

# Dynamic Neighborhood Structures in Parallel Evolution Strategies

**Klaus Weinert\***

**Jörn Mehnen\***

*Department of Machining Technology,  
University of Dortmund,  
44221 Dortmund, Germany*

**Günter Rudolph†**

*Department of Computer Science,  
University of Dortmund,  
44221 Dortmund, Germany*

---

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 robustly and quickly on a multimodal test function.

---

## 1. Introduction

---

Evolutionary algorithms (EA) 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 such as 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 EA.

The biological principle of evolution is inherently parallel. In nature, genetic interaction and selection of the fittest individuals happen

---

\*Electronic mail addresses: {weinert,mehnen}@isf.mb.uni-dortmund.de.

†Electronic mail address: rudolph@LS11.cs.uni-dortmund.de.

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 self-organizing processes that allow perpetuation of a high genetic variety in order to keep the species from becoming extinct. According to these observations, the analyses of EA with parallel evolution and dynamically interacting populations are 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 are 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) [1], which also comprises the fields of fuzzy logic and neural networks. These techniques are subsymbolic (numeric) and excel by their adaptability, fault tolerance, and high processing speeds when applied to complex problems. CI methods are often inherently parallel. Here, parallelization not only helps to increase performance of these strategies, the usage of structure and the distributed processing of data also introduces a new quality to these systems. In this article the focus will be on parallel EA.

### 2.1 Static population structures

Parallel EA with static population structures can be grouped as follows [2].

- *Panmictic model.* The population is not explicitly 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 number of individuals can migrate between the subpopulations on predetermined paths. 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, that is, 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 multipopulation models where data can migrate between any subpopulation [3]. Stepping stone models follow a more restricted communication scheme, that is, the subpopulations are arranged on a ring [2], or on a torus or grid [4].

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 does depend 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, such as 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, such as Cohoon’s approach of punctuated equilibria [5] or Rechenberg’s nested evolution strategies [6, 7], also show that structured populations have advantages when compared to panmictic approaches.

**Neighborhood models.** Neighborhood models are sometimes also called “diffusion models.” Due to the grid structure typically used, data spreads slowly through the population. Genes that have a good fitness are transferred from neighbor to neighbor and, thus, “infect” the population by a diffusion process.

Gorges-Schleuter in [8, 9] and Sprave in [10] introduced algorithms that were also implemented on parallel computer hardware. They show that a diffusion model applied to complex problems, such as the traveling 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 in [3] introduced an EA 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 and 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. At 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 are 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 ES running on a diffusion model. Great differences can be seen in the way mutation, recombination, and population replacement after selection is used. Like in GA, 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 strategy with dynamic neighborhood structures

The paragon for the alternative parallel optimization algorithm discussed in this section is the ES [6, 11]. The algorithm has the following structure.

```

-  $t := 0$ 
- 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  $\lambda$  offspring from the parents  $I_i$  and  $I_r$ 
    - mutate the  $\lambda$  offspring (object variables and step sizes)

```

- evaluate  $\lambda$  offspring by application of  $f$
- select the best individual from the  $\lambda$  offspring to replace individual  $I_i$  of population  $P^{t+1}$

During the initialization phase a beginning population  $P^0$  and an adjacency matrix  $L^0$  are used.  $L_i^0$  characterizes the local neighborhood of individual  $i = 1, \dots, N$ . Typically the neighbors are chosen from a set of randomly selected individuals. The size of the set fluctuates around a predetermined average value. Hence, isolated individuals or isolated subgraphs are 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 ES, namely recombination and mutation, are applied to each individual  $\lambda$  times in order to generate a local offspring population. The mating partners are chosen from the local neighborhood  $I_r$  of the individuals. 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. Two different neighborhood dynamics have been used in the algorithm.

**Halpern's deterministic neighborhood rule.** The deterministic rules used to adapt the neighborhood structures follow the scheme of Halpern in [3]. 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  $\chi$ .
- *Coupling rule.* Simultaneous with 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  $\omega$  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 in our algorithm they still realize a  $(1, \lambda)$ -ES search [11].

**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 succeeding 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 recombination with that neighbor yields an offspring that has a lower 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  $\phi$ ,  $\nu$  new connections to randomly chosen individuals of the population are established.

Although isolated individuals cannot occur, isolated subpopulations are still possible. These isolated structures 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, fitness, and neighborhood structure are 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 implementing an asynchronous EA, a synchronized generation scheme has been used. Earlier analyses have shown [12, 13] that asynchronous communication schemes imply complex dynamics in 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 scaling the number of individuals per processor according to the complexity of the problem and the performance of the computer. 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) [14] offers the opportunity of running the code on clusters of workstations 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, nonpanmictic).
- Parallel ES using the adaptive communication scheme (dynamic, nonpanmictic).

The standard ES, which was exhaustively tested by Sprave in [10], 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

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

and to the 10-dimensional Rastrigin's function

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

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 [11].

The settings of the parameters used for the strategies were motivated by recommendations of Kursawe in [15] where optimal parameters for a standard ES are empirically found and applied to the problems above based on extensive numerical experiments (see Table 1).

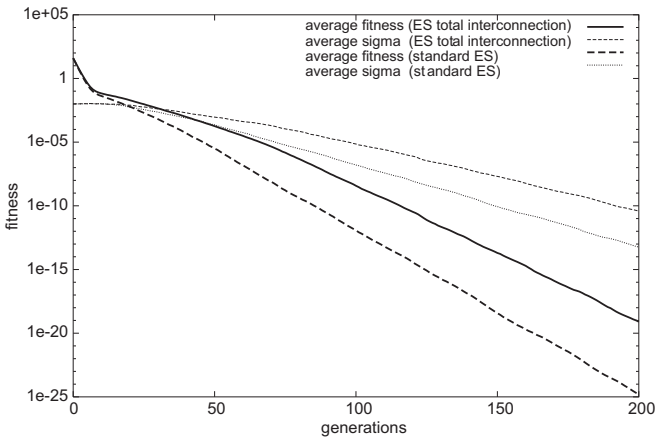
Since our previous numerical study had shown only small fluctuations in the results, we dared to base the series of experiments presented here on five independent repetitions per experiment.

## 5. Results

**Analyses of the static ES variants.** Figure 1 illustrates the behavior of a standard (100, 500)-ES [11] and a parallel ES with a total interconnectivity

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)	intermediate
recombination type (step sizes)	discrete
$\tau_0$	0.1826 (sphere model) 0.2236 (Rastrigin)
$\tau_i$	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 ES.



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



tion 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 five offspring elements per individual within each local environment 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 ES when applied to a sphere model. The step sizes follow the progression of the fitness. In this case a correct automatic step size adaptation is indicated.

The convergence velocity of the parallel ES is a little bit smaller than 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 EA on self-organizing dynamic lattices [3] can be controlled by specific model parameters. The deterministic rules of Halpern are influenced by the coupling parameter  $\chi$  and the decoupling parameter  $\omega$ . The probabilistic strategy is controlled by the threshold values  $\phi$  and  $\nu$ . In the following experiments the behavior of the strategies is discussed.

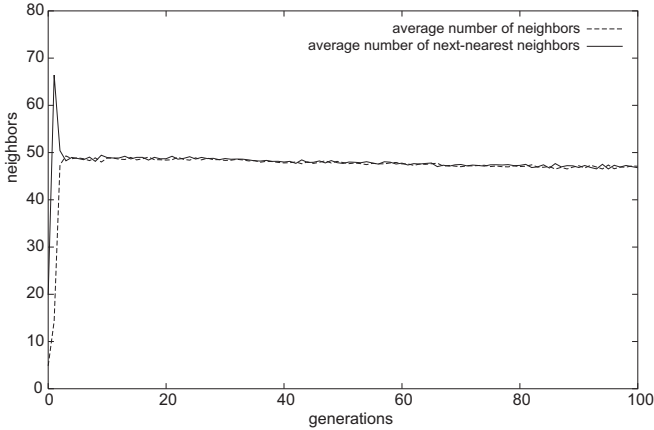
### ■ 5.1 Dynamic evolution strategy using Halpern's neighborhood rules

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  $\chi$  and  $\omega$  and the population size  $\nu$ , the number of direct neighbors  $\Gamma$  and next-nearest neighbors  $\Delta$  stabilizes after an initial oscillation phase to the manually estimated values  $\Gamma = \chi \cdot \mu / (\chi + \omega)$  and  $\Delta = \omega \cdot \mu / (\chi + \omega)$ , respectively.

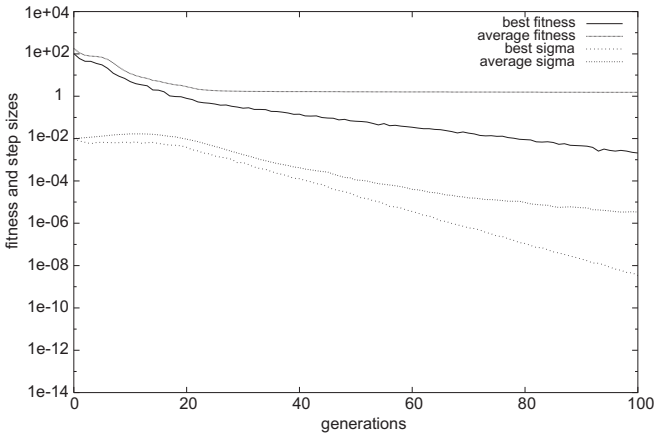
Due to the effect of isolation, the total number of neighbors and next-nearest neighbors decreases over time. This effect becomes more prominent the larger  $\chi$  is relative to  $\nu$ . Once an individual is isolated, it cannot establish new connections to other individuals. Hence, the total number of contacts within the population decreases over time.

Figure 3 shows the typical convergence behavior of the dynamic ES 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 EA 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  $\omega$  speeds up the initial stabilization process of the connectivity of the population.



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



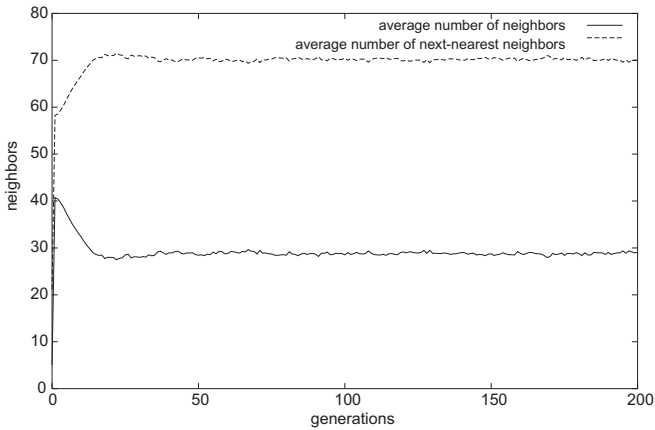
**Figure 3.** Fitness and step sizes of the dynamic ES with Halpern's neighborhood rules (applied to Rastrigin's function).

A stable average connectivity rate is an important precondition for a stable progression of the optimization process.

An analysis of the neighborhood structure showed that isolated sub-populations having more than one individual did not appear. This was true for both fitness functions.

## 5.2 Dynamic evolution strategy 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.

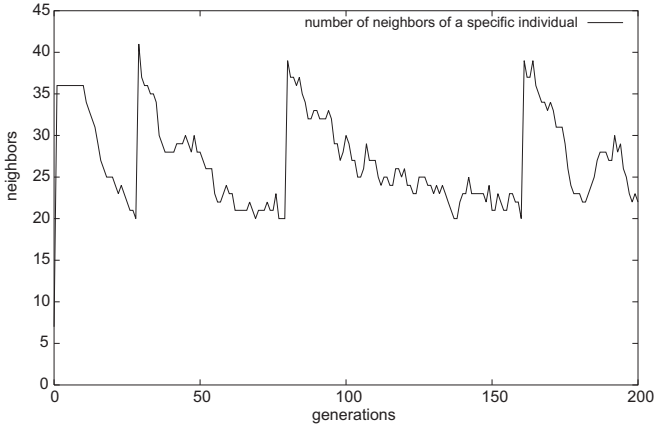
Figure 4 shows the dynamics of the connectivity in the population of the dynamic ES when applied to Rastrigin's function. Here, the parameter settings  $\phi = 20$  and  $\nu = 20$  are used. In the beginning, like in Halpern's scheme, an initial stabilizing phase appears. After this period, a constant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

The analyses of different parameter settings yielded that  $\phi = 20$  and  $\nu = 20$  are good choices. 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.

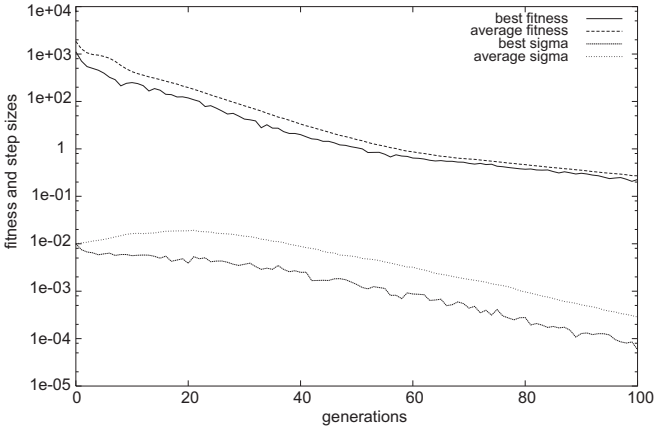
Figure 5 shows the progress of the number of neighbors of one randomly selected individual during the optimization process. The typical degeneration and re-establishing phases of the connectivities can be seen. On the average, the number of connections in the population is nearly constant (see Figure 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 size indicates that the step sizes are adapted in the typical way of an ES. The continuous decrease in the fitness values is a hint for a robust optimization process.

**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



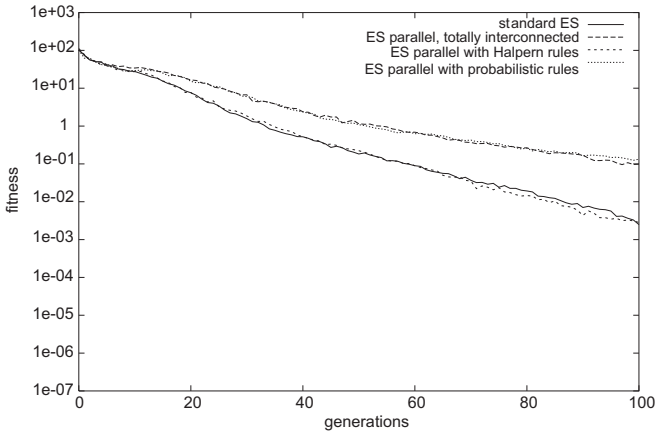
**Figure 5.** Number of direct neighbors using a probabilistic scheme.



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

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 Figure 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 prevent 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.

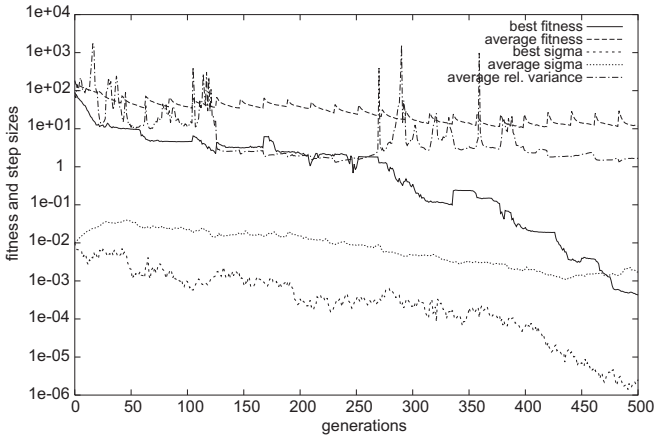


**Figure 7.** Comparison of the parallel ES variants.

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

An interesting improvement of the convergence probability for Rastrigin's function can be reached when the ES with probabilistic communication is modified in such 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 interconnecting any individual with any 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  $\gamma$  generations, reduces the "turbulences" induced by communication and, hence, increases the possibility of evolving 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 a probabilistic communication scheme is shown in Figure 8.

**Speed up.** The main goal in parallelizing an algorithm is to improve its performance. For EA, this means 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 8.** Dynamic multipopulation ES with probabilistic communication.

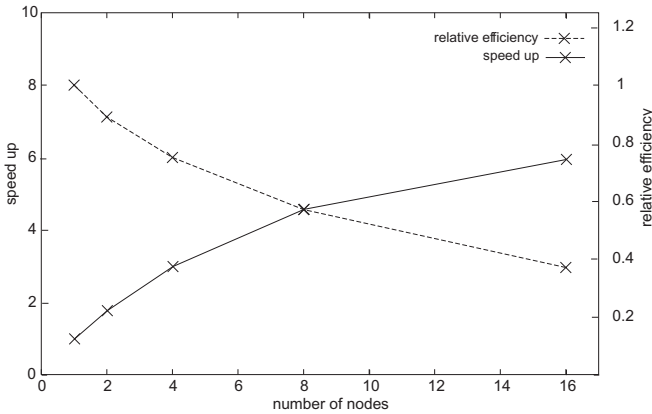
The parallel ES has been implemented on a parallel computer architecture with 16 processors. Therefore, the speed up of the dynamic parallel ES variants can be measured quantitatively.

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 [16]. 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, that is,  $s(p) = t_1/t_p$ . 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, and 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 amount of communication becomes larger as 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 (ES) using dynamic communication structures have been discussed. Both approaches use the standard genetic operators from conventional ES. This allows utilizing the benefit of an automatic step size adaptation *via* mutation and recombination. The basic population structure is



**Figure 9.** Relative speed up and relative efficiency.

a diffusion model. Motivated by the ideas of Halpern in [3], self-organized dynamic lattices are used. Following either deterministic or probabilistic rules, each individual is able to connect or disconnect within 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 and 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 of meeting better individuals for recombination is higher than in the panmictic case. The probabilistic re-establishing of the connections leads to a stable number of neighbors but disregards 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 slow strategy with a surpassing convergence probability, when compared to the other strategies.

The application of a parallel computer with 16 processors allowed a significant speed up of the ES. The experimentally 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 in the convergence probability and a little higher convergence velocity for 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 [13]. First tests already show very promising 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

- [1] C. Bezdek, “What is Computational Intelligence?” in *Computational Intelligence: Imitating Life*, edited by J. M. Zurada, R. J. Marks II, and C. J. Robinson (IEEE Press, New York, 1994).
- [2] J. Sprave, “A Unified Model of Non-Panmictic Population Structures in Evolutionary Algorithms,” in *Proceedings 1999, Congress on Evolutionary Computation (CEC’99)*, volume 2, Washington D.C., July 6–9, 1999, edited by P. J. Angeline and V. W. Porto (IEEE Press, Piscataway, NJ, 1999).
- [3] P. Halpern, *Evolutionary Algorithms on a Self-Organized Dynamic Lattice* (University of the Sciences in Philadelphia, 1999).
- [4] T. Maruyama, “On the Role of Decrease of Heterozygosity in Circular Stepping Stone Models,” *Theor. Publ. Biol.*, 1 (1970) 101–119.
- [5] J. P. Cohoon, S. U. Hedge, W. N. Martin, and D. S. Richards, “Punctuated Equilibria: A Parallel Genetic Algorithm,” in *Proceedings of the Second International Conference on Genetic Algorithms and their Applications*, edited by J. J. Grefenstette (Lawrence Erlbaum Associates, Hillsdale, NJ, 1987).
- [6] I. Rechenberg, *Evolutionsstrategie ’94* (Frommann-Holzboog, Stuttgart, 1994).
- [7] G. Rudolph, “Evolution Strategies,” in *Evolutionary Computation One—Basic Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz (Institute of Physics Publishing, Bristol, 2000).
- [8] M. Gorges-Schleuter, “ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategy,” *Proceedings of the Third ICGA*, edited by J. D. Schaffer (Morgan Kaufmann Publishers, San Mateo, 1989).



- [9] M. Gorges-Schleuter, “Explicit Parallelism of GAs through Population Structures,” *Proceedings PPSN 1, LNCS 496*, edited by H.-P. Schwefel and R. Männer (Springer-Verlag, 1991).
- [10] J. Sprave, “Ease—Evolutionary Algorithms Scripting Environment,” *Technical Report, CI 54/99*, ISSN 1433-3325 (University of Dortmund, 1999).
- [11] H.-P. Schwefel, *Evolution and Optimum Seeking* (Wiley Interscience, New York, 1995).
- [12] J. Mehnen, *Asynchronous Parallel Evolution Strategies*, Diploma Thesis, University of Dortmund, 1994.
- [13] J. Mehnen, *Evolutionäre Flächenrekonstruktion*, Ph.D. Thesis, University of Dortmund, Germany, published at the Vulkan Verlag, Essen, edited by K. Weinert, 2000.
- [14] P. S. Pacheco, *Parallel Programming with MPI* (Morgan Kaufmann Publishers, San Francisco, 1997).
- [15] F. Kursawe, *Grundlegende empirische Untersuchungen der Parameter von Evolutionsstrategien - Metastrategien*, Ph.D. thesis, University of Dortmund, Department of Computer Science, 1999.
- [16] H. Burkhardt, *Parallele Programmierung*, Informatik - Handbuch, edited by P. Rechenberg and G. Pomberger (Hanser Verlag, München, 1997).