Multi-Agent Based Process Scheduling Using Ant Colony Optimization: An Algorithmic Approach

[1] M. K. Ahmed, [2]A. Aliyuda and [3]A.Y. Umar

[1, 2, 3]Department of Mathematics, Computer Science Unit,

Gombe State University, Gombe, Nigeria

Abstract

Scarcities of resources coupled with distributive and complex nature of activities have caused various organizations and environments to face challenges when it comes to process scheduling. As such, there is need to use new programming paradigm (Agent Oriented Programming) in order to come up with efficient algorithm that will solve this problem. In this work, a new Agent Based Algorithm is presented using a Meta heuristic technique called Ant Colony Optimization (ACO) that can be used in Task Assignment (Process Scheduling) problem. The Proposed Algorithm finds the optimal possible solution with small computational efforts that will reduce the waiting time of multiple processes when competing for the various resources.

Keywords: Agent, Muti-Agent, Ant Colony Optimization, Process Scheduling;

Introduction

An agent is a computer system within an environment and with an autonomous behavior made for achieving the objectives that were set during its design (Wooldridge, 1999). A multi-agent system (MAS) is a system that contains a set of agents that interact via communications protocols and are able to act on their environment. Different agents have different spheres of influence, mainly because of their control (or at least an influence) on different parts of the environment. In some cases, these spheres of influence may overlap which causes dependency of reports between the agents (Wooldridge, 2002).

It is interesting to note that not all problems require agent based approach. Some problems need other approaches other than the agent based. Among the most important characteristics of problems that may require agent or multi-agent approach include: cases in which knowledge that is required to solve a problem is spatially distributed in different locations; the solution of a problem involves the coordination of the effort of different individuals with different skills and functions; the problems are quite complex and finding standard software engineering solution for them is difficult. In addition, Multi-Agent based approach can be employed in a situation where accessing the most relevant information as easily, flexibly and timely as possible becomes necessary (Moreno, 2003). In a multi-processing environment agent approach can easily be employed together with Ant Colony Optimization since it works quite well in such kind of environment.

Blum, (2011) gives behavior of each ant in nature, wandering randomly at first, laying down a pheromone trail, if food is found, then return to the nest laying down a pheromone trail, if pheromone is found, with some increased probability follow the pheromone trail, once back at the nest, go out again in search of food. However, pheromones evaporate over time, such that unless they are reinforced by more ants, the pheromones will disappear.

Ant colonies, and more generally social insect societies, are distributed systems that, in spite of the simplicity of their individuals, present a highly structured social organization. As a result of this organization, ant colonies can accomplish complex tasks that in some cases far exceed the individual capabilities of a single ant. The field of ‘‘ant algorithms’’ studies models derived from the observation of real ants’ behavior, and uses these models as a source of inspiration for the design of novel algorithms for the solution of optimization and distributed control problems (Dorigo and Stutzle, 2004).

Related Work

In Umarani et al., (2012), the authors find a feasible schedule for a given task set to a set of heterogeneous processors without exceeding the capacity of the processors, which is NP-Hard. The study uses a paradigm using Ant Colony Optimization (ACO) for arriving at a schedule. An attempt is made to arrive at a feasible schedule of a task set on heterogeneous processors ensuring load balancing across the processors. The heterogeneity of the processors is modeled by assuming different utilization times for the same task on different processors. The two parameters: average wait time of tasks and utilization of processors are computed using the First Come First Served (FCFS). This approach to the tasks assignment problem using ACO performs better with respect to the two parameters used compared to the FCFS algorithm but the time taken to come up with the schedule using ACO is slightly more than that of FCFS.

In a similar development, Chen and Cheng (2005) shows that the problem of determining a set of periodic tasks can be assigned to a set of heterogeneous processors in such a way that all timing constraints are met, in general, to be NP-hard. But in this case the authors used MAX-MIN Ant System (MMAS) for the update of the pheromone trials. The paper presents a new algorithm based on Ant Colony Optimization (ACO) Meta heuristic for solving this problem. The ACO approach to the task assignment problem is presented. Preliminary test shows that this approach has better performance than a GA heuristic and Baruah's approximation algorithm In addition to being able to search for a feasible assignment solution; the approach can further optimize the solution to reduce its energy consumption.

In this research MAX-MIN Ant System will also be used in the update of the pheromone because only one single ant is used to update the pheromone trails after each iteration. In addition to that, since the key to achieve best performance of ACO algorithms is to combine an improved exploitation of the best solutions found during the search with an effective mechanism for avoiding early search stagnation. MAX–MIN Ant System has been specifically developed to meet these requirements by (Stützle and Hoosb, 2000).

Bhat et al., (2011) Identify problems of meeting scheduling (MS) in the health care sectors. In which they identify the need to implement an agent based meeting scheduling system, which can schedule a meeting between Patients and Doctors. The main contribution of the research was creation of an automated meeting scheduling agent in health care domain that can serve the following functions: It allows the patients to input his/her meeting request, Negotiates with the agents of the other requested patients, Finds out its best fitting and free time slots, Compares them with the sent fitting slots of the patient-agents and find out the best ones, Reacts to the incoming patient request by sending back its best fitting free time slots, Shows all fixed meeting in a time table and allows the patient to input his/her preferences.

From the studied literature ACO has being used in both health care sector and task assignment of multi-processor environment but they failed to come with any general algorithm that can be used for task assignment in a competitive environment.

This study intends to come up with a new general algorithm using Ant Colony Optimization based Multi-Agent approach that can be used in the process assignment (task assignment problem).

Agent Based Process Scheduling with Ant Colony Optimization

The framework consists of Process Agent (PA), Resource Agent (RA), Common Agent (CA) and Main Agent (MA). The CA is a general agent that accepts request by the Processes and the kind of resources required by a particular Process. Now, the PA knows what tasks it has to perform and then it requests for the resources. The resource agent may be R1, R2, and R3 etc. Each RA has multiple time slots. The PA requests for this time slots. Multiple PA may request for the same slot. For each resource, a RA is created. In addition to these agents, an ACO agent is used to perform the ACO optimization. On arrival of the patients, this ACO agent is called to perform optimization using ACO and an optimized schedule is generated.

Ant Colony Optimization for Process Scheduling

Process Scheduling is the process of scheduling and sequencing the processes for various multiple resources in a particular domain. In this research we followed the MAX-MIN Ant System (MMAS) for us to update the pheromone trials. Regarding pheromone trial limits, an estimate of the upper bound, f (sbest) /ρ, is used to define τ_{max}, where sbest is the best-so-far solution, and ρ is the evaporation rate of pheromone trails. τ_{min} is defined to be τ. For a Process scheduling problem (PSP) consisting of n Processes (or jobs) P = {P_1,…….,P_n} and m resources R = {R_1,….,R_m}, each Process Pi has a set of sequential task (operations) O_{ij}, i=1,2,…,n; j=1,….n_i, $n_i \leq m$; where i is index of the Process and j is the index of the task or operation.

$$\tau_{\min} = \frac{\tau_{\max} \bullet (1 - \sqrt[n]{P^{best}})}{(avg - 1) \bullet \sqrt[n]{P^{best}}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 1[9]$$

$$\tau_{\min} = \frac{\frac{1}{1-\rho} \bullet (1 - \sqrt[n]{P^{best}})}{(avg - 1) \bullet \sqrt[n]{P^{best}}} \quad \ldots\ldots\ldots\ldots..2$$

Since,

$$\tau_{\max} = \frac{1}{1-\rho} \quad \ldots\ldots\ldots\ldots 3[9]$$

Where P^{best} = the probability of constructing best solution found.

n = the total number of items.

avg = average number of items to choose at every decision point when building a solution.

ρ = the evaporation rate of the pheromone. Its value is set to 0.7 based on the study made in a related literatures.

τ_{max} = 1/1-0.7

= 3.33

To get P^{best}, we have to get 3P3 = 6,

P^{best} =1/6

=0.166

avg= (1+2+3)/3

=2

$$\tau_{min} = \frac{3.33 \bullet (1 - \sqrt[3]{0.166})}{(2-1) \bullet \sqrt[3]{0.166}}$$

= 2.725

$$\tau_{min} = 2.725 \approx 2.73$$
$$\tau_{max} = 3.333 \approx 3.33$$

Initialization

An example of Process Scheduling Problem (PSP) for 3 processes and 3 resources are given in the table 1 below. The example initialization of pheromone update by the ant of the given PSP is described in table 2. The pheromone update is initialized with the random numbers between τ_{min} and τ_{max}.

Table 1: An Example of 3×3 Process Scheduling Problem

Process	**Task Sequence**	**Processing Time**		
P1	R1, R2, R3	2	1	3
P2	R3, R2, R1	3	1	2
P3	R1, R3, R2	2	3	1

Table 2: Initialization of Pheromone Update for the ith Ant

Task	**1**	**1**	**1**	**2**	**2**	**2**	**3**	**3**	**3**
Pheromone Update	X_{i1}	X_{i2}	X_{i3}	X_{i4}	X_{i5}	X_{i6}	X_{i7}	X_{i8}	X_{i9}
Values	19.10	16.50	18.95	18.00	17.00	19.77	18.65	17.65	16.89

Decoding Ant into Solutions

For decoding ant into a schedule, the first step is to sort the values of the pheromone update vector into ascending order. The next step is to arrange the tasks in the corresponding order as the values of the position vector, sorted in step 1. The resultant sequence of task and the corresponding position values are given in table 3.

Table 3: An Example of Sequenced Ant

Task	**1**	**3**	**2**	**3**	**2**	**3**	**1**	**1**	**2**
Pheromone Update	X_{i2} 16.50 (1)	X_{i9} 16.89 (2)	X_{i5} 17.00 (3)	X_{i8} 17.65 (4)	X_{i4} 18.00 (5)	X_{i7} 18.65 (6)	X_{i3} 18.95 (7)	X_{i1} 19.10 (8)	X_{i6} 19.77 (9)

Thus from Table 3, the operational based permutation π = (1, 3, 2, 3, 2, 3, 1, 1, 2) is obtained. An element of π with i value stands for P_i. The j th occurrence of i in π refers to O_{ij}, that is the j th task (operation) of patient. The precedence of the task is determined simply by the order of the elements of π. considering the precedence constraint of the tasks O11, O_{31}, O_{21}, O_{32}, O_{22}, O_{33}, O_{12}, O_{13}, O_{23}. Ordered tasks from π are ready to be scheduled. Table 4 below gives the Decoded Schedule using ACO.

Consider the operation-based permutation π = (1, 3, 2, 3, 2, 3, 1, 1, 2), the first element in the permutation is 1; therefore the 1st task of the 1st patient is processed on resource R1. In succession, the 2nd element in the permutation is 3, hence the 1st task of the 3rd patient is processed on R1, and then the 3rd element is 2, so the 1st task of the 2nd patient is processed on R3. Thus based on the above description the decode schedule in Table 4 is obtained.

Table 4: Decoded Schedule using ACO

R3	P2				P3					P1					
R2				P2				P3	P1						
R1	P1		P3		P2										
Timeslot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Process Scheduling Algorithm Using Ant Colony Optimization (ACO)

Based on the procedures that were followed in order to come up with the agent-based decoded schedule using the ACO, we present a new algorithm below which can be use in Process Scheduling in a multi-tasking environment using agent-based ACO approach.

Step1: Calculate the value of the τ_{min} and τ_{max}

Step2: Generate values at random between τ_{min} and τ_{max} as the Pheromone update for each of the three tasks of the three patients pi: X_{i1}, Xi_2, X_{i3}, X_{i4}, X_{i5}, X_{i6}, X_{i7}, X_{i8}, X_{i9} in the following order: 1, 1, 1, 2, 2, 2, 3, 3, 3.

Step3: for i ← 0 to 7 do

min ← i

for j ← I +1 to 8 do

if A[j] < A[min] min ← j

Swap A[i] and A[min]

// arranging the values of the pheromone update in ascending Order.

Step 4: The next step is to arrange the tasks in the corresponding order as the values of the pheromone update, sorted in step 3 to get the operational based permutation π.

Step 5: Get the Decoded Schedule based on the Operational Based Permutation where the first element in the permutation i denote the 1st task of pi to be processed.

Conclusion

The proposed agent oriented algorithm can take care of any process scheduling (task assignment) in a multi-tasking environment using ACO approach. The ACO Algorithm can easily be implemented using any of the agent development frameworks such as Java Agent Development framework (JADE), JADEX etc.

CORRESPONDING AUTHOR: M. K Ahmed; Email: kabirmka@yahoo.com;

Contact: +2348039285289

References

[1] Bhat, S., Sidnal, N. S., Malashetty, R. S. and Manvi,S. S. (2011). Intelligent Scheduling in Health Care Domain: *International Journal of Computer Science Issues, 3(5), No 3.* pp 214-224, vol. 8

[2] Blum, C. (2011). Ant Colony Optimization: Introduction and Recent Trends: *Science Direct*, *Physics of Life Review 353-373*

[3] Chen, H. and Cheng, A. M. K. (2005). Applying Ant Colony Optimization to the Partitioned Scheduling Problem for Heterogeneous Multiprocessors: *SIGBED Review 2(2):11-14*

[4] Dorigo, M. and Stutzle, T. (2004). Ant Colony Optimization*. London: A Bradford Book, The MIT Press, Cambridge, Massachusetts.*

[5] Stützle, T. and Hoosb, H. H. (2000). MAX–MIN Ant System: *Future Generation Computer Systems, 16 (2000) 889–914.*

[6] Umarani, S. G., Maheswari, V. U., Shanthi, P. and Siromoney, A. (2012). Tasks Scheduling using Ant Colony Optimization. *Journal of Computer Science 8 (8): 1314-1320*.

[7] Moreno, A. (2003). Medical Applications of Multi-Agent Systems. *CiteSeerX*.

[8] Wooldridge, J. M. (1999). *Intelligent agents, Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence.* Wiley & Sons.

[9] Wooldridge, M. (2002). *An Introduction to Multi-Agent Systems.* Wiley & Sons.