Dynamic Neighborhood Structures in Parallel Evolution Strategies

Klaus Weinert, Jörn Mehnen
Department of Machining Technology, University of Dortmund
44221 Dortmund, Germany
{weinert,mehnen}@isf.mb.uni-dortmund.de

Günter Rudolph
Department of Computer Science, University of Dortmund
44221 Dortmund, Germany
rudolph@LS11.cs.uni-dortmund.de

Parallelizing is a straightforward approach to reduce the total computation time of evolutionary algorithms. Finding an appropriate communication network within spatially structured populations for improving convergence speed and convergence probability is a difficult task. A new method that uses a dynamic communication scheme in an evolution strategy will be compared with conventional static and dynamic approaches. The communication structure is based on a so-called diffusion model approach. The links between adjacent individuals are dynamically chosen according to deterministic or probabilistic rules. Due to self-organization effects, efficient and stable communication structures are established that perform robust and fast on a multimodal test function.

1. Introduction

Evolutionary algorithms are numerical optimization algorithms that are inspired by the principle of biological evolution. A population of individuals represents possible solutions of an optimization problem. This set of solutions is continuously varied by genetic operators like recombination, mutation and selection. An iterative application of these operators leads to an adaptation of the individuals in an environment that is unknown a priori. The proper adjustment of the process parameters and the tuning of the interplay of the operators are essential for a successful deployment of evolutionary algorithms.

The biological principle of evolution is inherently parallel. In nature genetic interaction and selection of the fittest individuals happen asynchronously. Something similar to specialization of individuals can be observed if spatial or temporal separation becomes part of the evolutionary model. In nature separation and interaction of individuals are permanent selforganizing processes that allow the perpetuation of a high genetic variety for preventing the species to become extinct. Accordingly to these observations, the analyses of evolutionary algorithms with parallel evolving and dynamicly interacting populations are in the focus of the experiments described here.

2. Parallel Evolutionary Algorithms

The term 'evolutionary algorithm' (EA) is a superordinate concept of Genetic Algorithms (GA), Evolutionary Programming (EP), Genetic Programming (GP) and Evolution Strategies (ES). EA is a class of direct and randomly driven optimization algorithms that belongs to a more universal set of methods subsumed under the label Computational Intelligence (CI) [Bezdek94], which also comprises the fields of fuzzy logic and neural networks. These techniques are subsymbolic (numeric) and excel by their adaptability, fault tolerance, and a high processing speed when applied to complex problems. CI methods are often inherently parallel. Here, parallelization does not only help to increase the performance of these strategies. The usage of structure and the distributed processing of data introduces a new quality to these systems. In this article the focus will lie on parallel evolutionary algorithms.

2.1 Static Population Structures

Parallel evolutionary algorithms with static population structures can be grouped as follows [Sprave99]:

- Panmictic model: the population is not explicitely structured. Every individual can interact genetically with each other.
- Multipopulation models:
  - Migration model: a population consists of separated subpopulations. Each subpopulation has a panmictic structure. A limited amount of individuals can migrate between the subpopulations on predetermined pathes. The data is moved and not copied between the populations.
  - Pollination model: this model is similar to the migration model but the data is copied and not moved.
- Neighborhood models:
  - Metric neighborhood model: the individuals of one common population are structured according to their spatial relations. Interactions between individuals happen only between elements that are neighbors, i.e. they have the same distance relation.
  - Relational neighborhood model: this model is similar to the metric neighborhood model, except that neighboring individuals share a more general relation than just a spatial distance.

Multipopulation models In this context the terms 'island model' and 'stepping stone model' are often used. Island models are multi-population models where data can migrate between any subpopulation [Hal30]. Stepping stone models follow a more restricted communication scheme, i.e. the subpopulations are arranged on a ring [Kim53] or a torus or a grid [Mar70].
Theoretic investigations show that island models may behave panmictic when the migration rate reaches a certain level. This threshold does not depend on the population size, but depends on the number of islands and the migration rate between the islands. It can be shown that the speed of convergence and the convergence probability can be increased when very small migration rates are used (e. g. one migrant per generation). On multimodal problems like Rastrigin's function $f(x) = \sum_{i=1}^{30}(x_i^2 - 50\cos(2\pi x_i))$ multipopulation models show a higher convergence probability than panmictic strategies. Other multipopulation strategies like Cohoon's approach of *punctuated equilibria* [Coh87] or Rechenbergs nested evolution strategies [Rech94, Rud00] also show, that structured populations have advantages compared to panmictic approaches.

Neighborhood models Neighborhood models are sometimes also called 'diffusion models'. Due to their typically used grid structure, data spreads slowly through the population. Genes that have a good fitness are transfered from neighbor to neighbor and, thus, 'infect' the population by a diffusion process.
Gorges-Schleuter [Gor89, Gor91] and Sprave [Spr99] introduced algorithms that were also implemented on parallel computer hardware. They show, that a diffusion model applied to complex problems – like the travelling salesman problem – can have a higher convergence probability than panmictic EA. Diffusion models are not only interesting because they are robust. They also show complex dynamics and can be scaled easily on parallel computers by assigning subgrid structures of different sizes to each processor.

2.2 Dynamic Population Structures

Halpern [Hal99] introduced an evolutionary algorithm with a dynamic population structure. It is based on a structurally dynamic cellular automaton. Here, one individual is assigned to one corresponding cell of the cellular automaton. Starting with N cells, the initial connections of the cells are random but isolated individuals are not allowed. During the initialization phase, each individual is assigned a vector with randomly chosen real valued numbers. During the optimization process, the connectivity of the cells is changed according to deterministic rules that depend on the fitness of the individuals. A predefined percentage of connections to cells with comparatively bad fitness values are deleted. The same time a list of next-nearest neighbors is created and a certain percentage of new connections to the best individuals in that list is established. A next-nearest neighborhood relation characterizes two individuals that are interconnected with each other over a common neighbor. During the evolutionary phase, each individual mates by intermediate recombination with a randomly chosen partner within its local neighborhood. During this step $N/2$ new elements are generated that undergo – with a certain predefined probability – mutation by adding or subtracting small random increments. The set of $N/2$ offspring replaces the $N/2$ least fit elements of the general population. The surviving offspring are placed randomly on the grid sites to replace the least fit individuals. The algorithm is iterated as long as a termination criterion does not hold.

The approach of Halpern has many similarities with an Evolution Strategy running on a diffusion model. Great differences can be seen in the way mutation, recombination and population replacement after selection is used. Like in Genetic Algorithms an explicit step size adaptation is not used. The mutation rate is kept at a predefined fixed level and the step sizes are limited to small intervals.

3. An Alternative Evolution Strategies with Dynamic Neighborhood Structures

The paragon for the alternative parallel optimization algorithm discussed next is the Evolution Strategy, as introduced by Rechenberg [Rech94] and Schwefel [Sch95]. The algorithm has the following scheme:

- $t := 0$
- define initial population P^0
- define initial neighborhood structure L^0
- calculate matrix of next-nearest neighbors M^0

while termination condition not valid **do**
    $t := t + 1$
    for $\forall$ individuals $I_i \in I$ **do**
        - specify new neighborhood relations for each I_i by application of the neighborhood rule (new L_i^t)
        - specify new next-nearest neighbors of I_i (new M_i^t)
        for $\forall\, \lambda$ offspring **do**
            - select a mating partner for recombination I_r from the neighborhood of I_i
            - recombine λ offspring from the parents I_i and I_r
        - mutate the λ offspring (object variables and step sizes)
        - evaluate λ offspring by application of f
        - select the best individual from the λ offspring to replace individual I_i of population P^{t+1}

During the initalization phase a start population P^0 and a adjacency matrix L^0 is used. L_i^0 characterizes the local neighborhood of individual $i = 1, \ldots, N$. Typically the neighbors are chosen from a set of randomly selected individuals. The size of the set fluctuates around a predetermined average value. Hence, the case of isolated individuals or isolated subgraphs is allowed. The matrix M^t of next-nearest neighbors at iteration $t \geq 0$ can always be calculated from the current adjacency matrix L^t.

The main loop of the optimization algorithm consists of two parts: In the first phase the neighborhood structure (L^t) is altered depending on the user-defined rules for the neighborhood dynamics. Any connections between neighbors may be deleted or established during that step. In the second phase the genetic operators of an evolution strategy, namely recombination and mutation, are applied to each individual λ times in order to generate a *local* offspring population. The mating partners are chosen from the individuals' *local* neighborhood I_r. The best offspring solution is selected to replace the original individual I_i. Both phases of the algorithm are repeated as long as the termination criterion does not hold. In the algorithm two different neighborhood dynamics have been used:

Halpern's Deterministic Neighborhood Rule: The *deterministic* rules used to adapt the neighborhood *structures* follow the scheme of Halpern [Halp99]. The formation of the neighborhood depends directly on the fitness of the neighbors. Two types of rules are used:

- Decoupling rule:
  Each individual has a list of its neighbors that is sorted by fitness values. The connections to a certain percentage of least fit neighbors are detached. This number is determined by a parameter χ.

- Coupling rule:
  Simultanously to the generation of the list of neighbors used in the decoupling rule, a list of next-nearest neighbors is generated. According to this list, a percentage ω of new connections to the best next-nearest neighbors are established.

The fitness of each individual influences the structural development of its neighborhood: connections to individuals with a high fitness are preferred, hence, the number of their neighbors increases over time. Individuals that are least fit become more and more isolated. The rules allow that elements may become completely disconnected. The structure of the rules implies that once an individual is isolated this state is permanent. However, isolated individuals still share their contribution in the optimization process, because on our algorithm they still realize a $(1, \lambda)$-ES search.

Probabilistic Neighborhood Structure Adaptation Rule: According to the philosophy of CI, a system should adapt itself automatically to a problem. The coupling and decoupling rules introduced here take care of the fact that an individual should always have a certain number of neighbors. Furthermore, according to the evolutionary step size adaptation scheme, genetic interactions between individuals that proved to be good should have an effect on the next generations. Otherwise the interaction should be changed. Due to the discrete character of the connections the following rules have been introduced:

- Decoupling rule:
  A connection to a neighbor is detached, if the recombination with that neighbor yielded an offspring that has a worse fitness than the fitness of the best offspring individual generated at that site during the current generation.
- Coupling rule:
  If the number of connections of an individual falls below a certain threshold ϕ, ν new connections to randomly chosen individuals of the population are established.

Although isolated individuals cannot occur, isolated subpopulations are still possible. These isolations are not permanent (with high probability), because connections to any member of the population can be established during the coupling phase.

Communication Scheme The cellular population structure allows the storage of the data in a decentralized manner. The genetic code, the fitness, and the neighborhood structure is assigned to each individual. Communication happens directly between the individuals according to their local neighborhood scheme. A central master unit that controls the exchange of the data does not exist. Hence, the power of parallel communication can be used extensively. Although this scheme allows to implement an asynchronous evolutionary algorithm, a synchronized generation scheme has been used. Earlier analyses have shown [Meh94, Meh00] that asynchronous communication schemes imply complex dynamics to the evolutionary process that is not easy to control and a comparison with standard ES can become quite difficult.
Technically the individuals are gathered on single processor nodes. This allows to scale the number of individuals per processor according to the complexity of the problem and the performance of the computer. The changes in the communication structure during one generation cycle are transferred via broadcast functions that are especially designed for efficient communication.

Technical Details The software has been implemented on a parallel computer SGI Origin 2000 with 16 R10000 processors using a shared memory concept. The use of MPI (Message Passing Toolkit) [Pac97] offers the opportunity to run the code also on workstations clusters under LINUX or Solaris OS. The program has been implemented in C++.

4. Experiments

In order to characterize the effect of different parallelization schemes, four evolutionary approaches have been tested:

- standard ES (static, panmictic)
- parallel ES with total interconnectivity (static, panmictic)
- parallel ES using Halpern's communication scheme (dynamic, non-panmictic)
- parallel ES using the adaptive communication scheme (dynamic, non-panmictic)

The standard ES, which was exhaustively tested by Sprave [Spr99b], was used as a reference strategy. A direct comparison of the parallel approaches with a standard ES is not possible, because the genetic operators of the parallel and the standard strategy differ. However, in order to get an impression of the behavior of the parallel strategies, the results of the standard ES are cited. In order to compare a standard ES with its parallel counterpart, a virtually panmictic population has been modeled by a parallel ES using total interconnectivity between all individuals.

All strategies have been applied to a 30-dimensional sphere model

$$f(x) = \sum_{i=1}^{30} x_i^2 \tag{1}$$

and to the 10-dimensional Rastrigin's function

$$f(x) = 100 + \sum_{i=1}^{10} (x_i^2 - 10 \cdot cos(2 \cdot \pi \cdot x_i))\,. \tag{2}$$

The sphere model is a simple test function for which the behavior of a standard ES is well known. Rastrigin's function is a highly multimodal function that represents a big challenge for many optimization strategies. The global minimum of both functions is zero. A discussion of the application of a standard ES to these functions can be found in [Sch95].

The settings of the parameters used for the strategies were motivated by recommendations of Kursawe [Kur99] who empirically found optimal parameters for a standard ES applied to the problems above based on extensive numerical experiments.

parameter	settings
number of generations	200
number of individuals	100
number of offspring per individual	5 (sphere model) 2 (Rastrigin)
problem dimension	30 (sphere model) 10 (Rastrigin)
number of step sizes	1 (sphere model) n (Rastrigin)
initial object variables	[-2.0, ..., 2.0] (sphere model) [-5.0, ..., 5.0] (Rastrigin)
initial sigma values	0.01
recombination type (object variables)	intermedite
recombination type (step sizes)	discrete
τ_0	0.1826 (sphere model) 0.2236 (Rastrigin)
τ_i	not used (sphere model) 0.3976 (Rastrigin)
initial neighborhood structure	random
initial average number of neighbors of an individual	5

Table 1. Parameters for the structurally dynamic evolutions strategies

Each experiment has been repeated five times using the same parameter settings to get an impression of the variance of the results.

5. Results

Analyses of the Static ES Variants Figure 1 illustrates the behavior of a standard $(100, 500)$-ES and a parallel ES with a total interconnection communication scheme. Both strategies are panmictic and do not use a dynamic neighborhood structure. The difference lies in the genetic operators. The parallel algorithm produces within each local environment five offspring elements per individual and replaces the solution of the specific site by the best solution found in the local offspring population.

Both strategies show the typical linear (logarithmically scaled) convergence behavior of an evolution strategy when applied to a sphere model. The step sizes follow the progression of the fitness. In this case this indicates a correct automatic step size adaptation.

The convergence velocity of the parallel ES is a little bit smaller than

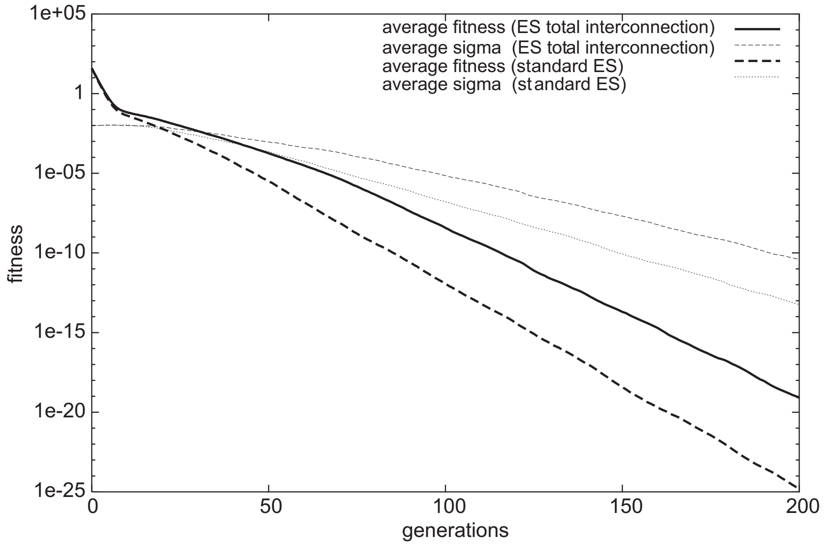

Figure 1. Comparison of a standard ES and an ES with a total interconnection scheme (applied to the sphere model).

the velocity of the standard ES. This is due to the local selection scheme of the parallel algorithm. However, qualitatively both strategies behave quite similar. One can state that the basic neighborhood structure used for the parallel algorithms does not influence the basic behavior of the optimization process.

Analyses of the Dynamic ES Variants The dynamics of the evolutionary algorithm on self-organizing dynamic lattices [Hal99] can be controlled by specific model parameters. The deterministic rules of Halpern are influenced by the coupling parameter χ and the decoupling parameter ω. The probabilistic strategy is controlled by the threshold values ϕ and ν. In the following experiments the behavior of the strategies is discussed.

a.) Dynamic ES using Halpern's neighborhood rules

Figure 2 illustrates the typical progression of the number of neighbors during the evolutionary optimization process for $\chi = 0.4$ and $\omega = 0.4$. Depending on the definition of χ and ω and the population size ν, the number of direct neighbors Γ and next-nearest neighbors Δ stabilizes after an initial oscillation phase to the manually estimated values $\Gamma = \frac{\chi \cdot \mu}{\chi + \omega}$ and $\Delta = \frac{\omega \cdot \mu}{\chi + \omega}$, respectively.

Due to the effect of isolation, the total number of neighbors and

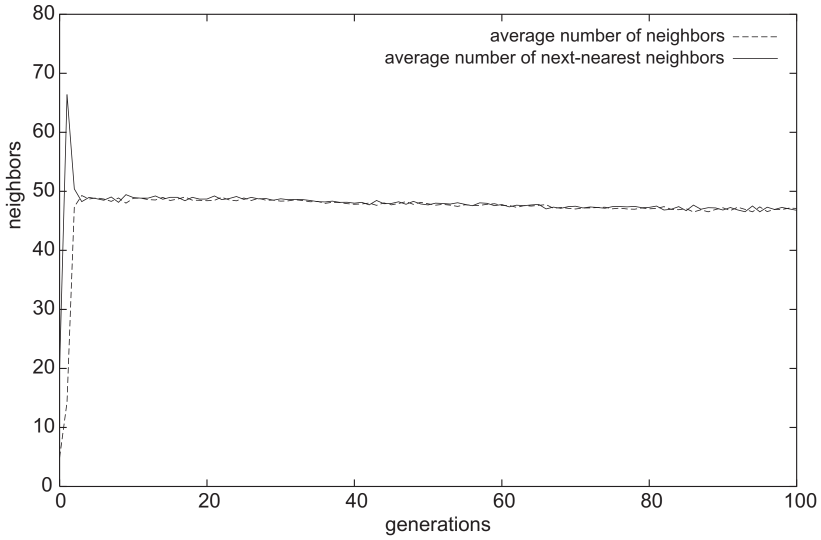

Figure 2. Number of direct neighbors and next-nearest neighbors (Halpern's rules, Rastrigin's function).

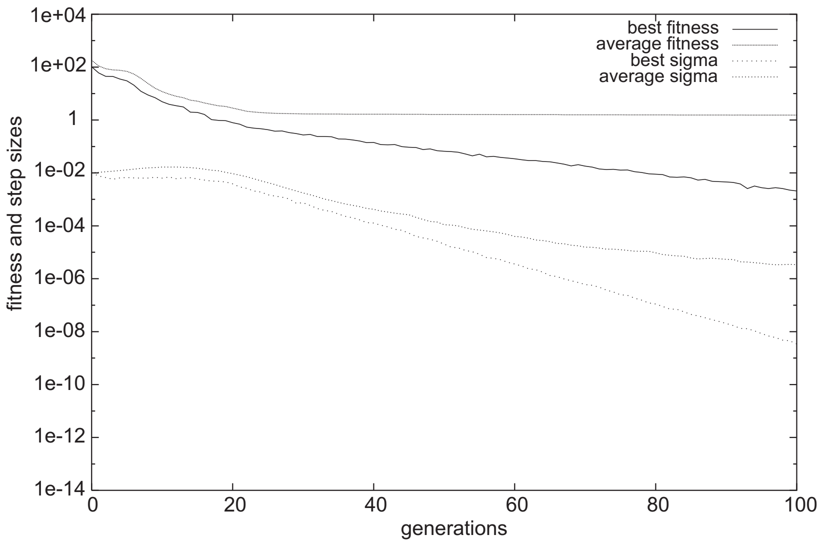

Figure 3. Fitness and step sizes of the dynamic ES with Halpern's neighborhood rules (applied to Rastrigin's function).

next-nearest neighbors decreases over time. This effect is the more prominent the larger χ becomes relative to ν. Once an individual is isolated, it cannot establish new connections to other individuals.

Hence, the total number of contacts within the population decreases over time.

Figure 3 shows the typical convergence behavior of the dynamic evolution strategy using Halpern's rules when applied to Rastrigin's function. Although the fitness function is quite complex, the optimization process shows no phases of stagnation. The convergence speed of this parallel evolutionary algorithm when applied to the sphere model is, of course, much higher. Nevertheless, the algorithm shows nearly linear convergence on both fitness functions.

The application of the dynamic ES with Halpern's neighborhood rules to Rastrigin's function shows the highest convergence velocity with the parameter settings $\chi = 0.4$ and $\omega = 0.4$. A high coupling value ω speeds up the initial stabilization process of the connectivity of the population. A stable average connectivity rate is an important precondition for a stable progression of the optimization process.

The analysis of the neighborhood structure shows that isolated subpopulations having more than one individual did not appear. This was true for both fitness functions.

b.) Dynamic ES using Probabilistic Neighborhood Rules

The following experiments illustrate the dynamics of the connectivity of the population and the convergence behavior of the dynamic ES with probabilistic neighborhood rules.

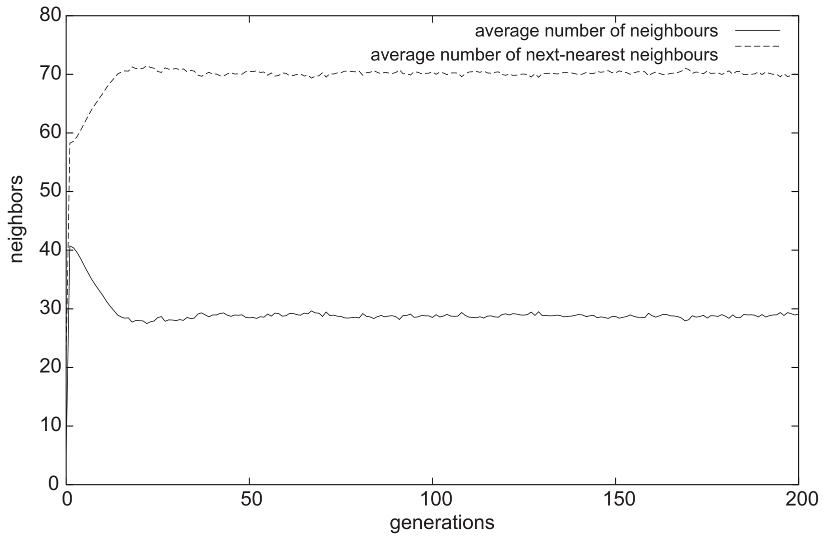

Figure 4. Number of direct neighbors and next-nearest neighbors.

Figure 4 shows the dynamics of the connectivity in the population of the dynamic ES when applied to Rastrigin's function. Here, the parameter setting $\phi = 20$ and $\nu = 20$ are used. In the beginning, like in Halpern's scheme, an initial stabilizing phase appears. After this period, a constant and non-decreasing number of connections in the population is established. This is due to the theshold value ϕ that prevents the phenomenon of isolation. The average number of neighbors of an individual can be estimated by $\Upsilon = \frac{2\phi+\nu}{2}$.

The analyses of different parameter settings yielded that $\phi = 20$ and $\nu = 20$ is a good choice. Smaller values reduced the convergence velocity of the algorithm, because the necessary genetic diversity that depends on the number of neighbors cannot be established. High parameter settings yielded neighborhood structures that more and more resemble completely interconnected populations that cannot use synergetic effects due to temporal genetic separation.

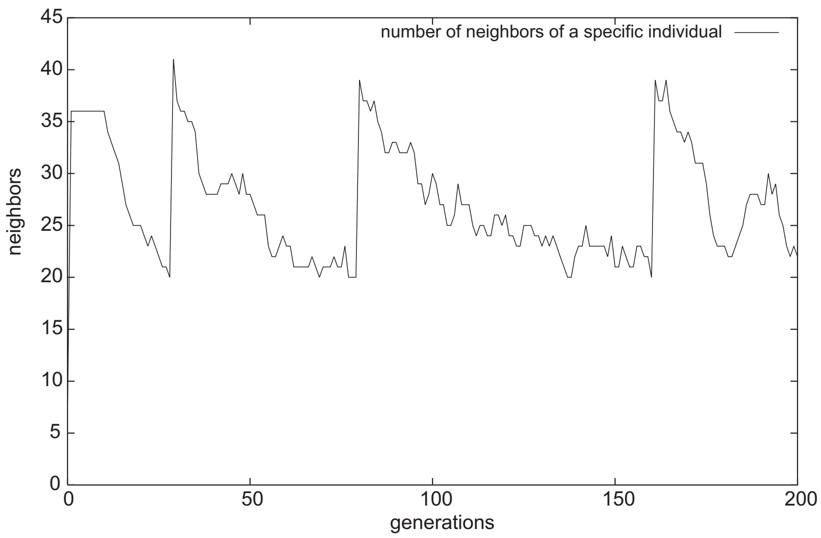

Figure 5. Number of direct neighbors using an probabilistic scheme.

Figure 5 shows the progress of the number of neighbors of one randomly selected individual during the optimization process. The typical degeneration and re-establishing phases of the connectivities can be seen. On the average, the number of connections in the population is nearly constant (see figure 4).

The convergence behavior of the dynamic ES with probabilistic neighborhood rules applied to Rastrigin's function is shown in Figure 6. The continuous reduction of the step sizes indicates that the step size are adapted in the typical way of an ES. The continuous decrease of the fitness values is a hint for a robust optimization process.

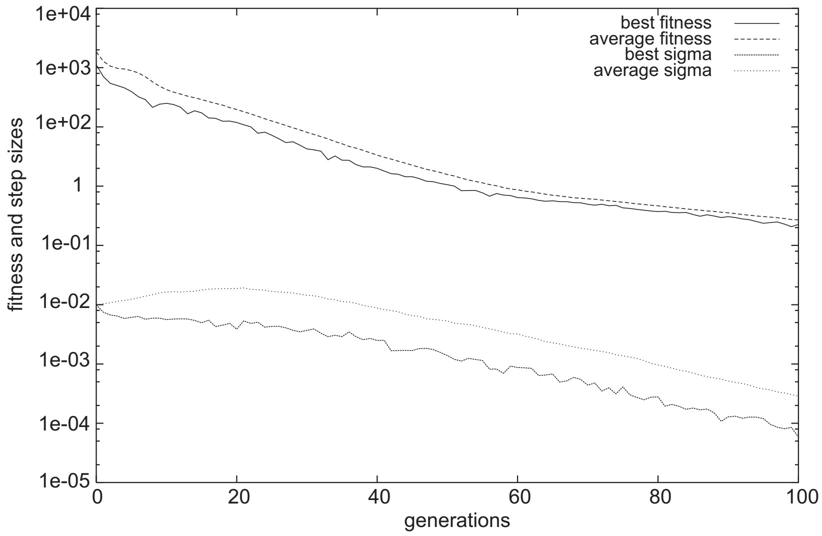

Figure 6. Fitness and step sizes of the dynamic ES with probabilistic neighborhood rules.

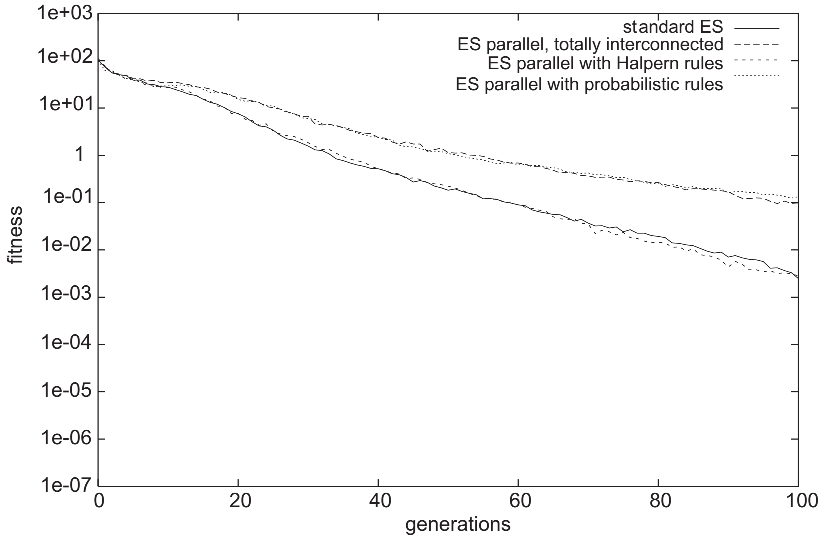

Figure 7. Comparison of the parallel ES variants.

Comparison of the ES Variants In order to compare the convergence velocity of the ES variants, a standard $(100, 500)$-ES, a parallel ES using a total interconnection communication scheme, and the two dynamic and parallel ES with either Halpern's or probabilistic neighborhood adaptation rules were used. The parameter settings of the previously discussed experiments were applied.

The parallel ES with Halpern's neighborhood rule shows the best convergence velocity when applied to Rastrigin's function (see fig. reffig-7). An additional advantage of the algorithm lies in its processing speed because the fitness functions can be evaluated in parallel. A disadvantage of Halpern's rules is its tendency to generate isolated individuals, that prevents the exploitation of the full synergetic effect of parallelsim. It should be noted, that the algorithm discussed here only uses Halpern's neighborhood rules while the genetic operators are taken from the standard ES.

The dynamic ES with the probabilistic neighborhood adaption rules showed a convercence velocity that is similar to the algorithm with totally interconnected populations.

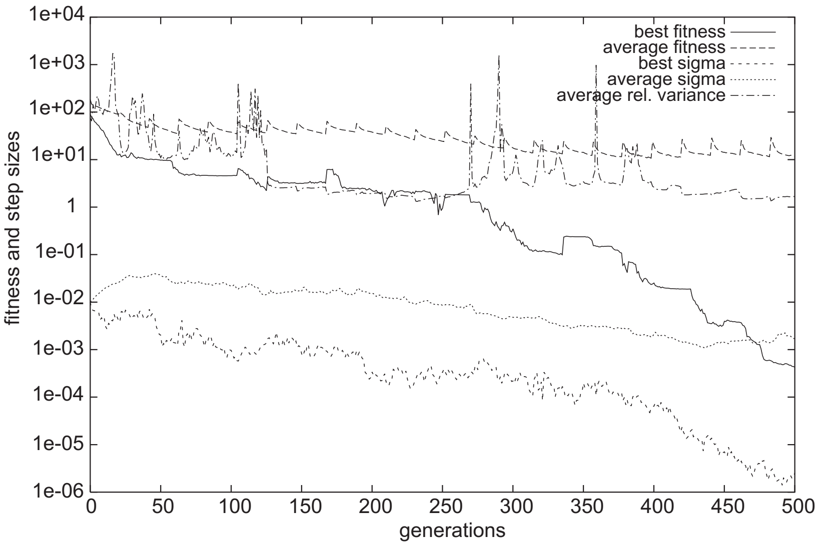

Figure 8. Dynamic multipopulation ES with probabilistic communication.

An interesting improvement of the convergence probability for Rastrigin's function can be reached when the ES with probabilistic communication is modified in a way that temporarily isolated subpopulations (clusters) can appear. Therefore, the parameter settings $\phi = 1$ and $\nu = 1$ are used. The extremely low number of connections yields subpopulations that are isolated for certain time periods. The probabilistic connection scheme allows to interconnect any individual with each other. Hence, a sort of *dynamic* stepping stone model with changing subpopulation structures is generated. An additional damping factor, that "freezes" the communication structure for a period of γ generations, reduces the "turbolences" induced by communication and, hence, increases the possibility to evolve good local solutions, that can be exchanged with the neighbors. This improves the genetic diversity and therefore the convergence probability but it also hampers the convergence velocity. Especially for difficult multimodal functions a robust optimization may be more necessary than a fast but 'greedy' strategy. The behavior of the parallel dynamic multipopulation strategy with probabilistic communication scheme is shown in Figure 8.

Speed Up The main goal in parallelizing an algorithm is to improve its performance. This means for evolutionary algorithms to speed up the optimization process by parallel execution of the genetic operators and/or the fitness function. Synergetic effects gained by parallelizing often allow an additional improvement of the robustness and the convergence probability of the optimization process.

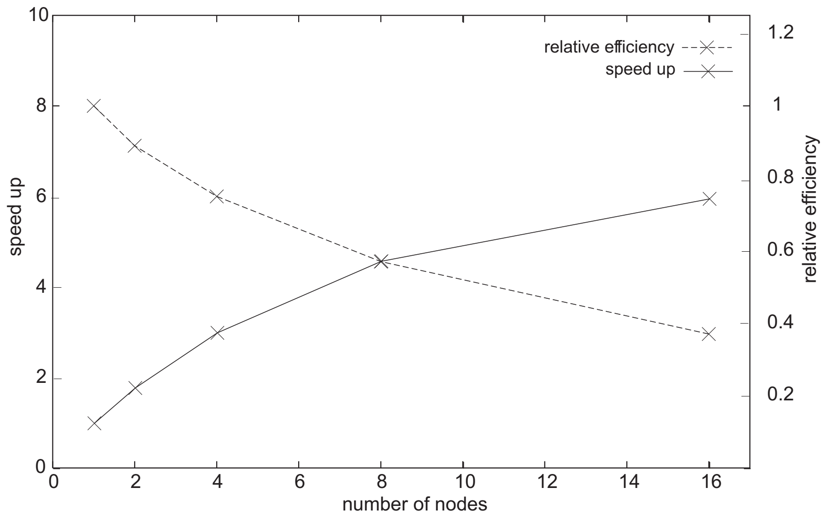

Figure 9. Relative speed up and relative efficiency.

The parallel ES has been implemented on a parallel computer architecture with 16 processors. Therefore, the speed up of the dynamic parallel ES variants can be measured quantitatively.

The execution time of a parallel program is the length of the time interval from the start of the first process to the end of the last process [Bur97]. The *relative speed up s(p)* is the quotient of the time t_1 needed by one processor to execute the parallel program to time t_p needed by p processors, i.e. $s(p) = t_1/t_p$. The *relative efficiency e(p)* is the quotient of the relative speed up $s(p)$ and the number of processors p. This quotient $e(p) = s(p)/p$ expresses the effective gain of the parallelization. In order to keep the experimental conditions constant, each experiment was run with 1, 2, 4, 8, 16 processors using constant population sizes (i.e., 208 individuals were used for one processor or 104 individuals each for two processors, and so forth).

Figure 9 shows that an increasing number of nodes yields a sub-linear relative speed up. The relative efficiency decreases because the ammount of communication becomes worse the more processors are used. Comparing the speed up and the effectiveness allows the conclusion that there exists a certain number of processors that marks a limit of efficiency (not reached here) gained by using additional processors.

6. Conclusions and Outlook

In this article two parallel implementations of Evolution Strategies using dynamic communication structures have been discussed. Both approaches use the standard genetic operators from conventional ES. This allows to utilize the benefit of automatic step size adaptation via mutation and recombination. The basic population structure is a diffusion model. Motivated by the ideas of Halpern [Hal99], self-organized dynamic lattices are used. Following either deterministic or probabilistic rules, each individual is able to connect or disconnect in its environment. Both rules use subpopulations of neighbors around each individual to produce offspring. The best individuals of the offspring population are selected to replace the parent individual. Connection and disconnection depend on the fitness of the neighbors or the fitness gain of the connection after reproduction, respectively. The deterministic rules tend to reduce the connections over time while the probabilistic rules re-establish new connections in order prevent isolation.
Compared with a reference model of a parallel ES with totally interconnected individuals, the parallel ES with deterministic communication rules has a much higher convergence velocity. This may be due to the fact that in the deterministic model connections to individuals with low fitness values are deleted and, hence, the probability to meet better individuals for recombinations is higher than in the panmictic case. The probabilistic re-establishing of the connections leads to a stable number of neighbors but does not consider the fitness of new neighbors. This yields a convergence velocity that is similar to the reference model.
Using a parallel ES with probabilistic communication rules that do not destroy links between individuals for some generations and that allow temporarily isolated subpopulations yielded a quite slow strategy with a – compared to the other strategies – surpassing convergence probability.
The application of a parallel computer with 16 processors allowed a significant speed-up of the ES. The experimental measured increase of

speed-up per processor follows a curve with positive but slowly decreasing derivatives. The amount of communication between the processors limits the gain from parallelization. The synergetic effects in parallel ES play a more important role. This is expressed by an increase of the convergence probability and a little higher convergence velocity of the parallel optimization algorithms when applied to multimodal problems.

The experience from the experiments discussed here will be used for practical applications in the field of evolutionary surface reconstruction [Meh00]. First tests already show very promizing results.

Acknowledgments

This project is part of the collaborative research center SFB 531 “Design and Management of Complex Technical Processes and Systems by Means of Computational Intelligence Methods” that is funded by the Deutsche Forschungsgemeinschaft. We would also like to thank Winfried Thieme, who did a lot of important basic work and experiments.

References

[Bez94] C. Bezdek *What is computational intelligence?*, pp. 1-12, In: J. M. Zurada, R. J. Marks II und C. J. Robinson (eds), Computational Intelligence: Imitating Life. IEEE Press, New York, NY, 1994.

[Bur97] H. Burkhart, *Parallele Programmierung*, Informatik - Handbuch, P. Rechenberg and G. Pomberger (eds), Hanser Verlag, München, 1997.

[Coh87] J. P. Cohoon, S. U. Hedge, W. N. Martin, and D. S. Richards *Punctuated Equilibria: A parallel genetic algorithm*, Proc. Second Int'l Conf. Genetic Algorithms and their Applications, J. J. Grefenstette (ed.), pp. 148-154, Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[Gor89] M. Gorges-Schleuter, *ASPARAGOS An asynchronous parallel genetic optimization strategy*, Proc. 3rd ICGA, J. D. Schaffer (ed.), Morgan Kaufmann Publishers, San Mateo, 1989.

[Gor91] M. Gorges-Schleuter, *Explicit Parallelism of GAs through Population Structures*, Proc. PPSN 1, LNCS 496, H.-P. Schwefel and R. Männer (Eds.), Springer Verlag, 1991.

[Hal99] P. Halpern, *Evolutionary Algorithms on a Self-Organized dynamic Lattice*,University of the Sciences in Philadelphia, 1999.

[Kur99] F. Kursawe, *Grundlegende empirische Untersuchungen der Parameter von Evolutionsstrategien - Metastrategien*, Doctoral Dissertation, University of Dortmund, Department of Computer Science, 1999.

[Mar70] T. Maruyama, *On the role of decrease of heterozygoty in circular stepping stone models*, Theor. Publ. Biol. 1, 101-119, 1970.

[Meh94] J. Mehnen, *Asynchronous Parallel Evolution Strategies*, Diploma Thesis, University of Dortmund, 1994.

[Meh00] J. Mehnen, *Evolutionäre Flächenrekonstruktion*, Ph.D. thesis, University of Dortmund, Germany, published at the Vulkan Verlag, Essen, K. Weinert (Eds.) 2000.

[Pac97] P. S. Pacheco, *Parallel Programming with MPI*, Morgan Kaufmann Publishers, San Francisco, 1997.

[Rau00] T. Rauber and G. Rünger *Parallele und verteilte Programmierung*, Springer, Berlin, 2000.

[Rech94] I. Rechenberg, *Evolutionsstrategie '94*, Frommann-Holzboog, Stuttgart, 1994.

[Rud00] G. Rudolph, *Evolution Strategies*, Evolutionary Computation 1 - Basic Algorithms and Operators, T. Bäck, D. B. Fogel and Z. Michalewicz (Eds.), pp. 81-88, Institute of Physics Publishing, Bristol, 2000.

[Sch95] H.-P. Schwefel, *Evolution and Optimum Seeking*, Wiley Interscience, New York, 1995.

[Spr99] J. Sprave *Ein einheitliches Modell für Populationsstrukturen in evolutionären Algorithmen*, Ph.D. thesis, University of Dortmund, 1999.

[Spr99b] J. Sprave *Ease - Evolutionary Algorithms Scripting Environment*, Reihe CI 54/99, SFB 531, University of Dortmund, 1999.