Significance of Locality and Selection Pressure in the Grand Deluge Evolutionary Algorithm

Günter Rudolph
Informatik Centrum Dortmund e.V.
Joseph-von-Fraunhofer-Str. 20
D–44227 Dortmund
Rudolph@LS11.Informatik.Uni-Dortmund.de

Joachim Sprave
Universität Dortmund
Fachbereich Informatik XI
D–44221 Dortmund
Sprave@LS11.Informatik.Uni-Dortmund.de

Abstract

This paper presents the results of a parameter study of the Grand Deluge Evolutionary Algorithm, whose special features consist of local interactions between individuals within a spatially structured population and a self–adjusting control mechanism of the selection pressure. Since both ingrediences are parametrizable this study aims at the identification of the significance and sensitivity of the parameter settings with regard to the performance of the algorithm, especially under the transition from one– to two–dimensional neighborhood patterns.

1 Introduction

In [10] we presented the Grand Deluge Evolutionary Algorithm (GDEA), which combines the traditional proportionate selection operator with a self–organizing acceptance threshold schedule. The population of the GDEA possesses a spatial structure to allow scalable parallel implementations, which means that the individuals are distributed over the vertices of a connected graph and that the genetic operators are applied locally in some neighborhood of each individual. This algorithm was embedded in the framework of *probabilistic automata networks* and could be proven to be globally convergent with probability one under the assumption that the genotypes of the individuals are binary strings. The parameter study made in [10] employed a multiple knapsack problem as objective function and investigated the significance of the parameters with regard to performance by varying the delay of adjusting the threshold values (selection pressure) and the neighborhood size (locality) in a ring topology, i.e., with one–dimensional neighborhood structures. While the overall performance of the GDEA was great compared to a traditional genetic algorithm (GA), the results were disappointing with respect to parallelism, where small neighborhood sizes are preferred to obtain low communication requirements. But instead the parameter study resulted in a relatively large optimal neighborhood size of about 40 individuals. Since the ring topology is only one possible implementation of the GDEA, the next step was to run the same experiments with a different topology. We chose a toroid grid for two reasons: it seems to be the most natural extension of a topology just to increase the dimension, and the torus is the most popular structure for parallel implementations at all.

A description of the GDEA is given in section 2, with emphasis on the design of local reproduction operators for individuals distributed over the vertices of a connected graph and the realization of a self–adjusting threshold control. Section 3 first presents our selection of neighborhood structures and test problems, before the results of the parameter study are discussed. Finally, we draw some conclusions in section 4.

2 Description of the Algorithm

It is assumed that the reader is familiar with the basics of evolutionary algorithms (EA). For a recent comprehensive overview see the monograph by Bäck [1]. The genetic operators of the GDEA for individuals with binary genotype are based on those of the traditional GA as described by Goldberg [4]. Since mutation and crossover remain unchanged they are not explicitly defined here. The changes only affect reproduction and offspring acceptance.

2.1 Local Reproduction

Since all individuals in a population compete with each other for the chance to produce offspring, a traditional EA needs to know the fitnesses of all individuals during the reproduction phase of the algorithm. This kind of global knowledge makes an algorithm unsuitable for an efficient parallel implementation. Therefore, most parallel implementations of EAs base on local reproduction rules [7, 5, 12, 8, 11] which can be applied simultaneously to smaller subsets of the population.

In order to be comparable to a standard GA, in [10] a localized proportionate selection was defined for a ring topology. In the following a more general definition is given which does not even depend on homogeneous neighborhood structures.

Let be n the population size, ℓ the dimension of the search space, $P^t = \{x_i^t \in \mathbb{B}^\ell : 0 \leq i < n\}$ the population at generation t, and $\mathcal{N}_\nu \in \mathcal{P}\{0, \ldots, n-1\}$ a set of indices defining the neighborhood $\mathcal{N}_\nu$ of the individual x_ν^t. $\mathcal{N}_\nu$ is a family (not a set) consisting of all x_k^t with $k \in \mathcal{N}_\nu$. The fitness function $F : \mathbb{B}^\ell \mapsto \mathbb{R}^+$ is normally the result of windowing and scaling techniques applied to the objective function. If the search space of the objective function is $D \neq \mathbb{B}^\ell$, e.g. $D \subseteq \mathbb{R}^N$, a mapping function $m : \mathbb{B}^\ell \mapsto D$ must be applied additionally.

The *ν-local relative fitness* of an index μ can now be defined as

$$p_\nu^t(\mu) := \frac{F(x_\mu^t)}{\sum\limits_{k \in \mathcal{N}_\nu} F(x_k^t)}$$

and the ν*-local cumulative relative fitness* of an index μ as

$$\mathrm{CRF}_\nu^t(\mu) := \sum_{k \in \mathcal{N}_\nu \,:\, k \leq \mu} p_\nu^t(k).$$

Proportionate selection can now be applied in a canonical way. For each parent to select, a random number ξ is drawn uniformly from $[0,1)$ and the individual with index k is chosen with

$$\mathrm{CRF}_\nu^t(k) = \min\{i \in \mathcal{N}_\nu : \mathrm{CRF}_\nu^t(i) \geq \xi\}.$$

As an example, figure 1 shows a small torus and the population indices of the individuals. In case of a von–Neumann neighborhood structure, the individuals inside of the dashed line belong to the neighborhood of the individual with index 7, so $\mathcal{N}_7$ would be $\{2,6,7,8,12\}$.

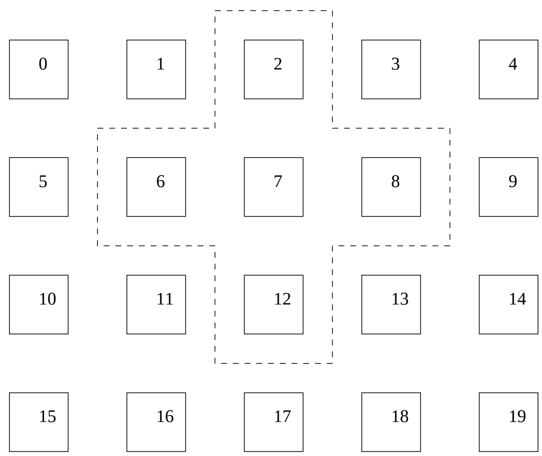

Fig. 1: Example of a neighborhood structure on a torus.

The following table lists the (fictional) fitness values and the resulting local relative fitnesses:

μ	2	6	7	8	12
$F(x_\mu^t)$	16	8	12	20	4
$p_7^t(\mu)$	$4/15$	$2/15$	$3/15$	$5/15$	$1/15$
$\mathrm{CRF}_7^t(\mu)$	$4/15$	$6/15$	$9/15$	$14/15$	$15/15$

The generation of offspring can be performed in parallel. For each position in the population, two parents are chosen from the neighborhood by local proportionate selection, and one child is generated by recombination and mutation. The individual at the current position is replaced by the new child if the latter is accepted, otherwise it remains unchanged.

2.2 Threshold Adjustment

As shown in [9], a standard GA with proportionate selection is not globally convergent to the optimum. Motivated by the *Grand Deluge Algorithm* of Dueck [3], an adaptive threshold acceptance schedule was added in [10]. For the convergence proof, the reader is referred to the original work. In the GDEA, the local threshold τ_k^t at index k and generation t is defined as

$$\tau_k^t := \begin{cases} F(x_k^0) & \text{, if } t < \delta \\ \max\{\tau_k^{t-1}, F(x_k^{t-\delta})\} & \text{, otherwise.} \end{cases} \tag{1}$$

The *threshold delay* $\delta \in \mathbb{N}$ specifies the lag of generations that a current fitness value will enter the threshold update rule (1). A new offspring at a given position in the population is only accepted if its fitness value exceeds the local threshold τ_k^t. This "tidal value" is the maximum of the fitness of the predecessor at this position δ generations in the past, and the tidal value of the last generation. Evidently, the local tides are monotonic rising by definition.

Since the value of δ determines how many generations without improvement are tolerated at most, it is a control parameter of the selection pressure. For $\delta = 1$, only improvements are accepted, whereas values beyond the maximum number of generations turn off the threshold acceptance. It conjunction with large neighborhoods, the latter case is very close to a traditional GA.

2.3 Outline of the Algorithm

The following pseudo code gives a sketch of the algorithm:

```
initialize population
REPEAT
  FOR EACH node
    select two neighbors
    recombine them
    mutate resulting offspring
    evaluate offspring
    IF F(offspring) > threshold
    THEN
       accept offspring
    ENDIF
    update local threshold
  ENDFOR
UNTIL maximum number of generations
```

3 Computational Experiments

3.1 Choice of Neighborhoods

In [10] we assumed that the population's directed graph $G = (V, E)$ with edges $E = \{(\nu, \mu) : \nu \in V,\ \mu \in \mathcal{N}_\nu\}$ was embedded into a processor network with bidirectional ring topology. To keep the (virtual) communication load low we decided to use neighborhoods of the following type: Let $R \in \mathbb{N}$ denote the *neighborhood radius* and $\mathcal{O} = \{a \in \mathbb{Z} : |a| \le R\}$ a set of offsets. The neighborhood set of the individual with label ν is $\mathcal{N}_\nu = \{(\nu + n + a) \bmod n : a \in \mathcal{O}\}$ where n is the population size. We shall say (with some lack of precision) that the population is living on a ring or that the optimization problem is treated on a ring whenever a neighborhood of the above type is used.

After our initial experiments on the ring [10] we wondered whether two–dimensional neighborhood patterns would result in a qualitative change of performance or a change of significance of the parameters controlling locality and selection pressure. Therefore we imagined that the individuals are living on a toroidal processor network and that each individual possesses the same two–dimensional neighborhood pattern. These patterns can be defined by a mask or matrix $M = (m_{ij})$ with an odd number of columns and rows whose central element refers to the current individual with label $\nu \in V$. An entry of M with $m_{ij} = 1$ indicates that the corresponding individual on the torus belongs to the neighborhood set of individual ν, otherwise the entry is zero. This is enough to calculate the neighborhood sets:

The matrix $M = (m_{ij})$ with r rows, c columns (r, c odd), $m_{ij} \in \{0, 1\}$ for all $i \in I = \{0, 1, \ldots, r-1\}$, $j \in J = \{0, 1, \ldots, c-1\}$ and $m_{r/2,c/2} = 1$ is called the *neighborhood mask*. The set

$$\mathcal{O}_M = \left\{ \left(i - \frac{r}{2} \,,\, j - \frac{c}{2} \right) \in \mathbb{Z}^2 : m_{ij} = 1,\ (i, j) \in I \times J \right\}$$

is termed the *offset set* of neighborhood mask M. For example, the neighborhood mask

$$M = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

defines a neighborhood pattern that is related to the maximum norm in $\mathbb{Z}^2$, i.e., the offset set is $\mathcal{O}_M = \{w \in \mathbb{Z}^2 : \|w\|_\infty \le 2\}$ or explicitly

$$\mathcal{O}_M = \{(0,0), (0,\pm 1), (0,\pm 2), (\pm 1, 0), (\pm 2, 0), (\pm 1, \pm 1), (\pm 1, \mp 1)\}.$$

Let the pair $(n, k) \in \mathbb{N}^2$ such that $n = k \cdot q$ with $q \in \mathbb{N}$ and where n is the population size. The function $\delta_k(\nu) = (\nu \text{ div } k\,,\ \nu \bmod k)$ with its inverse $\delta_k^{-1}(a, b) = a \cdot k + b$ will serve to map the population into a grid and vice versa. Now the neighborhood set can be defined easily:

$$\mathcal{N}_\nu = \{\delta_k^{-1}((\delta_k(\nu) + (i, j) + (q, k)) \bmod (q, k)) : (i, j) \in \mathcal{O}_M\}.$$

In this formalism the experiments made on the ring [10] can be described by setting $k = n$ and $\mathcal{O} = \{w \in \mathbb{Z}^2 : w_1 = 0, \|w\|_\infty \le R\}$.

While the neighborhood size in a ring can be increased gradually the neighborhood size defined by regular two–dimensional patterns increases in larger steps when usual distance measures (norms in $\mathbb{Z}^2$) are used. Therefore, the comparability of the effects of locality between one– and two–dimensional patterns would be hardly possible. As a consequence, we defined neighborhood masks whose patterns were inspired by chamfer–distances [2] in order to “smooth” the transitions to larger neighborhood sizes. For example, the matrix C below characterizes 9 different neighborhood masks with neighborhood sizes ranging between 5 and 49:

$$C = \begin{pmatrix} 9 & 8 & 7 & 5 & 7 & 8 & 9 \\ 8 & 6 & 4 & 3 & 4 & 6 & 8 \\ 7 & 4 & 2 & 1 & 2 & 4 & 7 \\ 5 & 3 & 1 & 0 & 1 & 3 & 5 \\ 7 & 4 & 2 & 1 & 2 & 4 & 7 \\ 8 & 6 & 4 & 3 & 4 & 6 & 8 \\ 9 & 8 & 7 & 5 & 7 & 8 & 9 \end{pmatrix}$$

The mask M_d is defined via $m_{ij} = 1$ if $c_{ij} \leq d$ for $d = 1, \ldots, 9$ and zero otherwise. Our experiments were made with patterns of the above type resulting in the neighborhood sizes $\{5, 9, 13, 21, 25, 29, 37, 45, 49, 81, 121, 169, 225\}$ where the steps between the last five sizes were enlarged intentionally to reduce the computation time required for our study.

3.2 Objective Functions

Our experiments were made on two problems: a pseudo–boolean and a pseudo–continuous one. The first one was a NP–hard multiple knapsack problem already investigated for populations on a ring in [10]. The problem can be formalized as follows:

$$\begin{aligned} f_1(x) = c^T x & \quad \rightarrow \max! \\ \text{s.t. } Ax \leq b & \end{aligned}$$

with $x \in \mathbb{B}^\ell$, $c \in \mathbb{R}_+^\ell$, $b \in \mathbb{R}_+^m$ and $A \in \mathbb{R}_+^{m,\ell}$. The constraints were included into the objective function by a penalty technique in the same manner as in [6]:

$$f_1(x) = c^T x - \beta \cdot c_{max} \quad \rightarrow \max! \; ,$$

where β denotes the number of violated constraints and c_{max} the largest entry in the cost vector c. Here, the problem had dimension $\ell = 50$ and $m = 5$ constraints.

The objective function of the second test problem was a version of the well–known Rastrigin function:

$$f_2(x) = 5000 - \sum_{i=1}^{20} \{ x_i^2 + 10 \, [\, 1 - \cos(2\, \pi \, x_i)] \} \quad \rightarrow \max!$$

where each x_i was represented by a Gray–coded binary string of length 20 such that $|x_i| \leq 5.24288$ for each $i = 1, \ldots, 20$. Thus, the string length of an individual is $\ell = 400$.

3.3 Computational Results

The population size was set to 500 for both the ring and torus topology. While the labels of the individuals in the ring were arranged in linear order, a grid of 20×25 was the basis of the labeling in the torus.

For the multiple knapsack problem, the neighborhood size $|\, \mathcal{N}_\nu \,|$ was varied from 3 to 200 for the ring topology (results taken from [10]), and from 5 to 225 for the torus. In both cases, the threshold delay δ ranged from 1, which can be seen as a local elitist selection, to 500 which was the maximum number of generations. For each combination of δ and $|\mathcal{N}_\nu|$ the success frequency, which is the ratio of the number of runs that found the global optimum to total number of runs to the, was calculated from 200 independent experiments.

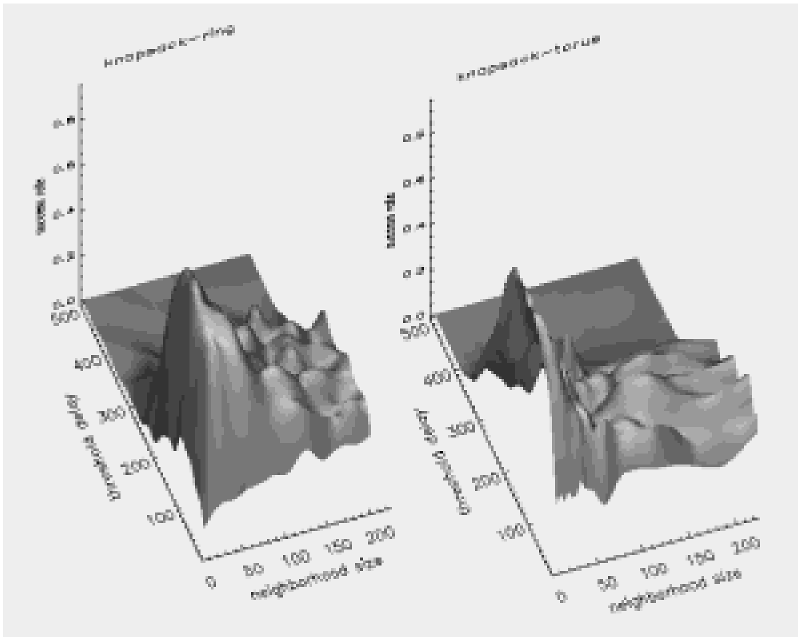

Fig. 2: Success frequency for the multiple knapsack problem with varying parameters.

Figure 2 summarizes the success frequencies depending on δ and $|\mathcal{N}_\nu|$ in both ring and torus topology, i.e., for one– and two–dimensional neighborhood patterns. In contrast to the results of the same experiment in the ring topology, the highest success rates in the torus were obtained by the smallest neighborhood sizes. In fact, the optimal settings were approximately $(|\mathcal{N}_\nu|, \delta) = (40, 100)$ for the ring and $(|\mathcal{N}_\nu|, \delta) = (5, 140)$ for the torus. These settings achieved a success frequency of about 85 % and 95 %, respectively. An interesting observation is the fact that the torus neighborhood yields better results when properly tuned, whereas the ring topology behaves more robust against missetting of the neighborhood sizes. But in both cases, it is obvious that selection pressure is the key to success, because too large threshold delays decrease the success rate almost independently of the neighborhood size.

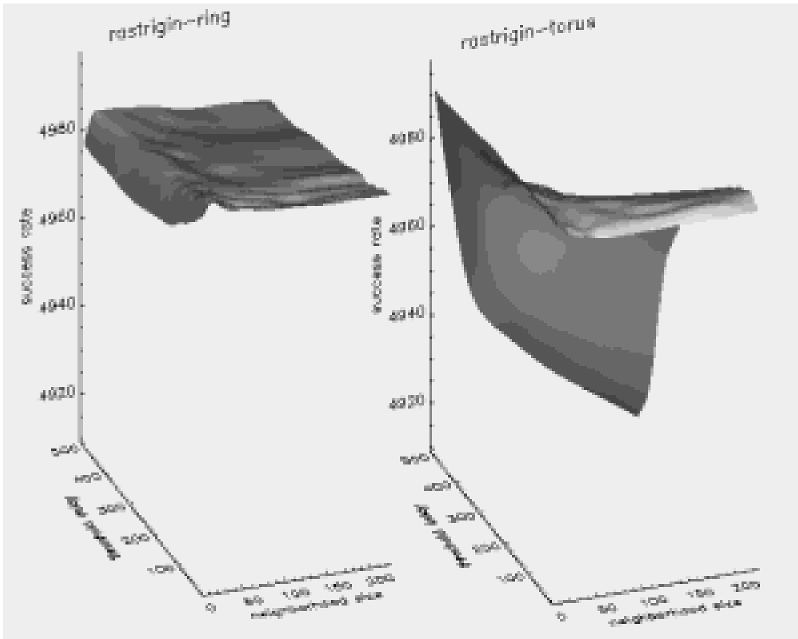

Fig. 3: Averaged best fitness for the Rastrigin test problem with varying parameters.

In contrast to the first test problem, success frequencies are not an appropriate quality measure for a continuous function such as the Rastrigin problem. Instead, the best results after 500 generations were averaged out of 100 runs. Figure 3 shows the response of the GDEA to the variation of δ and $|\mathcal{N}_\nu|$ for the Rastrigin function using ring and torus topology. Again, the best performance of the torus can be observed with the smallest neighborhood sizes and threshold delays significantly lower than half the number of generations. However, the importance of the neighborhood sizes appears to be even lower than in the first experiments.

The optimum neighborhood size for the ring is relatively large again, but the changes to worse values are almost not significant. Additionally, the ring is quite robust against variations of the threshold delay.

4 Conclusions

Speaking in terms of biology, the parameters examined in this paper are selection pressure (threshold delay) and locality (neighborhood size). Obviously, high selection pressure causes a speed-up of the GDEA on its way to the global optimum[1]. With respect to parallelism, we can state that locality does not necessarily improve EAs. On the other hand, it does not harm as long as there is sufficient selection pressure, so small neighborhood sizes should be used if communication bandwidth matters.

The different behaviors of the ring and the torus topology under the condition of equally sized neighborhoods means that locality is not provided by a the number of neighbors, but by the connectivity of the neighborhood structure. In

[1] Remember that the GDEA is globally convergent, so eventually, it will find the optimum.

order to examine this in a more general context, the authors are working on a definition of locality by means of graph theory.

References

[1] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, New York, 1996.

[2] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

[3] G. Dueck. New optimization heuristics: the great deluge algorithm and the record–to–record travel. *Journal of Computional Physics*, 104:86–92, 1993.

[4] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley, Reading (MA), 1989.

[5] M. Gorges-Schleuter. ASPARAGOS: an asynchronous parallel genetic optimization strategy. In J.D. Schaffer, editor, *Genetic Algorithms, Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 422–427. Morgan Kaufman, San Mateo, 1989.

[6] S. Khuri, Th. Bäck, and J. Heitkötter. The zero/one multiple knapsack problem and genetic algorithms. In E. Deaton, D. Oppenheim, J. Urban, and H. Berghel, editors, *Proceedings of the 1994 ACM Symposium on Applied Computing*, pages 188–193. ACM Press, New York, 1994.

[7] H. Mühlenbein, M. Gorges-Schleuter, and O. Krämer. Evolution algorithms in combinatorial optimization. *Parallel Computing*, 7:65–88, 1988.

[8] M.E. Palmer and S.J. Smith. Improved evolutionary optimization of difficult landscapes: Control of premature convergence through scheduled sharing. *Complex Systems*, 5:443–458, 1991.

[9] G. Rudolph. Convergence properties of canonical genetic algorithms. *IEEE Transaction on Neural Networks*, 5(1):96–101, 1994.

[10] G. Rudolph and J. Sprave. A cellular genetic algorithm with self–adjusting acceptance threshold. In *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 365–372. IEE, London, 1995.

[11] P. Spiessens and B. Manderick. A massively parallel genetic algorithm: Implementation and first analysis. In R.K. Belew and L.B. Booker, editors, *Proceedings of the Fourth Conference on Genetic Algorithms*, pages 279–286. Morgan Kaufmann, San Mateo, 1991.

[12] J. Sprave. Parallelisierung Genetischer Algorithmen zur Suche und Optimierung. Diplomarbeit, University of Dortmund, Department of Computer Science, 1990.