Hybrid Representation for Compositional Optimization and Parallelizing MOEAs

Felix Streichert, Holger Ulmer, and Andreas Zell

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

Abstract. In many real-world optimization problems sparse solution vectors are often preferred. Unfortunately, evolutionary algorithms can have problems to eliminate certain components completely especially in multi-modal or neutral search spaces. A simple extension of the real-valued representation enables evolutionary algorithms to solve these types of optimization problems more efficiently. In case of multi-objective optimization some of these compositional optimization problems show most peculiar structures of the Pareto front. Here, the Pareto front is often non-convex and consists of multiple local segments. This feature invites parallelization based on the 'divide and conquer' principle, since subdivision into multiple local multi-objective optimization problems seems to be feasible. Therefore, we introduce a new parallelization scheme for multi-objective evolutionary algorithms based on clustering.

Keywords: Multi-objective Evolutionary Algorithms (MOEAs); Solution Representation; Constrained Portfolio Selection Problem; Parallelizing MOEAs.

1 Introduction

In many real-world optimization problems sparse solution vectors are preferred over solution vectors where all the decision variables are non-zero. One example is given by compositional optimization of a recipe for a medical drug which requires to optimize the percentage of several hundreds of potential ingredients. Any pharmacist would prefer a simple recipe made of only few ingredients over a recipe made from all available ingredients for practical reasons. Another example is the inference of regulatory networks. Here, the problem is given by finding suitable parameters defining the dynamics of a regulatory network to fit a measured time series. In case the network structure is a priori unknown, the number of parameters to optimize is given by a fully connected network, but any practitioner knows that regulatory networks, especially biological ones, are often sparsely connected. Therefore, again sparse solution vectors are preferred. A final example, which will be used to illustrate our results, is the multi-objective portfolio selection problem. The portfolio selection problem is given by the task of how to invest a limited amount of money in multiple available assets. This

Dagstuhl Seminar Proceedings 04461
Practical Approaches to Multi-Objective Optimization
http://drops.dagstuhl.de/opus/volltexte/2005/251

problem is explained in detail in sec. 2. This problem also calls for sparse solution vectors. On the one hand, because many solutions on the Pareto front consist only of a limited subset of all available assets and on the other hand, an investor many be interested to limit the overall number of assets he is investing in. We introduce an alternative hybrid encoding for evolutionary algorithms (EAs), which combines a standard real-valued vector with an additional binary vector and which is able to outperform the standard encoding on compositional optimization problems.
These kind of optimization problems have a number of intriguing features. Each subset of non-zero parameters can be considered as a subspace of the true solution space and each of these subspaces has its own local optimum. Therefore, the search space is highly multi-modal. In case of multi-objective optimization this leads to multiple localized Pareto fronts. This feature invites parallelization based on the 'divide and conquer' principle, since subdivision into multiple local multi-objective optimization problems seems to be feasible. Unfortunately, without any a priori knowledge of the shape of the true Pareto front it is difficult to decide which subdivision scheme to use. Therefore, we suggest to use an adaptive subdivision scheme based on repeated clustering to identify suitable subdivisions for a 'divide and conquer' approach.
This paper is structured as follows, in sec. 2 we introduce the constrained portfolio selection problem. In sec. 3 we show that a hybrid encoding outperforms the standard encoding on the portfolio selection problem even in case the standard encoding is enhanced with a repair mechanism together with Lamarckism. Then, we present our results on parallelizing multi-objective evolutionary algorithms (MOEAs) using a clustering based parallelization scheme in sec. 4. Finally, we discuss the results in sec. 5.

2 The Portfolio Selection Problem

There are numerous optimization problems in the area of financial engineering for example index tracking, credit scoring, identifying default rules, time series prediction, trading rules, etc. But one of the most prominent is the portfolio selection problem. The portfolio selection problem is given by the task of how to distribute a limited amount of money between multiple assets available for a profitable investment strategy.
Markowitz made an early approach to give the portfolio selection problem a mathematical background, the Markowitz mean-variance model [10, 11]. This model assumes that an investor would always try to maximize the return of his investments while at the same time securing his investments from a possible loss. Therefore, the portfolio problem gives a multi-objective optimization problem, maximizing the expected return on the one hand and on the other hand minimizing the risk (variance) of the portfolio.
The Markowitz mean-variance model 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:

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

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

$$\text{subject to} \quad \sum_{i=1}^{N} w_i = 1 \quad (3)$$

$$\text{and} \quad 0 \leq w_i \leq 1 \quad (4)$$

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 historical 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 maximum 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. Since the mentioned constraints are hard constraints a repair mechanism is used all the time regardless of the representation used.
The portfolio selection problem will be used throughout this paper as an exemplary compositional optimization problem and multi-objective optimization problem for parallelization. The features and characteristics of the portfolio selection problem inspired both the hybrid representation used in sec. 3 and the clustering based parallelization scheme in sec. 4.

3 Hybrid Representation for Compositional Optimization

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. 16. For $K = 2$ there are several distinct regimes of two assets combinations that form the Pareto front. The same holds true for larger values of K. But the less restrictive the cardinality constraints are, the less distinct the regimes.
The problem to find the best combinations of assets for a portfolio resembles a one-dimensional binary knapsack problem. This kind of problem has already

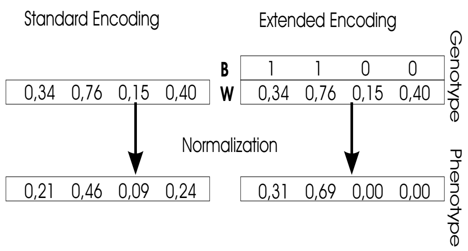

Fig. 1. Comparing the standard encoding to the hybrid encoding.

been addressed by means of EA using a binary genotype. We suggest to use the very same genotype in addition to the vector of decision variables **W**, see Fig. 1. Each bit of the bit-string **B** determines whether the associated asset will be 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 will be processed by the repair mechanism. With this hybrid encoding it is much easier for the EA to add or remove the associated asset simply by mutating the bit-string **B**.
The hybrid representation is altered by mutating/crossing each genotype **B** and **W** separately from each other. Binary operators are acting on the binary vector **B** and real-valued operators are acting on the real-valued decision vector **W**.
Alternatively, we discuss the impact of a repair mechanism, which is able to resolve the cardinality constraint in a deterministic way, with and without Lamarckism. With Lamarckism the EA with the standard encoding should also be able to explore sparse decision vectors **W** as efficiently as the hybrid encoding, since it is also limited to a sparse decision vector. Unfortunately, this strategy has two drawbacks. First, it is not as flexible as the hybrid encoding as it will be shown in sec. 3.3. And second, the repair mechanism is limited to optimization problem where the cardinality is a hard constraint. In case the cardinality of the decision vector is an optional goal as in case of inferring regulatory networks of unknown network topology, this strategy cannot be applied anymore.
A general comparison between the standard encoding and the hybrid encoding has been performed in [15]. A more detailed discussion on this encoding style and the general parameters that were used to obtain these results can be found in [16].

3.1 Experimental Results

The comparison of the different EA representation was performed on data sets given by Beasley [1] available at *http://mscmga.ms.ic.ac.uk/info.html*. 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 EA encodings and crossover operators. First, we compare the portfolio selection problem without cardinality constraints

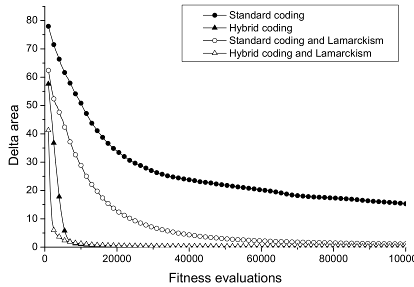

Fig. 2. Comparing the standard and the hybrid encoding with and without Lamarckism without cardinality constraints.

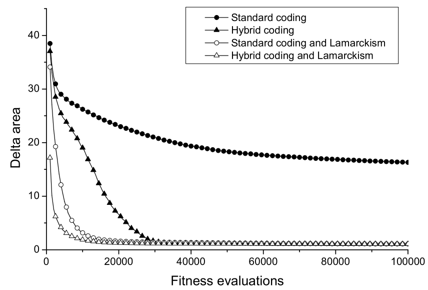

Fig. 3. Comparing the standard and the hybrid encoding with and without Lamarckism with $k = 4$.

and with cardinality constraints $k = 4$. In a second set of experiments we also add buy-in thresholds ($l_i = 0.1$) and roundlots constraints ($c_i = 0.02$) to the portfolio selection problem.

We measure the performance of the algorithms by calculating the S-metric [20], i.e. the area under the currently achieved Pareto front bounded by μ_{max} of the maximum return asset and $\sigma = 0$. We compare this area to the area under the Pareto front of the unconstrained portfolio selection problem calculated through quadratic programming also given in the benchmark data set. The percentage difference (Δ_{area}) of the MOEA calculated solution and the reference solution is to be minimized and gives the measure of quality. But only without any real-world constraints can this measure drop to zero, otherwise the Δ_{area} is limited by the constraints. Additionally, the Δ_{area} is limited due to the limited size of the archive population, which gives the Pareto front identified by the MOEA.

To obtain reliable results we repeat each MOEA experiment for 50 times for each parameter setting and problem instance. A single MOEA run is terminated after 100,000 fitness evaluations.

3.2 Results without Additional Constraints

Without Lamarckism the hybrid encoding clearly outperforms the standard encoding regardless of the cardinality constraint imposed, see Fig. 2 and 3. This is due to the fact that even without cardinality constraints the Pareto front contains portfolios consisting of only few available assets. But the standard encoding has difficulties to remove surplus assets and to obtain a sparse decision vector. Halfway this is caused by the effect of the repair mechanism used, which makes the search space neutral to some extend. Only the hybrid encoding is able to counter the effect of the neutral search space especially in case of $k = 4$.

With Lamarckism on the other hand the both representations perform significantly better. The standard encoding recovers from the negative effect of the repair mechanism, since the problem with the neutral search space is partially resolved by Lamarckism. Even without cardinality constraints the MOEA with

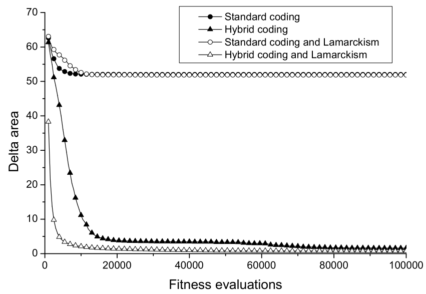

Fig. 4. Comparing the standard and the hybrid encoding with and without Lamarckism without cardinality constraints but with additional constraints.

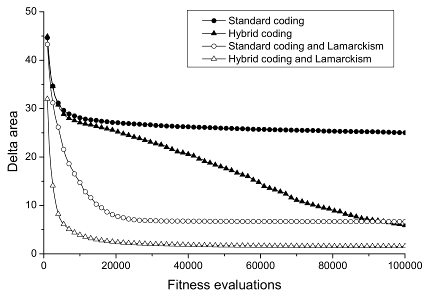

Fig. 5. Comparing the standard and the hybrid encoding with and without Lamarckism with $k = 4$ and additional constraints.

standard encoding is able to remove surplus assets more efficiently than before, because of constraint 3 and 4. With cardinality constraints the decision vector becomes as sparse as in case of the hybrid encoding. Therefore, the standard encoding with Lamarckism is able to equal the performance of the hybrid encoding. But still the hybrid encoding with Lamarckism performs significantly better than the standard encoding in both cases, because Lamarckism removes some of the neutrality of the search space for the hybrid encoding too and the cardinality of the bit vector is set to reasonably values immediately.

3.3 Results with Additional Constraints

With additional buy-in thresholds and roundlot constraints the repair mechanism again increases the neutrality of the search space. Especially the standard encoding suffers form this effect. Without Lamarckism the standard encoding finds only portfolios where the weights of the assets are $w_i \approx 1/k$, see Fig. 5. This also applies to the problem instance without cardinality constraints since the buy-in threshold imposes an implicit cardinality constraint of $k = 10$, see Fig. 4. With Lamarckism the neutrality of the search space is a little bit removed such that the MOEA with the standard encoding is able to explore portfolio with $w_i < 1/k$ but still the neutral search space prevents to remove surplus assets by keeping $w_i > l_i$. This is the reason why there is virtually no difference for the standard encoding with and without Lamarckism on the problem instance without explicit cardinality constraints, see Fig. 4. But while in case of $k = 4$ the standard encoding with Lamarckism is able to find better portfolios, it is still limited to a subspace of the true search space and converges prematurely, see Fig. 5.

The hybrid encoding is again able to counteract the negative effect of the neutral search space and to search efficiently for space decision vectors. And again the hybrid encoding performs better with Lamarckism than without, because it removes some of neutrality of the search space, see Fig. 4 and 5.

3.4 Exemplary Pareto fronts for the Portfolio Selection Problem

Fig. 16 gives exemplary results on the Hang Seng data set with 31 assets and the DAX data set with 85 assets, respectively. The obtained Pareto fronts illustrate the multiple local segments that make up the complete efficiency frontier for the cardinality constrained portfolio selection problem. Such structures for $k > 2$ could not have concluded from the results presented by Beasley et al. [4], although they pooled results from multiple optimization algorithms like Tabu Search, Simulated Annealing and Genetic Algorithms. Similar structures also occur in other optimization problems, since cardinality constraints may cause the search space to become highly multi-modal.
This observation caused us to believe that a 'divide and conquer' approach might be feasible on such problem instances where the search space is structured and can be subdivided into subspaces. Unfortunately, without a priori knowledge such structures can only be identified on the fly during the exploration of the search space. In a previous publication we suggested the use of clustering algorithms to identify and maintain niches in multi-modal search spaces [14]. It is most straightforward to use the same method to identify structure in multi-objective search spaces required for parallelization based on the 'divide and conquer' principle.

4 Utilizing the 'Divide and Conquer' Approach to parallelize MOEAs

A good overview over alternative parallelization strategies for MOEAs is given in [18]. Basically, three different approaches to parallelize MOEAs can be distinguished: the island model, the master-slave model and the diffusion model. Since we want to use the 'divide and conquer' approach only the island model is worth considering.
The most straightforward implementation of island MOEAs runs a number of MOEA populations independently, each trying to obtain the complete Pareto front and every m_{rate} generations migration takes place [13, 9]. Miki et al. also applied an island MOEA, but at regular intervals, the subpopulations are aggregated, sorted according to a randomly selected objective, and then redistributed onto different processors [12]. This approach allows the subpopulation to specialize on given objectives during the optimization, and was also used in [5, 6].
Another approach by Deb et al. uses the dominance principle [2] to guide the individual subpopulation to different sections of the global Pareto front [8]. Unfortunately, the individual search directions have to be set beforehand, which requires a priori knowledge of the shape of the Pareto front, and moreover this approach cannot be applied to concave Pareto fronts.
The cone separation technique uses a geometrical approach to subdivide a given Pareto front [3], see Fig. 6 for an example. A reference point R is given by the extreme values of the current Pareto front and R is the origin of k subdividing demarcation lines. The authors point out, that in order to have each subpopulation

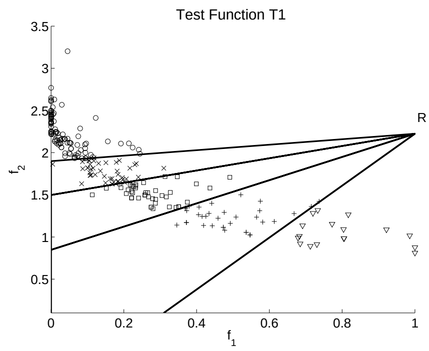

Fig. 6. Exemplary partitioning using the cone separation approach [3].

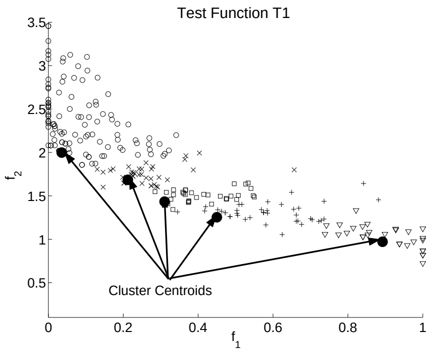

Fig. 7. Exemplary partitioning using k-Means ($k = 5$) on the Pareto front.

focusing on a specific region in objective space, the demarcation lines for each region have to be treated as zone constraints using the constrained dominance principle [7]. Again, this approach has several drawbacks. In case of discontinuous or non evenly distributed Pareto fronts small or empty subpopulations can be generated, which do not reflect any problem inherent structures. And finally, the geometrical subdivision scheme of cone separation becomes rather complicated in case of more than two objectives.

4.1 The Clustering Based Parallelization Scheme

Instead of choosing a subdivision scheme a priori using the dominance principle or a static geometrical approach, we decided to search for a suitable subdivision scheme on the fly by means of clustering algorithms. Each m_{rate} generations, all subpopulations $P_{i,remote}$ are gathered, the aggregated Pareto front P_{local} is clustered and all individuals are redistributed onto the available processors depending on the cluster centroids. For clustering, we decided to use k-Means clustering on the current Pareto front, because k-Means allows use to choose the number of clusters according to the number of available processors k=k (k does not correspond to the cardinality constraint of the previous section). In case the size of P_{local} is smaller than k, next level Pareto fronts are also used for clustering. We further distinguish between two variants for clustering, first a search space based clustering and second an objective space clustering. Fig. 7 gives an impression of an objective space based clustering.

One advantage of the clustering based parallelization scheme is that each subpopulation is guaranteed to be non-empty. Depending on the shape of the Pareto front this cannot be guaranteed for the cone separation approach. Since we used cone separation as a reference algorithm we decided to assign random individuals to an empty subpopulation to prevent any further complications.

To limit subpopulations to their specific region, we implemented zone constraints based on the constrained dominance principle [7] using the cluster centroids. In

```
g = 0;
for (i = 0; i < k; i++) do  P_{i,remote}.initialize();
foreach P_{i,remote} do P_{i,remote}.evaluate();
while isNotTerminated() do
    foreach P_{i,remote} do
        P_{i,remote}.evolveOneGeneration();
        P_{i,remote}.evaluate();
    end
    if (g%m_{rate} == 0) then
        /*Migration and/or partitioning scheme */
        P_{local}.initialize();
        foreach P_{i,remote} do P_{local}.addPopulation(P_{i,remote});
        foreach P_{local}.cluster(k) do P_{i,remote} = P_{local}.cluster(k).getCluster(i);
        if useConstraints then foreach P_{i,remote} do P_{i,remote}.addConstraints();
    end
    g = g +1;
end
```

Algorithm 1: General scheme of the clustering based parallelization scheme for MOEA, with k number of processors used, m_{rate} the migration rate, $P_{i,remote}$ a remote population and P_{local} the local population.

case an individual is assigned to a different cluster centroid than the current subpopulation belongs to, the individual is marked as invalid. This interpretation complies with the implementation of cone separation, but we also tested the parallel MOEAs without zone constraints.

4.2 Experimental Results

We compare the new clustering based parallelization scheme with both objective space based k-Means (kosMOEA) and search space based k-Means (kssMOEA) on four different test function, see appendix, to three other approaches. First, an island model MOEA implementation without migration (pMOEA). Secondly, an island model MOEA with migration (iMOEA) where the subpopulations can profit from each others achievements. And finally, the cone separation MOEA (csMOEA). We further compare the pMOEA and the iMOEA to both the cluster based and the cone separation MOEAs with zone constraints and without zone constraint.
We compare the different parallel MOEA implementations on four test functions, three of them given from literature (T1-T3) and one simplified portfolio selection problem (T4) with five available asset, a cardinality constraint of 2 and 25 additional decision variable to make the problem as complex as the previous benchmark functions. T4 has similar properties as the previously discussed portfolio selection problem but has only 4 local Pareto fronts.
To see how each implementation scales with increased number of processors, we use up to six processors on each test function. To allow comparison we decrease the size of the subpopulations $P_{i,remote}$ from 600 to 100 with increased number

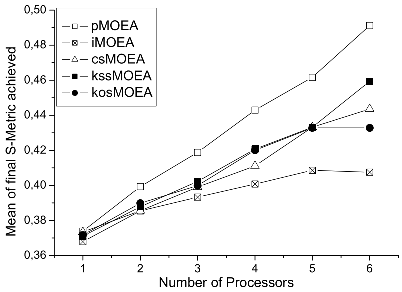

Fig. 8. 'Divide and conquer' approaches with zone constraints on T1.

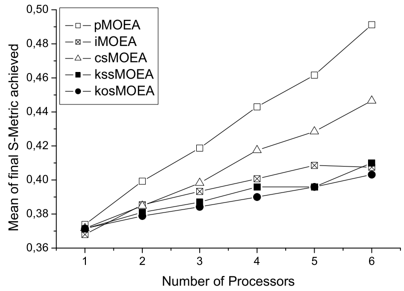

Fig. 9. 'Divide and conquer' approaches without zone constraints on T1.

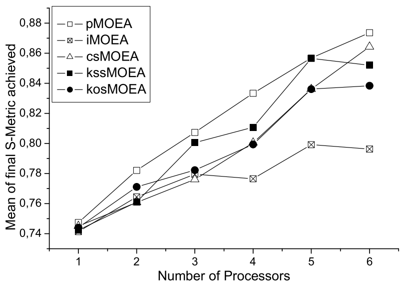

Fig. 10. 'Divide and conquer' approaches with zone constraints on T2.

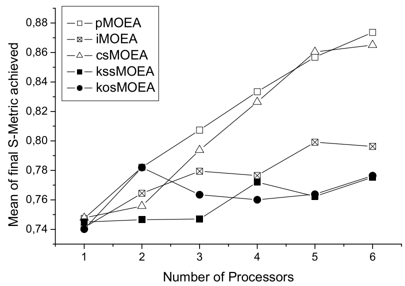

Fig. 11. 'Divide and conquer' approaches without zone constraints on T2.

of processors and use a real-valued NSGA-II with an archive size of $P_{i,remote}/2$, details can be found in [17].

For comparison, we use the hyper-volume under the accumulated population P_{local} (S-metric) of each parallel MOEA implementation averaged over 25 multi-runs for each problem instance and processor configuration. The S-metric is to be minimized, and typically converges to nonzero values.

Unfortunately, with zone constraints the 'divide and conquer' approaches hardly outperform the worst case implementation of the pMOEA on the T1 and T2 benchmark functions , see Fig. 10 and 8. This is rather discouraging, but can be explained by the structure of the T1-T3 benchmark function. A closer look at the T1-T3 test functions reveals that the Pareto fronts are not only contiguous in objective space, but also in search space. The Pareto-optimal solutions consist of vectors where $x_{i \neq 1} = 0$. A single solution on the true Pareto front could explore the whole Pareto front simply by mutating x_1.

Therefore, 'divide and conquer' cannot be applied successfully to these kind of problems. Instead of isolating subpopulation to solve supposed subproblem a holistic approach or heavy communication is necessary for a single good solution to explore the whole Pareto front. And in fact single processor approach with maximum population size performs best and the iMOEA with heavy communi-

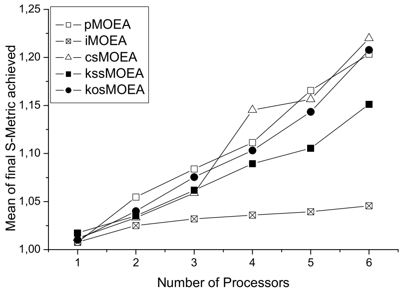

Fig. 12. ‘Divide and conquer’ approaches with zone constraints on T3.

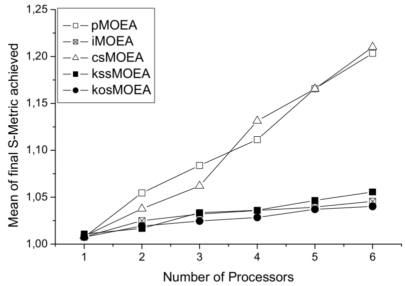

Fig. 13. ‘Divide and conquer’ approaches without zone constraints on T3.

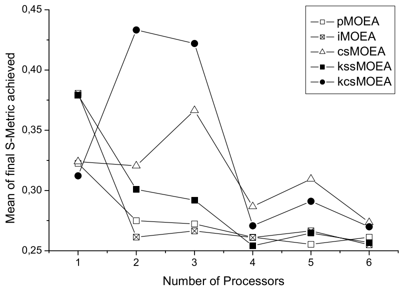

Fig. 14. ‘Divide and conquer’ approaches with zone constraints on T4.

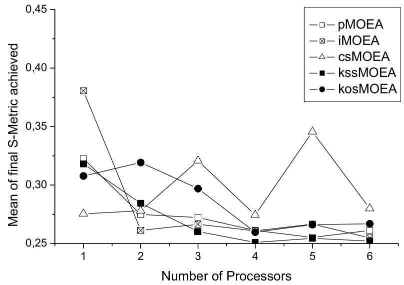

Fig. 15. ‘Divide and conquer’ approaches without zone constraints on T4.

cation is second best. Alternatively, we decided to remove the zone constraints to enable lateral exploration of the Pareto front. Indeed the kosMOEA and the kssMOEA perform significantly better without zone constraints, see Fig. 9 and 11. The csMOEA does not perform as well, because the non deterministic clustering algorithm causes additional lateral information exchange for the kssMOEA and kosMOEA compared to the static scheme of the csMOEA.

Both the T3 and the T4 on the other hand have local segments of the Pareto front. But the T3 is contiguous in search space. Again only without zone constraints the kssMOEA and the kosMOEA are able to equal the performance of the iMOEA on the T3 function, see Fig. 12 and 13. It is interesting to note that the kosMOEA slightly outperforms the kssMOEA without zone constraints. This can be accounted to the fact that the local segments of the Pareto front of the T3 benchmark function can only be separated in objective space and not in search space.

The T4 function is of course more qualified for a ’divide and conquer’ approach since the additional decision parameters are localized for each segment of the Pareto front and the resulting search space is truly multi-modal, which is also reflected in a higher noise rate in the results and the fact that multi processor runs outperform the single processor reference. Again the kosMOEA and the

kssMOEA perform significantly better without zone constraints and they seem to perform better in case the number of processors exceeds the number of local Pareto fronts, see Fig. 14 and 15. It is again interesting to note that on this problem instance the kssMOEA slightly outperforms the kosMOEA, which again corresponds to the fact that the local Pareto fronts are easier to distinguish in search space on T4 in contrast to the T3 benchmark function.
Basically, this shows that standard benchmark functions are not suited for 'divide and conquer' approaches for parallelization, but for master slave model or simple island models with heavy communication instead. We have shown that 'divide and conquer' approaches require certain internal structures in the search space to allow efficient subdivision into small subproblems, which can then be solved independently.

5 Conclusions

We have shown that certain problems, which require sparse decision vectors either caused by constraints or by the type of optimization problem, benefit from a mixed representation combining a real-valued and binary vector. Such a hybrid encoding allows EAs to search more efficiently for sparse decision vectors. We have further shown that a similar effect can be achieved through repair mechanisms together with Lamarckism. Unfortunately, this requires hard cardinality constraints to allow repair mechanisms and the repair mechanisms may introduce neutrality to the search space, which again may limit the efficiency of the optimization process. We have shown that the hybrid encoding is able to overcome such neutral search spaces to a certain extend and can be applied even on problem instances where no explicit cardinality constraints are given.
Additionally, we have shown that 'divide and conquer' approaches seem not be suitable for arbitrary problem instances, but require a certain internal structure in search space to allow a successful subdivision into small subproblems, which can then be solved independently. We have given at least a proof of concept for the clustering based parallelization scheme and have shown its efficiency on two benchmark problems. But additional experiments are necessary to distinguish between problem instances where 'divide and conquer' approaches are infeasible and problem instances where they can be advantageous.

Acknowledgements: This research has been funded by ALTANA Pharma AG, Konstanz, Germany.

References

1. J. B. Beasley. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research*, 8:429–433, 1996.
2. J. Branke, T. Kauler, and H. Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32:499–507, 2001.

3. J. Branke, H. Schmeck, K. Deb, and R. S. Maheshwar. Parallelizing multi-objective evolutionary algorithms: Cone separation. In *Congress of Evolutionary Computation (CEC 2004)*, pages 1952–1957, Portland, Oregon, USA, 2004. IEEE Press.
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. F. de Toro Negro, J. Ortega, J. Fernandez, and A. Diaz. PSFGA: A parallel genetic algorithm for multi-objective optimization. In F. Vajda and N. Podhorszki, editors, *Euromicro Workshop on Parallel Distributed and Network-Based Processing*, pages 849–858. IEEE, 2002.
6. F. de Toro Negro, J. Ortega, E. Ros, S. Mota, B. Paechter, and J. Martin. PSFGA: Parallel processing and evolutionary computation for multi-objective optimization. *Parallel Computing*, 30:721–739, 2004.
7. 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.
8. K. Deb, P. Zope, and A. Jain. Distributed computing of pareto-optimal solutions with evolutionary algorithms. In C. Fonseca, P. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization*, volume 2632 of *LNCS*, pages 534–549. Springer-Verlag, 2003.
9. H. Horii, M. Miki, T. Koizumi, and N. Tsujiuchi. Asynchronous migration of island parallel GA for multi-objective optimization problems. In L. Wang, K. Tan, T. Furuhashi, J.-H. Kim, and X. Yao, editors, *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 86–90, Nanyang University, Singapore, 2002.
10. H. M. Markowitz. Portfolio selection. *Journal of Finance*, 1(7):77–91, 1952.
11. H. M. Markowitz. *Portfolio Selection: efficient diversification of investments*. John Wiley & Sons, 1959.
12. M. Miki, T. Hiroyasu, and S. Watanabe. The new model of parallel genetic algorithm in multiobjective genetic algorithms. In *Congress of Evolutionary Computation (CEC 2000)*, volume 1, pages 333–340, 2000.
13. D. Quagliarella and A. Vicini. Subpopulation policies for a parallel multi-objective genetic algorithm with applications to wing design. *IEEE International Conference on Systems, Man and Cybernetics*, 4:3142–3147, 1998.
14. F. Streichert, G. Stein, H. Ulmer, and A. Zell. A clustering based niching EA for multimodal search spaces. In *6th International Conference on Artificial Evolution*, volume 2723 of *LNCS*, pages 293–304, Marseille, 27-30 October 2003.
15. 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*, pages 253 – 260. Springer, 2003.
16. F. Streichert, H. Ulmer, and A. Zell. Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. In *Congress of Evolutionary Computation (CEC 2004)*, pages 932–939, Portland, Oregon, USA, 2004. IEEE Press.
17. F. Streichert, H. Ulmer, and A. Zell. Parallelization of multi-objective evolutionary algorithms using clustering algorithms. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, Guanajuato, Mxico, 2005.

18. D. A. Van Veldhuizen. Considerations in engineering parallel multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):144–173, 2003.
19. E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
20. 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.

Appendix: Test Functions

The test functions T1-T3 are from [19] and have the basic structure of:

$$\begin{aligned} f_1(\bar{x}) &= x_1 \\ f_2(\bar{x}) &= g(\bar{x})h\left(f_1(\bar{x}), g(\bar{x})\right) \end{aligned} \tag{5}$$

with $\bar{x} \in [0,1]^n$ and $n = 30$. They differ only in the definition of $g(x)$, $h(x)$.
Test Function T1 has a convex Pareto front:

$$\begin{aligned} g(\bar{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \tag{6}$$

Test Function T2 has a concave Pareto front:

$$\begin{aligned} g(\bar{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ h(f_1, g) &= 1 - (f_1/g)^2 \end{aligned} \tag{7}$$

Test Function T3 has a discontinuous Pareto front:

$$\begin{aligned} g(\bar{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\ h(f_1, g) &= 2 - \sqrt{f_1/g} - (f_1/g)sin(10\pi f_i) \end{aligned} \tag{8}$$

Test Function T4 resembles the constrained portfolio selection problem minimizing risk (f_1) and loss (f_2) of N assets. We limit to $N = 5$ assets such that the number of local Pareto fronts is well in the number of processors used. The remaining $n - N$ span the search space:

$$\begin{aligned} f_1(\bar{x}) &= \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij} \\ f_2(\bar{x}) &= \sum_{i=1}^{N} x_i \cdot \mu_i + \sum_{i=N+1}^{n} x_i^2 \cdot x_{i \bmod N} \end{aligned} \tag{9}$$

μ_i	σ_{ij}				
0	1.0	0	0.1	0	0.3
1.0	0	0	0	0	0
0.2	0.1	0	0.7	0.3	-0.1
0.5	0	0	0.3	0.5	0
0.7	0.3	0	-0.1	0	0.2

with $n = 30$, $N = 5$, $x_i \in [0,1]$, $\sum_{i=1}^{N} x_i = 1$ and $\sum_{i=1}^{N} |sign(x_i)| = 2$.

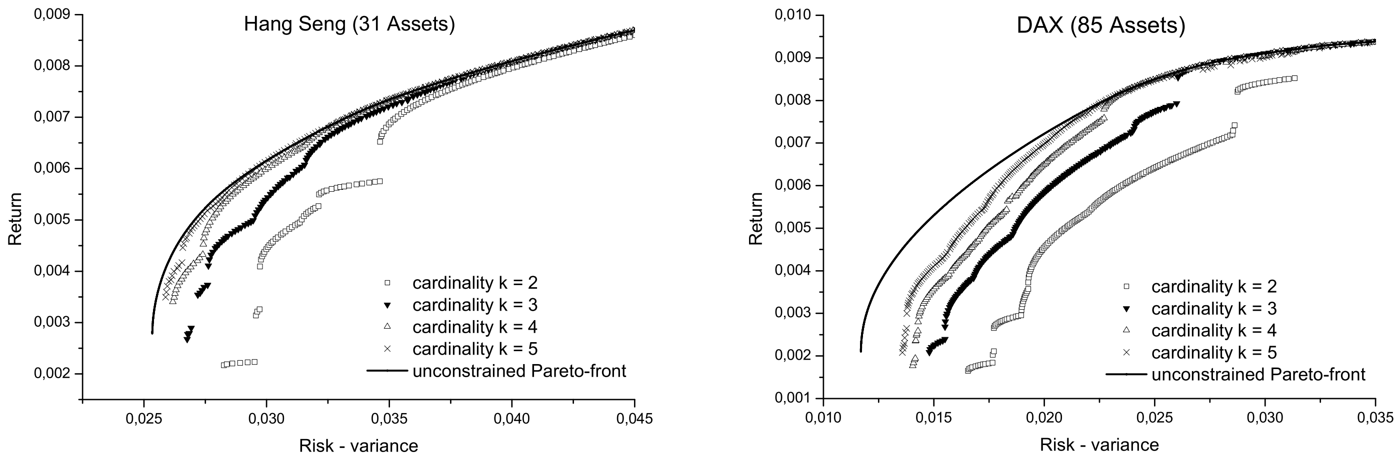

Fig. 16. Exemplary cardinality constrained Pareto fronts on the Hang Seng and the DAX data set.