Published in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 1246–1253, Edinburgh, Scotland, 2005.

Modeling and Analysis of Indirect Communication in Particle Swarm Optimization

Sabine Helwig, Christian Haubelt, and Jürgen Teich
Computer Science Department
University of Erlangen-Nuremberg, Germany
{*helwig, haubelt, teich*}*@codesign.informatik.uni-erlangen.de*

Abstract- Particle Swarm Optimization (PSO) has successfully been applied to many optimization problems. One particularly interesting aspect of these algorithms is to study the communication behavior of the particles. Often, a neighborhood topology is defined a priori and used throughout the optimization run. However, the cost of communication between particles has not been analyzed up to now. In this paper, we will propose a novel algorithm called DAPSO (Distributed Archives PSO) that makes use of stationary archives to establish an indirect communication architecture in the swarms. Moreover, we provide analytical results of the required communication energy in such a scenario. This might be especially important in robot swarms and sensor networks. The applicability of our new methodology will be shown on some selected test cases.

1 Introduction

Particle Swarm Optimization (*PSO*) [1] has successfully been applied to many optimization problems. One particularly interesting aspect of these algorithms is the communication behavior of the particles. Often, a *neighborhood topology* is defined a priori and used throughout the subsequent optimization run. There exists a variety of different neighborhood topologies [1, 2]. The most common are:

- All or *gbest*: All particles are connected.
- Circle(k) or *lbest(k)*: Each particle has exactly k fixed neighbors. The special cases k=2 and k=4 are called ring and von Neumann topology, respectively.
- Wheel: Every particle is connected with the same, central particle; it is, however, isolated from all other members of the population.

All these neighborhood relations are unweighted, undirected, and stay fixed during the whole optimization process. Besides the above, intuitive neighborhood structures, graphs representing the social network with specified average degrees, clustering coefficients (the percentage of the neighbors of a node that are adjacent to each other) or other characteristics can be generated. Studies comparing the above, traditional topologies and systematically constructed neighborhood graphs recommend the von Neumann topology [3], and conclude that graphs with an average degree of 4 and a low clustering coefficient are most promising for a wide range of test problems [4].

However, such an a priori definition of the neighborhood topology might be too expensive when applied in real world search spaces as is the case when using autonomous robots, often battery-powered, as particles, e.g. for detecting hazardous contaminants by mobile sensor networks [5]. In such a scenario, communication comes not for free. Since energy needed for communication depends on the distance, two adjacent particles in different regions of the search space either communicate by a very high effort or even not at all.

In order to overcome these problems, we propose a novel communication strategy based on *stationary archives*. This directly leads to the concept of what we call *indirect communication*: in our new algorithm called DAPSO (Distributed Archives Particle Swarm Optimization), particles only communicate with their nearest archive and the archives provide a *local guide* for each particle.

Using archives raises the question where to place them and how to distribute the solutions over the entire search space. In this paper, we investigate optimization problems using regularly placed archives and different communication schemes among the archives. Based on our novel algorithm, we will provide some analytical results of the expected required communication energy for regularly placed archives. We also show that it is possible to trade communication energy for convergence. To the best of our knowledge there does not exist any work on this particular topic.

The rest of the paper is organized as follows: In Section 2, we briefly review the basic PSO algorithm. In Section 3, we will present our novel DAPSO algorithm using indirect communication through regularly placed archives. The analysis of the expected communication energy is discussed in Section 4. Finally, Section 5 presents some selected test cases using our new algorithm.

2 Particle Swarm Optimization

Particle swarm optimization is based on the social behavior of individuals living together in groups. Each individual tries to improve itself by observing other group members and imitating the better ones. That way, the group members are performing an optimization which can be described with the following model [1]:

Every particle has a *position* $\vec{x}$ in the search space, is able to calculate the corresponding *objective function value*, and moves in the search space with a *velocity* $\vec{v}$. During the whole optimization process, all particles stay alive; they just change their position and velocity. The best position a particle has reached so far is called its *private guide* $\vec{p}$ and the best position ever visited by one of its neighbors is the particle's *local guide* $\vec{l}$.

In each iteration t, the particles' velocities are updated, directing each particle towards its local and its private guide and keeping a proportion of the old velocity:

$$\vec{v}_t = \omega \cdot \vec{v}_{t-1} + \vec{U}[0, \varphi_1] \cdot (\vec{p} - \vec{x}) + \vec{U}[0, \varphi_2] \cdot (\vec{l} - \vec{x}) \quad (1)$$

where $\vec{U}[m, n]$ is a vector of random numbers between m and n, φ_1 and φ_2 are the control parameters or acceleration constants and ω is called *inertia weight*.

In the basic algorithm, each particle is only influenced by its best neighbor, ignoring all others. However, choosing a stochastically or success-dependently weighted sum of the experiences of all neighbors of a particle seems to be a better approach [2, 6].

The updated velocity is then added to the position of the corresponding particle in order to move the particle to its new position:

$$\vec{x}_t = \vec{x}_{t-1} + \vec{v}_t \tag{2}$$

To enable the particles to escape local optima, the positions of a particle can be changed randomly with a low probability at the end of each iteration. This procedure is called *turbulence* or *craziness*.

3 Particle Swarm Optimization with Indirect Communication

As has been observed in the previous section, particles act independently, with no central instance telling them what to do. Although they work together by exchanging information and every particle is influenced by its neighbors, each particle makes its own calculations and decisions. For this reason, particle swarm optimization seems to be predestined for being used in groups of autonomous robots to solve optimization problems in real world search spaces. In such optimization problems, we assume that the evaluation of the objective functions needs input data measured in the corresponding search spaces (see Figure 1).

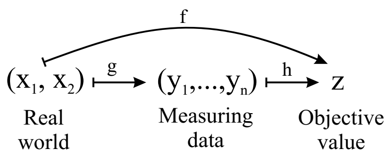

Figure 1: Autonomous robots move in a real world search space in which they measure the data needed to calculate the objective value

However, before transferring PSO strategies to groups of interacting robots, one has to consider the limited resources inherent to such systems. Especially, the limited energy budget must be of concern in case of battery-powered systems. In this paper, we will focus on a single aspect, namely the required *communication energy* by assuming that the robots communicate via radio transmission, to solve a given optimization problem.

In the original algorithm, the neighbors of a particle are defined before the start of the optimization process and communication can take place without knowing the positions of each other and over arbitrary distances. If the basic model is transferred to real swarms of interacting robots, communication cannot be assumed that simple. Since a robot does not know where the other robots are, only undirected communication makes sense. If each robot transmits its information with a predetermined power and all robots that receive this message are defined as its neighborhood, the swarm divides into sub-swarms, which are isolated from each other. Sending undirected broadcast messages over the whole search space is too expensive by means of energy or might even be impossible.

To overcome this problem, we propose a novel PSO algorithm called DAPSO (Distributed Archives PSO), which uses so-called *archives* placed equidistantly on a square grid in the two-dimensional real world search space (see Figure 2). Archives are able to receive and transmit messages as well as to store information. Instead of communicating directly, the particles interact with each other through the archives. To spread information over the search space, the archives communicate with each other according to a specified neighborhood topology.

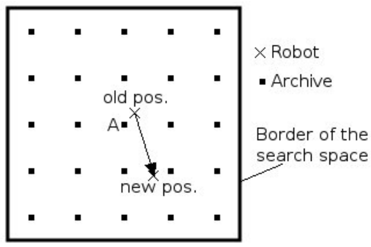

Figure 2: Archives are placed on a square grid in the search space. Particles, e.g. autonomous robots, only communicate with each other via archives.

Several advantages result from the introduction of archives: By increasing the number of archives, and therefore reducing the average distance from a particle to its nearest archive, the expected communication costs of the particles can be decreased. Because of the fixed positions of the archives, the particles can use three or more of them to determine their positions, and they can use directed transmission which is cheaper than undirected transmission. Furthermore, the particles are able to handle large search spaces even if they have a limited transmission range.

The use of archives allows an additional reduction of the communication costs: As the only social information a particle needs in each iteration is its local guide, it is not necessary that the archives forward all information they receive. Instead, the archives determine the local guide for each particle (which is the best position an archive has received so far) and transmit just this position.

Our new DAPSO algorithm, that uses distributed archives and that can be applied to groups of autonomous robots, is shown in Figure 3.

As mentioned above, the archives are able to communicate with each other. Here, three different neighborhood structures are studied (see Figure 4):

- Not connected: The archives are not connected. Information passes very slowly through the search space, as only the particles carry information from archive to archive.
- Nearest: Each archive is connected with its nearest neighbors. No equivalent structure can be defined on

DAPSO — Distributed Archives PSO

- Particles are initialized on random positions and with random velocities. Private guides are set to initial positions. Archives are initialized on a square grid in the search space.
- While no termination criterion is met
  - For each particle P:
    - P updates the spatially nearest archive A (see Figure 2) with its private guide
    - A tells P its best known position, P's local guide
    - P calculates its new velocity
    - Turbulence takes place with a certain probability
    - P updates its position
    - If necessary, velocity is truncated (search space borders)
    - P updates its private guide
  - Archive-archive communication, according to a specified neighborhood topology, takes place

Figure 3: Our new particle swarm optimization algorithm, called DAPSO uses indirect communication through distributed archives.

the basic algorithm where the particles communicate directly. The information flow is relatively fast: Let a^2 be the number of archives in a quadratic search space and let dim be the number of search space dimensions, then after a maximum of $dim \cdot (a-1)$ iterations new information has reached every archive.

- All: All archives are connected. The information flow is very fast: in each iteration, all information gained in the previous iteration is available for all particles. Thus, regarding the behavior of every single particle, using the all-topology on the archives is equivalent to defining an all-topology on the particles and to using only a single archive.[1]

[1] Both equivalences only hold if the particles act in parallel; i.e., each particle works with the information available from the last iteration, but does not use knowledge gained from other particles earlier in this iteration.

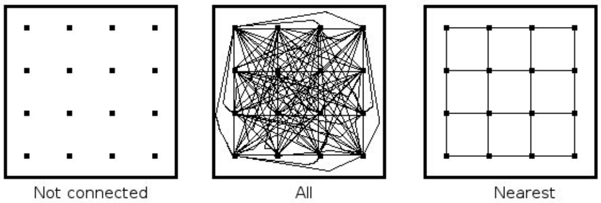

Figure 4: Different neighborhood topologies can be defined on the archives.

Communication through archives as described above strongly differs from the social interaction of the particles in the basic algorithm. If, for example, a ring topology is used in the basic algorithm, each particle has exactly two predefined neighbors. Assume that the particles I and K are the two neighbors of particle J, but I and K are not adjacent. If I finds a new best position, it will notify J about it. But J does not inform K until J itself has improved. Thus, the information flow is relatively slow. In our model, I will notify its nearest archive A about its best position, whereupon A informs all inquiring particles as well as its adjacent archives in the next iteration. Even if a neighborhood topology with a slow information flow is defined for the archives, soon all particles will use the same local guide. This may lead to premature convergence. In order to slow down the information flow, the archives exchange their data only every m-th iteration, where $m \geq 1$ is the user-defined *synchronization distance*.

4 Analysis of Communication Costs

By introducing the stationary archives, we are able to analyze the communication of the swarm. In the following section, we will provide first results for the expected required communication energy. Unfortunately, it is not possible to draw any conclusions for the expected convergence behavior, since this is dependent on the actual problem instance. In Section 5, we will provide some experimental results from selected test cases.

Since the archives are stationary, we assume that they have a significantly higher energy supply compared to the moving particles, and that the communication costs for the archives therefore are negligible.

To calculate the communications costs, we use the first order radio model [7] where $E_{elec} = 50nJ/bit$ is the energy needed to run the transmitter or receiver circuitry and $\epsilon_{amp} = 100pJ/bit/m^2$ the energy for the transmitter amplifier. The costs for transmitting a k-bit message over a Euclidean distance d are calculated according to the following equation:

$$E_T(k,d) = E_{elec} \cdot k + \epsilon_{amp} \cdot k \cdot d^2 \quad (3)$$

and the costs for receiving the same message are:

$$E_R(k) = E_{elec} \cdot k \quad (4)$$

If n is the number of iterations in the above algorithm, the whole communication costs for particle P in one optimization run are:

$$E = \sum_{t=0}^{n-1} (E_{T_{Pt}} + E_{R_{Pt}}) \quad (5)$$

where $E_{T_{Pt}}$ and $E_{R_{Pt}}$ are the transmission and reception costs of particle P in iteration t respectively.

In each iteration, every particle transmits exactly one message: it sends its private guide consisting of a position in the two-dimensional search space and its corresponding objective function value to the nearest archive. Let $3 \cdot l$ be the length of that message in bit. In addition, it receives exactly one message per iteration: its local guide consisting of a position in the search space and therefore having a length of $2 \cdot l$ bit. Thus, including Equations (3) and (4), Equation (5) can be transformed to:

$$E = 5 \cdot l \cdot E_{elec} \cdot n + \epsilon_{amp} \cdot 3 \cdot l \sum_{t=0}^{n-1} d_{Pt}^2 \quad (6)$$

where d_{Pt} is the Euclidean distance of particle P to its nearest archive in iteration t.

The expected communication costs for one particle then are:

$$\overline{E} = 5 \cdot l \cdot E_{elec} \cdot n + \epsilon_{amp} \cdot 3 \cdot l \cdot n \cdot \overline{d}^2 \quad (7)$$

where $\overline{d}$ is the average distance of a particle to its nearest archive.

To obtain an uniform behavior of the swarm at any place of the search space, an infinite search space is assumed. In finite search spaces, the swarm behavior at the borders differs from the usual swarm behavior according to a specified strategy, for example resetting the particle into the search space or assigning an objective function value of $+\infty$ for invalid regions [2].

Theorem 4.1. *Let d_A be the distance of two archives on a two-dimensional square grid in an infinitely large search space and assume that the communication costs of a particle can be calculated according to Equations (5)-(7). Then, the expected required communication costs of a single particle to run the DAPSO algorithm for n iterations evaluate to:*

$$\overline{E}(d_A) = 5 \cdot l \cdot E_{elec} \cdot n + \epsilon_{amp} \cdot 3 \cdot l \cdot n \cdot 0.146 \cdot d_A^2 \quad (8)$$

Proof. In a single run of the algorithm, the particles cluster in a small region around a local or global optimum. If nothing is known about the optimization problem to be solved, the probability for a randomly placed starting population of particles to end up in a certain region of the search space is uniformly distributed, since no position in search space is preferred.

The archives are placed on a square grid with distances d_A. The particles which are inside a square with side length d_A around archive A have A as their nearest archive. The average distance from one particle to its nearest archive is therefore the average distance from a point uniformly chosen in a square with side length d_A to the central point. It can be calculated as follows [8]:

$$\overline{d} = \frac{\sqrt{2} + \ln(1 + \sqrt{2})}{3} \cdot \frac{d_A}{2} \approx 0.383 \cdot d_A \quad (9)$$

which, including Equation (7), results in Equation (8). □

This result cannot be proven through experiments, since for a specific optimization problem the required energy for communication depends on the location of local optima, which are preferred regions for the particles to cluster. In addition, an unbounded search space has been assumed. However, if we have no a priori knowledge of the optimization problem to be solved, the expected required communication costs of a single particle to run the DAPSO algorithm for n iterations can be calculated according to Equation (8).

5 Experimental results

The effects of using archives with different neighborhood topologies and synchronization distances m (that means, the archives communicate every m-th iteration) and of the number of the archives on the quality of the solutions has been studied experimentally on a variety of frequently used test functions [9, 10]. The results using the DAPSO algorithm from Section 3 are presented in the following.

5.1 Test Functions

All chosen test functions are minimization problems; they are described below:

Sphere: $x_i \in [-100 \ldots 100], \forall i = 1 \ldots dim$

$$f(\vec{x}) = \sum_{i=1}^{dim} x_i^2 \quad (10)$$

Sphere (also called De Jong's function 1) is a very simple, unimodal function whose minimum is $f(0, \ldots, 0) = 0$.

Rastrigin: $x_i \in [-5.12 \ldots 5.12], \forall i = 1 \ldots dim$

$$f(\vec{x}) = 10 \cdot dim + \sum_{i=1}^{dim} \left(x_i^2 - 10 \cdot \cos\left(2 \cdot \pi \cdot x_i\right)\right) \quad (11)$$

The Rastrigin function has many local optima, which are regularly distributed. The global minimum is $f(0, \ldots, 0) = 0$.

Rosenbrock: $x_i \in [-2.048 \ldots 2.048], \forall i = 1 \ldots dim$

$$f(\vec{x}) = \sum_{i=1}^{dim-1} \left(100 \cdot \left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right) \quad (12)$$

Rosenbrock is an unimodal function whose optimum $f(1, \ldots, 1) = 0$ is inside a long and narrow valley.

Schwefel: $x_i \in [-500 \ldots 500], \forall i = 1 \ldots dim$

$$f(\vec{x}) = \sum_{i=1}^{dim} \left(-x_i \cdot \sin\left(\sqrt{|x_i|}\right)\right) \quad (13)$$

The global optimum $f(420.9687, \ldots, 420.9687) = -dim \cdot 418.9829$ is located far away from the second best local minimum, in one corner of the search space. There are many local optima.

Griewank: $x_i \in [-600 \ldots 600], \forall i = 1 \ldots dim$

$$f(\vec{x}) = \sum_{i=1}^{dim} \frac{x_i^2}{4000} - \prod_{i=1}^{dim} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \qquad (14)$$

With is many local minima regularly distributed over the search space, the Griewank function resembles the Rastrigin test function. The global minimum is $f(0, \ldots, 0) = 0$.

Michalewicz: $x_i \in [0 \ldots 3.14], \forall i = 1 \ldots dim$

$$f(\vec{x}) = -\sum_{i=1}^{dim} \left(\sin(x_i) \cdot \left(\sin\left(\frac{i \cdot x_i^2}{\pi}\right)\right)^{2 \cdot m}\right) \qquad (15)$$

Michalewicz is a parameterized, multi-modal function with many local optima located between plateaus. The higher the parameter m, the more difficult it is to find the global optimum. In our experiments, m is set to 10.

Shaffer's f6: $x_i \in [-100 \ldots 100], \forall i = 1, 2$

$$f(\vec{x}) = 0.5 + \frac{(\sin\sqrt{x_1{}^2 + x_2{}^2})^2 - 0.5}{(1 + 0.001(x_1{}^2 + x_2{}^2))^2} \qquad (16)$$

Shaffer's f6 is a two-dimensional test function with many local minima and the global optimum $f(0, 0) = 0$.

5.2 Settings

In the following experiments, the impact of using different neighborhood topologies for the archives (as shown in Figure 4) in combination with different synchronization distances m and different numbers of archives in the DAPSO algorithm on the quality of the solutions will be studied. The measuring data of the particles is a single value which has to be minimized. This means that h in Figure 1 is set to the identity and g is one of the above test functions with a two-dimensional input vector. The following parameters are set to commonly used values: The control parameters φ_1 and φ_2 are set to 1.49445 and the inertia weight ω is set to 0.729. This is equivalent to using the Type 1" constriction of Clerc [11] with $\varphi_1 = \varphi_2 = 2.05$, $\kappa = 1$ and $\chi = 0.729$. A population of 20 particles, which are initialized on random positions with random velocities, is used. Turbulence takes place with a probability of 0.01 by adding or subtracting a random value between 0 and one fifth of the side length of the search space to each component of the velocity vector. When a particle would leave the search space, it is set on the nearest border point instead (bound checking). Each experiment is terminated after $n = 500$ iterations and was repeated 100 times. In the diagrams and tables below, average values are presented.

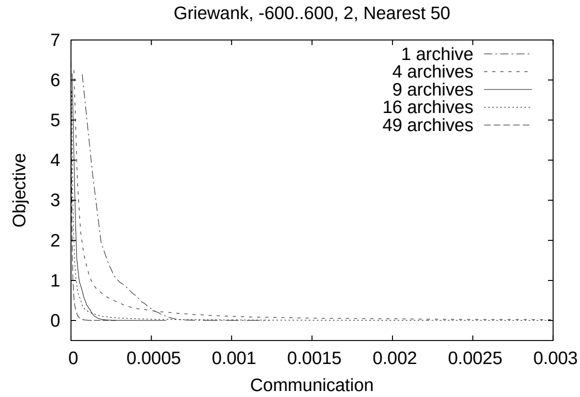

Figure 5: Communication-convergence trade-off for the Griewank function: At the beginning of the optimization process, the more archives are used the less energy is required to improve the solution.

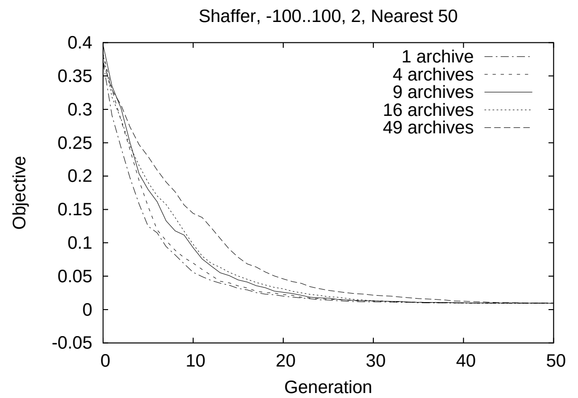

Figure 6: Effects of the number of archives: For most of the test functions, using less archives led to faster convergence at the beginning of the optimization process.

5.3 Results

In Figure 5, the trade-off between required energy amount for communication and convergence is shown for the Griewank function using the Nearest-topology with synchronization distance $m = 50$ (the so-called Nearest 50-topology). In all test functions, increasing the number of archives led to less communication costs for improving the solution at the beginning of the optimization process. However, since the particles end up in similar cluster regions around local optima for each run of the algorithm on a given test problem, the overall communication costs (shown in Table 1) highly depend on the objective function landscape. In most test functions, using four archives was most expensive (often significantly more expensive than using one archive), which means that the particles are clustering in the center of the search space. Since many test functions have a

Table 1: For each test function, the energy needed for the communication of a single particle in one run of the DAPSO-algorithm with 500 iterations is shown. In the rows labeled with *theor.*, the expected communication costs calculated according to Equation (8) are presented. The rows labeled with *exper.* show the average required communication costs of one particle in our experiments. All values are multiplied by 10^{-3}. The often significant differences between the theoretical and experimental results show that the communication costs of a particle highly depend on the fitness landscape. However, the analytical results provide a good estimation if nothing is known about the optimization problem.

Funct./No.Arch.	**1**	**4**	**9**	**16**	**49**
Schwefel theor.	22.082168	5.614292	2.564685	1.497323	0.573105
Schwefel exper.	47.936738	8.588782	2.570988	1.179344	0.339526
Rastrigin theor.	0.127302	0.125576	0.125256	0.125144	0.125047
Rastrigin exper.	0.125112	0.126723	0.125050	0.125402	0.125016
Griewank theor.	31.743322	8.029581	3.638147	2.101145	0.770272
Griewank exper.	1.232643	24.242026	0.592190	5.741837	0.279363
Michal. theor.	0.125216	0.125054	0.125024	0.125014	0.125004
Michal. exper.	0.125036	0.125105	0.125019	0.125019	0.125004
Sphere theor.	1.003286	0.344572	0.222587	0.179893	0.142924
Sphere exper.	0.155080	0.809473	0.137499	0.288174	0.128900
Rosenbr. theor.	0.125368	0.125092	0.125041	0.125023	0.125008
Rosenbr. exper.	0.125306	0.125015	0.125044	0.125058	0.125009
Shaffer theor.	1.003286	0.344572	0.222587	0.179893	0.142924
Shaffer exper.	0.158610	0.770888	0.140401	0.269730	0.130889

Table 2: For each test function, the average best found objective function value is shown for different numbers of archives, using the Nearest 50-topology.

Funct./No.Arch.	**1**	**4**	**9**	**16**	**49**	**Global Opt.**
Schwefel	-796.51237	-836.57249	-836.60724	-834.41256	-831.98906	-837.9658
Rastrigin	0	0	0	0	0	0
Griewank	0.002836543	0.002556502	0.002503729	0.002713728	0.002438109	0
Michalewicz	-1.9447746	-1.9485441	-1.9485441	-1.9485441	-1.94979	unknown
Sphere	0	0	0	0	0	0
Rosenbrock	0	0	0	0	0	0
Shaffer	0.003807991	0.003736249	0.003870557	0.003859896	0.003439982	0

symmetric search space around the origin and they are designed such that the global optimum is located in the origin, the center is preferred and the experimental results are not surprising. By considering the Schwefel test function, which has its optimum in one corner of the search space, the overall communication costs resemble the theoretical results presented in Section 4.

In order to investigate the effect of the number of archives on the convergence of the solutions, we will consider the experimental results of using the Nearest 50-topology with different numbers of archives (1, 4, 9, 16 and 49) in the following.

In all test functions besides the Schwefel and Rosenbrock function, the less archives were used, the faster the particles attained good regions at the beginning of the optimization process (see Figure 6). In Table 2, the average final objective function values using different numbers of archives are shown for each test function. In some functions, the number of archives had no influence on the final quality of the solutions. However, for solving the Schwefel, Michalewicz and Griewank function, using one archive resulted (on average) in worse solutions than using more archives (see also Figure 7, vertical error bars indicate the standard deviation).

As the swarm behavior by using a single archive is equivalent to using the *gbest*-topology in the basic algorithm, the use of archives not only reduces the communication costs but also has the ability to find solutions of equal or even better quality than the basic algorithm with *gbest*-topology.

In the following, the influence of the archives' neighborhood topology and synchronization distance is studied. Therefore, we use a fixed number of archives. The results are presented in Table 3. In many of the test functions, neither the neighborhood topology nor the synchronization distance had an impact on the quality of the final solutions (see Figure 8). In most test functions, a lower synchronization distance resulted in better solutions at the beginning of the optimization process (see Figure 9).

A fast information flow soon drives all particles in the same region of the search space and may lead to premature convergence. In the Michalewicz as well as in the Schwefel function, which are two of the harder test functions for the PSO algorithm, the final quality of the solutions by using a synchronization distance of 1 was worse than the quality of

Table 3: For each test function, the average best found objective function value is presented for different neighborhood topologies and synchronization distances, using a fixed number of 16 archives.

Funct./Top.	**Nearest 1**	**Nearest 20**	**Nearest 50**	**All 1**	**All 20**	**All 50**	**Not. con.**
Schwefel	-801.24915	-836.78161	-834.41256	-803.61871	-832.9335	-836.6376	-832.03711
Rastrigin	0	0	0	0	0	0	0
Griewank	0.003034700	0.002615949	0.002713728	0.002729550	0.002848139	0.003076153	0.002694705
Michalewicz	-1.9415793	-1.9472982	-1.9485441	-1.9389025	-1.946473	-1.9497474	-1.94979
Sphere	0	0	0	0	0	0	0
Rosenbrock	0	0	0	0	0	0	0
Shaffer	0.003338214	0.003419997	0.003859896	0.004679843	0.003126319	0.004851343	0.003279090

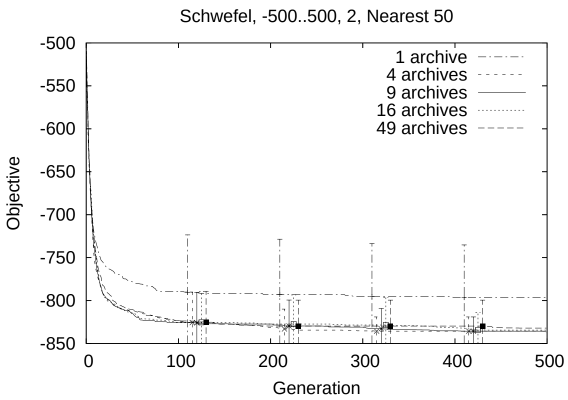

Figure 7: Effects of the number of archives on the convergence: In the Schwefel as well as in the Michalewicz function, using more archives resulted in better solutions. Average results as well as the standard deviation (error bars) are shown.

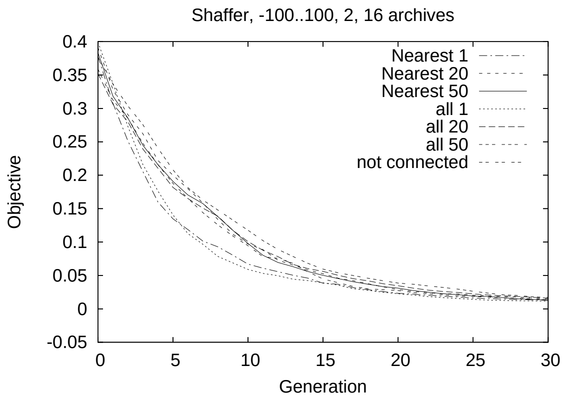

Figure 9: Effects of the neighborhood topology on the initial convergence: The faster the information flow (that means the lower the synchronization distance) the faster the particles attained good solutions at the beginning of the optimization process.

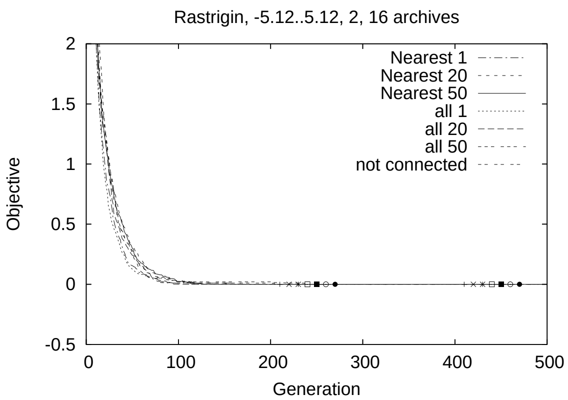

Figure 8: Effects of the neighborhood topology on the convergence: In many test functions, solutions of equal quality were found for any neighborhood topology and synchronization distance at the end of the optimization process.

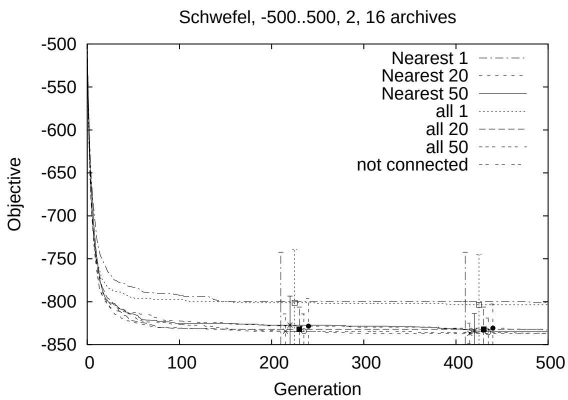

Figure 10: Effects of the neighborhood topology on the convergence: In the Schwefel and Michalewicz test functions, using a low synchronization distance resulted in bad quality solutions.

the solutions attained by slowing down the information flow (see Figure 10). As the swarm behavior using the all-Topology in combination with a synchronization distance of 1 (which is also called All 1-topology) is equivalent to using the original algorithm with *gbest*-topology, this result again proves that our novel approach yields to solutions of equal and even better quality as the standard algorithm with *gbest*-topology.

6 Conclusion and Future Work

As a priori defined neighborhood topologies might be too expensive with respect to the required communication energy when applied in real world search spaces, we proposed a novel algorithm, called DAPSO, that makes use of distributed stationary archives to establish an indirect communication architecture in particle swarm optimization. Based on this algorithm, first analytical results of the expected required communication energy have been presented. Finally, we applied our novel algorithm to selected test cases in order to investigate the trade-off between required communication energy and convergence to the optimal solution. The experimental results have been compared to the standard PSO with *gbest*-topology.

In future work, we would like to generalize our concept to search spaces with more than two dimensions and to compare our new approach to other versions of the PSO algorithm. Moreover, we will investigate the movement energy of particles, collision avoidance strategies, and the effect of using mobile archives. Furthermore, we intend to extend our work towards Multi-Objective Particle Swarm Optimization (MOPSO) [12].

Bibliography

[1] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Academic Press, 2001.

[2] Rui Mendes. *Population Topologies and Their Influence in Particle Swarm Performance*. PhD thesis, Departamento de Informática, Escola de Engenharia, Universidade do Minho, 2004.

[3] James Kennedy and Rui Mendes. Population Structure and Particle Swarm Performance. In David B. Fogel, Mohamed A. El-Sharkawi, Xin Yao, Garry Greenwood, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 1671–1676. IEEE Press, 2002.

[4] Rui Mendes and José Neves. What Makes a Successful Society? Experiments with Population Topologies in Particle Swarms. In Ana L. C. Bazzan and Sofiane Labidi, editors, *SBIA*, volume 3171 of *Lecture Notes in Computer Science*, pages 346–355. Springer, 2004.

[5] X. Cui, T. Hardin, R. K. Ragade, and A. S. Elmaghraby. A Swarm-based Fuzzy Logic Control Mobile Sensor Network for Hazardous Contaminants Localization. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, Fort Lauderdale, Florida, USA, 2004.

[6] Rui Mendes, James Kennedy, and José Neves. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210, 2004.

[7] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.

[8] Sabine Helwig. Roboterschwarm-Optimierung. Diplomarbeit, Hardware-Software-Co-Design, Department of Computer Science 12, University of Erlangen-Nuremberg, 2005.

[9] Rui Mendes, James Kennedy, and José Neves. Watch Thy Neighbor Or How The Swarm Can Learn From Its Environment. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 88–94, 2003.

[10] Hartmut Pohlheim. Geatbx: Genetic and Evolutionary Algorithm Toolbox for use with Matlab Documentation. http://www.geatbx.com/ver_3_5/fcnindex.html, 2004.

[11] Maurice Clerc and James Kennedy. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[12] Sanaz Mostaghim. *Multi-Objective Evolutionary Algorithms: Data Structures, Convergence, and Diversity*. PhD thesis, Faculty of Electrical Engineering, Computer Science and Mathematics, University of Paderborn, Germany, September 2004.