# Tuning PSO Parameters Through Sensitivity Analysis

**Thomas Beielstein[1], Konstantinos E. Parsopoulos[2], Michael N. Vrahatis[2]**

[1]Department of Computer Science XI, University of Dortmund, D–44221 Dortmund, Germany.
thomas.beielstein@LS11.cs.uni-dortmund.de
[2]Department of Mathematics, Artificial Intelligence Research Center (UPAIRC), University of Patras, GR–26110 Patras, Greece
{kostasp, vrahatis}@math.upatras.gr

**Abstract- In this paper, a first analysis of the Particle Swarm Optimization (PSO) method's parameters, using Design of Experiments (DOE) techniques, is performed, and important settings as well as interactions among the parameters, are investigated (screening).**

## 1 INTRODUCTION

Computer experiments and simulations play an important role in scientific research, since many real–world processes are so complex that physical experimentation is either time consuming or expensive. Simulation models often need high–dimensional inputs and they are computationally expensive. Thus, the simulation practitioner is interested in the development of simulation models, that enable him to perform the simulation with the minimum amount of simulation runs. To achieve this goal, design of experiments (DOE) methods have been developed and successfully applied in the last decades.

The DOE techniques can be applied to optimization algorithms, considering the run of an algorithm as an *experiment*, gaining insightful conclusions into the behavior of the algorithm and the interaction and significance of its parameters. In this paper, the parameters of the Particle Swarm Optimization method (PSO) are investigated using DOE techniques.

The rest of the paper is organized as follows: the PSO method and the DOE techniques are briefly described in Sec. 2 and 3, respectively. An experimental design (DESIGN I) is chosen and experiments (simulation runs) are performed. This design is discussed in detail in Sec. 4, and an improved design (DESIGN II) is developed. The analysis results in recommendations for the parameter setting of PSO on a special test function, that are statistically validated.

## 2 PARTICLE SWARM OPTIMIZATION

The term "Swarm Intelligence", is used to describe algorithms and distributed problem solvers inspired by the collective behavior of insect colonies and other animal societies [BDT99, KE01]. Under this prism, PSO is a Swarm Intelligence method for solving optimization problems. Its precursor was a simulator of social behavior, that was used to visualize the movement of a birds' flock. Several versions of the simulation model were developed, incorporating concepts such as nearest–neighbor velocity matching and acceleration by distance [ESD96, KE95]. When it was realized that the simulation could be used as a population–based optimizer, several parameters were omitted, through a trial and error process, resulting in the first simple version of PSO [EK95, ESD96].

Suppose that the search space is $D$–dimensional, then the $i$–th individual, which is called $particle$, of the population, which is called $swarm$, can be represented by a $D$–dimensional vector, $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^\top$. The *velocity* (position change) of this particle, can be represented by another $D$–dimensional vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^\top$. The best previously visited position of the $i$–th particle is denoted as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^\top$. Defining $g$ as the index of the best particle in the swarm (i.e. the $g$–th particle is the best), and let the superscripts denote the iteration number, then the swarm is manipulated according to the following two equations [ESD96]:

$$\begin{aligned} v_{id}^{n+1} &= v_{id}^n + c r_1^n (p_{id}^n - x_{id}^n) + c r_2^n (p_{gd}^n - x_{id}^n), \quad (1) \\ x_{id}^{n+1} &= x_{id}^n + v_{id}^{n+1}, \quad (2) \end{aligned}$$

where $d = 1, 2, \ldots, D$; $i = 1, 2, \ldots, N$, and $N$ is the size of the swarm; $c$ is a positive constant, called *acceleration constant*; $r_1$, $r_2$ are random numbers, uniformly distributed in $[0, 1]$; and $n = 1, 2, \ldots$, determines the iteration number.

However, in the first version of PSO, there was no actual control mechanism for the velocity. This could result in inefficient behavior of the algorithm, especially in the neighborhood of the global minimum. In a second version of PSO, this shortcoming was addressed by incorporating new parameters, called *inertia weight* and *constriction factor*. This version is described by the equations [ES98, SE98b, SE98a]:

$$\begin{aligned} v_{id}^{n+1} &= \chi \left( w v_{id}^n + c_1 r_1^n (p_{id}^n - x_{id}^n) + \right. \\ &\quad \left. + c_2 r_2^n (p_{gd}^n - x_{id}^n) \right), \quad (3) \\ x_{id}^{n+1} &= x_{id}^n + v_{id}^{n+1}, \quad (4) \end{aligned}$$

where $w$ is called *inertia weight*; $c_1$, $c_2$ are two positive constants, called *cognitive* and *social* parameter respectively; and $\chi$ is a *constriction factor*, which is used to limit velocity.

The role of the *inertia weight* $w$, in Eq. (3), is considered critical for the PSO's convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current one. Accordingly, the parameter $w$ regulates the trade–off between the global (wide–ranging) and local (nearby) exploration abilities of the swarm. A large inertia weight facilitates global exploration (searching new areas), while a small one tends to facilitate local exploration, i.e. fine–tuning the current search area. A suitable value for the inertia weight $w$ usually provides balance between global and local exploration abilities and consequently results in a reduction of the number of iterations required to locate the optimum solution. Initially, the inertia weight was constant. However, experimental results indicated that it is better to initially set the inertia to a large value, in order to promote global exploration of the search space, and gradually decrease it to get more refined solutions [SE98b, SE98a]. Thus, an initial value around $1.2$ and a gradual decline towards $0$ can be considered as a good choice for $w$ .

Proper fine–tuning of the parameters $c_1$ and $c_2$, in Eq. (3), may result in faster convergence of the algorithm, and alleviation of the local minima. An extended study of the acceleration parameter in the first version of PSO, is given in [Ken98]. As default values, $c_1 = c_2 = 2$ were proposed, but experimental results indicate that $c_1 = c_2 = 0.5$ might provide even better results. Recent work reports that it might be even better to choose a larger cognitive parameter, $c_1$, than a social parameter, $c_2$, but with $c_1 + c_2 \leqslant 4$ [CD01].

The parameters $r_1$ and $r_2$ are used to maintain the diversity of the population, and they are uniformly distributed in the range $[0, 1]$. The constriction factor $\chi$ controls on the magnitude of the velocities, in a way similar to the $V_{max}$ parameter, used in the first versions of PSO. In some cases, using both $\chi$ and $V_{max}$ may result in faster convergence rates. In all experiments in this paper, the PSO method described in Eqs. 3 and 4, with $\chi = 1$ was used.

Although PSO is an evolutionary technique, it differs from other evolutionary algorithm (EA) techniques. Three main operators are usually involved in EA techniques. The *recombination*, the *mutation* and the *selection* operator. PSO does not have a direct recombination operator. However, the stochastic acceleration of a particle towards its previous best position, as well as towards the best particle of the swarm (or towards the best in its neighborhood in the local version), resembles the recombination procedure in EA [ES98, Rec94, Sch75, Sch95]. In PSO the information exchange takes place only among the particle's own experience and the experience of the best particle in the swarm, instead of being carried from fitness dependent selected "parents" to descendants as in GA's. Moreover, PSO's directional position updating operation resembles mutation of GA, with a kind of memory built in. This mutation–like procedure is multi-directional both in PSO and GA, and it includes control of the mutation's severity, utilizing factors such as the $V_{max}$ and $\chi$.

| Symbol | Parameter |
|---|---|
| $D$ | search–space dimension |
| $N$ | swarm size |
| $v_0$ | initial velocity |
| $c_1$ | cognitive parameter |
| $c_2$ | social parameter |
| $w$ | inertial weight (max value) |
| $w_{\text{scale}}$ | scaling factor for inertia weight |
| $f$ | fitness function, optimization problem |
| $D$ | dimension of $f$ |
| $n_{\max}$ | maximum number of iterations |
| $N_{\text{exp}}$ | number of experiments for each scenario |
| $N_{\text{tot}}$ | total number of fitness function evaluations |
| $\sigma$ | noise level |
| $N_{\text{reeval}}$ | number of re-evaluations (noise) |

Table 1: DOE parameter.

| Var | Factor | I: Level | II: Level |
|---|---|---|---|
| $N$ | $A$ | $\{600, 900, 1200\}$ | $\{10, 30^{*}, 50\}$ |
| $w$ | $B$ | $\{0.6, 0.9, 1.2\}$ | $\{0.3, 0.6^{*}, 0.9\}$ |
| $w_{\text{scale}}$ | $C$ | $\{0.0, 0.25, 0.5\}$ | $\{0.0, 0.25, 0.5^{*}\}$ |
| $c_1$ | $D$ | $\{0.5, 1.0, 2.0\}$ | $\{0.5, 1.0^{*}, 2.0\}$ |
| $c_2$ | $E$ | $\{0.5, 1.0, 2.0\}$ | $\{1.75, 2.25^{*}, 2.75\}$ |

Table 2: DESIGN I and DESIGN II.

PSO belongs to the class of EAs, that does not use the "survival of the fittest" concept. It does not utilize a direct selection function. Thus, particles with lower fitness can survive during the optimization and potentially visit any point of the search space [ES98].

# 3 EXPERIMENTAL DESIGN

Experimental design provides an excellent way of deciding which simulation runs should be performed so that the desired information can be obtained with the least amount of experiments [BHH78, BD87, Kle87, KVG92, Kle99, LK00]. The input parameters and structural assumptions, that define a simulation model are called *factors*, the output value(s) are called *response(s)*. The different values of parameters are called *levels*. An experimental design is a set of factor level combinations. As mentioned in Sec. 2, fine–tuning of PSO parameters is an important task. The role of the inertia weight $w$, the relationship between the cognitive and the social parameters $c_1$ and $c_2$ or the determination of the swarm size are important to ensure convergence of the PSO algorithm. Applying the DOE model assumptions to PSO algorithms, we can define the parameter vector (cp. Tab. 1):

$$p = (N, v_0, c_1, c_2, w, w_{\text{scale}})^T \quad (5)$$

and the design vector:

$$d = (f, D, n_{\max}, N_{\text{exp}}, N_{\text{tot}}, N_{\text{reeval}}, \sigma)^T. \quad (6)$$

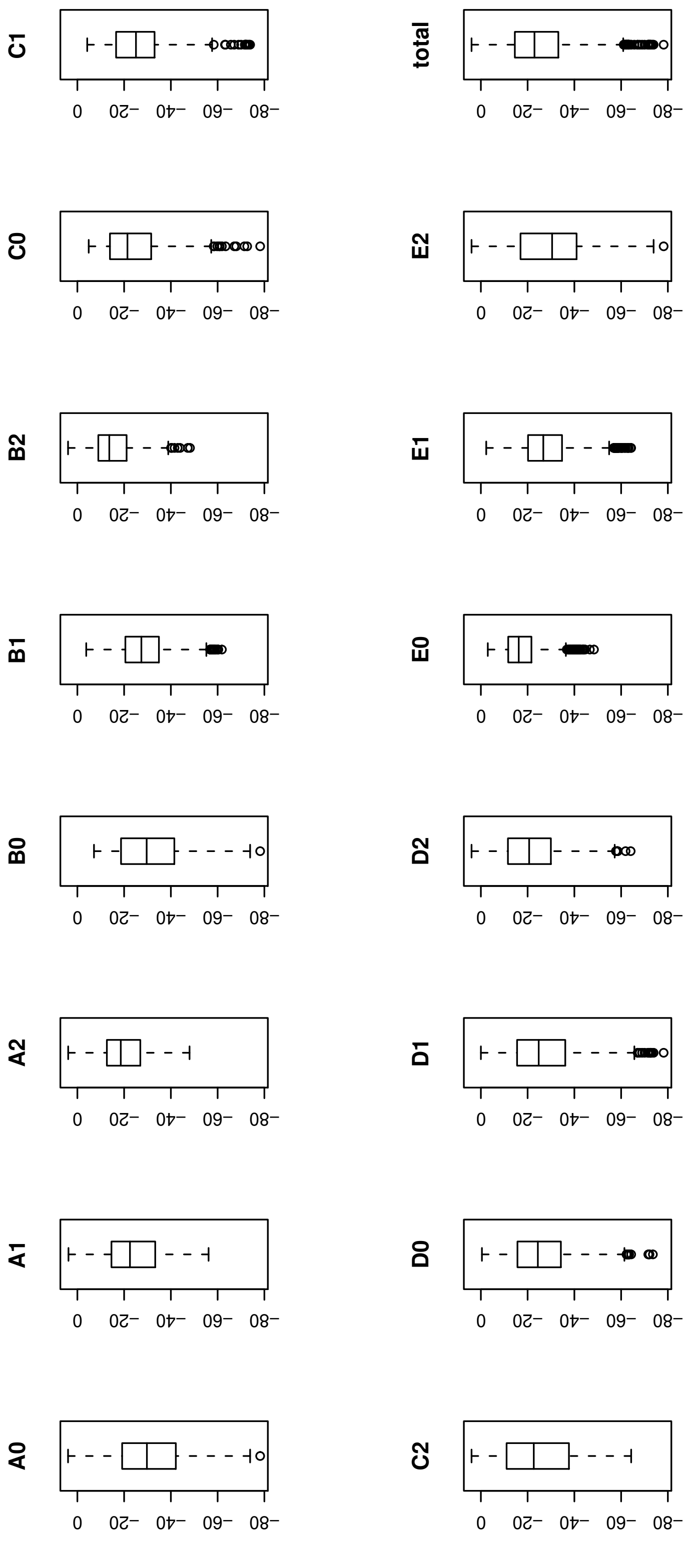


Figure 1: DESIGN I. Box-plot

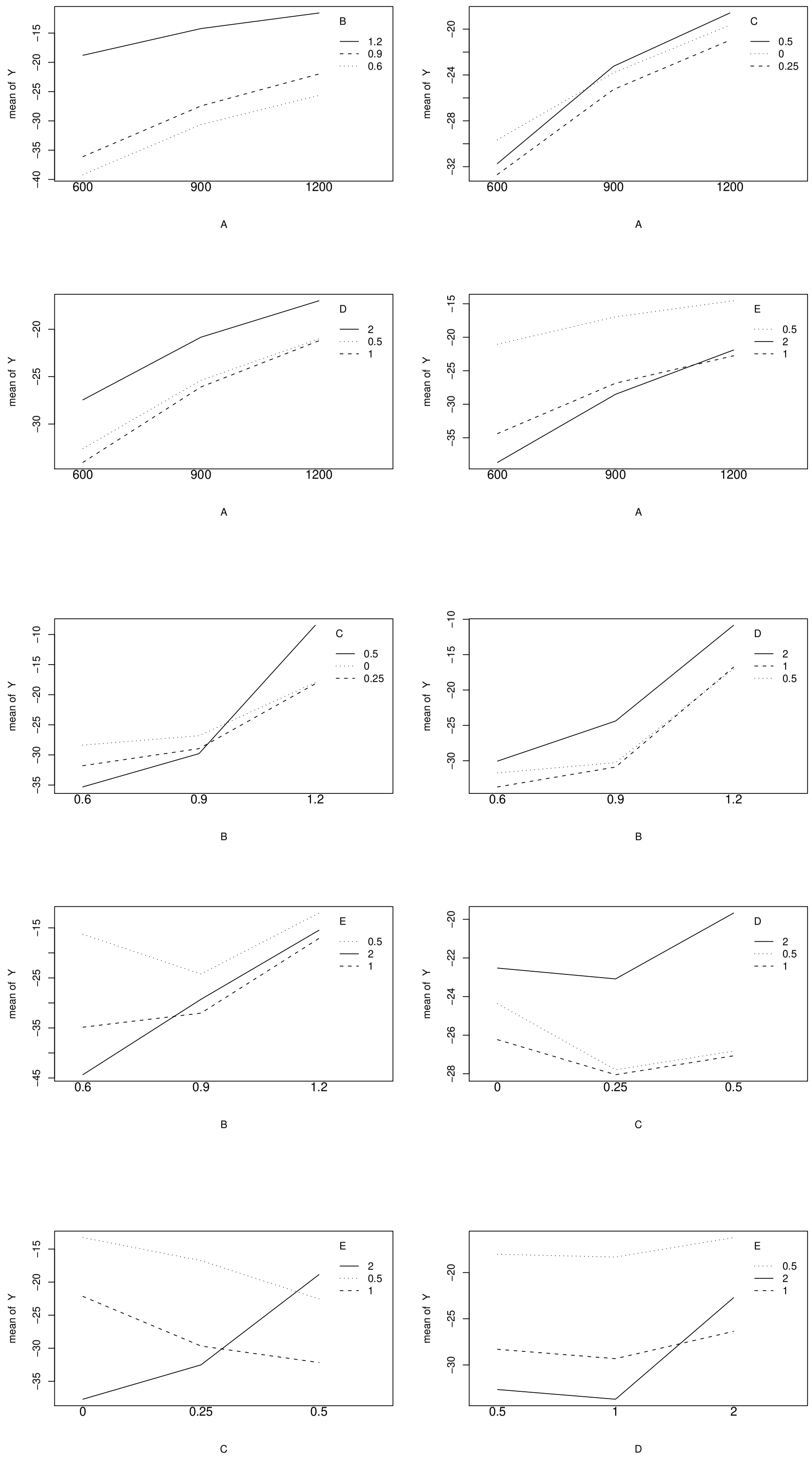


Figure 2: DESIGN I. Interaction plot

In the next section, we will apply DOE methods to improve the behavior of the PSO algorithm for a given optimization configuration. This task can be interpreted as the determination of optimal values of $p$ for a given problem design $d$ or as an analysis of a *regression meta-model* [KVG92].

## 4 SENSITIVITY ANALYSIS APPLIED TO PSO

Starting point for the first experiments is the minimization of the sphere function $f(x) = \sum_{i=1}^{D} x_i^2$, were $x \in \mathbb{R}^D$. Although the number of simulation runs required increases geometrically with $k$ for a $l^k$ factorial design as $k$ is increased, we chose a $3^5$ full factorial design for didactical purposes. $3^l$ factorial designs clarifies the geometrical interpretation of the results. Neglecting higher order interactions, fractional factorial designs reduce the required amount of simulation runs drastically [BHH78]. Efficiency and effectiveness of designs are discussed in [Kle87, KVG92]. The corresponding factor levels are shown in Tab. 2. To apply analysis of variance (ANOVA) techniques, we have to verify classical assumptions, e.g. the 'normality of errors' [DS98].

Examination of the fit reveals, that the main effects and some first order interactions are significant, but that it is necessary to use a response transformation $\ln Y$. After checking the fitted regression model, we investigate the effects and their interactions[1].

Box-plots provide an excellent visualization of the effects of a change of one factor level. For a factor $X$, $X0$ denotes the low, $X1$ the medium, and $X2$ the highest value. Regarding the swarm size $A$ in DESIGN I, we obtain: $A0$: $600$, $A1$: $900$, and $A2$: $1200$. Thus we can conclude from the first three plots in Fig. 1, that the swarm size should be set to the low $A0$ level. $B0$ outperforms the other $B$ factor level settings, but $C$ and $D$ should be set to their medium level values ($w_{\text{scale}} = 0.25$ resp. $c_1 = 1.0$), whereas $E$ should be set to its maximum value $2.0$.

Fig. 2 shows the interactions between two factors. The first figure reveals, that factor $A$ performs best at its low level $600$ – no matter how we set factor $B$. Obviously performs $B$ best, if we choose its low level $0.6$, too. But we have to be careful, since the ANOVA shows that the interaction $A : B$ is not significant.

Results from the analysis of DESIGN I lead to an improved DESIGN II. The following investigations were based on DESIGN II (corresponding settings marked with an asterisk in Tab. 2). We investigated the transferability of the results to other designs by modifying two parameters of the design vector $d$ (cp. Eq. 6): Dimension $D$ and noise $\sigma$.
Increasing the search-space dimension does not change the behavior of the PSO significantly: Fig.3 reveals the behavior of the PSO for $D = 24$ and $D = 96$ (instead of $D = 12$ in Fig. 2). As a first result, we can conclude: The general behavior of the PSO on the sphere model does not change, if the search space dimension is modified. Adding normal distributed noise to the fitness function value does change the behavior of the PSO and requires different strategy parameters to obtain convergence.[2]

[1]R, a language for data analysis and graphics was used to perform the ANOVA and to generate the graphics in this paper [IG96].

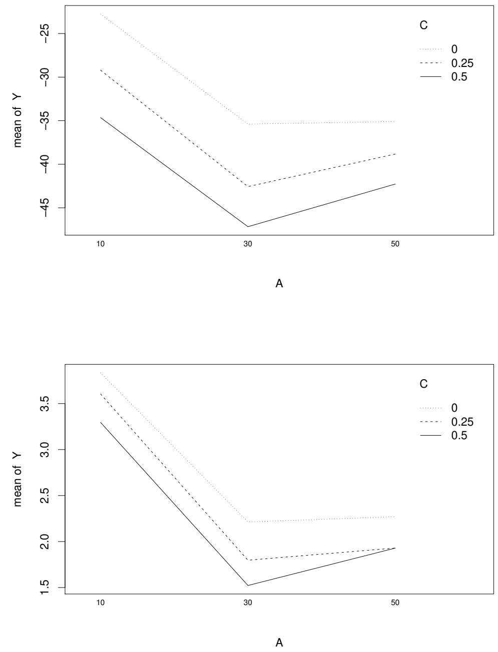


Figure 3: DESIGN II. Varying the search-space dimension: $D = 24$ (above) and $D = 96$ (below).

## 5 Summary and Outlook

In this paper, a first analysis of the PSO method's parameters, using DOE techniques, was introduced, and important parameter settings and interactions were extracted (screening).

Extension to other fitness functions and real–world problems, are needed to verify the insightful results obtained in this paper. Currently, experiments on the simplified elevator simulator (S-ring) [MAB+01], are performed to improve the efficiency of the design (fractional designs). A comparison of PSO and other stochastic search methods, especially EAs, will be given in future work.

### Acknowledgments

This work is a product of the Collaborative Research Center 'Computational Intelligence' (531), at the University of Dortmund and was supported by the Deutsche Forschungsgemeinschaft (DFG).

[2]These results will not be discussed here, but will be published in a forthcoming paper.

## Bibliography


[BD87] G. E. P. Box and N. R. Draper. *Experimental Model Building and Response Surfaces*. Wiley, 1987.

[BDT99] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.

[BHH78] G. E. P. Box, W. Hunter, and J. S. Hunter. *Statistics for experimenters*. Wiley series in probabilty and mathematical statistics: Applied probability and statistics. Wiley, 1978.

[CD01] A. Carlisle and G. Dozier. An Off-The-Shelf PSO. In *Proceedings of the Particle Swarm Optimization Workshop*, pages 1–6, 2001.

[DS98] N. R. Draper and H. Smith. *Applied regression analysis*. Wiley series in probability and statistics. Wiley, New York, 3 edition, 1998.

[EK95] R.C. Eberhart and J. Kennedy. A New Optimizer Using Particle Swarm Theory. In *Proceedings Sixth Symposium on Micro Machine and Human Science*, pages 39–43, Piscataway, NJ, 1995. IEEE Service Center.

[ES98] R.C. Eberhart and Y. Shi. Comparison Between Genetic Algorithms and Particle Swarm Optimization. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, editors, *Evolutionary Programming VII*, pages 611–616. Springer, 1998.

[ESD96] R.C. Eberhart, P. Simpson, and R. Dobbins. *Computational Intelligence PC Tools*. Academic Press, 1996.

[IG96] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[KE95] J. Kennedy and R.C. Eberhart. Particle Swarm Optimization. In *Proceedings IEEE International Conference on Neural Networks*, pages 1942–1948, Piscataway, NJ, 1995. IEEE Service Center.

[KE01] J. Kennedy and R.C. Eberhart. *Swarm Intelligence*. Academic Press, London, 2001.

[Ken98] J. Kennedy. The Behavior of Particles. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, editors, *Evolutionary Programming VII*, pages 581–590. Springer, 1998.

[Kle87] J. Kleijnen. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, New York, 1987.

[Kle99] J. Kleijnen. Validation of models: Statistical techniques and data availability. In P.A. Farrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans, editors, *Proceedings of the 1999 Winter Simulation Conference*, pages 647–654, 1999.

[KVG92] J. Kleijnen and W. Van Groenendaal. *Simulation - A Statistical Perspective*. Wiley, Chichester, 1992.

[LK00] A. M. Law and W. Kelton. *Simulation Modelling and Analysis*. McGraw-Hill Series in Industrial Egineering and Management Science. McGraw-Hill, New York, 3 edition, 2000.

[MAB+01] Sandor Markon, Dirk V. Arnold, Thomas Bäck, Thomas Beielstein, and Hans-Georg Beyer. Thresholding – a selection operator for noisy ES. In *Proc. Conference on Evolutionary Computation (CEC 2001)*, Seoul, Korea, May 27–30, 2001.

[Rec94] I. Rechenberg. Evolution Strategy. In J.M. Zurada, R.J. Marks II, and C. Robinson, editors, *Computational Intelligence: Imitating Life*, Piscataway, NJ, 1994. IEEE Press.

[Sch75] H.-P. Schwefel. *Evolutionsstrategie und numerische Optimierung*. Dr.-Ing. Thesis, Technical University of Berlin, Department of Process Engineering, 1975.

[Sch95] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.

[SE98a] Y. Shi and R.C. Eberhart. A Modified Particle Swarm Optimizer. In *Proceedings of the IEEE Conference on Evolutionary Computation*, pages 69–73, Piscataway, NJ, 1998. IEEE Press.

[SE98b] Y. Shi and R.C. Eberhart. Parameter Selection in Particle Swarm Optimization. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, editors, *Evolutionary Programming VII*, pages 591–600. Springer, 1998.