12-2014

Second Order-Response Surface Model for the Automated Parameter Tuning Problem

Aldy Gunawan

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

Second Order-Response Surface Model for the Automated Parameter Tuning Problem

Aldy Gunawan[1], Hoong Chuin Lau[1]
[1]School of Information Systems, Singapore Management University, Singapore
(aldygunawan@smu.edu.sg, hclau@smu.edu.sg)

Abstract - **Several automated parameter tuning procedures/configurators have been proposed in order to find the best parameter setting for a target algorithm. These configurators can generally be classified into model-free and model-based approaches. We introduce a recent approach which is based on the hybridization of both approaches. It combines the Design of Experiments (DOE) and Response Surface Methodology (RSM) with prevailing model-free techniques. DOE is mainly used for determining the importance of parameters. A First Order-RSM is initially employed to define the promising region for the important parameters. A Second Order-RSM is then built to approximate the center point as well as the final promising ranges of parameter values. We show how our approach can be embedded with existing model-free techniques, namely ParamILS and Randomized Convex Search, to tune target algorithms and demonstrate that our proposed methodology leads to improvements in terms of the quality of the solutions compared against the earlier work.**

Keywords – **Design of Experiment, Parameter Tuning, Response Surface Methodology, Second Order Model**

I. INTRODUCTION

Many combinatorial optimization problems are NP-hard. Heuristic algorithms (known as target algorithms) have been proposed to solve these problems, and the performance of them typically depends on the value of the underlying parameters. For example, a Simulated Annealing algorithm relies on a good choice of the initial temperature and the cooling factor.

The importance of fine-tuning parameters in a heuristic algorithm has been addressed by many authors (e.g. Barr et al. [1] and Fink and Voss [2]). In response to the need for a principled approach to find good parameter settings, several automated parameter tuning procedures (or configurators) have been proposed to search for good parameter values of a target algorithm.

The Automated Tuning problem is defined as follows: *Given a target algorithm TA parameterized by a set of parameters X with their respective intervals, a set of training instances I_{tr}, and a meta-function H(x) that measures the algorithm performance on a fixed parameter setting x over a set of problem instances, the goal is to determine a configuration x* such that H(x*) is minimized over I_{tr}.*

The configurators can generally be classified into: model-free and model-based approaches [3]. Model-free approaches are generally simple so they can be applied out-of-the-box, while model-based approaches uses fitting models to choose which configurations to investigate.

Some model-free approaches are the racing algorithm F-Race [4], ParamILS [5], and Randomized Convex Search (RCS) [6]. On the other hand, some model-based approaches, such as CALIBRA [7], SPO and its variants [8, 9] and SMAC [3], have their roots in statistics such as the Design of Experiments, Gaussian Stochastic Process and Tree-Based Regression.

A tuning framework based on DOE was proposed in Gunawan et al. [10] that combines a model-based (DOE methodology) with a model-free approach. The limitation of the framework is that only First Order-RSM (FO-RSM) is considered to define the promising initial range for the important parameters. In this paper, we propose a key improvement to FO-RSM, to allow it to find a better range for configurators. Once we reach the optimum region of the parameter values, a Second Order-RSM (SO-RSM) is built to further explore the region.

A Central Composite Design (CCD) is applied for fitting the second-order model [11]. A stationary point of the surface is then calculated and used as a new centre point. This center point would then be used to define a new parameter range for the model-free configurators.

II. AUTOMATED TUNING FRAMEWORK

The framework proposed [10] consists of three phases: screening, exploration and exploitation phases (Figure 1). In the following sub-sections, we provide a short description of each phase.

Algorithm: Automated Tuning Framework

TA: Target algorithm with *k* parameters

Screening Phase:
1. Run 2^k factorial design to identify *m* important parameters
2. Update the range for each important parameter

Exploration Phase:
1. Apply First Order-RSM to identify the promising region
2. Apply Second Order-RSM to identify the stationery point and final range for each important parameter

Exploitation Phase:
1. Apply the configurator to tune *TA* based on the final range for each parameter

Fig. 1. Automated Tuning Framework

2.1. Screening Phase

Initially, *k* parameters of *TA* are to be tuned, where each parameter p_i (discrete or continuous) lies within a numeric interval [l_i, u_i]. In the screening phase, a complete 2^k factorial design is applied to identify *m* parameters (*m*

$\leq k$) which have significant effects to the performance of *TA*, so called **important parameters**. A complete 2^k factorial design requires ($n \times 2^k$) observations, where n represents the number of replicates. Experiments are replicated to help identify the sources of variation and to better estimate the true effects of treatments.

The importance of a particular parameter p_i can be defined by conducting the significance test on the main effect of the parameter (with a significance level = 10%). Furthermore, the ranking of the important parameters is determined by the absolute values of the main effects of those important parameters. By doing so, one can determine which parameters should be carefully controlled including the direction of adjustment for these parameters. The range of important parameter p_i is then modified to a sub-interval $[l'_i, u'_i]$. On the other hand, unimportant parameters would be set to a constant value. In this phase, we still assume that we are at a region that is remote from the optimum, therefore, there is little curvature in the model and the linear model would be sufficient [12].

2.2. Exploration Phase

In the earlier work [10], an approach based on First Order-Response Surface Model (FO-RSM) was proposed. In this paper, we propose another approach, namely Second Order-Response Surface Model (SO-RSM). The main difference between both models is in the basic model used. The former employs a planar model to approximate the response, while the latter use a second-order model due to curvature of the true response surface. The details of both would be explained below.

2.2.1 First Order-Response Surface Model (FO-RSM)

Let m be the total number of important parameters ($m \leq k$) determined in the screening phase where each parameter p_i has a modified interval $[l'_i, u'_i]$ and its centre point value $(l'_i + u'_i)/2$ as well. In essence, we begin with a small region and aim to find a "promising" range for important parameters using steepest descent on the response surface. The target algorithm is run with respect to the parameter configuration space Θ which contains (2^m+1) possible parameter settings. An additional parameter setting is defined by the centre point value of each parameter.

We apply a factorial experiment design in order to build a first-order model. The underlying assumption is that the region can be approximated by a planar model, which is a reasonable assumption when the region is sufficiently small and far from the optimum. The planar model is given by the following approximating function:

$$Y = \beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m + \varepsilon \quad (1)$$

In order to test the significance of this model, we conduct two additional statistical tests: 1) *Interaction test*, mainly on testing whether any interaction between parameters; 2) *Curvature test*, mainly on testing whether the planar model is adequate to represent the local response function. As long as each test is not statistically significant, we can always assume that the planar model is adequate to represent the true surface of parameters. We then continue the process by applying steepest descent that allows us to move rapidly to the vicinity of the optimum. More precisely, we move sequentially along the path of steepest descent in the direction of the maximum decrease in the response [12]. For the details, please refer to Gunawan et al. [10].

2.2.2 Second Order-Response Surface Model (SO-RSM)

When the parameter values are relatively close to the optimum region, [12] shows that a second-order model to approximate the response due to the curvature in the true response surface is more appropriate. Using this idea, we build a second-order model that approximates the response surface:

$$Y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + \sum_{i=1}^{m} \beta_{ii} x_i^2 + \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \beta_{ij} x_i x_j + \varepsilon \quad (2)$$

Box and Wilson [13] introduce Central Composite Designs (CCD) for fitting the second-order model. In general, a CCD in m parameters requires 2^m factorial runs with $2m$ additional combinations called *axial points* along the coordinate axes of the **coded** parameter levels and at least one center point. The axial points are required to determine the coefficients of the second-order model. For example, a design with $m = 2$ requires (4+4+1) possible combination of parameter settings. The lower and upper bound for each parameter i are coded as -1 and 1, respectively. The coordinate for each combination is represented in Table I.

TABLE I CCD FOR $m = 2$

Combinations	Coordinate
2^m Factorial	(-1,-1), (-1,1), (1,-1), (1,1)
$2m$ axial points	(±α,0), (0,±α), where $\alpha = (2^k)^{1/4}$
1 centre point	(0.0)

In order to estimate the stationary point of Equation (2), we reformulate the formula as follows:

$$Y = \hat{\beta}_0 + x'b + x'Bx \quad (3)$$

where:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_m \end{bmatrix} \quad b = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \vdots \\ \hat{\beta}_m \end{bmatrix} \text{ and } B = \begin{bmatrix} \hat{\beta}_{11} & \hat{\beta}_{12}/2 & \ldots & \hat{\beta}_{1m}/2 \\ \hat{\beta}_{12}/2 & \hat{\beta}_{22} & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \hat{\beta}_{1m}/2 & \ldots & \ldots & \hat{\beta}_{mm} \end{bmatrix} \quad (4)$$

The stationary point of parameters is then calculated by the following formula:

$$x_s = -\frac{1}{2} B^{-1} b \quad (5)$$

Once we have calculated the stationary point, we treat it as the new centre point of important parameter p_i for defining the final range [l^{final}_i, u^{final}_i].

2.3. Exploitation Phase

In this phase, we find the optimal point in the region that is output from the exploration phase described above. This can be achieved by applying a configurator such as F-Race [4], ParamILS [5] or RCS [6]. In this paper, we experiment on two configurators, ParamILS and RCS. More precisely, ParamILS is applied to tune the Iterated Local Search algorithm for TSP [14], while RCS is applied to the hybrid Simulated Annealing and Tabu Search algorithm for QAP [15].

II. EXPERIMENTS

3.1. Experimental Setup

In order to evaluate the performance of our proposed approach, we solved some benchmark problems for the TSP and QAP (TSPLIB and QAPLIB). The problem instances are then divided into two categories: training and testing instances (Table II). We find the best parameter setting configuration of the target algorithm based on the set of training instances. The quality of the best configuration is then assessed based on the set of testing instances. For a particular parameter setting, we take the average of 10 runs. The objective is to measure the improvements in terms of the gap (i.e. percentage deviation) between the average objective values of the solutions obtained by our approach against the best known solutions.

TABLE II TRAINING AND TESTING INSTANCES

Problem	*N* training instances	*N* testing instances
TSP	47 instances	23 instances
QAP		
- *Unstructured instances*	11 instances	5 instances
- *Grid-based distance matrix*	24 instances	11 instances
- *Real-life instances*	14 instances	7 instances

TABLE III PARAMETERS FOR ILS ON TSP

Parameters	Type	Range	Definition
max_iter	Discrete	[100, 900]	number of iteration
perturb	Discrete	[1 ,10]	number of perturbations
non_imprv	Discrete	[1, 10]	number of allowable non-improving moves
opt_cho	Discrete	[3, 4]	perturbation strategy

For each problem, we analyze and compare three different scenarios, configurator, configurator+FO-RSM and configurator+SO-RSM. The amount of resources allocated (i.e. the number of iterations) are equal in order to ensure the fairness. Tables III and IV list the parameters that need to be tuned for each target algorithm including with their initial range values [10].

TABLE IV PARAMETERS FOR SA-TS ON TSP

Parameters	Type	Range	Definition
temp	Continuous	[100, 9000]	initial temperature
alpha	Continuous	[0.5, 0.95]	the cooling factor
length	Discrete	[5, 10]	the length of tabu list
pct	Continuous	[0.01. 0.10]	percentage of non-improvement iterations prior to intensification

3.2. Computational Results

3.2.1 Screening and Exploration Results

In the screening phase, our main focus is to determine which parameters are significantly important. For TSP, three parameters (*max_iter*, *perturb*, and *non_imprv*) are significantly important, while *opt_cho* is an unimportant parameter therefore we decide to set the value of this parameter to its lower bound value ($l_{opt\_cho} = 3$) since its β value (equation (1)) is a positive value.

We first observe the relationship between two most important parameters, *max_iter* and *perturb,* for TSP. We initially set *perturb* to different discrete values from 1 to 10. For a particular discrete parameter value, we then run the configurator to find out the best parameter value of *max_iter* = [100, 900]. Figure 2 visualizes the relationship between *max_iter* and *perturb*. Observing the best parameter configurations for both parameters, we notice that there is a strong negative interaction with the correlation coefficient = -0.736. This indicates that a planar model is not sufficient to predict the promising of parameter values since the interaction does exist.

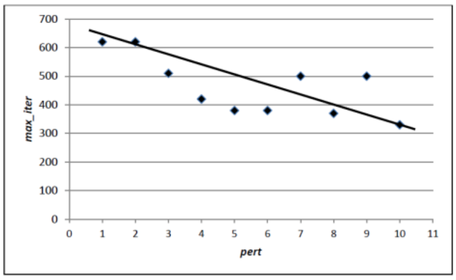

Fig. 2. Interaction Plot

TABLE V PARAMETER SPACE FOR ILS ON TSP

Parameters	Range	
	Exploration Phase (FO-RSM)	Exploration Phase (SO-RSM)
max_iter	[400, 600]	[340, 540]
Perturb	[1, 3]	[1, 3]
non_imprv	[4, 6]	5
opt_cho	3	3

In the exploration phase, we focus on the important parameters obtained from the screening phase. The promising region of important parameters is explored by using steepest descent method. We implement Central Composite Designs (CCD) for fitting the second-order model. By doing so, a better initial range including the center point of the range for the configurator can be approximated as summarized in Table V.

Table VI summarizes the results obtained for each class of the QAP problem. Due to space limitation, we only show the final exploration results of SO-RSM. We observe that only two parameters, *temp* and *alpha*, are significantly important. The rest are set to a constant value. Firstly, we observe that the optimal region would be obtained if we set *temp* and *alpha* at approximately within [4945-6945] and [0.95-0.99], respectively. By using equation (5), we obtain the stationary point (5945 and 0.97) for both parameters. This stationary point would be treated as the centre point in the next phase, exploitation phase. The final ranges [l^{final}_i, u^{final}_i] that would be used by a configurator are [5945-Δ_1, 5945+Δ_1] and [0.97-Δ_2, 0.97+Δ_2], respectively. Here, we set Δ_1 = 1000 and Δ_2 = 0.02. Similar observations can be concluded for other problem classes.

TABLE VI PARAMETER SPACE FOR SA-TS ON QAP

Parameters	Range		
	Unstructured instances	Grid-based distance matrix	Real-life instances
Temp	[4313, 6313]	[4945, 6945]	[5066, 7066]
Alpha	[0.94, 0.99]	[0.95, 0.99]	[0.76, 0.96]
Length	5	6	[6, 8]
Pct	0.01	0. 1	0.1

3.2.2. Exploitation Results

Given the output of the exploration phase, we apply a configurator to tune important parameters. By using the second-order model, we can provide a very good initial range for a configurator. The performance of three different approaches: 1) configurator, 2) configurator+FO-RSM and 3) configurator+SO-RSM, are compared. For the configurators, the initial range for the parameter values are taken from Tables III and IV.

TABLE VII BEST PARAMETER SETTING FOR ILS

Approach	*max_iter*	*perturb*	*Non_improv*	*opt_cho*
ParamILS	500	1	6	3
ParamILS+FO-RSM	600	1	6	3
ParamILS+SO-RSM	540	1	5	3

Tables VII and VIII summarize the best parameter setting obtained for each approach. A negative interaction between parameters *max_iter* and *perturb*. The best parameter setting given by all approaches are quite similar. Parameter *max_iter* has to be set into high value (= 500) while parameter *perturb* is set to a low value (= 1). For QAP problem, both parameters *temp* and *alpha* have to be set to high values since both have a positive interaction. We take the average of 10 runs on each training instance for a particular parameter setting. In order to compare all approaches, we calculate the following descriptive statistics: grand mean of average objective function value, average of standard deviation objective function value and average of coefficient of variance (CV).

TABLE VIII BEST PARAMETER SETTING FOR SA-TS

Approach	*temp*	*alpha*	*length*	*pct*
RCS				
- unstructured instances	7000	0.950	7	0.01
- grid-based distance matrix	7000	0.950	7	0.10
- real-life instances	6886	0.930	10	0.03
RCS+FO-RSM				
- unstructured instances	6348	0.935	5	0.01
- grid-based distance matrix	4238	0.945	6	0.10
- real-life instances	6000	0.950	5	0.10
RCS+SO-RSM				
- unstructured instances	6313	0.990	5	0.01
- grid-based distance matrix	4945	0.950	6	0.10
- real-life instances	6174	0.960	8	0.10

Table IX gives the details of tuning results for TSP problem. By implementing ParamILS+SO-RSM, the percentage deviations between the average objective function value of the solutions obtained and the best known/optimal solutions are only 3.423% and 3.893% for training and testing instances, respectively. The averages of standard deviation objective function value and the averages of CV are also lower than other approaches for both training and testing instances. A low value of CV indicates lower sensitivity and less dispersion of the results. We conclude that by using SO-RSM, ParamILS can actually perform better by finding the best parameter setting within a smaller promising region.

Similar observation can also be made for the QAP. Based on the results in Table X (we do not report the entire results due to space limitation), we have the following findings. As we expected, RCS+SO-RSM have considerably higher percentages of improvement than other approaches, RCS and RCS+FO-RSM. For instance, in *unstructured instances* class, the percentage deviations from the optimal/best known solutions are only 1.077% and 0.643% for the training and testing instances, respectively. In general, the average of standard deviation and coefficient of variance of RCS+SO-RSM are the least for the majority of the instances.

V. CONCLUSION

In this paper, we have shown that our proposed approach can improve [10] which is based on DOE and First-Order Response Surface Model. We paid special attention to the issues of curvature and interaction between parameters, especially when the first-order model is no longer sufficient in predicting algorithm performance.

For future works, we plan to test the effectiveness of our approach on target algorithms with larger number of parameters. Finally, we observe that the difficulty of parameter tuning also depends on parameters interactions, hence search space analysis on the parameter values is a promising area that deserves further investigation.

ACKNOWLEDGMENT

This research is supported by Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

TABLE IX PARAMETER TUNING FOR ILS ON TSP

	Training Instances			**Testing Instances**		
Method	Grand mean of average objective value	Average of std dev objective value	Average of coefficient of variance	Grand mean of average objective value	Average of std dev objective value	Average of coefficient of variance
ParamILS	3.930	0.656	0.147	4.103	0.754	0.195
ParamILS+FO-RSM	3.690	0.656	0.243	3.423	0.599	0.193
ParamILS+SO-RSM	3.423	0.599	0.193	3.893	0.821	0.183

TABLE X PARAMETER TUNING FOR SA-TS ON QAP

	Training Instances			**Testing Instances**		
Method	Grand mean of average objective value	Average of std dev objective value	Average of coefficient of variance	Grand mean of average objective value	Average of std dev objective value	Average of coefficient of variance
Unstructured instances						
RCS	2.060	0.623	0.518	1.759	0.909	0.518
RCS+FO-RSM	2.022	0.508	0.486	1.708	0.788	0.486
RCS+SO-RSM	1.077	0.256	0.215	0.643	0.184	0.512
Grid-based distance matrix						
RCS	0.485	0.358	0.785	0.769	0.648	0.757
RCS+FO-RSM	0.422	0.250	0.807	0.467	0.484	1.174
RCS+SO-RSM	0.379	0.230	0.782	0.414	0.273	0.693
Real-life instances						
RCS	8.727	4.528	0.597	5.179	2.384	1.006
RCS+FO-RSM	8.618	4.627	0.803	4.881	1.385	0.550
RCS+SO-RSM	6.512	4.225	0.718	4.369	2.485	0.415

REFERENCES

[1] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende, and W. R. Stewart, “Designing and reporting on computational experiments with heuristic methods,” *Journal of Heuristics*, vol. 1, pp. 9-32, 1995.

[2] A. Fink, S. and Voss, “Solving the continuous flow-shop scheduling problem by metaheuristics,” *European Journal of Operational Research*, vol. 151, pp. 400-414, 2003.

[3] F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Sequential model-based optimization for general algorithm configuration,” in *Proc. of the Learning and Intelligent Optimization, 5th International Conference* (*LION 5*). LNCS, vol. 6683, pp. 507-523. Springer, Heidelberg, 2011

[4] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp, “A racing algorithm for configuring metaheuristics,” In Morgan Kaufmann (editor) *Proc. of the Genetic and Evolutionary Computation Conference* (*GECCO 2002*), pp. 11-18, San Francisco, CA, USA, July 9-13 2002.

[5] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle, “ParamILS: an automatic algorithm configuration framework,” *Journal of Artificial Intelligence Research*, vol. 36, pp. 267-306, 2009.

[6] H. C. Lau, and F. Xiao, “Enhancing the speed and accuracy of automated parameter tuning in heuristic design,” in *Proc. of the 8th Metaheuristic International Conference* (*MIC 2009*), Hamburg, Germany, July 13-16 2009.

[7] B. Adenso-Díaz, and M. Laguna, “Fine-tuning of algorithms using fractional experimental designs and local search,” *Operations Research*, vol. 54, no 1, pp. 99-114, 2006.

[8] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss, “Sequential parameter optimization,” in *Proc. of the 7th IEEE International Conference on E-Commerce Technology* (*CEC-05*), pages 773-780, München, Germany, July 19-22 2005.

[9] F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. Murphy, “Time-bounded sequential parameter optimization,” in *Proc. of the Learning and Intelligent Optimization, 4th International Conference* (*LION 4*). LNCS, vol. 6073, pp. 281-298. Springer, Heidelberg, 2010.

[10] A. Gunawan, H. C. Lau, and Lindawati, “Fine-tuning algorithm parameters using the design of experiments approach,” in *Proc. of the Learning and Intelligent Optimization, 5th International Conference* (*LION 5*). LNCS, vol. 6683, pp. 278-292. Springer, Heidelberg, 2011

[11] R. O. Kuehl, “Design of experiments: statistical principles of research design and analysis,” 2 edn. Pacific Grove, California: Duxbury/Thomson Learning, 2000.

[12] D. C. Montgomery, ”Design and analysis of Experiments,” 6 edn. John Wiley and Sons Inc, 2005.

[13] G. E. P. Box, and K. G. Wilson, “On the experimental attainment of optimum conditions,” *Journal of the Royal Statistical Society B*, vol. 13, pp. 1-45, 1951.

[14] H. R. Lourenco, O. C. Martin, and T. Stutzle, “Iterated Local Search,” Handbook of Metaheuristics, International Series in Operations Research & Management Sci., vol. 57, pp. 320-353. Springer, Heidelberg, 2003.

[15] K. M. Ng, A. Gunawan, and K. L. Poh, “A hybrid Algorithm for the quadratic assignment problem,” in *Proc. of the International Conference on Conference on Scientific Computing*, pages 14-17, Nevada, USA, July 14-17 2008.