

*Full Length Research Paper*

## **Complete Neighbourhood Search Heuristic Algorithm for Portfolio Optimization**

**Collether John**

University of Dar es Salaam, Department of Electronics and Telecommunications,  
P. O. Box 33335, Dar es Salaam

Email of corresponding author: nyalusicj@yahoo.com; coletha@udsm.ac.tz

### **ABSTRACT**

*In portfolio optimization, the fundamental goal of an investor is to optimally allocate investments between different assets. Mean-variance optimization methods make unrealistic assumptions to solve the problem of optimal allocation. On the other hand, when realistic constraints like holding size and cardinality are introduced it leads to optimal asset allocation which differ from the mean variance optimization. The resulting optimization problem become quite complex as it exhibits multiple local extrema and discontinuities. Heuristic algorithms work well for the complex problem. Therefore, a heuristic algorithm is developed which is based on hill climbing complete (HC-C). It is utilized to solve the extended portfolio optimization problem. In order to validate its performance, the proposed HC-C is tested with standard portfolio optimization problem. Experimental results are benchmarked with the quadratic programming method and threshold accepting (TA) algorithm.*

*Key words: Optimization, Portfolio, Heuristics, Hill Climbing, Algorithm, Threshold Accepting.*

### **INTRODUCTION**

Optimization techniques are search methods whereby the goal is to find a solution to an optimization problem such that a given quantity is optimized subject to a set of constraints. 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 distribute the initial wealth across the available assets in order to meet the investor's objectives and constraints as much as possible (Maringer, 2008; Meucci, 2005; Markowitz, 1959; Markowitz, 1952). Markowitz (1992) introduced the mean-variance optimization as a quantitative tool to allow this allocation by considering the trade-off between risk and return. The risk

is measured by the variance of the future asset returns. The assumptions of the normality of the returns or of the quadratic investor's preferences allow the simplification of the utility optimization problem in a relatively easy to solve quadratic program. To dismiss the normality hypothesis in order to account for the fat-tailed ness and the asymmetry of the asset returns, heuristic approaches and other alternative approaches attempt to conform the assumptions to reality. The resulting optimization problem become quite complex as it exhibits multiple local extrema and discontinuities.

Heuristic algorithms are usually used to deal with this complex/extended portfolio optimization problem (Maringer, 2008; Muralikrishna, 2008; Streichert and

Tamaka-Tamawaki, 2006; Crama and Schyns, 2003; Gilli and Kellezi, 2000).

There are some heuristic algorithms that have been applied to extended portfolio optimization problem like simulated annealing (Kirkpatrick et al. (1983) in (Muralikrishna, (2008)), and with constraints and trading restrictions in (Crama and Schyns (2003)), and also threshold acceptance (Dueck and Scheuer (1990)), being the most established heuristic algorithm, has been applied in (Gilli and Schumann (2012)), Gilli and Schumann (2010), Winker and Maringer (2007), Maringer (2005), Winker (2001), Gilli and Kellezi (2000)). In John (2021) application of guided local search (GLS) is proposed to solve the problem.

In this paper, a new proposed algorithm based on HC-C is benchmarked with a threshold accepting algorithm under standard portfolio optimization problem. The objective is to come up with a more effective and more efficient hill climbing algorithm to tackle the complex portfolio optimization problem. The proposed algorithm is effective and efficient in tackling complex system than TA.

## METHODS AND MATERIALS

In Markowitz model optimization, the aim is to maximize the expected return,  $R$ , while diminishing incurred risk,  $\sigma$  (measured as standard deviation/variance), (Markowitz, 1959; Markowitz, 1952).

Given return ( $R_p$ ) of a portfolio and variance ( $\sigma_p^2$ ) of portfolio, the equation to maximize is given by:

$$\text{Max}(\lambda E(R_p) - (1 - \lambda)\sigma_p^2) \quad (1)$$

Subject to the expected return:

$$E(R_p) = \sum_i w_i E(R_i) \quad (2)$$

The portfolio return variance:

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_i \sigma_j \rho_{ij} \quad (3)$$

where:  $\rho_{ij} = 1$  for  $i=j$ .

Constraints:

$$\sum_i w_i = 1 \quad \text{for } 0 \leq w_i \leq 1 \quad (4)$$

where the expected return of each asset is  $E(R_i)$ , each asset variance is  $\sigma_i$ , and each asset weight is  $w_i$ .

It is noted that equation (1) reflects the trade-off between return ( $R_p$ ) and risk ( $\sigma_p$ ) of portfolio. By solving the above problem for different values of  $\lambda \in (0, 1)$ : the efficient line/frontier is then identified. If  $\lambda=1$ , the model will search for the portfolio with highest possible return regardless of the variance. If  $\lambda=0$ , the minimum variance portfolio (MVP) will be identified. So higher values of  $\lambda$  put more emphasis on portfolio's expected return and less on its risk, (Markowitz, 1952). Equation (4) represents the constraints on the weights that they must not exceed certain bounds.

The function below measures the quality of a portfolio. The function calculates the 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.

### (a) Design of proposed HC-C algorithm

The hill climbing complete algorithm is denoted as HC-C. HC stands for Hill Climbing, C stands for complete search of neighbourhood according to the neighbourhood functions defined below. HC-C explores the possibility of changing the relative weight of every asset in each step, hence the name "complete" is added.

#### Solution

The solution is represented by a vector of numbers ( $y_1, \dots, y_n$ ). The element in position  $i$  represents the relative weight of the capital invested in stock  $i$ . The vector of numbers ( $y_1, \dots, y_n$ ) 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:



<p style="text-align: center;"> <b>end if</b>  <b>end while</b>                  Until halting criterion is met  <b>End.</b> </p>
---

**Table 2: Pseudo Code of a function for searching for better neighbouring solution**

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

**Table 3: Pseudo Code for calculating objective function value**

<p> <b>Function Objectvalue (x, λ, retasset, dev, fitvalue)</b>  <b>Begin</b>                  retpor t = scalar/dot product (retasset, x)                  risk = x * dev * x'                  fitvalue = λ * retpor - (1 - λ) * risk  <b>return</b> fitvalue  <b>End</b> </p>	<p>                 % Calculate effective expected return of portfolio %                  % Calculate effective risk/variance of portfolio %                  % Calculate objective/objective value according to equation (1) above. %             </p>
--	---

## RESULTS AND DISCUSSION

### Benchmarking the Algorithm

The working of the algorithm HC-C is first illustrated by solving the standard Markowitz model as described in equations (1), (2), (3) under basic constraints (4). The model is also solved using quadratic programming method and then threshold acceptance (TA), which is a well-

established heuristic algorithm in portfolio optimization used as benchmarks.

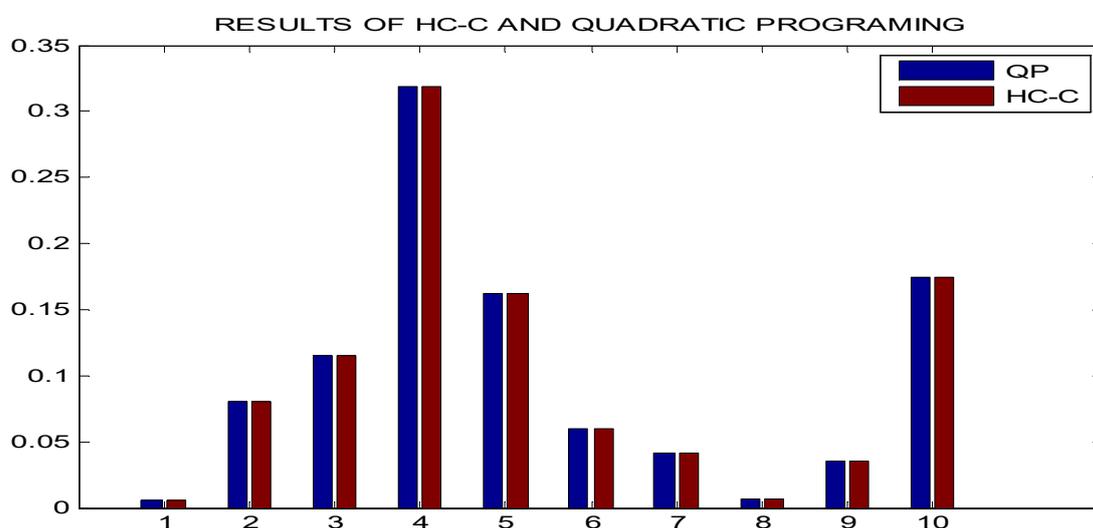
The data used were daily returns over 1001 days DAX stock exchange. The proposed HC-C algorithm is tested on 10 assets example. The results are compared with Quadratic Programming (QP), which is a standard method. Below in Table 4 and corresponding bar charts in Figure 1, are experimental results on benchmarking HC-C with QP. They are the percentage values

in a Table 4 and corresponding bar charts of the weights of 10 assets portfolio. They were obtained by finding minimum variance portfolio (Markowitz model with  $\lambda=0$  in expression (1) above) by quadratic programming method and by the proposed HC-C algorithm. The QP produces exact solution so results by HC-C are compared

with QP results to see how accurate the method is. The values show the relative weights (of total budget) to be invested in each asset. The results (weights) by HC-C algorithm is from the best solution after 100 iterations.

**Table 4: Results on benchmarking HC-C with Quadratic Programming (QP)**

algorithm	Weight in each asset									
QP	0.0053	0.0802	0.1150	0.3191	0.1622	0.0599	0.0419	0.0067	0.0356	0.1741
HC-C	0.0053	0.0802	0.1150	0.3192	0.1622	0.0599	0.0419	0.0067	0.0356	0.1740



**Figure 1: Composition of the optimal portfolio for HC-C (red) and QP (blue)**

It is observed from the results in Table 4 and bar chart form showed by Figure 1 that solutions obtained by HC-C do not differ much from the exact solution by QP.

Moreover, the variance/risk parameter was calculated from the weights obtained by both methods QP and HC-C. Both methods attained the same low portfolio risk of  $6.9751e-005$ . Attaining the same value of risk as QP depicts that the proposed HC-C algorithm attain very accurate solutions. In fact, Figure 1 shows the results of the HC-C algorithms in comparison to QP. In that case, the blue bars are that of QP while the red ones depicts the performance of HC-C respectively. The similar height bars of HC-

C and QP also depict that the HC-C algorithm gives very accurate solutions. To benchmark HC-C with TA, both algorithms are applied on a benchmark problem to solve standard Markowitz model as described in equations (1), (2), (3) under basic constraints (4). They are then tested on 100 assets portfolio. The experimental results of HC-C are compared with those of TA.

The results in Table 4 show the values of objective function, number of functional evaluations required to reach final objective value, and average time in seconds for one iteration to converge to local maximum (final solution). The best final objective

value is the highest objective function value obtained in all 100 runs. The Mean, STD and worst of final objective values in all the 100 runs are given. The Mean and STD of Number of functional evaluations to reach final objective value, of the 100 runs, are also given. The **HC-C (9e+5)** is HC-C with 9e+5 iterations.

**Table 4: Results on Portfolio optimization on 100 stocks (100 runs)**

Algorithm		HC-C (9e+5)	TA (9e+5)
Best final objective value		5.96e-4	5.88e-4
Final objective value	Mean	5.95e-4	5.63e-4
	STD	2.47e-6	3.46e-5
	Worst	5.72e-4	7.2563e-5
No. of Functional evaluations to final objective value	Mean	3.0e+5	3.0e+5
	STD	7400	1770
Average time for 1 run (in sec)		100	704.7

STD =Standard Deviation

**Evaluating the results**

In the numerical experiments presented in Table 5, the best final objective value is higher in HC-C than in TA which means that HC-C is good at escaping local

optimum and locating high quality solutions with a given enough number of iterations (in this case 100 runs). Also, the mean of final objective value of HC-C is higher (0.000595) than that of TA (0.000563). The worst final objective of value of HC-C is a far better (0.000572) than that of TA (0.000559). The STD of mean of final objective value of TA (3.46e-5) is more than ten times that of HC-C (2.47e-6). These results demonstrate that HC-C is better as it is more accurate and reliable than TA. The time, in seconds it takes for one iteration to converge to best final objective value of HC-C is 100 and that of TA is 704.7 implying that HC-C required far less time than TA. Therefore, it is far more expensive (time wise) to compute neighbourhood function using TA than HC-C algorithm.

What makes TA cumbersome is that, 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.

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 MATLAB (R2010a). In Table 5 the effectiveness and efficiency of the algorithms are compared.

**Table 5: Comparison of HC-C with TA**

Algorithm	Effectiveness	Efficiency
TA (Dueck and Scheuer (1990))	Well established algorithm in portfolio optimization (Winker (2001), Hoos and Tsang (2006), Radziukynienė (2008), Gilli and Schumann (2010). and Gilli and Schumann (2012)).	Efficient
HC-C	More effective in finding better solution than TA	more efficient than TA

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.

## CONCLUSION AND RECOMMENDATIONS

In this paper, a hill climbing complete (HC-C) algorithm has been proposed and implemented. The proposed HC-C algorithm is used to tackle the portfolio optimization problem using standard Markowitz model, where a budget constraint is imposed and no short-selling is permitted. The HC-C algorithm performs very well according to the results of benchmarking with QP giving accurate solutions. The results also have shown that HC-C algorithm is more effective and efficient than TA which is a well-known algorithm for portfolio optimization problem. In fact, the HC-C algorithm is quite easy to understand and implement. Therefore, it true to recommend this algorithm be utilised in other areas of application that require effective and efficient algorithm for optimization.

## REFERENCES

- Agnieszka K.A., (2004). Phosphorus Crama, Y., Schyns, M., (2003). Simulated annealing for complex portfolio selection problems, *European Journal of Operational Research*, 150:546–571,
- Di Tollo, G., and Roli, A., (2008). Metaheuristics for the Portfolio Selection Problem, *IJOR Vol. 5, No. 1*, 130-135,
- Dueck, G., Scheuer, T., (1990). Threshold accepting: a general-purpose optimization algorithm, *Journal of Computational Physics*; 90:161–175
- Dueck, G., Winker, P., (1992). New concepts and algorithms for portfolio choice, *Applied stochastic models and data analysis*, Vol 8, 159-178.
- Gilli, M., Kellezi, E., (2000). Heuristic Approaches for Portfolio Optimization, 6<sup>th</sup> intl. Conf. On Computing and Economics, Barcelona, July.
- Gilli, M., Maringer, D., Schumann, E., (2011). Numerical Methods and optimization in Finance, Elsevier Inc. chp 12, 13
- Gilli, M., Schumann, E., (2010). Portfolio Optimization with Threshold Accepting: a Practical Guide, Ch. 9, In Stephen E. Satchell (Ed), *Optimizing Optimization: The Next Generation of Optimization Applications and Theory*. Elsevier,
- Gilli, M., Schumann, E., (2012). Heuristic Optimisation in Financial Modelling, *Annals of Operations Research*, Vol. 193, No. 1, pp. 129-158, ([http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1277114](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1277114))
- Gilli, M., Winker, P., (2008). A Review of heuristic optimization methods in econometrics, Swiss Finance Institute Research Paper No. 08-12. Available at SSRN: <http://ssrn.com/abstract=1140655>, June.
- Glover, F. & Kochenberger, G.A. (ed.), (2006). *Handbook of metaheuristics*, Vol. 57, Springer Science & Business Media.
- Hoos, H.H., Tsang, E. P.K., (2006). Local Search Methods, *Handbook of Constraint Programming*, pp. 245, Ed. F. Rossi, P. van Beek and T. Walsh c, Elsevier.
- John, C., (2021). Application of guided local search (GLS) in portfolio optimization problem, *Tanzania Journal of Science*, Vol 47(1):194-203.
- Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., (1983). Optimization by simulated annealing, *Science* 220 (4598) 671–680, 1983.
- Maringer, D., (2005). *Portfolio management with heuristic optimization*, Dordrecht, Springer.
- Maringer, D., (2008). *Heuristic Optimization for Portfolio*

- management. Nov, IEEE Comp. Intelligence Magazine.
- Markowitz, (1952). Portfolio selection. J. Finnan. pp 77-91.
- Markowitz, (1959). Portfolio selection: Efficient diversification of investments. John Wiley and Sons.
- Meucci, A., (2005). Risk and asset allocation, Springer.
- Muralikrishnan, V., (2008). Optimization by Simulated annealing (portfolio optimization), Global association of risk professionals, 45-48, Jun/jolly.
- Prigent, J., (2007). Portfolio optimization and performance analysis, Chapman & Hall/CRC.
- Sharpe, W. F., (2000). Portfolio theory and capital markets. McGraw-Hill.
- Streichert, F., Tamaka-Tamawaki, M., (2006). The effect of local search on the constrained portfolio selection problem. In Proceedings of the IEEE World Congress on Evolutionary Computation (CEC2006), pages 2368– 2374.
- Tsang, Edward. (1993). Foundations of Constraint Satisfaction, Academic Press, chapter 8.
- Voudouris, C., Tsang, E.P.K. & Alsheddy, A., (2010). Guided local search, Chapter 11, in M. Gendreau & J-Y Potvin (ed.), Handbook of Metaheuristics, Springer, 321-361.
- Winker, P., (2001). Optimization Heuristics in Econometrics- Application of Threshold accepting, John Wiley and sons.
- Winker, P., and Maringer, D., (2007). The Threshold Accepting Optimization Algorithm in Economics and Statistics, In: EJ Kontoghiorges and C Gatu (Eds.), Optimisation, Econometric and Financial Analysis. Advances in Computational Management Science Vol. 9, Springer, Berlin, Heidelberg.