Control of Parallel Population Dynamics by Social-Like Behavior of GA-Individuals*

Dirk C. Mattfeld[1], Herbert Kopfer[2] and Christian Bierwirth[2]

[1] LRW Computing Center, 28334 Bremen, Germany
[2] Department of Economics, University of Bremen

Abstract. A frequently observed difficulty in the application of genetic algorithms to the domain of optimization arises from premature convergence. In order to preserve genotype diversity we develop a new model of auto-adaptive behavior for individuals. In this model a population member is an active individual that assumes social-like behavior patterns. Different individuals living in the same population can assume different patterns. By moving in a hierarchy of "social states" individuals change their behavior. Changes of social state are controlled by arguments of plausibility. These arguments are implemented as a rule set for a massively-parallel genetic algorithm. Computational experiments on 12 large-scale job shop benchmark problems show that the results of the new approach dominate the ordinary genetic algorithm significantly.

1 Introduction

A major topic of recent research in genetic algorithms (GA) focuses on the ability to control population dynamics. The difficulty in reaching this goal is caused by the complex mechanisms of selection. Optimal balancing of selection pressure means to adjust it as sharply as possible without destroying genotype diversity. In this paper we consider a new approach to control convergence by means of social-like behavior patterns (SBP). The phenomena of social hierarchies and individual behavior can be found in many natural populations. Thus it seems worthwhile to explore the power of social mechanisms within the GA paradigm.

Section 2 gives a brief survey on related approaches. Section 3 introduces our metaphor of social-like behavior. We show how to translate SBP into a local recombination strategy and how to implement it in massively-parallel GA's. Section 4 presents the test platform which is a standard model of job shop scheduling. The job-shop specific components of local recombination are briefly described. Finally we report our computational results.

2 Controlled Convergence

The most frequently observed difficulty in the application of standard GA's to large-scale optimization problems results from premature convergence. While GA heuristics perform well for small problems they often suffer from strong sub-

* Supported by the Deutsche Forschungsgemeinschaft (Project Parnet)

optimality in larger ones. Population dynamics of standard GA's is known to be responsible for premature convergence. In order to improve the average fitness of a population the best individuals are given the highest selection rates for reproduction. Hence a dynamic is enforced that decreases genotype diversity continuously. Thus the reproduction scheme's ability to build new solutions becomes smaller and smaller and eventually the GA converges.

Several approaches of maintaining a population's diversity were proposed. Some direct approaches focus on genetic operators. Obviously, a more disruptive crossover delays the reduction of genotype diversity. The success of these strategies is considered to be highly problem specific. More general strategies are going by the term of controlled convergence.

2.1 Mating Strategies

Whenever proportional mate selection leads to premature convergence more sophisticated mating strategies can be used to slow down convergence. Due to this analogy mates practice incest if their genotypes are too similar. Some attempts were made to avoid such constellations. Goldberg introduced a mechanism into mate selection called sharing functions. In this approach the fitness of individuals is used as an indicator for genotypical similarity. More direct approaches of incest prevention are based on the measure of similarity between genotypes by means of the hamming distance. A brief survey of such techniques can be found in a paper of Davidor [4]. Another approach proposed by Eshelman and Schaffer controls mate selection by a threshold [7]. The threshold forbids mating if it dominates the hamming distance of mating candidates. Whenever a population cannot progress, the threshold value is decremented in the next generation.

2.2 Structured Populations

Further approaches to control convergence result from structuring the entire population. This is either reached by splitting up population into islands (migration model) or by defining overlapping neighborhoods which cover it entirely (diffusion model). Both models restrict immediate data flow between individuals to local areas of the population. Thereby the nature-like feature of ecological niches is introduced into population dynamics. This feature is expected to preserve genotype diversity within useful bounds.

Since 1988, when Mühlenbein and Gorges-Schleuter developed the diffusion model [11] this kind of population structure received increasing attention. The probably most exciting feature of neighborhood structures is the explicit introduction of parallelism. It enables individuals to act without centralized control. Mate-selection and offspring-replacement are done locally. For this reason the diffusion model fits the restrictions of massively-parallel environments. A combined approach of traditional GA's and diffusion models is shown by the ECO-framework of Davidor [4]. A brief outline of other local mating strategies proposed is published in a recent paper of Gorges-Schleuter [8].

3 Incorporating Patterns of Social Behavior

In structured populations derived from diffusion models mating is restricted to a small number of nearby individuals. Hence global premature convergence is alleviated at the expense of incest in the neighborhood.

While the environment changes constantly at a slow pace for the individuals, evolution from generation to generation works well. If locality is introduced, the evolutionary process is too fast to solve some of the problems faced by individuals of the population. Our approach extends the global GA adaptation towards the adaptation of a single individual to its neighborhood. Each individual must respond to its own specific environmental conditions. Thus individuals are able to change behavior usefully as a function of immediate changes of the environment.

The psychologist school of Behaviorism became important in the early days of the 20^{th} century. Staats gives a comprehensive survey in [12]. He emphasizes that complex functional behavior of an individual is learned and that environmental events can affect the individuals behavior. Thorndike laid the foundations in 1898 with his "law of effect": One effect of successful behavior is to increase the probability that it will be used again in similar circumstances. Rewards granted in case of success lead to patterns of behavior, called habits. In 1947 Doob extended the formal learning theory to the consideration of attitudes. He suggested that attitudes are anticipatory responses which can mediate behavior. An attitude can be seen as a disposition to react favorably or unfavorably to a class of environmental stimuli. Staats notes that in social interactions attitudes are formed by social rewards which stimulate reinforcement on a certain behavior.

3.1 The Metaphor

We borrowed the basic ideas of our metaphor from the Behaviorists. As shown in figure 1 we classify individual behavior by three general cases. The initial attitude of individuals is an established one, i.e. they all act cooperatively within their environments. Secondly, the elitist attitude follows a conservative behavior pattern. The last attitude is a more critical one, which tends to be risk-prone. The actual behavior of each pattern is rewarded in terms of social interaction. Again we classify three simple responses which are defined by reinforcements. An individual can be pleased, satisfied or disappointed. The success of the actual behavior carried out may change its attitude and therefore changes its habit in a similar situation within the near future. The individual will react differently and may receive a different reinforcement on the same environmental situation.

In most cases a cooperative individual will be satisfied and therefore does not change its attitude. If pleased by success of its habit, next time it will tend to act conservative trying to keep its previous performance level. With this elitist attitude an individual can only be satisfied or disappointed by the success of its habit. In case of disappointment it will change back to the established attitude. Failing on cooperative behavior brings up a critical attitude of the individual towards its neighborhood environment. It will then tend towards a more risk-prone behavior. The critical attitude is kept so long as a disappointing response

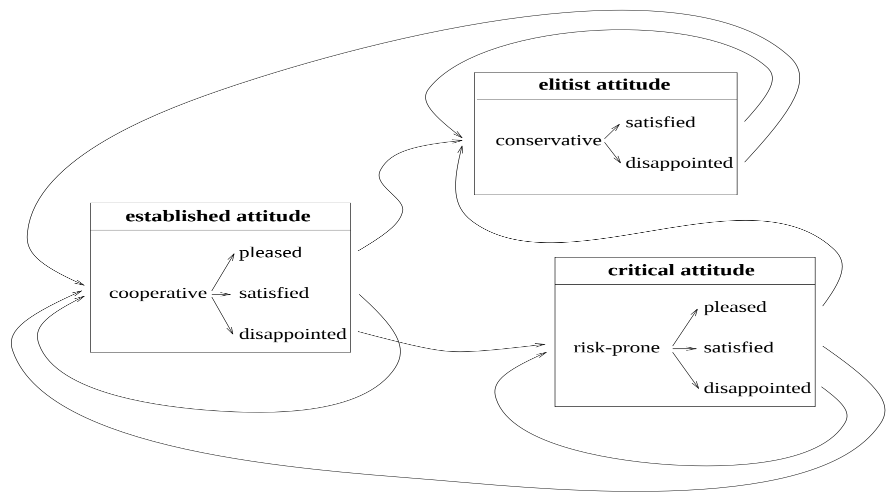

Fig. 1. Scheme of social behavior patterns.

is still received. If the individual is satisfied by the result of its behavior, it may change to the established pattern again. In rare cases a risk-prone individual will receive a pleasing response. Then it changes towards the elitist attitude.

Don't expect figure 1 to be a blueprint of the complete transition structure of the attitude changes. In fact the response on a certain behavior gives only a rough idea of which attitude may be suitable for the next trial. In general, attitudes are changed only after a number of identical reinforcements. Strong reinforcements can lead to immediate attitude changes, while, in general, weak and moderate reinforcements lead to memory adjustments only.

3.2 Translating the Metaphor into the Model

In our population model individuals reside on a torodial square grid. Mating of individuals is restricted to the North, South, West and East neighbors. Selection is carried out locally by the ranking scheme of 40%, 30%, 20% and 10% from the best to the worst-fit neighbor. An individual is replaced in the population if the fitness of its offspring is at most 2.5% worse.

In order to implement social behavior patterns we transform our metaphor into a local recombination strategy. The established attitude corresponds to co-operation with one of the neighbors by crossover. The critical attitude corresponds to a mutation. The conservative behavior tries to save the reached state. The individual performs no active operation (i.e. is sleeping) to avoid replacement by offspring. Figure 2 show these operations in boldface.

First an individual compares its fitness with the neighborhood. If the fitness is superior to all neighbors, the conservative behavior will cause the individual to sleep. If several best individuals exist in one neighborhood none of them will be superior. For this reason incorporating SBP does not introduce an elitist strategy into PGA. An inferior individual determines its attitude. The actual behavior

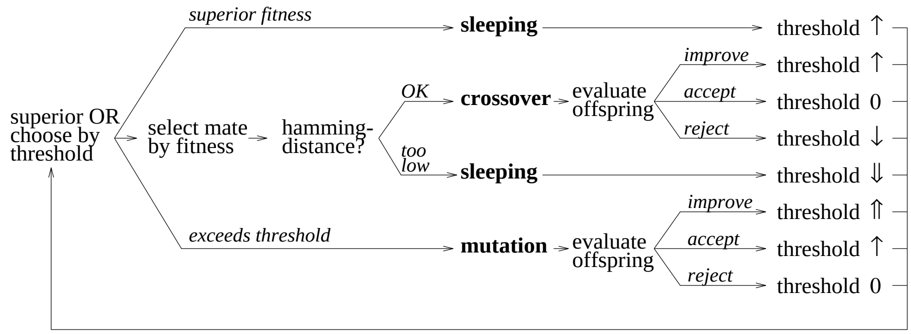

Fig. 2. Control model of local recombination.

is drawn probabilistically from a threshold. Initially the threshold is set to 1, which enforces crossover. Decreasing the threshold increases the probability of mutations. In case of crossover, the hamming distance to the selected mate is evaluated. If mates differ in less than 1% of their genes it is not worthwhile to try a crossover. Again, the individual sleeps, but now because of a different reason. If crossover or mutation is carried out, the offspring is evaluated. Either an offspring dominates its parents (improve), or the acceptance rule decides whether to replace the individual (accept/reject).

Summing up all distinct operations we count 8 responses which are tied to reinforcements of the threshold. We modify the threshold by rules of plausibility. The symbols "$\Uparrow\Downarrow\downarrow\uparrow$" express the degree of change of the threshold. This rule set attempts to adjust the behavior of each single individual towards the environment of its actual neighborhood. In our implementation the threshold vector $(+0.02, +0.05, 0, -0.02, -0.20, +0.15, +0.05, 0)$ performed well. This setting reacts adaptively on incest occurrence with a strong decrease of the threshold. It favors risky behavior by mutations in further generations. If a mutation succeeds, the threshold is increased which in turn leads to crossover.

4 Job Shop Scheduling

Job shop scheduling is known to be a difficult combinatorial optimization problem. It has become a popular platform for comparison of modern heuristics, e.g. Simulated-Annealing (Laarhoven et. al. [9]) and Tabu-Search (Taillard [13], Dell'Amico et. al. [6]). Furthermore many GA approaches were proposed, e.g. the Giffler-Thompson-GA (Davidor et. al. [4]). Nevertheless, concerning scheduling Simulated-Annealing and especially Tabu-Search perform better than GA's.

The standard problem of job shop scheduling can now be described roughly. A production program containing n jobs is released. Each job is split into m operations that must be processed by m dedicated machines in a predefined technological order. The processing times of operations are known and considered as tasks on machines. While several assumptions concerning the way of processing tasks are made (e.g. no preemption of operations) a schedule has to be found that optimizes a certain measure of performance. In order to compare computational results we consider the measure that is predominantly used in literature: To minimize of the total completion time, i.e the makespan.

PGA uses a genetic representation of the job shop problem described in [3]. In this approach a solution's genotype is defined by a permutation with repetition. The permutation contains $n \times m$ genes from the alphabet $\{J_1, \ldots J_n\}$, where J_k denotes job k. Within a chromosome each gene is repeated exactly m times. E.g., if $n = 2$ and $m = 3$ the chromosome "$J_2\, J_1\, J_1\, J_2\, J_2\, J_1$" represents a feasible genotype. The i-th occurrence of a gene in the string refers to the i-th operation of the named job. Crossover is done by a generalized order-crossover technique [3]. It preserves the n/m permutation-structure and inherits nearly half of the relative gene-order from both parents to the offspring. Mutation results from a position-based random-change of an arbitrary gene.

An elegant way of modeling the job shop problem via a disjunctive graph is given by Adams et. al. [1]. Each oriented acyclic graph in the disjunctive graph represents a feasible solution. The cost of the longest-path in such a graph gives the makespan of the represented solution. Solving a job shop problem means to find the acyclic oriented graph with the minimal longest-path.

Evaluation of genotypes is done in three steps. First we transform a chromosome into an acyclic graph and calculate its longest path. Second we apply local-search to reduce makespan by exchanging adjacent operations on the longest path as described by Taillard [13]. Finally the reoptimized graph is transformed back into a genotype.

5 Computational Results and Analysis

Two versions of PGA were run on a suite of 12 problems. It includes the famous "10×10" and "20×5" Muth-Thompson benchmarks and 10 other difficult problems, see Applegate and Cook [2]. In version 1 PGA was running alone whereas version 2 incorporates the SBP model (parameterized as described in 3.2). All routines are written in C++ by massively use of the LEDA-library [10]. The algorithms run on a SUN/10 workstation.

5.1 The Muth-Thompson Benchmarks

PGA and PGA+SBP were run for a total of 200 iterations. The population size was set to 100, termination occurred after 150 generations. Experimental results appear in table 1. The column "Trials" refers to the average number of genotype evaluations of a single run. A single evaluation requires about 1 millisecond

		PGA+SBP				PGA			
Problem	Opt	Average	Best	Var	Trials	Average	Best	Var	Trials
mt (10x10)	930	947	930	8.2	10935	950	934	9.2	15000
mt (20x5)	1165	1188	1165	10.3	11738	1190	1173	10.7	15000

Table 1. Comparison of results (makespan) on the Muth-Thompson problems.

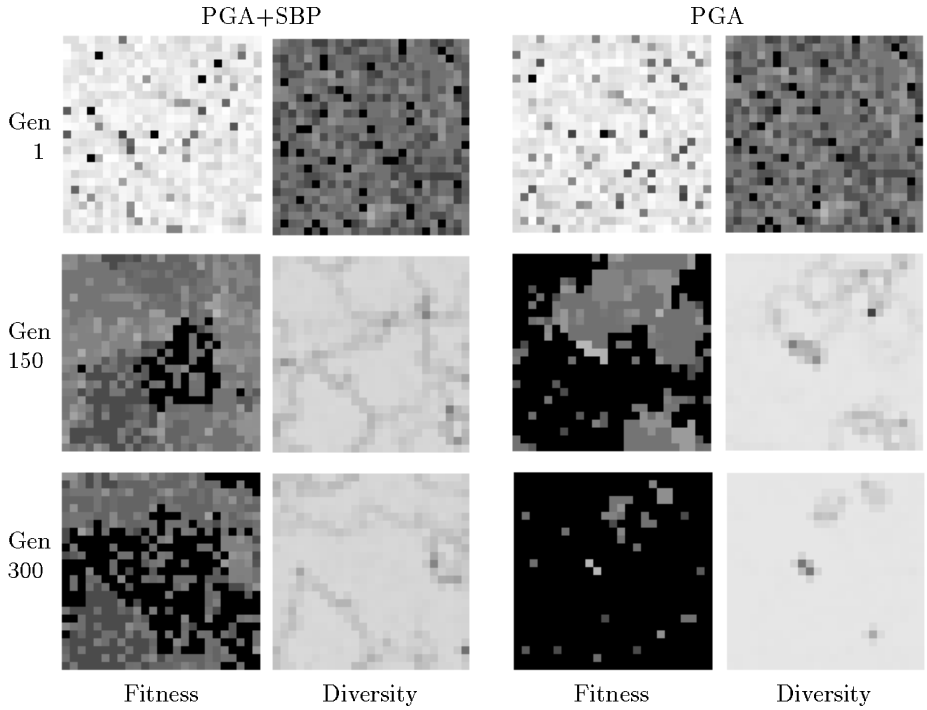

Fig. 3. Convergence process of arbitrary PGA and PGA+SBP runs.

for both problems, thus PGA solves a problem in less than 3 minutes. The computational amount arising from incorporating SPB increases evaluation time by approximately 5%. But the smaller number of genotype evaluations leads to a shorter runtime of 2.3 minutes in average for PGA+SBP.

Figure 3 illustrates the evolution of 625 individuals solving the "mt10x10" problem. Snapshots on the initial, the medium and the final generation show the distribution of fitness and diversity on the torodial grid. We calculate the diversity of individuals by the average hamming distance to 4 neighbors. Greyscale refers to levels of fitness and diversity by using darker colors to indicate larger values (min makespan $\equiv$ max fitness).

It can be seen that in the PGA-run as well as in the PGA+SBP-run average fitness increases at the expense of decreasing diversity. PGA permits highly fit individuals to spread rapidly into areas of lower fitness in their vicinity. After 150 generations the population is dominated by two large fitness plateaus. Moderate genotype diversity exclusively remains in the lower plateau. This niche is completely driven out by the upper plateau after another 150 generations. Uniformity of genotypes inside the dominating plateau prevents further progress. Promising search is limited to the borderlines of plateaus.

To the contrary, PGA+SBP evolves different regions of similar genotypes simultaneously. Borderlines between such regions can be seen clearly on the diversity map after 150 generations. The corresponding fitness map still shows

		PGA+SBP			PGA		
Problem(Size)	Known	Average	Best	Trials	Average	Best	Trials
la21 (15x10)	1048 [6]	1061.5	1053	24956	1064.9	1053	30000
la24 (15x10)	*935 [2]	948.0	938	25335	954.2	938	30000
la25 (15x10)	*977 [2]	989.1	977	25733	990.8	984	30000
la27 (20x10)	1242 [6]	1265.7	1236	24797	1266.7	1256	30000
la29 (20x10)	1180 [13]	1214.4	1184	25013	1222.9	1185	30000
la38 (15x15)	1203 [6]	1222.4	1201	24026	1234.1	1206	30000
la40 (15x15)	*1222 [2]	1243.5	1228	47881	1254.1	1233	57600
abz7 (15x20)	667 [13]	684.6	672	45812	685.4	675	57600
abz8 (15x20)	670 [9]	697.9	683	44664	698.7	687	57600
abz9 (15x20)	691 [9]	712.6	703	45004	715.8	704	57600

*optimal.

Table 2. Results for 10 "tough" job shop problems (see [2], table V).

heterogeneity, i.e. SBP prevents the building of rigid fitness plateaus. Even after 300 generations the situation has hardly changed. Although regions of higher fitness have grown the population is still active. Summing up, both versions of PGA lead to optimal or near-optimal solutions of the Muth-Thompson benchmarks. Although both versions come along different population dynamics, the reached quality of solutions differs insufficiently in order to allow general statements.

5.2 Large-Scale Benchmarks

The application of SBP to moderate job shop problems (up to 100 operations) cannot show all properties of the SBP model. The improvement by genetic search is so fast that the decreasing diversity does not prevent the algorithms from finding good solutions. Therefore we focus on the 10 most difficult problems of the 53 benchmark-suite provided by Applegate and Cook.

Now, PGA and PGA+SBP were run for a total of 30 iterations on each problem. The population size was set to 100 in 6 instances and to 144 in 4 even more difficult cases. Termination was set to 300 and 400 generations respectively. Experimental results appear in table 2. The computation time for a single evaluation scales up dramatically with problem size. E.g. we need a total computation time of 30 minutes to solve a 20×10 problem on a single workstation. It can be seen from the table, that PGA+SBP decreases the number of trials by 20%.

For all problems the average makespan generated by PGA+SBP is significantly better than the corresponding PGA result. For the relatively small 15×10 problems the best found solutions are similar in both strategies. The best of all found solutions were always generated by PGA+SBP. These solutions are within a 13-unit range of the best known solutions. For the problems "la27" and "la38" new best solutions were found. Notice that "la27" is solved by a makespan of 1236. This value differs only one unit from the theoretical lower bound, thus we assume the problem to be solved to optimality.

Figure 4 compares arbitrary runs of PGA and PGA+SBP solving "la27". The upper pictures show fitness progress for the population's average and for the best

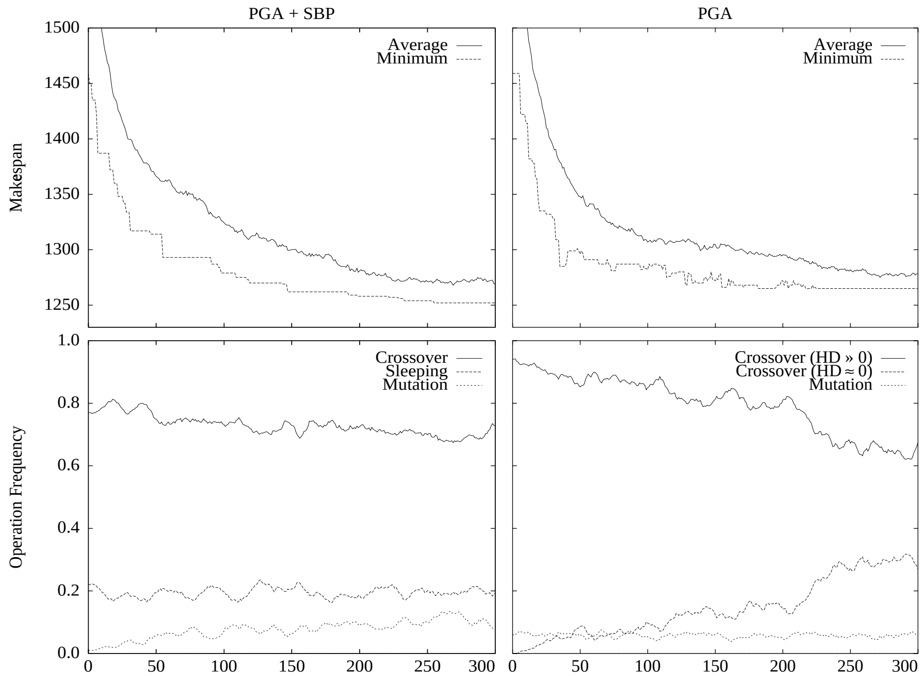

Fig. 4. Population dynamics of PGA and PGA+SBP.

individual. Compared with PGA the PGA+SBP strategy slows convergence in early generations and retains average fitness longer. PGA stagnates in generation 200 whereas PGA+SBP generates improvements continuously.

The lower pictures report the operation frequency over time. PGA uses only two genetic operators which are crossover and mutation. The probability of mutation is fixed at 5%. Exemplary crossover operations are divided into two classes. A crossover falls into the first class if the mates differ by at least 1% of their genes. This class is shown in the upper curve. It can be seen that the condition to match class one decreases approximately linear in time. To the contrary, the increasing lower curve documents the incest rate. This rate refers to mating of individuals inside dominating fitness plateaus (compare figure 3).

The corresponding picture of PGA+SBP indicates a totally different population dynamic. Whereas the crossover frequency decreases slowly with increase of mutations, sleeping frequency is fixed at about 20%. Whenever incest occurs the affected individual sleeps in this generation and increases the probability of mutations in future generations. Actually, sleeping has two distinct reasons: Sleeping caused by incest trials and, secondly, by superior fitness within a neighborhood (see figure 2). During the initial phase of algorithm sleeping is mainly caused by superior fitness, whereas during later phases sleeping is mainly triggered by incest trials. Amazingly, adding both cases leads to a nearly constant frequency of sleeping during the whole process. This seemly stationary sleeping process is conjectured to be the most important condition for SBP to work successfully. Bear in mind that solely local adjustment of genetic operators leads to this overall balance of population dynamics.

6 Summary

This paper presents a first approach to incorporate individual behavior into genetic algorithms. Applying our approach to job shop scheduling enabled us to find some new best solutions for a suite of difficult benchmark problems. We do not claim that incorporating SBP into GA's can be a successful strategy for all kinds of optimization problems, e.g. the emulation of our "risk-prone" behavior pattern by mutations might fail. This is the case if a mutation/hill-climbing strategy performs poorly on the problem. Furthermore our parameterizing of reinforcements of attitudes might seem arbitrary. Of course, its evidence is derived from plausibility but the setting requires at least some computational experience from the domain of the optimization problem. In conclusion, our results encourage the combination of long-term learning by inheritance and short-term learning by behavior in further research of simulated evolution.

References

1. Adams, J., Balas, E., Zawack, D.: The Shifting Bottleneck Procedure for Job Shop Scheduling. Management Science **34** (1988) 391–401
2. Applegate, D., Cook, W.: A Computational Study of the Job-Shop Scheduling Problem. ORSA Journal on Computing **2** (1991) 149–156
3. Bierwirth, C.: A Generalized Permutation Approach to Job Shop Scheduling with Genetic Algorithms.OR Spektrum (to appear)
4. Davidor, Y.: A Naturally Occurring Nice & Species Phenomenon: The Model and First Results. Proc. of ICGA **4**, Morgan Kaufmann (1991) 257–263
5. Davidor, Y., Yamada, T., Nakano, R.: The ECOlogical Framework II, Improving GA Performance at Virtually Zero Cost. Proc. of ICGA **5**, Morgan Kaufmann (1993) 171–176
6. Dell'Amico, M., Trubian, M.: Applying Tabu-Search to the Job Shop Scheduling Problem. Annals of Operations Research **41** (1993) 231–252
7. Eshelman, L.J., Schaffer, J.D.: Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. Proc. of ICGA **4**, Morgan Kaufmann (1991) 115–122
8. Gorges-Schleuter, M.: Comparison of Local Mating Strategies in Massively Parallel Genetic Algorithms. Proc. of PPSN **2**, North-Holland (1992) 553–562
9. van Laarhoven, P.J., Aarts, E.H., Lenstra, J.K.: Job Shop Scheduling by Simulated Annealing. Operations Research **40** (1993) 113–125
10. Mehlhorn, K., Näher, S.: LEDA, A Library of Efficient Data Types and Algorithms. TR **A 05/89** University of Saarbrücken (1989)
11. Mühlenbein, H., Gorges-Schleuter, M.: Die Evolutionsstrategie: Prinzip für parallele Algorithmen. GMD Annual Report (1988)
12. Staats, A.W.: Social behaviorism. The Dorsey Press, Illinois (1975)
13. Taillard, E.: Parallel Taboo Search Techniques for the Job Shop Problem. TR **ORWP 89/11**, Ecole Polytechnique de Lausanne (1989)