Research Journal of Applied Sciences, Engineering and Technology 11(1): 109-117, 2015
DOI: 10.19026/rjaset.11.1682
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: March 7, 2015 Accepted: April 1, 2015 Published: September 05, 2015

Research Article

Nature-inspired Parameter Controllers for ACO-based Reactive Search

[1]Rafid Sagban, [2]Ku Ruhana Ku-Mahamud and [2]Muhamad Shahbani Abu Bakar
[1]Department of Computer Science, University of Babylon, Babylon, Iraq
[2]School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

Abstract: This study proposes machine learning strategies to control the parameter adaptation in ant colony optimization algorithm, the prominent swarm intelligence metaheuristic. The sensitivity to parameters' selection is one of the main limitations within the swarm intelligence algorithms when solving combinatorial problems. These parameters are often tuned manually by algorithm experts to a set that seems to work well for the problem under study, a standard set from the literature or using off-line parameter tuning procedures. In the present study, the parameter search process is integrated within the running of the ant colony optimization without incurring an undue computational overhead. The proposed strategies were based on a novel nature-inspired idea. The results for the travelling salesman and quadratic assignment problems revealed that the use of the augmented strategies generally performs well against other parameter adaptation methods.

Keywords: Ant colony optimization, combinatorial problems, parameter control, swarm intelligence metaheuristics

INTRODUCTION

As combinatorial optimization solvers, Ant Colony Optimization (ACO) algorithms are accorded much work to improve their efficiency and usability. Parameter adaptation and reactive search are autonomous search techniques for improving the behavior of the algorithms by integrating control in the solving process (Broderick *et al*., 2014). The control takes either a machine learning fashion, as in reactive search, or the self-tuning fashion, as in parameter adaptation. In ACO, both fashions respond to a similar need, which is the balance between two opposing processes, namely, exploration and exploitation (Solnon, 2010). They have to be maintained during the search. These radical methods are used for improving the flexibility of ACO algorithm by an "on-the-fly" replacing the bad-performance.

The performance of ACO algorithm relates to how efficient it explores the search space, while also being able to exploit the most promising regions. In reactive search techniques, the searching process integrates machine learning with searching process. Integration with ACO requires memory-based models to record the past history of search and escape mechanism by restarting the search from a new random point when it shows no improvement in the quality of solutions (Khichane *et al*., 2009; Sagban *et al*., 2014). However, more efforts still need to be directed to the use of machine learning triggers to detect the current state of search whether it is exploration or exploitation.

Parameter adaptation techniques modify the parameters during the run in different manners: deterministic, adaptive and self-adaptive (Stützle *et al*., 2012). The first manner assumes that all Combinatorial Optimization (CO) problems are the same in their global characteristics and this is not true. The second manner adapts to the local characteristics of the regions of the search space through a feedback. It has the advantage of no augmenting in the complexity of the problem. On the other hand, it has two limitations: a complexity of implementation and presenting new hyper-parameters which also need to be tuned. The third manner has the advantage of tuning parameters "for free", where its implementation is simple and there is no hyper-parameters that need to be tuned. Besides increasing the complexity of the problem, it is linked to the structure of the algorithm. There is an ongoing interest in parameter adaptation strategies in ACO literature (Stützle *et al*., 2012). Although these strategies were invented early in ACO history, their applications are ineffective compared with those proposed for other meta heuristics such as genetic algorithms.

This study proposes three strategies based on the type of exploration measures involved to monitor the progress of the search. The first strategy utilizes the improvement in the quality of solutions as implicit proxy for current exploration behavior. The second strategy exchanges the proxy with explicit strategy for exploration indication called ACOustic (Sagban *et al*., 2015), while the third strategy hybridized both

Corresponding Author: Rafid Sagban, Department of Computer Science, University of Babylon, Babylon, Iraq

indications. Max-Min Ant System (MMAS) (Stützle and Hoos, 2000) is used as a test-bed to evaluate the performance of the proposed strategies. Two classical combinatorial optimization problems are used in the evaluation. These are Travelling Salesman Problem (TSP) and Quadratic Assignment Problem (QAP).

ANT COLONY OPTIMIZATION

ACO takes inspiration from the foraging behavior of several ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization exploits a similar mechanism in solving optimization problems. The ACO metaheuristic is based on a generic problem representation and the definition of the ants' behavior (Dorigo and Stützle, 2010). Given this formulation, the ants in ACO build solutions to the CO problem by moving concurrently and asynchronously on a predefined construction graph. Considering the *TSP* problem, there are some aspects that need to be characterized.

A finite set of components of the problem: $C = \{c_1, c_2, \ldots, c_n\}$, where n is the number of components of the TSP problem:

- A sequence of the states of the problem over the elements of C, such that each sequence $S = <c_i, c_j, \ldots, c_h, \ldots>$. The set of all sequences is denoted by S.
- A set of candidate solutions S^* is a subset of S.
- A set of feasible solutions N is a subset of S.
- A non-empty set of optimal solutions.
- A cost g (s, t) is associated with each candidate solution.

TSP (S, f, Ω) has to be mapped to a complete connected graph called construction graph $CG_{TSP} = (C, L)$, where C is the set of nodes of the graph and L is the connections of those nodes. The artificial ants will walk randomly on CG_{TSP} to build solutions of the TSP problem. The pheromone trail value τ and heuristic value can be associated with C or L.

A finite set C of solution components need to be derived. This set C is used to assemble solutions for the CO problem. Next, a set of pheromone values τ is defined. The set τ is called the pheromone model and is commonly recognized as a parameterized probabilistic model. The pheromone model is probabilistically used to generate solutions based on the solution components. To achieve this, the model associates the solution components to the pheromone values $\tau_i \in \tau$ which formed the central components of the ACO metaheuristic. In general, the ACO approach attempts to solve an optimization problem by iterating the following two steps (Dorigo and Stützle, 2010):

- Constructing candidate solutions by using the pheromone model. The pheromone model, as mentioned earlier, is a parameterized probability distribution over the solution space.
- Modifying the pheromone values by using candidate solutions in a way that it is deemed to bias future sampling towards high quality solutions.

The interaction between the two steps is presented in Fig. 1 which illustrates a conceptual abstraction about how ACO metaheuristic solves CO problems. This is an informal high-level description about this functionality. The ACO metaheuristic defines the way of the solution construction and the pheromone update. These are used to implement problem specific or centralized actions that cannot be performed by a single ant.

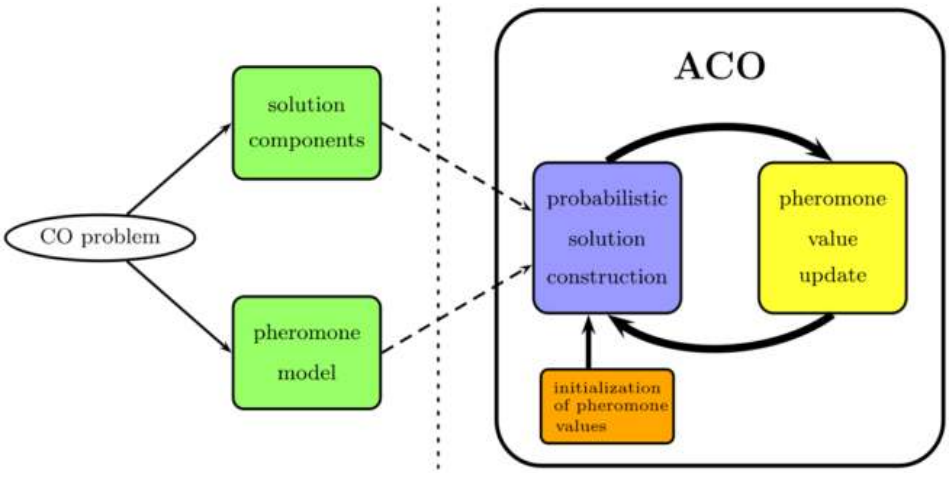

Fig. 1: The conceptual framework of ACO

PARAMETERS ADAPTATION IN ACO

Parameter control methods work directly on the values of the parameters while solving the combinatorial problem. The schemes of control are governed by the type of information used as evidence for changing parameters' values. This section reviews the recent work in the field of parameter adaptation in ACO.

In deterministic parameter adaptation, the information is not predicted during the run and a set of deterministic rules is allowed to trigger the change. In this way, there is no feedback from the search that needs to be collected. This kind of adaptation is useful for improving the anytime behavior of ACO (Lopez-Ibanez and Stützle, 2014). Stützle *et al*. (2010) examined two deterministic MMAS variants. In the first one, the parameter of the number of ants (m) starts with (1) and then increases by (1) every (10) iterations until the value becomes equal to the number of variables (n). In the second variant, the parameter of exploration/exploitation, denoted by q_0, decreases starting from 0.99 until 0.0. Both variants showed good results in the context of anytime behavior and quality of solutions. The same strategy was followed by Liu *et al*. (2011a) by considering more parameters. This approach can be regarded as an offline schema for adapting parameters. The value of parameters is adapted based on a pre-scheduled number of iterations. The problem with such deterministic assignment of parameter values is that the number of iterations needed for convergence is unknown. In fact, devising proper values for parameters must be adapted based on the search progress.

In adaptive approach, parameters are modified according to either the diversity of the pheromone trails or the quality of solutions. Searching behavior of the algorithm is considered. Neyoy *et al*. (2013) implemented fuzzy logic controller to adjust the pheromone concentrate parameter, denoted by α. The rule of adaptation relied on λ-branching factor as an exploration indicator. Various fuzzy systems were proposed to control the diversity of solutions in order to maintain the exploration and exploitation in ACO (Olivas *et al*., 2014). Collings and Kim (2014) augmented the use of fuzzy controller for the adaptation base, stagnation detection and control. Two fuzzy controllers were proposed by Liu *et al*. (2011b) to adjust the parameters of the number of ants and the evaporation rate dynamically. Wang (2013) used the proxy of entropy information as an exploration indicator for adaptaion.

The parameter settings can be encoded with the search space, in which explicitly results the search-adaptive approach, or implicitly results the self-adaptive approach. In the first category, the work of Melo *et al*. (2010) considered the ACS parameters: α, β, ρ and q_0 in their adaptation strategy mutation operator for exchanging the best setting with the worst one. The proposed mechanism has contributed a new measurement tool to indicate disturbance of parameters and then each parameter to be disturbed will be substituted by the best one in the best colony. In the second category, the most state-of-the-art methods (Pellegrini *et al*., 2012) exist as exemplified by Khichane *et al*. (2009) and Randall (2004). The rationale way for the adaptation is by evolving parameters based on an extra pheromone matrix maintained solely for this purpose known by parameters matrix. The self-adaptive methods played a key role in solving several CO problems. However, they did not improve the performance of ACO in TSP and QAP (Pellegrini *et al*., 2012), except the work of Randall (2004) which has shown explicitlygood results for TSP and QAP. To fill the gap, there is an emphasis on adoptingsuccessful methodologies such as those in evolutionary metaheuristics.

THE PROPOSED CONTROLLERS

This section is about the proposal of ACO-based Adaptive Parameters' Selection method (APS_{ACO}) in addressing the parameter setting problem. Two components are needed to address the problem, namely the parameters' selection and the reward assignment. In Fig. 2, APS_{ACO} will determine which of the parameters' values can be applied for current iteration/sec.

Nature-inspired strategies proposed to operate APS_{ACO}, include the Exploration-based Reward Assignment (ERA), Quality-based Reward Assignment (QRA) and Unified Reward Assignment (URA). The parameters' selection selects a parameter according to its empirical quality during the last iterations. The selected parameter is applied and its impact is transformed into a reward, using reward assignment, to be used in updating the quality of parameters. The reinforcement learning process in APS_{ACO} is guided by those two components.

In parameters' selection, the desirability of selecting the given parameter value V is affected proportionally by its empirical quality $Q = \{q(v_{11}), q(v_{12}), \ldots, q(v_{1m_1}), \ldots, q(v_{km_k})\}$ where k is the number of parameters and m is the number of values for the k^{th} parameter. Each parameter is associated with a range of values. The bounds of the ranges are set based on Dorigo and Stutzle (2010). The j^{th} parameter value for the i^{th} parameter, i.e., the value v_{ij}, is selected as follows:

$$s(v_{ij}) = l_i + \frac{p(v_{ij})}{m}(u_i - l_i) \quad (1)$$

where, $1 \leq i \leq k$ *and* $1 \leq j \leq m$:

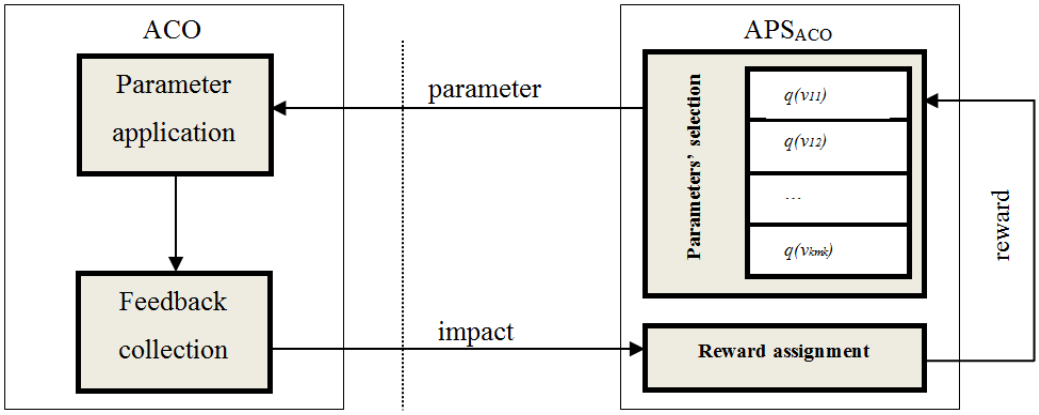

Fig. 2: The general scheme ACO-based adaptive parameters' selection

$$p\ (v_{ij}) = \ q\ (v_{ij})/\sum_{l=0}^{n} q\ (v_{il}) \qquad (2)$$

where, l_i is the lower bound value of i^{th} parameter, u_i is the upper bound value of i^{th} parameter, m is the number of values and p is the proportional selection probability for value the v_{ij}. In the first application, the value of each parameter is chosen as the halfway point in its range. After that, the values are selected proportionally. If the application performs badly, then the desirability of application of the parameter has to be decreased, otherwise it will be rewarded. This is by firstly evaluating the effect of the selected values and then updating the empirical quality of the selected values.

In reward assignment, the empirical quality of each parameter gain rewards only if the application of the parameter achieves impact for the optimization process. The impact determined by feedback collection strategy is translated into rewards denoted as $r\ (v_{ij})$. The reward will be added to the previous quality of the perspective value as follows:

$$q\ (v_{ij}) = (1 - \ \rho)\ .q\ (v_{ij}) + \ \rho\ .r\ (v_{ij}) \qquad (3)$$

The value of ρ is automatically assigned within the recommended range. Based on this formula, there is another instance of exploration versus exploitation: the best values are extensively used, while other values which need to be tried from time to time are not considered yet. To find good trade-off of the two processes, the bounding strategy is involved where the quantity τ_{min} represents the minimum selection probability for all the parameters' values:

$$q\ (v_{ij}) = (1 - \ \gamma)\ .q\ (v_{ij}) + \ \gamma\ .\tau_{\min} \qquad (4)$$

where, the value of γ is automatically assigned by the proposed APS_{ACO} method itself. Through this component, the effect of parameter value choices on the search is transformed into rewards. It involves the exploration state, the quality of solutions or both for rewards' calculation, i.e., the proposals of ERA, QRA and URA strategies. The idea of the proposed controllers is inspired from the acoustical mimicry to the sounds of ant queens by some parasites to control the food foraging behavior of some insects. For the mimicry to be succeeded, the parasites use a kind of feedback collection process. The nature-inspired feedback collector proposed by Sagban *et al*. (2015) to characterize the exploration in the ACO algorithm. Its idea presents in the following subsections.

The biological schema: Rapid and effective communication between ants is a key attribute that enables them to live in dominant, fiercely protected societies. *Myrmica* ant colonies, in particular, are exploited by social parasites called *Maculinea* butterflies (Barbero *et al*., 2012). The process of *Trophallaxis* (i.e., distributing liquid food from the 'social stomach') between attendance workers and other nest-mates is the main process in food foraging behavior of ants. The worker ants produce acoustics during the process. The *Maculinea* larvae interfere with the *Myrmica* system and produce similar acoustics to that of the colony. The high number of worker ants leads to a low relatedness between nest-mates. A greater variance in nest-mates ACOustic signals leads to a higher likelihood of being infested (Barbero *et al*., 2009). Through this indicator, the larva can decide the optimal point to leave the colony before it is discovered by other ants. The larva is able to evaluate the situation inside the nest whether to leave or stay. If the relatedness between nest-mates becomes high, then the likelihood of being clustered around the larva will become low. This is an indication to the larva to explore another nest before being killed; otherwise the larva will continue to exploit the current nest until further notice. The ACOustic reaction in this process can be simplified in three basic components as shown in Fig. 3.

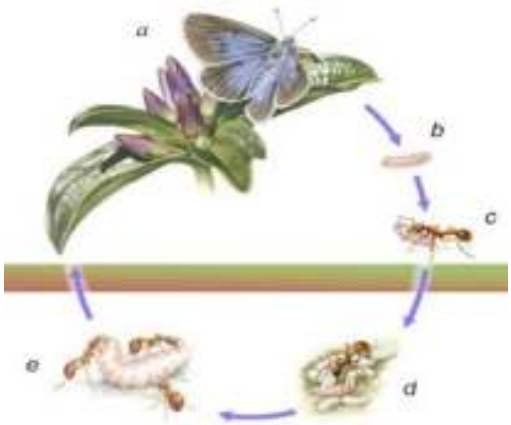

Fig. 3: The *Myrmica* ants-*Maculinea* larvae system

A colony of artificial ants is inspiring its characteristics from the real ant's foraging behavior. The construction graph simulates the environment that ants and larvae agents are moving on. For larvae agents, the interaction with the new environment is highly related with state of penetration, i.e., learning process. The agents can decide whether to continue with the current exploitation or to explore another environment. To simulate the process of characterizing the state of penetration, statistical analyzing and the agglomerative clustering algorithms are developed in this study.

The computational schema: The way of characterizing the state of penetration is used as a didactic tool to explain the idea behind the ACOustic proposal. The behavior of ACO algorithm is described in terms of exploration and exploitation processes. According to the biological scheme, the natural scheme in parasites-ants system models is then translated into procedures for feedback collection during the problem solving.

For the modelling, let a construction graph $G = (N, A)$ represent a CO problem, where N is the set of nodes; A is the set of arcs; $|A| = a$ and $|N| = n$. The fitness landscape of the given CO problem is defined by: P is a population set which includes all solutions to the CO problem where each solution $s \in P$ is assigned a fitness value $f(s)$; and has a structure of neighborhood $N \subseteq P \times P$. A colony of artificial ants performs a biased walk in this landscape with the goal of finding low $f(s)$ (in case of minimization problems). The set $C_p(t)$ represents the collection of acoustics (sounds) that emanate from the landscape traversed by the ants of a perspective colony at time t where $C_p(t) \subseteq P(t) \times P(t)$ where c_i and c_{i+1} are two acoustics which belong to $C_p(t)$ where $c_i = \{x_1, x_2, \ldots, x_a\}$; $c_{i+1} = \{y_1, y_2, \ldots, y_a\}$ and where the long of each ACOustic signal is equal to a. The relatedness between two nest-mates is defined by the similarity between their acoustics. Two acoustics c_{i+1} and c_i are considered as similar if their similarity neighborhood SN is below a predefined threshold X. For the implementation, the section walks-through the implementation of Algorithm 1.

Algorithm 1: *ACOustic* ()
Input: X
Define: $C = \{C_1, C_2, .., C_{|a|}\} = \{\{c_1\}, \{c_2\},.., \{c_{|a|}\}\}$
miniDist = $\min_{c_h, c_k \in C} d_{c_h, c_k}$
for each $C_h, C_k \in C$ do

$$D = d_{c_h, c_k} = \sqrt{\sum_{i=1}^{a} (x_i^h - x_i^k)^2}$$

end-foreach
repeat
$C_h = C_k \cup C_h$
$C = C \setminus \{C_k\}$
foreach $C_w \in C \setminus \{C_h\}$ do

$$d_{c_h, c_w} = d_{c_w, c_h} = \min \{d_{c_h, c_w}, d_{c_k, c_w}\}$$

end-foreach
miniDist = $\min_{c_h, c_k \in C} d_{c_h, c_k}$
nc = | C |
until (miniDist≤X)
rltdnss←mr-C_{Num}/stdr
return *rltdnss*
end-algorithm

In Algorithm 1, the minimum distance miniDist is calculated from the distance matrix that is generated earlier. The nearest two clusters are united, the distance matrix is recalculated and finally miniDist and number of clusters C_{Num} are updated as in Algorithm 3. The nearest neighborhood threshold X is entered, vector of acoustics clusters C_i is defined and other variables such as miniDist, C_{Num}, $NC_{throushold}$ and *max* are initialized. A matrix of Euclidian distances between acoustics is generated and the statistical medians are calculated as follows:

$$mr = \left(\sum_{i=1}^{max-1} \sum_{j=i+1}^{max} (D_{i,j}/m)\right) / ((max^2 - max)/2) \quad (5)$$

$$vr = \sum_{i=1}^{max-1} \sum_{j=i+1}^{max} (C_{ij} - mr)^2 \quad (6)$$

$$stdr = \sqrt{\frac{vr}{max-1}} \quad (7)$$

The number of clusters are combined and returned as a relatedness quantifier denoted as *rltdnss*.

The implementation of APS_{ACO} controllers: The contributed controller emerged to the body of ACO algorithmic framework as shown in Algorithm 2.

Algorithm 2: APS_{ACO}

Set the number of parameters to k
Set the number of values to m

Set the maximum and minimum ranges of parameters' values
Discretize the ranges R based on value of m
for i = 1 to k do
    $v_i \leftarrow r_i$
d←m/2
for i = 1 to k do
for j = 1 to m do
        $q_{ij} \leftarrow \tau_{min}$//It can be set to τ_0 if another ACO variant is applied except the MMAS
        $s_{ij} \leftarrow v_{id}$
while (not termination_condition ()) do
        select_param ()
        construct_solutions ()
        update_ pheromone ()
        assign_rewards ()
end-while
end-algorithm

In initialization, the probability and the quality vectors are initiated. In select_param, the parameters' values are either selected as the halfway point in their ranges, if that is the first application, or selected proportionally to be involved in the search. The ants construct their solutions and update the memory of the pheromone. The effect of the just applied parameters' values is transformed into rewards by the assign_rewards. It is based on the feedback collected from the search and it updates the quality of the current parameters' values. The amount of rewards assigned depends on the way of feedback collected, whether it focuses on the improvement in quality, the improvement in exploration behavior or the relative improvement in both of them.

In ERA and URA strategies, the exploration identified in terms of the so-called relatedness (rltdnss) amount between ants produced by ACOustic indicator, while it identified based on the quality of solutions in QRA.

Exploration-based strategy: The ERA strategy, the exploration identified in terms of the relatedness amount between ants produced by ACOustic as follow:

$$f(rltdnss) = \begin{cases} expl r = expl r + 1 \; if rltdnss > X_{rltdnss} \\ expl = expl + 1 \; otherwise \end{cases} \quad (8)$$

where, the value of $X_{rltdnss}$ is the first relatedness value captured when the number of clusters decreased. It is worth mentioning that this characterization function is deactivated during the stagnation of the search. The stagnation is flagged when the solutions S_k since the last best restart i_{last} did not improve for the last ϵ iterations (e.g., 250 iterations), i.e., if $f(S_k) \geq f(S_{gb})$ and $i\text{-}i_{last} > \epsilon$. The rewards amount is derived from the impact of the application of parameter values which is calculated as follows:

$$r(v_{ij}) = \frac{expl r^2}{expl} \quad (9)$$

It is worth mentioning that the values of the exploration/exploitation quantifiers, i.e., *explr* and *expl*, are very sensitive to the value of the nearest neighborhood threshold. The higher the threshold is, the more sensitive the quantifying becomes. In the beginning of the search, the amount of exploration starts higher than the exploitation one. With this property, the behavior of the algorithm is automated in various phases of the search. This automates the trade-off between exploration and exploitation in response to the current state of the search.

Unified reward assignment: The URA strategy relies on the improvement on both the quality of solutions and the exploration of the current search in assessing the effect of the current parameters' values. The rewards are calculated as follows:

$$r(v_{ij}) = C_{num}/global\_avg \quad (10)$$

where, the C_{num} is the number clusters calculated in Algorithm 1. Based on this equation, the exploration behavior plays a fundamental role in determining the amount of rewards. The higher the number of clusters is, the higher the reward becomes.

Quality-based reward assignment: This QRA strategy relies on the improvement in the quality of solutions in assessing the effect of the current parameters' values. The median of the objective functions of the current population is used as an effect's proxy for the application of selected parameters' values. The value of rewards is calculated as follows:

$$r(v_{ij}) = {}^{1}/_{global\_avg} \quad (11)$$

The value of global_avg is the median of the objective function for the solutions that is recorded in the population-based memory.

Experimental design: The goal of the empirical analysis is to evaluate the proposed APS_{ACO} against the state-of-the-art adaptation methods proposed for ant colony optimization. The implementation of four self-adaption methods is based on the work of two groups. The works of Randall (2004) and Förster *et al*. (2007) are in the first group, while the works of Martens *et al*. (2007) and Khichane *et al*. (2009) are in the second group. To capture the contribution of the groups independently of the problems or the algorithms for which they were proposed, the works of Randall (2004) and Khichane *et al*. (2009). are followed. In the first group, namely RandallG, the parameters' values are on-

Table 1: The TSP and QAP instances and their size, in terms of the number of cities and facilities/locations respectively

TSP			QAP		
Name	Size	Best-known cost	Name	Size	Best-known cost
eil51	51	426	nug15	15	1150
st70	70	675	nug20	20	2570
eil76	76	538	tai25a	25	1167256
gr96	96	55209	tai35a	35	2422002
rd100	100	7910	ste36a	36	9526
bier127	127	118282	tho40	40	240516
d198	198	15780	sko49	49	23386

line selected based on the Randall way, where the parameters are independent, e.g., the parameters β, ρ, γ and q_0. In the second group, namely Khichane *G*, the parameters' values are on-line selected based on the Khichane way where the parameters are interdependent, e.g., the parameters α and β. The search space for the parameter values must be known in advance and discretized in both groups except in the second group where the values are optimized a priori. Both groups are in the same level at which they manage parameters. The rewards given to the parameters' values selected during the run are based on the best-so-far ant in colony-level rather than ant-level. The ant-level setting is omitted because most the parameters are colony-wise so that they cannot be adapt multiple settings in each iteration. The number of parameter values *m* remains constant at 20. The ranges for the parameters q_0, ρ and γ are bound between the constant values of 0 and 1; parameter β is bound between the constant values of 5 and 1 and parameter α is bound between 1 and 2.

The experiments are conducted on Windows 8 64-bit operating system, processor Intel Core i3-3217U with CPU @ 1.80 GHz, RAM 4 GB. Each experiment is executed 10 times to avoid the stochastic behavior. A maximum of 10 sec is used as a termination condition to the run of particular algorithms. The QAP and TSP instances are selected from QAPLIB and TSPLIB repositories as in Table 1.

RESULTS

Non-parametric descriptive statistics are used to report the results. This is because the distribution of results is highly non-normal. The cost results are reported as the Relative Percentage Deviation (RPD) from the best known solution cost. This is calculated as follows: RPD = (((the result cost - the best known cost)) / (the best known cost)) ×100. Note that "min", "med" and "max" represent the minimum, median and maximum RPD respectively. The runtime is recorded as the number of CPU seconds required to obtain the best solution within a particular run. Table 2 reports the results of the conducted experiments.

In Table 2, the results reveal that the APS_{ACO} with QRA shows very good results on QAP and less in TSP as compared with other state-of-the-art methods. In TSP, while sometimes RandallG finds the best quality solutions (such as st70, gr96 and d198), the overall behavior of APS_{ACO} with QRA is better for small TSP problems without local search. In QAP, the KhichaneG sometimes finds the best solution (such as tai35a). However, the overall performance of the proposed method is better. In Table 3, the comparison of the three proposed strategies of rewards assignment is depicted.

The QRA strategy was the best performance, while the ERA showed less performance and the URA came in last. The design of each proposed strategies determines the suitable situation to apply any of them. For example, when a restart mechanism is applied, the URA will be a promising choice because of its trend to increase the current exploration. In Fig. 4, the overall performance of the proposed strategies outperformed the state-of-the-art methods for TSP instances.

With small size instances, the three methods are the best. The QRA strategy was the best among all. In some cases, the ERA strategy is outperformed (such as gr96 and rd100). However, when the size of the problem

Table 2: The results of the comparison of quality-based APS_{ACO} (Xrltdnss = 0.8) method with Randall G and Khichane G methods on the TSP instances

	Randall G				Khichane G				APS_{ACO} (QRA)			
	Cost			Runtime	Cost			Runtime	Cost			Runtime
TSP problem	Min.	Med.	Max.	Med.	Min.	Med.	Max.	Med.	Min.	Med.	Max.	Med.
eil51	0.23	0.77	2.11	1.80	0.46	0.86	1.17	4.40	0.00	0.11	0.46	2.41
st70	0.14	1.51	3.25	6.52	0.74	2.28	4.14	4.26	0.44	0.91	2.37	6.49
eil76	0.18	1.41	2.97	3.77	0.92	1.89	2.60	4.86	0.00	0.29	0.92	5.70
gr96	0.33	1.39	4.68	4.22	1.91	3.08	4.76	5.61	0.57	1.13	1.55	4.73
rd100	0.05	1.10	3.53	5.27	1.01	2.64	6.11	6.91	0.32	0.76	1.87	8.24
bier127	1.61	2.96	5.24	8.20	2.80	3.76	4.98	3.39	1.53	2.48	3.61	4.99
d198	1.96	4.21	6.82	7.74	3.07	4.96	6.84	8.89	2.20	3.94	5.48	6.14
nug15	0.00	0.00	0.00	0.22	0.00	0.00	0.00	0.19	0.00	0.00	0.00	0.05
nug20	0.00	0.04	0.15	2.75	0.00	0.01	0.15	2.64	0.00	0.00	0.00	1.76
tai25a	1.40	2.20	2.70	4.77	1.68	2.24	2.61	5.54	1.21	1.88	2.55	4.50
tai35a	2.98	3.22	3.60	4.50	2.58	3.12	3.51	6.13	2.61	3.02	3.38	3.77
ste36a	2.45	3.30	4.20	6.14	1.61	3.14	4.42	4.13	0.92	2.26	3.19	3.71
tho40	1.52	1.88	2.11	5.31	1.39	1.98	2.32	6.33	1.05	1.73	2.08	4.92
sko49	1.08	1.38	1.75	6.06	0.95	1.32	1.54	4.51	0.85	1.18	1.40	5.44

Min.: Minimum; Max.: Maximum

Table 3: The results of the comparison of quality-based APS_{ACO} (Xrltdnss = 0.8) method with Randall G and Khichane G methods on the QAP instances

	APS_{ACO} (QRA)				APS_{ACO} (URA)				APS_{ACO} (ERA)			
	Cost			Runtime	Cost			Runtime	Cost			Runtime
QAP problem	Min.	Med.	Max.	Med.	Min.	Med.	Max.	Med.	Min.	Med.	Max.	Med.
eil51	0.00	0.11	0.46	2.41	0.23	0.37	0.70	4.94	0.00	0.28	1.17	2.63
st70	0.44	0.91	2.37	6.49	0.44	1.20	3.25	4.35	0.29	0.90	1.92	5.80
eil76	0.00	0.29	0.92	5.70	0.00	0.44	1.11	5.70	0.00	0.61	0.92	4.65
gr96	0.57	1.13	1.55	4.73	0.36	1.03	3.28	5.01	0.38	0.98	2.82	8.12
rd100	0.32	0.76	1.87	8.24	0.05	1.18	2.98	5.91	0.01	0.13	0.91	6.63
bier127	1.53	2.48	3.61	4.99	1.61	2.75	4.57	6.35	2.14	3.20	4.22	6.08
d198	2.20	3.94	5.48	6.14	2.74	4.40	6.36	7.79	5.15	8.79	7.43	3.51
nug15	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.22	0.00	0.00	0.00	0.27
nug20	0.00	0.00	0.00	1.76	0.00	0.04	0.15	3.93	0.00	0.01	0.15	3.52
tai25a	1.21	1.88	2.55	4.50	1.21	2.01	2.42	3.67	1.30	1.90	2.30	5.70
tai35a	2.61	3.02	3.38	3.77	3.13	3.34	3.72	5.09	2.60	3.10	3.40	5.36
ste36a	0.92	2.26	3.19	3.71	2.09	3.00	3.82	5.96	1.40	2.90	4.00	4.11
tho40	1.05	1.73	2.08	4.92	1.26	1.77	2.15	5.45	0.90	1.84	2.30	5.97
sko49	0.85	1.18	1.40	5.44	1.06	1.39	1.71	5.54	1.23	1.41	1.63	4.67

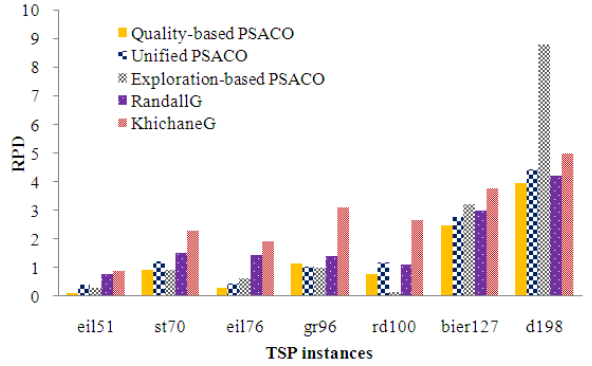

Fig. 4: The results of the comparison of quality-based APS_{ACO}, exploration-based APS_{ACO} (Xrltdnss = 0.8) and unified APS_{ACO} (Xrltdnss = 0.8) methods with the state-of-the-art methods on the TSP instances

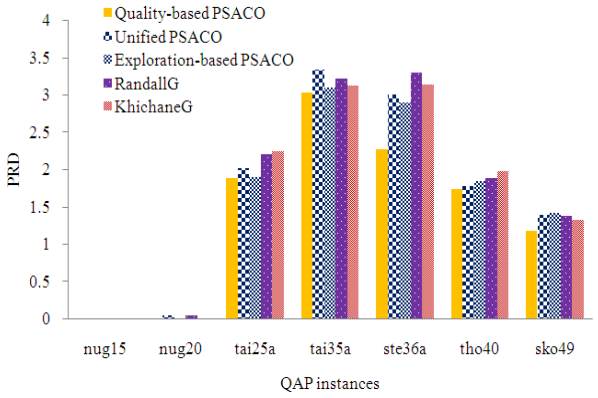

Fig. 5: The results of the comparison of quality-based APS_{ACO}, exploration-based APS_{ACO} (Xrltdnss = 0.8) and unified APS_{ACO} (Xrltdnss = 0.8) methods with the state-of-the-art methods on the QAP instances

increases, the ERA worsens the performance because of its additional computations. The URA strategy is more robust. In Fig. 5, the overall performance of the proposed strategies outperformed the state-of-the-art methods for QAP instances.

With small size instances, the URA has not much improved the quality of solution compared with QRA, ERA and KhichaneG. The QRA strategy produced the best average quality of solutions in all experiments. With small size instances, the URA does not improve much the quality of solution as compared to QRA, ERA and KhichaneG. The QRA strategy produced the best average quality of solutions in all experiments. It can be concluded that the simplicity and the effectiveness in calculating the rewards in each of the proposed strategies are critical in determining the contribution of the perspective strategy on the search.

CONCLUSION

The parameter controllers proposed in this study are to supplement the limitation of parameters' selection in ACO-based reactive search. Simple statistics about exploration behavior with machine learning procedure of clustering were the soul of the proposals. Two dependent components are highlighted in the schema of parameter adaptation. These are the parameters' selection and the reward assignment. For the parameters' selection, the implementation of the first group is followed. For the rewards assignment, three contributed strategies, denoted by QRA, ERA and URQ, are proposed for improving the performance of parameter adaptation in ACO. Three variants of APS_{ACO} algorithm can be alternated by varying the proposal of the perspective strategy. In the design of QRA, the general improvement in the quality of solutions used a proxy for the impact of the selected parameters. In ERA and URQ, the feedback from the search process is collected based on novel swarm intelligence idea. The idea is an attempt to emulate the acoustical mimicry phenomena in several host-parasites systems of some insects in nature. By the said contribution, the parameters' selection problem in ACO is addressed. The proposed APS_{ACO} variants are evaluated against each other and against the state-of-the-art methods. The results confirmed the effectiveness

of the proposed controllers. Future work can be focused on generalizing the proposed controllers for other swarm intelligence algorithms.

ACKNOWLEDGMENT

The authors wish to thank the School of Computing in Universiti Utara Malaysia, Kedah, Malaysia for the funding and the administration of this study.

REFERENCES

Barbero, F., S. Bonelli, J.A. Thomas, E. Balletto and K. Schönrogge, 2009. Acoustical mimicry in a predatory social parasite of ants. J. Exp. Biol., 212(Pt 24): 4084-4090.

Barbero, F., D. Patricelli, M. Witek, E. Balletto, L.P. Casacci, M. Sala and S. Bonelli, 2012. *Myrmica* ants and their butterfly parasites with special focus on the acoustic communication. Psyche, 2012(2012): 11, Article Id: 725237.

Broderick, C., R. Soto, E. Monfroy and F. Johnson, 2014. Self-adaptive systems: Facilitating the use of combinatorial problem solvers. Proceeding of the Part I, International Conference, HCI International 2014. Heraklion, Crete, Greece, June 22-27, 434: 503-508.

Collings, J. and E. Kim, 2014. A distributed and decentralized approach for ant colony optimization with fuzzy parameter adaptation in traveling salesman problem. Proceeding of the IEEE Symposium on Swarm Intelligence (SIS, 2014). Florida, U.S.A., pp: 1-9.

Dorigo, M. and T. Stützle, 2010. Ant Colony Optimization: Overview and Recent Advances. In: Gendreau, M. and J. Potvin (Eds.), Handbook of Metaheuristics. Springer US, New York, USA, pp: 227-263.

Förster, M., B. Bickel, B. Hardung and G. Kókai, 2007. Self-adaptive ant colony optimisation applied to function allocation in vehicle networks. Proceeding of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO, 2007). ACM Press, London, UK, pp: 1991-1998.

Khichane, M., P. Albert and C. Solnon, 2009. An ACO-based reactive framework for ant colony optimization: First experiments on constraint satisfaction problems. Learn. Intell. Optimiz., 5851: 119-133.

Liu, X., C. Yang and X. Liu, 2011a. Optimization of vehicle routing problem based on max-min ant system with parameter adaptation. Proceeding of the 7th IEEE International Conference on Computational Intelligence and Security. Hainan, China.

Liu, Y., G. Ang, H. Chen, Z. Zhao, X. Zhu and Z. Liu, 2011b. An adaptive fuzzy ant colony optimization for feature selection. J. Comput. Inform. Syst., 4(7): 1206-1213.

Lopez-Ibanez, M. and T. Stützle, 2014. Automatically improving the anytime behaviour of optimisation algorithms. Eur. J. Oper. Res., 235(3): 569-582.

Martens, D., M. De Backer, R. Haesen, S. Member, J. Vanthienen and M. Snoeck, 2007. Classification with ant colony optimization. IEEE T. Evolut. Comput., 11(5): 651-665.

Melo, L., F. Pereira and E. Costa, 2010. MC-ANT: A Multi-Colony ant algorithm. In: Collet, P. *et al*. (Eds.), EA, 2009. LNCS 5975, Springer-Verlag, Berlin, Heidelberg, pp: 25-36.

Neyoy, H., O. Castillo and J. Soria, 2013. Dynamic fuzzy logic parameter tuning for ACO and its application in TSP problems. In: Castillo, O., P. Melin and J. Kacprzyk (Eds.), Studies in Computational Intelligence (SCI): Recent Advances on Hybrid Intelligent Systems. Springer, Heidelberg, Germany, 451: 259-271.

Olivas, F., F. Valdez and O. Castillo, 2014. A fuzzy system for parameter adaptation in ant colony optimization. Proceeding of the IEEE Symposium on Swarm Intelligence. Orlando, FL, pp: 1-6.

Pellegrini, P., T. Stützle and M. Birattari, 2012. A critical analysis of parameter adaptation in ant colony optimization. Swarm Intell., 6(1): 23-48.

Randall, M., 2004. Near parameter free ant colony optimisation. In: Dorigo, M. *et al*. (Eds.), ANTS 20004. LNCS 3172, Springer-Verlag, Berlin, Heidelberg, pp: 374-381.

Sagban, R., K.R. Ku-Mahamud and M. Shahbani, 2014. Reactive memory model for ant colony optimization and its application to TSP. In Proceedings of the 4th International Conference on Control System, Computing and Engineering (ICCSCE, 2014). IEEE, Penang, Malaysia, pp: 310-315.

Sagban, R., K.R. Ku-Mahamud and M.S.A. Bakar, 2015. ACOustic : A nature-inspired exploration indicator for ant colony optimization. Sci. World J., 2015(2015): 11, Article ID 392345.

Solnon, C., 2010. Ant Colony Optimization and Constraint Programming. Weley and Sons Ltd., U.S.A.

Stützle, T. and H.H. Hoos, 2000. MAX–MIN ant system. Future Gener. Comp. Sy., 16(8): 889-914.

Stützle, T., M. Maur and M. López-Ibáñez, 2010. Pre-scheduled and adaptive parameter variation in MAX-MIN ant system. Proceeding of the IEEE Congress on Evolutionary Computation (CEC, 2010), pp: 1-8.

Stützle, T., L. Manuel, P. Pellegrini, M. Maur, M.M. De Oca, M. Birattari and M. Dorigo, 2012. Parameter adaptation in ant colony optimization. In: Hamadi, Y. *et al*. (Eds.), Autonomous Search. Springer-Verlag, Berlin, Heidelberg, pp: 191-215.

Wang, Y., 2013. Adaptive ant colony algorithm for the VRP solution of logistics distribution. Res. J. Appl. Sci. Eng. Technol., 6(5): 807-811.