

# A Parameter Tuning Framework for Metaheuristics Based on Design of Experiments and Artificial Neural Networks

Felix Dobsław

*Abstract*—In this paper, a framework for the simplification and standardization of metaheuristic related parameter tuning by applying a four phase methodology, utilizing *Design of Experiments* and *Artificial Neural Networks*, is presented. Metaheuristics are multi-purpose problem solvers that are utilized on computational optimization problems for which no efficient problem-specific algorithm exists. Their successful application to concrete problems requires the finding of a good initial parameter setting, which is a tedious and time-consuming task. Recent research reveals the lack of approach when it comes to this so called *parameter tuning* process. In the majority of publications, researchers do have a weak motivation for their respective choices, if any. Because initial parameter settings have a significant impact on the solutions quality, this course of action could lead to suboptimal experimental results, and thereby a fraudulent basis for the drawing of conclusions.

*Keywords*—Parameter Tuning, Metaheuristics, Design of Experiments, Artificial Neural Networks.

## I. INTRODUCTION

Metaheuristics, such as Evolutionary Algorithms (EA), Particle Swarm Optimization (PSO), or Ant Colony Optimization (ACO), are multi-purpose optimization methods that can be applied to a large set of computational optimization problems. They become selected when the finding of an optimal solution is very expensive in time or space complexity, and where no efficient problem-specific algorithm pre-exists.

Each metaheuristic has a predefined set of parameters that has to be initialized before an execution. The metaheuristics adaptation requires the calibration of these parameters with respect to the problem at hand. This activity is called *parameter tuning*. An appropriate initial parameter setting has a significant impact on the solving progress, such as the exploitation or exploration rate of the search space, and therefore the quality of the solution. The importance of parameter tuning is discussed in [1], in the context of EA. Furthermore, there exists no general optimal initial parameter setting for metaheuristics. This means that for any metaheuristic, an optimal initial parameter setting can vary considerably from problem to problem, and even between problem instances.

Figlali et al. reveal in [2] the lack of approach when it comes to parameter tuning, using ACO. In the majority of investigated publications, researchers have a weak motivation for their choice of parameters, if any.

In this paper, a new general framework for an automated parameter tuning for metaheuristics, is presented. In contrast

to other automated tuning methods such as in [3] or [4] for PSO, or the ones mentioned in [5] for ACO, is the method proposed here applicable to any metaheuristic.

The methodology aggregates four phases, 1. *Problem Description*, 2. *Training*, 3. *Parameter Retrieval* and 4. *Execution*. It combines the advantages of an automated Design of Experiments (DoE) for efficient parameter tuning and Artificial Neural Networks (ANN) for the recognition of good initial parameter settings for new problem instances.

The paper is structured as follows: This section proceeds with short introductions to parameter control and tuning, Design of Experiments, and Artificial Neural Networks. Section two, *The Framework*, presents the proposed framework, and its phases, in further detail. In section three, *Discussion and Concluding Remarks*, the applicability and relevance of the approaches are discussed, and section four, *Future Work*, concludes the paper by giving record of ongoing and potential prospective research.

### A. Parameter control and tuning

In [1], Eiben et al. lift the importance of parameter control and tuning in the context of EA. Even though narrowed to EA, the discussion applies to parameter control and tuning to metaheuristics in general. Where parameter tuning addresses the finding of good static parameters *before* the execution of a metaheuristic to a problem, parameter control addresses the dynamic change of parameters *throughout* the execution. A combination of both approaches is generally required for the finding of satisfactory solutions.

### B. Design of Experiments

Design of Experiments (DoE) is a stochastic framework for the conduct of representative experiments [6]. It attempts to minimize the amount of required experiments for an analysis, while maintaining high quality results. Experiments are considered to have input variables (factors) and output variables (responds). DoE suggests a well-organized approach, combining experiments with extreme values and representative experiments, so called "center points" (see Figure 1). A common objective with DoE is to optimize the factors by comparing and evaluating the quality of the responses. DoE has successfully been applied as a tool for the manual parameter tuning of particular computational optimization problems (e.g., [7] and [8]).

F. Dobsław is with the Information Technology and Media department, Mid Sweden University, Östersund, Sweden e-mail: Felix.Dobsław@miun.se.

Manuscript received December 30, 2009

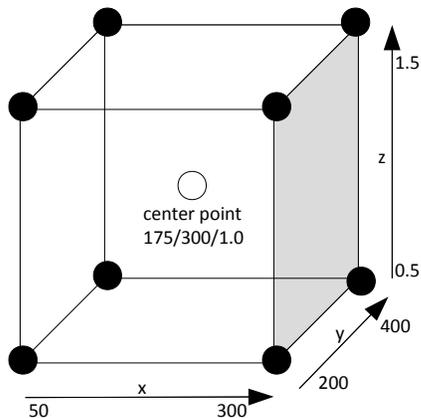


Figure 1. An illustration of a 3-factor full factorial DoE, highlighting the center and extreme point experiments within the cube.

### C. Artificial Neural Networks

An Artificial Neural Network (ANN) is a statistical regression tool that can be utilized for the modeling of complex systems, or the finding of patterns in data. ANN's have the capability to 1) learn from experience, 2) generalize, and 3) serve as a universal functional approximator [9]. They are inspired by biological nervous systems, utilizing the concepts of different neuron types that are linked via synaptic connections, resulting in a so called neural network. A special characteristic of neural networks is, that their knowledge is distributed throughout the neuron connections. The output behavior of an ANN depends on its neuron model, inner architecture, initial weighting and the cost function. ANN's are especially useful in areas where the functional behavior of the systems to be simulated is hard or even impossible to figure out ("black boxes"). The ANN model is computationally equivalent to the Universal Turing Machine model.

## II. THE FRAMEWORK

In this paper, a framework that tackles the parameter tuning issue by semi-automation, utilizing a combination of DoE and ANN, is proposed. Subsequently, the four process phases, 1. *Problem Description*, 2. *Training*, 3. *Parameter Retrieval* and 4. *Execution*, are explained in further detail. The explanation is supported by Figure 2, which illustrates the whole parameter tuning process, component-wise. To provide a better understanding, a scenario is pursued in the text, where appropriate. In this scenario, it is assumed that the user tries to solve the Traveling Salesman Problem (TSP) by applying the standard PSO 2007<sup>1</sup> metaheuristic. TSP is one of the classic NP-complete combinatorial optimization problems in the literature. Given a list of cities, specified by their coordinates, the task for "the travelling salesman" is to find a shortest tour, visiting each listed city exactly once.

<sup>1</sup><http://www.particleswarm.info/>

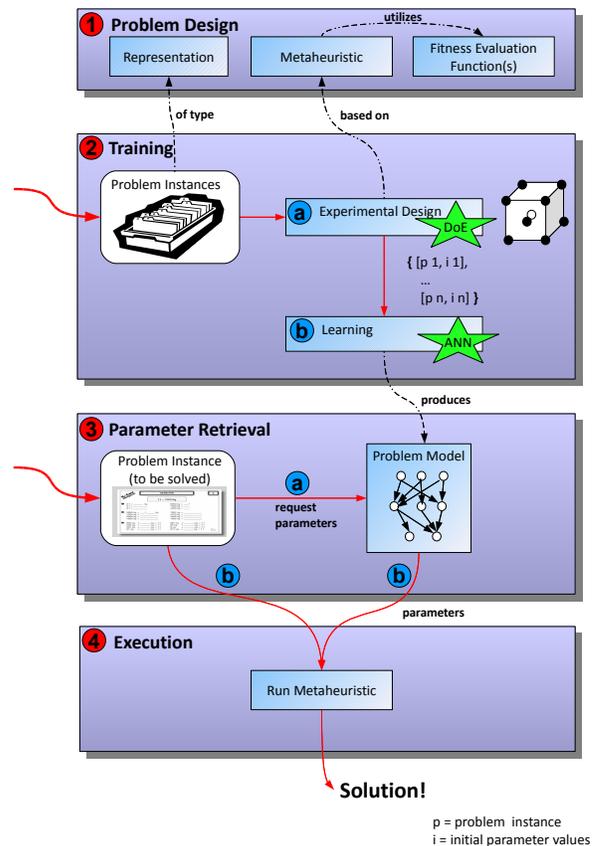


Figure 2. The four phases of the methodology: 1. *Problem Design*, 2. *Training*, 3. *Parameter Retrieval* and 4. *Execution*, illustrated component-wise.

### A. Phase 1: Problem Design

First, the problem at hand has to be specified. This includes its computer-readable representation, the definition of a fitness evaluation function for assessment of the quality of the results, and the choice of a metaheuristic to approach it. In the case of TSP, a representation usually is an array of two-dimensional coordinates (latitude  $\times$  longitude), representing the cities for a tour. The fitness evaluation function could simply measure the total length of a complete tour. Depending on the choice of metaheuristic, more or less effort has to be put into its adaptation to the problem. EAs, for instance, require the definition of reproduction operations, such as mutation or crossover, for an incremental fitness improvement. For PSO, this is sometimes referred to as swap sequencing (see e.g., [10]).

### B. Phase 2: Training

In the training phase, a problem model is generated from a two-stage process. The first stage, *Experimental Design*, concerns the optimization of initial parameter values by applying an automated DoE to a finite training set of problem instances. In stage two, *Learning*, an ANN becomes trained with the data from stage one, in order to later on give recommendations for initial parameters for any given problem instance.

param	factor	data type	min	max
$A_1$	population size	integer	2	100
$A_2$	inertia weight (velocity impact)	decimal	0.8	1.2
$A_3$	c1 (cognitive parameter)	decimal	1	3
$A_4$	c2 (social parameter)	decimal	1	3

Table I

AN EXAMPLE OF HOW THE STANDARD PSO FACTORS COULD BE DEFINED FOR DoE IN THE FIRST STAGE OF THE TRAINING PHASE.

Without loss of generality, let  $P$  be the set that compiles all instances of the problem at hand,  $P$ .  $A_1, \dots, A_m, m \in \mathbb{N}$  are said to be the initial, metaheuristically specific, parameters with their respective domains. Let then  $I_P = \{(a_1, \dots, a_m) | a_j \in A_j, j \in \underline{m}\}$  be the set of all possible initial parameter value combinations for  $P$ .

1) *Experimental Design*: DoE's factors for the factorial design are the metaheuristic parameters  $A_1, \dots, A_m$ , described by their data type and respective extreme values, *min* and *max*. For standard PSO they may look as in Table I. This table is used as a starting point for experimental design, independent of  $P$ . However, finding good extreme values for the parameters is a non-trivial task in itself. An investigation of former achievements with the metaheuristic at hand can reveal commonly applied settings that can work as an indicator for where minimal and maximal bounds could be settled.

Each problem instance is assessed independently for the finding of near-optimal initial parameter values. For this reason an iterative factorial design with repetitive experiments has been proposed. The user is here able to specify the amount of iterations. One iteration means the conduct of a full-scale factorial design with extreme value and center point simulations. The results from a former iteration are used for a screening into most promising areas of the search space. In the case of Figure 3, while trying to maximize the fitness, another DoE experimental series would be suggested with bounds in the upper corner, indicated by the cube. That way, the accuracy of the findings improves iteratively, with the caveat of a longer experimental execution time. In case of a limitation to one screening for each iteration, the increase of execution time is linear (assuming that the experiments require about the same runtime). For each problem instance  $p \in P$  the most promising exploited parameter setting  $i_p \in I_P$  is suggested, after the last iteration, as its best known initial parameter vector.

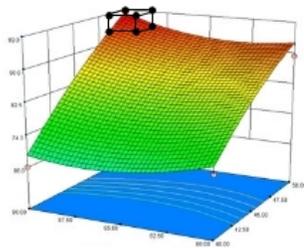


Figure 3. An exemplified behavioral landscape after a DoE iteration. Red color means high fitness; green color low fitness.

The training result is a set  $T_{res} = \{(p, i_p) | p \in P, i_p \in I_P\}$ .

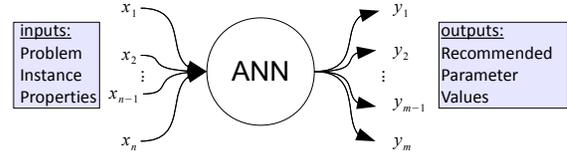


Figure 4. The ANN's inputs are problem-specific. Provided with the property values of the problem instance, the ANN replies with recommended initial parameter values for the execution of the instances.

For TSP, problem instances could be arrays of the ten largest cities in Germany and Sweden respectively,  $p_{ger}, p_{swe} \in P$ . After the experimental design, the resulting training set, applying standard PSO, might be:

$$T_{res_{tsp\_pso}} = \{[p_{ger}, [70, 1.11, 2.0, 2.23]], [p_{swe}, [65, 0.86, 1.98, 2.9]]\}$$

This should only be seen as an example, since the number of two problem instances would in general not suffice for the inference of a representative problem model.  $T_{res}$  is of high importance to the upcoming learning stage.

2) *Learning*: The ANN is trained to recommend near-optimal initial parameters, based on the properties of so far unknown problem instances. To do so, the ANN is fed with  $T_{res}$ , for supervised learning. See Figure 4 for the ANN's structure. Its inputs  $X_1, \dots, X_n$  are problem properties, describing the concerned problem instance as precisely as possible. The problem properties for TSP, for instance, naturally include the number of cities on a map. But they might be extended by the general density or average distances to the nearest neighbor or other relevant properties. Here is where the user understanding of the problem can make a significant impact on the quality of the models. The ANN's output parameters  $Y_1, \dots, Y_m$  are equal to  $A_1, \dots, A_m$  from the experimental design.

In response to the training, the ANN builds a regression model that captures an understanding of the problem at hand. The trained ANN serves from here on as the problem model.

### C. Phase 3: Parameter Retrieval

The ANN can now be utilized as an oracle, to receive good initial parameter values  $(y_1, \dots, y_m) \in Y_1 \times \dots \times Y_m$  for any given  $p \in P$ , represented by a vector  $(x_1, \dots, x_n) \in X_1 \times \dots \times X_n$ . Committed requests are treated in real time.

### D. Phase 4: Execution

The last stage is the execution of the algorithms with the initial parameter setup  $(y_1, \dots, y_m) \in Y_1 \times \dots \times Y_m$ , as recommended in phase 3.

## III. DISCUSSION AND CONCLUDING REMARKS

A framework for the simplification and standardization of metaheuristic-related parameter tuning by applying a four phase methodology, utilizing DoE and ANN, was presented.

The author's position is that parameter tuning is as relevant to metaheuristics as parameter control is. Appropriate initial values for parameters are of high importance for a metaheuristic's performance [1]. Optimal results are generally neither obtained if parameters are inappropriately initialized, nor if there exists no satisfactory dynamic parameter control. Parameter control addresses another issue, which the author sees as a problem of methodology in itself. An issue that is of high relevance, but which, however, has not been addressed in the context of this paper.

The main advantage of the proposed parameter tuning methodology is that it can be applied to any attempt to solve a computational optimization problem with a metaheuristic.

The training phase of the frameworks is composed of the *experimental design* and *learning* stages. These stages are linked together by a well-defined interface, which allows for modularity, meaning here the exchange of components. Thus, DoE and ANN are not compulsory parts of the methodology, as long as they are substituted by compatible means. DoE and ANN were selected due to their respective applicability for experimental design and learning.

The methodology's time-consuming fraction is the experimental training stage. Initial parameter values for new problem instances can be retrieved in real time from the ANN, once the training phase is over.

The quality of the recommended parameter values depends on

- 1) the quality of the DoE recommendations.
- 2) the quality (diversity) of the training set for the ANN.
- 3) the problem properties for ANN training.
- 4) the suitability of ANN's network properties (initial weighting, cost function, ...).

#### IV. FUTURE WORK

The future work will be allocated in 1) the sharp design of the inner architecture of the DoE and ANN compartments for the training phase, and 2) the experimental verification of the applicability of the framework.

##### A. DoE and ANN Design

One question to answer is if a single factorial design type for DoE is generally applicable. Otherwise, the user has to be added into the loop in the DoE design stage to ensure a solid basis for further inference. The ANN's inner architecture, neuron design, cost functions, regression type and initial weighting have to be experimentally evaluated and decided upon. The problem of overfitting has to be tackled. The resulting models have to be tested independently, and furthermore, compared to standard optimization problems with well-performing state of the art algorithms and parameter tuning methods.

##### B. Verification

The compartments have to be combined and integrated into the framework in a sense that final conclusions could be drawn about

- the quality of recommended parameters from the DoE in comparison to known optimal or 'good' parameter settings.
- the quality of recommended parameters from the ANN in comparison to the recommended parameter settings, obtained from the DoE.
- the quality of recommended parameters from the ANN in comparison to known optimal or 'good' parameter settings.
- the impact of the framework on the solution quality, as well as the total execution time.

A state of the art survey about metaheuristic-related parameter tuning, as well as experimental work regarding the previously mentioned questions are in working progress.

#### ACKNOWLEDGMENT

The author would like to thank Åke Malmberg, Keith Downing, Gerasimos Antzoulatos and Anders Olsson for prolific discussions and comments.

#### REFERENCES

- [1] E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.
- [2] N. Figlali, C. Özkale, O. Engin, and A. Figlali, "Investigation of Ant System parameter interactions by using design of experiments for job-shop scheduling problems," *Computers & Industrial Engineering*, vol. 56, pp. 538–559, 2009.
- [3] G. Tewolde, D. Hanna, and R. Haskell, "Enhancing performance of pso with automatic parameter tuning technique," pp. 67–73, 2009.

- [4] Y. Cooren, M. Clerc, and P. Siarry, "Performance evaluation of tribes, an adaptive particle swarm optimization algorithm," *Swarm Intelligence*, vol. 3, no. 2, pp. 149–178, 2009.
- [5] K. Wong and Komarudin, "Parameter tuning for ant colony optimization: A review," pp. 542–545, 2008.
- [6] L. Eriksson, E. Johansson, C. Kettaneh-Wold, and S. Wikström, Wold, *Design of Experiments, Principles and Applications*. MKS Umetrics AB, 2008.
- [7] O. Kramer, B. Gloger, and A. Goebels, "An experimental analysis of evolution strategies and particle swarm optimisers using design of experiments," pp. 674–681, 2007.
- [8] Y. Lucas, A. Domingues, D. Driouchi, and S. Treuillet, "Design of experiments for performance evaluation and parameter tuning of a road image processing chain," *Eurasip Journal on Applied Signal Processing*, 2006.
- [9] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.
- [10] K.-P. Wang, L. Huang, C.-G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem," vol. 3, pp. 1583–1585, 2003.



**Felix Dobsław** received a BSc. degree in Informatics from the University of Hamburg and a MSc. degree in Computer Science with a major in Distributed Systems from Mid Sweden University. He is currently pursuing PhD studies at the Information Technology and Media (ITM) department in Östersund, Sweden. His research interests include parameter tuning, metaheuristics, Evolutionary Algorithms, Swarm Intelligence, Particle Swarm Optimization, Ant Colony Optimization, and Context Modeling.