Differential Evolution for Adaptive System of Particle Swarm Optimization with Genetic Algorithm

Pham Ngoc Hieu[1] and Hiroshi Hasegawa[2]
[1]*Functional Control Systems, Graduate School of Engineering and Science, Shibaura Institute of Technology, Saitama, Japan*
[2]*College of Systems Engineering and Science, Shibaura Institute of Technology, Saitama, Japan*

Keywords: Adaptive System, Differential Evolution (DE), Genetic Algorithm (GA), Multi-peak Problems, Particle Swarm Optimization (PSO).

Abstract: A new strategy using Differential Evolution (DE) for Adaptive Plan System of Particle Swarm Optimization (PSO) with Genetic Algorithm (GA) called DE-PSO-APGA is proposed to solve a huge scale optimization problem, and to improve the convergence towards the optimal solution. This is an approach that combines the global search ability of GA and Adaptive plan (AP) for local search ability. The proposed strategy incorporates concepts from DE and PSO, updating particles not only by DE operators but also by mechanism of PSO for Adaptive System (AS). The DE-PSO-APGA is applied to several benchmark functions with multi-dimensions to evaluate its performance. We confirmed satisfactory performance through various benchmark tests.

1 INTRODUCTION

Several modern heuristic algorithms as population-based algorithms Evolutionary Algorithms (EAs) have been developed for solving complex numerical optimization. The most popular EA, Genetic Algorithm (GA) (Goldberg, 1989) has been applied to various multi-peak optimization problems. The validity of this method has been reported by many researchers. However, it requires a huge computational cost to obtain stability in the convergence to an optimal solution. To reduce the cost and to improve stability, a strategy that combines global and local search methods becomes necessary. As for this strategy, Hasegawa et al. proposed a new evolutionary algorithm called an Adaptive Plan system with Genetic Algorithm (APGA) (Hasegawa, 2007).

Among the modern meta-heuristic algorithms, a well-known branch is Particle Swarm Optimization (PSO), first introduced by Kennedy and Eberhart (2001). It has been developed through simulation of a simplified social system, and has been found to be robust in solving optimization problems. Nevertheless, the performance of the PSO greatly depends on its parameters and it often suffers from the problem of being trapped in the local optimum. To resolve this problem, various improvement algorithms are proposed for solving a variety of optimal problems (Clerc and Kennedy, 2002).

A new evolutionary algorithm known as Differential Evolutionary (DE) was recently introduced and has garnered significant attention in the research literature (Storn and Price, 1997). Compared with other techniques and EAs, it hardly requires any parameter tuning and is very efficient and reliable. As PSO has memory, knowledge of good solutions is retained by all particles, whereas in DE, previous knowledge of the problem is discarded once the population changes. Moreover PSO and DE both work with an initial population of solutions. Therefore, combining the searching ability of these methods seems to be a reasonable approach (Das et al., 2008).

In this paper, we purposed a new strategy using DE for Adaptive Plan system of PSO with GA to solve a huge scale optimization problem, and to improve the convergence to the optimal solution called DE-PSO-APGA.

The remainder of this paper is organized as follows. Section 2 describes the basic concepts of DE, Section 3 explains the algorithm of proposed strategy (DE-PSO-APGA), and Section 4 discusses about the convergence to the optimal solution of multi-peak benchmark functions. Finally Section 5 includes some brief conclusions.

Ngoc Hieu P. and Hasegawa H..
Differential Evolution for Adaptive System of Particle Swarm Optimization with Genetic Algorithm.
DOI: 10.5220/0004107202590264
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (ECTA-2012), pages 259-264
ISBN: 978-989-8565-33-4

2 DIFFERENTIAL EVOLUTION

Differential Evolution (DE) is an EA proposed by Storn and Price (1997), also a population-based heuristic algorithm, which is simple to implement, requires little or no parameter tuning and is known for its remarkable performance for combinatorial optimization.

2.1 Basic Concepts of DE

DE is similar to other EAs particularly GA in the sense that it uses the same evolutionary operators such as selection, recombination, and mutation. However the significant difference is that DE uses distance and direction information from the current population to guide the search process. The performance of DE depends on the manipulation of target vector and difference vector in order to obtain a trial vector.

2.1.1 Mutation

Mutation is the main operator in DE. For a D-dimensional search space, each target vector $X_{i,G}$, the most useful strategies of a mutant vector are:

DE/rand/1

$$V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \tag{1}$$

DE/best/1

$$V_{i,G} = X_{best,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \tag{2}$$

DE/target to best/1

$$\begin{aligned} V_{i,G} = X_{i,G} &+ F \cdot (X_{best,G} - X_{i,G}) \\ &+ F \cdot (X_{r_2,G} - X_{r_3,G}) , \end{aligned} \tag{3}$$

DE/best/2

$$\begin{aligned} V_{i,G} = X_{best,G} &+ F \cdot (X_{r_1,G} - X_{r_2,G}) \\ &+ F \cdot (X_{r_3,G} - X_{r_4,G}) , \end{aligned} \tag{4}$$

DE/rand/2

$$\begin{aligned} V_{i,G} = X_{r_1,G} &+ F \cdot (X_{r_2,G} - X_{r_3,G}) \\ &+ F \cdot \left(X_{r_4,G} - X_{r_5,G}\right) , \end{aligned} \tag{5}$$

where r_1, r_2, r_3, r_4, $r_5 \in [1,2,\ldots,NP]$ are mutually exclusive randomly chosen integers with a initiated population of NP, and all are different from the base index i. G denotes subsequent generations, and $F > 0$ is a scaling factor which controls the amplification of differential evolution. $X_{best,G}$ is the best individual vector with the best fitness (lowest objective function value for a minimization) in the population.

DE/rand/2/dir

$$V_{i,G} = X_{r_1,G} + F/2 \cdot (X_{r_1,G} - X_{r_2,G} - X_{r_3,G}) \tag{6}$$

DE/rand/2/dir (Feoktistov and Janaqi, 2004) incorporates the objective function information to guide the direction of the donor vectors. $X_{r_1,G}$, $X_{r_2,G}$, and $X_{r_3,G}$ are distinct population members such that $f(X_{r_1,G}) \leq \{f(X_{r_2,G}), f(X_{r_3,G})\}$.

2.1.2 Crossover

To enhance the potential diversity of the population, a crossover operation is introduced. The donor vector exchanges its components with the target vector to form the trial vector:

$$U_{ji,G+1} = \begin{cases} V_{ji,G+1}, (rand_j \leq CR) \; or \; (j = j_{rand}) \\ X_{ji,G+1}, (rand_j \geq CR) \; and \; (j \neq j_{rand}) \end{cases} \tag{7}$$

where $j = [1,2,\ldots,D]$; $rand_j \in [0.0,1.0]$; CR is the crossover probability takes value in the range [0.0,1.0], and $j_{rand} \in [1,2,\ldots,D]$ is the randomly chosen index.

2.1.3 Selection

Selection is performed to determine whether the target vector or the trial vector survives to the next generation. The selection operation is described as:

$$X_{i,G+1} = \begin{cases} U_{i,G}, & f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, & f(U_{i,G}) > f(X_{i,G}) \end{cases} \tag{8}$$

2.2 Variants of DE

In this section, we discuss about an introduction of the most prominent DE variants that were developed and appeared to be competitive against the existing best-known real parameter optimizers. Some of these variants are:

DE using Arithmetic Recombination (Price et al., 2005)

$$\begin{aligned} V_{i,G} = X_{i,G} &+ k_i \cdot (X_{r_1,G} - X_{i,G}) \\ &+ F' \cdot (X_{r_2,G} - X_{r_3,G}) \end{aligned} \tag{9}$$

where k_i is the combination coefficient, which can be a constant or a random variable distribution from [0.0,1.0], and $F' = k_i \cdot F$ is a new constant parameter.

DE/rand/1/either-or

$$V_{i,G} = \begin{cases} X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}), \; rand_i(0,1) < p_F \\ X_{r_0,G} + k \cdot (X_{r_1,G} + X_{r_2,G} - 2X_{r_0,G}) \end{cases} \tag{10}$$

Price et al. (2005) proposed the state-of-the-art where the trial vectors that are pure mutants occur with a probability p_F and those that are pure recombinants occur with a probability $1 - p_F$ $(0.0 < p_F < 1.0)$.

Note that p_F is a parameter that determines the relative importance of a mutation and arithmetic recombination schemes, Price et al. recommended a value 0.4 for it, and the parameter $k = 0.5 \cdot (F + 1)$ as a good choice for a given F.

3 DE-PSO-APGA STRATEGY

With a view to global search, we proposed the new algorithm using DE for Adaptive Plan system of PSO with GA named DE-PSO-APGA. The DE-PSO-APGA aims at getting the direction from PSO operator to adjust into adaptive system of APGA based on alternative operators of DE scheme. In addition, for a verification of APGA search process, refer to Ref. (Hasegawa, 2007).

3.1 Algorithm

The proposed DE-PSO-APGA starts like the usual DE algorithm up to the point where the trial vector is generated. If the trial vector satisfies the conditions given by (8), then the algorithm enters the PSO operator to get the direction and generates a new candidate solution with adaptive system of APGA. The inclusion of APGA process turns helps in maintaining diversity of the population and reaching a global optimal solution.

Pseudocode of DE-PSO-APGA

```
begin
   initialize population;
   fitness evaluation;
   repeat until (termination) do
      DE update strategies;
      PSO activated;
      APGA process;
   end do
   renew population;
end.
```

DE Update Strategies

- DE-PSO-APGA1 by (1);
- DE-PSO-APGA2 by (2);
- DE-PSO-APGA3 by (3);
- DE-PSO-APGA4 by (4);
- DE-PSO-APGA5 by (5);
- DE-PSO-APGA6 by (6);
- DE-PSO-APGA7 by (9);
- DE-PSO-APGA8 by (10);

PSO Operator

We are concerned here with conventional basic model of PSO (Kennedy and Eberhart, 2001). In this model, each particle which make up a swarm has information of its position x_i and velocity v_i (i is the index of the particle) at the present in the search space. Each particle aims at the global optimal solution by updating next velocity making use of the position at the present, based on its best solution has been achieved so far p_{ij} and the best solution of all particles g_j ($j = [1,2,\ldots,D]$, D is the dimension of the solution vector), as following equation:

$$v_{ij}^{G+1} = w v_{ij}^{G} + c_1 r_1 \left(p_{ij}^{G} - X_{ij}^{G}\right) + c_2 r_2 \left(g_j^{G} - X_{ij}^{G}\right) \tag{11}$$

where w is inertia weight; c_1 and c_2 are cognitive acceleration and social acceleration, respectively; r_1 and r_2 are random numbers uniformly distributed in the range [0.0, 1.0].

In our strategy, the concept of time-varying has been adapted (Shi and Eberhart, 1999). The inertia weight w in (11) linearly decreasing with the iterative generation as below:

$$w = (w_{\max} - w_{\min}) \frac{iter_{\max} - iter}{iter_{\max}} + w_{\min} \tag{12}$$

where $iter$ is the current iteration number while $iter_{\max}$ is the maximum number of iterations, the maximal and minimal weights $w_{\max}$ and $w_{\min}$ are respectively set 0.9, 0.4 known from experience.

The acceleration coefficients are expressed as:

$$c_1 = (c_{1f} - c_{1i}) \frac{iter_{\max} - iter}{iter_{\max}} + c_{1i} \tag{13}$$

$$c_2 = (c_{2f} - c_{2i}) \frac{iter_{\max} - iter}{iter_{\max}} + c_{2i} \tag{14}$$

where c_{1i}, c_{1f}, c_{2i} and c_{2f} are initial and final values of the acceleration coefficient factors respectively. The most effective values are set 2.5 for c_{1i} and c_{2f} and 0.5 for c_{1f} and c_{2i} as in (Eberhart and Shi, 2000).

APGA Process

Adaptive Plan with Genetic Algorithm (APGA) (Hasegawa, 2007) that combines the global search ability of a GA and an Adaptive Plan (AP) with excellent local search ability is superior to other EAs, Memetic Algorithms (MAs) (Smith et al., 2005). The APGA concept differs in handling DVs from general EAs based on GAs. EAs generally encode DVs into the genes of a chromosome, and handle them through GA operators. However, APGA completely separates DVs of global search and local search methods. It encodes Control variable vectors (CVs) of AP into its genes on Adaptive system (AS). Moreover, this separation strategy for DVs and chromosomes can solve

MA problem of breaking chromosomes. The control variable vectors (CVs) steer the behavior of adaptive plan (AP) for a global search, and are renewed via genetic operations by estimating fitness value. For a local search, AP generates next values of DVs by using CVs, response value vectors (RVs) and current values of DVs according to the formula:

$$X_{i,G+1} = X_{i,G} + AP(C_G, R_G) \quad (15)$$

where $AP()$, X, C, R, and G denote a function of AP, DVs, CVs, RVs and generation, respectively.

3.2 Adaptive Plan (AP)

It is necessary that the AP realizes a local search process by applying various heuristics rules. In this paper, the plan introduces a DV generation formula using velocity update from PSO operator that is effective in the convex function problem as a heuristic rule, because a multi-peak problem is combined of convex functions. This plan uses the following equation:

$$AP(C_G, R_G) = scale \cdot SP \cdot PSO \cdot (\nabla R) \quad (16)$$

$$SP = 2 \cdot C_G - 1 \quad (17)$$

where ∇R denote sensitivity of RVs, constriction factor *scale* randomly selected from a uniform distribution in [0.1,1.0], and velocity update *PSO* by (11).

Step size *SP* is defined by CVs for controlling a global behavior to prevent it falling into the local optimum. $C = [c_{i,j}, \ldots, c_{i,p}]$; $(0.0 \leq c_{i,j} \leq 1.0)$ is used so that it can change the direction to improve or worsen the objective function, and C is encoded into a chromosome by 10 bit strings (shown in Figure 1). In addition, i, j and p are the individual number, design variable number and its size, respectively.

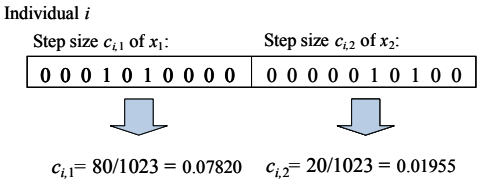


Figure 1: Encoding into genes of a chromosome.

3.3 GA Operators

3.3.1 Selection

Selection is performed using a tournament strategy to maintain the diverseness of individuals with a goal of keeping off an early convergence. A tournament size of 2 is used.

3.3.2 Elite Strategy

An elite strategy, where the best individual survives in the next generation, is adopted during each generation process. It is necessary to assume that the best individual, i.e., as for the elite individual, generates two behaviors of AP by updating DVs with AP, not GA operators. Therefore, its strategy replicates the best individual to two elite individuals, and keeps them to next generation. As shown in Figure 2, DVs of one of them (Δ symbol) is renewed by AP, and its CVs which are coded into chromosome arent changed by GA operators. Another one ($\circ$ symbol) is that both DVs and CVs are not renewed, and are kept to next generation as an elite individual at the same search point.

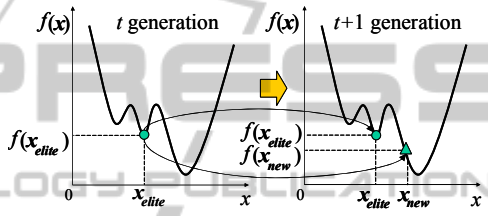
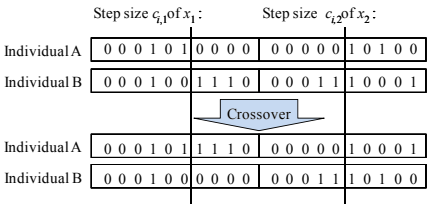


Figure 2: Elite strategy.

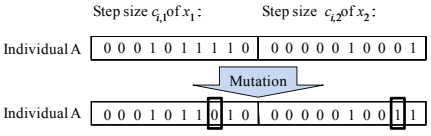
3.3.3 Crossover and Mutation

In order to pick up the best values of each CV, a single point crossover is used for the string of each CV. This can be considered to be a uniform crossover for the string of the chromosome as shown in Figure 3(a).

Mutation are performed for each string at mutation ratio on each generation, and set to maintain the strings diverse as shown in Figure 3(b).



(a)



(b)

Figure 3: Crossover and Mutation.

3.3.4 Recombination

At following conditions, the genetic information on chromosome of individual is recombined by uniform random function.

- One fitness value occupies 80% of the fitness of all individuals;
- One chromosome occupies 80% of the population.

4 NUMERICAL EXPERIMENTS

The numerical experiments are performed 25 trials for every function. The initial seed number is randomly varied during every trial. The population size is 100 individuals and the terminal generation is 1500 generations. Parameters setting for the experiments are given in Table 1.

4.1 Benchmark Functions

We estimate the stability of the convergence to the optimal solution by using five benchmark functions - Rastrigin (RA), Griewank (GR), Ridge (RI), Ackley (AC), and Rosenbrock (RO). These functions are given as follows:

$$RA : f_1 = 10n + \sum_{i=1}^{n} \{x_i^2 - 10cos(2\pi x_i)\} \quad (18)$$

$$RI : f_2 = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2 \quad (19)$$

$$GR : f_3 = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}} \right) \quad (20)$$

$$AC : f_4 = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e \quad (21)$$

$$RO : f_5 = \sum_{i=1}^{n} [100(x_{i+1} + 1 - (x_i + 1)^2)^2 + x_i^2] \quad (22)$$

Table 2 summarizes their characteristics, and design range of DVs. The terms epistasis, peaks, steep denote the dependence relation of the DVs, presence of multi-peak and level of steepness, respectively. All functions are minimized to zero ($ESP = 1.7e^{-308}$), when optimal DVs $X = 0$ are obtained.

Table 1: Setting parameters for solving Benchmarks.

	Control Parameter	Set value
DE	Scaling factor	$F \in [0.1, 1.0]$
	Crossover	$CR = 0.5$
PSO	Inertia weight	$w_{\max} = 0.9$
		$w_{\min} = 0.4$
	Coefficients	$c_{1i} = c_{2f} = 2.5$
		$c_{1f} = c_{2i} = 0.5$
GA	Selection	0.1
	Crossover	0.8
	Mutation	0.01
Population size 100; Terminal generation 1500		

Table 2: Benchmark functions and design range.

Func	Epistasis	Peaks	Steep	DVs range
RA	No	Yes	Average	[-5.12,5.12]
RI	Yes	No	Average	[-100,100]
GR	Yes	Yes	Small	[-600,600]
AC	No	Yes	Average	[-32,32]
RO	Yes	No	Big	[-30,30]

4.2 Experiment Results

The experiment results, averaged over 25 trials with RO function are shown in Table 3. The solutions of all strategies reach their global optimum solutions. When the success rate of optimal solution is not 100%, "-" is described.

From the results via optimization experiments, we employed DE-PSO-APGA2 using DE/best/1 update strategy for the DE-PSO-APGA algorithm, and recommended a value 0.1 for the scaling factor F. Additionally, the average results of all benchmark functions with 30 and 100 dimensions by the DE-PSO-APGA are given in Table 4. The success rate of optimal solution is 100% with all benchmark functions. "Mean" indicates average of minimum values obtained, and "Std." stands for standard deviation.

In summary, its validity confirms that this strategy can reduce the computation cost and improve the stability of the convergence to the optimal solution.

4.3 Comparison

Table 5 shows the comparison of PSO, DE, DE-PSO (Pant et al., 2008) and the DE-PSO-APGA. As a result, DE-PSO-APGA algorithm outperformed PSO, DE and DE-PSO in all benchmark functions. Therefore, it is desirable to introduce this method for the new evolution strategy.

Overall, DE-PSO-APGA was capable of attaining robustness, high quality, low calculation cost and ef-

Table 3: Average of minimum values obtained over 25 trials with RO function ($D = 30$, population size 50, $CR = 0.5$).

Strategy	Scaling factor F									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
DE-PSO-APGA1	-	-	-	-	-	-	-	-	-	-
DE-PSO-APGA2	0	0	0	0	0	0	0	-	-	-
DE-PSO-APGA3	2.96e-30	5.06e-30	2.9e-197	2.9e-197	2.8e-197	0	0	-	-	-
DE-PSO-APGA4	0	0	0	0	0	-	-	-	-	-
DE-PSO-APGA5	-	-	-	-	-	-	-	-	-	-
DE-PSO-APGA6	5.31e-30	1.6e-118	-	-	-	-	-	-	-	-
DE-PSO-APGA7	-	-	-	-	-	-	-	-	-	-
DE-PSO-APGA8	-	-	-	-	-	-	-	-	-	-

Table 4: Results by DE-PSO-APGA with population size 100 ($F = 0.1$, $CR = 0.5$). Mean indicates average of minimum values obtained, "Std." stands for standard deviation.

Func	Dim.	Max Gen.	Mean (Std.)
RA	30	500	0.00e+00 (0.00e+00)
	100	1500	0.00e+00 (0.00e+00)
RI	30	500	0.00e+00 (0.00e+00)
	100	1500	0.00e+00 (0.00e+00)
GR	30	500	0.00e+00 (0.00e+00)
	100	1500	0.00e+00 (0.00e+00)
AC	30	500	4.44e-16 (0.00e+00)
	100	1500	4.44e-16 (0.00e+00)
RO	30	500	0.00e+00 (0.00e+00)
	100	1500	0.00e+00 (0.00e+00)

Table 5: Comparison results of PSO, DE, DE-PSO and DE-PSO-APGA ($D = 30$, population size 30, max generation 3000).

Func	PSO	DE	DE-PSO	DE-PSO -APGA
RA	37.82 (7.456)	2.531 (5.19)	1.614 (3.885)	0 (0)
RI	- -	- -	- -	0 (0)
GR	0.018 (0.023)	0 (0)	0 (0)	0 (0)
AC	1.0e-08 (1.9e-08)	7.3e-15 (7.7e-16)	3.7e-15 (0)	4.4e-16 (0.0e+0)
RO	81.27 (41.22)	31.14 (17.12)	24.2 (12.31)	0 (0)

ficient performance on many benchmark problems.

5 CONCLUSIONS

To overcome the computational complexity, a new strategy using DE for Adaptive Plan system of PSO with GA called DE-PSO-APGA has been proposed to solve a huge scale optimization problem, and to improve the convergence to the optimal solution. Then, we verify the effectiveness of the DE-PSO-APGA by the numerical experiments performed five benchmark functions.

We can confirm that the DE-PSO-APGA dramatically reduces the calculation cost and improves the convergence towards the optimal solution.

REFERENCES

Clerc, M. and Kennedy, J. (2002). The particle swarm-explosion, stability and convergence in a multidimensional complex space. In *IEEE Trans. Evol. Comput.*

Das, S., Abraham, A., and Konar, A. (2008). Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives. In *Studies in Computational Intelligence (SCI)*.

Eberhart, R. C. and Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proc. Congr. Evol. Comput.*

Feoktistov, V. and Janaqi, S. (2004). Generalization of the strategies in differential evolution. In *Proc. 18th IPDPS*.

Goldberg, D. E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley.

Hasegawa, H. (2007). Adaptive plan system with genetic algorithm based on synthesis of local and global search method for multi-peak optimization problems. In *6th EUROSIM Congress on Modelling and Simulation*.

Kennedy, J. and Eberhart, R. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers.

Pant, M., Thangaraj, R., Grosan, C., and Abraham, A. (2008). Hybrid differential evolution - particle swarm optimization algorithm for solving global optimization problems. In *Third International Conference on Digital Information Management (ICDIM)*.

Price, K., Storn, R., and Lampinen, J. (2005). *Differential Evolution - A Practical Approach to Global Optimization*. Springer, Berlin, Germany.

Shi, Y. and Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proc. IEEE Int. Congr. Evol. Comput.*

Smith, J. E., Hart, W. E., and Krasnogor, N. (2005). *Recent Advances in Memetic Algorithms*. Springer.

Storn, R. and Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. In *Journal Global Optimization*.