Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem

Felix Streichert, Holger Ulmer, and Andreas Zell

Centre for Bioinformatics Tübingen (ZBIT), University of Tübingen,
Sand 1, 72076 Tübingen, Germany,
{streiche, ulmerh, zell}@informatik.uni-tuebingen.de
http://www-ra.informatik.uni-tuebingen.de/

Abstract. In financial engineering the problem of portfolio selection has drawn much attention in the last decades. But still unsolved problems remain, while on the one hand the type of model to use is still debated, even the most common models cannot be solved efficiently, if real world constraints are added. This is not only because the portfolio selection problem is multi-objective, but also because constraints may turn a formerly continuous problem into a discrete one. Therefore, we suggest to use a Multi-Objective Evolutionary Algorithm and compare discrete and continuous representations. To meet constraints we apply a repair mechanism and examine the impact of Lamarckism and the Baldwin Effect on several instances of the portfolio selection problem.

1 Introduction

One prominent problem in financial engineering is portfolio selection, i.e. the problem how to invest money most profitable in multiple assets available. In this paper we investigate the application of a Multi-Objective Evolutionary Algorithm (MOEA), a heuristic that is virtually independent of the underlying portfolio selection model used. We investigate the impact of several coding schemes and the application of a repair mechanism together with Lamarckism on the constrained portfolio optimization problem.

First, we give a short introduction to the portfolio selection problem in sec. 1.1 and the related work in sec. 1.2. Then we explain details of the MOEA, the repair mechanism and the different coding schemes we applied in sec. 2. Results on several problem instances are shown in sec. 3 and finally conclusions and an outlook on future work are given in sec. 4 and sec. 5, respectively.

1.1 The Portfolio Selection Problem

The Markowitz mean-variance model [11, 12] gives a multi-objective optimization problem, with two output dimensions. A portfolio p consisting of N assets with specific volumes for each asset given by weights w_i is to be found, which:

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

$$\text{maximizes the return of the portfolio}: \quad \mu_p = \sum_{i=1}^{N} w_i \cdot \mu_i, \tag{2}$$

$$\text{subject to}: \quad \sum_{i=1}^{N} w_i = 1 \quad \text{and} \tag{3}$$

$$0 \leq w_i \leq 1 \tag{4}$$

where $i = 1, .., N$ is the index of the asset, N represents the number of assets available, μ_i the estimated return of asset i and σ_{ij} the estimated covariance between two assets. Usually, μ_i and σ_{ij} are to be estimated from historic data.

While the optimization problem given in equ. 1 and equ. 2 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 maximal number of assets used in the portfolio, $\sum_{i=1}^{N} \text{sign}(w_i) = K$.

Buy-in thresholds give the minimum amount that is to be purchased, i.e. $w_i \geq l_i \quad \forall \quad w_i > 0; i = 1, .., N$.

Roundlots give the smallest volumes c_i that can be purchased for each asset, $w_i = y_i \cdot c_i; \quad i = 1, .., N$ and $y_i \in \mathbb{Z}$.

These constraints are often hard constraints, i.e. they must not be violated. Other real world constraints like sector/industry constraints, immunization/duration matching and taxation constraints can be considered as soft constraints and should be implemented as additional objectives, since this yields the most information. While we do consider the above hard constraints, we currently do not include soft constraints in our experiments, but plan to examine their impact in our future work.

1.2 Related Work

One of the first groups to apply Genetic Algorithms (GA) on the portfolio selection problem were Tettamanzi et al. [1, 10, 9]. They transformed the multi-objective optimization problem (MOOP) into a single-objective problem by using a trade-off function. They used multiple GA populations with individual trade-off coefficients to identify the complete Pareto front. More recently, Crama et al. applied Simulated Annealing (SA) to the portfolio selection problem [5]. They especially pointed out that SA and similar heuristics like GA have the major advantage that they can be easily applied to any kind of portfolio selection model with arbitrary constraints without much modification. For the same reason Beasley et al. compared Tabu Search, SA and GA on the portfolio selection to evaluate their performance [4]. They solved the MOOP by interpreting one objective as constraint and optimizing the other one. The constraint was altered iteratively to get the complete Pareto front. As a conclusion they found that no individual heuristic performed better than the other ones and that only a pooled result of all three heuristics produced a satisfying Pareto front.

Unfortunately, the papers using Evolutionary Algorithms (EA) did not apply multi-objective EAs (MOEA) to the portfolio selection problem, although MOEA have shown to be very useful on similar multi-objective optimization problems [8, 6, 16].

2 Multi-Objective Evolutionary Algorithm

Our MOEA strategy uses a generational GA population strategy with a population size of 500 individuals. We apply tournament selection with a tournament group size of 8 together with objective space based fitness sharing with a sharing distance of $\sigma_{share} = 0.01$ [7]. The selection prefers individuals that are better than other individuals in at least one objective value, i.e. which are not dominated by other individuals. To maintain the currently known Pareto front we use an archive of 250 individuals and use this archive as elite to achieve a faster speed of convergence. Details of this MOEA strategy can be found in [13]. We use one-point mutation with a mutation probability of $p_m = 0.1$ and a discrete 3-point-crossover with $p_c = 1.0$ on all genotypes. For binary genotypes bit-flip mutation is used and in case of the real-valued genotype a gaussian random number with $\sigma = 0.05$ is added to a random decision variable. These parameters for the operators were selected to allow a fair comparison. The general parameters were found in preliminary experiments [14].

As representations we decided to compare bit-string based genotypes using binary or gray-coding to a real-valued genotype. On discretized problem instances, caused by additional roundlot constraints, we also investigated the size of the bit-string from a 32bit 'continuous' and a 7bit 'discrete' representation.

Preliminary experiments indicated that pareto-optimal solutions for the portfolio selection problem are rarely composed of all available assets, but only a limited selection of the available assets, especially in case of cardinality constraints, see Fig. 2. This selection problem resembles a one-dimensional binary knapsack problem, which has already been addressed by means of EA using a binary representation. Therefore, we suggest to use the very same representation in addition to the vector of decision variables $\mathbf{W}$, see Fig. 1. Each bit of the bit-string $\mathbf{B}$ determines whether the associated asset is an element of the portfolio or not, so that the actual value of the decision variable is $w'_i = b_i \cdot w_i$. This is the value that is processed by the following repair algorithm. With this hybrid representation it is much easier for the GA to add or remove the associated assets simply by mutating the additional bit-string. The hybrid representation is altered by mu-

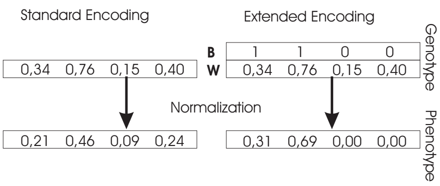

Fig. 1. Comparing the standard representation to the hybrid representation.

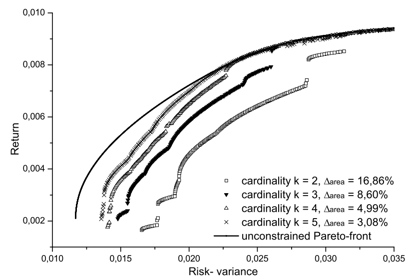

Fig. 2. Solutions generated by EA with the hybrid representation on the *DAX* data set with 81 assets as given in [2].

tating/crossing each genotype element (**B** and **W**) separately from each other. The extended GA is abbreviated KGA (Knapsack-GA).

The GA implementation used encodes each decision variable in the desired range, $w_i \in \{0, 1\}$, but especially the additional constraints given in sec. 1.1 are rather restrictive. Therefore, it is impractical to outright reject all infeasible solutions. This is the reason why we applied a repair algorithm, which searches for the next feasible solution.

To do so the repair algorithm first removes all surplus assets from the portfolio to meet the cardinality constraints by setting the $N - K$ smallest values of w_i to zero and also those assets whose weights are below the given buy-in threshold. For those $w_i > 0$ remaining, the weights are normalized such that $w'_i = l_i + \frac{w_i - l_i}{\sum(w_i - l_i)}$. To meet round-lot constraints the algorithm rounds the $w_i > 0$ to the next round-lot level, $w''_i = w'_i - (w'_i \ \ mod\, c_i)$, after cardinality repair, buy-in repair and normalization was applied. The remainder of the rounding process, $\sum_i (w'_i \ \ mod\, c_i)$, is spent in quantities of c_i on those w''_i, which had the biggest values for $w'_i \ \ mod\, c_i$ until all of the remainder is spent. Since the repair algorithm is deterministic, an individual is always assigned to the same phenotype after repair if the genotype did not change.

Since in a basic implementation the repair mechanism would only determine the phenotype of a GA individual, we compare the performance of the GA with and without Lamarckism to further examine the effect of the repair mechanism. With Lamarckism alters genotype of a GA individual is altered by coding the phenotype back onto the genotype.

3 Experimental Results

The comparison of the different GA implementations is performed on a public benchmark data set provided by Beasley [2]. The numerical results presented here were performed on the *Hang Seng* data set with 31 assets. On this data set we use several combinations of real world constraints to compare the performance of the different GA representations. First, we compare the cardinality constrained portfolio selection problem without and with use of Lamarckism. In a second set of experiments we also add real-world constraints like buy-in thresholds and roundlot constraints.

To compare the performance of the MOEAs we use the S-metric that calculates the hyper volume under the Pareto front [17]. We take the percentage difference (Δ_{area}) between the hyper volume of the Pareto front found by the MOEA and a reference solution of the unconstrained portfolio selection problem, compare Fig. 2, Δ_{area} is to be minimized.

To obtain reliable results we repeat each GA experiment for 50 times for each parameter setting and problem instance. A single GA run is terminated after 100,000 fitness evaluations. We then calculate the mean value, the standard deviation, the maximum and minimum values and the 90 % confidence intervals of the Δ_{area} value to evaluate the performance of each GA setting.

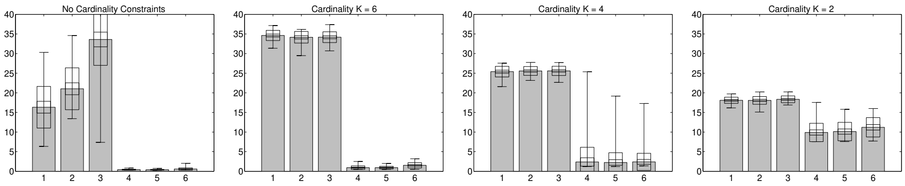

Fig. 3. Δ_{area} for the experiments on the Hang Seng data set $l_i = 0$ and $c_i = 0$ (1: GA binary-coding, 2: GA gray-coding, 3: GA real-valued, 4: KGA binary-coding, 5: KGA gray-coding, 6: KGA real-valued)

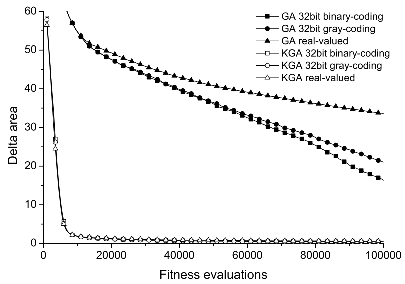

Fig. 4. Δ_{area} on the Hang Seng data set with $K = N$, $l_i = 0$ and $c_i = 0$

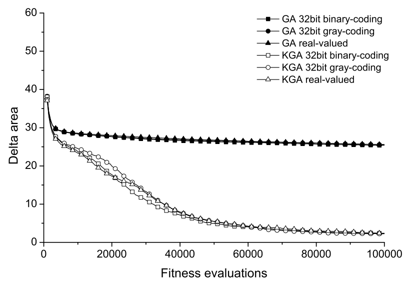

Fig. 5. Δ_{area} on the Hang Seng data set with $K = 4$, $l_i = 0$ and $c_i = 0$

3.1 Results without Additional Constraints

In our experiments we distinguish further between experiments with and without Lamarckism. On the one hand Lamarckism is said to cause premature convergence, while the Baldwin effect on the other hand leads to a neutral search space, which may enable the GA to escape local optima, see [15] for further details. We show that the applied repair mechanism has a quite unexpected result on the constrained portfolio selection problem if Lamarckism is not applied.

Without Lamarckism. On the simplest problem instance without additional constraints the behavior of the hybrid KGA representation clearly outperforms the standard representation on all problem instances, see Figs. 3 - 5. Without cardinality constraints the hybrid KGA nearly instantly converges to very good values of Δ_{area} independent of the coding scheme used for the genotype. Only in case of $K = 2$ the real-valued KGA performs slightly worse than the bit-string based KGAs.

When the standard GA is used on the portfolio selection problem without cardinality constraints, the different genotype coding schemes can be clearly distinguished, see Fig. 4. Here the real-coded GA performs worst, while the binary-coding is better than the gray-coding. But when cardinality constraints are used no such distinctions can be made anymore. This is due to the combined effect of cardinality constraints and the applied repair mechanism. The repair algorithm always selects the K biggest w_i to be part of the portfolio. The remaining w_i are

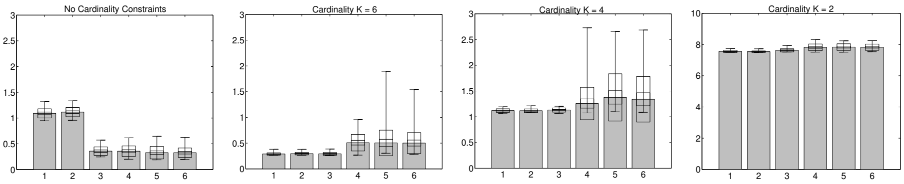

Fig. 6. Δ_{area} for the experiments on the Hang Seng data set with Lamarckism, $l_i = 0$ and $c_i = 0$ (1: GA binary-coding, 2: GA gray-coding, 3: GA real-valued, 4: KGA binary-coding, 5: KGA gray-coding, 6: KGA real-valued)

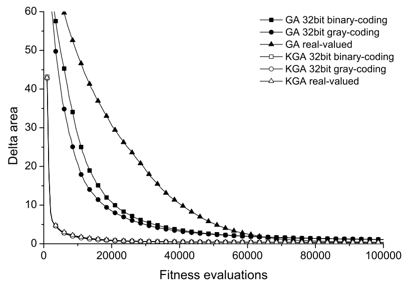

Fig. 7. Δ_{area} on the Hang Seng data with Lamarckism, $K = N$, $l_i = 0$ and $c_i = 0$

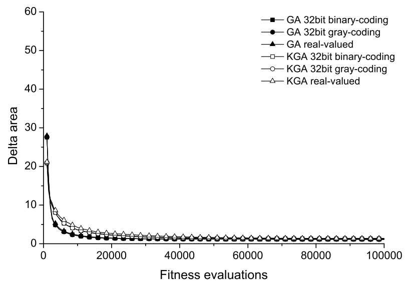

Fig. 8. Δ_{area} on the Hang Seng data with Lamarckism, $K = 4$, $l_i = 0$ and $c_i = 0$

normalized to values of $w'_i \approx 1/K$. The other $N - K$ asset weights are subject to genetic drift, since there is no selection pressure toward sparse vectors $\mathbf{W}$. If any of the previously selected w_i drops out of the portfolio due to mutation or crossover, the biggest of the $N - K$ asset weights takes its place and the values are again normalized to $w'_i \approx 1/K$. This way the standard GA only searches the subspace of portfolio of size K with weights $w_i \approx 1/K$.

With Lamarckism. With cardinality constraints and Lamarckism the standard GA inherits some properties of the hybrid KGA. Since the repair mechanism removes the surplus assets from the portfolio and Lamarckism removes them from the genotype, the standard GA also acts on a sparse vector of $\mathbf{W}$ like the hybrid KGA. This way the standard GA can add and remove assets to and from the portfolio as easily as the hybrid KGA. Now the standard GA is also able to explore the complete subspace of possible portfolio combinations, see Fig. 6. The standard GA even outperforms the KGA reading speed convergence and reliability of the results for $K < N$.

Without cardinality constraints this effect is not as strong, although the results of the standard GA are much better and the speed of convergence is increased notably, see Fig. 7. Here Lamarckism removes neutrality from the search space, which enables the standard GA to remove surplus assets more efficiently and thereby the standard GA converges much faster, compare Fig. 7 to Fig. 4.

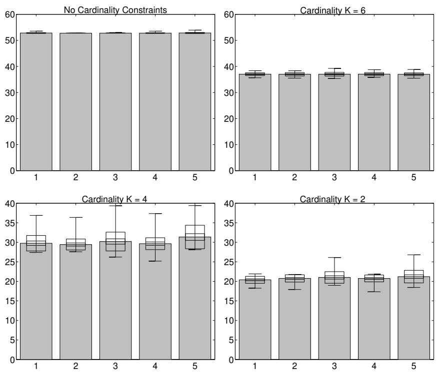

Fig. 9. Δ_{area} for the experiments on the Hang Seng data set with $l_i = 0.08$ and $c_i = 0.008$ (1: GA 32bit binary-coding, 2: GA 7bit binary-coding, 3: GA 32bit gray-coding, 4: GA 7bit gray-coding, 5: GA real-valued)

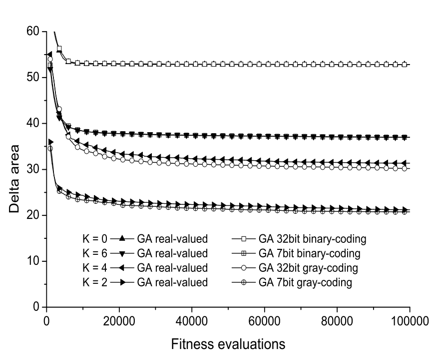

Fig. 10. Δ_{area} on the Hang Seng data set with several cardinality constraints, $l_i = 0.08$ and $c_i = 0.008$

But also the performance of the KGA is increased due to Lamarckism. Although no better results are found the KGA converges significantly, faster especially with increasing cardinality constraints, compare Fig. 8 to Fig. 5.

Unfortunately, all the GA representations perform so well on this problem instance with $K < N$ that no clear distinctions can be made. Only for $K = N$ the real-valued GA converges slower than the bit-string based GA, see Fig. 7, but outperforms both bit-string based standard GAs regarding the quality of the results, see Fig. 6. But these differences also vanish with the application of the hybrid KGA.

3.2 Results with Additional Constraints

With additional real-world constraints the previously continuous portfolio selection problem becomes a discrete one. Therefore, we extend the group of examined representations with an additional discrete representation using a bit-string limited to 7bit instead of 32bit.

To increase comprehensibility we examine the results separately, first for the standard GA and then for the hybrid KGA representation.

Standard GA without Lamarckism. Here the very same effect as in sec. 3.1 occurs: the standard GA implementation suffers from premature convergence, see Fig. 9 and Fig. 10. Again this is due to the neutrality of the search space caused by the repair mechanism. But now it applies to all problem instances since even without cardinality constraints the additional buy-in threshold acts like a cardinality constraint of $K = 12$. The neutral search space causes the GA

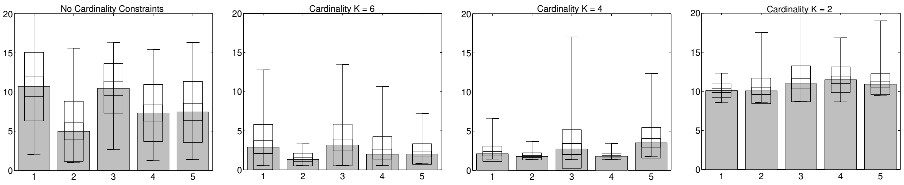

Fig. 11. Δ_{area} for the experiments on the Hang Seng data set with Lamarckism, $l_i = 0.08$ and $c_i = 0.008$ (1: GA 32bit binary-coding, 2: GA 7bit binary-coding, 3: GA 32bit gray-coding, 4: GA 7bit gray-coding, 5: GA real-valued)

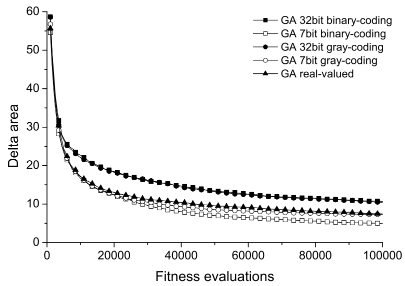

Fig. 12. Δ_{area} on the Hang Seng data set with Lamarckism, $K = N$, $l_i = 0.08$ and $c_i = 0.008$

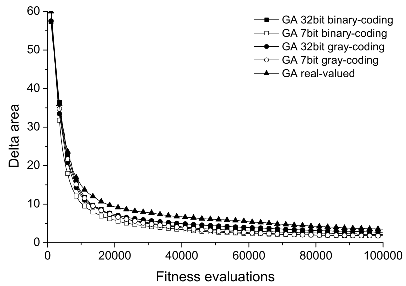

Fig. 13. Δ_{area} on the Hang Seng data set with Lamarckism, $K = 4$, $l_i = 0.08$ and $c_i = 0.008$

to search a subspace of the true search space and again the subspace consists only of portfolios of size K with weights $w_i \approx 1/K$.

Fig. 10 shows the convergence behavior on each problem instance. On each problem instance a bit-string based representation is compared to the real-valued representation. Basically, they all converge to the very same local optimum in the previously described subspace, but the real-valued representation performs slightly worse than the bit-string based representations.

Standard GA with Lamarckism. Again with Lamarckism the negative effect of the neutral search space is removed, see Fig. 11. And again the standard GA becomes much more efficient, since it is able to search the space of sparse portfolios more efficiently. The convergence speed of the standard GA once more matches the behavior of the hybrid KGA in the previous examples, see Fig. 12 and Fig. 13.

Regarding the different coding schemes the real-valued coding performs slightly better than the 32bit codings. But comparing the 7bit coding to the 32bit coding, the 7bit coding performs much better than the 32bit coding and also outperforms the real-valued representation, see also Fig. 12 and Fig. 13. Most likely this is due to the reduced search space of the 7bit coding and the greater impact of the mutation operator. While the confidence intervals for the different representations are clearly separated for $K = N$ and $K = 4$, the differences decrease with increasing cardinality constraints.

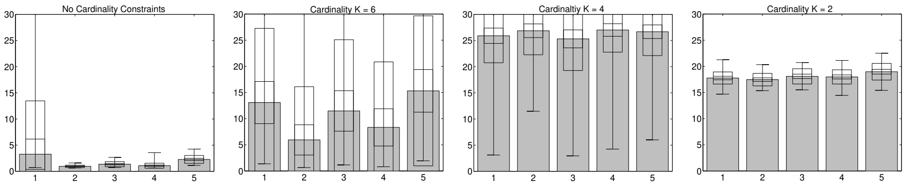

Fig. 14. Δ_{area} for the experiments on the Hang Seng data set with $l_i = 0.08$ and $c_i = 0.008$ (1: KGA 32bit binary-coding, 2: KGA 7bit binary-coding, 3: KGA 32bit gray-coding, 4: KGA 7bit gray-coding, 5: KGA real-valued)

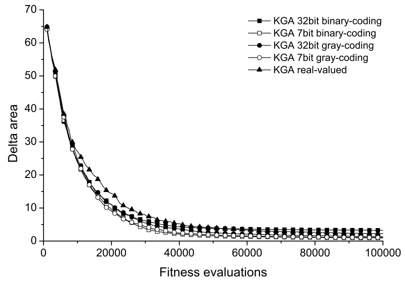

Fig. 15. Δ_{area} on the Hang Seng data set with $K = N$, $l_i = 0.08$ and $c_i = 0.008$

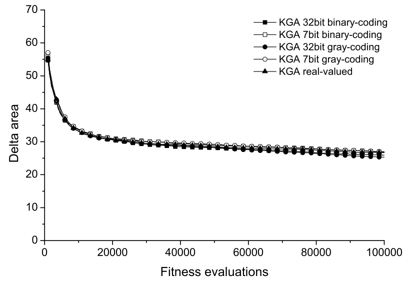

Fig. 16. Δ_{area} on the Hang Seng data set with $K = 4$, $l_i = 0.08$ and $c_i = 0.008$

Hybrid KGA without Lamarckism. Even without Lamarckism the hybrid KGA is not prone to the same premature convergence as the standard GA, compare Fig. 14 to Fig. 11. But while without cardinality constraints the hybrid KGA performs rather well, see Fig. 15, this is not the case with increasing cardinality constraints, see Fig. 16. Although the mean results of the hybrid KGA are still better than the results of the standard GA and the best runs of the hybrid KGA are considerably better, the overall results of the KGA without Lamarckism can be rejected as being too bad and also too unreliable.

When comparing the different coding schemes, again the real-valued KGA performs worst on all problem instances, see Fig. 14. The 32bit codings usually perform slightly better than the real-valued representation, except for some extreme outliers in case of the 32bit binary-coding for $K = N$. But the confidence intervals between the 32bit codings and the real-valued coding are not as clearly separated. But the confidence intervals for 32bit and 7bit coding are clearly separated at least for weak cardinality constraints, $K = N$ and $K = 6$, and show that the 7bit coding outperforms the 32bit coding. With increasing cardinality constraints these differences are again leveled out. Regarding the comparison between binary and gray-coding no reliable conclusions can be made, since the confidence intervals have a significant overlap.

Hybrid KGA with Lamarckism. The application of Lamarckism gives the driving edge to the hybrid KGA, see Fig. 17. In some instances the results are

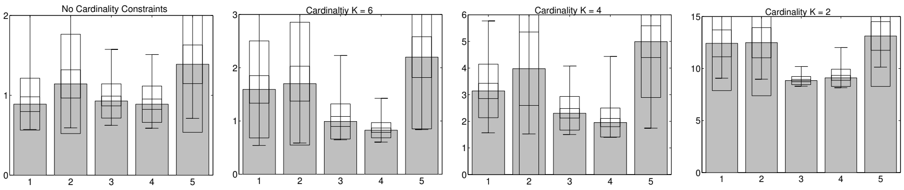

Fig. 17. Δ_{area} for the experiments on the Hang Seng data set with Lamarckism, $l_i = 0.08$ and $c_i = 0.008$ (1: KGA 32bit binary-coding, 2: KGA 7bit binary-coding, 3: KGA 32bit gray-coding, 4: KGA 7bit gray-coding, 5: KGA real-valued)

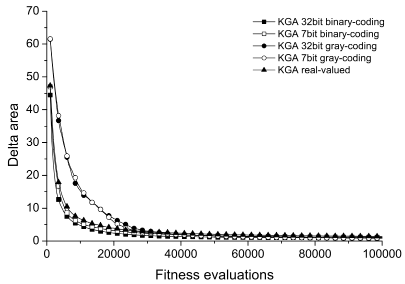

Fig. 18. Δ_{area} on the Hang Seng data set with Lamarckism, $K = N$, $l_i = 0.08$ and $c_i = 0.008$

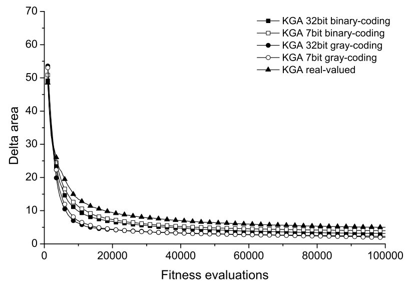

Fig. 19. Δ_{area} on the Hang Seng data set with Lamarckism, $K = 4$, $l_i = 0.08$ and $c_i = 0.008$

so good, that we believe the fixed size of the archive population may become a limiting element.

Comparing the different representations the real-valued representation performs again worst. Second best is the binary-coding, but astonishingly the previously observed advantage of the 7bit coding is reversed in this case. Gray-coding on the other hand performs best on all problem instances and again the confidence intervals indicate a significant advantage of the 7bit gray-coding over the 32bit gray-coding. In this case the general advantage of the gray-coding is even maintained for increasing cardinality constraints and actually becomes more and more obvious.

Regarding the speed of convergence the gray-coding is slightly slower in the beginning for $K = N$, see Fig. 18, but it catches up and finally produces the best results. With increasing cardinality constraints the gray-coding performs better and outperforms the other coding schemes regarding speed of convergence and the quality of the final result, see Fig. 19 and Fig. 17.

4 Discussion

There are several conclusions that can be drawn from the experimental results presented in this paper. First, we were able to prove that the proposed hybrid KGA representation performed better than the standard GA on this problem class regardless of the problem instance and the genotype representation used for **W**. The KGA produced better results and converged faster than the standard GA. We could support the argument, that the advantage of the hybrid KGA is based on the efficient removal of surplus assets, by reproducing the very same effect for the standard GA on the problem instances without additional constraints, with cardinality constraints and Lamarckism. Although the positive effect of Lamarckism on the standard GA was not as strong if real-world constraints were added.

We also showed that the standard GA without Lamarckism is prone to premature convergence, since the neutrality of the search space causes the GA to get stuck in an suboptimal subspace. The KGA on the other hand was not prone to such premature convergence even without Lamarckism.

Regarding the different coding schemes we were able to show that on average the real-valued coding performed worst on all problem instances. But there were only negligible differences between the binary and the gray-coding if no additional constraints were applied. We could also prove that with additional constraints the 'discrete' 7bit coding performed better than the 32bit coding on both bit-string based codings, most likely because the mutation and crossover operators become more effective.

Overall, the hybrid KGA with 7bit gray-coding and Lamarckism turned out to be best on the most interesting problem instances with additional real-world constraints.

5 Future Work

Our future work will concentrate on evaluating the performance of alternative MOEA implementations on the portfolio selection problem. We believe that the choice of the MOEA strategy will become crucial, if more real-world constraints are added like sector/industry constraints, immunization/duration matching and taxation constraints, which may increase the output dimension of the portfolio selection problem.

Another area of improvement could be the application of more sophisticated local search heuristics. There are numerous alternatives to the simple search for feasible solutions, but they have to be carefully evaluated regarding their ability to handle real-world constraints.

Finally, we plan to extend our experiments to other models for portfolio selection like the Black-Litterman model [3], since the Markowitz mean-variance model suffers from two major drawbacks: first, it is rather complicated to gather the necessary data and estimate μ_i and σ_{ij} from historic data and secondly, the Markowitz model is very sensitive to estimation errors of μ_i and σ_{ij}.

References

1. S. Arnone, A. Loraschi, and A. Tettamanzi. A genetic approach to portfolio selection. *Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems*, 3:597–604, 1993.

2. J. B. Beasley. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research*, 8:429–433, 1996.
3. F. Black and R. Litterman. Global portfolio optimization. *Financial Analysts Journal*, pages 28–43, September-October 1992.
4. T.-J. Chang, N. Meade, J. B. Beasley, and Y. Sharaiha. Heuristics for cardinality constrained portfolio optimization. *Computers and Operations Research*, 27:1271–1302, 2000.
5. Y. Crama and M. Schyns. Simulated annealing for complex portfolio selection problems. Working paper GEMME 9911, Universit de Lige, 1999.
6. K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
7. D. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms*, pages 41–49, 1987.
8. J. Knowles and D. Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 98–105, Mayflower Hotel, Washington D.C., USA, 1999. IEEE Press.
9. A. Loraschi and A. Tettamanzi. An evolutionary algorithm for portfolio selection in a downside risk framework. *Working Papers in Financial Economics*, 6:8–12, June 1995.
10. A. Loraschi, A. Tettamanzi, M. Tomassini, and P. Verda. Distributed genetic algorithms with an application to portfolio selection problems. In D. W. Pearson, N. C. Steele, and R. F. Albrecht, editors, *Artificial Neural Networks and Genetic Algorithms*, pages 384–387, Wien, 1995. Sringer.
11. H. M. Markowitz. Portfolio selection. *Journal of Finance*, 1(7):77–91, 1952.
12. H. M. Markowitz. *Portfolio Selection: efficient diversification of investments*. John Wiley & Sons, 1959.
13. N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
14. F. Streichert, H. Ulmer, and A. Zell. Evolutionary algorithms and the cardinality constrained portfolio selection problem. In D. Ahr, R. Fahrion, M. Oswald, and G. Reinelt, editors, *Operations Research Proceedings 2003, Selected Papers of the International Conference on Operations Research (OR 2003), Heidelberg, September 3-5, 2003*. Springer, 2003.
15. D. L. Whitley, V. S. Gordon, and K. E. Mathias. Lamarckian evolution, the baldwin effect and function optimization. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature – PPSN III*, pages 6–15, Berlin, 1994. Springer.
16. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.
17. E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.