A Multi-Objective Continuous Genetic Algorithm for Financial Portfolio Optimization Problem

Yacine KESSACI
Worldline
Zone Industrielle A, Rue de la Pointe
Seclin 59113, France
Yacine.Kessaci@worldline.com

ABSTRACT

Financial portfolio management with both quantity and cardinality constraints can be modeled as a NP-hard continuous optimization problem where two objectives are optimized: maximizing the return of the portfolio and minimizing its risk. In this paper, we propose a work that aims at developing and tuning a multi-objective continuous genetic algorithm that gives the best Pareto-set of portfolios with different trade-offs between objectives. Experiments have been conducted using realistic pricing history of the CAC40 stock market. The reported results show the configuration of the genetic algorithm with the best performance.

KEYWORDS

Financial portfolio management, real world problem, continuous optimization problem, multi-objective optimization, genetic algorithm

ACM Reference format:
Yacine KESSACI. 2017. A Multi-Objective Continuous Genetic Algorithm for Financial Portfolio Optimization Problem. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3075977

1 INTRODUCTION

Maximizing the returns obtained from the stocks is one of the main questions since the establishment of the financial markets. This problem has been modeled as a portfolio containing different amounts of stocks [3] under the name of Modern Portfolio Theory (MPT).

The underlying addressed problem is a variation of the MPT problem integrating a risk-free asset and both quantity and cardinality constraints.

This problem can be formalized as a multi-objective continuous optimization problem where two criteria, the return and the risk of the portfolio, have to be maximized and minimized respectively under the aforementioned constraints.

Contrary to the original MPT, the constraints make this problem NP-hard [2]. Therefore, metaheuristic methods – and evolutionary algorithms more specifically – are good candidates to tackle this problem [4].

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3075977

1.1 Problem Modeling

The objective functions of our approach aim to **minimize** the portfolio risk (*Risk*) and **maximize** the portfolio return (*Return*) while respecting different constraints which are: the number of stocks to deal with (*cardinality*), the maximum quantity to invest in each stock (*quantity*), the maximum amount to keep safe as a risk-free asset (*quantityRFA*), having only positive investments (weights) and keeping the sum of weights equaling 1.

2 GENETIC ALGORITHM

2.1 Problem Encoding

A portfolio contains $N + 1$ assets: N stocks and 1 risk-free asset (RFA) that may be managed. We choose to encode a portfolio as a vector of real values of size $N + 1$.

2.2 Genetic Algorithm Specification

Our proposed algorithm is based on the NSGA-II [1] a multi-objective genetic algorithm (GA) with an archiving policy based on the crowding distance. The instantiation of GA requires the definition of variation – crossover and mutation – operators and a selection mechanism to evolve and improve the quality of the obtained solutions. After a fixed number of generations it gives a set of Pareto optimum solutions.

Mutation operator. First, two integers i and j such that $1 \leq i < j \leq N + 1$ are randomly generated. Then, according to a probability value, either the swap operator is applied and the weights of the stocks i and j are swapped or the shift operator is applied and the weight of stock i is assigned to stock j and all the weights of the stocks between i and j are shifted to the left. In NSGA-II, the mutation is applied on a solution according to a probability p_m.

Crossover operator. It has been specifically designed and is inspired by a version of a continuous crossover mechanism [5]. Let s_1 and s_2 as the first and the second parent respectively the offspring s'_1 is generated as follows:

- generate $\beta \in [0; 1]$ randomly.
- generate two integers i and j such that $1 \leq i < j \leq N + 1$.
- keep in a temporary s all values of s_1.
- apply the formula $s'_1 = \beta \times s_1 + (1 - \beta) \times s_2$ between each couple of values of s_1 and s_2 and stores the result in s'_1.
- replace the values of s'_1 by the ones of s for the ones located before i and after j.

The solution s'_2 is generated using the same method by considering s_2 as the first parent and s_1 as the second parent and by generating a new value of β.

	crossover		mutation	selection
Config.	p_c	*strategy*	p_m	*strategy*
A1	0.25	rand	0.35	stoch. tournament
A2	0.25	rand	0.35	fixed tournament (size 10)
A3	0.25	rand	0.35	wheel (size 10)
A4	0.45	rand – split – keep zero	0.6	wheel (size 10)
A5	0.45	rand – split – RFA ≠ 0	0.6	wheel (size 10)
A6	0.45	full – split – keep zero	0.6	wheel (size 10)
A7	0.45	full – split – RFA ≠ 0	0.6	wheel (size 10)

Table 1: Configurations of GA.

The indexes i and j can be either randomly generated (`rand` crossover) or set respectively to 1 and $N+1$ in order to consider the whole chromosome (`full` crossover). Besides, the crossover that we propose distinguishes between the type of the assets. It alternates between modifying the part of the chromosome containing the volatile assets (stocks) and the part with the risk-free asset. Hence it crosses only the assets belonging to the same type (`split` crossover). That helps to explore in a clever way the search space by avoiding the total destruction of the solution. The separation of the crossover process between the asset types lets use the RFA as a pivot around which we explore the neighborhood of the stocks.

Note that some of the constraints of the portfolio management problem are integrated in the crossover process. However, some others can be missed after the variation of the solution Both a checking and a fixing process are called to solve the remaining issues in order to keep the solution feasible.

In the same way as the mutation probability, a crossover probability p_c is defined.

3 EXPERIMENTS AND RESULTS

3.1 Experimental Protocol

GA evolves a population of 100 solutions during a number of 1000 iterations (generations). Because of the stochasticity of the GA, the performance of each tested GA is evaluated from 30 independent executions. The reported results are the average performance over these runs. In this experiments we use the following values for the constraints: a maximum quantity per stock of 0.3, a cardinality of 10 and, a quantity of Risk-free Asset of 0.6.

The CAC40 stock exchange history that we have used extends from 02/04/2013 to 07/27/2015 (18 months) where 17 months has been used to predict the information of the stocks and 1 to evaluate the performance of the different GA configurations. The prediction estimates both the risk and the expected return. Besides, in our experiments, a portfolio solution is always encoded as a vector of size 41. 40 stocks of the CAC40 plus one risk-free asset. The number of invested stocks (strictly positive weights) is defined by the $cardinality$.

3.2 Tuning of GA

The performance of a genetic algorithm may be improved with a good parameters setting. In this section, we compare the performance of seven different configurations of our GA. Table 1 reports the seven configurations of the evaluated GAs. Two sets of configurations are clearly differentiable: (A1, A2, A3) and (A4, A5, A6, A7). Indeed the crossover and mutation probabilities are smaller in the first set than in the second one *i.e.,*the GAs of the second set enable more diversification. Besides the crossover strategy differs by the `split` possibility that differentiates the volatile stocks from the RFA. Configurations A4, A5, A6, A7 propose to test the crossover ability of improvement when are considered the whole stocks of each individual (`full`) or only the stocks between two random positions (`rand`). Moreover, the strategy `keep zero` forces the stocks with a zero weight in one of the parents to remain equal to zero while the strategy `RFA ≠ 0` forbids to generate offspring individuals with a null weight for the risk-free asset.

Indicator	Wilcoxon signed-rank test
Hypervolume	A7 ≫ A5 ≈ A6 ≫ A4 ≫ A1 ≈ A2 ≈ A3
Epsilon	A7 ≫ A5 ≈ A6 ≫ A4 ≫ A1,A2,A3 and A1 ≈ A2,A3 and A2 ≫ A3

Table 2: Comparison between 7 configurations of GA for the hypervolume metric and the epsilon metric.

The performance of each configuration of the multi-objective GA is computed from the Pareto front outputted using two binary metrics: hypervolume difference and epsilon difference indicators [6]. Table 2 gives the results of the non-parametric statistical Wilcoxon signed-rank test (p-value set to 0.05) of the 7 configurations of GA for both hypervolume and epsilon metrics. One configuration statistically outperforms the other ones for both hypervolume and epsilon metrics. Indeed, the statistical test demonstrates the robustness of configuration A7. The results highlighted the importance of the variation operator especially the crossover. The tuning process designated the A7 crossover configuration as to be the one with the best impact on the solutions' quality.

4 CONCLUSION

In this paper, we presented a Pareto multi-objective genetic algorithm for constrained portfolio management problem . The objective is to maximize the return and reduce the risk of the portfolios according to a prediction model. This model relies on trend indicators for computing the expected returns and on variance covariance matrix for estimating the risks.

Our approach has been evaluated over the CAC40 stock exchange history of 1 month. Our experimentation process included the evaluation of the performance of the genetic algorithm based on 7 variation operators configurations. The tuning showed that the crossover of the A7 configuration was the most efficient one.

The major perspective of this work is to integrate the genetic algorithm within a portfolio management tool in real conditions.

REFERENCES

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. Apr. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on* 6, 2 (Apr), 182–197. DOI: http://dx.doi.org/10.1109/4235.996017
[2] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., NY, USA.
[3] Harry M. Markowitz. 1991. *Portfolio Selection: Efficient Diversification of Investments, 2nd Edition.* Wiley.
[4] K. Metaxiotis and K. Liagkouras. 2012. Multiobjective Evolutionary Algorithms for Portfolio Management: A comprehensive literature review. *Expert Systems with Applications* 39, 14 (2012), 11685 – 11698.
[5] Alden H. Wright. 1991. Genetic Algorithms for Real Parameter Optimization. In *Foundations of Genetic Algorithms.* Morgan Kaufmann, 205–218.
[6] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. 2003. Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on* 7, 2 (2003), 117–132.