A Background Study on Ant Colony Optimization Metaheuristic and its Application Principles in Resolving Three Combinatorial Optimization Problems

Jamaludin Sallim[1], Wan Muhammad Syahrir Wan Hussin[2], Rosni Abdullah[3], Ahamad Tajudin Abdul Khader[4]

[1, 3, 4] School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia
jamals@cs.usm.my, rosni@cs.usm.my, tajudin@cs.usm.my

[2] Faculty of Computer Systems and Software Engineering,
University Malaysia Pahang, 25000, Kuantan, Pahang, Malaysia
wmsyahrir@ump.edu.my

Abstract—Ant Colony Optimization (ACO) is a metaheuristic that inspired by the behaviour of real ant colonies and can be considered as a one of recent metaheuristic approach that has been proposed. In this paper, we review the approach by ACO metaheuristic in resolving three the combinatorial optimization problems (COPs). The chosen COPs are: Traveling Salesman Problem; Sequential Ordering Problem; and Network Routing Problem that exploit a general ACO framework. For each problem, we review on the identified ACO application principles which comprise four basic issues that play an important role in these COPs. The four basic issues are: construction graph; constraint; pheromone trails and heuristic information; and solution construction.

INTRODUCTION

Ant Colony Optimization (here-on referred to as ACO) is a recent-proposed metaheuristic that inspired by the behaviour of real ant colonies which enables ants to find shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit a substance called *pheromone* on the ground. When they decide about a direction to go, they choose with higher probability paths that are marked by stronger pheromone concentrations. This basic behavior is the basis for a cooperative interaction which leads to the emergence of shortest paths. ACO is widely been used to resolve combinatorial optimization problems (here-on referred to as COPs) and rapidly become a popular optimization technique. In this paper we only review three COPs which consist of *Traveling Salesman Problem*, *Sequential Ordering Problem*, and *Network Routing Problem*. Initially the first ACO algorithm which called *Ant System* (AS) is introduced by Marco Dorigo (1991) and was later developed, finally leading to the formalization of the approach as ACO by Dorigo and Di Caro (1999) and Dorigo, Bonabeau and Theraulaz (2000).

The term *heuristic* can be referred as something relating to or using a problem-solving technique in which the most appropriate solution of several found by alternative methods is selected at successive stages of a program for use in the next step of the program. A 'metaheuristic', which defined by [3], is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems or can be seen as a general purpose heuristic method designed to guide an underlying problem specific heuristic toward promising regions of the search space containing high quality solutions. Therefore, a metaheuristic is a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to specific problem. Fred Glover and Manuel Laguna, (Tabu Search, 1998) quoted that "A metaheuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality." Dorigo and Stutzle (2004) also mentioned that the use of metaheuristic has significantly increased the ability of finding very high-quality solutions to hard, practically relevant combinatorial problems in a reasonable time.

In general definition by wikipedia, combinatorial optimization is a branch of optimization in applied mathematics and computer science that related to operations research, algorithm theory and computational complexity theory that sit at the intersection of several fields, including artificial intelligence, mathematics and software engineering. Combinatorial optimization algorithms solve instances of problems that are believed to be hard in general, by exploring the usually-large solution space of these instances. They achieve this by reducing the effective size of the space, and by exploring the space efficiently. [3] stated that combinatorial optimization problems involve finding values for discrete variables such that the optimal solution with respect to a given objective function is found. Many optimization problems of practical and theoretical importance are of combinatorial nature.

M. Affenzeller and R. Mayrhofer (2002) stated that problems of combinatorial optimization are characterized by their well-structured problem definition as well as by their huge number of action alternatives in practical application areas of reasonable size. Especially in areas like routing, task allocation, or scheduling such kinds of problems often occur. Their advantage lies in the easy understanding of their action alternatives and their objective function. Therefore, an objective evaluation of the quality of action alternatives is possible in the context of combinatorial optimization problems [10].

ACO Metaheuristic

In this section, we extract the information about ACO metaheuristic (we follow precisely the description in [2]):

A. Ant Colonies

Ants are social insects that live in colonies and, because if their collaborative interaction, they are capable of showing complex behaviours and to perform difficult tasks from an ant's local perspective. A very interesting aspect about ant colonies is their ability to find shortest path between the ants' nest and the food sources. This fact is especially noticeable having in mind that in many ant species ants are almost blind, which avoids the exploitation of visual clues. While walking between their nest and food sources, some ants deposit a chemical called *pheromone*. If no pheromone trails are available, ants move essentially at random, but in the presence of pheromone they have a tendency to follow a trail.

B. From Real Ants To Artificial Ants

ACO algorithms take inspiration from the behaviour of real ant colonies to solve combinatorial optimization problems. They are based on a colony of artificial ants, that is, simple computational agents that work cooperatively and communicate through artificial pheromone trails. ACO algorithms are essentially construction algorithms: in each algorithm iteration, every ant constructs a solution to a the problem by traveling on a construction graph. Each edge of the graph, representing the possible steps the ant can make, has associated two kinds of information that guide the ant movement:

1) Heuristic information, which measures the heuristic preference of moving from node *r* to node *s*, i.e., of traveling the edge α_{rs}. It is denoted by η_{rs}. This information is not modified by the ants during the algorithm run.

2) (Artificial) pheromone trail information, which measures the "learned desirability" of the movement and mimics the real pheromone that natural ants deposit. This information is modified during the algorithm run depending on the solutions found by the ants. It is denoted by τ_{rs}.

This section introduces the steps leading from real ants to ACO. It should be noted for the following that ACO algorithms present a double perspective:

1) On the one hand, they are an abstraction of some behavioral patterns of natural ants related to the shortest path searching behavior.

2) On the other hand, they include several features that do not have a natural counterpart, but that allow to develop algorithms for obtaining good solutions to the problem tackled (for example, the use of heuristic information to guide the ant movement).

ACO Metaheuristic Framework

ACO metaheuristic framework consists of three parts gathered in the *ScheduleActivities* construct. The *ScheduleActivities* construct does not specify how these three activities are scheduled and synchronized. This is up to the algorithm designer [13].

```
WHILE termination conditions not met DO
  ScheduleActivities
    AntBasedSolutionConstruction()
    PheromoneUpdate()
    DaemonActions() {optional}
  END ScheduleActivities
ENDWHILE
```

A. AntBasedSolutionConstruction()

An ant constructively builds a solution to the problem by moving through nodes of the construction graph *G*. Ants move by applying a stochastic local decision policy that makes use of the pheromone values and the heuristic values on components and/or connections of the construction graph. While moving, the ant keeps in memory the partial solution it has built in terms of the path it was walking on the construction graph.

B. PheromoneUpdate()

When adding a component c_i to the current partial solution, an ant can update the values of the pheromone trails that where used for this construction step. This kind of pheromone update is called *online step-by-step pheromone update*. Once an ant has built a solution, it can (by using its memory) retrace the same path backward and update the pheromone trails of the used components and/or connections according to the quality of the solution it has built. This is called *online delayed pheromone update*. Another important concept in Ant

Colony Optimization is *pheromone evaporation*. Pheromone evaporation is the process by means of which the pheromone trail intensity on the components decreases over time. From a practical point of view, pheromone evaporation is needed to avoid a too rapid convergence of the algorithm toward a sub-optimal region. It implements a useful form of *forgetting*, favoring the exploration of new areas in the search space.

C. DaemonActions()

Daemon actions can be used to implement centralized actions which cannot be performed by single ants. Examples are the use of a local search procedure applied to the solutions built by the ants, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process from an non-local perspective. As a practical example, the daemon can observe the path found by each ant in the colony and choose to deposit extra pheromone on the components used by the ant that built the best solution. Pheromone updates performed by the daemon are called *offline pheromone updates*.

In discrete optimization problems, D. Angus (2006) stated that the ACO meta-heuristic framework can be applied by having a finite set of components with connections between these components (with associated costs). In summary, ACO is described as being responsible for the scheduling of three processes [9]:

1) Ants generation & activity
2) Pheromone trail evaporation
3) Daemon actions

Three COPs: General Overview Of The Problems

A. Traveling Salesman Problem

The traveling salesman problem is the problem faced by salesman who, starting from his home town, wants to find a shortest possible trip through a given set of customer cities, visiting each city once before finally returning home. The TSP can be represented by a complete weighted graph $G = (N , A)$ with N being the set of $n = | N |$ nodes (cities), A being the set of arcs fully connecting to the nodes. Each are $(i,j) \epsilon$ A is assigned a weight $d_{i,j}$ which represents the distance between cities i and j [3]. [4] stated that the traveling salesman problem plays a central role in ant colony optimization because it was the first problem to be attacked by these methods (see (Dorigo, 1992; Dorigo *et al.*, 1991; Dorigo *et al.*, 1996)). The reasons the TSP was chosen are:

1) it is relatively easy to adapt the ant colony metaphor to it
2) it is a very dif-ficult problem (*NP*-hard),
3) it is one of the most studied problems in combinatorial optimization (Lawler *et al.*, 1985;

B. Sequential Ordering Problem

The sequential ordering problem with precedence constraints (SOP) was first formulated by Escudero (1988) to design heuristics for a production planning system. It consists of finding a minimum weight Hamiltonian path on a directed graph with weights on the arcs and the nodes, subject to precedence constraints among nodes [12].

C. Network Routing Problem

Based on [3], Let a telecommunications network be defined by a set of nodes N, a set of of links between nodes L_{net}, and the cost $d_{i,j}$ associated with the links. Then the network routing problem (NRP) is the problem of finding a minimum cost paths among all pairs of nodes in the network. It should be noted that if the costs $d_{i,j}$ are fixed, then the NRP is reduced to a set of minimum cost path problems, each of which can be solved efficiently via a polynomial time algorithm like Dijkstra's algorithm (Dijkstra, 1959). The problem becomes interesting for heuristic approach once, as happens in real world applications like routing in communication networks, costs (e.g., data traffic in links) or the network topology varies in time.

ACO Application Principles

M. Dorigo and T. Stützle (2004) had identified some basic issues for ACO application principles:

C. Construction Graph

Given the respective formulation, artificial ants build solutions by performing randomized walks on the completely connected graph $G_C = (C,L)$ whose nodes are components C, and the set L fully connects the components C. The graph G_C is called *construction graph* and elements of L are called *connections*.

D. Constraint

The problem constraint Ω (t) are implemented in the policy followed by artificial ants. The choice of implementing the constraint in the construction policy of the artificial ants allows a certain degree of flexibility. In fact, depending on COP considered, it may be more reasonable to implement the constraints in hard way, allowing the ants to build only feasible solutions, or in a soft way, in which case the ants can build infeasible solutions (i.e candidate solution in $S \setminus \check{S}$) that can be penalized as a function of their degree of infeasibility.

E. Pheromone Trails and Heuristic Information

Initially, ants explore the area surrounding their nest in a random manner. As soon as an ant finds a source of food, it evaluates quantity and quality of the food and carries some of this food to the nest. During the return trip, the ant deposits a pheromone trail on the ground. The quantity of pheromone deposited, which may depend on the quantity and quality of the food, will guide other ants to the food source. The indirect communication between the ants via the pheromone trails allows them to find the shortest path between their nest and food sources. This functionality of real ant colonies is exploited in artificial ant colonies in order to solve optimization problems. In ACO algorithms the pheromone trails are simulated via a parameterized probabilistic model that is called the pheromone model. The pheromone model consists of a set of model parameters whose values are called the pheromone values. The basic ingredient of ACO algorithms is a constructive heuristic that is used for probabilistically constructing solutions using the pheromone values.

F. Solution Constructions

Solution Construction is a process of an ant constructively builds a solution to the problem by moving through nodes of the construction graph *G*. Ants move by applying a stochastic local decision policy that makes use of the pheromone values and the heuristic values on components and/or connections of the construction graph. While moving, the ant keeps in memory the partial solution it has built in terms of the path it was walking on the construction graph.

Application Principles For COPs

M. Dorigo and T. Stützle (2004) had described four elements or issues of ACO application principles that need to be considered [3]. As mentioned earlier, we choose three COPs, based on the diversity of the problems.

A. Traveling Salesman Problem

Construction graph. The construction graph is identical to the problem graph: the set of components *C* corresponds to the set of nodes (i.e., $C = N$), the connections correspond to the set of arcs (i.e., L = A), and each connection has a weight which corresponds to the distance d_{ij} between nodes *i* and *j*. The states of the problem are the set of all possible partial tours.

Constraints. The only constraint in the TSP is that all cities have to be visited and that each city is visited at most once. This constraint is enforced if an ant at each construction step chooses the next city only among those it has not visited yet (i.e., the feasible neighborhood N^k_i of ant *k* in city *i*, where k is the ant's identifier, comprises all cities that are still unvisited).

Pheromone trails and heuristic information. The pheromone trails $\tau_{i,j}$ in the TSP refer to the desirability of visiting city *j* directly after *i*. The heuristic information η_{ij} is typically inversely proportional to the distance between cities *i* and *j*, a straight-forward choice being $\eta_{ij} = 1/ d_{i,j}$. In fact, this is also the heuristic information used in most ACO algorithms for the TSP.

Solution construction. Each ant is initially put on a randomly chosen start city and at each step iteratively adds one still unvisited city to its partial tour. The solution construction terminates once all cities have been visited.

B. Sequential Ordering Problem

Construction graph. Similar to TSP, the set components C contains of set of the nodes. Solutions are permutation of the elements of C, and costs (lengths) are associated with the connections between nodes.

Constraints. The only significant difference between the applications of ACO to the SOP and to the TSP is the set of constraints: while building solutions, ants choose components only among those that have not yet been used and, if possible, satisfy all precedence constraint.

Pheromone trails and heuristic information. As in the TSP case, pheromone trails are associated with connections, and the heuristic information can, for example, be chosen as the inverse of the costs (lengths) of the connections.

Solution Construction. Ants build solutions iteratively by adding, step by step, new unvisited nodes to the partial solution under construction. They choose the new node to add by using pheromone trails, heuristic, and constraint information.

C. Network Routing Problem

Construction graph. The construction graph is the graph $G_C = (C,L)$, where *C* corresponds to the set of nodes, *N*, and *L* fully connects G_C. Note that $L_{net} \epsilon L$.

Constraints. The only constraint is that ants use only connections $l_{ij} \epsilon L_{net}$

Pheromone trails and heuristic information. Because the NRP is, in reality, a set of minimum cost path problems, each connection $l_{ij} \epsilon L$ should have many different pheromone trails associated. For example, each connection l_{ij} could have associated one trail value τ_{ijd} for each possible destination node d an ant located in node i can have. Each arc can also be assigned a heuristic value η_{ij} independent of the final destination. The heuristic value η_{ij} can be set, for example, to a value inversely proportional to the amount of traffic on the link connecting nodes *i* and *j*.

Solution construction. Solution construction is straightforward. In fact, the S-ACO algorithm presented in chapter 1, section 1.3.1, is an example of how to proceed. Each ant has a source node *s* and a destination node, *d*, and moves from *s* to *d* hoping from one node to the next, until node *d* has been reached. When ant *k* is located at node *i*, it choose the next node *j* to move to using a probabilistic decision rule which is a function of the ant's memory, of local pheromones, and heuristic information.

Summary

Combinatorial optimization problems arise in many practical and theoretical problems which are very hard to resolve. ACO metaheuristic, which based on a generic problem representation and the definition of ants' behaviour, has been proposed to overcome these problems. ACO has application principles that consist of four issues or elements need to be considered. It is very important to researchers to identify and determine these four issues or elements in order to start resolving the problems.

References

[1] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243–278, 2005.

[2] O. Cordón, F. Herrera, T. Stützle, A Review of the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends**.** *Mathware and Soft Computing 9:2-3* (2002) 141-175.

[3] M. Dorigo and T. Stützle, 2004. *Ant Colony Optimization*. MIT Press, Cambridge, MA.

[4] M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation (CEC99), pages 1470–1477. IEEE, 1999.

[5] M. Dorigo and G. Di Caro. The Ant Colony Optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw Hill, London, UK, 1999.

[6] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.

[7] M. Dorigo and T. Stützle. The Ant Colony Metaheuristic: Algorithms, Applications and Advances. Technical report. IRIDIA, 2000.

[8] D. Angus. The Ant Colony Optimization: From Biological Inspiration to an Algorithmic Framework. Technical report. CISCP, 2006.

[9] M. Affenzeller and R. Mayrhofer. Generic Heuristic for Combinatorial Optimization Problems. Proceedings of the 9th International Conference on Operational Research (KOI), pp. 83-92. 2002.

[10] M. Dorigo, V. Maniezzo and A. Colorni, The Ant Systems: Optimization by a Colony of Cooperating Agents, IEEE Transactions on Systems, Man, and Cybernetics – Part B, 26(2) (1996) 29-41.

[11] L.M Gambardella, M.Dorigo,. An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem, INFORMS Journal on Computing, vol.12(3), pp. 237-255, 2000

[12] C. Blum and M. Manfrin, Online website. Available: www.metaheuristic.org

[13] M. Dorigo, Online website. Available: www.aco-metaheuristic.org

[14] General definition. Online. Available: www.wikipedia.org