خوارزمية OAPJ لتحسين الوظائف المجدولة في المتجر

د.ندى الحكاك
قسم علوم الحاسبات
كلية بغداد للعلوم الاقتصادية الجامعة
dr.nada@baghdadcollege.edu.iq
nadahakkak@hotmail.com

المستخلص

احدى المواضيع البحثية المهمة في مجال نظم التشغيل هي الوظائف المجدولة في المتجر. هذا البحث يقدم خوارزمية OAPJ لتحسين العمل لتحسين بيئة عمل وظائف جدولة المتجرمع ضمان افضل استغلال لاجهزة هذه البيئة. تم اختبار الخوارزمية من خلال بيئة محاكاة للوظائف او العمليات التي لها فترات عمل مختلفة تم انشاءها عشوائيا لكي تنفذ على اجهزة او ماكنات عمل في بيئة مصنع افتراضية. الخوارزمية المقترحة استخدمت تقنية مستعمرة النمل المثلى لاتخاذ القرار عند تنقل العمليات من ماكنة عمل الى اخرى مع اعتماد قيمة الفترة الكلية لانهاء العمل كأداة قياس لكفاءة العمل، وتم استخدام تقنيتي مبدأ تفضيل العمل الاقصر و دائرة روبن مع معدل قيمة الشريحة الزمنية كخوارزميات جدولة. ولقد اعطت تقنية العمل الاقصر اولا وقت بقاء اقل للعمليات.

كلمات مفتاحية: جدولة الوظائف في المتجر، تقنية مستعمرة النمل المثلى، اداة قياس كفاءة العمل، العمل الاقصر اولا والواصل اولا، الوظيفة الاولوية النملة، جدولة دائرة روبن، شريحة زمنية.

Abstract

One of the interested research topics in the field of operating systems is Job Shop Scheduling. This paper produced Optimized Ant Preempted Job (OAPJ) algorithm to enhance the work of job shop environment in order to get better utilization for the machines. OAPJ algorithm is tested over a simulated environment for processes with burst time that are generated randomly. The simulated environment represents a factory environment with multiple machines as nodes and the products are the processes. Ant Colony Optimization ACO algorithm was used for process movement across the machines with fitness function of minimizing the makespan time, meanwhile on each machine, Preempted Shortest Job First SJF and Round Robin RR with mean time slice value, were used to manage scheduling on those machines. The preempted Shortest Job First had better results than Round Robin.

Keywords: Job Shop Scheduling, Makespan, Ant Colony Optimization, Fitness, Preempted Shortest Job First, Ant Preempted Job, Round Robin.

Introduction

Job-Shop Scheduling Problem (JSP) is an NP complete problem (Non-Deterministic Polynomial time complete). JSP deals with machine's allocation to a specific operation, with the effect of time interval in order to minimize the total time for job-processing and to avoid executing two jobs at the same time and on the same machine (Wickramasinghe & W.R.M.U.K., ٢٠٠٣).

ACO is a metahuristic search approach; it is used for sloving multiple types of NP-hard problems, like classical JSP, and others (Mahdavinejad & R.A., 2010).

This paper proposed (OAPJ) algorithm to manage and control the work of job shop scheduling problem environment by using scheduling algorithm to manage work on each node on the network graph that constructed upon the ACO with fitness function as minimum waiting time. Hence, the main objective of this paper is to minimize the total time where process or product can spent in a system or factory that contains multiple machines.

However, most papers and previous work related to JSP problems used ACO to arrange the machines with simple scheduling problems, but did not use preempted SJF scheduling algorithm and compare it with another scheduling algorithm like Round Robin scheduling algorithm with time slice value calculated for the average of execution times of jobs on a certain machine.

This paper is organized as follows: first review about ACO and JSP environment, second related work, third the proposed idea of APJ-algorithm with an example for implementing the algorithm, fourth conclusion with future work, finally references.

Related Work

Many research have been done in the area of JSP, some of them constraint on clustering the machines according to specific features and depending on ACO algorithm with multiple tests on changing the ACO parameters to get better results, on the other side some others worked on the scheduling part for each machine, as we can see in the following part of related work section.

Mainly, (Kaur and kaur, 2015) used ACO technique on JSP environment without any optimization to minimize makespan.

(Rui et al 2014) proposed an idea of changing the pheromone value of certain paths, after each cycle by using scheduler builder on each solution. It was done after a process of computing multiple values and selects the best one. However, they used the (AS-JJS) Ant system to search on specific area of the graph, in order to get better results.

Moreover, (Ponnambalam et al 2014) connected the work between FJSP and ACO based heuristic. And (You-Xin, M et al 2009) presented multiple types of scheduling problems, like; multiprocessor scheduling, JSP, and other types with applying ACO on those problems.

Furthermore, (Mahdavinejad & R.A., 2010) presented a new technique for calculating makespan values; by working on a single-processor JSP environment with an optimized ACO algorithm based on priority technique. The proposed idea was a hybrid technique from SA, GA, and ACO algorithms.

However, they clustered the machines according to their functions, and for each process there was two times; processing setup time, and processing time. All allocations are affected by machine's priority.

Moreover, (Huang et al 2008) proposed a new algorithm for combining ACO with fast taboo in order to minimize makespan in JSP, they optimized their work by defining their own pheromone trail with the help of dynamic greedy heuristic method.

Furthermore, (Omkumar et al 2008) implemented traditional JSP and compared the results with several dispatching rules by using multiple values of ACO algorithm to minimize makespan time. While, (Zhang et al 2006) applied ACS on JSP with multiple values; β, α, and ρ also they added two extra nodes to the graph. They called them dummy-start and dummy-end to make boundaries for job finishing.

Finally, (Montgomery et al 2006) proposed a new approach for computing the operation's priority with the help of applying dispatching rules on each machine. They integrated work with fuzzy processing.

Generally, no work has been done to cover the best utilization of JSP machines, so there is a probability of being idle. Hence, this work used ACO for better rout and two types of scheduling algorithms in order to get better results in machines' utilization. In the same time the products should spend the minimum time in the system.

Scheduling algorithms

One of the main factors that affect operating systems performance is CPU utilization. This can be achieved by using a scheduling algorithm that prevent the CPU from being idle, in other words allocating all available resources in best way. This paper used Shortest Job First (SJF) algorithm and Round Robin scheduling algorithms to be built with ACO work, but in Round Robin the time slice is calculated as we previously discusses in (Jarrah and Hakkak, 2014). The following are two examples of how preempted SJF and RR work.

Example1 (preempted SJF)

Process	Burst	Arrival
P_2	2	0
P_7	4	2
P_4	2	6

2	4	2	
P_2	P_7	P_4	
0	2	6	8

$W\text{-}P_2=0-0=0$

$W\text{-}P_7=2-2=2$

$W\text{-}P_4=6-6=0$

<u>Example2 (RR with mean time slice)</u>

Process	Burst	Arrival
P_2	2	0
P_7	4	2
P_4	2	6

2	2	2	2	
P_2	P_7	P_4	P_7	
0	2	4	6	8

Q or time slice=8/3=2

$W\text{-}P_2=0-0=0$

$W\text{-}P_7=(2-2)+(6-4)=2$

$W\text{-}P_4=6-6=0$

Ant Colony Optimization (ACO)

A metaheuristic is a feature of multiple types of algorithms; those algorithms are be used with heuristic concepts and applied on different kinds of problems (Darquennes, 2005). As an example, we will constraint on ACO. It's one type of metaheuristic search method, which is inspired by real ant's behavior (Aggarwal et al, 2012).

However, ACO also known as "Ants System"; were proposed first for solving traveling salesman problem. Mainly, the area of cities is represented as a graph or network of nodes. While traveling ants build new solution on that graph, were the graph's edges contain two types of information's that help ants in its decision of move. Hence, there should be some pre-work or pre processing to should be done for information's gathering about the graph; also there will be another kind of information's to be modified during the work of the algorithm. One of the main factors that affect the performance is the number of ants that followed the path's pheromone (Liu et al, 2012).

However, when ants walk between source-node and destination-node it uses the pheromone from its tail to mark the path as the best choice. This pheromone represents a long-term memory about the ant's searching process that is updated by the same ant. Moreover, when arriving into path intersection, where decision should be made to follow the best path; the ant should decide after applying some steps for probabilistic decision depending on the amount of pheromone. Finally, the best path or the shortest path will be marked with the greatest amount of pheromone (Alonso et al, 2005).

Hence the ants are blind; the pheromone is the main communication tool between the ants, so they would be guided by the strength of path's pheromone.

Furthermore, there are two types of pheromone's deposit; those are:

١- Online step by step pheromone update; direct deposit after each move.

٢- Online delayed pheromone update; one update is made after end of solution (Yu & H., 2014) and (Contreras et al, 2008).

Ant Colony optimization algorithm may be summarized as follow:

١- Initialize ACO parameters (Q, τ(0), α, β, ρ), t-iteration, i-source city, J-destination city, k-ant index.
   Where $\eta_{ij}=1/d_{ij}$ …….. Equation(1)

٢- Loop
   a. Randomly Position num_ant ants on num-city cities.
   b. For step=1 to num-city
      i. For k=1 to num_ant
         ١. $P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\propto}.[\eta_{ij}(t)]^{\beta}}{\sum_{h \epsilon j_i^k}[\tau_{ij}(t)]^{\propto}.[\eta_{ij}(t)]^{\beta}}$ ……. Equation(2)

      End for

ii. If $R_n < P_{ih}$ then move to town h
Else repeat R_n and compare again

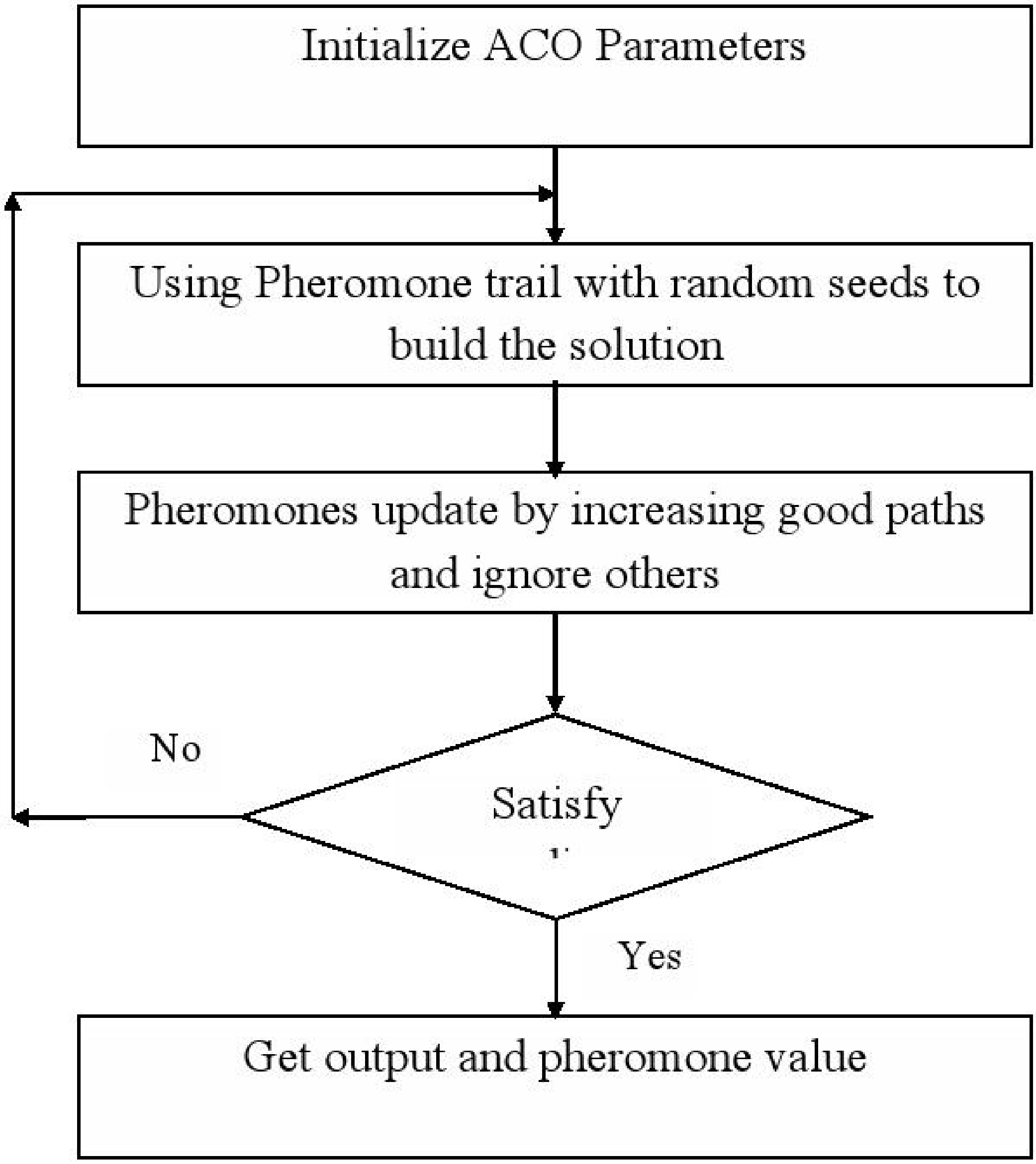
iii. $\Delta\tau_{ij}(t) = \frac{Q}{L_K}$ ……….. Equation (3)

End for

c. $\tau_{ij}(t + ١) = (١ - \rho).\tau_{ij}(t) + \Delta\tau_{ij}(t)$ ……….Equation (4)

٣- Until end condition (Selvi & Umarani, 2012).

The following figure represents the searching process within ACO method (Jia &H., 2015):



<u>Figure (1):</u> Ant Colony Optimization ACO process

Job shop Scheduling Problem (JSP)

When having multiple resources with a specific time to do some tasks, scheduling is the best choice as a solution to this problem (Błażewicz et al, 1996). Those kinds of problems should be represented in a graphic or network way, which contains nodes to represents the resources (or operations) (Bertsimas et al, 1999).

For any process there is a process planning in which certain machine type is assigned to certain process. Hence, we need an integration method to connect scheduling with process planning to optimize processing and avoid bottleneck problems. All this can be done when having alternative machines and routings. Job shop needs scheduling algorithms to manage the multiple copies of some machines (or resources). And some of those machines may become busy for long terms (Aggoune & R., 2004).

In JSP world, scheduling represents some multiple stages that jobs or product should go through, but it would be bounded by start-time and end-time to get the final task in complete way (Moghaddas & Houshm, 2008).

Proposed Idea

Before discussing our proposed idea, some assumptions should be reviewed first.

- Some critical machines would appear with multiple copies, to prevent delay.
- In order to get best performance we need an integration between process ‘operation and scheduling to manage those operations, with the availability of multiple routes, to avoid congestion.
- The benefit of multiple routes is to enhance machine’s utilization, to prevent idle problem.
- There would be a number of machines (M), with multiple jobs (N), as (J_1 to J_n), where each job or product needs multiple operations on different machines, in order to complete its processing.
- There is no job priority, instead job preemption is allowed in scheduling. Also it compared with round robin method.
- The main objective is to complete all operations with the minimum makespan time; this is also the fitness function of the heuristic method.

Optimized Ant Preempted Job (OAPJ) Algorithm, main steps

Part-1 (Request to access):

١- Request for graph type (for example ACO) to represent the JSP problem and number of nodes (or machines) to build the environment.

٢- Request the scheduling type on the nodes (for example preempted shortest job first, and Round Robin), or it may be as multi layer.

Part-2 (ACO + preempted SJF or RR with mean time slice):

١- Draw ACO graph or network for JSP problem and identify the machines of each node with describing the alternative machines, and the weight values of edges.

٢- Initialize the jobs or products, and the machines they need proceed in order to complete their work.

٣- Apply ACO algorithm to move jobs or products, between nodes or machines.

٤- For scheduling, on each node (or machine) apply preempted SJF algorithm, or apply Round Robin algorithm, results comparison.

٥- Update Pheromone values on edges, for future choose of routes.

٦- If not in destination, then go to step-2 part 3 to re-apply ACO with ignoring visited nodes or machines, and in case of "equal nodes functions" they should be treated as routers. Else, if its destination then apply scheduling algorithm on the destination node.

Part-3 (Makespan Calculation):

For each job calculate the Makespan time value; as follows:

$$\text{Makespan} = \Sigma \text{ waiting time} + \Sigma \text{ Processing time} + \Sigma \text{ Path time}$$

The following figure, figure (2), represents the proposed idea:

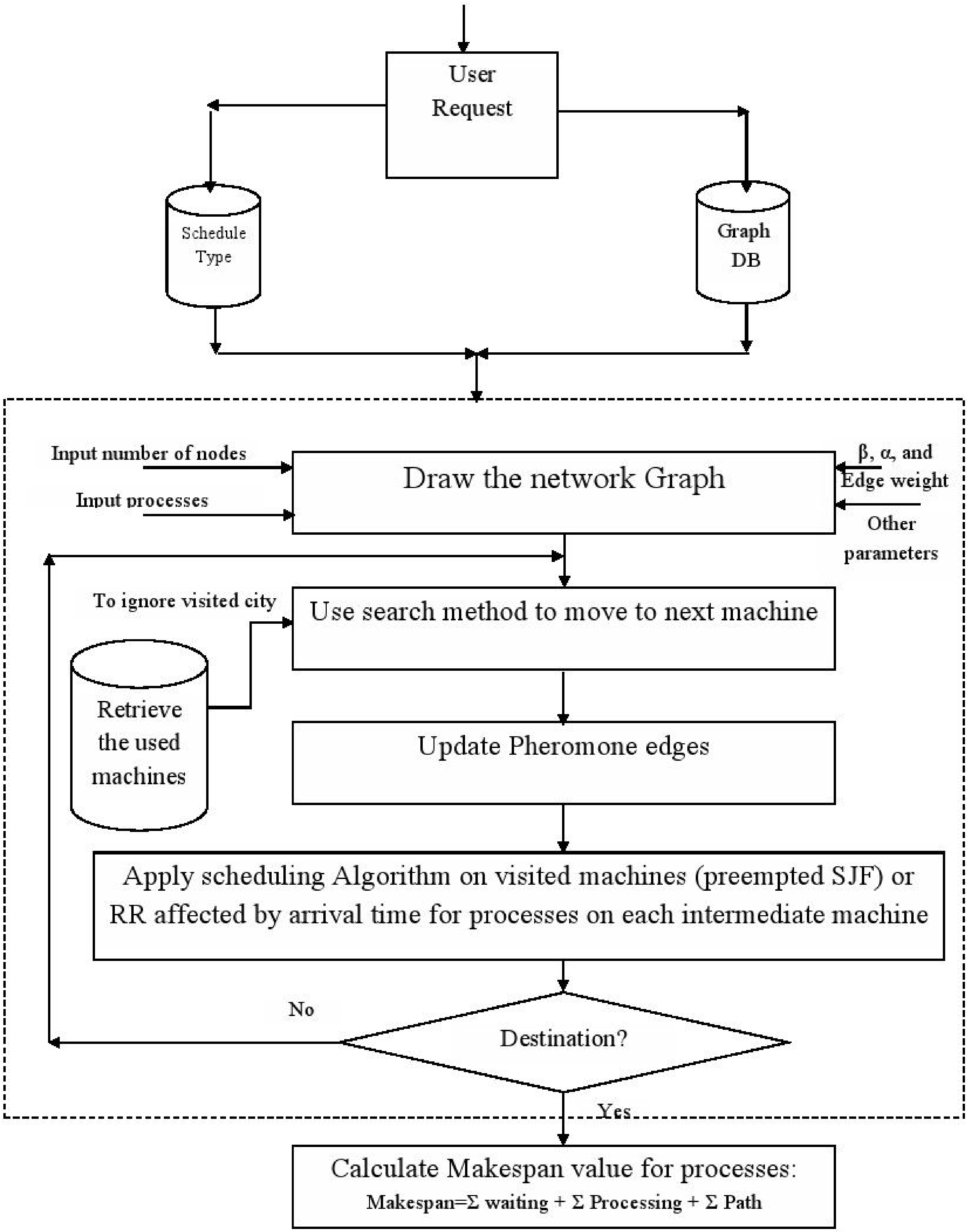


Figure (2): OAPJ algorithm

Example:

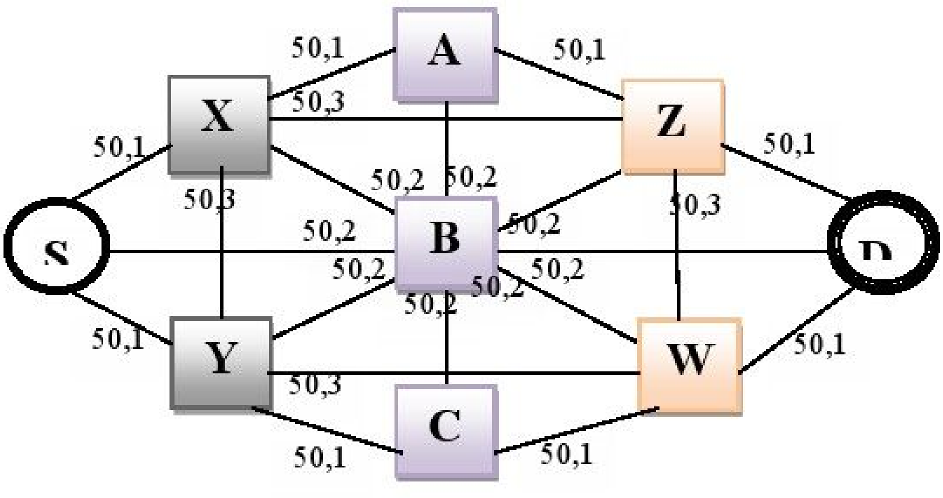


Figure (3): Machine's network before processing

Assumptions:

- Q=50, T_0=1, α=1.5, β=2, and ρ=0.2
- Alternative nodes (equal in their functions), are: (A,B,C), (X,Y), and (Z,W) ; all interior nodes are bounded by start-node (S), and end-node (D).
- Some ants may prefer executing nodes than routing nodes.

The following is the graph, figure (4), is a result of implementing figure (3) with OAPJ algorithm:

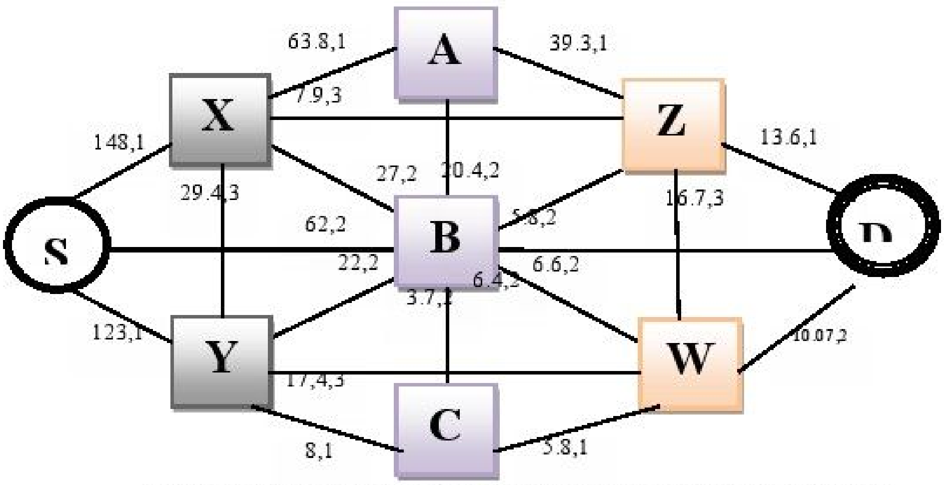


Figure (4): Machine's network after processing

The following table, table (1), contains all the values for implementing the proposed algorithm (OAPJ). There were ten different types of products called as processes, with different burst time for each process. The table also contains the calculated waiting time using two different scheduling algorithms and path distance with final makespan value.

Mainly OAJP was implemented in C# environment with the use of XML files as database to store pre used information to improve OAPJ-algorithm.

P-name	**P-Burst**	**P-waiting with SJF-preempted With non zero arrival time**	**P-waiting with Round Robin (q=average of bursts) With non-zero arrival time RR_2**	**P-path**	**P-makespan With SJF-preempted**	**P-makespan With Round Robin (q=average of bursts)**
P_1	7	8	4	4	(7+8+4)=19	(7+4+4)=15
P_2	11	2	2	7	20	20
P_3	16	8	5	8	32	29
P_4	21	20	25	6	47	52
P_5	26	33	39	6	65	71
P_6	6	7	17	5	18	28
P_7	11	1	9	6	18	26
P_8	16	23	23	7	46	46
P_9	21	25	37	7	53	65
P_{10}	26	24	30	4	54	60

Table (1): Example's values and results of OAJP algorithm with preempted SJF and RR

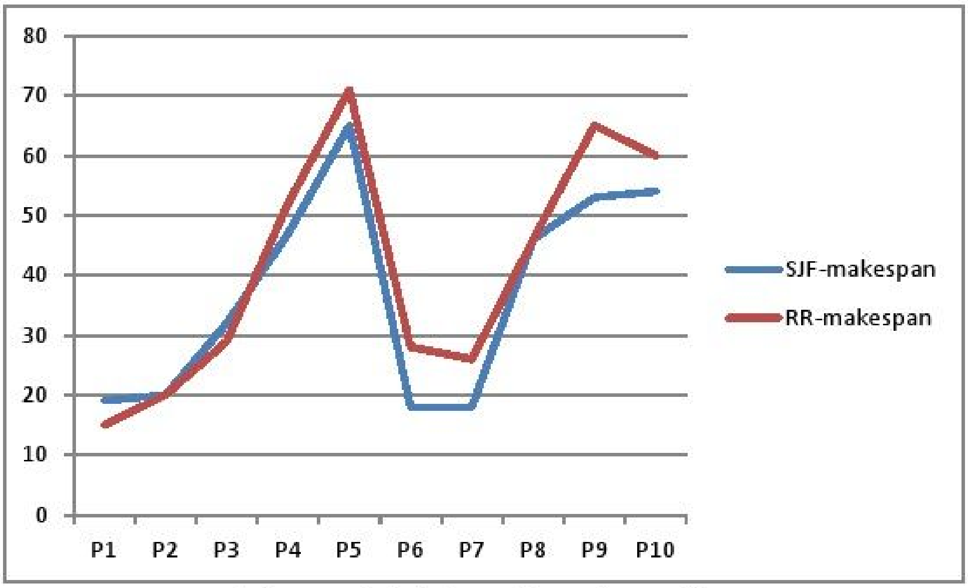


Figure (٥): Experiment results

However, figure (5) shows makespan value when using each type of scheduling algorithms. Generally SJF gave better results than RR.

Conclusion

Machines unavailability is a huge problem for manufacturing systems; hence many researchers were interested in the job shop scheduling problem.

This paper is interested in minimizing total spent time for jobs in those manufacturing systems; which denoted by makespan time value, with best utilization for the machines

The paper proposed OAPJ algorithm that constrained on enhancing JSP environment; which is constructed using ACO and scheduled with preempted SJF and RR scheduling algorithms, in order to minimize makespan value.

However, the results showed in figure (5) that applying preempted SJF scheduling algorithm gives better results, less makespam value, than RR algorithm in ACO environment.

Future Work

١. In order to reduce network overhead, we should use threshold limitation on nodes or machines to get different alternative routes.

٢. Use the same simulation area to implement the ACO on the same network with modifying its parameters; β, α, and ρ. Because β, α values affects the probability equation of ACO; while ρ affect the updated pheromone equation.

References

[1] Aggarwal, M. and Saroj, D., 2012. Compute travelling salesman problem using ant colony optimization. *International Journal of Computing and Business Research ISSN (Online): 2229-6166 Proceedings of I-Society 2012 at GKU.*

[2] Aggoune, R., 2004, June. Two-Job Shop Scheduling Problems with Availability Constraints. In *ICAPS* (pp. 253-259).

[3] Al Jarrah U. and Al Hakkak N., 2014, Time Slice Optimization in Round Robin Scheduling Algorithm, Conference, Madent Al-Elem University College.

[4] Alonso, S., Cordón, O., De Viana, I.F. and Herrera, F., 2005. Integrating evolutionary computation components in ant colony optimization. *Recent Developments in Biologically Inspired Computing*, p.148.

[5] Bertsimas, D. and Gamarnik, D., 1999. Asymptotically optimal algorithms for job shop scheduling and packet routing. *Journal of Algorithms*, *33*(2), pp.296-318.

[6] Błażewicz, J., Domschke, W. and Pesch, E., 1996. The job shop scheduling problem: Conventional and new solution techniques. *European journal of operational research*, *93*(1), pp.1-33.

[7] Contreras, M.A., Chung, W. and Jones, G., 2008. Applying ant colony optimization metaheuristic to solve forest transportation planning problems with side constraints. *Canadian Journal of Forest Research*, *38*(11), pp.2896-2910.

[8] Darquennes, D., 2005. Implementation and Applications of Ant Colony Algorithms. *M. Sc. Tesis, University of Namur, Belgium.*

[9] Huang, K.L. and Liao, C.J., 2008. Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers & operations research*, *35*(4), pp.1030-1046.

[10] Jia, H., 2015. A novel hybrid optimization algorithm and its application in solving complex problem. *International Journal of Hybrid Information Technology*, *8*(2), pp.1-10.

[11] Kaur G. and kaur S., 2015, Job Shop Problem To Minimize Makespan Time With Ant Colony Optimization Approach, Journal Australian Journal of Information technology and Communication, Volume II, Issue I, ISSN 2203-2843.

[12] Liu, F., 2012. A dual population parallel ant colony optimization algorithm for solving the traveling salesman problem. *Journal of Convergence Information Technology*, *7*(5), pp.66-74.

[13] Mahdavinejad, R.A., 2010. A new approach to job shop-scheduling problem. *Journal of Achievements in Materials and Manufacturing Engineering*, *41*(1-2), pp.200-206.

[14] Moghaddas, R. and Houshm, M., 2008. Job-shop scheduling problem with sequence dependent setup times.

[15] Montgomery, J., Fayad, C. and Petrovic, S., 2006, September. Solution representation for job shop scheduling problems in ant colony optimisation. In *International Workshop on Ant Colony Optimization and Swarm Intelligence* (pp. 484-491). Springer Berlin Heidelberg.

[16] Omkumar, M. and Shahabudeen, P., 2008. Ant colony optimization for multilevel assembly job shop scheduling. *International Journal of Applied Management and Technology*, *6*(2).

[17] Ponnambalam, S.G., Jawahar, N. and Girish, B.S., 2014. An ant colony optimization algorithm for flexible job shop scheduling problem. New Advanced Technologies, 2010.

[18] Rui, Z., Shilong, W., Zheqi, Z. and Lili, Y., 2014. An ant colony algorithm for job shop scheduling problem with tool flow. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *228*(8), pp.959-968.

[19] Selvi V. and Umarani R., 2012, An Ant Colony Optimization For Job Scheduling To Minimize Makespan Time, International Journal of Computer & Communication Technology (IJCCT) ISSN (ONLINE): 2231 - 0371 ISSN (PRINT): 0975 –7449 Vol-3, Iss-3.

[20] Wickramasinghe, W.R.M.U.K., 2003. *Hybrid Ant Colonies for Parallel Algorithms* (Doctoral dissertation, University of Colombo).

[21] You-Xin, M., Jie, Z. and Zhuo, C., 2009, December. An overview of ant colony optimization algorithm and its application on production scheduling. In *Innovation Management, 2009. ICIM'09. International Conference on* (pp. 135-138). IEEE.

[23] Zhang, J., Hu, X., Tan, X., Zhong, J.H. and Huang, Q., 2006. Implementation of an ant colony optimization technique for job shop scheduling problem. *Transactions of the Institute of Measurement and Control*, *28*(1), pp.93-108.