Identifying Early Termination Criteria to Meet Accuracy Requirements in the MMAS Heuristic

Chase W. Gruszewski and Matthew H. Henry

***Abstract*—This paper examines the selection of algorithm termination criteria in the Max Min Ant System to meet specified accuracy requirements. Three separate types of termination criteria: iterations, total stagnations, and iteration stagnations are examined for predictability. The study takes a Design of Experiments approach, using Box-Behnken Response Surface Methodology to examine the interactions between algorithm parameters, problem characteristics and solution accuracy relative to the algorithm's solution at convergence. The response surfaces are tested for sensitivity and validated for predictive use. We find that it is possible to predict termination criteria that will meet accuracy requirements and that the iterations termination criterion provides the most predictable accuracy.**

***Index Terms*—Ant colony optimization, design of experiments, metaheuristic algorithms, multi-agent learning.**

I. INTRODUCTION

Ant Colony Optimization (ACO) is a metaheuristic designed to provide approximate solutions to a variety of complex combinatorial problems. As a stochastic algorithm, its implementation in a system or decision-making framework requires setting algorithm parameters and selecting termination criteria. Significant research effort has focused on tuning specific algorithm parameters and the influence of the problem set characteristics on solution quality. However, the selection of termination criteria has received little attention: most researchers have set the criteria without any experimental design or otherwise rigorous basis. Multiple types of termination criteria have been used in practice, including: time limits, number of tours constructed and number of stagnant iterations. The lack of focus on any particular criterion has complicated the problem of determining optimal termination criteria.

If a correlation between algorithm settings, problem instance characteristics, required accuracy, and optimal termination criteria can be established, researchers can conclude algorithm runs sooner with confidence in the quality of the solution. Our research aims to determine if termination criteria can be established in advance, with assurances of reasonable accuracy at termination. We compare multiple types of termination criteria to determine which yield the most predictable results.

Our approach uses a Design of Experiments that runs algorithms for an extended period and then looks backwards, asking the question "If it was only necessary to be within a certain percentage of the algorithm's solution at convergence, how early could the algorithm have been stopped?" We find that, when using iterations and stagnations as termination criteria, the optimal algorithm stopping point can be predictably modeled and applied. Conversely, we concluded that using the number of tours as a basis for termination is not as reliable. The remainder of this paper discusses our approach using the Traveling Salesman Problem (TSP) as a mechanism for evaluating the suitability of different termination criteria and presents the results in more detail.

Manuscript received May 12, 2018; revised July 7, 2018.

C. W. Gruszewski was with Johns Hopkins University, Baltimore MD 21075 USA (e-mail: cgrusze1@jhu.edu).

M. H. Henry is with the Applied Physics Laboratory, Johns Hopkins University, Baltimore MD 21075 USA (e-mail: mhhenry@jhu.edu).

II. BACKGROUND

The ACO metaheuristic falls within the category of swarm intelligence, where complex solutions are derived from the iterative application of simple rules and calculations, with minor variations. Each iteration impacts the behavior of the swarm, and the swarm collectively gravitates to an acceptable, though not always optimal, solution or action.

ACO's methodology is derived from its namesake: ants. An ant colony, through the use of pheromone deposition, is able to identify the shortest path to a food source, in spite of the fact that each individual ant is virtually blind. Ants deposit pheromones along the shorter path more frequently, resulting in a stronger pheromone trail, which subsequent ants reinforce by traversing the path [1]. Conversely, pheromones on longer paths evaporate, as they are traveled less frequently. Once all the pheromone has evaporated on a path, subsequent ants are no longer attracted to that potential route.

ACO models this behavior through the implementation of a mechanism for stigmergy in algorithms and multi-agent systems [2]. In computational algorithms, this requires the inclusion of a pheromone variable, stored on the solution graph. In multi-agent systems, the stigmergy implementation requires an environmental abstraction or agent that each ant can query and influence [3], [4].

ACO is particularly effective in solving dynamic problems, where decisions and option costs change during solution construction [3]. This is analogous to the metaheuristic's natural metaphor. If an obstacle is placed in the path of an existing pheromone trail, the ant colony will follow the trail to the point of an obstruction, and then switch from path exploitation to exploration in order to find a suitable detour around the obstacle. Once a detour is discovered, the pheromone trail is reestablished and the colony will revert to exploitation of the known path. Similarly, once the obstacle is removed, the colony will progressively reestablish the original, shortest path to the food.

doi: 10.18178/ijmlc.2018.8.4.710

As a result, ACO has been used to solve problems where incomplete information is available or the problem evolves while the solution is being calculated [5]. It has been applied in both computational algorithms for solving defined problems and in multi-agent systems, e.g. automated fault management systems designed to monitor, diagnose, and solve system health in real time. It has also been used in situations where the detection of several unknown and unrelated conditions may be required to draw a conclusion, such as software virus detection [6] or to solve complex nonlinear problems with higher levels of dimensionality [7].

III. Related Work

Marco Dorigo [8] introduces ACO as a mechanism to solve complicated problems with non-convex solution spaces. His initial implementation, dubbed the Ant Colony System (AS), modeled the collective problem-solving behaviors of cooperating ants by way of virtual "pheromone" deposition and evaporation. This enabled a stochastic search of the solution space of a problem with iterative reinforcement of promising solutions.

Dorigo and Gambardella improved the performance of AS by introducing the concept of an elitist strategy [9]. In an elitist strategy, only the ant that constructs the best tour in an iteration deposits pheromone along the traveled edges, as opposed to all ants. Additionally, they introduced the exploitation/exploration ratio, which is a percentage of time that an ant will automatically choose the lowest cost edge, instead of a probability distribution.

Stutzle and Hoos introduced the Max Min Ant System (MMAS) as an improvement over previous ant systems [10]. Its key features are upper and lower limits on pheromone levels on edges, in addition to an elitist strategy. With the adoption of pheromone limits, they introduced the concept of convergence, where each node has only one edge with a pheromone value of τ_{max} and all other edges have a value of τ_{min}. Using this definition, they demonstrated that a termination criterion of 2500n tour constructions achieved convergence in every case.

Dorigo and Stutzle surveyed variations of ACO and found that the implementations with stronger exploitation of the "learned" pheromone trails tended to perform better [3]. They also noted the importance of balancing α and β in addition to elitist strategies, candidate lists, and exploitation/exploration ratios. The ACO code currently available at (http://iridia/ulb.ac.be/~mdorigo/ACO/aco-code/public-software.html) includes the implementation of MMAS with these additional parameters.

Ridge and Kudenko outlined a framework for using Design of Experiments based on Response Surface Methodology (RSM) and Minimum Run Resolution V Design to address the multiple degrees of freedom in tuning ACO parameters [11]. They developed a model that correlated parameter settings, solution set size and the problem set's standard deviation to both algorithm accuracy and run time. They used iteration stagnations (number of iterations without improvement) as a termination criterion. This approach provided the basis for using RSM to model MMAS behavior. However, by balancing time and accuracy in the same response surface, the model produces optimizations on complex problems that converge prematurely, and have unpredictable accuracies.

They extended their methodology to demonstrate the significance of the problem set's standard deviation of edge costs on ACO performance [12]. In this study, they used iterations for the stopping criterion and noted that a "more formal detection of possible differences introduced by different stopping criteria would have required a different experiment design."

Stutzle *et al.* found that the original AS was insensitive to parameter tuning [13]. However, the MMAS was more responsive and improved performance as they adjusted parameters on pre-scheduled criteria. They utilized time limits for termination criteria to study the impact of parameter tuning.

Fallahi *et al.* examined and updated Ridge & Kudenko's Design of Experiments methodology [11] with a comprehensive framework for metaheuristics and improved modeling of the response surface. [14] They took the viewpoint that the number of algorithm iterations is a design factor to be tuned to the specific problem at hand, and used response surface modeling to optimize the termination criteria for clusters of specific problem instances. However, the only termination criterion used in their research was the number of iterations. They also balanced algorithm accuracy and algorithm run time within the same response model, causing competing objectives for parameter tuning.

Byerly and Uskov attempted to further improve the ACO performance by oscillating the α and β parameters throughout the run, in order to prevent premature convergence in a MMAS based ACO design [15]. In this experiment, the α and β parameters were varied in a sinusoidal pattern during shortened ACO runs, to derive higher quality solutions more quickly. They used multiple termination criteria including the total number of iterations, the number of stagnations, and clock time. These criteria were selected arbitrarily, with the objective of stress-testing the parameter oscillation approach.

Our research differs from previous work in that we do not attempt to optimize algorithm run time and algorithm accuracy simultaneously. Instead, we set out to determine the earliest point at which we can have a high degree of confidence that the algorithm's solution is within a required accuracy, and can therefore, be terminated.

Our study also differs from past work by using a Box-Behnken Response Surface Methodology. Previous work used Central Composite Design (CCD) due to its well-known usage rather than for any specific technical reason. Within the CCD family, many researchers selected the Face Centered Composite (FCC) design, because it allows construction of the model with parameters entirely within the design space, as opposed to Circumscribed Central Composite (CCC) designs, which require parameters to be set at extremes outside the design space.

The use of Box-Behnken offers a few advantages over FCC. Primarily, it does not require all factors to be simultaneously set to their extreme values—thus reducing the resource load created by combined factor extremes. Additionally, the Box-Behnken model produces a fully rotatable design unlike the FCC [16]. Finally, the Box-Behnken model permits a reduced number of runs, resulting in lower resource demands. Box-Behnken only requires 130 runs for a nine-factored model, as opposed to the FCC, which requires 156 runs.

IV. Preliminaries

A. Symmetric Traveling Salesman Problem

We apply the MMAS to finding solutions to the symmetric Traveling Salesman Problem (TSP) in our experiments. A TSP consists of a series of cities at fixed distances from each other. The objective in the TSP is to find the shortest route that visits each city once, and then returns to the beginning. A symmetric TSP is an undirected graph, which means the distance and pheromone values between two cities are equal in both directions.

B. Essential Elements of ACO

In the original implementation of ACO as the Ant Colony System (AS) [8], ants traverse a problem graph across decision nodes via edges that connect them. At each decision node, the ant randomly selects an edge based on the probability distribution given by (1).

Within (1): *P* is the probability that an ant on node *i*, at time *t*, will select the path to node *j* from the number of available paths *k*; τ is the pheromone information stored on the path; η is the heuristic information at each option; α and β are tuned parameters, which adjust the importance of pheromone information and heuristic information, respectively [3].

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} * [\eta_{il}]^{\beta}} \quad (1)$$

Each ant continues traveling the graph until it has completed a tour. Once a tour is completed, the pheromone on the edges of the graph evaporates and ants deposit pheromone according to (2) where ρ is the evaporation rate, L_k is the tour for ant *k*; *m* is the number of ants, and $\Delta\tau^{k}_{ij}$ is the pheromone deposition by ant *k* along its path at time *t*.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

$$\text{where } \Delta\tau_{ij}^{k} = \frac{1}{L_k} \quad (2)$$

C. Max Min Ant System (MMAS)

ACO implementation requires that the potential solution set, or search area, be modeled as a graph consisting of decision points as nodes and options as edges that connect the decision points. [5] Each edge has a heuristic value, η, specific to the problem at hand. With the TSP, the heuristic value for each edge between cities is 1/d, where d is distance between cities.

Each edge is also characterized by a pheromone value, τ. Unlike the heuristic value, which is unique to the problem instance and static throughout the algorithm run, the pheromone value changes by pheromone deposition and evaporation in each algorithm iteration.

Fig. 1 is a graphical representation of this concept with solid lines representing heuristic information and dotted lines representing pheromone information. The graph shows the paths available to an ant on node 1. In a generic ACO implementation, the probability of an ant traveling from node 1 to any of the other 4 nodes is calculated by (1).

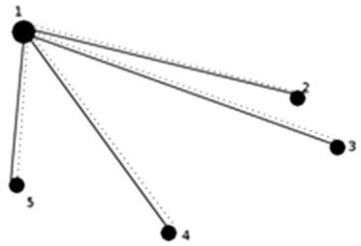

Fig. 1. An ACO problem representation.

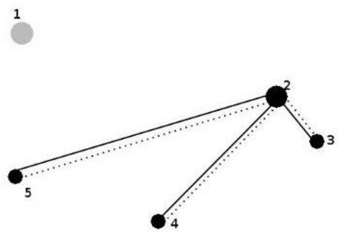

Fig. 2. Options after traveling from node 1 to 2.

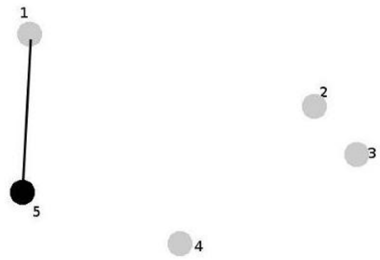

Fig. 3. An ant concludes its TSP tour.

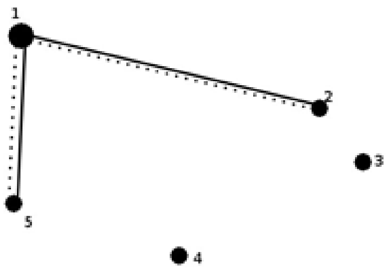

Fig. 4. Candidate list length of 2.

If the ant travels from node 1 to node 2, its path options would then appear as in Fig. 2. In the case of a TSP instance, the ant continues constructing a tour until it has traveled to all graph nodes then returns to its starting location, as seen in Fig. 3. This is a complete tour.

The MMAS algorithm builds on this process with some variations. Ants, the quantity of which is assigned the variable *m*, are randomly placed on nodes throughout the graph at the beginning of each iteration. Each ant then begins constructing a tour using candidate lists.

Candidate lists enumerate the number of options an ant evaluates for travel at each node. Truncating a candidate list can provide computational advantages. When a candidate list is truncated, only the top candidate edges, based on values calculated by Fig. 1, are evaluated at each node. For example, Fig. 4 shows the paths an ant on node 1 could possibly take if the candidate list length were two. The length of the candidate list, assigned the variable *nnAnts*, is an algorithm parameter that we employ in the evaluation.

Another parameter we employ is the exploitation/exploration ratio assigned the variable *q*0. At each decision node, a random number, q, is chosen between zero and one. If q is less than $q0$, then the ant chooses to exploit: it deterministically selects to travel along the available edge with the highest value calculated by (1). If q is greater than $q0$, then the ant chooses to explore by probabilistically selecting the next edge to travel from the candidate list according to the probability distribution given by (1).

Once every ant in an iteration finishes a complete tour, the total distance of each tour is compared. The shortest tour in that iteration is the iteration best solution s^{ib} where i is the iteration, b denotes best, and the ant that traveled it is the *iteration best ant*.

Pheromone is evaporated from every edge in the graph according to the algorithm's evaporation factor, ρ. Next, each edge along the path traveled by *iteration best ant* receives a pheromone deposition equal $1/s^{ib}$. No other edges receive additional pheromone.

Finally, edge pheromone values are adjusted to ensure no edge has a pheromone value outside of the algorithm's maximum/minimum (MaxMin) pheromone range. Pheromone values falling outside the MaxMin range are adjusted to the Max or Min limit as appropriate. If a new *global best* tour has been found, then the MaxMin ranges are adjusted according to (3) and (4) [10]. Pheromone updating is the last step of an MMAS iteration. If the termination criterion has not yet been met, then the ants are removed from the graph and a new iteration is started with updated pheromone levels at each edge. If the termination criterion has been met, then the algorithm run concludes, and the best solution found across all iterations is reported.

$$\tau_{max} = \frac{1}{1-\rho}\frac{1}{s^{gb}} \tag{3}$$

$$\tau_{min} = \frac{\tau_{max}}{2n} \tag{4}$$

V. Problem Description

The problem we address here is to answer the following two questions:

1) Given a set of MMAS parameter values, can we prescribe a termination criterion with a measurable degree of confidence in its quality? Specifically, can we estimate a model that prescribes a specific termination criterion for a given set of MMAS parameter values and problem characteristics?
2) What factors contribute most to the optimal termination criterion?

There are multiple options for selecting an ACO termination criterion. Generally, researchers have used a maximum number of algorithm iterations, algorithm tours, a time limit, or a number of iterations of stagnation. The number of tours is equal to the number of iterations multiplied by the number of ants, as each ant constructs one tour during an iteration. The time limit criterion, while commonly used and included in the source code, is not a reproducible result without identical hardware and configurations. Therefore, we did not examine time limits in this study. In the case of limiting the number of iteration stagnations, the algorithm terminates after a predetermined number of iterations are performed without improvement in the algorithm's best tour cost.

A. Solution Quality

We use relative error, defined as the percentage difference between the algorithm's solution and some baseline solution, as the measure for a solution's quality. In each instance where relative error is used, we have identified the baseline used in that instance.

B. Optimal Termination Criteria

We define optimal termination criteria as criteria that will stop the algorithm run as soon as the solution reaches a required level of accuracy relative to the algorithm's solution at convergence.

VI. Methodology

A. Objectives

There were several objectives to this experiment. The primary objective was to determine if termination criteria, for a required accuracy, can be predictably modeled. Secondarily, we wanted to identify which type of termination criteria provided the most predictable results. Finally, we wanted to examine which algorithm parameters and problem characteristics have the greatest effect on the optimal termination criteria for the MMAS ACO, and how broadly these effects could be applied.

B. Box-Behnken RSM

Response Surface Methodology (RSM) is a design of experiments technique used to create a model where the response of interest is affected by many variables [17].

Box-Behnken designs for RSM provide data to fit linear and quadratic models for both first and second order coefficients with rotatable designs. The design must have three equally spaced levels per factor: low, medium and high. Unlike CCD RSM, Box-Behnken does not contain a factorial or fractional factorial design. Instead, these designs combine factorials with incomplete block designs [19], where a block is a balanced set of treatments for a specific group of variables.

The treatments in Box-Behnken include the mid-point of the process area, and the mid-point of the edges of the cube encapsulating the process area. [16] In our nine-factor experiment, Box-Behnken prescribes varying three factors across 12 blocks, with additional measurements at the midpoint to create a full RSM.

To analyze the significance and strength of the effects in the response, we generate an Analysis of Variance (ANOVA) table with the data from the experiment. In the analysis, an error rate of 0.05 is used to determine whether an effect is significant. The F-value, which is the ratio of variance between groups to the variance within groups, is used to rank the strength of the effects.

C. Factors (Independent Variables)

1) MMAS parameters

The factors listed in Table I were used in this study and are described as follows:

α and β are parameters in (1). High and low ranges were selected based on the ranges used in previous literature [11].

ρ denotes the evaporation rate in (2).

m denotes the number of ants that are randomly placed at starting nodes at the beginning of an algorithm iteration. This is also the number of tours constructed during each algorithm iteration. The upper bound of this range was limited to 50%, based on previous research indicating higher numbers of ants are not beneficial to algorithm performance [11], [14].

*q*0 refers to the exploitation/exploration ratio, discussed in Section IV.

nnAnts denotes the candidate list length, which is the maximum the maximum number of cities that will considered at each node. The lower bound of this range was set to 5% in order to allow some degree of exploration in all problem sets. The upper bound of this range was limited to 85% as previous research [11] indicates higher values are not beneficial to algorithm performance.

TABLE I: BOX-BEHNKEN FACTOR LEVELS

Variable	Description	Low (-1)	Medium (0)	High (+1)
MMAS Algorithm Parameters				
α	pheromone influence factor	1.0	7.0	13.0
β	heuristic influence factor	1.0	7.0	13.0
ρ	pheromone Evaporation Factor	0.01	0.5	0.99
m	Number of Ants as a % of n	1%	25%	50%
$q0$	Exploitation/exploration ratio	0.01	0.5	0.99
nnAnts	length of candidate list as a % of n	5%	45%	85%
Problem Characteristics				
n	Number of Nodes in Problem Instance	100	250	400
σ_η	Standard Deviation of Edge Costs	30	50	70
ε	Require Accuracy (Maximum relative error compared to best tour at 1000n iterations)	0.1%	1.55%	3%

2) *Problem characteristics*

n, the number of nodes, refers to the number of cities in a TSP instance, where there is a one-to-one correspondence between nodes and cities.

σ_η, standard deviation of edge costs, refers to the population standard deviation of all the distances between cities within a TSP. The parameter ranges for the problem characteristics were selected based on values used in previous research [12], [14], and the need to constrain the size of the problem instances to meet resource constraints.

ε, the required accuracy, is a percentage of maximum acceptable relative error, compared to the algorithm run's solution at 1000n iterations.

We assume that all algorithm runs converge before 1000n iterations. For most treatments, this assumption meets or exceeds the findings in [10], that all runs converge after 2500n tours. The only exception is those treatments where both the number of nodes (n) and the number of ants (m) are set to 100 and 1%, respectively. In these treatments, we verified convergence was reached before algorithm termination.

D. Responses

1) *Termination criteria*

We evaluated three separate termination criteria: the number of iterations; the total number of stagnations; and the number of effective iteration stagnations. We generated a response surface for each, where the basis was composed of the variable in Table I and the response variable was a count of the number of iterations, stagnations, and the number of iteration stagnations, respectively.

The first measure is a count of the number of iterations the algorithm has gone through before reaching a solution within the required accuracy.

The second measure is a count of the number of iterations that have occurred in the algorithm run without resulting in improvement. This number accumulates over the run and does not reset when an iteration results in an improvement. For example, as shown in the solid box Fig. 5, the algorithm found a best tour cost solution of 15391 at iteration 150; the number of total number of stagnations at this point was 140, the sum of all stagnations that have occurred in the algorithm run up to that point.

```
begin problem TSP-00
seed 1507313303
begin try 0
best 15646   iteration 1      tours 64      time 0.042    stagnations 0
best 15592   iteration 2      tours 126     time 0.089    stagnations 0
best 15548   iteration 6      tours 374     time 0.273    stagnations 3
best 15537   iteration 10     tours 622     time 0.457    stagnations 3
best 15511   iteration 12     tours 746     time 0.550    stagnations 1
best 15462   iteration 13     tours 808     time 0.595    stagnations 0
best 15459   iteration 94     tours 5830    time 4.314    stagnations 80
best 15450   iteration 143    tours 8868    time 6.571    stagnations 48
best 15420   iteration 148    tours 9178    time 6.805    stagnations 4
best 15391   iteration 150    tours 9302    time 6.897    stagnations 1
best 15364   iteration 251    tours 15564       time 11.566     stagnations 100
best 15294   iteration 414    tours 25670       time 19.105     stagnations 162
best 15212   iteration 504    tours 31250       time 23.264     stagnations 89
best 15194   iteration 1029   tours 63800       time 47.130     stagnations 524
best 15153   iteration 1665   tours 103232      time 75.845     stagnations 635
best 15115   iteration 15278      tours 947238      time 701.270      stagnations 13612
best 15108   iteration 92581      tours 5740024     time -72.893      stagnations
end try 0
```

Fig. 5. A sample improvement log from a MMAS run.

The third measure is frequently used as the basis for a termination criterion [11], [15]. It is the number of iterations that have passed without any improvement since the last improvement. Note that the number of iteration stagnations does not uniformly increase throughout the course of a MMAS algorithm run in general.

Because our experiment determines run-time accuracy after the fact, we must account for the non-uniform increase in iteration stagnations by recording the largest number of iteration stagnations to have occurred within the algorithm run plus one.

The rationale behind this process is to examine and model the frequently used iteration stagnation criteria in a manner that can be used for prediction. Based on the non-progressive behavior observed in Fig. 5, recording the most recent number of iteration stagnations to determine iteration stagnations would result in early termination of algorithm runs before the required accuracy was reached.

Using the example in Fig. 5, when the algorithm has found the best solution of 15391 in the solid box, the recorded number of iteration stagnations is equal to 80+1=81, because 80 is the maximum number of iteration stagnations to have occurred up to that solution. Alternately, when the algorithm has found the best solution of 15294, in the dashed box, the recorded number of iteration stagnations is equal to 162+1=163, because the most recent number of iteration stagnations is the maximum number that have occurred.

2) Convergence accuracy

We generated a fourth response surface for the relative error in the algorithm's solution at 1000n iterations, compared to the best solution found for the TSP problem instance. This response surface was generated for two reasons. The first was to provide a comparison for the tradeoffs in optimizing an algorithm's termination criteria and its convergence accuracy. The second objective was to compare the results of the Box-Behnken model to the results of previous research, which also studied relative error.

Optimally, we would have deterministically computed the global optimum for each of the problem instances and used this value as the baseline for relative error comparison. However, resource constraints made this approach prohibitive.

E. Problem Instances

Because specific problem characteristics were independent variables in this experiment, we created unique TSP problems to match the parameters in Table 1. Nine problem instances were generated using a version of the DIMACS portmgen code available from the 8th DIMACS Challenge Benchmark library at http://dimacs.rutgers.edu/Challenges/TSP/download.html. We ported the original code over to Python and modified it to accommodate the selection of custom standard deviations and means for the edge costs. All instances were generated with a mean edge cost of 200.

F. Computation

We conducted algorithm runs across seven separate, similar, but not identical machines provisioned with LINUX Ubuntu 16.10 distributions. Because algorithm run time was not considered as part of this experiment, we did not consider nuisance factors such as background processes and variations to be consequential to the results.

We modified the original code base to accommodate parallel processing using the Open MP library—allowing for parallel tour construction during each iteration. We added logging to track of the number of stagnations that occur between improvements. Finally, as part of experiment controls, we removed the branching factor reset, time limit termination, and tour limit termination processes from the code.

We performed experimental design generation and statistical calculations with Minitab 18.

G. Model Fitting

Our Box-Behnken design uses the nine factors outlined in Table I, with 10 center points per replicate, and 10 replicates, resulting in a total of 1300 ACO runs.

TABLE I: TOP FIVE RANKED EFFECTS FOR EACH OF THE THREE TERMINATION CRITERIA RESPONSE SURFACES

Iterations		**Total Stagnations**		**Effective Iteration Stagnations**	
Parameter	**F-Value**	**Parameter**	**F-Value**	**Parameter**	**F-Value**
α	1014	α	910	nnAnts	797
ρ	908	nnAnts	764	α	740
nnAnts	741	ρ	763	α^2	588
α^2	454	α^2	695	$nnAnts^2$	430
$nnAnts^2$	369	ρ^2	655	ρ	400

All four responses produced non-normal data with peaks near zero and a long, one-sided tail. ANOVA testing requires normal data to determine an effect's significance and relative strength. Therefore, we used the Box-Cox transformation method [20] to produce a normal distribution for analysis. In this methodology, response values are raised to a power λ to produce data that is a normal distribution for analysis. For each response variable, the statistical software calculated an optimal value of λ, through brute force, to produce a normal distribution for model fit. The λ values for iterations, total stagnations, iteration stagnations, and convergence accuracy were -0.279, -0.334, -0.319, and 0.242 respectively.

This research differs from previous research that used RSM with the ACO metaheuristic, in that outliers were not removed or replaced at part of the model fitting process. This decision was based in the research objective to determine the predictability of the termination criteria. Because outliers represent real data, their inclusion is necessary for an objective assessment of the models' predictive values.

VII. RESULTS

After generating the four response surfaces from data taken from MMAS algorithm runs we analyzed the first and second order effects in the models; evaluated the models' predictive fit to the data, conducted trial runs to validate model use; and conducted a sensitivity analysis to determine how broadly our findings could be applied.

A. Significant Effects

1) Termination criteria

In Table II, the top five effects, ranked by ANOVA F-Values, are shown. Each of the surface models shows the most significant factors to be α, ρ, and nnAnts in each case. Figs. 7-9 shows the specific impact of each of these first order effects to be generally the same for all of the termination criteria, with an increase in any of these parameters resulting in earlier optimal termination criteria.

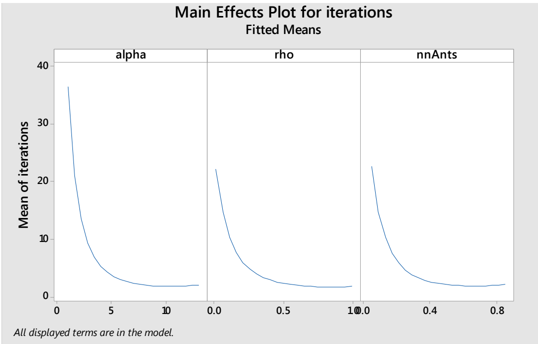

Fig. 6. Main first order effects for iterations response.

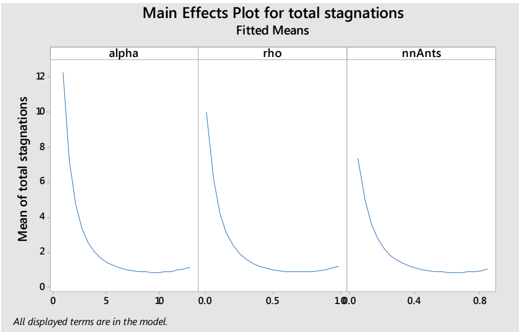

Fig. 7. Main first order effects for total stagnations response.

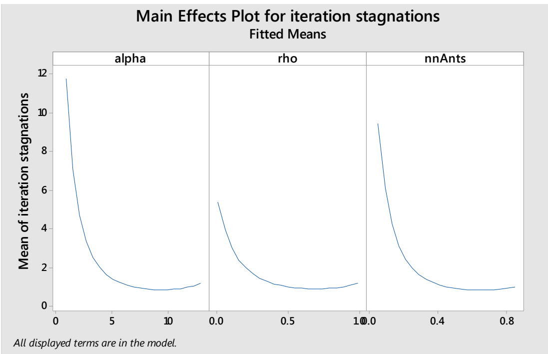

Fig. 8. Main first order effects for iteration stagnations.

Comparing these factors with the plots of the convergence accuracy model in Fig. 10, we see there is a direct tradeoff in modifying the α, ρ, and nnAnts parameters between early algorithm accuracy and convergence accuracy. By increasing α, ρ and nnAnts the algorithm reaches the required accuracy, relative to its final solution at convergence, much earlier in the run. However, these same parameter modifications significantly deteriorate the algorithm's solution at convergence.

Second order effects had a much weaker impact on the response surfaces, accounting for approximately 15% of the variation in each model. The most significant second order effect for the termination criteria was ρ*nnAnts, which had an F-value of 247.2 in the iterations model. This effect is shown in Fig. 6.

While only the top five important factors are discussed here, 35 out of 55 coefficients were shown to be statistically significant in the termination criteria response models. This high number of significant factors indicates the screening experiments, as performed in previous literature [14], should generally be avoided.

An additional noteworthy finding is the factors that did not have a strong impact. The number of ants, m, did not have a significant effect on the optimal termination criteria. This indicates that, because the number of tours constructed is directly proportional to the number of ants, the use of tours is a poor criterion for algorithm termination.

On the other hand, the number of nodes is a significant factor in the model, which indicates that using multiples of the problem size for algorithm termination limit, such as 1000n, can maintain algorithm performance across a range of problem sets.

2) Convergence accuracy

The most significant effects in the model were β, q0, the interaction of β*$q0$, and nnAnts. These findings align well with previous literature [11] [14], validating the use of the Box-Behnken RSM for ACO modeling.

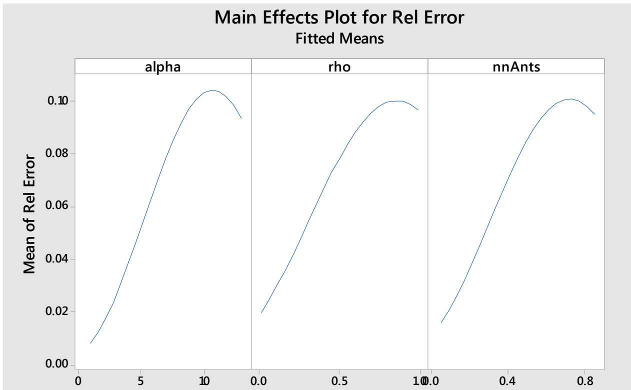

Fig. 9. α, ρ, and nnAnts first order effects for convergence accuracy model.

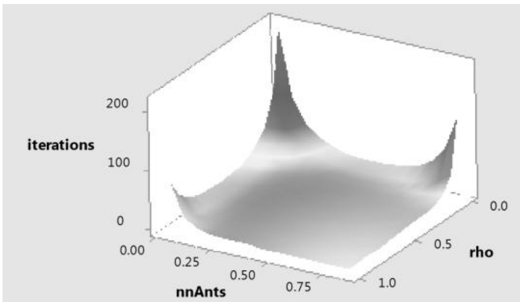

Fig. 10. ρ*nnAnts effect plot for the iterations response.

B. Predictive Measures

We evaluated each of the three models for fit by calculating the $R^2_{\text{Prediction}}$ metric, which is an estimation of the amount of variability the model is able to describe in new observations. $R^2_{\text{Prediction}}$ is calculated using (6) with the prediction sum of squares (PRESS), given by (5) [15]. A higher value of $R^2_{\text{Prediction}}$, indicates a more predictive model. Table III shows each response surface had a $R^2_{\text{Prediction}}$ value above 75%, with the iterations and total stagnations termination criteria closer to 80%, indicating a slightly higher predictive value.

Regarding our original objectives, these results indicate that termination criteria can be reliably modeled, and that the number of iterations is the preferred termination criterion, in terms of predictability.

$$\boldsymbol{PRESS} = \sum_{i=1}^{n}[\boldsymbol{y_i} - \acute{y}_{(i)}]^2 \quad (5)$$

$$R^2_{\text{Predictio } n} = \frac{PRESS}{SS_T} \quad (6)$$

TABLE III: $R^2_{\text{PREDICTION}}$ VALUES FOR THE RESPONSE SURFACES GENERATED FOR EACH OF THE RESPONSES STUDIED

Termination Criteria	$R^2_{\text{Prediction}}$
Iterations	80.22%
Total stagnations	79.56%
Iteration stagnations	75.65%

C. Model Validation

Models were validated with the objective that response surfaces provide termination criteria for a given set of parameters with a 95% degree of confidence that the algorithm is within the accuracy band.

The models generated were validated using three new problem instances and MMAS parameters shown in Table IV. Each new instance was run five times, to a stopping point of 1000n iterations. The model's 95% confidence, upper bound stopping criteria was then compared to the actual optimal termination criteria in Table IV.

All three types of termination criteria used in the validation run produced results which met the required accuracy. The actual optimal stopping criterion for each ACO run was well below the model's upper bound.

TABLE IV: VALIDATION PROBLEM PARAMETERS AND TERMINATION CRITERIA PREDICTED BY RESPONSE SURFACES

Factor	VProb1	VProb2	VProb3
α	1	2	4
β	12	3	4
n	150	300	200
σ_η	40	60	50
ρ	0.02	0.5	0.98
m	10% (15)	5% (15)	40%(80)
$q0$	0.8	0.3	0.1
nnAnts	40%	10%	60%
ε	2%	0.5%	1%
Modeled iterations	41	745	5
Actual iterations (average)	6.8	8.8	2
Modeled total stagnations	72	263	2
Actual number of total stagnations (average)	4.6	1.4	1
Modeled iteration stagnations	165	6211	3
Actual iteration stagnations (average)	3.8	1.6	1

D. Sensitivity Analysis

We performed a sensitivity analysis to determine how broadly the response surfaces could be applied to larger ranges of parameters.

Due to resource constraints, only the required accuracy (ε) was adjusted. First the response surfaces were rebuilt with ε levels of 1%, 3%, and 5%. The same observations from the initial experiment were used.

The resulting termination criteria response surfaces were significantly different. The first order effects α and ρ maintained a similar impact as before. However, nnAnts exhibited significantly decreased influence, as seen in a comparison of Fig. 11 to Fig. 12. The new model also had higher relative F-value rankings for q0 and ε, indicating a reordering of the strength of first order effects. Additionally, the impact of second order effects became more prominent, accounting for approximately 20% of the model's variation. The shape of the second order interaction also changed, as seen in comparing Fig. 12 with Fig. 6.

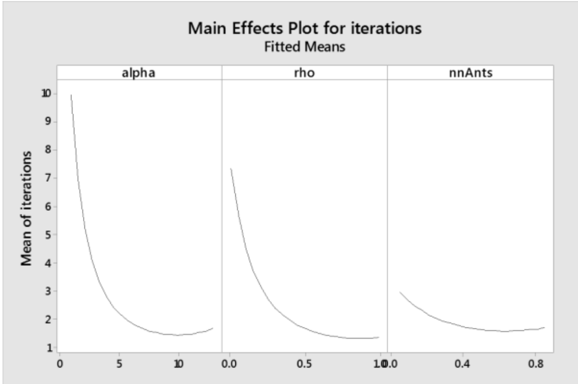

Fig. 11. First order effects plot for iterations response in sensitivity analysis.

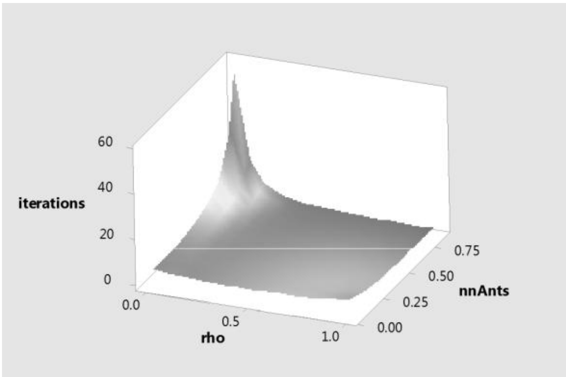

Fig. 12. ρ*nnAnts response surface for iterations response in sensitivity analysis.

By achieving significant changes in the termination criteria in the responses through slight modifications in ε, we found that there is a high degree of specificity in the response surfaces for a given set of problem characteristics. Broad conclusions should not be drawn between a response surface, its effects and a differing data set. This finding aligns well with previous research that has concluded that optimal ACO parameter settings are highly problem specific [15].

VIII. CONCLUSION

A. Discussion

Our study finds that termination criteria, for a given set of parameters, problem characteristics, and required accuracy, can be predicted using response surface methodology. While all types of termination criteria yielded predictive models, the iterations criterion yielded the best results across all bands of ε.

Finally, our sensitivity analysis shows the effect of each factor changes with even small alterations to the process area. This indicates that generalizations on parameter impact are a poor substitute for a model fitted to specific problem characteristics.

B. Recommendations

Our finding that an early termination criterion can be predicted for a required accuracy provides an alternative to attempting to balance two competing measures, such as time and accuracy, in a single response. Instead, a set of observations can be used to create multiple response surfaces to model a metaheuristic's behavior at both early termination and convergence. These can then be iteratively analyzed to meet specific accuracy requirements.

With regard to selecting a type of termination criteria, the number of iterations termination criterion maintained the highest $R^2_{\text{Prediction}}$ across the studies and is the conceptually easiest to employ. Therefore, we recommend iterations as the preferred method for terminating an ACO run, unless an application has a specific need to choose an alternate method. We also identified Box-Behnken method to be useful in developing predictive models for the ACO algorithm. It is recommended in cases where the algorithm is expected to run in a region of normally distributed problem characteristics due to the lower number of required observations and its rotatability.

C. Future Work

This study did not examine the modeling of time as termination criteria, as it is not reproducible without identical equipment and configurations. However, modeling the time termination criteria may be useful, particularly when ACO is considered for use in fielded system.

We also limited the scope of this study to the symmetrical TSP in order to build on previous research. Validating RSM for parameter and termination criteria tuning in other problem areas would be a useful examination of the breadth of the method's applicability.

Finally, this and previous research indicated that the ACO parameter optimums switch between high and low extremes at some yet undefined threshold. A study examining decision trees and stochastic machine learning algorithms may provide insight into this behavior.

REFERENCES

[1] J. L. Fernandez-Marquez, G. D. Serugendo, S. Montaga, M Viroli, and J. L. Arcos, "Description and composition of bio-inspired design patterns: a complete overview," *Natural Computing*, pp. 43-67, 2013.

[2] F. Zambonelli *et al.*, "Developing pervasive multi-agent systems with nature-inspired coordination," *Pervasive and Mobile Computing*, pp. 236–252, 2015.

[3] M. Dorigo and T. Stutzle, "The ant colony optimization metaheuristic: Algorithms, applications, and advances," *Metaheuristcs Handbook*, pp. 251-285, 2003.

[4] D. Weyns, A. Omicini, and J. Odell, "Environment as a first class abstraction in multiagent systems," *Autonomous Agent Multi-Agent Systems*, pp. 5-30, 2007.

[5] H. M. Maier *et al*., "Ant colony optimization for design of water distribution systems," *Journal of Water Resources Management and Planning*, pp. 200-209, 2003

[6] J. N. Haack, G. A. Fink, W. M. Maiden, S. J. Templeton, and E. W. Pulp, "Ant-based cyber security," in *Proc. the 8th International Conference on Information Technology New Generations*, 2011, pp. 918-926.

[7] J. M. Szemis, G. C. Dandy, and H. R. Maier, "A multiobjective ant colony optimization approach for scheduling environmental flow management alternatives with application to the River Murray, Australia," *Water Resources Research*, pp. 6393-6411, 2013

[8] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation Politecnico di Milano, Italy, 1992.

[9] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 53-66, 1997.

[10] T. Stutzle and H. H. Hoos, "Max-min ant system," *Future Generation Computer Systems*, pp. 889-914, 2000.

[11] E. Ridge and D. Kudenko, "Tuning the performance of MMAS heuristic," *Engineering Stochastic Local Search Algorithms Designing, Implementing and Analyzing Effective Heuristics*,. Berlin Heidelberg : Springer-Verlag, pp. 46-60, 2007.

[12] E. Ridge and D. Kudenko, "Determining whether a problem characteristic affects heuristic performance: A rigorous design of experiments approach," *Recent Advances in Evolutionary Computation for Combinatorial Optimization*,Springer, pp. 21-35, 2008.

[13] Stutzle, Thoms *et al*., "Parameter adaptation in ant colony optimization," *Autonomous Search*, pp. 195-211, 2011.

[14] M. Fallahi, S. Amiri, and Y. Masoud, "A parameter tuning methodology for metaheuristics based on design of experiments," *International Journal of Engineering and Technology sciences (IJETS)*, pp. 497-521, 2014.

[15] A. Byerly and A. Uskov, "A novel approach to avoiding early stagnation in Ant Colony Optimization algorithms," *International Journal of Knowledge-based and Intelligent Engineering Systems*, pp. 113-121, 2016.

[16] National Institute of Standards and Technology. Comparisons of response surface designs. *Engineering Statistics Handbook*. [Online] Available: http://www.itl.nist.gov/div898/handbook/pri/section3/pri3363.htm

[17] D. C. Montgomery, *Design and Analysis of Experiments.* Hoboken, NJ: John Wiley & Sons, pp. 470-472, 478-554, 2013

[18] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimaization by a colony of cooperating agents," *IEEE Transactions on System, Man, and Cybernetics, Part B Cybernetics*, vol. 26, 1996.

[19] G. E. P. Box and D. W. Behnken, "Some new three level designs for the study of quantitative variables," *Technometrics*, vol. 2, no. 4, pp 455-475, 1960.

[20] G. E. P. Box and D. R. Cox, "An analysis of transmorfations," *Journal of the Royal Statistical Society*, Series B, vol. 2, pp. 211-252, 1964.

Chase W. Gruszewski holds a masters of science in systems engineering from Johns Hopkins University, in Baltimore Maryland and a bachelors of science ocean engineering from the United States Naval Academy in Annapolis Maryland.

He currently works as a programmer at CWPDEV in Washington State.

Matthew H. Henry holds a doctor of philosophy in systems and information engineering from the University of Virginia.

He is a lecturer with the Johns Hopkins University Whiting School of Engineering. He has over twenty years in research and development in the aerospace, automotive, and cyber domains. He has published dozens of papers in peer-reviewed journals and conferences and has written several book chapters. His current work focuses on new algorithmic approaches to autonomy and complex adaptive systems.