Hybrid Particle Swarm Optimiser with Breeding and Subpopulations

Morten Løvbjerg
EvALife project group
Dept. of Computer science
University of Aarhus
DK-8000 Aarhus C
Denmark
lovbjerg@daimi.au.dk
phone: +45 89423357

Thomas Kiel Rasmussen
EvALife project group
Dept. of Computer science
University of Aarhus
DK-8000 Aarhus C
Denmark
kiel@daimi.au.dk
phone: +45 89423357

Thiemo Krink
EvALife project group
Dept. of Computer science
University of Aarhus
DK-8000 Aarhus C
Denmark
krink@daimi.au.dk
phone: +45 89423358

Abstract

In this paper we present two hybrid Particle Swarm Optimisers combining the idea of the particle swarm with concepts from Evolutionary Algorithms. The hybrid PSOs combine the traditional velocity and position update rules with the ideas of breeding and subpopulations. Both hybrid models were tested and compared with the standard PSO and standard GA models. This is done to illustrate that PSOs with breeding strategies have the potential to achieve faster convergence and the potential to find a better solution. The objective of this paper is to describe how to make the hybrids benefit from genetic methods and to test their potential and competetiveness on function optimisation.

1 Introduction

The Particle Swarm Optimisation (PSO) algorithm was originally introduced in [Kennedy95] as an alternative to the standard Genetic Algorithm (GA). The PSO was inspired by insect swarms and has since proven to be a competitor to the standard GA when it comes to function optimisation. Since then several researchers have analysed the performance of the PSO with different settings, e.g., neighbourhood settings ([Kennedy99, Suganthan99]). Work presented in [Shi98] describes the complex task of parameter selection in the PSO model. Comparisons between PSOs and the standard GA were done analytically in [Eberhart98] and also with regards to performance in [Angeline98]. Angeline points out that the PSO performs well in the early iterations, but has problems reaching a near optimal solution in several real-valued function optimisation problems. Both Eberhart and Angeline conclude that hybrid models of the standard GA and the PSO, could lead to further advances.

We present such a hybrid model. The model incorporates one major aspect of the standard GA into the PSO, the reproduction. In the following we will refer to the used reproduction and recombination of genes only as "breeding". Breeding is one of the core elements that makes the standard GA a powerful algorithm. Hence our hypothesis was that a PSO hybrid with breeding has the potential to reach a better optimum than the standard PSO.

In addition to breeding we introduce a hybrid with both breeding and subpopulations. Subpopulations have previously been introduced to standard GA models mainly to prevent premature convergence to suboptimal points ([Spears94]). Our motivation for this extension was that the PSO models, including the hybrid PSO with breeding, also reach suboptimal solutions. Breeding between particles in different subpopulations was also added as an interaction mechanism between subpopulations.

The introduced hybrids were tested against both standard PSO and standard GA models.

The next section presents the structures of the hybrid PSO models. Section 3 describes the experimental settings used to find the results described in section 4. The experimental results are discussed in section 5 and finally section 6 summarises the study.

2 Model

The traditional PSO model, described by [Kennedy95], consists of a number of particles moving around in the search space, each representing a possible solution to a numerical problem. Each particle has a position vector ($\vec{x}_i$), a velocity vector ($\vec{v}_i$), the position ($\vec{p}_i$) and fitness of the best point encountered by the particle, and the index (g) of the

best particle in the swarm.

In each iteration the velocity of each particle is updated according to their best encountered position and the best position encountered by any particle, in the following way

$$\vec{v}_i = \chi(w\vec{v}_i + \vec{\varphi}_{1i}(\vec{p}_i - \vec{x}_i) + \vec{\varphi}_{2i}(\vec{p}_g - \vec{x}_i))$$

where χ is known as the *constriction coefficient* described in [Clerc99], w is the *inertia weight* described in [Shi98B, Shi98] and $\vec{p}_g$ is the best position known for all particles. φ_1 and φ_2 are random values different for each particle and for each dimension. If the velocity is higher than a certain limit, called V_{max}, this limit will be used as the new velocity for this particle in this dimension, thus keeping the particles within the search space.

The position of each particle is updated in each iteration. This is done by adding the velocity vector to the position vector, i.e.,

$$\vec{x}_i = \vec{x}_i + \vec{v}_i$$

The particles have no neighbourhood restrictions, meaning that each particle can affect all other particles. This neighbourhood is of type *star* (fully connected network), which have been shown to be a good neighbourhood type in [Kennedy99].

The structure of the hybrid model is illustrated in figure 1.

```
begin
    initialise
    while (not terminate-condition) do
        begin
            evaluate
            calculate new velocity vectors
            move
            breed
        end
end
```

Figure 1: The structure of the hybrid model.

The breeding is done by first determining which of the particles that should breed. This is done by iterating through all the particles and, with probability pb (*breeding probability*), mark a given particle for breeding. Note that the fitness is not used when selecting particles for breeding. From the pool of marked particles we now select two random particles for breeding. This is done until the pool of marked particles is empty. The parent particles are replaced by their offspring particles, thereby keeping the population size fixed.

The position of the offspring is found for each dimension by arithmetic crossover on the position of the parents, i.e.,

$$child_1(x_i) = p_i * parent_1(x_i) + (1.0 - p_i) * parent_2(x_i)$$

$$child_2(x_i) = p_i * parent_2(x_i) + (1.0 - p_i) * parent_1(x_i)$$

where p_i is a uniformly distributed random value between 0 and 1. The velocity vectors of the offspring is calculated as the sum of the velocity vectors of the parents normalised to the original length of each parent velocity vector.

$$child_1(\vec{v}) = \frac{parent_1(\vec{v}) + parent_2(\vec{v})}{|parent_1(\vec{v}) + parent_2(\vec{v})|}|parent_1(\vec{v})|$$

$$child_2(\vec{v}) = \frac{parent_1(\vec{v}) + parent_2(\vec{v})}{|parent_1(\vec{v}) + parent_2(\vec{v})|}|parent_2(\vec{v})|$$

The arithmetic crossover of positions and velocity vectors used were empirically tested to be the most promising. The arithmetic crossover of positions in the search space is one of the most commonly used crossover methods with standard real valued GAs, placing the offspring within the hypercube spanned by the parent particles. The main motivation behind the crossover is that offspring particles benefit from both parents. In theory this allows good examination of the search space between particles. Having two particles on different suboptimal peaks breed could result in an escape from a local optimum, and thus aid in achieving a better one.

We used the same idea for the crossover of the velocity vector. Adding the velocity vectors of the parents results in the velocity vector of the offspring. Thus each parent affects the direction of each offspring velocity vector equally. In order to control that the offspring velocity was not getting too fast or too slow, the offspring velocity vector is normalised to the length of the velocity vector of one of the parent particles.

Finally, the starting position of a new offspring particle is used as the initial value for this particle's best found optimum ($\vec{p}_i$).

2.1 Subpopulation Model

The motivation for introducing subpopulations is to restrict the gene flow (keeping the diversity) and thereby attempt to evade suboptimal convergence.

The subpopulation hybrid PSO model is an extension of the just described breeding hybrid PSO model. In this new model the particles are divided into a number of subpopulations. The purpose of the subpopulations is that each subpopulation has its own unique best known optimum. The velocity vector of a particle is updated as before except that the best known position ($\vec{p}_g$ in the formula) now refers to the best known position within the subpopulation that the particle belongs to. In terms of the neighbourhood topology suggested by Kennedy in [Kennedy99], each subpopulation has its own *star neighbourhood*.

The only interaction between subpopulations is if parents from different subpopulations breed. Breeding is now possible both within a subpopulation but also between different subpopulations. An extra parameter called *probability of same subpopulation breeding* (psb) determines whether a given particle selected for breeding is to breed within the same subpopulation (probability psb), or with a particle from another subpopulation (probability $1 - psb$).

Replacing each parent with an offspring particle ensures a constant subpopulation size.

3 Experimental Settings

Both the PSOs and the standard GA were tested on four benchmark problems, all minimisation problems. The first two functions were unimodal while the last two were multimodal with many local minima. All functions are designed such that their global minimum was at or near the origin of the search space.

The first test function was the generalised sphere function given by the equation

$$f_1(x) = \sum_{i=1}^{n} x_i^2$$

where x is a n dimensional real-valued vector and x_i is the ith element of that vector. The second function is the generalised Rosenbrock function given by the equation

$$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

The third function is the generalised Griewank function.

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^{n} (x_i - 100)^2 - \prod_{i=1}^{n} cos(\frac{x_i - 100}{\sqrt{i}}) + 1$$

The fourth and final test function is the generalised Rastrigin function which is given by the equation

$$f_4(x) = \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i) + 10)$$

These four functions have been commonly used in other studies on particle swarm optimisers (e.g. [Kennedy99, Shi99]).

The initial population is usually uniformly distributed over the entire search space. According to [Angeline98] this can give false indications of relative performance - especially if the search space is symmetric around the origin where many test functions have their global optimum. To prevent this, and to ease comparison with other models, the asymmetric initialisation method used in [Angeline98] was used.

Table 1: Search space and asymmetric initialisation ranges for each test function.

Function	Search space	Initialisation range
f_1	$-100 \leq x_i \leq 100$	$50 \leq x_i \leq 100$
f_2	$-100 \leq x_i \leq 100$	$15 \leq x_i \leq 30$
f_3	$-600 \leq x_i \leq 600$	$300 \leq x_i \leq 600$
f_4	$-10 \leq x_i \leq 10$	$2.56 \leq x_i \leq 5.12$

Search space and initialisation ranges for the experiments are listed in table 1. The number of generations run for each test function was set to 1000, 1500 and 2000 corresponding to the dimensions 10, 20 and 30 of the test functions respectively.

In both the standard PSO model and the hybrid model, the upper limits for φ_1 and φ_2 were set to 2.0, and a linearly decreasing inertia weight starting at 0.7 and ending at 0.4 was used. The constriction coefficient χ was set to 1. The maximum velocity (V_{max}) of each particle was set to be half the length of the search space in one dimension (for instance $V_{max} = 100$ for f_1 and f_2).

Two sets of experiments were conducted; Experiments with breeding alone and experiments with both breeding and subpopulations.

Research done in [Shi98] regarding scalability of the standard PSO have shown that the performance of the standard algorithm is not sensitive to the population size. Experiments with the hybrid model confirm this result. Based on these results the population size in the experiments was fixed to 20 particles in order to keep the computational requirements low.

In the experiments with subpopulations, the population size for the *whole* system was also 20. The size of each subpopulation was fixed throughout each run at $\frac{20}{subpopulations}$ particles.

The probability for breeding (pb) was empirically found to have its optimal setting at 0.2, which with 20 particles on average gives a total of two breedings per generation.

In the experiments with subpopulations, the best setting regarding the probability for breeding within the same subpopulation (psb) was determined empirically by examining the results for different settings. The number of subpopulations used in the experiments was 2, 3, 4 and 6. Table 2 shows the relation between the number of populations and the setting for this probability that appeared to be optimal.

The standard GA that we used was a real-valued GA with random initialisation, tournament selection with tournament size two, arithmetic crossover with random weight, Gaussian mutation with distribution $N(0, \alpha)$ where α is

Table 2: Probability for breeding within same subpopulation compared to number of populations

Populations	Psb
1	1.0
2	0.6
3	0.3
4	0.0
6	0.0

linearly decreasing from 1 to 0. Crossover and mutation probabilities for each of the four test functions are listed in table 3. In order to get a fair comparison between the models, with regards to the total number of evaluations, a population size of 20 individuals was also selected for the GA. This was done even though the standard GA often requires larger population sizes in comparison to the standard PSO model [Angeline98]. Other studies [Shi99] show that the standard PSO model with different population sizes have almost the same performance, so the low population size seems to be fair when analysing the PSO model.

Table 3: Crossover and mutation probability used in standard GA.

Function	Crossover prob.	Mutation prob.
f_1	0.60	0.30
f_2	0.50	0.30
f_3	0.50	0.40
f_4	0.20	0.02

A total of 100 runs for each experiment were conducted.

4 Experimental Results

Tables 4 and 5 list a representative set of results from the conducted experiments. The tables list the test function, the dimensionality of the function, the number of generations the algorithm was run and the average best fitness for the best particle found for the 100 runs of the four test functions respectively. Standard error for each value is also listed. Table 4 shows results for the experiments with the hybrid PSO without subpopulations. The table also list the corresponding average best fitness of both the standard PSO and the standard GA with the same settings (where they are applicable) as described in the previous section. Results for experiments with subpopulations are listed in table 5. Note that the hybrid PSO with one subpopulation in table 5 corresponds to the hybrid PSO in table 4.

Figures 2 to 7 are graphs corresponding to the reported experiments.

Figures 2 to 5 show the average best fitness for each generation for both the standard PSO model, the standard GA and the hybrid model. The graphs illustrate a representative set of experiments for functions with a dimensionality of 30. The hybrid model in these figures are without subpopulations (i.e. one subpopulation). Note that the figure with the Griewank function only illustrates two experiments, since the standard GA was unable to achieve a reasonable result (see table 4).

Figures 6 and 7 show the average best fitness for each generation for both the standard PSO model and the hybrid model. The graphs illustrate experiments with both a unimodal (Rosenbrock) and a multimodal test function (Griewank) both of 30 dimensions. The graphs for the hybrid model correspond to experiments with a varying number of subpopulations. The graphs for the standard PSO model are the same as in the previous figures.

Tables 4 and 5 with corresponding figures 2 to 5 show results for the standard PSO supporting the results in [Shi99].

In experiments with the Sphere function the standard PSO achieved better results and had much faster convergence than both the standard GA and the hybrid model with one subpopulation. The GA and the hybrid model found similar values but the hybrid model had a faster convergence speed than the GA. When the number of subpopulations in the hybrid model was increased the best fitness got worse. This happened in all of the experiments.

With the Rosenbrock function, the standard PSO had a better performance than both the GA and the hybrid model. The hybrid model only had a fitness comparable to that of the standard PSO when the test functions were of low dimensionality. When the dimensionality of the test functions were higher, the GA accomplished better results than the hybrid model. The convergence speed of the GA and the hybrid model was better than that of the standard PSO.

In the experiments with the Griewank function, the GA failed to achieve a reasonable result compared to the other models. The hybrid model had a faster convergence than the standard PSO, but achieved a marginally worse best value.

In experiments with the Rastrigin function, the hybrid model was better than both the standard GA and the standard PSO model with both a faster convergence and also a better best value found.

Table 4: Average best fitness of 100 runs for experiments without subpopulations (Average best fitness±standard error).

f	Dim.	Gen.	Std. PSO	Std. GA	Hybrid
f_1	10	1000	2.98E-33±4.21E-33	2.43E-04±1.14E-05	2.42E-04±2.17E-05
f_1	20	1500	3.03E-20±9.27E-21	0.00145±6.22E-05	0.00212±2.75E-04
f_1	30	2000	6.29E-13±7.64E-14	0.00442±1.78E-04	0.01203±6.33E-04
f_2	10	1000	43.049±11.554	109.810±6.212	43.521±16.047
f_2	20	1500	115.143±19.871	146.912±10.951	169.112±21.535
f_2	30	2000	154.519±24.512	199.730±16.285	187.033±22.960
f_3	10	1000	0.08976±0.00498	283.251±1.812	0.09078±0.03306
f_3	20	1500	0.03601±0.00298	611.266±3.572	0.00459±0.01209
f_3	30	2000	0.01504±0.00241	889.537±3.939	0.09911±0.00106
f_4	10	1000	4.8021±0.2323	3.1667±0.2237	3.0599±0.1535
f_4	20	1500	21.3917±0.7885	16.8732±0.6007	11.6590±0.3602
f_4	30	2000	46.9712±1.3206	49.3212±1.1204	27.8119±0.8059

Table 5: Average best fitness of 100 runs for experiments with subpopulations (Average best fitness±standard error). ("Hybrid (i)" is the hybrid model with i subpopulations).

f	Dim.	Gen.	Hybrid (1)	Hybrid (2)	Hybrid (4)	Hybrid (6)
f_1	10	1000	2.42E-04±2.17E-05	3.796E-05±9.22E-05	0.00223±9.13E-04	0.02124±0.00641
f_1	20	1500	0.00212±2.75E-04	0.00175±2.28E-04	0.00566±0.00185	0.04597±0.00721
f_1	30	2000	0.01203±6.33E-04	0.17396±4.56E-04	0.02023±0.00349	0.05669±0.00738
f_2	10	1000	43.521±16.047	51.701±13.761	63.369±14.006	81.283±14.907
f_2	20	1500	169.112±21.535	129.570±14.880	108.391±16.928	137.236±19.619
f_2	30	2000	187.033±22.960	196.554±14.733	279.390±19.468	247.724±31.822
f_3	10	1000	0.09078±0.03306	0.46423±0.03700	0.69206±0.02758	0.74694±0.01844
f_3	20	1500	0.00459±0.01209	0.02231±0.02121	0.09885±0.01883	0.34306±0.03072
f_3	30	2000	0.09911±0.00106	0.06316±0.00121	0.16389±0.00913	0.37501±0.02842
f_4	10	1000	3.0599±0.1535	3.5615±0.1478	3.6840±0.2611	6.8036±0.4657
f_4	20	1500	11.6590±0.3602	12.9158±0.3107	11.6379±0.5308	11.7054±0.5992
f_4	30	2000	27.8119±0.8059	38.5897±0.6455	29.5827±1.0649	29.1747±0.9449

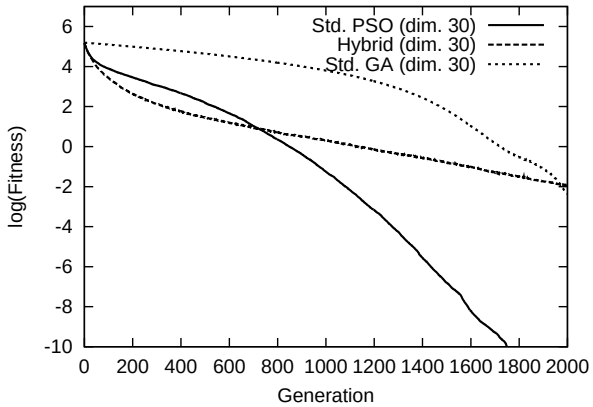

Figure 2: Standard PSO versus hybrid model for Sphere function with one population.

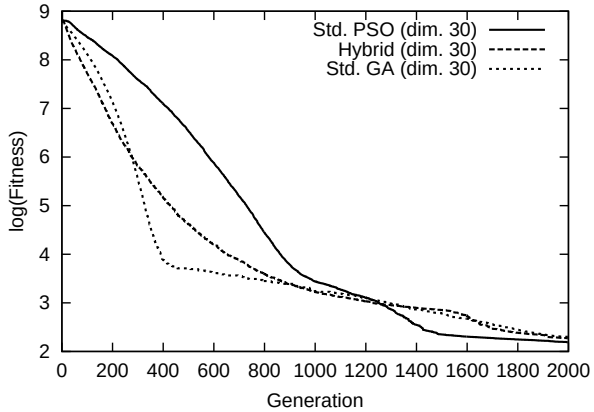

Figure 3: Standard PSO versus hybrid model for Rosenbrock function with one population.

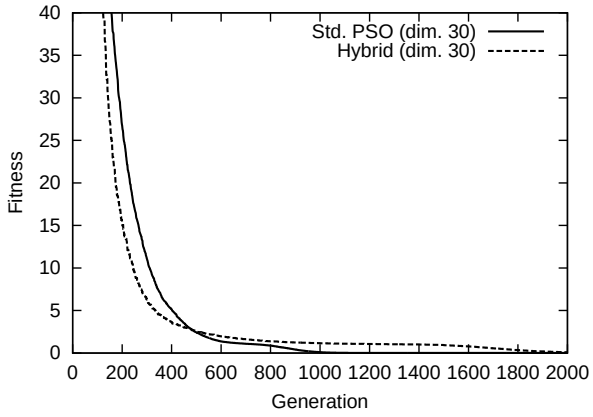

Figure 4: Standard PSO versus hybrid model for Griewank function with one population.

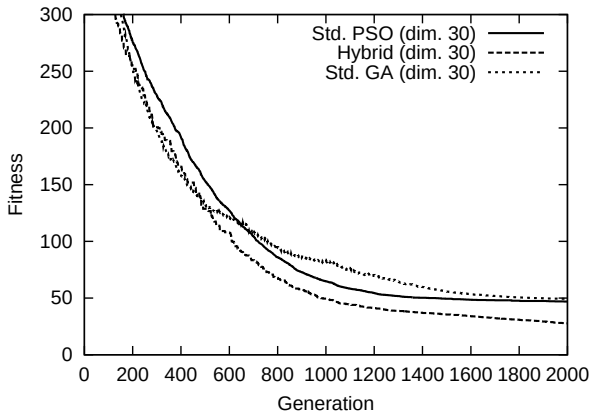

Figure 5: Standard PSO versus hybrid model for Rastrigin function with one population.

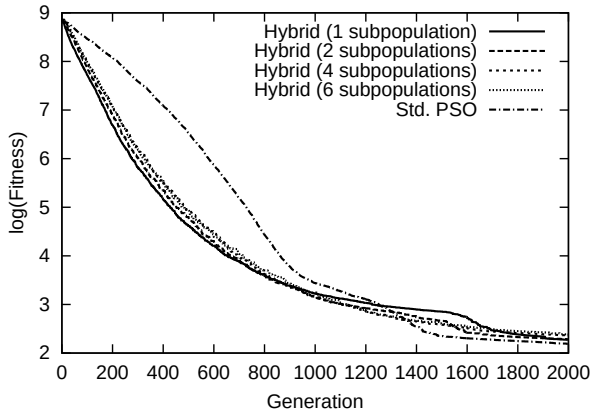

Figure 6: Hybrid model with different number of subpopulations versus standard PSO (Rosenbrock 30 dim.).

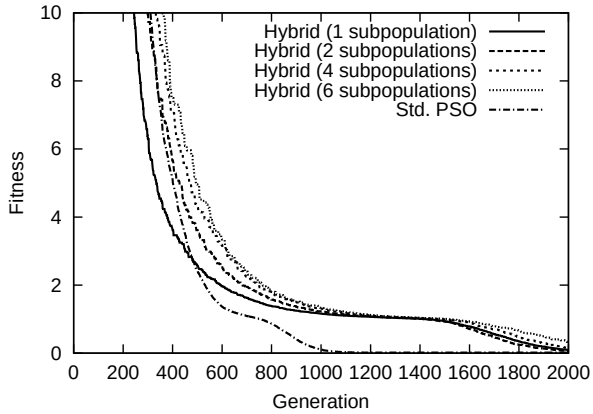

Figure 7: Hybrid model with different number of subpopulations versus standard PSO (Griewank 30 dim.).

5 Discussion

Tables 4 and 5 show a comparison of the performances in the standard PSO model, the standard GA, and the breeding PSO hybrid with regards to the optimum found.

Looking at the unimodal functions Sphere (f_1) and Rosenbrock (f_2) both the hybrid and the standard GA seem to outperform by the standard PSO. As mentioned in section 2 the offspring are initialised with a clean memory, i.e., the previously best found solution of a new particle is its starting point in the search space. This should provide a form of diversity since new particles are unaware of previously found optima. The purpose of adding diversity to the standard PSO is to tackle the problem of avoiding sub-optimal solutions. When we try to avoid sub-optimal solutions we run the risk of not beeing able to find a close to optimal solution because the particles takes longer to converge. This could be why the hybrid model suffers in experiments with unimodal functions.

Looking at the multimodal functions Griewank (f_3) and Rastrigin (f_4) the hybrid model should have a better chance of outperforming the standard PSO, because of the extra diversity. Table 4 does not show an improvement for the Griewank function, but figure 4 shows that the hybrid model converges faster than the standard PSO model. The standard GA was not able to reach a reasonable optimum in any of the experiments with the Griewank function. This is probably due to the fairly small population size in the GA. Table 4 along with figure 5 show the improvements for the Rastrigin function. Here both faster convergence is achieved and an improvement in the best solution is found. These results could be because of the design of the crossover operator that allows offspring particles to escape local optima (see section 2). The results seem to show

the potential of particle breeding regarding the multimodal problems.

Table 5 as well as figures 6 and 7 show no further increase in performance when subpopulations were introduced. Comparisons between the approach with one subpopulation (equal to the standard breeding PSO hybrid) and cases with more than one subpopulation show that the introduction of subpopulations only outperforms the standard breeding PSO hybrid in the Rosenbrock 20-dimensional function. In all other experiments the hybrid model with subpopulations performs worse than the standard PSO model. This is probably because the particles are distributed in several subpopulations which yields a subpopulation size that is too low.

The setting of psb, the probability of breeding within the same subpopulation, could be the cause of the performance deterioration. When the number of subpopulations is increased, the number of particles in each subpopulation is decreased. Having only a few particles in a subpopulation limits the effect of breeding within this subpopulation. Our experiments confirm that it was better to use a lower psb when the number of subpopulations increases, as seen in Table 2. A low psb implies that the probability for breeding between subpopulations is high which of course reduces the effect of subpopulations, in that the amount of gene flow in the total population is kept somewhat constant. These results suggest that the introduction of this specific subpopulation construction to the hybrid model does not generally improve the performance of particle swarms.

6 Conclusions and Future Work

In this paper a hybrid model based on the standard Particle Swarm Optimiser (PSO) and the standard Genetic Algorithm (GA) was introduced. The hybrid model was basically the standard PSO combined with arithmetic crossover. Furthermore, the notion of subpopulations in the hybrid model was introduced, also from the genetic algorithm field.

Four models were used in comparison, namely the standard PSO model, the standard GA and the two hybrid models. Parameters for each model were empirically tuned for each model yielding interesting results regarding the hybrid models. We found that the probability of breeding (pb) for a given particle had its optimum around 0.2. The optimal setting for the probability for breeding between subpopulations (psb) was in our case found to depend on the number of subpopulations. This result indicates that the model would work better with larger subpopulation sizes or other interaction constructions between subpopulations

On unimodal test functions (Sphere and Rosenbrock) the hybrid model was outperformed by the standard PSO and GA models regarding a comparison of the best optima found. Yet, the hybrid model had a marginally faster convergence than both the standard PSO and GA models. On multimodal test functions (Griewank and Rastrigin) the hybrid model performed better. The optima found by the hybrid were better or identical to those of the standard PSO model and the convergence speed was marginally faster.

Future work should cover the grounds of other subpopulation constructions. We chose breeding to model interaction between subpopulations, but other schemes such as migration should be investigated. Larger subpopulationsizes should also be investigated and compared to other evolutionary algorithms that uses subpopulations.

References

[Angeline98] P. J. Angeline, "Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences", Evolutionary Programming VII (1998), Lecture Notes in Computer Science 1447, 601–610. Springer.

[Clerc99] M. Clerc, "The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization", Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1951–1957. IEEE Press.

[Eberhart98] R. C. Eberhart and Y. Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimization", Evolutionary Programming VII (1998), Lecture Notes in Computer Science 1447, 611–616. Springer.

[Kennedy95] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, 1942–1948. IEEE Press.

[Kennedy99] J. Kennedy, "Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance", Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1931–1938. IEEE Press.

[Shi98] Y. Shi and R. C. Eberhart, "Parameter Selection in Particle Swarm Optimization", Evolutionary Programming VII (1998), Lecture Notes in Computer Science 1447, 591–600. Springer.

[Shi98B] Y. Shi and R. C. Eberhart, "A modified Particle Swarm Optimiser", IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.

[Shi99] Y. Shi and R. C. Eberhart, ”Empirical Study of Particle Swarm Optimization”, Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1945–1950. IEEE Press.

[Spears94] W. M. Spears, ”Simple subpopulation schemes”, Proceedings of the Evolutionary Programming Conference 1994, pp. 296-307.

[Suganthan99] P. N. Suganthan, ”Particle Swarm Optimiser with Neighbourhood Operator”, Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1958–1962. IEEE Press.