LINEAR NEIGHBORHOOD EVOLUTION STRATEGY

JOACHIM SPRAVE
Department of Computer Science, University of Dortmund
44221 Dortmund, Germany
E-Mail: sprave@ls11.informatik.uni-dortmund.de

ABSTRACT

Global selection in panmictic evolutionary algorithms often causes premature convergence. Approaches with non-panmictic populations have been made mostly in parallel implementations to eliminate global control. This results in new types of selection, which may also improve the reliability of sequential evolutionary algorithms. Most previous approaches in the past were based on genetic algorithms (GA), but this paper focuses evolution strategies (ES). Three ES-based approaches with a ring topology are introduced, one with a GA-like kind of mating selection, and two which are very close to the standard (μ, λ)-ES and $(\mu + \lambda)$-ES. They have been compared to traditional ES in numerical experiments. The results show the dependence of convergence reliability and neighborhood sizes, and that local selection can improve the exploitation facility of ES with only a linear slow-down in exploration.

1. Parallel Evolutionary Algorithms

1.1. Interaction Models

There are at least two different approaches to parallel evolutionary algorithms (EA). The first is called the *migration model* and results in coarse grain distributed applications. A couple of instances of a standard EA are running on different processors, providing a migration facility to exchange genetic material between populations. The travelers can be chosen in many ways – you can allow the best to travel, urge the worst, choose travelers randomly or anything else.

Communication can be accomplished asynchronously to prevent the processors from being idle. By tuning the local population sizes and migration frequencies these algorithms can be scaled to a balanced usage of processing and communication resources.

The subpopulations are still panmictic: for each individual, any other individual in the same population is a potential mating partner.

The second approach is known as *diffusion model*, *plant pollination model*[5] or *neighborhood model*. The individuals are placed in an arbitrary space, e.g., on a two-dimensional grid. On top of this topology it is easy to define some kind of neigh-

borhood. By constraining evolutionary processes to this neighborhood, the need for any kind of global knowledge is eliminated. Mapping this to a grid of processors results in a fine grain approach for parallel EAs. In contrast to migration, these algorithms lack the feature of scalability. Nevertheless, because of their regular structure, they are well-suited for data parallel computers, such as the Connection Machine.

The neighborhood model is not only an approach for parallelism, it is as well a better model of nature. The existence of a global selection authority is a very strong simplification in classic EA.

To implement the two-dimensional neighborhood model on MIMD-machines, the processors themselves are connected to a torus, and a subgrid of individuals is placed on every processor. Now the borders must be exchanged between adjacent subgrids after every generation. Because EAs are very robust, there is no need for synchronization of the tasks placed on different processors.

An implementation like this, based on a genetic algorithm (GA), is DIOGENES[11]. It has shown that the size of the subgrids is the parameter to balance computation and communication, thus the parallelism is scalable as in the migration model. Numerical results from DIOGENES have shown that local selection can improve the ability of EAs to find the global optimum of multi-modal problems.

1.2. Local Selection in Genetic Algorithms

There are many parallel EAs using the neighborhood model, and most are GA-based. These approaches have in common that a new selection operator had to be introduced.

To generate the successor of an individual for the next generation, Collins and Jefferson[4] took two series of samples from the neighborhood and chose the best of each series as parents for the new individual.

Sprave[11] made experiments with different selection strategies. The best results were achieved by a deterministic selection: each individual was replaced by the offspring of itself and its fittest neighbor.

Spiessens and Manderick[10] took a selection scheme as close as possible to sequential GA, a local proportionate selection: first, each individual is replaced by one of its neighbors, selected according to the local fitness distribution. After that, each individual selects a mating partner from its neighborhood (which itself has been selected by local proportionate selection in the step before), to produce a successor for this position in next generation.

All neighborhood GAs have shown that the diversity in a population can be preserved for many generations. Thus, exploration of the search space is improved, and

the exploitation is not necessarily worse than in a panmictic GA.

2. Linear Neighborhood Evolution Strategy

Using local selection also in sequential EAs should lead to more diversity during the optimization process, and thus increase the probability of discovering the global optimum of multimodal objective functions. This approach looks very much like a *Cellular Automaton* (CA). There are two violations of the rules which Wolfram[12] defined a CA to possess: the transitions are not deterministic, and, in case of an ES, the states are not discrete.

In the theory of cellular automata, most analysis concerns one-dimensional or *linear* cellular automata, because they are simple but have nearly all the properties of multi-dimensional CA. So, the linear topology seems to be a good choice for analyzing cellular EA, too.

2.1. Local Selection Schemes

The next three sections describe three local selection schemes for ES. Although they are described for a ring topology in this paper, they are well-suited for any kind of topology as long as a neighborhood of more than two individuals is defined for each place in the population.

The *Linear-Neighborhood-ES* (LNES) is an ES with a neighborhood defined by using the index of each individual. The neighborhood with radius ρ of the individual a_i consists of

$$\mathrm{NB}_\rho(i) := \{a_{(i-\rho)}, .., a_{(i+\rho)}\} \tag{1}$$

where the indices are calculated modulo population size so that the result is a ring topology.

2.2. Local Mating Selection

The difference of selection schemes of GA-like proportionate selection, ranking and so on, and the ES's (μ, λ)-selection can be reduced to different acceptance probability functions. Nevertheless, selection in GA is more or less intended as a *mating selection*, wherein the best individuals have most offspring. In ES, a surplus of offspring is produced and only the best of them survive and each has the same chance to be chosen as a parent.

While it is easy to constrain the GA's mating selection to a neighborhood, the extinctive selection used in ES is inherently global. Of course, it is possible to transfer the GA's selection scheme to ES by choosing two parents from the neighborhood by

proportionate selection, and replace the central individual by their offspring in the next generation.

The neighborhood size is typically small to maximize local effects. Therefore, the simple mating selection described below should produce similar results like ranking or proportionate selection, except for the fact that, as long as the neighborhood size is not greater than the population size, one individual cannot be chosen twice to produce one offspring.

For each individual a_i, do synchronously (in parallel or pseudo-parallel):

- Choose the two best from its neighborhood.
- Recombine them to one offspring named a'_i.
- Mutate a'_i.
- Replace a_i by a'_i.

Recombination and mutation operators are the same as in traditional ES (for details see Schwefel and Bäck[2]). Note that if the neighborhood is the whole population, this selection becomes a $(2, P)$-ES (where P is the size of the ring topology) and, if ρ is at least P, a $(1, P)$.

2.3. Local (μ, λ)-Selection

To make comparison to standard ES easy, a (μ, λ)-selection was combined with the neighborhood model. Assuming that λ is a multiple of the population size μ, the calculation of a new generation in a (μ, λ)-LNES works as follows:

For each individual a_i, do synchronously (in parallel or pseudo-parallel):

- Choose λ/μ times two parents randomly from $\mathrm{NB}_\rho(i)$ and recombine them to $a'_{i,j}, \quad 1 \leq j \leq \lambda/\mu$.
- Replace a_i by a'_i which is chosen as the best of the $a'_{i,j}$.
- Mutate a'_i as in a panmictic ES.

Now there are two extreme values for ρ:

- $\rho = \lfloor \mu/2 \rfloor$
  This case is very close to a (μ, λ)-ES, but not the μ best are the parents of the next generation. Instead, there are μ sets of λ/μ individuals created, and the best of each set is taken to the next generation. The selection pressure is weaker, but because the parents are taken randomly from the whole population, it should be still sufficient to generate an efficient search.

- $\rho = 0$
  This is nothing but a $(1, \lambda/\mu)$-ES, started μ times in parallel.

Another interesting case is $\rho = 1$, the smallest symmetrical neighborhood which is possible. It is expected to be slower in unimodal problems, and better in multimodal functions.

2.4. Local $(\mu + \lambda)$-Selection

If the offspring is only accepted when it is better than its local predecessor, the algorithm behaves very much like a $(\mu + \lambda)$-ES. The quality function value of each position in the population is monotonic decreasing (in case of minimization). If there is a lower boundary for the step sizes, the proof of global convergence for the panmictic $(+)$-ES[3] is valid for this kind of ES, too.

3. Results

All test functions are minimization problems taken from a comparison of genetic algorithms and evolution strategies by Hoffmeister and Bäck[6]. For the sphere model, all evaluations are averaged over 50 runs with different seed values for the random number generator, and n different step sizes σ_i were used (where n is the dimension of the test function). For all test functions discrete recombination was taken for x_i, and intermediate recombination for the step sizes σ_i. All step sizes are varied self-adaptive, correlated mutations have not been used at all.

The main objective of the experiments was to observe the effects produced by local selection. A given parameter setting was tested with neighborhood radius ρ varying from 1 to $\mu/2$.

3.1. Sphere Model

The first test function is the sphere model

$$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2 \qquad ; \vec{x} \in \mathbf{R}^n \tag{2}$$
$$n = 30; \quad -5.12 \leq x_i \leq 5.12$$

This function can show the ability of the algorithms to proceed straight to the optimum in unimodal environments. Figure 1 shows the best function value found after $500,000$ function evaluations varying the neighborhood radius from 1 to 50. Population size μ was set to 100 for LNES. The $(+)$ and $(,)$ strategies produced six children per position each generation, while the mating-LNES does not need to produce a surplus. For comparison, a $(100,600)$-ES and a $(10,60)$-ES were run with the same initialization.

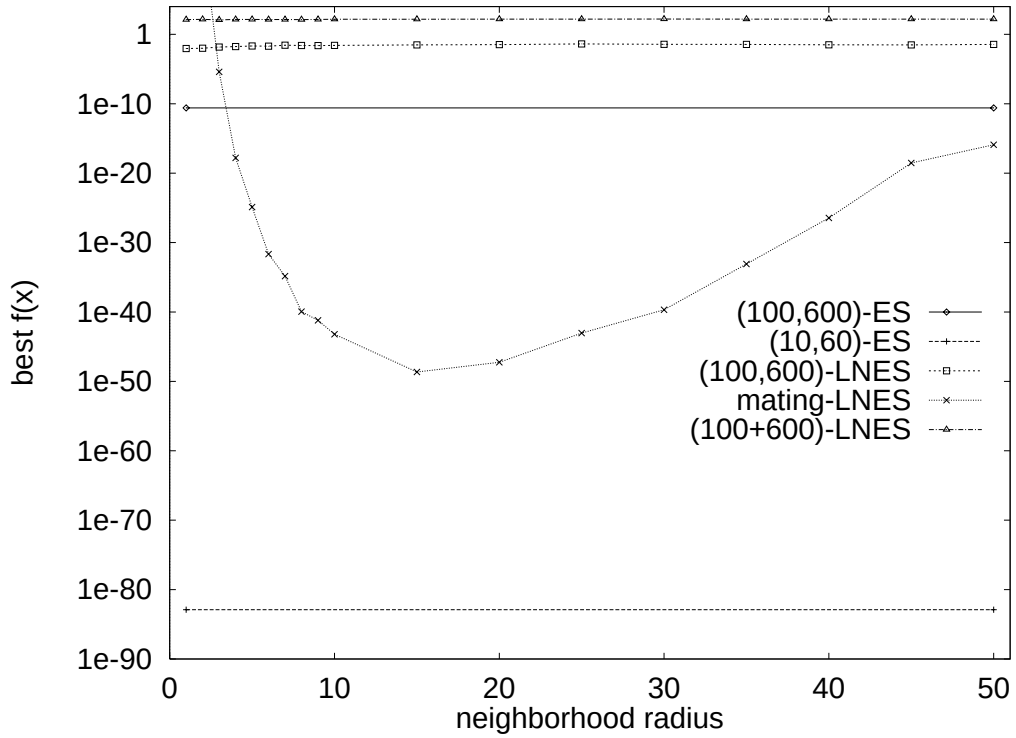

Figure 1: Evaluation of f_1

The local $(\mu + \lambda)$-strategy fails to make any progress towards the optimum (the worst initial value which is possible is 768.2). It is difficult already for a panmictic $(\mu + \lambda)$ strategy to learn n step sizes, but it is obviously impossible for the local variant.

Mating selection works well with a neighborhood radius of six or more. This is a strong hint that this kind of selection also needs a ratio of at least six children per

parent to learn individual step sizes, as suggested by Schwefel[8]. Beyond that ratio, it has a strong emphasis on local search.

Although the (μ, λ)-LNES is much slower than traditional ES, it has the same kind of linear convergence in quadratic environments. In contrast to the former strategy, it performs much better with very small neighborhood sizes, which is very advantageous for parallel implementations: the smaller the neighborhoods, the less communication is required.

3.2. Generalized Rastrigin Function

The generalized Rastrigin function is a very difficult problem for panmictic ES, as shown by Hoffmeister and Bäck[6], and it was used by Rudolph[7] to show the ability of the migration model ES to find the global optimum in multimodal environments.

$$f_7(\vec{x}) = nA + \sum_{i=1}^{n} x_i^2 - A\cos(\omega x_i) \qquad ;\ \vec{x} \in \mathbf{R}^n \tag{3}$$

$$n = 20; \quad A = 10; \quad \omega = 2\pi; \quad -5.12 \leq x_i \leq 5.12$$

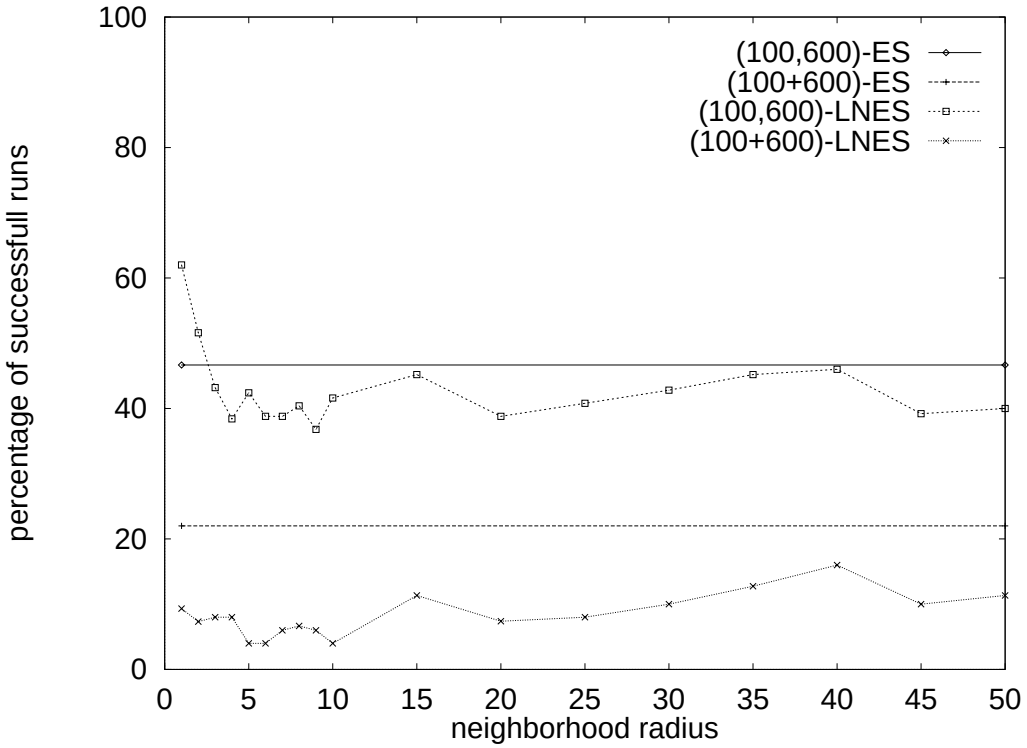

Figure 2: Evaluation of f_7

Figure 2 shows the percentage of successful runs (from 250 runs in total) dependent on the neighborhood radius. Because all variants with multiple step sizes failed to find the global optimum within 100 runs, they are not drawn.

The most successful runs, about 65%, were performed by the $(100, 600)$-LNES with one step size for all x_i and small neighborhood sizes. Larger neighborhood sizes are worse, but nevertheless nearly as good as the panmictic $(100, 600)$-ES. A $(10, 60)$-ES did not find the optimum even once in 150 runs.

The mating-LNES suffers from premature convergence: its emphasis on local search is too strong even with only one step size: very few runs were successful.

3.3. Shekel's Foxholes

Shekel's Foxholes[9] is known as a hard problem for ES. It consists of a large plateau with a few holes of different depths. On the plateau, an ES gets no hint which direction to go to. Once the population falls into one of the holes, it is nearly impossible for the procedure to leave it and search for another.

$$\frac{1}{f_5(\vec{x})} = \frac{1}{K} + \sum_{j=1}^{25} \frac{1}{c_j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \qquad ; \ \vec{x} \in \mathbf{R}^2 \tag{4}$$

$$K = 500; \quad f_5(a_{1j}, a_{2j}) \approx c_j = j$$

$$a_{1j} = 16\,(j \bmod 5 - 2); \qquad a_{2j} = 16\,(\lfloor j/5 \rfloor - 2)$$

To avoid that some initial points are already placed in holes, the start population is not sampled from an interval including the holes as in the comparison of Hoffmeister and Bäck[6], but initialized far away from the holes: $x_i = 10,000$ for all individuals. A $(100, 500)$-LNES with neighborhood size 1 is still able to find the global optimum with a high probability.

To understand the internal process, a visualization method has been applied to LNES which is very often used for Cellular Automata: each objective function value is mapped to a color, so each generation can be displayed as a line of dots. Drawing these lines one below the other shows the fitness distribution inside the population over the time. Figure 3 shows a typical run of LNES on Shekel's Foxholes. In the beginning, the entire population is white, which indicates individuals searching on the plateau. Every hole is mapped to a greyscale, the darker the deeper. It can be seen that, while some points of the population are already in holes, the rest of the population keeps on creeping around on the plateau. The global optimum, painted in black, has been found several times and is expanding slowly but steadily over the population. Eventually, after a number of generations depending on neighborhood sizes, the optimal solution would take over the whole population.

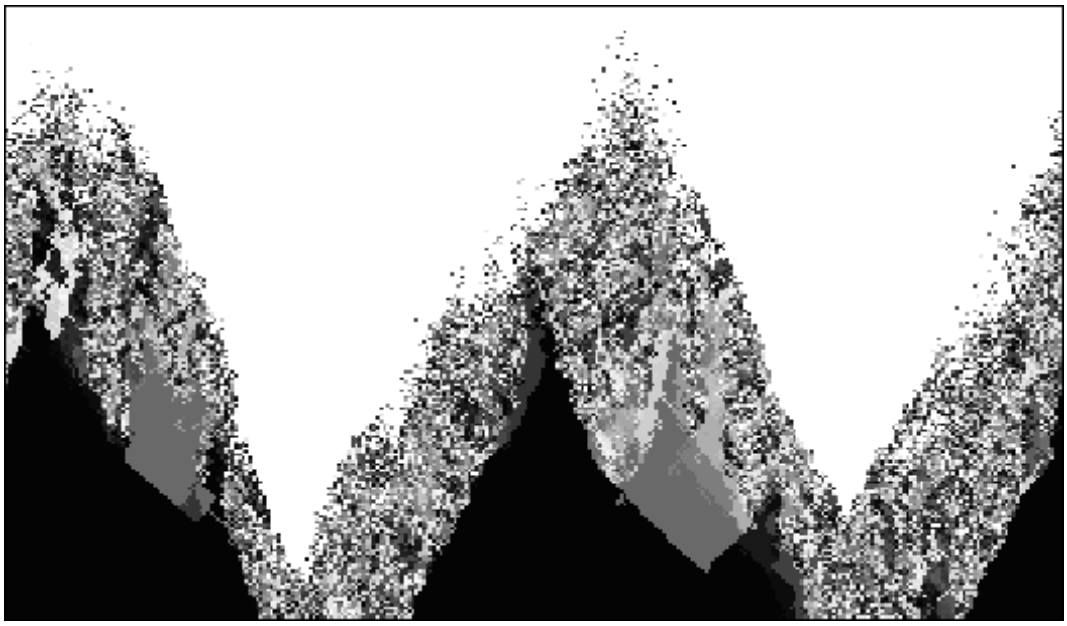

Figure 3: Typical fitness distribution for Shekel's Foxholes

4. Summary

Linear neighborhood ES have properties that are similar to traditional ES. Except for the mating selection, the emphasis on local search is much weaker than in traditional ES, but the exploration facility is much better instead, because they are able to explore a set of attractive areas in parallel.

LNES are free of global evolutionary operators, and are therefore well-suited for parallel implementations. The granularity of parallel implementations can be tuned by the local population size and the neighborhood radius. As already shown[11], the neighborhood model for evolutionary algorithms also works with asynchronous communication to exchange the overlapping neighborhoods.

The local mating selection introduced here is an alternative approach to provide the selection pressure which is necessary for self adaption of step sizes, but it is not suited for global optimization.

1. Richard K. Belew and Lashon B. Booker, editors. *Proceedings of the Fourth International Conference on Genetic Algorithms and their Applications*, University of California, San Diego, USA, 1991. Morgan Kaufmann Publishers.
2. Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.
3. Joachim Born. *Evolutionsstrategien zur numerischen Lösung von Adaptationsaufgaben.* Doctoral dissertation A, Humboldt-University, Berlin, 1978.

4. Robert J. Collins and David R. Jefferson. Selection in massively parallel genetic algorithms. In Belew and Booker[1], pages 249–256.
5. David E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989.
6. Frank Hoffmeister and Thomas Bäck. Genetic algorithms and evolution strategies — similarities and differences. Papers on Economics & Evolution 9103, European Study Group for Evolutionary Economics (ESGEE), November 1991.
7. Günter Rudolph. Globale Optimierung mit Parallelen Evolutionsstrategien. Diploma thesis, University of Dortmund, Department of Computer Science, Dortmund, Germany, July 1990.
8. Hans-Paul Schwefel. Collective phenomena in evolutionary systems. In *Preprints of the 31st Annual Meeting of the International Society for General System Research, Budapest*, volume 2, pages 1025–1033, June 1987.
9. J. Shekel. Test functions for multimodal search techniques. In *Fifth Annual Princeton Conference on Information Schience and Systems*, 1971.
10. Piet Spiessens and Bernard Manderick. A massively parallel genetic algorithm: Implementation and first analysis. In Belew and Booker[1], pages 279–286.
11. Joachim Sprave. *Parallelisierung Genetischer Algorithmen zur Suche und Optimierung.* Diploma thesis, University of Dortmund, December 1990.
12. Stephen Wolfram. Statistical mechanics of cellular automata. In S. Wolfram, editor, *Theory and Applications of Cellular Automata*, chapter 1, pages 7 – 50. World Scientific Publishing, Singapore, 1986.