Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Improved auto control ant colony optimization using lazy ant approach for grid scheduling problem

Pawan Kumar Tiwari, Deo Prakash Vidyarthi *

School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

HIGHLIGHTS

- This work proposes an improvement in auto controlled ant colony optimization method for grid scheduling problem.
- Auto Control mechanism dynamically adapts changes in GSP through the ant.
- The concept of Lazy Ants has been introduced, which fully exploits the solution around the best ant.
- The algorithm is effective in producing better result as well as efficient in computational time over some contemporary ACO, PSO, GA, QGA and AACO for GSP.

ARTICLE INFO

Article history:
Received 16 March 2015
Received in revised form
25 October 2015
Accepted 27 January 2016
Available online 6 February 2016

Keywords:
Auto controlled Ant Colony Optimization (AACO)
Grid Scheduling Problem (GSP)
Quantum Genetic Algorithm (QGA)
Simulated Annealing (SA)
Particle Swarm Optimization (PSO)

ABSTRACT

An auto controlled ant colony optimization algorithm controls the behavior of the ant colony algorithm automatically based on a priori heuristic. During the experimental study of auto controlled ACO algorithm on grid scheduling problem, it was observed that the induction of lazy ants not only reduces the time complexity of the algorithm but also produces better results on the given objectives. Lazy ants are basically a mutated version of active ants that remain alive till the fitter lazy ants are generated in the successive generations. This work presents an improved auto controlled ACO algorithm using the lazy ant concept. Performance study reveals the efficacy and the efficiency achieved by the proposed algorithm. A comparative study of the proposed method with some other recent meta-heuristics such as auto controlled ant colony optimization algorithm, genetic algorithm, quantum genetic algorithm, simulated annealing and particle swarm optimization for grid scheduling problem exhibits so.

1. Introduction

Ant colony optimization (ACO) is an evolutionary computational technique proposed by Dorigo et al. [1–4]. It is a population based stochastic optimization approach inspired from the ant movement and can be applied to the combinatorial optimization problems with minimal changes. It transcribes the cooperative behavior of the ant colony in solving an optimization problem. During the traversal in search of food, ants secrete a chemical trail called pheromone. The role of the trail is to share the experience of the ants about their already traversed journey. Though, ACO provides a better solution to a wide class of combinatorial optimization problems but the parameter mapping of this technique with the tuned constraint of the given problem is a difficult task. A typical distributive cooperation is observed in the ant colony using the smell of a chemical secreted by each ant during the path construction. Message of good constructive path is conveyed to the ant colony by the pheromone concentration deposited on the path. Pheromone evaporation also takes place with the passage of time otherwise a path of forgotten ants may mislead other ants forever. Due to this evaporation, the attraction of a less frequently used path diminishes for other ants. The stochastic nature of the algorithm is derived from the fact that at each step ants take the decision based on two parameters; attractiveness and trail. Attractiveness, in this context, is the prior information of the promising solution and trail represent the posterior information of the previously obtained good solution. Combination of the above two information is achieved nicely by a probabilistic formula, which plays a crucial role in the exploration of the solutions.

The ACO algorithm was first applied on Traveling Sales Person (TSP) problem [1,2]. A plenty of ACO variants, since then, have been developed [3]. Ant system (AS) [1], Ant Colony System (ACS) [2], Max-Min Ant System (MMAS) [5], Rank based Ant System (RAS) [6], etc. are some of them that has attracted the attention

* Corresponding author.
E-mail addresses: pawantiwari6@gmail.com (P.K. Tiwari), dpv@mail.jnu.ac.in (D.P. Vidyarthi).

http://dx.doi.org/10.1016/j.future.2016.01.017
0167-739X/© 2016 Elsevier B.V. All rights reserved.

of researchers. In ACS, the inclusion of random proportion choice rule for the movement from one state to another reduces the computational time drastically in comparison to the earlier proposed AS [1,2]. These algorithms can be viewed as the interplay of the procedures such as parameter initialization, ant initialization, construction of solution, local pheromone updating, global pheromone updating and termination criteria. ACO and its variants have been widely applied in various fields such as vehicle routing [7], network routing [8], robotics [9], job-shop scheduling [10], grid scheduling [11–16], etc.

Grid scheduling is a well-known NP-class problem exhibiting a very large search space of possible solutions [16–20]. Not only the grid resources are dynamic even the load on these resources varies from time to time and hence a good example of dynamic environment. In order to obtain a near optimal solution, many meta-heuristics have been applied for this as discussed in [16,21–28]. Some of these are based on ACO focusing on the minimization of turnaround time. GA has also been frequently applied for grid scheduling problem [21–23]. The performance evaluations of the models are done on the benchmark data from [29]. Liu et al. proposed fuzzy particle swarm optimization (PSO) based model for GSP and could show its efficacy over GA and simulated annealing (SA) [24]. An AACO model is proposed to study the effect of inter-process communication in optimizing turnaround time with grid scheduling [16]. It has been compared with few other models that use GA for GSP [24,28]. It is observed that AACO model outperforms GA based models. QGA algorithm has also been applied and compared with GA for GSP in [27].

Liu et al. [24] observed that traditional ACO based model takes much time in the convergence. AACO achieves better solution in less number of iteration, due to auto control mechanism of heuristic information, in comparison to other models as depicted in Liu et al. and hence reduces the computational time drastically. In one of our previous work, an Auto controlled ACO (AACO) to observe the Inter Process communication (IPC) for the scheduled jobs in a Computational Grid (CG) was proposed in [16]. The major contribution of AACO algorithm was auto controlling the heuristic information at each step of the solution construction, which was not seen in any other ACO variants applied for the grid scheduling problem. In fact on TSP also, none of the ACO variants update heuristic information. In all of these, heuristic information of ant movement from a city to another is reciprocal value of the distance between the two cities and hence is a constant value known a-priori. In any dynamic environment, the desirability of a particular action (e.g. node selection for task allocation in CG) depends on the time (step of ant movement) which can be realized and observed in nature as well. This nature of dynamic CG environment is well incorporated in the AACO model through auto controlling the heuristic information.

In a biological study, it was observed that in some ant colonies ant switches from active state to lazy state [30]. Inspired from this, an Improved Auto-Controlled Ant Colony Optimization (IAC-ACO) is proposed in this work for GSP in which, in each iteration, a portion of ants changes its states from active state to lazy state. Lazy ants enhance the ability of exploitation process with lesser computational cost. The newly proposed IAC-ACO retains all other features of AACO and incorporates the concept of lazy ants with more sophisticated auto control mechanism for parameter updation.

The outline of the paper is as follows. In Section 2 the grid scheduling problem, for which the proposed approach has been applied, is presented. In Section 3, the concept of lazy ant as well as the proposed IAC-ACO method for grid scheduling problem has been elaborated. In Section 4, experimental verification and analysis of the proposed method is given. Section 5 concludes the work.

2. Grid scheduling problem: an application

Grid is composed of different types of processing units, storage units, scientific instrumentation etc. called the nodes of the grid system. Scheduling is done to optimize some characteristic parameters related to system or job such as throughput, system utilization, turnaround time, waiting time, response time, fairness, level of security, availability etc. [15]. Scheduling of the job/tasks is an essential activity of any computing systems in order to enhance the performance of the system. An intensive research has been done in this area and the results have been widely publicized.

Scheduling in the grid is entirely different from the scheduling on a uniprocessor system wherein a single objective of maximal CPU utilization is heeded by keeping CPU busy all through [31]. With the emergence of OGSA (Open Grid Service Architecture), newer scheduling models and algorithms are in demand for addressing new concerns in the grid environment. The complexity of scheduling problem increases with the size of the grid which makes it highly infeasible [32,33].

Generally, there are three phases in grid scheduling. Phase one discovers the set of resources according to the requirements of the job/task. Phase two collects the information on queuing length, reliability and the speed of the node to match it with the job. In the third phase, allocation and execution takes place. A global grid scheduler, also called resource broker, acts as an interface between the user and the distributed grid resources hiding the complexities of the computational grid. Resource broker performs all the three phases of scheduling for the submitted jobs in a grid. Besides, it also monitors the progress on the job execution along with adapting to the changes, variation in resource share availability, and failures etc. The detail about the grid scheduling problem may be extracted from [15,16,26–28].

In OGSA, users submit their jobs unaware of other users' job. So, there might be a queue formation before the grid nodes. According to queuing theory [34,35], scheduling problem in computational grid falls under M/M/k model. This work considers the turnaround time [16] as a parameter to be optimized for a submitted job, the fitness function of which has been derived as below.

Let T denote the set of submitted tasks of a given job in CG at any instance of time and N denote the set of available nodes in the CG. A schedule q is basically a mapping from the set of tasks to the set of nodes as given in Eq. (1).

$$q : T \to N. \tag{1}$$

Let Sch denote the set of all such schedules i.e. $Sch = \{q \mid q : T \to N\}$.

The fitness of a particular schedule q, denoted by L_q, can be obtained by reciprocating the total time of a given job i.e. $T_q + \Omega_q$ as given in Eq. (2). Two cases have been considered; one in which no IPC is involved (WC-case) and the other in which IPC (IPC-case) is involved. For WC-case, fitness of the schedule is calculated keeping $\Omega_q = 0$ in Eq. (2).

$$L_q = \frac{1}{\left|T_q + \Omega_q\right|}. \tag{2}$$

T_q is the total time taken by the grid for a schedule q and is computed as given in Eq. (3).

$$T_q = Max_{i=1}^{m}\left(\left(\sum_{j=1}^{l}\frac{\lambda_i}{\mu_i\left(\mu_i - \lambda_i\right)} + \frac{1}{\mu_i}\right)\cdot N_j \cdot \chi_{ji}\right) \tag{3}$$

Ω_q is the total time taken by the grid for IPC between the tasks for the schedule q which is obtained as given in Eq. (4).

$$\Omega_q = \underset{\substack{1\le i,k\le m\\ i<k}}{Max}\left[\sum_{r=1}^{l}\sum_{s=r+1}^{l}\left\{\sum_{t=0}^{hd(k,i)-1} f(\lambda,\mu)\right\} C_{rs}\cdot \chi_{kr}\,\chi_{is}\right] \tag{4}$$