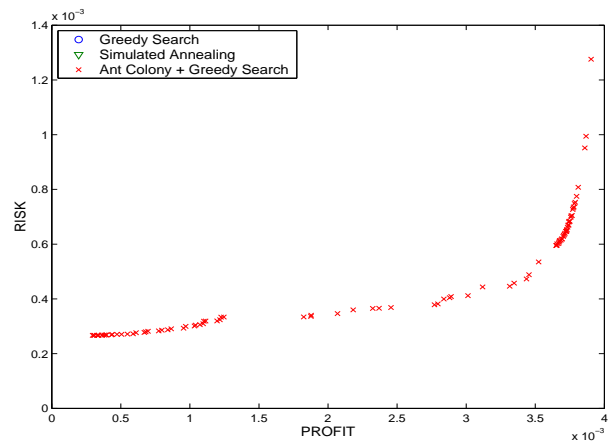


Figure 14: Nikkei 225 pareto border for $K = 4$

- ACO modeling has proven itself to be a very competitive approach. Not originally designed for multi-objective problems, it has found the most extreme solutions, exploring places of the search space that the other approaches have not been able to reach.

Bibliography

- [1] H. H. Hoos and T. Stützle, *Stochastic Local Search - Foundations and Applications*. Morgan Kaufmann, 2005.
- [2] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [3] M. Dorigo and T. Stützle, *Ant Colony Optimization*. The MIT Press, 2004.
- [4] C. A. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer Academic Publishers, May 2002.
- [5] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: John Wiley & Sons, 2001.
- [6] V. Pareto, *Cours D'Economie Politique*, F. Rouge, Ed., Lausanne, 1896, vol. I and II.
- [7] T. J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, "Heuristics for cardinality constrained portfolio optimisation," *Computers and Operations Research*, vol. 27, no. 13, pp. 1271–1302, 2000.
- [8] C. R. Harvey, J. C. Liechty, M. W. Liechty, and P. Müller, "Portfolio selection with higher moments," Duke University, 2004. Working paper.
- [9] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*. New York: Wiley, 1959.
- [10] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [11] T. Stützle and H. H. Hoos, "Improving the ant system: A detailed report on the max-min ant system," FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. AIDA-96-12, August 1996.
- [12] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol. 41, pp. 1069–1072, 1990.