Fine-Tuning Algorithm Parameters Using the Design of Experiments Approach

Aldy Gunawan, Hoong Chuin Lau, and Lindawati

School of Information Systems, Singapore Management University,
80 Stamford Road, S(178902), Singapore
{aldygunawan,hclau,lindawati.2008}@smu.edu.sg

Abstract. Optimizing parameter settings is an important task in algorithm design. Several automated parameter tuning procedures/configurators have been proposed in the literature, most of which work effectively when given a good initial range for the parameter values. In the Design of Experiments (DOE), a good initial range is known to lead to an optimum parameter setting. In this paper, we present a framework based on DOE to find a good initial range of parameter values for automated tuning. We use a factorial experiment design to first screen and rank all the parameters thereby allowing us to then focus on the parameter search space of the important parameters. A model based on the Response Surface methodology is then proposed to define the promising initial range for the important parameter values. We show how our approach can be embedded with existing automated parameter tuning configurators, namely ParamILS and RCS (Randomized Convex Search), to tune target algorithms and demonstrate that our proposed methodology leads to improvements in terms of the quality of the solutions.

Keywords: parameter tuning algorithm, design of experiments, response surface methodology.

1 Introduction

It is well-known that good parameter settings have a significant effect on the performance of an algorithm (Eiben et al., 1999; Hutter et al., 2010). For example, a simulated annealing algorithm is sensitive to the cooling factor, while a tabu search algorithm relies on a good choice of the tabu tenure. Many of the works we witness to date propose algorithms where the underlying parameters are set either arbitrarily without explanation, or conveniently choose parameter values that have been reported in previous studies.

In response to the need for a principled approach to find good parameter settings, several automated approaches have been proposed in recent years. For model-based approaches, Díaz and Laguna (2006) developed CALIBRA which employs a Taguchi fractional experimental design followed by a local search procedure. The former focuses on providing the starting point of the experiment, while the latter continues to search for the best parameter configuration. This procedure can only handle up to five parameters and focuses on the main effects of parameters without exploiting the

C.A. Coello Coello (Ed.): LION 5, LNCS 6683, pp. 278–292, 2011.

interaction effects between parameters. SPO+ (Hutter et al., 2010) is an improved model-based technique extended from the Sequential Parameter Optimization framework that constructs predictive performance models to focus attention on promising regions of a design space, aimed at tuning target algorithms with continuous parameters and a single problem instance at a time. F-Race (Birattari et al., 2002) is the specialization of the generic class of racing algorithms for configuration of metaheuristics.

For model-free approaches, Hutter et al. (2009) presented a local search approach, ParamILS, for algorithm configuration which is suited for discrete parameters. Again, ParamILS only considers changing one single parameter value at a time. Much potential in the use of statistical testing methods as well as RSM in algorithm configuration problems were also discussed. Randomized Convex Search (RCS) was recently proposed to handle both discrete and continuous parameter values (Lau and Xiao, 2009). The underlying assumption of RCS is that the points lie inside the convex hull of a certain number of the best points (parameter configurations).

The Design of Experiments (DOE) is a well-established statistical approach that involves experiment designs for the empirical modeling of processes (see for example Montgomery, 2005). Some typical applications of DOE include 1) evaluation and comparison of basic design configurations, 2) evaluation of different materials, and 3) selection of design parameters. The proposal for exploiting DOE for algorithm parameter tuning is in fact not new. Barr et al. (1995) discussed the design of computational experiments to test heuristic methods and provided guidelines for such experimentation. The performance of algorithm in computation experiments was affected by algorithm factors which include initial solution construction procedures and any parameters employed by the heuristic. The authors suggested the use of DOE in the process of planning an experiment.

Parsons and Johnson (1997) used statistical techniques, a central composite design embedded a fractional factorial design, to build a response surface for four parameters. This approach was applied to a genetic algorithm with applications to DNA sequence assembly. More recently, Ridge and Kudenko (2007) used the DOE approach to build a predictive model of the performance of a combinatorial optimization heuristic over a range of heuristic tuning parameter settings. However, the approach was only applicable to tuning Ant Colony System for the Travelling Salesman problem. There was no further comparison with other automated tuning approaches.

The Response Surface methodology (RSM) is a model-based approach within DOE that can be used to quantify the importance of each parameter, support interpolation of performance between parameter settings as well as extrapolation to previously-unseen regions of the parameter space (Hutter et al., 2010). Recently, Caserta and Voss (2009) adapted the RSM to fine-tune their Corridor Method for solving a block relocation problem in container terminal logistics. The values of parameters were restricted to discrete intervals due to the problem characteristics. Caserta and Voss (2010) presented a simple mechanism aimed at automatically fine tuning only a single parameter, the corridor width, of the corridor method for solving the DNA sequencing problem.

This paper describes a sequential experimental approach for screening and tuning algorithm parameters. Our approach is grounded on the DOE methodology as follows.

Consider an algorithm (called the target algorithm) to solve a particular problem that requires a number of parameters to be set prior to the execution of the algorithm. A factorial experiment design is applied to first screen and rank the parameters. Parameters which are determined to be *unimportant* (in that the solution quality is insensitive to the values of these parameters) are set to some constant values so that the resulting parameter space that needs to be explored is reduced. A first-order polynomial model based on RSM is then built to define the promising initial range for the important parameter values. We apply our proposed approach to two different automated tuning configurators, ParamILS (Hutter et al., 2009) and RCS (Lau and Xiao, 2009). Each configurator is applied to a target algorithm for solving the Traveling Salesman Problem (TSP) and Quadratic Assignment Problem (QAP), respectively.

In summary, the major contributions/highlights of this paper are as follows:

1. We propose the use of a factorial experiment design that enables to screen and rank the algorithm parameters. The screening process helps us to identify those unimportant parameters so they can be set into constant values. By focusing on important parameters, we reduce the parameter search space and target our search on the promising regions of the important parameter search space.
2. We propose the use of RSM to define the promising initial range for important parameter values that can be embedded to automated tuning procedures for improving the quality of solutions.

The remainder of this paper is organized as follows. Section 2 describes our proposed automated tuning framework. Section 3 provides a computational analysis of our proposed approach applied to two problems. Finally, we provide some concluding perspectives and future research plans in Section 4.

2 Automated Tuning Framework

The Automated Tuning problem is defined as follows:

Definition**:** *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}.*

In this paper, we assume all parameters to lie within numeric intervals. An example of the function value $H(x)$ is the average percentage deviation of the solution values obtained by *TA* using x as the parameter setting from the optimal values over the given set of instances. In our paper, the goal is to optimize x over the given set of training instances I_{tr} and subsequently verify the quality of this parameter setting on a set of testing instances.

A high-level view of our proposed automated tuning framework is given in Figure 1. The framework consists of three phases, (1) screening, (2) exploration, and (3) exploitation phases. In the following, we discuss the details of each phase.

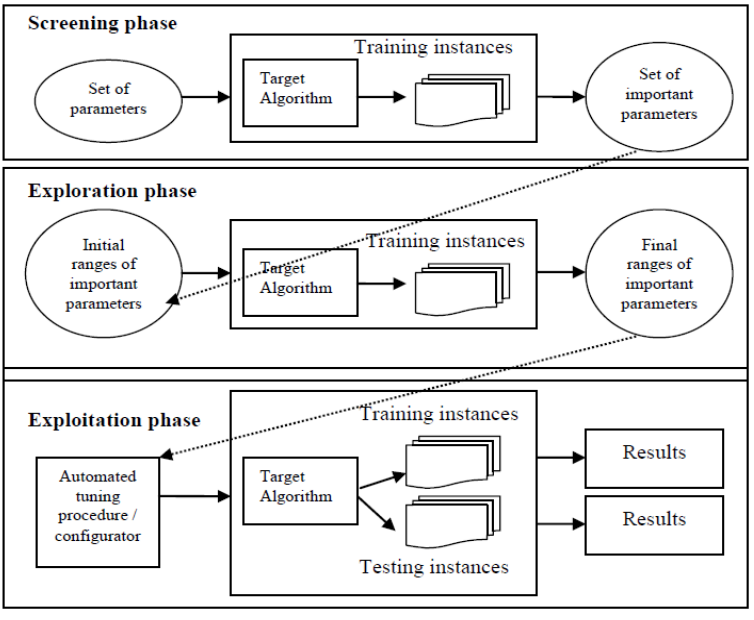

Fig. 1. Automated tuning framework

2.1 Screening Phase

Let k denote the number of parameters of the target algorithm to be tuned, and each parameter p_i (discrete or continuous) lies within a numeric interval $[l_i, u_i]$. In this phase, we perform screening to determine which parameters are significantly important thereby reducing the number of parameters under consideration. For this purpose, we apply a 2^k factorial design which consists of k parameters, where each parameter p_i only has two levels (l_i and u_i). A complete design requires $(2 \times 2 \times \ldots \times 2) \times n = n \times 2^k$ observations where n represents the number of replicates.

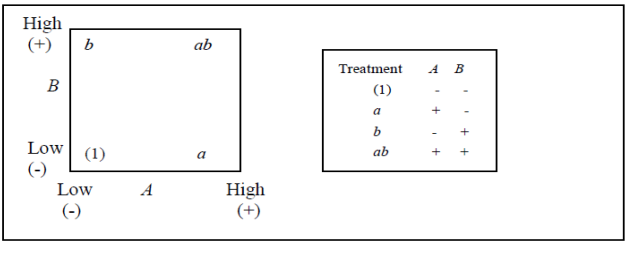

Fig. 2. The 2^2 factorial design

As an example, consider there are two parameters, A and B. Figure 2 shows the 2^2 design with treatment combinations are represented as the corners of the squares.

The signs + and – denote the values of l_i and u_i of each parameter p_i, respectively. In general, a treatment combination is represented by a series of lowercase letters (Montgomery, 2005). For example, treatment combination a indicates that parameters A and B are set to u_A and l_B, respectively. To estimate this treatment combination, we average n replications obtained. By using equations (1)–(3) and some other statistical testing (Montgomery, 2005), we can further examine the main effects of parameters A, B and the two-factor interaction AB as well.

$$A = \frac{1}{2n}\left[a + ab - b - (1)\right] \tag{1}$$

$$B = \frac{1}{2n}\left[b + ab - a - (1)\right] \tag{2}$$

$$AB = \frac{1}{2n}\left[ab + (1) - a - b\right] \tag{3}$$

The **importance** of a particular parameter is defined by conducting the test of significance on the main effect of the parameter. We choose a significance level (α = 5%) for our purpose. To further determine the **ranking** of the important parameters, we look at the absolute values of the main effects of those important parameters. By doing so, we can determine which parameters should be carefully controlled including the direction of adjustment for these parameters (see Figure 3 for illustration). The result in Figure 3 is obtained with the MINITAB statistical software.

Fractional Factorial Fit: Obj versus A, B, C

Estimated Effects and Coefficients for Obj (coded units)

Term	Effect	Coef	SE Coef	T	P
Constant		8.625	0.3062	28.17	0.000
A	1.750	0.875	0.3062	2.86	0.021
B	-3.000	-1.500	0.3062	-4.90	0.001
C	1.000	0.500	0.3062	1.63	0.141
A*B	-0.500	-0.250	0.3062	-0.82	0.438
A*C	-1.000	-0.500	0.3062	-1.63	0.141
B*C	-0.750	-0.375	0.3062	-1.22	0.256
A*B*C	0.250	0.125	0.3062	0.41	0.694

Analysis of Variance for Obj (coded units)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	3	52.2500	52.2500	17.4167	11.61	0.003
2-Way Interactions	3	7.2500	7.2500	2.4167	1.61	0.262
3-Way Interactions	1	0.2500	0.2500	0.2500	0.17	0.694
Residual Error	8	12.0000	12.0000	1.5000		
Pure Error	8	12.0000	12.0000	1.5000		
Total	15	71.7500				

Fig. 3. Statistical results of the screening phase

From Figure 3, we observe that the main effects of A and B are significant since the p-values of both effects are less than 5%. In terms of ranking, B is the most dominant parameter, followed by A. Assuming that our objective function is a minimizing function, we modify the range of each significant parameter by the main effect value of the parameter. For instance, parameter A should be set to a *low* value since its coefficient is positive; hence the range of parameter A is modified to $[l'_A, u'_A] = [l_A, l_A+2\Delta]$, where Δ is a constant. (This notation will become clear in the next section.)

For each unimportant parameter, we simply set to a constant value by the main effect value of the parameter; if the value is positive, we set the parameter to a low value (in our case, it is set to its lower bound l_c). The analysis of variance confirms our interpretation of the effect estimates. Both parameters *A* and *B* exhibit significant main effects.

2.2 Exploration Phase

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 $[l'_i, u'_i]$ (as defined in Section 2.1) as well as its centre point value $(l'_i + u'_i)/2$. The Exploration phase is summarized in the following figure.

Procedure ExplorationPhase
Input: *TA*: Target Algorithm with m parameters,
Θ: Parameter Configuration Space, defined by each parameter p_i having initial range $[l'_i, u'_i]$;
I: Set of Training Instances;
Output: Modified configuration space, each parameter with modified interval.

Procedure:
1: Run *TA* with respect to configuration space Θ on *I*;
2: Implement 2^{m+1} factorial design on m parameters ;
3: Conduct the interaction and curvature tests. If at least one of the tests is statistically significant, stop. Otherwise, go to Step 4;
4: Build a planar model of significant parameters;
5: Apply steepest descent to define a new centre point for each important parameter p_i;
6: Update the range of each important parameter p_i and generate a new $[l'_i, u'_i]$;
7: If at least one parameter p_i with either $l'_i < l_i$ or $u'_i > u_i$, stop. Otherwise, go to Step 1;

Fig. 4. Exploration phase

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 (each parameter has two possible values, with an additional parameter setting defined by the centre point value of each parameter).

We apply a factorial experiment design in order to build a first-order (planar) 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:

$$Y = \beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m + \varepsilon \tag{4}$$

In order to test the significance of this model, we conduct two additional statistical tests:

- *Interaction test*. This test is mainly on testing whether any interaction between parameters. This can be done by looking at the significance of the estimated coefficient between two parameters (for instance, β_{ij}).
- *Curvature test*. This test is mainly on testing whether the planar model is adequate to represent the local response function.

As long as each test is not 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 Y (Box and Wilson, 1951). The path is proportional to the signs and magnitudes of the equation (4). For example, if β_A (coefficient of parameter A) is the largest absolute coefficient value compared against other coefficient values, the step size of another parameter i is calculated by β_i/β_A. Several points along this path of steepest descent would be generated. A point with the minimum objective function value is then selected as the new centre point. A new set of l_i and u_i values for each parameter p_i as well as a new parameter configuration space Θ are then determined.

We illustrate the steepest descent step as follows. Assuming two parameters, A and B, where A has the larger absolute coefficient value (ties broken randomly). We first generate n possible values of x_A and x_B as follows: the values of x_A are set to arbitrary values (e.g., 0.1, 0.2, …, 0.9), whereas the corresponding values of x_B are calculated by $(\beta_B/\beta_A)\times x_A$. Finally, the n possible parameter values for A and B are calculated as follows:

$$V_A^n = x_A \times \frac{(u_A - l_A)}{2} + \frac{(l_A + u_A)}{2} \tag{5}$$

$$V_B^n = x_B \times \frac{(u_B - l_B)}{2} + \frac{(l_B + u_B)}{2} \tag{6}$$

We then run the target algorithm with these n parameter values of A and B. The parameter setting with the minimum objective function value, denoted by V_A^{best} and V_B^{best}, is selected as a new centre point. The range of is parameter then modified as [V_i^{best} -Δ, V_i^{best} +Δ] where Δ is a constant.

From statistical point of view, the region of planar local optimality is indicated by the existence of either interaction or curvature. Hence, we conduct the experiments until either interaction test or curvature test is statistically significant and proceed to the exploitation phase.

2.3 Exploitation Phase

In this phase, we drop the planarity assumption and devote our attention to finding the optimal point in the region output from the exploration phase. This is achieved by applying an automated tuning procedure, such as ParamILS (Hutter et al., 2009) or RCS (Lau and Xiao, 2009).

In this study, ParamILS is applied to tune the Iterated Local Search algorithm (Lourenco et al., 2003) for the Traveling Salesman Problem, while RCS is applied to the hybrid algorithm combining Simulated Annealing and Tabu Search (Ng et al., 2008) for the Quadratic Assignment Problem (QAP).

3 Experimental Results

In this section, we report a suite of computational results and analysis obtained from our proposed approach. All the experiments are run on a Intel (R) Core (TM)^2 Duo CPU 2.33 GHz with 1.96GB RAM that runs Microsoft Windows XP.

To evaluate the performance of our proposed automated tuning framework, we conduct two different experiments: 1) test ParamILS on Traveling Salesman Problem (TSP), and 2) test RCS on Quadratic Assignment Problem (QAP). For each experiment, two different scenarios, configurator+DOE (1st scenario) and configurator (2nd scenario), would be analyzed and compared. In this case, the amount of resources allocated (i.e. the number of iterations) are fixed. For instance, suppose the number of iterations of ParamILS and DOE are x and y respectively, the number of iterations of the 1st scenario is x+y, while the number of iterations of the 2nd scenario is set to z, with $z = x+y$.

The main purpose is to show that our approach can lead to 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. We show that our proposed approach could provide better solutions for both discrete and continuous parameter values.

3.1 Traveling Salesman Problem (TSP)

The target algorithm to solve TSP is the Iterated Local Search (ILS). In this paper, we used the implementation from Halim et al. (2007). Four parameters that need to be tuned are as follows (Table 1):

- *Maximum_number_of_iterations* that limits the number of iterations for running the algorithm.
- *Perturbation_strength* that limits the number of times required for running the perturbation.
- *Non_improving_moves_tolerance* that limits the number of non-improving moves to be accepted.
- *Perturbation_choice* that selects the perturbation strategy.

Table 1. Parameter space for ILS on TSP

Parameters (p_i)	Range
Maximum_number_of_iterations (max_iter)	[100, 900]
Perturbation_strength (perturb)	[1, 10]
Non_improving_moves_tolerance (non_imprv)	[1, 10]
Perturbation_choice (opt_cho)	[3, 4]

In this screening process, the parameter space for *max_iter*, *perturb*, *non_improv* are reduced to [100, 500], [1, 5] and [1, 5] respectively. We started by selecting 47 instances from the 70 instances (TSPLIB) as training instances while the rest (23 instances) are treated as testing instances. For a particular parameter setting, we take the average of 10 runs on the training instances. The details of the experiment would be explained below.

3.1.1 Screening Phase

As described in Section 2.1, we focus on determining which parameters are significantly important. Figures 5 and 6 present the results of a 2^4 factorial design with $n = 10$ replicates using the factors mentioned in Table 1. The numerical estimates of the effects indicate that the effect of *max_iter*, *perturb*, and *non_imprv* are significant (with p-value < 5%), while the effect of *opt_cho* appears small. Based on the coefficient value of parameter *opt_cho* obtained, we decide to set the value of this parameter to its lower bound value (l_{opt_cho}). As we can see from Figure 6, only three parameters (*max_iter*, *perturb*, and *non_imprv*) have significant effects. The dotted line represents the cut-off limit associated with that significance level.

Fractional Factorial Fit: Dev versus Max_iter, Perturb, ...

Estimated Effects and Coefficients for Dev (coded units)

Term	Effect	Coef	SE Coef	T	P
Constant		3.9972	0.05187	77.06	0.000
Max_iter	-0.2430	-0.1215	0.05187	-2.34	0.021
Perturb	0.9924	0.4962	0.05187	9.57	0.000
Non_impr	0.2277	0.1139	0.05187	2.20	0.030
Opt_cho	0.1966	0.0983	0.05187	1.90	0.060
Max_iter*Perturb	-0.0676	-0.0338	0.05187	-0.65	0.515
Max_iter*Non_impr	-0.0225	-0.0112	0.05187	-0.22	0.829

Analysis of Variance for Dev (coded units)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	4	45.376	45.3765	11.3441	26.35	0.000
2-Way Interactions	6	2.053	2.0528	0.3421	0.79	0.576
3-Way Interactions	4	0.992	0.9918	0.2479	0.58	0.681
4-Way Interactions	1	0.003	0.0025	0.0025	0.01	0.939
Residual Error	144	61.996	61.9964	0.4305		
Pure Error	144	61.996	61.9964	0.4305		
Total	159	110.420				

Fig. 5. Statistical results of the screening phase

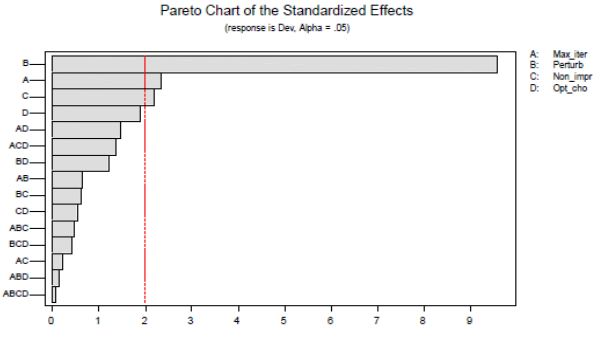

Fig. 6. Screening phase of ILS Algorithm

3.1.2 Exploration Phase

In this phase, we focus on three important parameters obtained from screening phase. We apply a factorial experiment design in order to build the first-order model. In order to test the significance of the first-order model, we conduct two additional statistical testing: interaction and curvature tests. As described earlier, as long as theses two additional tests are not significant, we can always assume that the first – order model is adequate to represent the true surface of parameters. Table 2 summarizes the parameter space of parameters along the path of the steepest descent.

Table 2. Parameter space for ILS Algorithm

Parameters	Range
	Exploration_1
max_iter	[400, 600]
Perturb	[1 ,3]
non_imprv	[4, 6]
opt_cho	3
Objective function value (%)	3.811

3.1.3 Exploitation Phase

In this phase, we use ParamILS to further explore neighbor parameters, given the information about the parameter values from exploration phase. Here, we would like to show that by using the DOE approach, we can provide a very good initial range for the parameter values.

Table 3. Parameter space for ILS on TSP

Parameters	Type	Range	
		ParamILS	ParamILS + DOE
Maximum_number_of_iteration	Discrete	[100, 900]	[400, 600]
Perturbation_strength	Discrete	[1, 10]	[1, 3]
Non_improving_moves_tolerance	Discrete	[1, 10]	[4, 6]
Perturbation_choice	Discrete	[3, 4]	3

Table 4. Parameter tuning for ILS on TSP

Algorithms	Mean
ParamILS (training instances)	2.653
ParamILS + DOE (training instances)	2.513
ParamILS (testing instances)	4.103
ParamILS + DOE (testing instances)	4.066

For comparison purpose, we also run ParamILS with the initial range for the parameter values (Table 3). The default parameter setting is based on the lower bound value of each parameter. The details tuning results for both ParamILS and ParamILS + DOE are given in Table 4. We observe that the results obtained by ParamILS + DOE

are better than those of ParamILS. We can conclude that DOE approach could lead to improvements in terms of the solution quality. The percentage deviations between the average objective function value of the solutions obtained and the best known/optimal solutions are only 1.117 % and 1.710% for training and testing instances, respectively.

3.2 Quadratic Assignment Problem (QAP)

In this experiment, the target algorithm to solve QAP is the hybrid algorithm (Ng et al. 2008). The hybrid algorithm involves using the Greedy Randomized Adaptive Search Procedure (GRASP) to obtain an initial solution, and then using a combined Simulated Annealing (SA) and Tabu Search (TS) algorithm to improve the solution. There are four parameters to be tuned, which are listed as follows:

- Initial temperature of SA algorithm (*temp*)
- Cooling factor (*alpha*)
- Length of tabu list (*length*)
- Percentage of number of non-improvement iterations prior to intensification strategy (*pct*).

In order to evaluate the performance of our proposed approach, we decided to solve some benchmark problems from a library for research on the QAP (QAPLIB) which have been studied and solved by other researchers (Burkard et al., 1997). According to Taillard (1995), the instances of QAPLIB can be classified into four classes: unstructured (randomly generated) instances, grid-based distance matrix and real-life instances and real-life-like instances. Due to the limitation of the target algorithm that can only solve symmetric instances with zero diagonal values, we only focus on some instances from three classes: unstructured (randomly generated) instances, grid-based distance matrix and real-life instances.

3.2.1 Screening Phase

We selected a certain number of instances for training and testing instances for each class (Table 5). Table 6 summarizes the initial range for each parameter value. Only parameter *length* is a discrete parameter while the rest are continuous ones.

Table 5. Training and testing instances for each class

Class	Training instances	Testing instances
Unstructured (randomly generated) instances	11 instances	5 instances
Grid-based distance matrix	24 instances	11 instances
Real-life instances	14 instances	7 instances

Table 6. Parameter space for hybrid algorithm on QAP

Parameters	Type	Range
Temp	Continuous	[100, 7000]
Alpha	Continuous	[0.5, 0.95]
Length	Discrete	[5, 10]
Pct	Continuous	[0.01. 0.10]

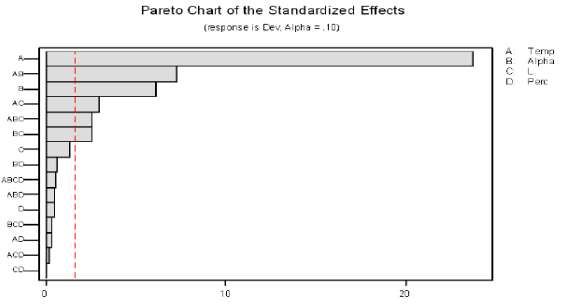

Fig. 7. Screening phase of the hybrid algorithm (unstructured instances)

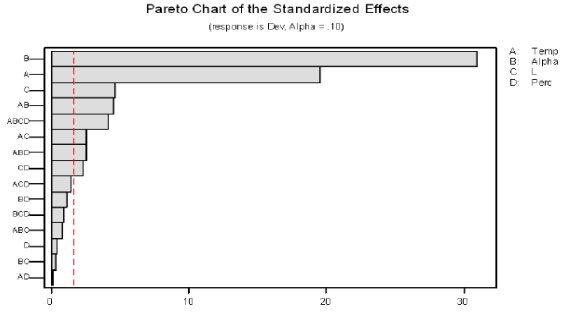

Fig. 8. Screening phase of the hybrid algorithm (grid-based distance matrix)

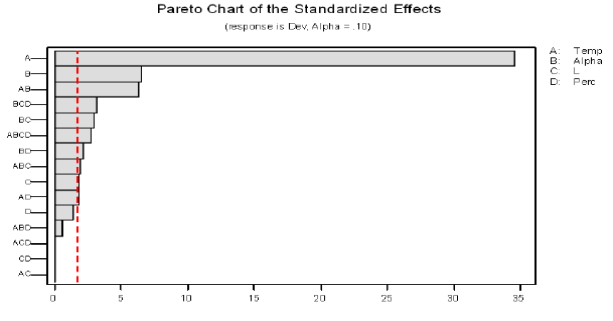

Fig. 9. Screening phase of the hybrid algorithm (real-life instances)

3.2.2 Exploration Phase

In this phase, we again focus on important parameters obtained from screening phase. By applying the same approach discussed in Section 3.1, we conduct the experiment until the first-order model is not appropriate for each class.

The parameter spaces of parameters along the path of the steepest descent are summarized in Tables 7, 8 and 9. We observe that the objective function value would decrease subsequently when we reach the promising region of the parameter values. The last column for each table represents the final range for each parameter that would be used as an input in exploitation phase.

Table 7. Parameter space for hybrid algorithm on QAP (unstructured instances)

Parameters	Range	
	Exploration_1	Exploration_2
Temp	[4000, 6000]	[4378, 6348]
Alpha	[0.85, 0.95]	[0.935, 0.945]
Length	5	5
Pct	0.01	0.01
Objective function value (%)	2.517	2.108

Table 8. Parameter space for hybrid algorithm on QAP (grid-based distance matrix)

Parameters	Range	
	Exploration_1	Exploration_2
Temp	[4000, 6000]	[4238, 6238]
Alpha	[0.85, 0.95]	[0.935, 0.945]
Length	[4, 6]	6
Pct	0.1	0.1
Objective function value (%)	0.591	0.425

Table 9. Parameter space for hybrid algorithm on QAP (real-life instances)

Parameters	Range
	Exploration_1
Temp	[4000, 6000]
Alpha	[0.85, 0.95]
Length	[4, 6]
Pct	0. 1
Objective function value (%)	9.255

3.2.3 Exploitation Phase

In this phase, the final range for parameter values obtained from exploration phase would be compared with the default configuration of RCS (Table 10). The results obtained by testing two different scenarios, RCS and RCS + DOE, are given in Table 11. We can conclude that RCS + DOE outperforms RCS in all groups of instances. We obtained improvements of results over RCS for both training and testing instances.

Table 10. Parameter space for hybrid algorithm on QAP

Parameters	Range			
	RCS	RCS + DOE (unstructured instances)	RCS + DOE (grid-based distance matrix)	RCS + DOE (real-life instances)
Temp	[100, 7000]	[4378, 6348]	[4238, 6238]	[4000, 6000]
Alpha	[0.5, 0.95]	[0.935, 0.945]	[0.935, 0.945]	[0.85, 0.95]
Length	[5, 10]	5	6	[4, 6]
Pct	[0.01. 0.10]	0.01	0.10	0.1

Table 11. Parameter Tuning for Hybrid Algorithm on QAP

Algorithms	Mean (unstructured instances)	(grid-based distance matrix)	(real-life instances)
RCS (training instances)	1.100	0.630	3.264
RCS + DOE (training instances)	0.938	0.190	2.822
RCS (testing instances)	1.595	1.158	6.770
RCS + DOE (testing instances)	1.518	0.754	5.985

4 Conclusion

This paper proposes an automated tuning framework based on the Design of Experiments (DOE) approach. We demonstrate that our approach can be adapted to address the parameter tuning problem for target algorithms that find approximate solutions to two combinatorial optimization problems, TSP and QAP. We show that the proposed approach performs very well for both discrete and continuous parameter value settings.

One limitation of a factorial experiment design is that the number of experiments increases exponentially with the number of parameters. Fractional factorial designs offer a manageable alternative, which uses only some subset of a full factorial design's run.

In ParamILS and RCS, the neighborhoods of the current parameter setting are usually randomly selected. For future extensions to this work, we can consider using a second-order response surface model which is usually required when the experimenter is relatively close to the optimum.

References

1. Adenso-Diaz, B., Laguna, M.: Fine-Tuning of Algorithms Using Fractional Experimental Design and Local Search. Operations Research 54(1), 99–114 (2006)
2. Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C., Stewart, W.R.: Designing and Reporting on Computational Experiments with Heuristic Methods. Journal of Heuristics 1, 9–32 (1995)
3. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A Racing Algorithm for Configuring Metaheuristics. In: Proc. Of the Genetic and Evolutionary Computation Conference, pp. 11–18. Morgan Kaufmann, San Francisco (2002)
4. Box, G., Wilson, K.: On the Experimental Attainment of Optimum Conditions. Journal of the Royal Statistical Society Series b 13, 1–45 (1951)
5. Burkard, R.E., Karisch, S.E., Rendl, F.: QAPLIB – A Quadratic Assignment Problem Library. Journal of Global Optimization 10, 391–403 (1997)
6. Caserta, M., Voß, S.: A Math-Heuristic Algorithm for the DNA Sequencing Problem. In: Blum, C., Battiti, R. (eds.) LION 4. LNCS, vol. 6073, pp. 25–36. Springer, Heidelberg (2010)
7. Caserta, M., Voß, S.: Corridor Selection and Fine Tuning for the Corridor Method. In: Stützle, T. (ed.) LION 3. LNCS, vol. 5851, pp. 163–175. Springer, Heidelberg (2009)

8. Halim, S., Yap, R., Lau, H.C.: An Integrated White+Black Box Approach for Designing and Tuning Stochastic Local Search. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 332–347. Springer, Heidelberg (2007)
9. Hutter, F., Hoos, H.H., Leyton-Brown, K., Murphy, K.: Time-Bounded Sequential Parameter Optimization. In: Blum, C., Battiti, R. (eds.) LION 4. LNCS, vol. 6073, pp. 281–298. Springer, Heidelberg (2010)
10. Hutter, F., Hoos, H.H., Leyton-Brown, K., Stützle, T.: ParamILS: An Automatic Algorithm Configuration Framework. Journal of Artificial Intelligence Research 36, 267–306 (2009)
11. Lau, H.C., Xiao, F.: A Framework for Automated Parameter Tuning in Heuristic Design. In: 8th Metaheuristics International Conference, Hamburg, Germany (2009)
12. Lourenco, H.R., Martin, O.C., Stutzle, T.: Iterated Local Search. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics. International Series in Operations Research & Management Sci., vol. 57, pp. 320–353. Springer, Heidelberg (2003)
13. Montgomery, D.C.: Design and analysis of Experiments, 6th edn. John Wiley and Sons Inc., Chichester (2005)
14. Ng, K.M., Gunawan, A., Poh, K.L.: A hybrid Algorithm for the Quadratic Assignment Problem. In: Proc. International Conference on Conference on Scientific Computing, Nevada, USA, pp. 14–17 (2008)
15. Parsons, R., Johnson, M.: A Case Study in Experimental Design Applied to Genetic Algorithms with Application to DNA Sequence Assembly. Journal of Mathematical and Management Sciences 17(3), 369–396 (1997)
16. Ridge, E., Kudenko, D.: Tuning the Performance of the MMAS Heuristic. In: Stützle, T., Birattari, M., Hoos, H.H. (eds.) SLS 2007. LNCS, vol. 4638, pp. 46–60. Springer, Heidelberg (2007)
17. Taillard, E.D.: Comparison of Iterative Searches for the Quadratic Assignment Problem. Location Science 3(2), 87–105 (1995)