Science Arena Publications
International Journal of Business Management
Available online at www.sciarena.com
2017, Vol, 2 (1): 91-111

A Decision Support System for Cardinality Constrained Portfolio Selection Problem

H. Dehghan Dehnavi[1], S.A. Seyed-Alagheband[1*], A.N. Sadrabadi[2]
[1]Department of Management, Yazd Branch, Islamic Azad University, Yazd, Iran
[2]Faculty of Industrial Management, Yazd University, Yazd, Iran

Abstract: *In this paper, we scrutinize the problem of finding the optimal possible tradeoff of risk against return in relation to the standard mean-variance portfolio selection model; including cardinality constraints that limit a portfolio to have a specified number of assets, and to impose limits on the proportion of the portfolio held in a given asset. Chang et al [2] were the first to introduce cardinality constrained portfolio optimization problem and included several algorithms to solve the problem. In this paper, a novel hybrid meta-heuristic algorithm based on genetic algorithm and simulated annealing is developed to improve the results obtained in [2]. Besides, a decision support system based on the hybrid meta-heuristic algorithm is developed to help the investor do tradeoffs between possible portfolios and decide on a suitable portfolio among the assets.*

Keywords: *portfolio selection, decision support system, hybrid heuristic algorithm*

INTRODUCTION

Markowitz assumed that asset returns follow a multivariate normal distribution. For a set of assets, the set of portfolios that offer the minimum risk for a given level of return form the efficient frontier. The portfolios on the efficient frontier can be obtained by quadratic programming (QP). The strengths of this approach are that QP solvers are available and efficient in terms of computing time. From a practical point of view, however, the Markowitz is too basic as it ignores many of the indisputable constraints such as trading constraints, size of the portfolio, etc. Including such constraints in the formulation results in an NP-hard problem which will not be solved in reasonable computational time.

Several researchers have attempted to attack this problem by a variety of techniques (decomposition, cutting planes, interior point methods, . . .), but there appears to be room for much improvement on this front. In particular, exact solution methods fail to solve large-scale instances of the problem. Therefore, in this paper, we investigate the ability of a hybrid meta-heuristic algorithm to deliver high-quality solutions for this model enriched by cardinality constraint and floor & ceiling constraint.

The remainder of the paper is as follows: Section 2 introduces the portfolio selection model that we want to solve. Section 3 presents a brief literature review of the portfolio selection problems. The proposed hybrid algorithm and its details are presented in section 4. Finally, section 5 illustrates a decision support system based on the developed algorithm.

2. Problem Definition

2-1 Unconstrained problem

The problem of selecting a portfolio among n assets was formulated by Markowitz in 1952. In this model, each asset is characterized by a return varying randomly with time. The risk of each asset is measured by the variance of its return. The basic assumption is that the investor aims to design a portfolio which minimizes risk while achieving a predetermined expected return, say R^*. Mathematically, the problem can be formulated as follows for any value of R^*:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \sigma_{ij}$$

Subject to:

$$\sum_{i=1}^{n} w_i \mu_i = R^*$$

$$\sum_{i=1}^{n} w_i = 1$$

$$0 \leq w_i \leq 1 \text{ for } i = 1. \dots . N$$

N : the number of available assets,
μ_i : the expected return of asset i (i=1,2,…,N),
σ_{ij} : the covariance between assets i and j (i=1,2,…,N; j=1,2,…,N),
R^* : the desired expected return.
w_i : the proportion ($0\leq w_i \leq 1$) held of asset i (i=1,2,…,N)

The first equation minimizes the total variance (risk) associated with the portfolio. The second one demonstrates that the portfolio has an expected return of R^*. The third equation ensures that the proportions add to one. This formulation is a simple nonlinear (quadratic) programming problem for which computationally effective algorithms exist so there is little difficulty in calculating the optimal solution for any particular data set.

2-2 Efficient frontier

By resolving the above formulation for different values of R^*, the efficient frontier is obtained, a smooth non-decreasing curve that represents the set of Pareto-optimal (non-dominated) portfolios. (Figure 1)

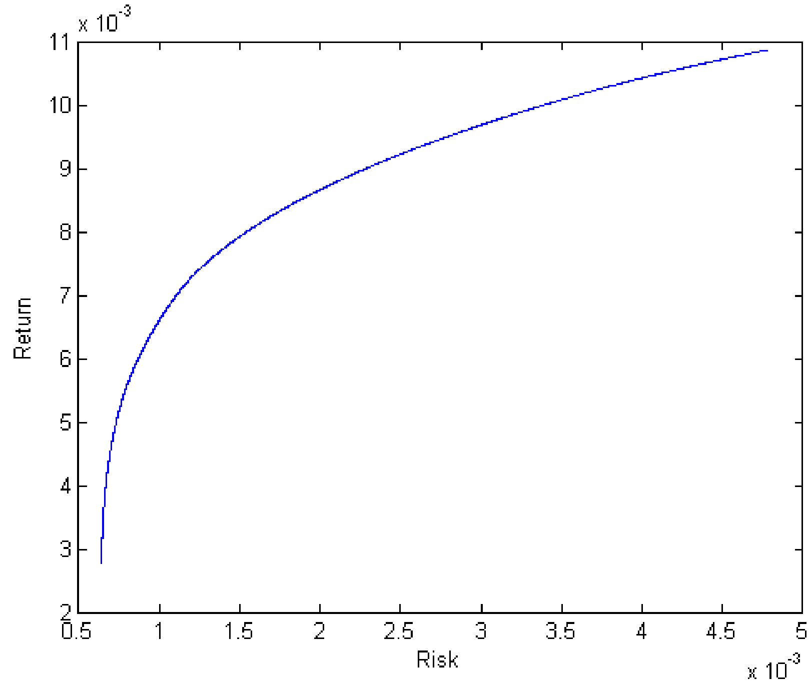


Figure 1: An example efficient frontier

For the unconstrained case it is standard practice to trace out the efficient frontier by introducing a weighting parameter λ ($0 \leq \lambda \leq 1$). The efficient frontier is obtained by varying the parameter λ.

$$\min \lambda(\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}) - (1-\lambda)(\sum_{i=1}^{n} w_i \mu_i)$$

Subject to:

$$\sum_{i=1}^{n} w_i = 1$$

$$0 \leq w_i \leq 1 \;\text{ for } i = 1. \ldots . N$$

The case $\lambda = 0$ represents maximize expected return (irrespective of the risk involved) and the optimal solution will involve just the single asset with the highest return. Furthermore, the case $\lambda = 1$ represents minimize risk (irrespective of the return involved) and the optimal solution will typically include a number of assets.

2-3 Constrained problem

In order to extend our formulation to the constrained case let:

K be the desired number of assets in the portfolio,
ε_i be the minimum proportion that must be held of asset i if any of asset i is held,
δ_i be the maximum proportion that can be held of asset i if any of asset i is held,

where we must have $0 \leq \varepsilon_i \leq \delta_i \leq 1$. In practice ε_i represents a minimum transaction level for asset i and δ_i limits the exposure of the portfolio to asset i. Introducing zero-one decision variables:

$$z_i = \begin{cases} 1 & \textit{if any of asset i is held} \\ 0 & \textit{otherwise} \end{cases}$$

the constrained portfolio optimization problem is expressed as below:

$$\min \lambda(\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}) - (1-\lambda)(\sum_{i=1}^{n} w_i \mu_i)$$

Subject to:

$$\sum_{i=1}^{n} w_i = 1$$

$$\sum_{i=1}^{N} z_i = K$$

$$\varepsilon_i z_i \leq w_i \leq \delta_i z_i \qquad 0 \leq w_i \leq 1 \text{ for } i = 1. \ldots . N$$

$$z_i \in [0.1] \qquad i = 1. \ldots . N$$

The first equation minimizes the total risk associated with the portfolio whilst the second one ensures that the portfolio has an expected return of R^*. The third equation ensures that the proportions add to one whilst the forth one ensures that exactly K assets are held. The fifth equation ensures that if any of asset i is held ($z_i = 1$) its proportion w_i must lie between ε_i and δ_i, whilst if none of asset i is held ($z_i = 0$) its proportion w_i is zero. The last equation is the integrality constraint.

3. Literature review

There have been several financial models for portfolio selection problems. The well-known Markowitz model [1] developed in 1952 attracted the attention of many researchers as it provided a simple model for the selection of best portfolios. Since then, a lot of research is done on portfolio selection based on Markowitz model.

The simple model, as mentioned before, is a QP which can be solved by exact algorithms. However, if more realistic cases are investigated, the problem, in many cases, is proved to be an NP-hard problem. Thus, many heuristic algorithms are developed to solve this sort of problems. Shapcott [21] was one of the first to use genetic algorithms for solving the portfolio selection problem. Genetic Algorithms are the most popular algorithms in the context of portfolio selection problems [2, 4, 15, 19, 21, 24, 25, 26, 27]. More recently, Suksonghong et al. developed several multi-objective genetic algorithms for solving portfolio optimization problems in the electricity market [30].

Many other algorithms like Simulated Annealing (SA), Tabu Search (TS), Ant Colony Optimization (ACO) and Neural Network are used for this kind of problems. Crama and Schyns [3] developed a Simulated Annealing algorithm to for a complex version of the problem. Rolland [20] used Tabu Search to solve a special type of problem. Doerner [7,8] presented an ant colony optimization to solve multi-objective portfolio selection problems. The neural networks are also popular in this context. Fernandez and Gomez [9] presented a neural network approach for this problem. Furthermore, a version of this problem considering skewness and using neural network was developed by Yu et al [28]. More recently, Ruiz and Suarez used memetic algorithm to address the portfolio selection problem with cardinality constraints and piecewise linear transaction costs [31].

Chang et al [2] also tackled our problem, using the cardinality constraint in the context of portfolio selection for the first time. They used 3 meta-heuristic algorithms - Genetic Algorithm, Tabu Search and Simulated Annealing - to solve this problem. This paper seeks to improve the results obtained in [2]. Table 1 demonstrates a summary of the literature review in portfolio selection context.

Author	Year	Proposed Algorithm(s)
Shapcott	1992	Genetic Algorithm
Arnone et al.	1993	Genetic Algorithm
Loraschi et al.	1995	Distributed Genetic Algorithm
Speranze	1996	Heuristic Algorithm
Rolland	1997	Tabu Search
Vedarajan	1997	Genetic Algorithm
Chang et al.	2000	Genetic Algorithm, Simulated Annealing, Tabu Search
Wang et al.	2001	Multi-Objective Genetic Algorithm
Doerner et al.	2001	Ant Colony Optimization
Maringer	2001	Ant Colony Optimization
Crama and Schyns	2003	Simulated Annealing
Kellerer and Maringer	2003	Hybrid Local Search
Maringer and Winker	2003	Memetic Algorithm
Doerner et al.	2004	Ant Colony Optimization
Fernandez and Gomez	2005	Neural Network
Kendall and Su	2005	Particle Swarm Optimization
Gomez et al.	2006	Hybrid Search
Mous et al.	2006	Comparison Between Genetic Algorithm and Particle Swarm Optimization
Yan et al.	2007	Hybrid Genetic Algorithm and Particle Swarm Optimization
Lin and Liu	2008	Fuzzy Genetic Algorithm
Yu et al.	2008	Neural Network
Deng et al.	2012	Particle Swarm Optimization
Suksonghong et al.	2014	Multi Objective Genetic Algorithms
Ruiz and Suarez	2017	Memetic Algorithm
Kumar and Mishra	2017	Artificial Bee Colony Algorithm

Table 1: A summary of literature review

Recent studied portfolio selection problems are mostly heading in two directions: fuzzy problems [32,33] and multi-objective algorithms [30]. But few articles have studied on the development of Decision Support Systems in portfolio selection context. Recently, Jalota et al. proposed a decision support system for portfolio selection with uncertain parameters [34].

4. Proposed hybrid algorithm

A version of problem similar to our case was first proposed by Chang et al [2]. They presented 3 heuristic algorithms: Genetic Algorithm, Tabu Search and Simulated Annealing. The results obtained in their paper demonstrate that improvements can be achieved through developing more effective algorithms. In this paper, a genetic algorithm is presented to develop a good solution for each λ. The solutions, then, will be improved by a simulated annealing algorithm.

4-1 Evaluation

Both the GA and SA algorithms require a function to make the solutions feasible and also evaluate the solutions. Therefore, the evaluation function is first described.

4-1-1 Chromosome representation

The chromosomes defined have 2 distinct parts, a set of K distinct assets and K real numbers s_i. s_i can be interpreted as the share of the free portfolio proportion $(1-\sum \varepsilon_j)$ associated with asset i. After the evaluation, another line is added showing the w_i related to each asset i.

4-1-2 Chromosome evaluation

To evaluate the chromosome, the w_i related to each asset i should be calculated. Not all possible chromosomes correspond to feasible solutions (because of the constraint relating to the limits on the proportion of an asset that can be held). However, when evaluating each solution, the simple procedure was used in order to try and ensure that the evaluated solution was feasible.

To explain our representation and Algorithm 1 further suppose that we have N=10, K=2 and ε_i=0.1 for any i. One GA solution might therefore be Q= {3, 7} and {s_3=0.9, s_7=0.5}. This means that assets 3 and 7 are in the portfolio. The free portfolio proportion is$(1-\sum \varepsilon_j) = 0.8$, since each of the two assets must have a proportion in the portfolio of at least 0.1. Hence we interpret this GA representation of {s_3=0.9, s_7=0.5} to mean that the share of the free portfolio proportion devoted to asset 3 is $s_3/(s_3+s7)$ =0.9/1.4=0.6429. Hence the proportion w_3 associated with asset 3 in the portfolio is given by 0.1+0.6429(0.8), i.e. the minimum proportion plus the appropriate share of the free portfolio proportion, hence w_3=0.6143. Similarly the proportion w_7 associated with asset 7 is $w_7=0.1+ (s_7/ (s_3+s_7))\ 0.8=0.3857$. Note that these values for w_3 and w_7 both satisfy the lower proportion limits and sum to one.

In this algorithm, we can ensure that the constraints relating to the lower limits ε_i are satisfied in a single algorithmic step. However we need an iterative procedure to ensure that the constraints relating to the upper limits δ_i are satisfied. Thus, in the cases which w_i of asset i is over δ_i, w_i are set to δ_i and the remaining w_i are calculated again without the consideration of such assets. After the floor and ceiling constraints are satisfied, the modified chromosome is evaluated and its fitness is obtained.

4-2 Genetic Algorithm

The proposed GA, as the main framework of the algorithm, plays an important role in producing good initial solutions to be improved by the SA algorithm. Here, we introduce the main parts of the GA.

4-2-1 Selection of parents

In this problem, parents are chosen by binary tournament selection which works by forming two pools of individuals, each consisting of two individuals drawn from the population randomly. The individuals with the best fitness, one taken from each of the two tournament pools, are chosen to be parents.

4-2-2 Crossover and mutation operators

Children in the GA are generated by uniform crossover. In uniform crossover, two parents have a single child. If an asset i is present in both parents, it is present in the child (with an associated value s_i randomly chosen from one or other parent). If an asset i is present in just one parent, it has probability 0.5 of being present in the child.

Children are also subject to mutation, multiplying by 0.9 or 1.1 (chosen with equal probability) the value ($\varepsilon_i + s_i$) of a randomly selected asset i. This mutation corresponds to decreasing or increasing this value by 10%.

4-2-3 Verification of the size of portfolio

After the child is generated, it may contain more than or less than K assets. If it has more than K assets, the assets with the least w_i are eliminated until the portfolio contains exactly K assets. If it has less than K assets, the assets which were in the parents but are not in the child are used to fulfill the empty places of the portfolio. If no such assets are available, assets which are not included in the portfolio are randomly added in the portfolio with $s_i = 0$.

4-2-4 Algorithm procedure

- The inputs are:
  - Limits matrix (bounds)
  - Mean and variance matrix (meanvar)
  - Correlation matrix (corr)
- For E λ, do the following:
  - Generate a random population with size 'popsize'
  - Evaluate the population with 'evaluate' function
  - For 'iteration' iterations, do the following:
    - Select the parents with 'binary tournament' method
    - Perform the crossover operator for reproduction
    - Perform mutation on the produced child
    - Verify that the child has exactly K assets
    - Evaluate the child
    - (improve the child with 'SA' function)
    - Update the population (the child should replace the worst member of population)
  - Update λ
- end

4-3 Simulated Annealing

In this paper, the SA heuristic is used as a neighboring mechanism to improve the GA solutions. The SA modules can be described as follows.

4-3-1 Moving operator

The move operator corresponds to taking all assets present in the portfolio of K assets and multiplying their values by 0.9 and 1.1. This means that the number of neighbors which we need to evaluate is 2K.

4-3-2 Algorithm procedure

- The inputs are:
  - Initial solution obtained from GA
  - Limits matrix (bounds)
  - Mean and variance matrix (meanvar)
  - Correlation matrix (corr)
  - λ and the best fitness for it
  - Number of temperature decrease (step)
  - Iterations in each temperature (MaxIter)
  - Decrease ratio (α)
- do for 'MaxIter' iterations:
  - Select a random asset i from the chromosome and multiply its w_i in 0.9 or 1.1
  - Evaluate the new chromosome with 'evaluate' function
  - If the fitness of new chromosome is better, it replaces the current solution, otherwise, the replacement is made with the probability of $e^{(-\Delta f/temp)}$
- Decrease the temperature with 'α' ratio (T=α*T)
- end

4-4 Test problem

To test our heuristic algorithm, we examined three test data sets used in Chang et al [2]. We considered the Hang Seng (Hong Kong), DAX 100 (Germany) and S&P 100 (USA).

All of the test problems solved in this paper are available at http://mscmga.ms.ic.ac.uk/.

4-5 MPE[1] calculation

The results should be compared to the UEF. Figure 2 shows a sample solution and UEF. Suppose that (x_i, y_i) is the discrete (x-coordinate: standard deviation, y-coordinate: return) values on the UEF. For a portfolio with (x^*, y^*), let j correspond to $y_j = \min [y_i \mid y_i \geq y^*]$ and k correspond to $y_k = \max [y_i \mid y_i \leq y^*]$ (i.e. y_j and y_k are the closest y-coordinates bracketing y^*). Simple geometry enables us to say that the value x^{**} associated with the x-direction linearly interpolated point on the UEF with $y=y^*$ (i.e. looking horizontally) is $x^{**} = x_k + (x_j - x_k)[(y^* - y_k)/(y_j - y_k)]$. A convenient percentage deviation error measure for this direction is then $| 100(x^* - x^{**})/x^{**} |$ (note here that no value is calculated if either j or k do not exist).

In a same way, y^{**} associated with the y-direction linearly interpolated point on the UEF with $x=x^*$ (i.e. looking vertically) is $y^{**} = y_k + (y_j - y_k)[(x^* - x_k)/(x_j - x_k)]$. A convenient percentage deviation error measure for this direction is then $| 100(y^* - y^{**})/y^{**} |$ (note here that no value is calculated if either j or k do not exist).

[1] Mean Percentage Error

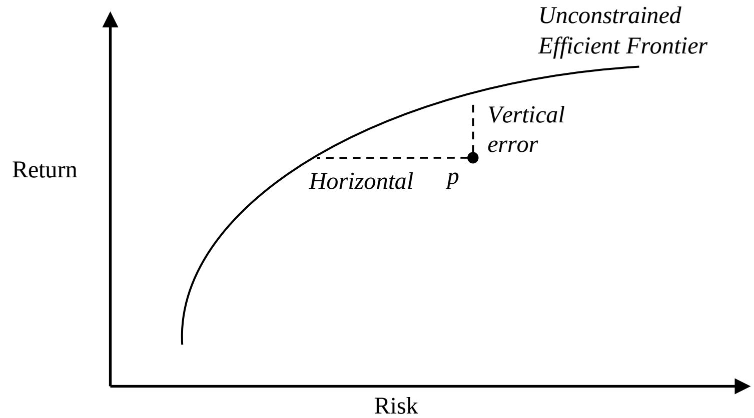


Figure 2: MPE calculation

4-6 The hybrid algorithm

The hybrid genetic algorithm and simulated annealing heuristic can now be developed. First, the genetic algorithm initializes a good feasible solution. The solution is then improved by the stochastic simulated annealing algorithm. By the end of the process, a set of solutions for each λ is obtained. The result, MPE, is calculated by making a comparison between the solutions and UEF.

4-7 Computational results

In this section, we present computational results for the hybrid heuristic algorithm we have presented above. Note here that all of the computational results presented in this section are for our hybrid heuristic as coded in MATLAB and run on a Pentium 4 computer with 2028 MB RAM and 4.00GHz CPU.

To achieve the best results, first, a parameter setting and tuning is performed. We examined 50 different λ values. With regard to the number of iterations T, we used T = 3000 for GA heuristic and T=2*N (N is the number of assets) for SA heuristic. We examined the model in two cases: unconstrained and constrained cases. We used a population size of 20 in the first case and 150 in the second one for GA. We also used 150 steps of temperature decrease in the unconstrained case and 100 steps in the constrained case for SA heuristic.

If K equals the number of all assets and 'bounds' are between 0 and 1, it would be the unconstrained case, but for constrained case, we used K=10 and bounds between 0.01 and 1 to be able to compare our results with test problems. Table 2 presents the results.

Index	**Number of assets in portfolio (K)**	**Mean Percentage Error (%)**	**Time (seconds)**
Hang Seng	31	0.0062	181.86
	10	1.1287	182.37
DAX	85	0.0074	390.43
	10	2.5181	233.83
S&P	98	0.0473	833.13
	10	2.8685	275.86

Table 2: Results

Here, we compared our results with the results obtained from [2] shown in Table 3. It is clear that out hybrid heuristic algorithm is capable of achieving better results in comparison with the individual algorithms presented in [2] and mostly improves the results in terms of error and running time of the algorithm. A tradeoff between unconstrained efficient frontier (UEF) and our unconstrained results obtained from S&P is illustrated in Figure 3.

Index	N	Chang et al. [2] GA Heuristic	Time (seconds)	Hybrid GA.SA Heuristic	Time (seconds)
Hang Seng	31	0.0202	621	0.0062	181.86
DAX	85	0.0136	10332	0.0074	390.43
S&P	98	0.0084	15879	0.0473	833.13

Table 3: Results for the unconstrained case

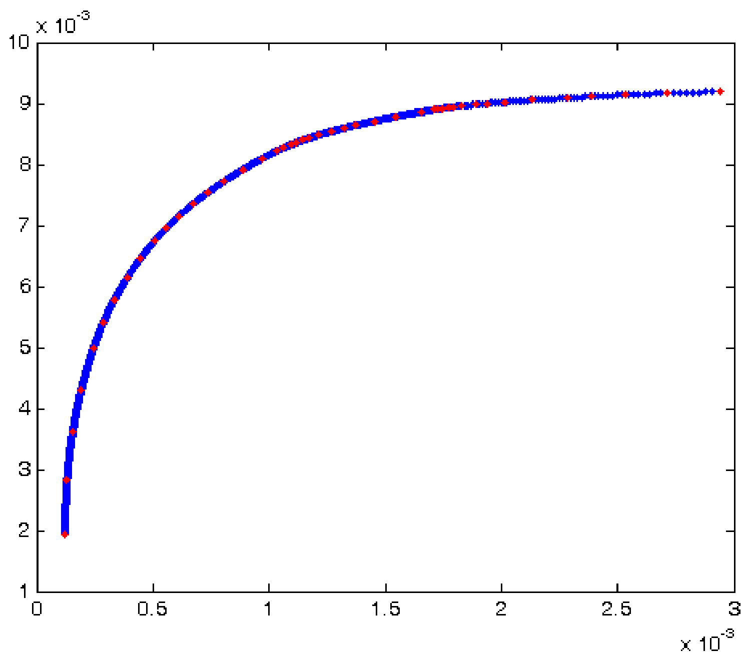


Figure 3: S&P tradeoff curve for unconstrained case

For the constrained case, however, the algorithm is not as capable as it was in unconstrained case and cannot dominate the results obtained from [2] as shown in Table 4. Of course, there is no reason to verify the results of [2], but unfortunately, we could not validate our hybrid heuristic algorithm in the constrained case. A tradeoff between unconstrained efficient frontier (UEF) and our constrained results obtained from S&P is illustrated in Figure 4.

Index	N	Chang et al. [2] GA Heuristic	Time (seconds)	Hybrid GA.SA Heuristic	Time (seconds)
Hang Seng	31	1.0974	172	1.1287	182.37
DAX	85	2.5424	544	2.5181	233.83
FTSE	89	1.1076	638	2.8685	275.86

Table 4: Results for the constrained case

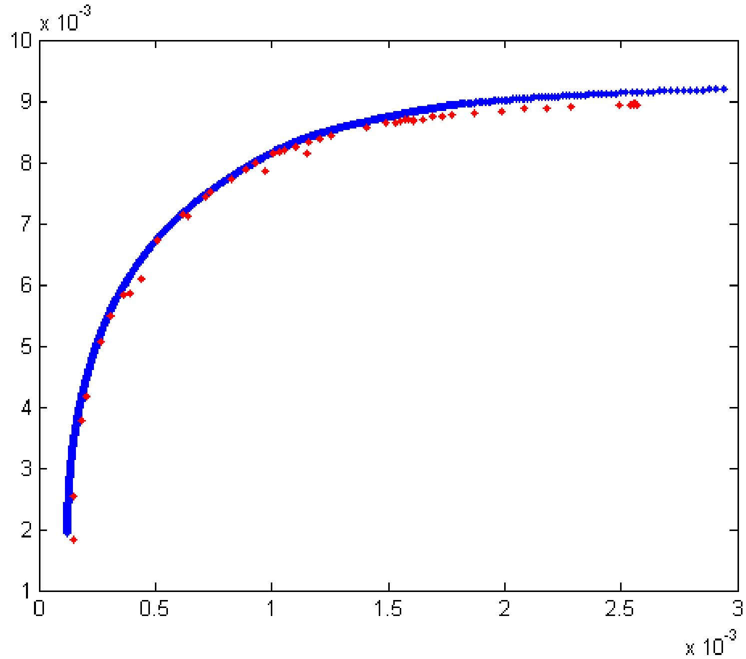


Figure 4: S&P tradeoff curve for constrained case (K=10)

Note here that the constrained case is compared to an unconstrained efficient frontier as there are no exact algorithms to solve it and the constrained results are better if their distance from the unconstrained efficient frontier get smaller.

Here, the decision maker has an explicit view of the available possibilities and can do tradeoffs between different situations, depending on which range of risk or return he/she can bear. To provide a software package for such decision makers, in the next section, a Decision Support System is presented.

5. Decision Support System for Portfolio Selection

As mentioned earlier, many investors face the problem of selecting an appropriate portfolio of assets in which to invest. To solve this problem, many systems have been generated. But, the point is that the problem of portfolio selection is so much developing and has been enriched by more realistic constraints. Therefore, the model base of those systems should be updated in order to be able to effectively correspond to the needs of such investors.

Any Decision Support System has 3 parts: Database, Model Base and Graphical User Interface (GUI). The Model Base is presented in Section 4; enriched by the hybrid algorithm. Here, we first present our prototyping approach and then, present the remaining modules of our system.

5-1 Prototyping method

Prototyping is a small-scale version of the system under investigation. To use prototyping, we should be familiar with the prototyping methods. There are 4 types of prototyping approaches:

1. Illustrative or throwaway: This prototype is designed for the purpose of illustration and gaining feedback and If user does not like the prototype, it will be thrown away an a new prototype may be designed;

2. Simulated Prototyping (Step Forward Prototyping): Provides models that behave as if they were parts of the desired information systems. This model is interactive, since the model can be refined and enhanced.

3. Functional Prototyping: This method is similar to Similar to the simulated methodology, but in contrast, it provides models that represent a more complete set of system functions.

4. Evolving Prototyping: This method starts from a small-scale system and then, it evolves into the final system by continuously adding new features and upgrading the existing features.

In this case, a throwaway approach is used, because here, it is easier and more user-oriented to construct a small comprehensive prototype than a phase-to-phase one. A throwaway procedure is shown in Figure 5.

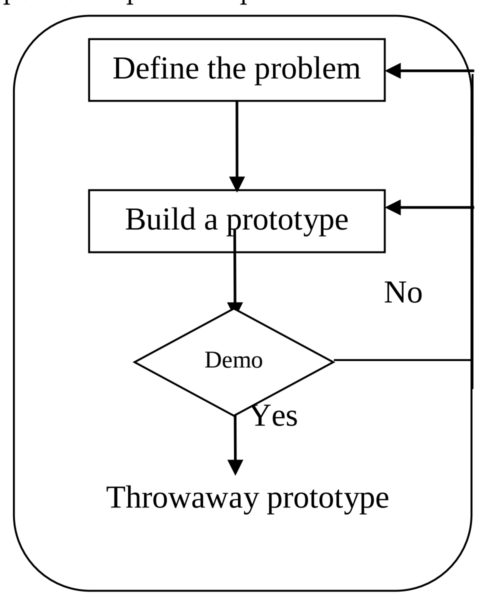
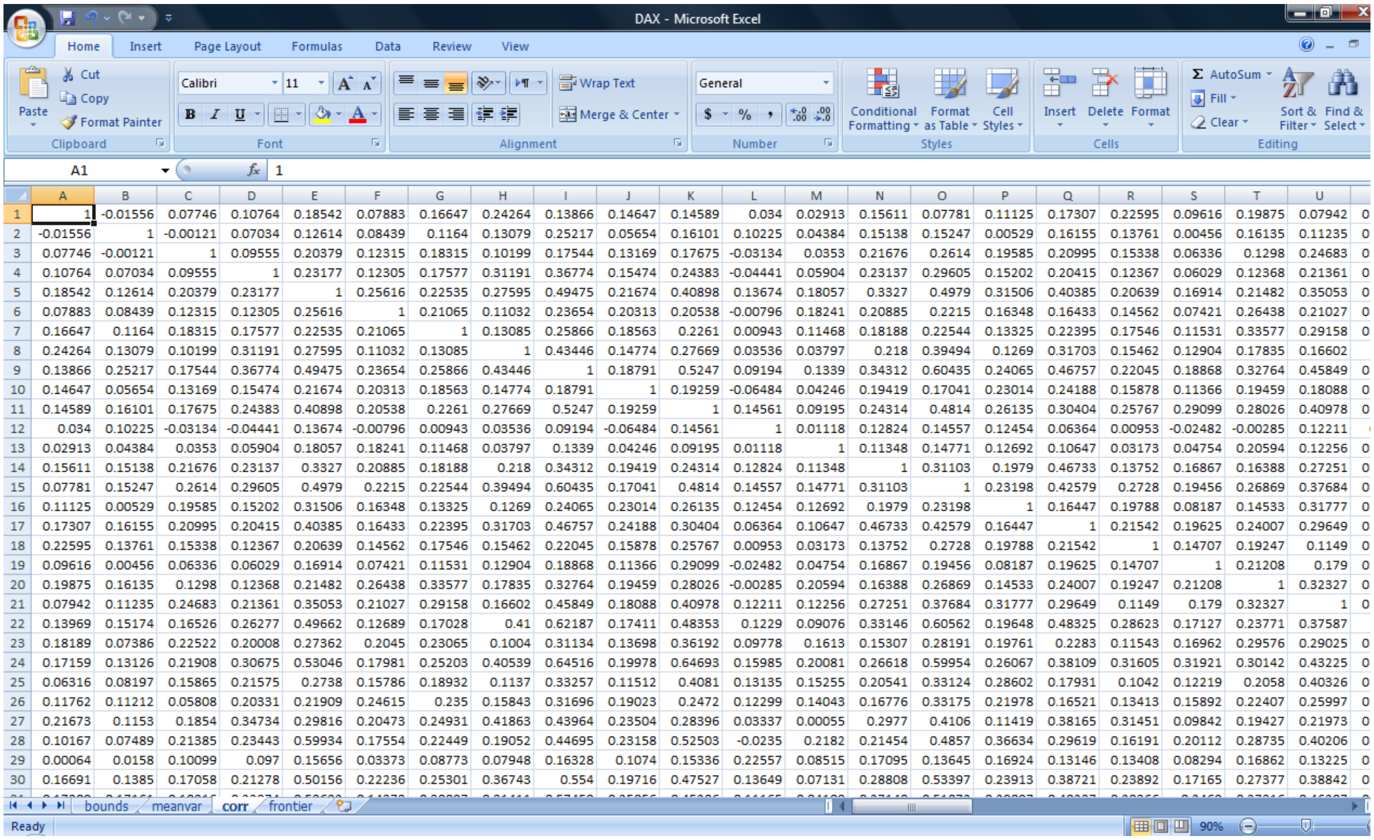


Figure 5: Throwaway prototype

5-2 Database

The databases used in this paper are three excel workbooks for each of three data sets. Each workbook has four excel sheets: the first sheet contains the ‘bounds’ matrix which is the floor and ceiling constraint. The mean and variance of each asset is included in the second sheet marked as ‘meanvar’ sheet. The third sheet has the correlation matrix and the forth one includes the numbers which when plotted, we have the UEF. Those are all the data we need to use in this paper. Figure 6 demonstrates the 'correlation' sheet, DAX100 database.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	1	-0.01556	0.07746	0.10764	0.18542	0.07883	0.16647	0.24264	0.13866	0.14647	0.14589	0.034	0.02913	0.15611	0.07781	0.11125	0.17307	0.22595	0.09616	0.19875	0.07942
2	-0.01556	1	-0.00121	0.07034	0.12614	0.08439	0.1164	0.13079	0.25217	0.05654	0.16101	0.10225	0.04384	0.15138	0.15247	0.00529	0.16155	0.13761	0.00456	0.16135	0.11235
3	0.07746	-0.00121	1	0.09555	0.20379	0.12315	0.18315	0.10199	0.17544	0.13169	0.17675	-0.03134	0.0353	0.21676	0.2614	0.19585	0.20995	0.15338	0.06336	0.1298	0.24683
4	0.10764	0.07034	0.09555	1	0.23177	0.12305	0.17577	0.31191	0.36774	0.15474	0.24383	-0.04441	0.05904	0.23137	0.29605	0.15202	0.20415	0.12367	0.06029	0.12368	0.21361
5	0.18542	0.12614	0.20379	0.23177	1	0.25616	0.22535	0.27595	0.49475	0.21674	0.40898	0.13674	0.18057	0.3327	0.4979	0.31506	0.40385	0.20639	0.16914	0.21482	0.35053
6	0.07883	0.08439	0.12315	0.12305	0.25616	1	0.21065	0.11032	0.23654	0.20313	0.20538	-0.00796	0.18241	0.20885	0.2215	0.16348	0.16433	0.14562	0.07421	0.26438	0.21027
7	0.16647	0.1164	0.18315	0.17577	0.22535	0.21065	1	0.13085	0.25866	0.18563	0.2261	0.00943	0.11468	0.18188	0.22544	0.13325	0.22395	0.17546	0.11531	0.33577	0.29158
8	0.24264	0.13079	0.10199	0.31191	0.27595	0.11032	0.13085	1	0.43446	0.14774	0.27669	0.03536	0.03797	0.218	0.39494	0.1269	0.31703	0.15462	0.12904	0.17835	0.16602
9	0.13866	0.25217	0.17544	0.36774	0.49475	0.23654	0.25866	0.43446	1	0.18791	0.5247	0.09194	0.1339	0.34312	0.60435	0.24065	0.46757	0.22045	0.18868	0.32764	0.45849
10	0.14647	0.05654	0.13169	0.15474	0.21674	0.20313	0.18563	0.14774	0.18791	1	0.19259	-0.06484	0.04246	0.19419	0.17041	0.23014	0.24188	0.15878	0.11366	0.19459	0.18088
11	0.14589	0.16101	0.17675	0.24383	0.40898	0.20538	0.2261	0.27669	0.5247	0.19259	1	0.14561	0.09195	0.24314	0.4814	0.26135	0.30404	0.25767	0.29099	0.28026	0.40978
12	0.034	0.10225	-0.03134	-0.04441	0.13674	-0.00796	0.00943	0.03536	0.09194	-0.06484	0.14561	1	0.01118	0.12824	0.14557	0.12454	0.06364	0.00953	-0.02482	-0.00285	0.12211
13	0.02913	0.04384	0.0353	0.05904	0.18057	0.18241	0.11468	0.03797	0.1339	0.04246	0.09195	0.01118	1	0.11348	0.14771	0.12692	0.10647	0.03173	0.04754	0.20594	0.12256
14	0.15611	0.15138	0.21676	0.23137	0.3327	0.20885	0.18188	0.218	0.34312	0.19419	0.24314	0.12824	0.11348	1	0.31103	0.1979	0.46733	0.13752	0.16867	0.16388	0.27251
15	0.07781	0.15247	0.2614	0.29605	0.4979	0.2215	0.22544	0.39494	0.60435	0.17041	0.4814	0.14557	0.14771	0.31103	1	0.23198	0.42579	0.2728	0.19456	0.26869	0.37684
16	0.11125	0.00529	0.19585	0.15202	0.31506	0.16348	0.13325	0.1269	0.24065	0.23014	0.26135	0.12454	0.12692	0.1979	0.23198	1	0.16447	0.19788	0.08187	0.14533	0.31777
17	0.17307	0.16155	0.20995	0.20415	0.40385	0.16433	0.22395	0.31703	0.46757	0.24188	0.30404	0.06364	0.10647	0.46733	0.42579	0.16447	1	0.21542	0.19625	0.24007	0.29649
18	0.22595	0.13761	0.15338	0.12367	0.20639	0.14562	0.17546	0.15462	0.22045	0.15878	0.25767	0.00953	0.03173	0.13752	0.2728	0.19788	0.21542	1	0.14707	0.19247	0.1149
19	0.09616	0.00456	0.06336	0.06029	0.16914	0.07421	0.11531	0.12904	0.18868	0.11366	0.29099	-0.02482	0.04754	0.16867	0.19456	0.08187	0.19625	0.14707	1	0.21208	0.179
20	0.19875	0.16135	0.1298	0.12368	0.21482	0.26438	0.33577	0.17835	0.32764	0.19459	0.28026	-0.00285	0.20594	0.16388	0.26869	0.14533	0.24007	0.19247	0.21208	1	0.32327
21	0.07942	0.11235	0.24683	0.21361	0.35053	0.21027	0.29158	0.16602	0.45849	0.18088	0.40978	0.12211	0.12256	0.27251	0.37684	0.31777	0.29649	0.1149	0.179	0.32327	1
22	0.13969	0.15174	0.16526	0.26277	0.49662	0.12689	0.17028	0.41	0.62187	0.17411	0.48353	0.1229	0.09076	0.33146	0.60562	0.19648	0.48325	0.28623	0.17127	0.23771	0.37587
23	0.18189	0.07386	0.22522	0.20008	0.27362	0.2045	0.23065	0.1004	0.31134	0.13698	0.36192	0.09778	0.1613	0.15307	0.28191	0.19761	0.2283	0.11543	0.16962	0.29576	0.29025
24	0.17159	0.13126	0.21908	0.30675	0.53046	0.17981	0.25203	0.40539	0.64516	0.19978	0.64693	0.15985	0.20081	0.26618	0.59954	0.26067	0.38109	0.31605	0.31921	0.30142	0.43225
25	0.06316	0.08197	0.15865	0.21575	0.2738	0.15786	0.18932	0.1137	0.33257	0.11512	0.4081	0.13135	0.15255	0.20541	0.33124	0.28602	0.17931	0.1042	0.12219	0.2058	0.40326
26	0.11762	0.11212	0.05808	0.20331	0.21909	0.24615	0.235	0.15843	0.31696	0.19023	0.2472	0.12299	0.14043	0.16776	0.33175	0.21978	0.16521	0.13413	0.15892	0.22407	0.25997
27	0.21673	0.1153	0.1854	0.34734	0.29816	0.20473	0.24931	0.41863	0.43964	0.23504	0.28396	0.03337	0.00055	0.2977	0.4106	0.11419	0.38165	0.31451	0.09842	0.19427	0.21973
28	0.10167	0.07489	0.21385	0.23443	0.59934	0.17554	0.22449	0.19052	0.44695	0.23158	0.52503	-0.0235	0.2182	0.21454	0.4857	0.36634	0.29619	0.16191	0.20112	0.28735	0.40206
29	0.00064	0.0158	0.10099	0.097	0.15656	0.03373	0.08773	0.07948	0.16328	0.1074	0.15336	0.22557	0.08515	0.17095	0.13645	0.16924	0.13146	0.13408	0.08294	0.16862	0.13225
30	0.16691	0.1385	0.17058	0.21278	0.50156	0.22236	0.25301	0.36743	0.554	0.19716	0.47527	0.13649	0.07131	0.28808	0.53397	0.23913	0.38721	0.23892	0.17165	0.27377	0.38842

Figure 6: DAX100 Database

5-3 Graphical User Interface

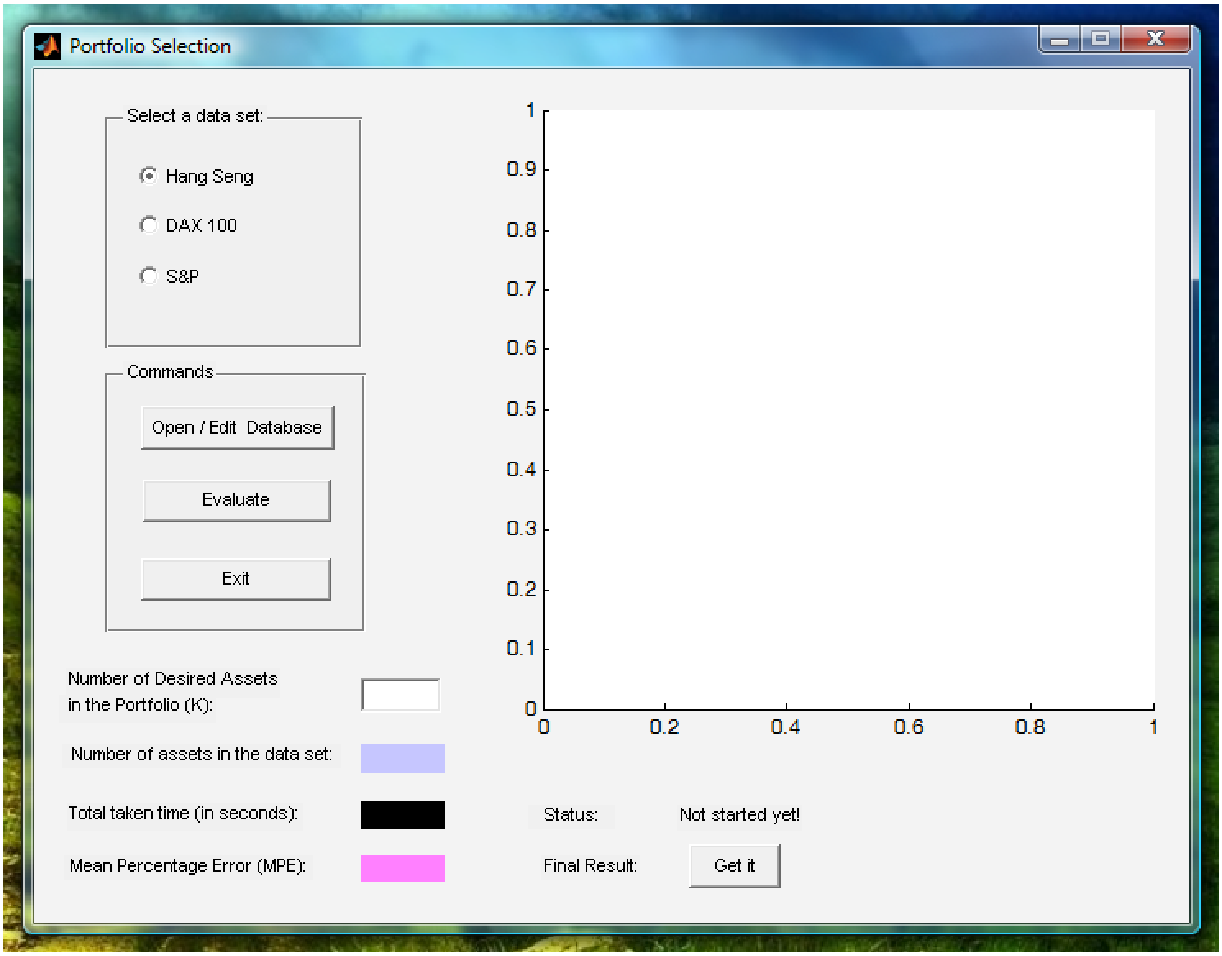


Figure 7: Blank GUI

Here, the procedure of using the program is explained. First, the user chooses the data set he/she would like to select the favorable portfolio from. There are 3 possible choices: Hang Seng, DAX 100 and S&P. Then, the user can try one of the following commands from the 'Commands' panel:

- Open/Edit Database: by pressing this button, the selected data set will open. Then, the user can see, verify or if necessary, edit the data which is available in four excel sheets. (Figure 7)
- Evaluate: by pressing this button, the evaluation will be performed.
- Exit: Pressing this button will close the GUI.

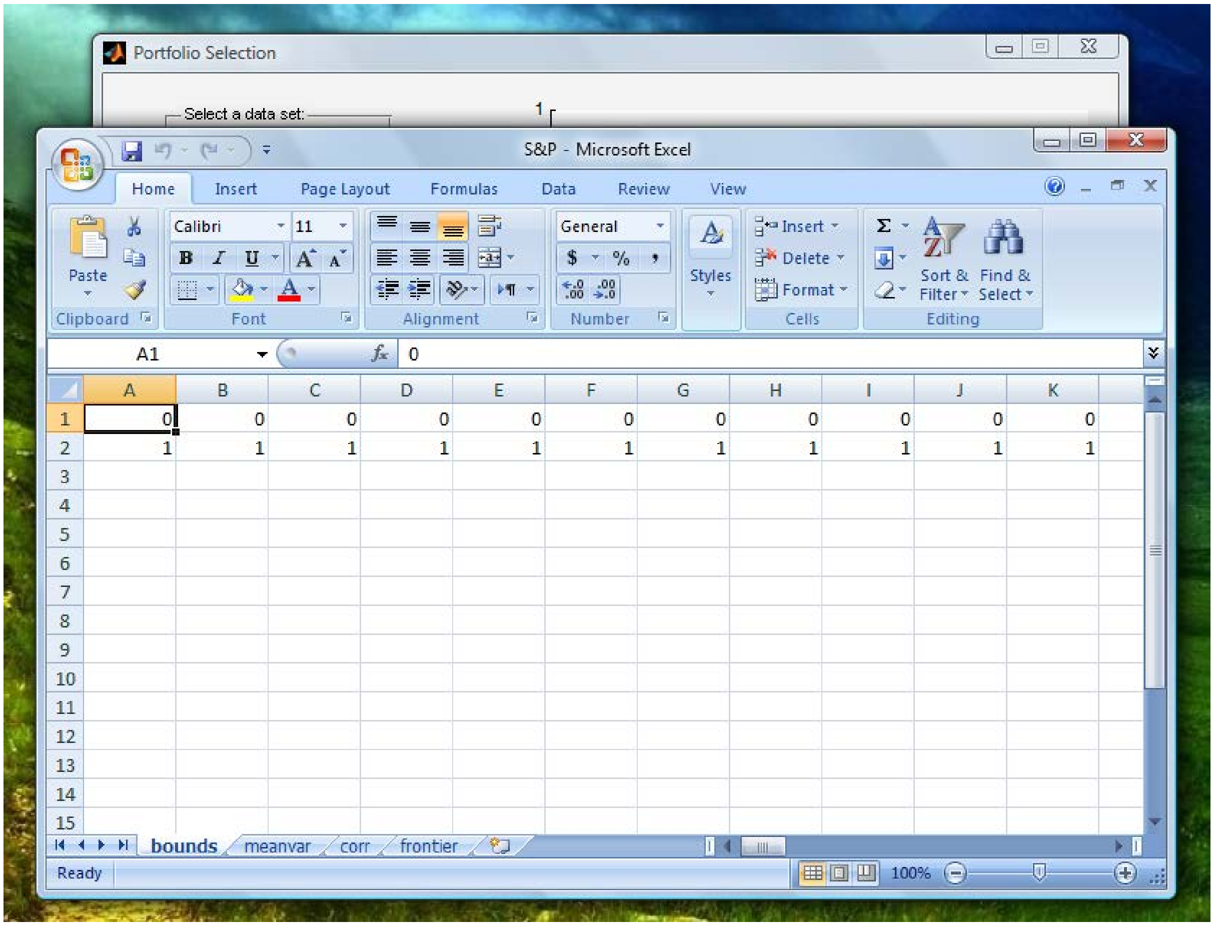


Figure 7: Database in MS Excel format opened by the GUI

To choose a suitable portfolio with predetermined constraints (K and bounds), first, the user should identify the constraints. K is identified in the available text box on the GUI and the floor and ceiling ratios is available in the database.

If the constraints are valid, the evaluation will be started and the status bar will be changed to the 'Running…' status. During the evaluation, the 'Axes' tool shows the progress. After the evaluation is completed, the total time and MPE of the evaluation process will be demonstrated in the relevant text boxes and the status box will be changed to 'Completed!'. By pressing the 'Get it' button, the user can get the final results of the evaluation which is an excel workbook named 'Result.xlsx'. This file contains the lambdas in each row and their relevant evaluation, risk and return. Figure 8 demonstrates a running instance of the portfolio selection problem. In this instance, the data set is DAX100, K=10 and the bounds for all assets is set to 0.01 and 1. Figure 9 shows the final result of that instance. Table 5 also shows the results obtained from excel by clicking the 'Get it' button.

Note here that all the results obtained from this system have one parameter setting which is considered approximately suitable for all cases, unconstrained or constrained. As a result, not all the cases have good results, but in many cases, we could get reasonable results.

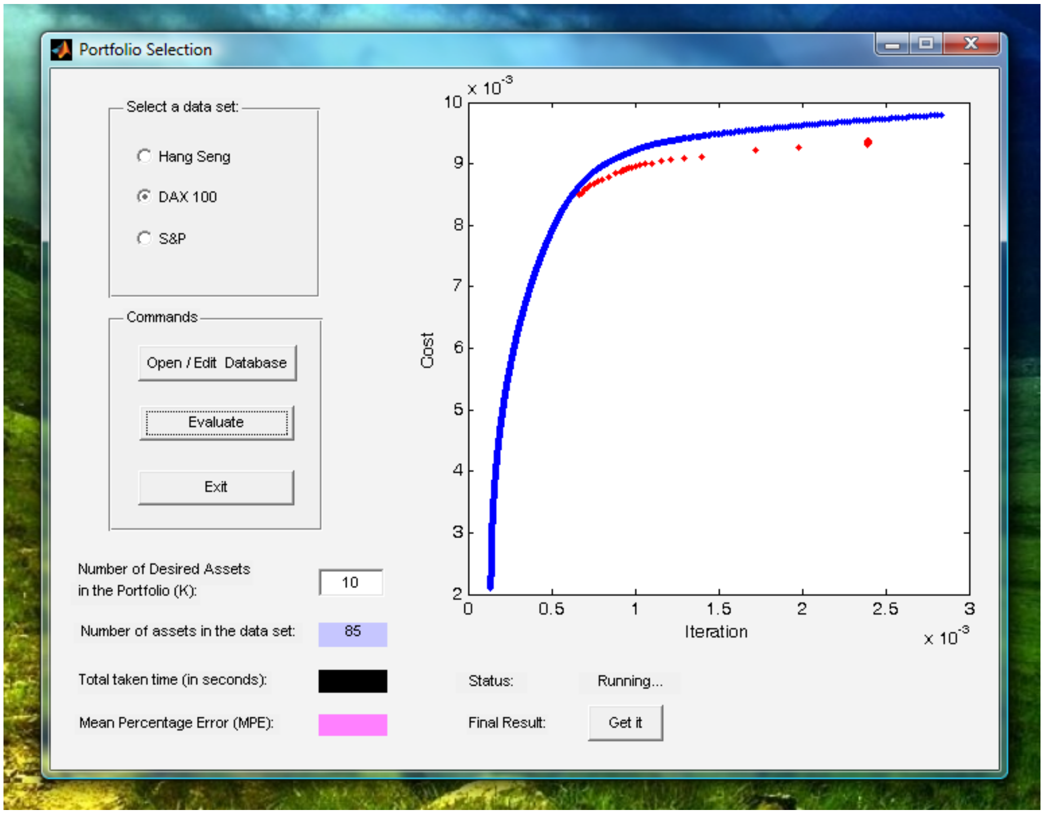


Figure 8: running instance of the problem

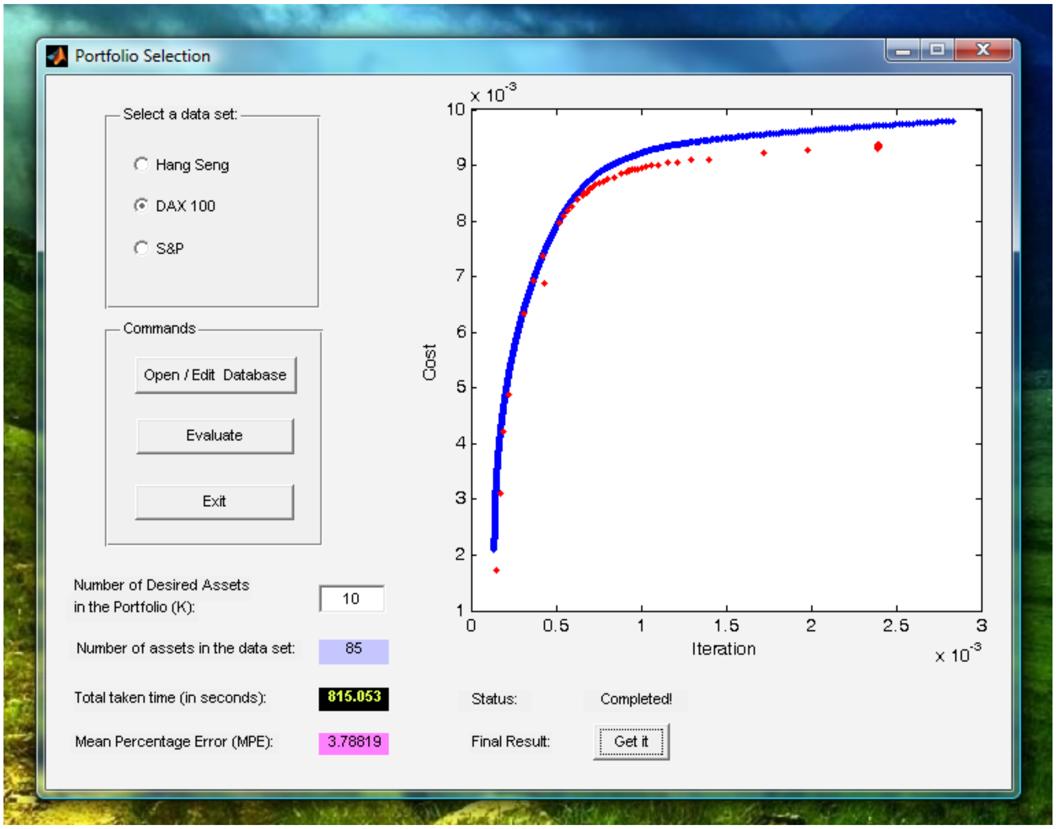


Figure 9: completed evaluation of the instance

Lambda	Evaluatior	Risk	Return
0	0.00937	0.0024	0.00937
0.020408	0.009093	0.002394	0.009332
0.040816	0.008892	0.002392	0.009372
0.061224	0.008649	0.00239	0.009369
0.081633	0.008398	0.002394	0.009357
0.102041	0.008168	0.002394	0.009368
0.122449	0.007916	0.002398	0.009355
0.142857	0.007683	0.002396	0.009363
0.163265	0.007408	0.002396	0.009321
0.183673	0.007198	0.001976	0.009262
0.204082	0.00698	0.001723	0.009211
0.22449	0.006677	0.002393	0.009303
0.244898	0.006536	0.001396	0.009108
0.265306	0.006338	0.001296	0.009094
0.285714	0.006127	0.001214	0.009063
0.306122	0.005921	0.001155	0.009042
0.326531	0.005709	0.001102	0.009011
0.346939	0.005506	0.001063	0.008995
0.367347	0.005298	0.001031	0.008973
0.387755	0.005099	0.001003	0.008963
0.408163	0.004889	0.000981	0.008937
0.428571	0.00469	0.000965	0.008931
0.44898	0.004489	0.000944	0.008917
0.469388	0.004288	0.000929	0.008904
0.489796	0.004091	0.000922	0.008903
0.510204	0.003884	0.000912	0.008881
0.530612	0.003688	0.000886	0.008858
0.55102	0.00348	0.000842	0.008785
0.571429	0.003291	0.000802	0.008748
0.591837	0.003099	0.000774	0.008717
0.612245	0.002905	0.000753	0.00868
0.632653	0.002714	0.000729	0.008645
0.653061	0.002522	0.000712	0.008611
0.673469	0.002335	0.000694	0.008582
0.693878	0.002134	0.00068	0.008511
0.714286	0.00195	0.000668	0.008496
0.734694	0.001765	0.000654	0.008464
0.755102	0.001584	0.000624	0.008391
0.77551	0.001397	0.000592	0.008271
0.795918	0.00122	0.000565	0.00818
0.816327	0.001043	0.000542	0.008087
0.836735	0.000864	0.000524	0.007976
0.857143	0.000611	0.000433	0.006875
0.877551	0.000529	0.000427	0.00738
0.897959	0.000378	0.000366	0.006925
0.918367	0.000235	0.000308	0.006341
0.938776	9.13E-05	0.000222	0.00489
0.959184	-6.2E-06	0.000186	0.004221
0.979592	-0.0001	0.000171	0.003119
1	-0.00015	0.00015	0.00172

Table 5: Final Results of the instance

In such case, the investor can easily do tradeoffs and decide on a suitable portfolio among the assets.

6. Conclusions and future research

In this paper, we presented a decision support system for a constrained portfolio selection problem. The main contribution was in the model base of the system which was a hybrid GA and SA algorithm. Then, we explained how the system including database and GUI was developed.

Future investigations include the consideration of more constraints and the modification of the algorithm to make it more powerful in getting better results, especially in the constrained cases. Due to the practical nature of the portfolio optimization in this developing financial market, it would be very intriguing to incorporate complex decision support systems in portfolio selection problems, taking more realistic constraints into consideration.

References

[1] H. Markowitz (1952). Portfolio selection. Journal of Finance, 7(1):77-91.

[2] T.J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha (2000). Heuristics for cardinality constrained portfolio optimisation. Computers & Operations Research, 27(13):1271-1302.

[3] Y. Crama and M. Schyns (2003). Simulated annealing for complex portfolio selection problems. European Journal of Operational Research, 150:546-571.

[4] S. Wang, D. Lin, and H. Yan (2001). A multi objective genetic algorithm for portfolio selection. Technical report, Institute of Systems Science, Academy of Mathematics and Systems Science Chinese Academy of Sciences, Beijing, China.

[5] U. Derigs, and N.H. Nickel (2003). Meta-heuristic based decision support for portfolio optimization with a case study on tracking error minimization in passive portfolio management. OR Spectrum, 25(3):345-378.

[6] L. Diosan (2005). A multi-objective evolutionary approach to the portfolio optimization problem. In International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. IEEE Conference Proceedings.

[7] K. Doerner, W. Gutjahr, R. Hartl, C. Strauss, and C. Stummer (2001). Ant colony optimization in multiobjective portfolio selection. In Proceedings of the 4th Metaheuristics International Conference, MIC'2001, Porto, Portugal, 243-248.

[8] K. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer (2004). Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Annals of Operations Research, 1:79-99.

[9] A. Fernandez, and S. Gomez (2007). Portfolio selection using neural networks. Computers and Operations Research, 34: 1177-1191.

[10] M.A. Gomez, C.X. Flores, and M.A. Osorio (2006). Hybrid search for cardinality constrained portfolio optimization. In Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, USA, 1865-1866.

[11] J. Matatko, J.E. Fieldsend, and M. Peng (2004). Cardinality constrained portfolio optimisation. In Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'04), Lecture Notes in Computer Science (LNCS 3177), 788-793.

[12] H. Kellerer, and D. Maringer (2003). Optimization of cardinality constrained portfolios with a hybrid local search algorithm. OR Spectrum, 25(4):481-495.

[13] G. Kendall, and Y. Su (2005). A particle swarm optimisation approach in the construction of optimal risky portfolios. In Proceedings of the 23rd IASTED International Multi-Conference on artificial intelligence and applications, 140-145.

[14] A. Loraschi, and A. Tettamanzi (1996). An evolutionary algorithm for portfolio selection within a downside risk framework. In Forecasting Financial Markets, Series in Financial Economics and Quantitative Analysis, 275-285.

[15] A. Loraschi, A. Tettamanzi, M. Tomassini, and P. Verda (1995). Distributed genetic algorithms with an application to portfolio selection problems. Artificial Neural Networks and Genetic Algorithms, 384-387.

[16] R. Mansini, and M.G. Speranza (1999). Heuristic algorithms for the portfolio selection problem with minimum transaction lots. European Journal of Operational Research,114(2):219-233.

[17] D.G. Maringer (2001). Optimizing portfolios with ant systems. In International ICSC congress on computational intelligence: methods and applications, 288-294.

[18] D.G. Maringer, and P. Winker (2003). Portfolio optimization under different risk constraints with modified memetic algorithms. Technical Report 2003-005E, University of Erfurt, Faculty of Economics, Law and Social Sciences.

[19] L. Mous, V.A.F. Dallagnol, W. Cheung, and J. van den Berg (2006). A comparison of particle swarm optimization and genetic algorithms applied to portfolio selection. In Proceedings of Workshop on Nature Inspired Cooperative Strategies for Optimization NICSO, 109-121.

[20] E. Rolland (1997). A tabu search method for constrained real number search: applications to portfolio selection. Technical report, Department of accounting and management information systems, Ohio State University, Columbus, USA.

[21] J. Shapcott (1992). Index tracking: Genetic algorithms for investment portfolio selection. Technical report, Edinburgh Parallel Computing Centre, The University of Edinburgh, JCMB, The King's Buildings, Mayfield Road, Edinburgh.

[22] M.G. Speranza (1996). A heuristic algorithm for a portfolio optimization model applied to the Milan stock market. Computers and Operations Research, 23(5):433-441.

[23] F. Streichert, H. Ulmer, and A. Zell (2003). Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In Selected Papers of the International Conference on Operations Research, Heidelberg, Germany, Springer Verlag, 253-260.

[24] G. Vedarajan, L.C. Chan, and D.E. Goldberg (1997). Investment portfolio optimization using genetic algorithms. In Late Breaking Papers at the Genetic Programming 1997 Conference. Stanford Bookstore, Stanford University, California, 255-263.

[25] S. Arnone, A. Loraschi, and A. Tettamanzi (1993). A genetic approach to portfolio selection. Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems, 3(6):597-604.

[26] W. Yan, R. Miao, and S. Li (2007). Multi-period semi-variance portfolio selection: Model and numerical solution, Applied Mathematics and Computation, 194: 128–134.

[27] C.C. Lin, and Y.T. Liu (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. European Journal of Operational Research, 185: 393–404.

[28] L. Yu, S. Wang, and K.K. Lai (2008). Neural network-based mean–variance–skewness model for portfolio selection. Computers and Operations Research, 35: 34 – 46.

[29] X.T. Deng, S.Y. Wang, and Y.S. Xia (2000). Criteria, models and strategies for portfolio selection. Advanced Modeling and Optimization, 2(2): 79-104.

[30] K. Suksonghong, K. Boonlong, and K.L. Goh (2014). Multi-objective genetic algorithms for solving portfolio optimization problems in the electricity market. International Journal of Electrical Power & Energy Systems, 58: 150-159.

[31] R. Ruiz -Torrubianoa, A. Suárez (2015). A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs. Applied Soft Computing, 36: 125-142.

[32] Y.J. Liu, and W.G. Zhang (2015). A multi-period fuzzy portfolio optimization model with minimum transaction lots. European Journal of Operational Research, 242 (3): 933-941.

[33] R. Saborido, A.B. Ruiz, J.D. Bermúdez, E. Vercher, and M. Luque (2016). Evolutionary multi-objective optimization algorithms for fuzzy portfolio selection. Applied Soft Computing, 39: 48-63.

[34] H. Jalota, M. Thakur, and G. Mittal (2017). A credibilistic decision support system for portfolio optimization. Applied Soft Computing, 59: 512-528.

[35] G.F. Deng, W.T. Lin, and C.C. Lo (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. Expert Systems with Applications, 39 (4): 4558-4566.

[36] D. Kumar, and K.K. Mishra (2017). Portfolio optimization using novel co-variance guided Artificial Bee Colony algorithm. Swarm and Evolutionary Computation, 33: 119-130.