HyGLEAM - An Approach to Generally Applicable Hybridization of Evolutionary Algorithms

Wilfried Jakob

Forschungszentrum Karlsruhe, Institute for Applied Computer Science,
P.O. Box 3640, 76021 Karlsruhe, Germany
jakob@iai.fzk.de

Abstract. Most successful applications of Evolutionary Algorithms to real world problems employ some sort of hybridization, thus speeding up the optimization process but turning the general applicable Evolutionary Algorithm into a problem-specific tool. This paper proposes to combine Evolutionary Algorithms and generally applicable local searchers to get the best of both approaches: A fast, but robust tool for global optimization. The approach consists of four different kinds of hybridization and combinations thereof, which are tested and compared using five commonly used benchmark functions and three real world applications. The results show the superiority of two hybridization types, with which reductions in the number evaluations of up to a factor of 100 could be achieved.

1 Motivation

When looking to the papers reporting about real world applications at the major EA conferences in the last ten years, it becomes clear that nearly all of them used some sort of hybridization with problem-specific local searchers or they applied other means of inserting problem-specific knowledge into the evolutionary process like e.g. special genetic operators. The commonly paid price for the achieved speed-up is the lost in generality of the resulting hybrid.

Already Holland suggested in 1975 to use GAs as a kind of preprocessor and finalize the optimization with local searchers [1]. 24 years later Goldberg and Voessner [2] pointed out that is not easy to find an appropriate procedure for distributing the computing time between global and local search and came to the conclusion that this is still an open question. They also stressed that nearly all serious EA applications use some sort of hybridization but that there is still a lack of investigations on suitable types of integration, which are not related to specific applications or limited to certain problem fields.

This paper contributes to this discussion by

- using only general applicable local searchers,
- comparing three well-known types of hybridization, a modified form of one type, and meaningful combinations of them
- introducing a new method of controlling the interaction of the basic algorithms, and
- testing the approach with a wide range of different types of applications to check the generality of the approach.

2 Methods of Hybridization

For a generally applicable hybridization of an EA, four general alternatives exist:

1. Pre-optimization of the start population
   This can be applied to the entire population or a fraction of it, and it provides the evolution with valid solutions of more or less good quality to start with.
2. Post-optimization of the EA results
   EAs are known to converge slowly. Thus, an improvement can be expected by stopping the evolution after approaching the area of attraction of the global optimum and leaving the rest to the local search.
3. Direct integration
   Optimizing every or the best offspring of one mating only causes the EA to operate over the peaks of the fitness landscape exclusively rather than to treat the valleys and slopes, too. The offspring's genotype can be updated (Lamarckian evolution) or left unchanged (Baldwinian evolution). As both methods which are usually applied to domain-specific local searchers are controversially discussed in literature [3, 4], this was also investigated. Orvosh and Davis recommend updating 5% of the accepted offsprings only [5].
4. Delayed direct integration
   Variant of direct integration, where the evolution works on its own until a criterion similar to the one used for switching from evolutionary to local search for post-optimization is fulfilled.

Pre-optimization can be combined with the other methods, while a fusion of direct integration and post-optimization does not appear to be meaningful.

3 Basic Algorithms Used for Hybridization

To comply with the goal of general applicability, the EA must allow for combinatorial optimization and parameter strings of dynamic length as required by some applications like design optimization [6] or collision-free robot path planning [7]. Especially because of the last requirement, GLEAM (General Learning Evolutionary Algorithm and Method) [7,8], an EA comprising elements of real coded GAs and the Evolution Strategy (ES) was chosen for testing, see also [9]. Among others, GLEAM uses mutation operators influenced by the ES in so far, as small parameter changes are more likely than greater ones. Mutation can also change the gene order and add or delete genes in the case of dynamic chromosomes. GLEAM uses ranking-based selection and elitist offspring acceptance. It is stressed that the introduced hybridization shall work with any other EA, too.

Suitable local search algorithms must be derivation-free and able to handle restrictions in order to preserve the general applicability of the resulting hybrid. Two well-known procedures from the sixties were chosen, since they meet these requirements and are known to be powerful local search procedures: The Rosenbrock algorithm [10] using one start point and the Complex method [11], because it can exploit multiple start points as they are delivered by an EA. The Rosenbrock procedure stops when the rate of changes of the main search direction decreases below a certain value and when the distances covered become too small. This is controlled by an external strategy parameter, here called *precision*. The Complex procedure stops when no improvement is achieved in five consecutive iterations. Schwefel gives a detailed description of both algorithms together with experimental results [12].

As this paper focuses on hybridization and due to the lack of space the basic algorithms have been described here very briefly only and the interested reader is referred to given literature.

4 Controlling the Basic Algorithms

Concerning real world problems neither the structure of the fitness landscape nor the optimum or its area of attraction are known in advance. But as computation of the fitness function frequently is time-consuming, it is possible to perform more sophisticated calculations to estimate when to switch from global to local search.

Fig. 1 shows the typical progress of an EA run. Stagnation phases of the overall quality can be identified easily, e.g. A, B or C. But which one shall be selected for terminating the evolution? This cannot be derived from stagnation only. A better measure is the genotypic diversity within the population. If the population consists of a few genotypically different sub-populations (niches) only, which are of minor difference, then stagnation can be expected, which provides little chance for greater progress. Hence, stagnation phases like in Fig. 1 may be used to trigger a check for niche constitution. Another trigger may be the number of generations without offspring acceptance.

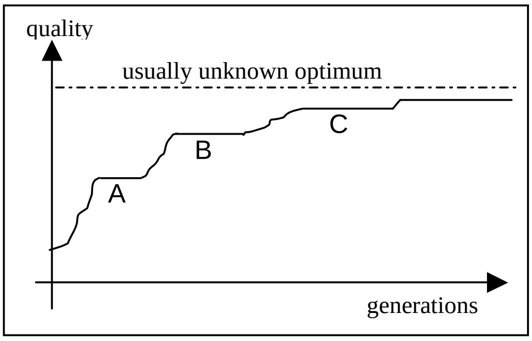

Fig. 1. Typical progress during the run of an EA

4.1 Distance Measures for Chromosomes

To estimate the genotypic diversity, distance measures for chromosomes must be defined, which quantify the parameter distance, the different gene ordering, and the common genes in the case of dynamic chromosomes. Distance functions reported in literature are often too specialized for the problem on hand, rather than to serve as a solution here [9]. Measures should be independent of the application in so far as they must not be influenced by actual parameter ranges or the number of genes and should be within a fixed range. The measures defined here vary between 0 (identity) and 1 (maximum difference). They comply with the four metric axioms, but the proofs and the calculation of $dist_{\max}$ are omitted here due to the lack of space. They can be found on the following web page: `http://www.iai.fzk.de/~jakob/hy_gleam/`

Fixed-length Chromosomes with Irrelevant Gene Order. For this chromosome type, the calculation of the parameter distance Δ_{par} of two chromosomes C_1 and C_2 is sufficient. It is defined as follows:

$$\Delta_{par}(C_1, C_2) = \frac{1}{n} \sum_{i=1}^{n} \frac{|par_{i,1} - par_{i,2}|}{ub_i - lb_i} \tag{1}$$

where n: number of all parameters of all genes
$par_{i,j}$: i-th parameter of chromosome j
lb_i, ub_i: lower and upper limits of the i-th parameter

Fixed-length Chromosomes with Relevant Gene Order. The positional difference $pd_{1,2}(G_i)$ of one gene G_i of two fixed-length chromosomes C_1 and C_2 is defined by their sequential numbering and the calculation of the absolute difference of their indices. This leads to the positional distance of two chromosomes Δ_{pos}:

$$\Delta_{pos}(C_1, C_2) = \frac{1}{dist_{\max}} \sum_{i=1}^{len} pd_{1,2}(G_i) \tag{2}$$

where len: length of the chromosomes ($len > 1$)
$dist_{\max}$: distance maximum of all genes within one chromosome

For $dist_{\max}$ two cases for odd and even chromosome lengths must be considered:

$$dist_{\max,\,even} = \frac{len^2}{2} \qquad dist_{\max,\,odd} = \frac{len^2 - 1}{2} \tag{3}$$

The overall distance Δ of fixed-length chromosomes with relevant ordering is defined as the mean of Δ_{par} and Δ_{pos}.

Variable-length Chromosomes. The goal is to determine the precise difference of similar chromosomes, while the exact value of discrepancy of dissimilar ones is of less interest. So the resulting measure may be inexact for more different chromosomes, thus yielding a less complex formula which reduces the computational effort for the fairly frequent distance calculations.

For two chromosomes C_1 and C_2 the set of common genes G_{com} may not be empty. Otherwise, the overall distance is set to 1. As genes in chromosomes of variable length may occur several times, they are treated in the sequence of their indexing and Δ_{par} and Δ_{pos} are defined over G_{com}, where $dist_{\max}$ is taken from the shorter chromosome. This may lead to an oversized value of Δ_{pos}, which may exceed 1 especially for chromosomes with large differences. Thus, Δ_{pos} is limited to 1 and the error is accepted, as it increases with the chromosome difference.

The difference of the presence of genes Δ_{gp} in two non-empty chromosomes is calculated as follows:

$$\Delta_{gp} = 1 - \frac{card(G_{com})}{\max(len(C_1), len(C_2))} \tag{4}$$

The overall distance Δ of variable-length chromosomes is defined as the mean of the three distances, with Δ_{gp} being weighted three times, because Δ_{par} and Δ_{pos} are defined over the common set of genes only.

4.2 Control Criteria for the Basic Algorithms

For estimating the genotypic diversity the individuals are assumed to be in linear order. A niche is formed by adjacent individuals with a fitness greater than half of the global best and with Δ being smaller than the strategy parameter ε. The fitness threshold is introduced to ignore chromosomes of less quality, as they do not contribute to niching of individuals of high fitness. The center individual of a niche is called its representative. Niches, whose representatives differ by less than ε, are merged regardless of their position. The resulting amount of niches N and the maximum difference of their representatives $\Delta_{N,max}$ are compared to the strategy parameters ε_{Pop} and N_{max} and the population is considered to be *converged*, if:

$$\Delta_{N,\max} \leq \varepsilon_{Pop} \quad \text{and} \qquad N \leq N_{\max} \tag{5}$$

If this criterion is fulfilled the evolutionary search is stopped and the results are handed over to the local searchers for post-optimization or the local procedures are added to the process of offspring generation in case of delayed direct integration.

5 Experimental Results

In the experiments, five test functions taken from GENEsYs [13] and three real world problems were used. Here, they shall be described very briefly only and the interested reader is referred to the literature.

- **Schwefel's sphere** in the range of $[-10^{10},10^{10}]$ and with a target value of 0.01, a unimodal problem known to be easy for ES, but hard for GA.
- **Shekel's foxholes**, a simple multimodal function, easy for EA, but hard for local searchers (target value: exact minimum)
- Generalized **Rastrigin function** (target value: 0.0001) and
- **Fletcher's function** (target value: 0.00001), both of considerable complexity
- **Fractal function** with an unknown minimum. Target value here: -0.5 (from [13]).
- **Design optimization** of a micro optical communication device considering fabrication tolerances as described in detail in [14]. Despite its only 3 parameters, the task involves some difficulty, because it is of extreme multimodal nature.
- The **resource optimization** is based on the scheduling task of 87 batches in chemical industry, where varying numbers of workers are required during the different batch phases [15]. The maximum number of workers per shift (human resource) and the production time shall be reduced to the largest possible extent. Restrictions like due dates of batches, necessary pre-products from other batches, and the availability of shared equipment must also be adhered to. Allocation conflicts are solved by the

sequence of the batches within a chromosome. But as this can be overwritten by suitable changes of the starting times, the combinatorial aspect is limited.

- The objective of the **robot path planning** task is to move a robot along a line as straightly as possible from a starting to a destination point and avoid collisions with itself and some obstacles by controlling the robot axis motors [7]. As the number of necessary motor commands is not known in advance, the chromosomes must be of dynamic length and the order of the commands is essential. Due to a command which tells the robot control to keep the actual motor settings for a specified amount of control cycles, this task has both integer and real parameters.

Table 1. Properties of the eight test cases and the results for the algorithms applied separately.

Experiment	Combinatorial optim.	Parameters	Modality	Implicit restrictions	Success rate for best run		
					GLEAM	Rosenbrock	Complex
Sphere	no	30 real	unimodal	no	0	100	0
Foxholes	no	2 real	multimod.	no	100	3	1
Rastrigin	no	20 real	multimod.	no	100	0	0
Fletcher	no	5 real	multimod.	no	100	10	10
Fractal	no	20 real	multimod.	no	100	0	0
Design	no	3 real	multimod.	no	100	15	12
Resource	(yes)	87 int	multimod.	yes	94	0	0
Robot Path	yes	dynamic mixed	multimod.	yes	100	0	0

These test cases cover a wide range of different application types, as shown in Table 1. The shortcuts and strategy parameters of the algorithms and hybrids, used in the figures below, are as follows:

- G: GLEAM: the population size (p)
- R: Rosenbrock: precision: 10^{-2} (s), 10^{-4} (m), 10^{-6} (l), 10^{-8} (xl), 10^{-9} (xxl)
- C: Complex: none
- Ri: Rosenbrock-initialized start population. Percentage of pre-
- Ci: Complex-initialized start population. optimized individuals
- PR: Rosenbrock post-optimization: see below
- PC1S: Complex post-optimization: each solution is 1 start point (1S)
- PC1C: Complex post-optimization: all solutions form 1 start complex (1C)
- GR, GC: Direct Integration of Rosenbrock or Complex: Lamarck-rate (l), local optimization of all or only the best offspring of one mating (all, best)
- GdR, GdC: Delayed direct integration with Rosenbrock and Complex respectively

The niching is controlled by three parameter settings for ε and ε_{Pop}, P1=(0.005, 0.01), P2=(0.002, 0.005), P3=(0.001, 0.002), and by N_{max} which varies between 2 (p<20) and 5 (p>90) with the population size. The success rate and the average amount of evaluations based on 100 runs (resource and the design task: 50 runs) for each parameterization (job) are taken for comparison. For the resource and the robot problem the local searchers work on the parameters only and leave the combinatorial aspects to the EA. A total of 182,000 jobs consuming 7 cpu-years on 22 sun workstations (ultra sparc) were needed for the experiments.

5.1 Results of the Test Cases

In Fig. 2 the results of the best jobs with 100% success rates are shown for each test case and hybridization method, and they are compared with the best run of GLEAM. Schwefel's sphere is somewhat exceptional, as GLEAM has no success despite the unimodality of the problem. The success of the Rosenbrock procedure is more or less unrealistic, because no one would start with such an extreme precision, and with less there is no success at all up to high precision. The post-optimization is mentioned in brackets, because even the unsuccessful runs deliver very-high-quality solutions. The results of Shekel's Foxholes show that a fairly easy task for an EA, but hard for local searchers, cannot be improved largely by hybridization. The EA solves the problem so well that the overhead imposed by hybridization is too costly. The Rastrigin function shows an unusual behavior in so far, as GLEAM still works with extremely small population sizes leaving little or no room for improvements by hybridization. In Fig. 3 this case is compared with the typical behavior of an EA using the robot path planning task. With population sizes below a certain value, success rates are expected to drop and the number of evaluations to increase. For the resource and the robot tasks, special precisions of the Rosenbrock algorithm of 0.6 and 0.5 respectively were necessary, as there was no convergence with any standard setting. Due to the extended chromosome types of the last two tasks, further niching control parameterizations were tested, and P0 (0.04, 0.08) led to a remarkable success for resource optimization. Together with Fletcher's function, this task yields the most impressive improvements, while the very difficult fractal function and the design problem show significant success, too.

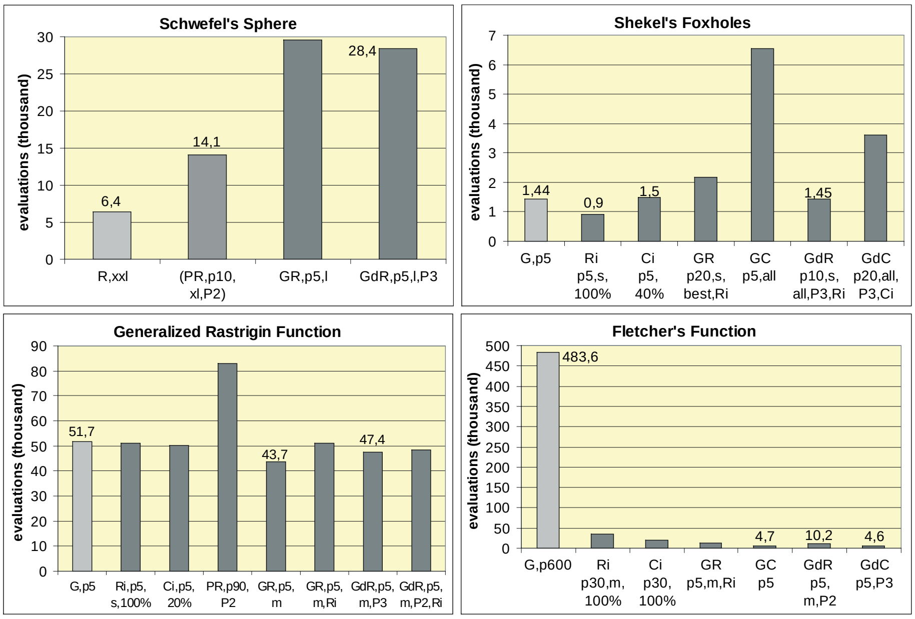

Fig. 2. Results for each test case and hybridization method reaching a 100% success rate. The best jobs per parameterization are shown and compared. Acronyms see above list

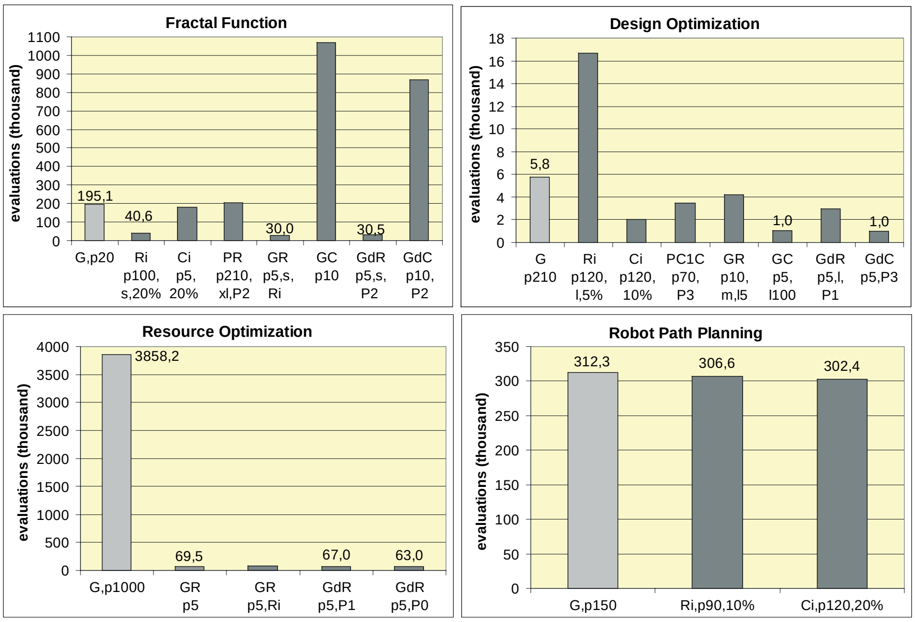

Fig. 2 (cont.). Results for each test case and hybridization method reaching a 100% success rate. The best jobs per parameterization are shown and compared. Acronyms see above list.

The examined hybridization approaches reach their limits, when the ordering of a dynamic amount of genes is of vital importance to success, as it is the case with the robot example. No relevant improvement can be stated in this case.

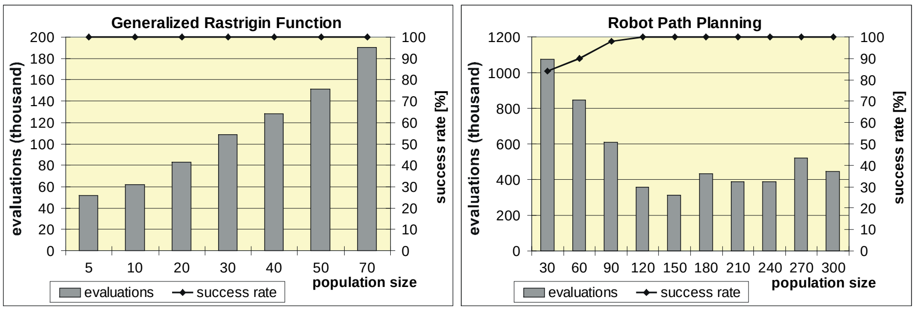

Fig. 3. GLEAM jobs: Untypical success of the Rastrigin function at very small population sizes and usually observed behavior in case of the robot path planning task (100 runs per setting).

5.2 Comparison of the Different Kinds of Hybridization

The different kinds of hybridization are compared on the basis of the achieved improvement of the best job compared to the best GLEAM job. This cannot be done for Schwefel's sphere, because GLEAM was not successful. But it can be stated that post-optimization and (delayed) direct integration yield very good results, see Fig. 2. Furthermore, the robot task will not contribute to the comparison, as there was no improvement worth mentioning. In Fig. 4 the results of the remaining test cases are summarized.

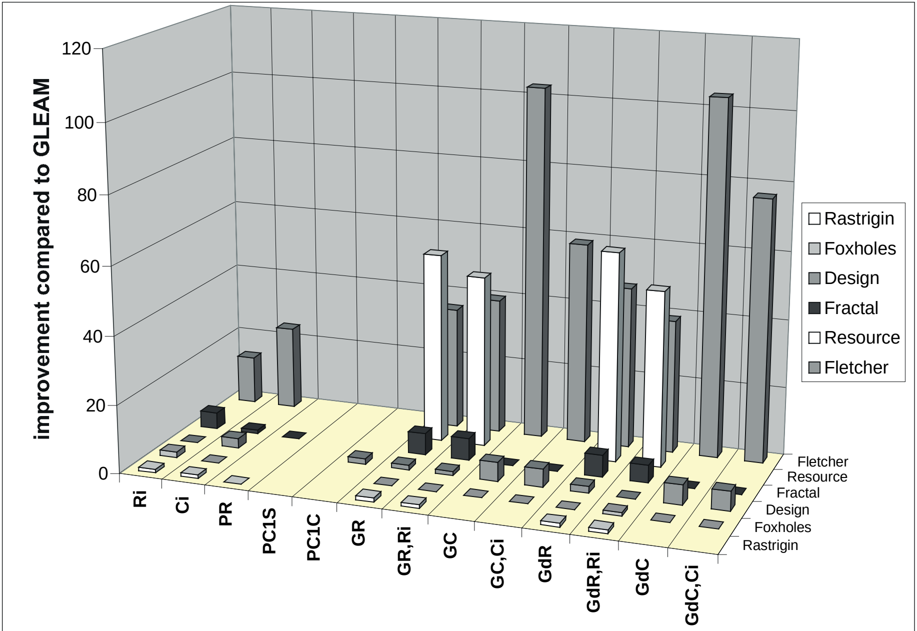

Fig. 4. Comparison of the different kinds of hybridization. Empty fields indicate no sufficient success rate (below 100%), while flat fields stand for fulfilling the optimization task, but with greater effort than GELAM. For PC1S no 100% success rate was reached in any test case.

Conclusions

From the extensive investigation of 13 different kinds of hybridization, based on eight test cases comprising one challenging unimodal problem, simple and complex multimodal tasks, and real world problems involving combinatorial optimization as well as dynamic parameter strings, the following conclusions can be drawn, see also Fig. 4:

1. Though post-optimization improves the results obtained by the EA in most cases, it does not lead to sufficient success, as the introduced control mechanism based on niche detection does not guarantee a stop of evolution only, when the area of the attraction of the global optimum is reached.

2. Direct or delayed direct integration of the Rosenbrock procedure works in all cases and yields very good results as long as the problem is not too simple like in the case of Shekel's foxholes. Pure Lamarckian evolution and local optimization of the best offspring in nearly all cases is superior to Baldwinian evolution or optimization of all offsprings of one mating. The delayed integration based on the niching

control algorithm improves the undelayed version by up to 20% less evaluations. Small population sizes between 5 and 20 are sufficient.

3. Pre-optimization helps in most cases but direct integration is better.
4. Hybridization with the Complex algorithm does not always work, but if it does, superior results can be produced. Hybridization with the Rosenbrock procedure is more reliable, but not always as successful as using the Complex.

Although these conclusions are based on the test cases investigated, it can be assumed that they are not limited to them. Thus, it can be recommended to use delayed or undelayed direct integration of the Rosenbrock algorithm to speed up the EA while maintaining the properties of reliable global search and general applicability. With the Rosenbrock procedure, evaluations were found to be reduced by the magnitude of 60 and by using the Complex algorithm instead, a factor of up to 100 can be achieved.

This paper was written using the terms and definitions of VDI/VDE 3550 [16].

References

1. Holland, H.J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
2. Goldberg, D.E, Voessner, S.: Optimizing Global-Local Search Hybrids. In: W. Banzhaf et al. (eds.): Proc. GECCO'99, Morgan Kaufmann, San Mateo, CA (1999) 220-228
3. Whitley, D., Gordon, V., Mathias, K.: Lamarckian Evolution, The Baldwin Effect and Funct. Opt. In: Davidor, Y. et al.: Proc. PPSN III, LNCS 866, Springer, Berlin (1994) 6-14
4. Gruau, F., Whitley, D.: Adding Learning to the Cellular Development of Neural Networks: Evolution and the Baldwin Effect. Evol. Comp. 1, Vol.3 (1993) 213-233
5. Orvosh, D., Davis, L.: Shall We Repair? Genetic Algorithms, Combinatorial Optimization, and Feasibility Constraints. In: Forrest, S. (ed): 5th ICGA, M. Kaufmann (1993) 650
6. Jakob, W., Quinte, A., et al.: Opt. of a Micro Fluidic Component Using a Parallel EA and Simulation Based on Discrete Element Methods. In: Hernandez, S., et al.: Computer Aided Design of Structures VII, Proc. of OPTI'01, WIT Press, Southampton (2001) 337-346
7. Blume, C.: GLEAM - A System for Intuitive Learning. In: Schwefel, H.P., Männer, R. (eds.): Proc. of PPSN I, LNCS 496, Springer, Berlin (1990) 48-54
8. Blume, C., Jakob, W.: GLEAM – an Evolutionary Algorithm for Planning and Control Based on Evolution Strategy. Conf. Proc. GECCO 2002, Vol. Late Breaking Papers, (2002)
9. Jakob, W.: HyGLEAM: Hybrid GeneraL-purpose Evolutionary Algorithm and Method. In: Callaos, N. et al. (eds.): Proc. SCI'2001, Vol. III, IIIS, Orlando, (2001) 187-192
10. Rosenbrock, H.H.: An Automatic Method for Finding the Greatest or Least Value of a Function. Comp. Journal 3 (1960) 175-184
11. Box, M.J.: A New Method of Constrained Optimization and a Comparison with Other Methods. Comp. Journal 8 (1965) 42-52
12. Schwefel, H.-P.: Evolution and Optimum Seeking. John Wiley & Sons, Chichester (1995)
13. Bäck, T.: GENEsYs 1.0, ftp://lumpi.informatik.uni-dortmund.de/pub/GA
14. Gorges-Schleuter, M., Jakob, W., Sieber, I.: Evolutionary Design Optimization of a Micro-optical Collimation System. In: Zimmermann, H.J. (ed.): Proc. Eufit'98, Verlag Mainz, Aachen (1998) 392-396
15. Blume, C., Jakob, W.: Cutting Down Production Costs by a New Optimization Method. In: Proc. of Japan-USA Symposium on Flexible Automation. ASME (1994)
16. Beyer, H.-G., et al.: Evolutionary Algorithms – Terms and Definitions. VDI/VDE-Richtlinie-3550, Blatt 3, Gründruck (Engl. vers. to be published in 2002). VDI, Düsseldorf (2001)