2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Using Harmony Search with Multiple Pitch Adjustment Operators for the Portfolio Selection Problem

Nasser R. Sabar and Graham Kendall, *Senior Member IEEE*

Abstract—**Portfolio selection is an important problem in the financial markets that seeks to distribute an amount of money over a set of assets where the goal is to simultaneously maximize the return and minimize the risk. In this work, we propose a harmony search algorithm (HSA) for this problem. HSA is a population based algorithm that mimics the musician improvisation process in solving optimization problems. At each iteration, HSA generates a new solution using a memory procedure which considers all existing solutions and then perturbs them using a pitch adjustment operator. To deal with different instances, and also changes in the problem landscape, we propose an improved HSA that utilizes multiple pitch adjustment operators. The rationale behind this is that different operators are appropriate for different stages of the search and using multiple operators can enhance the effectiveness of HSA. To evaluate and validate the effectiveness of the proposed HSA, computational experiments are carried out using portfolio selection benchmark instances from the scientific literature. The results demonstrate that the proposed HSA is capable of producing high quality solutions for most of the tested instances when compared with state of the art methods.**

I. INTRODUCTION

Portfolio selection (PS) is of interest to researchers and practitioners, due to its importance in financial engineering [1], [2], [3]. PS is concerned with how to invest a given amount of money over a set of assets. The main goal is to maximize the return and minimize the risk. Given a set of assets, investors select a subset of those assets to form a single portfolio that could simultaneously maximize their return and minimize the risk. However, aiming for higher returns will usually result in higher risk. Consequently, asset selection is the most crucial part and determining the best combination of assets is not a trivial task. Furthermore, the risk of a formed set of assets might be less than an individual asset [4], [5].

Markowitz [1], [2] introduced the mean–variance PS model that takes into consideration the expected return and risk of the formed portfolio. The Markowitz PS model is treated as a quadratic programming problem and thus, when the number of assets becomes large, an efficient algorithm that can find the optimal solution within a reasonable time is not known to exist [5], [6]. Researchers have therefore resorted to heuristic and meta-heuristic algorithms to find good quality solutions in acceptable amount of time. Heuristic and meta-heuristic algorithms can help investors in determining a portfolio that can satisfy their particular demand [4]. Example of heuristics and meta-heuristic algorithms that have been proposed for PS are: genetic algorithm [4], tabu search [4], simulated annealing [4], particle swarm optimization [7] and evolutionary systems [8], [9], [10], [11], [12].

In this work, we propose a Harmony Search Algorithm (HSA) for portfolio selection. HSA is a population-based algorithm that mimics the musician improvisation process in solving optimization problems [13]. In HSA, a new solution is generated using a memory procedure, which is then perturbed using a pitch adjustment operator. The perturbation step of HSA is analogous to the mutation operator in genetic algorithms. Thus, as in genetic algorithms, where different mutation operators are suited to different instances or different stages of the search, different pitch adjustment operators might be needed to deal with different instances or problem landscape changes [14], [15], [16]. Therefore, to enhance the effectiveness of HSA, we propose an improved HSA that utilizes multiple pitch adjustment operators in such a way that different operators may be used at different points of the search. The PS benchmark instances [17] that have been adopted by other researchers are used to evaluate the performance of the proposed HSA. The results demonstrate the effectiveness of the proposed HSA over other algorithms that have been presented in the scientific literature.

II. PROBLEM DESCRIPTION

Markowitz's mean–variance model has been criticized for considering unrealistic assumptions that might not exist in the real world [4], [5], [6]. Thus, some extensions and improvements have been proposed in the literature. A notable extension is the constrained portfolio problem that involves cardinality and boundary constraints that aim to reduce the transaction costs and avoid small/large holdings. The cardinality constraint restricts the number of assets that can be included in each portfolio. The boundary constraint restricts the proportion of each asset in the formed portfolio within a lower and upper bound. In this work, we consider the formulation of the extended model that involves cardinality and boundary constraints [4], [5], [6]:

$$\text{minimize } \lambda\left[\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \alpha_{ij}\right] + (1-\lambda)\left[-\sum_{i=1}^{n} w_i \mu_i\right] \quad (1)$$

$$\text{Subject to } \sum_{i=1}^{n} w_i = 1 \quad (2)$$

Nasser R. Sabar is with The University of Nottingham Malaysia Campus, Jalan Broga, 43500 Semenyih, Selangor, Malaysia (e-mail: Nasser.Sabar@nottingham.edu.my)
Graham Kendall is with The University of Nottingham, UK and also with The University of Nottingham Malaysia Campus, Jalan Broga, 43500 Semenyih, Selangor, Malaysia (e-mail: Graham.Kendall@nottingham.edu.my)

$$\sum_{i=1}^{n} s_i = K \quad (3)$$

$$\varepsilon_i s_i \le w_i \le \delta_i s_i, \; i = 1,...,n \quad (4)$$

$$s_i \in \{0,1\}, \; i = 1,...,n \quad (5)$$

where n represents the total number of assets, w_i represents the proportion of the i^{th} asset, α_{ij} is the connivance between i^{th} and j^{th} assets, λ is the risk aversion, λ=[0, 1], μ_i represents the expected return of the i^{th} asset, K represents the preferred invested assets in a portfolio, s_i is a decision variable representing whether the i^{th} asset has been selected or not, and ε_i and δ_i respectively represent the upper and lower bounds. The above equations are treated as a mixed integer programming. There is no known efficient algorithm that can solve these models in reasonable times. Portfolio selection can be considered as a combination of two sub-problems: the problem of selecting the subset of assets to form a portfolio and the problem of deciding the proportions of the selected assets. Therefore, in this work, we propose an improved harmony search algorithm for PS.

III. THE PROPOSED ALGORITHM

The harmony search algorithm (HSA), proposed in [13], is a population-based stochastic search algorithm that imitates the musical improvisation process. Similar to other population-based algorithms, HSA operates on a population of solutions that is iteratively improved over a number of generations. At each generation, HSA generates a new solution using three procedures; harmony memory consideration, random consideration and pitch adjustment. Then the new solution will replace the worse one in the population if it is better in terms of quality [18]. HSA has five steps illustrated in Figure 1 and described below.

Step 1: **Initialize HSA parameters**. This step is concerned with setting the main parameters of the HSA, these being:

- Harmony memory size (HMS), which represents the population size or the number of solutions to be stored in the harmony memory (HM).
- Harmony memory consideration rate (HMCR). This parameter is used during the solution generation process which decides whether the components or the decision variables of the new solution should be selected from the existing ones in the HM or randomly created. HMCR takes a real value between zero and one.
- The pitch adjustment rate (PAR) takes a real value between zero and one, and is used to decide whether to adjust the components that have been chosen from the HM.
- The maximum number of generations or improvisations (MNI) represents the stopping condition, based on the number of iterations.

In this work, the PS parameters are also set in this step; the maximum cardinality and boundary constraints

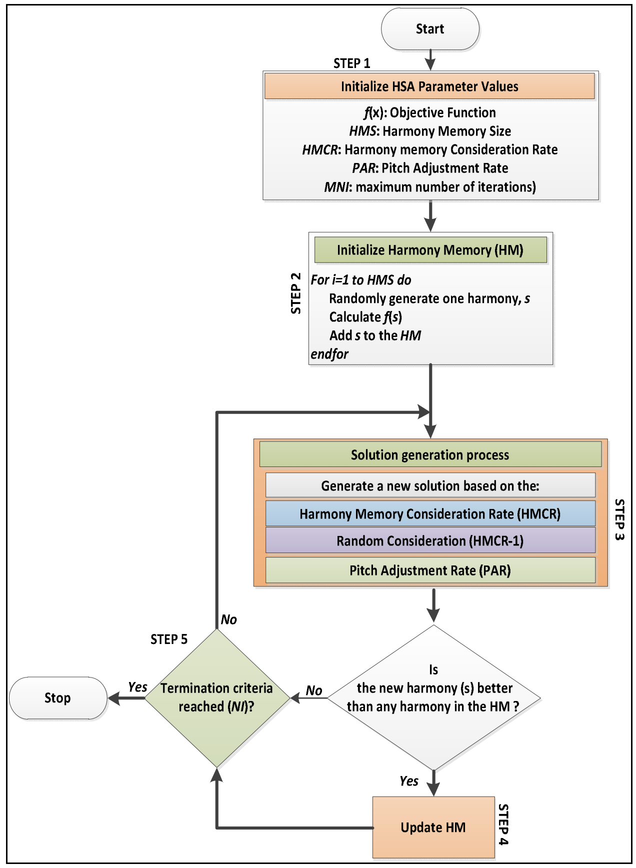

Figure 1: HSA algorithm

Step 2: Initialize the harmony memory (HM). HM contains a set of solution and its size is equal to HMS. In this step, HSA creates a set of the solutions using either a random or heuristic method and then adds them to the HM. To deal with PS, in this work, each solution is represented by a two-dimensional vector and the vector size is equal to the total number of assets, n. Figure 2 shows an example of PS solution representation, where the first row of the vector represents the cardinality which takes either 0 or 1, where 1 indicates that the corresponding asset is selected, while 0 indicates non-selection. The second row of the vector represents the boundary value of the selected asset which takes a real value within the predefined boundary constraints. In this work, the set of the HM initial solutions are randomly generated by assigning for each vector cell (solution decision variable) of the first row either 0 or 1 while makes sure that the maximum cardinality is respected. Next, we assign for each of the selected asset, a real number within the predefined boundary constraints that is represented by the second row of the vector. Next, we calculate the fitness value of the created solution using Equation (1) and add the solution to the HM. We repeat this procedure until the number of generated solutions is equal to HMS.

The portfolio index	*1*	*2*	*3*	.	.	.	*N*
The cardinality	1	1	0	0	0	1	1
The boundary value	0.3	0.5	0	0	0	0.4	0.1

Figure 2: An example of the HSA solution representation for the PS

Step 3: **Generates or improvises a new solution**. This step generates (improvises) a new solution from scratch according to HMCR and the PAR values using the following rules:

- ***Memory consideration rule***. This rule first creates an empty solution (an empty vector) with size equal to the total number of assets, *n*. Then it loops through the solution decision variables (vector cells) one by one and decides to either select the value of the current decision variable from the existing solutions in HM or randomly set it according to the HMCR value. More precisely, this rule generates, for each decision variable, a random number, *r*, between zero and one. Then, if *r* is less than HMCR, select one solution from HM at random and set the current decision variable of the new solution same as the corresponding decision variable of the selected HM solution. Otherwise, the current decision variable is randomly initialized. Once a complete solution is generated, we use Equations (1-5) to check the constraint violations (the maximum cardinality and boundary constraints) and the fitness value.

- ***Pitch adjustment rule (PAR)***. The pitch adjustment rule further adjusts the decision variable values that have been selected from the HM solutions according to the PAR value. Precisely, this rule generates, for each decision variable that was selected from the HM solutions, a random number, *r*, between zero and one. If *r* is less than PAR, the value of this decision variable will be adjusted by adding or subtracting a predetermined value from it. In this work, the pitch adjustment rule is responsible for two tasks. Firstly, since the newly generated solution is not guaranteed to be feasible because the cardinality constraint might be violated. The pitch adjustment rule seeks to turn an infeasible solution into a feasible one by randomly selecting one decision variable and then flipping its value. That is, if the value of the selected decision variable is 0 it will be changed to 1 and initialize its boundary; otherwise it will be changed to 0. This process is repeated until the infeasible solution becomes feasible. Secondly, the pitch adjustment rule acts as a local search algorithm that seeks to further improve the current solution for a certain number of iterations using multiple adjustment operators. We use multiple adjustment operators due to the fact that different instances have different characteristics that may require different operators to effectively explore the search space. Thus by using multiple adjustment operators we can utilize their strengths to cope with the landscape changes that may occur [15]. The proposed multiple pitch adjustment operators work as follows: take the current solution as an input and repeat the following for a predefine number of iterations (we use 20 non-improving iterations fixed based on preliminary experiments), randomly select one decision variable and check its value. If the value of the selected decision variable is 1, then randomly select one of the following three operators to adjust the boundary of the selected asset within the predefined range:

  - A parameterized Gaussian mutation, $N(0, \sigma2)$, where σ=0.5 is the standard division.
  - Same as above but $\sigma = 0.3$.
  - $x_j = x_j + F * (x_{1,j} - x_{2,j})$ where x_j is the current decision variable value, $x_{1,j}$ is the decision variable of the best solution in the population, $x_{2,j}$ is the decision variable of the worst solution in the population and F=0.1 [19].

  Next calculate the fitness value of the adjusted solution using Equation (1). If the fitness value of the adjusted solution is better than the current one replace the current solution with the adjusted solution. Otherwise discard it and start a new iteration.

Step 4: Update HM. This step compares the fitness value of the newly generated solution with the worse one in HM. The worse solution in HM will be replaced by the new one if the new one has a better fitness value.

Step 5: The termination condition. This step decides whether to terminate HSA or start a new iteration.

IV. Experimental Setup

This section first discusses the characteristics of the selected benchmark instances followed by the parameter settings of the proposed algorithm. The proposed algorithm is implemented in Java using a PC running Linux Ubuntu OS with 2.2 GHz Quad-Core processor and 2 GB RAM.

A. Benchmark Instances

The benchmark instances that are available via the OR-library [17] are used to evaluate the performance of the proposed algorithm against the state of the art methods. The benchmark has five different instances that represent the weekly prices for five different countries. The main characteristics of these instances are presented in Table 1, where *n* is the total number of the assets, *K* the maximum number of assets in a formed portfolio (cardinality), ε_i (i=1,…, n) minimum limit of asset's proportion and δ_i (i=1,…, n) the maximum limit of asset's proportion [17].

TABLE 1 THE CHARACTERISTICS OF PS BENCHMARK

#	*Name*	*Country*	*n*	*k*	ε_i	δ_i
1-	Hang Seng	Hong Kong	31	10	0.01	1
2-	DAX 100	Germany	85	10	0.01	1
3-	FTSM 100	UK	89	10	0.01	1
4-	S&P 100	USA	98	10	0.01	1
5-	Nikkei	Japan	255	10	0.01	1

B. *Parameter settings*

The proposed algorithm has a few parameters that need be set in advance and they were set based on a preliminary experiment. The utilized parameter values are reported in Table 2. The parameter λ of Equation (1) was tested using 51 different values and each value is tested for 1000*n fitness evaluations (the same as in [4] and [7]).

TABLE 2 THE PARAMETER SETTINGS

#	*Name*	*Value*
1-	Harmony memory size, *HMS*	30
2-	Harmony memory consideration rate, *HMCR*	0.8
3-	Pitch adjustment rate, *PAR*	0.8
4-	Pitch adjustment stopping condition	20 non-improving iterations
5-	Maximum number of generations, *MNI*	1000*n fitness evaluations

V. RESULTS AND COMPARISONS

In this section, we first evaluate the effectiveness of using multiple pitch adjustment operators within HSA. Therefore, we tested four different HSA variants as follows:

- MHSA: the proposed HSA that use multiple pitch adjustment operators (three operators).
- HSA1: utilize the first pitch adjustment operator only.
- HSA2: utilize the second pitch adjustment operator only.
- HAS3: utilize the third pitch adjustment operator only.

The results of the four HSA variants (MHSA, HSA1, HSA2 and HSA3) are compared using a Wilcoxon test with 0.05 critical level (the results of 31 runs). The *p*-value of MHSA versus HSA1, HSA2 and HSA3 is presented in Table 3, where "+" indicates that the MHSA is statistically better than the compared algorithm (*p*-value < 0.05), "-" indicates that the compared algorithm is better than MHSA (*p*-value > 0.05) and "=" indicates both algorithms have similar performance (*p*-value=0.05). As can be seen from Table 3, MHSA is statistically better than HSA1 and HSA2 on all tested instances. MHSA is better than HSA3 on 3 out of 5 tested instances, not statistically significant on 1 instance and performs the same as HSA3 on 1 instance. These positive results justify the use of multiple pitch adjustment operators within HSA in order to deal with various instances as well as the issue of landscape changes.

TABLE 3 THE P-VALUE OF THE COMPARED HSA VARIANTS

	MHSA vs.	**HSA1**	**HSA2**	**HSA3**
#	***Name***	***p*-value**	***p*-value**	***p*-value**
1-	Hang Seng	+	+	=
2-	DAX 100	+	+	-
3-	FTSM 100	+	+	+
4-	S&P 100	+	+	+
5-	Nikkei	+	+	+

We now compare MHSA results with the following algorithms that have been proposed in the scientific literature:

- Tabu search algorithm (TS) proposed in [4].
- Simulated annealing (SA) proposed in [4].
- Genetic algorithm (GA) proposed in [4].
- Particle swarm optimization (PSO) proposed in [7].

As in [4] and [7], the average results of MHSA over 31 independent runs are compared with *TS*, *SA*, *GA* and *PSO* based on the minimum mean percentage error (MP%) which is shown in Table 4, where the best obtained results are highlighted in bold font. As the table indicates, the proposed MHSA obtained the best results for 4 out of 5 tested instances and being slightly inferior on one instance (S&P 100 instance). Considering the average results (last row in Table 4), MHSA produced the best average results compared to other algorithms (*GA*, *SA*, TS and *PSO*).

TABLE 4 THE RESULTS OF MHSA COMPARED TO OTHER ALGORITHMS

#	*Name*	*MHSA*	*GA*	*SA*	*TS*	*PSO*
1-	Hang Seng	**1.0950**	1.0974	1.0957	1.1217	1.0953
2-	DAX 100	**2.5411**	2.5424	2.9297	3.3049	2.5417
3-	FTSM 100	**1.0731**	1.1076	1.4623	1.6080	1.0628
4-	S&P 100	1.6898	1.9328	3.0696	3.3092	**1.6890**
5-	Nikkei	**0.6726**	0.7961	0.6732	0.8975	0.6870
Average		**1.41432**	1.49526	1.8461	2.04826	1.41516

In Table 5 we compare the computational time (seconds) of MHSA against the compared algorithms, where the best computational time is indicated in bold. As shown in Table 5, the computational time of MHSA is lower than the other algorithms on all tested instances. Given the results presented in Tables 4 and 5, we can conclude that the proposed MHSA is an effective solution method for portfolio selection as it has obtained good quality results for all tested instances within a small computational time compared to other algorithms. The results also demonstrate that the use of multiple pitch adjustment operators does assist HSA in obtaining good results for all tested instances.

TABLE 5 THE COMPUTATION TIME OF MHSA COMPARED TO OTHER ALGORITHMS

#	*Name*	*MHSA*	*GA*	*SA*	*TS*	*PSO*
1-	Hang Seng	**0.3**	172	79	74	4.8
2-	DAX 100	**12.1**	544	210	199	26.8
3-	FTSM 100	**22.15**	573	215	246	31.4
4-	S&P 100	**25.4**	638	242	225	36.6
5-	Nikkei	**67.7**	1964	553	545	75.8

VI. CONCLUSION

This work has proposed a harmony search algorithm for the constrained portfolio selection problem. Harmony search algorithm is a population-based algorithm that operates on a population of solutions and iteratively improves them for a predefined number of iterations. Our proposed harmony search algorithm uses two types of pitch adjustment procedures. The first one aims to turn an infeasible solution into a feasible one. Whilst, the second one is a local search algorithm that seeks to improve the current solution using multiple pitch adjustment operators in order to deal with different instance characteristics as well as landscape changes that might occur during the search process. The performance of the proposed algorithm is validated using the constrained portfolio selection problem benchmark instances. The results demonstrate that the proposed algorithm outperforms other algorithms that have been proposed in the scientific literature, on 4 out of 5 tested instances. The computational time of the proposed algorithm is lower than other algorithms across all instances. These results indicate that the proposed algorithm is an effective solution method for the constrained portfolio selection problem.

REFERENCES

[1] H. Markowitz, "Portfolio selection*," *The journal of finance,* vol. 7, pp. 77-91, 1952.

[2] H. Markowitz, *Portfolio selection: efficient diversification of investments*: John Wiley and Sons, New York, 1959.

[3] H. Varian, "A portfolio of Nobel laureates: Markowitz, Miller and Sharpe," *The Journal of Economic Perspectives,* vol. 7, pp. 159-169, 1993.

[4] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, "Heuristics for cardinality constrained portfolio optimisation," *Computers & Operations Research,* vol. 27, pp. 1271-1302, 2000.

[5] R. Moral-Escudero, R. Ruiz-Torrubiano, and A. Suarez, "Selection of optimal investment portfolios with cardinality constraints," in *IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, 2006, pp. 2382-2388.

[6] A. Fernández and S. Gómez, "Portfolio selection using neural networks," *Computers & Operations Research,* vol. 34, pp. 1177-1191, 2007.

[7] G.-F. Deng, W.-T. Lin, and C.-C. Lo, "Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization," *Expert Systems with Applications,* vol. 39, pp. 4558-4566, 2012.

[8] G. Kendall and Y. Su, "Imperfect evolutionary systems," *Evolutionary Computation, IEEE Transactions on,* vol. 11, pp. 294-307, 2007.

[9] G. Kendall and Y. Su, "A particle swarm optimisation approach in the construction of optimal risky portfolios," in *In Proceedings of the 23rd IASTED International Multi-Conference Artificial Intelligence and Applications*, 2005, pp. 140-145.

[10] G. Kendall and Y. Su, "Learning with imperfections-a multi-agent neural-genetic trading system with differing levels of social learning," in *2004 IEEE Conference on Cybernetics and Intelligent Systems*, 2004, pp. 47-52.

[11] G. Kendall, "A multi-agent based simulated stock market-testing on different types of stocks," in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.* , 2003, pp. 2298-2305.

[12] G. Kendall and Y. Su, "The co-evolution of trading strategies in a multi-agent based simulated stock market through the integration of individual learning and social learning," in *The 2003 International Conference on Machine Learning and Applications (ICMLA'03)*, 2003, pp. 200-206.

[13] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation,* vol. 76, pp. 60-68, 2001.

[14] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan*, et al.*, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society,* vol. 64, pp. 1695-1724, 2013.

[15] N. R. Sabar, M. Ayob, G. Kendall, and Q. Rong, "Grammatical Evolution Hyper-Heuristic for Combinatorial Optimization Problems," *Evolutionary Computation, IEEE Transactions on,* vol. 17, pp. 840-861, 2013.

[16] N. R. Sabar, M. Ayob, R. Qu, and G. Kendall, "A graph coloring constructive hyper-heuristic for examination timetabling problems," *Applied Intelligence,* vol. 37, pp. 1-11, 2012.

[17] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society,* vol. 41, pp. 1069-1072, 1990.

[18] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences,* vol. 233, pp. 126-140, 2013.

[19] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization,* vol. 11, pp. 341-359, 1997.