# Evolutionary Algorithms and the Cardinality Constrained Portfolio Optimization Problem

Felix Streichert, Holger Ulmer, and Andreas Zell

Center for Bioinformatics Tübingen (ZBIT), University of Tübingen,
Sand 1, 72076 Tübingen, Germany,
streiche@informatik.uni-tuebingen.de

**Abstract.** While the unconstrained portfolio optimization problem can be solved efficiently by standard algorithms, this is not the case for the portfolio optimization problem with additional real world constraints like cardinality constraints, buy-in thresholds, roundlots etc. In this paper we investigate two extensions to Evolutionary Algorithms (EA) applied to the portfolio optimization problem. First, we introduce a problem specific EA representation and then we add a local search for feasible solutions to improve the performance of the EA. All algorithms are compared on the constrained and unconstrained portfolio optimization problem.

## 1 Introduction

Evolutionary Algorithms (EA) have been successfully applied to many optimization problems in science and technology. Some researchers also used EA on financial engineering problems like the portfolio optimization problem [4,1,8]. In this paper we compare the impact of several EA solution representations and the application of local search on the performance of EA on the portfolio optimization problem.

## 2 The Portfolio Optimization Problem

Using the standard Markowitz mean-variance approach [6], the unconstrained portfolio optimization problem is given as

$$\text{minimizing the variance of the portfolio}: \sum_{i=1}^{N}\sum_{j=1}^{N} w_i \cdot w_j \cdot \sigma_{ij}, \quad (1a)$$

$$\text{maximizing the return of the portfolio}: \sum_{i=1}^{N} w_i \cdot \mu_i, \quad (1b)$$

subject to

$$\sum_{i=1}^{N} w_i = 1\ , \quad (2a)$$

$$0 \leq w_i \leq 1 \quad ; \quad i = 1, .., N \quad (2b)$$

where N is the number of assets available, $\mu_i$ the expected return of asset $i$, $\sigma_{ij}$ the covariance between asset $i$ and $j$, and finally $w_i$ are the decision variables giving the composition of the portfolio.

The optimization problem as given in Equ. 1 is a multi-objective optimization problem with two competing objectives. First, to minimize the variance (risk) of the portfolio and at the same time to maximize the return of the portfolio. Equ. 2 gives the minimum constraints for a feasible portfolio.
While this portfolio optimization problem is a quadratic optimization problem for which computationally effective algorithms exist, this is not the case if real world constraints are added:
**Cardinality Constraints** restrict the number of assets in the portfolio.

$$\sum_{i=1}^{N} \text{sign}(w_i) = k \tag{3}$$

**Buy-in Thresholds** give the acquisition prices for each asset.

$$l_i \leq w_i; \quad i = 1, .., N \tag{4}$$

**Roundlots** give the smallest volumes $f_i$ that can be purchased.

$$w_i = y_i \cdot f_i; \quad i = 1, .., N \text{ and } y_i \in \mathbf{Z} \tag{5}$$

Other real world constraints can include sector/industry constraints, immunization/duration matching and taxation constraints, but these will not be addressed in this paper.

## 3 The Optimization Algorithm

To solve the multi-objective optimization problem we use a Multi-Objective Evolutionary Algorithm (MOEA) having two different EA representation types with an additional problem specific extension for each representation. To further improve the results we apply a local search for feasible solutions. There are two approaches how to incorporate local search into EA (Memetic Algorithms) either using Lamarckism or relying on the Baldwin effect [10].

### 3.1 Evolutionary Algorithms

EAs are population based stochastic optimization heuristics inspired by Darwin's Evolution Theory. An EA searches through a solution space in parallel by evaluating a set (population) of possible solutions (individuals). An individual gives a solution by representing the decision variables $w_i$.
An EA starts with a random initial population $P_0$. Then the 'fitness' of each individual is determined by evaluating the objective function, Equ. 1. After the best individuals $P'_t$ are selected, new individuals for the next generation $P_{t+1}$ are created from $P'_t$. New individuals are generated by altering the individuals of $P'_t$ through random mutation and by mixing the decision variables of multiple parents through crossover. Then the generational cycle repeats

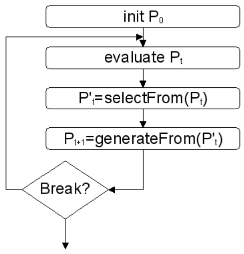


**Fig. 1.** EA scheme

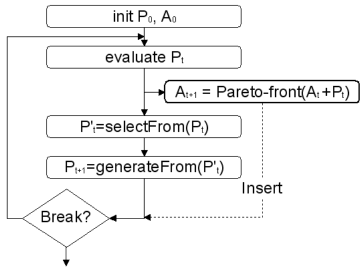


**Fig. 2.** Multi-Objective EA

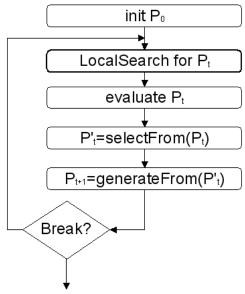


**Fig. 3.** Memetic Alg.

until a breaking criteria is fulfilled, see Fig. 1 for a basic scheme.
There are several alternative EA implementations but only two will be considered in this paper: Genetic Algorithms (GA) [5] are based on a binary representation for real value decision variables and rely on big populations and the crossover operation. Evolutionary Strategies (ES) [9] on the other hand use a direct real value representation and apply sophisticated mutation operators. An extensive overview on EAs can be found in [2].

### 3.2 Multi-Objective Evolutionary Algorithms

Due to the population based search strategy and the simple selection strategy EAs are easy to extend to multi-objective optimization problems. First, by using selection based on multiple objective values like the Pareto-dominance criteria. Secondly by adding an archive population $A_t$ used to maintain the currently known Pareto-front. Zitzler gives a guide to MOEAs in his Ph.D. Thesis [12].
During multi-objective optimization two goals are to be reached. On the one hand the solutions should be as close to the global Pareto-optimal front as possible and on the other hand the solutions should also cover the whole Pareto-front. The first goal is often achieved through elitism by replacing random individuals in $P_t$ with individuals on the Pareto-front $A_t$, see Fig. 2. The second goal can be achieved by punishing individuals that are too close together (Fitness Sharing).

### 3.3 Memetic Algorithms

Memetic Algorithms (MA) [7] extend EA by adding an arbitrary (possibly problem specific) local search heuristic before evaluating the population $P_t$, see Fig. 3. There are two alternatives of how to integrate the local search [10], first by updating only the enhanced objective values (fitness) for each individual (Baldwin Effect) or by also updating the decision variables, so that they can be inherited to the next generation (Lamarckism). An example for MA on the portfolio optimization problem is given in [8].

## 4 Experimental Settings

In our experiments we apply a generational GA population strategy with a population size of 500 individuals. We use tournament selection with a tournament group size of 8 together with objective space based fitness sharing with a sharing distance of $\sigma_{share} = 0.01$. The selection mechanism prefers individuals that are better than other individuals in at least one objective value, i.e. are not dominated by another individual. To maintain the currently known Pareto-front we use an archive of 250 individuals and use $A_t$ as elite to achieve a faster speed of convergence. Details of this MOEA strategy can be found in [11].

We use two different standard EA representations on the portfolio optimization problem: First, a GA binary representation with a 32 bit standard binary encoding (genotype) for each $w_i$ (phenotype). A 3-point-crossover and one-point-mutation is used on the GA genotype, with a crossover probability of ($P_c = 1.0$) and mutation probability of ($P_m = 0.01$). Secondly we use an ES with a real-valued solution representation. In this case the phenotype equals the genotype. We apply a discrete 3-Point-Crossover ($P_c = 0.5$) and local mutation with one strategy parameter for each decision variable ($P_m = 1.0$) on the ES genotype. These parameters were selected from preliminary experiments.

The idea to use a problem specific representation for the portfolio optimization problem is based on the fact that portfolios on the Pareto-front are rarely composed of all available assets, but only a limited selection of assets. The actual composition of the assets in the portfolio resembles a one-dimensional binary knapsack problem. To allow easy removal and adding of assets to the portfolio we added an auxiliary binary bit-mask $b_i$ together with the decision variables $w_i$ to represent a solution. Each bit $b_i$ determines whether the associated asset will be element of the portfolio or not, $w'_i = b_i \cdot w_i$. Both EAs were enhanced with this additional 'knapsack' representation. The extended EAs will be referred to as Knapsack-GA (KGA) and Knapsack-ES (KES).

The second extension is made to improve the number of feasible solutions generated by the EA. Instead of punishing or rejecting infeasible solutions, we apply a 'local search heuristic' to convert an infeasible solution into a feasible one. For example to hold Equ. 2 we limit the range of the EA solution representation and use a standardization step $w'_i = w_i / \sum_{j=1}^{N} w_j$. For cardinality constraints we set all but the $k$ biggest decision variables $w_i$ to zero before standardization. Similar mechanisms were applied for buy-in thresholds and roundlot constraints.

## 5 Results

The comparison of the different EA implementations was performed on benchmark data sets given by Beasley [3]. The numerical results presented here are

performed on the *Hang Seng* data set with 31 assets.
To compare the performance we measure the percentage difference ($\Delta_{area}$) between the area below the Pareto-front generated by the EA and the area below the unconstrained Pareto-front given as reference solution, see Fig. 8. For each experiment 50 independent runs each with 100.000 fitness evaluations were made. For these we calculate the mean, standard deviation, maximum and minimum values and the 90 % confidence interval of the $\Delta_{area}$, which is to be minimized.

### 5.1 Adding the Knapsack representation to the EA individuals

When comparing the GA and ES against the KGA and KES without additional constraints (no $l_i$ and $f_i$) the extended versions clearly outperform the standard EA approaches, see Fig. 4. Especially the KES shows very good convergence behavior. Only in case of $k = 2$ the GA and ES are able to catch up with the extended EA representations. This shows that the assumption that the portfolio optimization problem resembles the binary knapsack problem holds true even without cardinality constraints and that the extended representation is able to search more efficiently than the standard EAs.
With additional buy-in thresholds and roundlots ($l_i = 0.1$ and $f_i = 0.02$) all algorithms performed much worse, see Fig. 5. Although single runs of the extended EAs find reasonable good solutions, the results are rather unreliable.

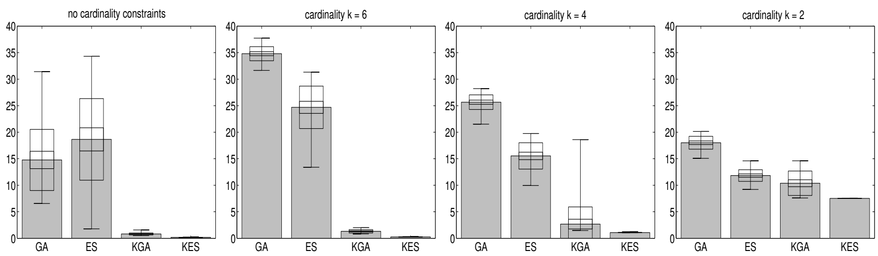


**Fig. 4.** $\Delta_{area}$ for Hang Seng without $l_i$ and $f_i$

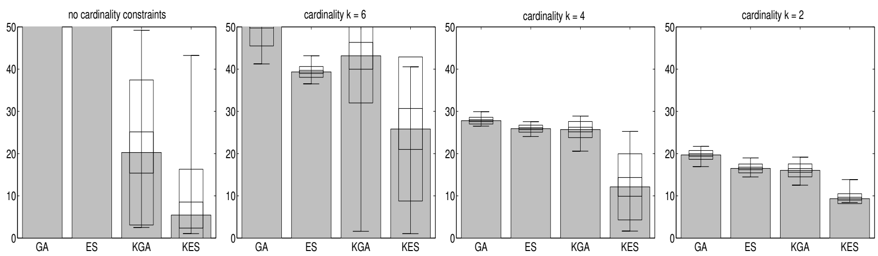


**Fig. 5.** $\Delta_{area}$ for Hang Seng with $l_i = 0.1$ and $f_i = 0.02$

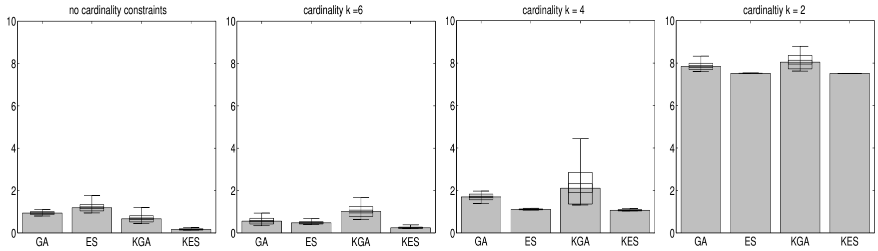


**Fig. 6.** $\Delta_{area}$ for Hang Seng with Lamarckism and without $l_i$ and $f_i$

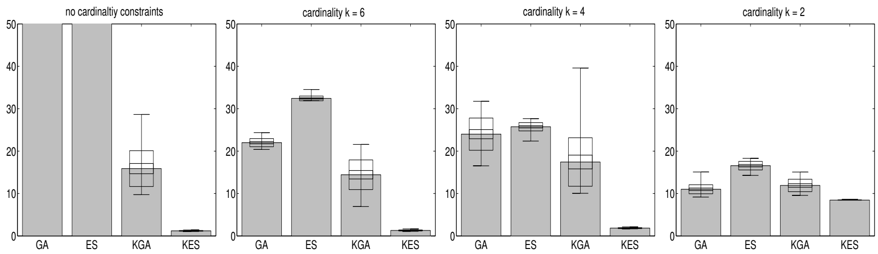


**Fig. 7.** $\Delta_{area}$ for Hang Seng with Lamarckism and $l_i = 0.1$ and $f_i = 0.02$

### 5.2 Applying Lamarckism

The main advantage of the extended EA seems to be the ability to easily change the content of the portfolio. The same effect could be achieved if local search with Lamarckism is used. With Lamarckism assets removed by local search would cause the decision variables to be set to zero. Therefore the resulting vector of decision variables $w_i$ would be sparse. On such a sparse vector single mutations and crossover could cause major changes of the content of the portfolio. With this, standard EAs could move as easily through the space of assets combinations as the extended EA.
The experiments with Lamarckism supports this view, see Fig. 6. The standard EAs become competitive to the extended EAs and especially the ES behaves nearly as good as the KES except for the unconstrained portfolio optimization. In the latter case, Lamarckism can not have the same effect since no assets are removed through local search.
In case of additional constraints all algorithms perform much better and the results become more reliable, see Fig. 7. Even the extended EAs benefit from the Lamarckism because the extended representation is as improper for the discretization of the search space through the roundlot constraints as the standard EAs are for searching for portfolios with limited cardinality. Remarkably, the KES with Lamarckism seems to perform as well as in the unconstrained case. Most likely the discretization of the Pareto-front through the limited archive size levels the effect of the roundlot constraint.

## 6 Conclusions and Future Work

We have shown that the extended EA with the additional knapsack representation is able to solve the portfolio optimization problem more efficiently than the standard EA approaches, due to the improved search capabilities regarding the possible combinations of assets in a portfolio. The KES showed to be superior to the KGA due to the more appropriate ES real-value representation of the decision variables $w_i$. We were also able to produce the same effect by using the Memetic feasibility search in combination with Lamarckism. In this case the standard EAs were able to draw level with the extended EAs and as before the ES with Lamarckism produced better results than the GA with Lamarckism and even better than the KGA.
With additional constraints all algorithms performed not as well except a few good outliers when using the extended EA. Again, Lamarckism is able to improve the algorithms. With Lamarckism the KES reliably produces results only sightly worse than in the unconstrained case. Preliminary experiments hint that a discrete representation for KGA performs much better in case of roundlot constraints and produces equally good results.
Our future research will concentrate on improving the Multi-Objective EA and comparing alternative Multi-Objective EAs on the portfolio selection problem. Another area of improvement could be the local search. There are numerous alternatives to the simple search for feasible solutions, but they have to be carefully evaluated regarding the ability to handle additional constraints.

## Acknowledgements

This research has been funded by the German Federal Ministry of Research and Education (BMBF) under contract No. 03C0309E.

## References

1. S. Arnone, A. Loraschi and A. Tettamanzi (1993) A Genetic Approach to Portfolio Selection, in Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems (3)6:597-604
2. T. Baeck, D.B. Fogel, Z. Michalewicz editors (1997) Handbook of Evolutionary Computation, Oxford University Press, New York, and Institute of Physics Publishing, Bristol
3. J. B. Beasley (1996) Obtaining test problems via the Internet, Journal of Global Optimization, 8:429-433
4. T.-J. Chang, N. Meade, J.B. Beasley and Y.M. Sharaiha (2000) Heuristics for cardinality constrained portfolio optimization, Computers and Operations Research, 27:1271-1302
5. J. Holland (1972) Adaption in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Systems, The University Press of Michigan Press, Ann Arbor

6. H. M. Markowitz (1952) Portfolio Selection, *Journal of Finance*, 1(7):77–91
7. P. Moscato (1989) On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program, C3P Report 826
8. F. Schlottmann, and D. Seese (2001) A Hybrid Genetic-Quantitative Method for Risk-Return Optimization of Credit Portfolios, Proceedings of the Conference of Quantitative Methods in Finance, University of Technology, Sydney, Australia, p. 55
9. H.-P. Schwefel (1995) Evolution and Optimum Seeking, John Wiley & Sons, Inc., New York
10. D.L. Whitley and V.S. Gordon and K.E. Mathias (1994) Lamarckian Evolution, The Baldwin Effect and Function Optimization, in Parallel Problem Solving from Nature (PSN III), Y. Davidor and H.-P. Schwefel and R. Männer eds., Springer, pp. 6-15
11. N. Srinivas, Kalyanmoy Deb (1994) Multiobjective Optimization Using Non-dominated Sorting in Genetic Algorithms, Evolutionary Computation (2)3:221-248
12. E. Zitzler (1999) Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Swiss Federal Institute of Technology (ETH) Zurich. TIK-Schriftenreihe Nr. 30, Diss ETH No. 13398, Shaker Verlag, Germany, ISBN 3-8265-6831-1

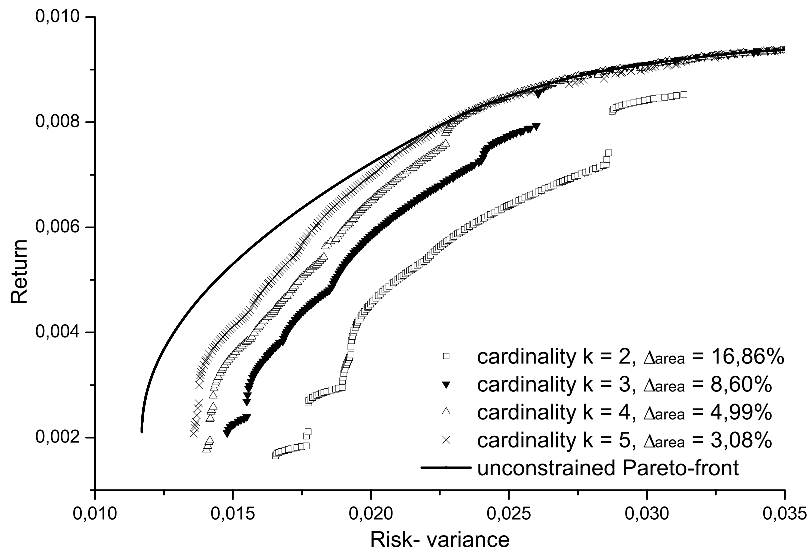


**Fig. 8.** Solutions generated by EA on the *DAX* data set