Tanzania Journal of Science 47(1): 194-203, 2021
ISSN 0856-1761, e-ISSN 2507-7961

Application of Guided Local Search (GLS) in Portfolio Optimization

Collether John
University of Dar es Salaam, College of Information and Communication Technology, Department of Electronics and Telecommunications, P. O. Box 33335, Dar es Salaam, Tanzania
Email: nyalusicj@yahoo.com; coletha@udsm.ac.tz

Received 11 Oct 2020, Revised 22 Dec 2020, Accepted 28 Dec 2020, Published Feb 2021

Abstract

Portfolio optimization is a major activity in any operating business. Conventional portfolio optimization research makes simplifying assumptions; for example, they assume no constraint in how many assets one holds (cardinality constraint). They also assume no minimum and maximum holding sizes (holding size constraint). Once these assumptions are relaxed, conventional methods become inapplicable, and hence new methods are needed to tackle this challenge. Threshold Accepting is an established algorithm in the extended portfolio optimization problem. In this paper, an algorithm called Guided Local Search (GLS) is applied using an accurate and efficient designed hill climbing algorithm, named HC-C-R. GLS sitting on HC-C-R is for the purpose of solving the extended portfolio optimization problem. The improved hill climbing algorithm is tested on standard portfolio optimization problem. Results are compared (benchmarked) with the Threshold Accepting (TA) algorithm, a well-known algorithm for portfolio optimization and are also compared with its original algorithm HC-C-R. Results show that GLS sitting on HC-C-R is more effective than HC-C-R and the algorithms are more effective than TA.

Keywords: Portfolio Optimization, Algorithm, Guided Local Search, GLS, Threshold Acceptance.

Introduction

The portfolio optimization problem is a problem concerning asset allocation and diversification for maximum return with minimum risk. The problem is to find the portfolio weights, i.e. how to most appropriately distribute the initial wealth across the available assets, in order to meet the investor's investment objectives and constraints (Markowitz 1952, Markowitz 1959, Meucci 2005, Maringer 2008).

Markowitz (1952) came up with a parametric optimization model for the problem of asset allocation and diversification for maximum return with minimum risk, which has become the foundation for Modern Portfolio Theory (MPT) or Markowitz theory or Mean-Variance model. To apply the Markowitz model to practical problems using the standard/traditional methods like quadratic programming, strong assumptions and simplifications of the real market situations have to be made.

Markowitz model considers what is termed as standard portfolio optimization. In the standard portfolio optimization problem, the constraints taken into account are budget and no-short selling. In reality however, portfolio optimization has realistic constraints to be incorporated, such as holding sizes, cardinality, transaction costs, portfolio size or additional requirements from investors and authorities. When these realistic constraints are added to portfolio optimization, the problem quickly becomes too complex to be solvable by standard optimization methods. When the assumptions and simplifications of the real market situations are relaxed and realistic constraints added, now we have an extended portfolio optimization problem. Here the Markowitz solution and the conventional methods like quadratic programming become inapplicable. Heuristic methods are usually used to deal with this extended portfolio optimization problems (Dueck and Winker 1992, Streichert and Tamaka-Tamawaki 2006, Maringer 2008, Gilli and Kellezi 2000, Crama and Schyns 2003, Muralikrishna 2008). The most established heuristic algorithm used in extended portfolio optimization problems being Threshold Accepting (Maringer 2005, Winker and Maringer 2007, Gilli and Kellezi 2000, Winker 2001, Gilli and Schumann 2010, Gilli and Schumann 2012). The new algorithm proposed below is benchmarked with Threshold Accepting algorithm under standard portfolio optimization problem.

The objective

The objective of the research was to produce more effective and more efficient heuristic algorithm for the extended portfolio optimization problem. In this research, a heuristic algorithm is designed, investigated and then applied to portfolio optimization problem under some constraints of the market. The produced algorithm is implemented in solving the standard portfolio optimization problem. The problem is to find the portfolio weights, i.e. how to distribute the initial wealth across the available assets, in order to meet the investor's objectives and constraints. The significance of the research lies in efficient portfolio selection/optimization and also in efficient investment management (Markowitz 1959).

Modern Portfolio Theory (MPT) or Markowitz Theory or Markowitz Model

Markowitz's standard portfolio optimization model (Markowitz 1952, Markowitz 1959) is a mathematical framework for describing and assessing return and risk of a portfolio of assets, using returns, volatilities and correlations. Markowitz (1952) introduced what is known as the mean-variance principle, where future returns are regarded as random numbers and expected value (mean) of the returns E(r) and their variance (whose square root is called standard deviation/ risk) capture all the information about the expected outcome and the likelihood and range of deviations from it (Markowitz 1952, Markowitz 1959).

Objective function

In the standard Markowitz model, Equation (1), (2) and (3) under basic constraints (4) and (5), the goal is to maximize the expected return, R, while diminishing incurred risk, σ (measured as standard deviation/variance) (Markowitz (1952).

Given return ($R_{\mathbf{p}}$) of a portfolio and variance ($\sigma^2_{\mathbf{p}}$) of portfolio, the equation to maximize is

$$\text{Max } (\lambda.E(R_{\mathbf{p}}) - (1-\lambda).\sigma^2_{\mathbf{p}}) \quad (1)$$

Subject to

- Expected return:
  $$E(R_{\mathbf{p}}) = \sum_i w_i \, E(R_i) \quad (2)$$
- Portfolio return variance:
  $$\sigma^2_{\mathbf{p}} = \sum_i \sum_j w_i w_j \sigma_i \sigma_j \, \rho_{ij} \quad (3)$$
  $$\rho_{ij} = 1 \qquad \text{for } i = j$$
- Constraints:
  $$\sum_i w_i = 1 \quad (4)$$
  $$0 \le w_i \le 1 \quad (5)$$

Where the expected return of each asset is $E(R_i)$, each asset variance is σ_i, and each asset weight is w_i.

From the Equation (1), the trade-off between return ($R_{\mathbf{p}}$) and risk ($\sigma_{\mathbf{p}}$) of portfolio is reflected. The efficient line/frontier is then identified by solving the above problem for different values of $\lambda \in (0, 1)$: If $\lambda = 1$ the model will search for the portfolio with highest possible return regardless of the variance. With $\lambda = 0$, the minimum variance portfolio (MVP) will be identified. Higher values of λ put more emphasis on portfolio's expected return and less on its risk. (Markowitz 1952). Equations (4) and (5) are the constraints on the weights that they must not exceed certain bounds. The most important constraints are budget and return constraints since they characterize the main part of the portfolio problem (Di Tollo and Roli 2008). The return constraint is when the investor requires a certain level of profit from his investment with minimum risk. The budget constraint is when the investor has to invest all the money/capital in the portfolio. However, return constraints can only be satisfied for a historical portfolio (Sharpe 2000, Korn 1997, Prigent 2007, Markowitz 1952 and Markowitz 1959).

Characteristics of heuristic optimization techniques

The core of heuristic methods is an iterative principle that includes stochastic elements in generating new candidate solutions and/or in deciding whether these replace their predecessors while still incorporating some mechanisms that prefer and encourage improvements (Maringer 2008, Winker and Maringer 2007, Glover and Kochenberger 2006, Voudouris et al. 2010). They seek to converge to the optimum in the course of the iterated search. They are flexible and not so restricted to certain forms of constraints (Gilli and Kellezi 2000, Winker 2001, Gilli and Winker 2008, Gill et al. 2011). Heuristic techniques solve optimization problems by repeatedly generating new solutions and testing them. The stopping criterion of the heuristic algorithms is usually a fixed number of steps or if the quality of the solution does not improve after a given or specified number of iterations or both (Winker and Maringer 2007). Therefore, heuristic techniques address problems with a well-defined objective function and model (Maringer 2008).

Summary on some of heuristic portfolio optimization techniques

Simulated annealing (Kirkpatrick et al. 1983) is a type of local search algorithm that accepts all new points that are superior to the current solution according to the objective function, but also, with a certain probability, and accept inferior points. By accepting inferior points, the algorithm avoids being trapped in local minima, and is able to explore more widely for better solutions. The probability of accepting an inferior point decreases over time, following a cooling schedule on the "temperature". When the temperature falls to 0, Simulating Annealing behaves exactly like hill climbing. It has been applied for portfolio selection (Muralikrishna 2008), and with constraints and trading restrictions according to Crama and Schyns (2003).

Threshold Accepting (TA) (Dueck and Scheuer 1990 and Winker and Maringer 2007) can be seen as a variation of simulated annealing, except that there is no introduction of temperature. Instead of accepting inferior new points with a certain probability, it accepts only the points that fall below a fixed threshold. TA was originally proposed by Dueck and Scheuer (1990) as a deterministic and faster variant of the original Simulated Annealing algorithm. As Threshold Accepting avoids the probabilistic acceptance calculations of simulated annealing, it may locate an optimal value faster than the actual simulated annealing technique. In Threshold Accepting algorithm, the best solution obtained depends on some parameters such as the initial threshold value, the threshold decreasing rate and the number of permutations. The initial threshold and threshold decreasing rate are fixed such that a number of iterations can be carried before the algorithm stops (Winker and Maringer 2007).

In this paper, Threshold Accepting is used as a benchmark algorithm to the proposed hill climbing algorithms in solving the standard Markowitz model.

Materials and Methods

The objective of the paper is to produce more effective and more efficient heuristic algorithm for the extended portfolio optimization problem.

Design of HC-C-R

In the design of the method/algorithm HC-C-R, the following is a representation of the solution. As an approach, a solution is represented by a vector of numbers (y_i, …,y_n). The element in position *i* represents the relative weight of the capital invested in stock *i*. The vector of numbers (y_i, …,y_n) are normalized to make sure that the weights in all the assets add up to 1.

The percentage/weight to be invested in stock *i* is x_i, where: $x_i = y_i / \sum_{i=1}^{n} y_i$
One advantage of using this representation is that the vector, y, may take any number without violation of budget constraint that the weights add up to 100%.

Neighbourhood function for the hill climbing algorithm (HC-C-R)

The neighbourhood function of HC-C-R involves Thresh hold Percentage (ThP) to be reduced over time. ThP is a small percentage by which elements of y will be varied to get the next neighbour. In other words, it searches the neighbourhood with finer and finer steps. The following, HC-C-R algorithm is proposed for portfolio optimization.

Elements of vector y are randomly generated. The number of elements of y is equal to number of asset/stocks. The randomly picked position in y is denoted as pos. ThP is a small percentage, which we refer to as threshold percentage, by which elements of y will be varied to get the next neighbour.

The sequence of all the positions of the elements of initial random solution y is randomized (so that the elements are not sequentially picked). If first position in the random sequence gives no better solution, next position is picked and so on. Thus, HC-C-R searches a larger space. This will potentially help it to find better solutions. The cost of doing so is increased time.

The randomly picked position in y is denoted as pos. The neighbourhood definition is to pick one position (pos) in the current solution. After picking the random position in the current solution, one neighbour is obtained by adding ThP to that position and another is obtained by subtracting ThP to the same position. This gives two neighbours (two possible candidate solutions) at a time to be compared with the current solution, at random order. The first better candidate solution (neighbour) to be picked becomes the current solution out of the possible candidate solutions. On getting a better solution, the sequence of the positions of the elements of y is again randomized. The overall procedure is repeated for a number of iterations, or until local maximum. In HC-C-R, ThP is reduced over time. That is after a pre-set number of iterations or on reaching local maximum, ThP is repeatedly reduced to be half the previous value until it reaches the pre-set minimum ThP value, denoted as minThP.

Given mean returns of all stocks in column vector denoted as *retasset*, given assets' co-variances/deviations matrix, denoted as *dev*, and given λ as the level of risk aversion; Figure 1 is the procedure for HC- C-R. Function in Figure 2 is to search for better neighbouring solution.

Pseudo code for HC-C-R

Procedure HC-C-R (ThP, minThP, **λ, retasset, dev)**	
Randomly generate initial current solution y	
Begin	
Do While ThP>minThP	
Repeat	
pick random position, pos, in current solution y	
yplus = y	%Generate yplus from current solution%
yminus = y	%Generate yminus from current solution%
yplus(pos) = yplus(pos)*(1 + ThP)	% Get neighbour of current solution. %
yminus(pos) = yminus(pos)*(1 - ThP)	%Get second neighbour of current soln%
yb4=y	% keep record of current solution y %
y **= move_to_neighbour** (y, yplus, yminus, λ, retasset, dev)	% Pick a better neighbour solution. %
Randomly change the sequence of the positions	% Provides randomness. %
while y=yb4 **do**	% Looks for better solution in the random sequence. (pos) is any position. %
yplus = y	% Generate yplus from current solution %
yminus = y	%Generate yminus from current solution%
yplus(pos) = yplus(pos)*(1 + ThP)	% Get neighbour of current solution. %
yminus(pos) = yminus(pos)*(1 - ThP)	%Get second neighbour of current soln%
if (all positions in the sequence have been checked for better solution) **then** break while loop	
end if	
end while	
Until halting criterion is met	
ThP=ThP/2	
End While	
End	% Halting criterion was no neighbour is better than current solution or maximum number of iteration is reached.

Figure 1: Hill climbing procedure of HC- C-R.

Pseudo code for a **function for searching for better neighbouring solution**

Function move_to_neighbour (y, yplus, yminus, λ, retasset, dev) **Begin**	
$x_i = y_i / \sum_{i=1}^{n} y_i$	% Find weights, x, of all the assets n in portfolio%
$xplus_i = yplus_i / \sum_{i=1}^{n} yplus_i$	% Find weights, xplus, of all assets n %
$xminus_i = yminus_i / \sum_{i=1}^{n} yminus_i$	% Find weights, xminus, of all assets n%
xvalue = **objectvalue** (x, λ, retasset, dev)	% Calculate objective value of portfolio x and denote as xvalue. %
xplusvalue = **objectvalue** (xplus, λ, retasset, dev)	% Calculate objective value of portfolio xplus and denote as xplusvalue%
xminusvalue = **objectvalue** (xminus, λ, retasset, dev)	%Calculate objective value of portfolio xminus and denote as xminusvalue. %
if xplusvalue>xvalue **then** y=yplus **end if**	% Return yplus if it is better than y. %
if xminusvalue>xvalue **then** y=yminus **end if**	% Return yminus if it is better than y. %
return y **End**	

Figure 2: Function to search for better neighbouring solution.

The function in Figure 3 measures the quality of a portfolio. The function calculates the objective/objective value from Equation (1). The mean returns and co-variances of all assets/stocks are initially calculated from the daily prices in the main program. They are used to calculate the expected return and risk of a portfolio. The return and risk of a portfolio calculated are used to measure the quality of a portfolio.

Pseudo code for calculating objective function value

Function object value (x, λ, retasset, dev) **Begin**	
retpor t = scalar/dot product(retasset, x)	%Calculate effective expected return %
risk = x*dev*x’	% Calculate effective risk/variance %
fitvalue = λ*retport – (1 - λ)*risk **return** fitvalue	%Calculate objective/objective value according to equation (1) above. %
End	

Figure 3: Function to calculate objective/fitness value.

Application of GLS

Below is the pseudo code of Guided Local Search, GLS (Voudouris et al. 2010) method applied in finding the optimum portfolio of n assets. Figure 4 shows GLS application using HC-C-R.

Pseudo code of GLS

<table>
<tr><td>
Procedure Guided Local Seach (p, g, λ , [I1, . . . , IM], [c1, . . . ,cM], M)
Begin
k=0;
s0 is randomly generate initial solution (p);
    for i=1: M do
        pi =0;
        h=g+ λ*∑ pi *Ii;
    while Stopping criterion do
    begin
sk+1 =Hill-climbing method HC-C-R (sk,h);
    for i=1: M do
        utili =Ii(sk+1) ∗ci/(1+ pi);
    for each i such that utili is maximum do
        pi = pi+1;
        k=k+1;
    end
s∗ is best solution found with respect to objective function g;
    return s∗;
End
</td><td>
% set all penalties to 0 %
% define the augmented objective function %

%the method HC-C-R is described figure 1 above%
% compute the utility of features %
/% penalize features with maximum utility %
</td></tr>
</table>

Figure 4: Procedure of GLS.

Where *p* is the problem, *g* is the objective function, *h* is the augmented objective function, λ is a parameter, *Ii* is the indicator function of feature *i*, *ci* is the cost of feature *i*, *M* is the number of features, *pi* is the penalty of feature *i*,.

Results

Benchmarking the Algorithms using Threshold Accepting

HC-C-R and GLS are benchmarked on the Markowitz model, Equations (1), (2), and (3) under basic constraints (4) and (5). They are tested on 100 assets portfolio. The results are compared with Threshold Accepting, which is a well-established Hill Climbing algorithm in portfolio selection and optimization.

The assets and their return data used for applications in the algorithms are from DAX stock exchange. The data used were daily returns over 1001 days.

The following are the algorithms that are evaluated:

HC-C-R: HC-C with reducing ThP

GLS: Guided Local search

The following are the explanations of notations of the algorithms used in presenting results:

HC-C-R (0.1, 0.01, 9e+5) is HC-C-R with starting ThP = 0.1, half ThP every 9e+5 iterations, until ThP is below 0.01. The above number of iterations was given on every ThP but the program was to stop on reaching a local optimum.

GLS (700): Guided Local Search with 700 iterations sitting on HC-C-R (0.1, 0.01, 500).

Table 1 shows experimental results on the portfolio optimization on 100 stocks from DAX stock exchange; taken after 100 runs. The results show the values of objective function, number of functional evaluations required to reach final objective value, and average time in seconds for one run to converge to local maximum (final solution). The Best Final Objective value is the highest objective function value obtained in all 100 runs. Final objective values obtained in each run were recorded and so below are the mean, standard deviations (STD) and Worst of Final objective values in all the 100 runs. The mean and STD of number of functional evaluations to reach final objective value, of the 100 runs, are also given.

Table 1: Experimental results on portfolio optimization on 100 stocks, after 100 runs

Algorithm		**GLS** (700) (on **HC-C-R** (0.1, 0.01, 500))	**HC-C-R** (0.1, 0.01, 9e+5)	**Threshold Accepting** (9e+5)
Best final objective value		0.000596	0.000596	0.000588
Final objective value	Mean STD Worst	0.000596 1.4e-10 0.000596	0.000595 3.32e-6 0.000572	0.000563 3.46e-5 7.2563e-5
No. of functional evaluations to final objective value	Mean STD	2.1e+5 2.8e+3	3.2e+4 943	3.0e+5 1770
Average time for 1 run (sec)		43.37	28.05	704.7

STD = Standard Deviation

Discussion

GLS and HC-C-R are better than TA: GLS on HC-C-R and HC-C-R managed to attain higher best final objective value (0.000596) than Threshold Accepting (0.000588). The best final objective values are higher and similar in GLS and HC-C-R, showing that the methods are more robust than Threshold Accepting as they better escape local optima.

To understand the significance of the difference in final objective value we look at the best final objective value of HC-C-R which is 0.000596. This translates to a return of 0.14% and a risk of 1.34% one day after investment, of the 100 stocks considered. The best final objective value of Threshold Accepting, 0.000588, translates to a return of 0.13% and a risk of 1.54% one day after investment. The following days could include compounded interest on the original capital. From the return and risk figures, it is observed that you incur more risk but expect less return when you use the Threshold Accepting rather than HC-C-R to find an optimal portfolio.

The mean of final objective value of HC-C-R is higher (0.000595) than that of Threshold Accepting (0.000563). The worst final objective of HC-C-R is a lot better (0.000572) than that of Threshold Accepting (7.2563e-5). The STD of mean of final objective value of HC-C-R(3.32e-6) is 10 times less that of Threshold Accepting (3.46e-5).

The number of functional evaluations for HC-C-R was 3.2e+4 while that of Threshold Accepting was 3.0e5. HC-C-R was faster as it required less number of functional evaluations. The STD of the number of functional evaluations of HC-C-R (943) is less than that of Threshold Accepting (1770). Considering the time in seconds for one run to converge to best final objective value, Threshold Accepting (704.7), required more time than HC-C-R (28.05). This shows that it is far more expensive (time wise) to compute neighbourhood function of Threshold Accepting than that of HC-C-R.

A t-test was performed on final objective values and on the number of functional evaluations to final objective of the 100 runs. Both outcomes displayed a rejection of the null hypothesis at the 5% (default value) significance level. The t-test was performed using Mat-lab (R2010a).

Furthermore, to use Threshold Accepting, one has to first calculate and sort threshold sequences according to a certain problem. These are the sequences by which poor solutions will be accepted to avoid being trapped in a local optimum. The process makes Threshold Accepting quite cumbersome.

GLS is better than No GLS: By adding penalties every time there was a local optimum, GLS managed to attain the best final objective value in all 100 runs. The mean of the final objective value is the same as the value of best final objective value, that is 0.000596, and the STD of the mean of the final objective value is exceedingly small (1.3997e-10). The worst final objective value was also the same as the best final objective value (0.000596). This demonstrates that GLS on HC-C-R was better in reliability to find best final objective value than HC-C-R as HC-C-R attained lower mean of final objective value (0.000595) and the worst final objective value was a lot lower than that of GLS.

The overall results demonstrate that GLS (sitting on HC-C-R) manages to find better solutions (higher mean of the final objective value) than HC-C-R and TA.

Following the results of the experiments above on benchmarking the algorithms with TA are summarized in a Table 2.

Table 2: Summary on benchmarking the algorithms with TA

Algorithm	**Effectiveness**	**Efficiency**
TA (Dueck and Cheuer 1990)	Well established algorithm in portfolio optimization (Winker 2001, Radziukynienė and Žilinskas 2008, Gilli and Schumann 2010, Hoos and Tsang 2006, and Gilli and Schumann 2012)	Efficient
HC-C-R	More effective in finding better solution than **TA**	More efficient than **TA** and more efficient than **GLS (**Voudouris et al. 2010) (sitting on HC-C-R**)**
GLS (Voudouris et al. 2010) (sitting on HC-C-R**)**	More effective and reliable in finding better solution than **HC-C-R**	More efficient than **TA**

Conclusion

GLS and HC-C-R have been described and implemented in portfolio optimization problem. They were tested on the Markowitz model; in finding weights for 100 stocks in portfolio optimization, where a budget constraint is imposed and no short-selling is permitted. The results demonstrate that GLS sitting on HC-C-R manages to find better solutions than the other algorithms. The small standard deviations observed show that GLS and HC-C-R find solutions more robust than Threshold Accepting with GLS able to find best final objective value in all 100 runs. In future more realistic, non-linear constraints like transaction costs will be incorporated. The hill climbing algorithms are also quite easy to understand and to implement. So these algorithms also have wider application areas other than portfolio optimization, for instance, in different research areas of science. Also the hill climbing algorithms will be combined with evolutionary algorithms like genetic algorithms, to give hybrid algorithms for portfolio optimization and other applications.

Acknowledgment

I thank my employer; University of Dar es Salaam for all support and encouragement.

References

Crama Y and Schyns M 2003 Simulated annealing for complex portfolio selection problems. *Eur. J. Oper. Res.* 150: 546–571.

Di Tollo G and Roli A 2008 Meta-heuristics for the portfolio selection problem. *Int. J. Oper. Res.* 5(1): 13-35.

Dueck G and Scheuer T 1990 Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *J. Comput. Phys.* 90: 161–175.

Dueck G and Winker P 1992 New concepts and algorithms for portfolio choice. *Appl. Stochastic Models Data Anal.* 8: 159-178.

Gilli M and Kellezi E 2000 Heuristic Approaches for portfolio optimization. In proceedings of the *6th International. Conference on Computing and Economics* Barcelona Spain.

Gilli M and Winker P 2008 A review of heuristic optimization methods in econometrics. *Swiss Finance Institute Research* Paper No. 08-12.

Gilli M and Schumann E 2010 Portfolio optimization with threshold accepting: a practical guide In: Satchell SE (Ed) *Optimizing Optimization: The Next Generation of Optimization Applications and Theory*. Chapter 9 Elsevier.

Gilli M, Maringer D and Schumann E 2011 Numerical methods and optimization in finance. Elsevier Inc. chapter 12, 13.

Gilli M and Schumann E 2012 Heuristic optimisation in financial modelling *Ann. Oper. Res.* 193(1): 129-158.

Glover F and Kochenberger GA (Eds) 2006 Handbook of metaheuristics,(Vol. 57). Springer Science & Business Media.

Hoos HH and Tsang EPK 2006 Local search methods in: Rossi F van Beek P and Walsh T (Eds) *Handbook of Constraint Programming* pg. 135 Elsevier.

Kirkpatrick S, Gelatt Jr CD Vecchi P 1983 Optimization by simulated annealing. *Science* 220(4598): 671-680.

Korn R 1997 Optimal portfolios: stochastic models for optimal investment and risk management in continuous time. World Scientific Publishing Co. Pte. Ltd.

Maringer D 2005 Heuristic optimization. Portfolio management with heuristic optimization, pp. 38-76, Dordrecht Springer.

Maringer D 2008 Heuristic optimization for portfolio management *IEEE Comput. Intel. Mag.* 3(4): 31-34.

Markowitz H 1952 Portfolio selection. *J. Finance* 7(1): 77-91.

Markowitz H 1959 Portfolio selection: efficient diversification of investments. John Wiley and Sons New York City.

Meucci A 2005 Risk and asset allocation. Springer–Verlag Berlin.

Muralikrishnan V 2008 Optimization by simulated annealing (portfolio optimization) *Global Association of Risk Professionals* 45-48 Jun/July.

Prigent J 2007 Portfolio optimization and performance analysis. Chapman & Hall/ CRC Press, New York.

Radziukynienė I, Žilinskas A 2008 Evolutionary methods for multi-objective portfolio optimization In: *Proceedings of the World Congress on Engineering* (Vol. 2, pp. 1155-1159), London UK.

Sharpe W F 2000 Portfolio theory and capital markets McGraw-Hill New York.

Streichert F, Tamaka-Tamawaki M 2006 The effect of local search on the constrained portfolio selection problem. *IEEE World Congress on Evolutionary Computation (CEC2006)* 2368-2374.

Voudouris C Tsang EPK and Alsheddy A 2010 Guided local search In: Gendreau M Potvin JY (Eds) *Handbook of Metaheuristics* 3rd ed (pp. 321-361), Springer Boston, MA .

Winker P 2001 Optimization heuristics in econometrics: application of threshold accepting. John Wiley and sons New York.

Winker P and Maringer D 2007 The threshold accepting optimisation algorithm in economics and statistics. In: Kontoghiorges EJ and Gatu C (Eds.) Optimisation, Econometric and Financial Analysis. *Advances in Computational Management Science* Vol 9, Springer, Berlin, Heidelberg.