International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:6, No:12, 2012

Jobs Scheduling and Worker Assignment Problem to Minimize Makespan using Ant Colony Optimization Metaheuristic

Mian Tahir Aftab, Muhammad Umer, and Riaz Ahmad

***Abstract*—**This article proposes an Ant Colony Optimization (ACO) metaheuristic to minimize total makespan for scheduling a set of jobs and assign workers for uniformly related parallel machines. An algorithm based on ACO has been developed and coded on a computer program Matlab®, to solve this problem. The paper explains various steps to apply Ant Colony approach to the problem of minimizing makespan for the worker assignment & jobs scheduling problem in a parallel machine model and is aimed at evaluating the strength of ACO as compared to other conventional approaches. One data set containing 100 problems (12 Jobs, 03 machines and 10 workers) which is available on internet, has been taken and solved through this ACO algorithm. The results of our ACO based algorithm has shown drastically improved results, especially, in terms of negligible computational effort of CPU, to reach the optimal solution. In our case, the time taken to solve all 100 problems is even lesser than the average time taken to solve one problem in the data set by other conventional approaches like GA algorithm and SPT-A/LMC heuristics.

***Keywords*—**Ant Colony Optimization (ACO), Genetic algorithms (GA), Makespan, SPT-A/LMC heuristic.

I. Introduction

THE management of available resources is always a key decision with an aim to maximize or minimize certain parameters like maximizing utilization of the machines and minimizing total flow time or makespan of the tasks. The problems of scheduling jobs onto machines have been researched by number of scholars as this area further leads to the solution of number of other identical problems being faced in the industry. The selected problem is quite significant as it is close to the real world problems as well as it is important from theoretical point of view. Makespan is the total length of the schedule (that is, when all the jobs have gone thru their processing) and is the objective function in our case with an aim to minimize it using an *Ant Colony Optimization approach.*

Mian Tahir Aftab is with National University of Sciences & Technology (NUST), Islamabad, Pakistan at its School of Mechanical & Manufacturing Engineering (SMME), (phone: +92-321-6003013; e-mail: seoccsf7@hotmail.com).

Muhammad Umer is with National University of Sciences & Technology (NUST), Islamabad, Pakistan at its School of Mechanical & Manufacturing Engineering (SMME) as a student.

Riaz Ahmad is also is with National University of Sciences & Technology (NUST), Islamabad, Pakistan at its School of Mechanical & Manufacturing Engineering (SMME).

II. Literature Review

A. Job Shop Scheduling and Worker Assignment

The problem of scheduling jobs on the machines has been quite famous amongst the researchers [1]-[3]. In this regard, the issue of minimizing the total time for completion of all jobs (makespan) is especially of great significance as it has been proven to be an NP-hard with even for the case of two parallel machines [4]. Similarly, the problems of worker assignment have also drawn significant attention of the researchers in the past. It has been learnt that in most of the cases, the number of workers are ignored and are not generally taken into account while scheduling the jobs. However, in number of practical situations, if the number of workers is increased or decreased, it significantly decreases or increases the overall makespan. Hence, in all such cases, the number of workers is to be taken as another variable which affect the objective function of the problem. Po-Chienge Hu [5]-[6] made a significant effort and established a mathematical relationship between the jobs, machines and workers. He solved these sets of problems for the parallel machines model with an aim to minimize different objective functions like total tardiness and total flow time. In this regard, a heuristic of shortest processing time (SPT) and a procedure of largest marginal contribution (LMC) were employed for solving the problems. Subsequently, Chaudhary and Drake [9]-[10]-[11] developed a Genetic Algorithm (GA) and solved the same set of problems with an aim to minimize total flow time, total tardiness and total makespan.

B. Ant Colony Optimization

The metaheuristic named Ant Colony Optimization was developed back in 1991 [13] by taking an inspiration from the working of natural ants. It was observed that a colony of real ants while searching for their food, initially follow different tracks to reach the source of food. The ants which take the shortest route, will obviously reach the destination in minimum time as compared to other ants which followed the longer tracks. During this travelling, all ants leave a trail of a biological material named pheromone. This pheromone plays a vital role in the ant colonies as it is medium thru which they indirectly communicate with each other and guide their colleagues to follow a particular route. Higher the concentration of pheromone on a track, there will be a higher probability to follow that route by the following ants. Additionally, this pheromone evaporates after some time if more quantities are not added to it. Now, while the first batch of ants start their journey from food source to their nest, it again follows different routes with a slightly higher probability of following the same earlier shortest route due mild concentration of the pheromone on this track. This cycle goes on and after some more iterations, it has been proven thru a number of real time experiments that all ants converged to the shortest route. Getting an inspiration from this biological aspect for the real ants, an algorithm based on artificial ants was developed and named as Ant Colony Optimization (ACO) to solve a wide range of optimization problems in the academia as well as in the real world. In this algorithm, initially one ant constructs a random solution and subsequently, the artificial pheromone is either updated for good quality solutions or evaporated for inferior solutions [16]. Another aspect which is essentially required to guide the ACO algorithm is the Heuristic information to produce good quality solutions. This heuristic information about the problem helps in adding remaining components of the solution to the already partial solution. The heuristic information is based on the knowledge of the problem and cannot be amended till end of the solution process.

III. Problem Formulation

A. Hypothesis

In classical parallel machine models, the number of workers are mostly ignored and not taken into account. However, in some practical situations, the number of workers doe's effect the completion time of a job, thus, adding more workers to the machine, reduces the completion time. As explained earlier, there can be a mathematical relationship between the job processing time and the number of workers assigned to work on the job. In the conventional scheduling problem for parallel machines, generally, there have been efforts to address two critical issues:

1. Screening jobs at the machines, and then 2. Sequencing these jobs on the machine

But in this research, we need to address two challenges :

1. How the jobs are to be scheduled onto the machines, and
2. How to deploy workers on these machines?

Our aim is to minimize the makespan which is the objective function and is defined as total time to complete all the jobs. As a goal of this research, an algorithm based on a metaheuristic named as Ant Colony Optimization (ACO) has been developed to solve the same set of 100 problems [12]. Subsequently, the performance of ACO has been compared with the earlier approaches like GA [9]-[10]-[11] and with SPT-A/LMC heuristic approach [5]-[6]. In order to gauge the performance of our ACO algorithm and as a litmus test, two parameters have been selected which are quality of the solution obtained and the calculation effort (processing time of the CPU) to reach these solutions.

B. Notations of the Problem

As explained earlier, un-like typical scheduling problems, in this particular case, an additional variable of workers is also to be taken into account while scheduling the jobs. Some notations that will be used are as given below :

a) $A_i, B_i \text{ and } E_i$ are the integers which follow uniform distribution, such that $0 \leq A_i < 10$, $0 \leq B_i < 800$, $1 \leq E_i < 10$
b) n = number of jobs waiting to be processed
c) m = number of parallel machine in the shop
d) P_i= process time of job i
e) W_i = number of workers assigned to machine j

C. Assumptions for the Problem

n jobs and W workers are to be scheduled and assigned to m parallel machines with following assumptions in place :

a) Every job is to go thru only one processing operation.
b) Each job takes different times on different machines and is also affected by the number of workers assigned to that machine.
c) Splitting or pre-emption of the job is not allowed;
d) Aim is to minimize the makespan, being an objective function.
e) Any job can be processed on any machine.
f) There is negligible time to set up the machines and can be ignored.
g) At any given time, there will be no more than one job under process on a machine.
h) There is negligible time for transportation of the jobs between machines.
i) The number of machines, jobs and workers is constant and shall not change during processing time.
j) All W workers have identical competence while working on any combination of the job/ machine.
k) All n jobs, W workers and m machines are ready at time T_0.
l) The simplest form of the processing time function is:

$$P_i(W_j) = A_i + B_i/(E_i * W_j)$$

where $P_i(W_j)$ is the processing time of job i that is processed on machine j in which the number of W_j workers are assigned on it, A_i is a fixed constant and not affected by the number of workers, $B_i/(E_i * W_j)$ is a variable portion and depends on the number of workers.

D. General Structure of the Proposed ACO Algorithm

In our particular case, we have to route 12 jobs thru 3 uniformly related parallel machines as well as we have to allocate 10 workers to these 3 machines in such a way that the total time to complete all the 12 jobs is minimum. General structure of the proposed ACO is as given below:

Part-I (Scheduling of Jobs)

Initiate trails of the pheromone and parameters. Set n as the set of schedulable jobs for m machines and w as the available/ assignable workers.

While (termination condition is not met) *do*

International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:6, No:12, 2012

For each ant in the colony *do*
Apply local search.
For ($n = 2,3,\ldots,12$) *do*
Spot all un-scheduled jobs and embrace them in the partial solution.
Update the pheromone values in the local iterations loops based on the partial solutions recorded up till now.
end for
Apply local search algorithm to the iteration best solution and in case of improvement, update the best solution found so far.
Apply global updating rule to the best solution found so far or the iteration best solution.

Part-II (Assigning the workers)
Once all jobs have been assigned, then further improve the globally constructed solutions by assigning remaining 9 workers so as to further minimize the processing timepi.
Again apply global updating rule to the best solution discovered so far.
end for
end while

IV. Analysis of the Results

With an aim to gauge the performance of our Ant Colony Optimization (ACO) based algorithm as compared to the two mentioned conventional approaches, following two yard sticks have been selected:

a) Computational Effort of the CPU i,e, time taken to solve all 100 problems by different approaches; and
b) Average deviation from the ideal solution (accrued from exhaustive search method)

The above-mentioned ACO algorithm was coded in a computer application named Matlab® and applied on all 100 problems. The results and screen shot of the program are given in Table I and screen shots of a problem solution computed by MatLab® is given in Fig. 1.

TABLE I
Comparison of the Results Accrued from ACO with the Results of Conventional Approaches

S No	Type of approach	Processor's speed	Time taken to solve all 100 problems	Average Error from Exhaustive Search method)
1	Exhaustive Search Method (all 4842288 possible options are to be explored)	P-IV, 2.0 GHz	680397 seconds (8 days)	-
2	GA	P1V 1.7 GHz	7469 seconds (2 hrs)	0.20%
3	SPT-A/ LMC	P-IV, 2.0 GHz	300 seconds (5 mins)	5.84%
4	ACO	P-IV, 1.7 GHz	0.16 seconds	3.40%

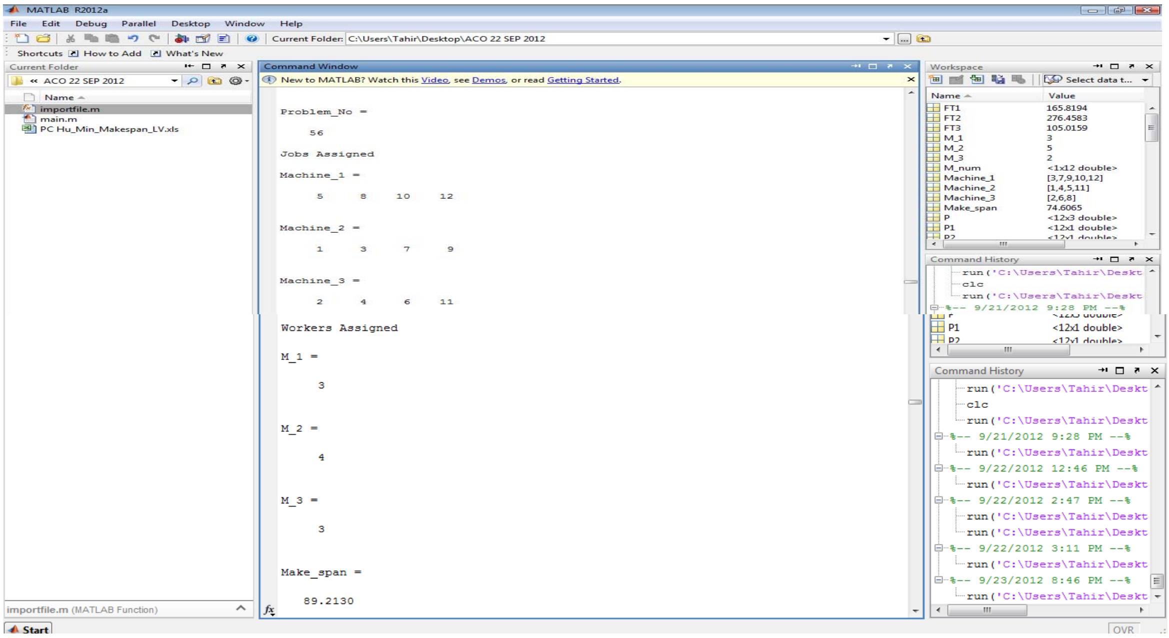

Fig. 1 Screen shot of MatLab® showing the result of Problem No 56 (as a sample problem)

V. Conclusion

The results accrued from this ACO algorithm shows that it is an efficient metaheuristic to solve a wide range of NP-hard combinatorial optimization problems like the one solved in this article as compared to conventional approaches like GA etc. However, it depends on the comprehension of the designer how the problem is comprehended and the way an algorithm is developed. For future research, other variants of the ACO like Elitist ant system, Rank-based ant system, Max-Min ant system, Ant colony system etc shall be investigated. In this regards, all these extensions of the ACO shall be applied on one bench marked/ known NP-hard combinatorial optimization problem and then the results shall be compared to comment on the strengths & weaknesses of each variant of the ACO.

References

[1] B. Chen, C. N. Potts, G. J. Woeginger, "A review of machine scheduling: Complexity, algorithms and approximability", in: D. Z. Du, P. M. Pardalos (Eds.), *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, Dordrecht, 1998, pp. 21–169.

[2] T. C. E. Cheng, C. C. S. Sin, "A state-of-the-art review of parallel-machine scheduling research", *European Journal of Operational Research*, vol. 47, 1990, pp. 271-292.

[3] E. Mokotoff, "Parallel machine scheduling problems: A survey", *Asia-Pacific Journal of Operational Research*, vol. 18, 2001, pp. 193-242.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[5] P-C. Hu, "Minimizing total tardiness for the worker assignment scheduling problem in identical parallel-machine models", *International Journal of Advanced Manufacturing Technology*, vol. 23(5–6), 2004, pp. 383-388.

[6] P-C. Hu, "Minimizing total flow time for the worker assignment scheduling problem in the identical parallel-machine models", *International Journal of Advanced Manufacturing Technology*, vol. 25(9–10), 2005, pp. 1046-1052.

[7] P-C. Hu, "Further study of minimizing total tardiness for the worker assignment scheduling problem in the identical parallel machine models", *International Journal of Advanced Manufacturing Technology*, vol. 29, 2006, pp. 165-169.

[8] P-C. Hu, "Further study of minimizing total flow time for the worker assignment scheduling problem in the identical parallel machine models" *International Journal of Advanced Manufacturing Technology*, vol. 29(7–8), 2006, pp. 753-757.

[9] I. A. Chaudhary and P. R. Drake, "Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms", I*nternational Journal of Advanced Manufacturing Technology*, vol. 42, 2008, pp. 581-594.

[10] I. A. Chaudhary, "Minimizing flow time for the worker assignment problem in identical parallel machine models using GA", *International Journal of Advanced Manufacturing Technology*, 2009, DOI 10.1007/s00170-009-2323-1.

[11] I. A. Chaudhary, "Minimizing makespan for the worker assignment problem in identical parallel machine models using GA", Proceedings of the World Congress on Engineering 2010 Vol III WCE 2010, June 30 - July 2, 2010, London, U.K.

[12] P-C. Hu. Online research data set available at http://sparc.nfu.edu.tw/~pchu/research_data.htm.

[13] Marco Dorigo and Krzysztof Socha, "An Introduction to Ant Colony Optimization", Technical Report No. TR/IRIDIA/2006-010, April 2006, Last revision: April 2007.

[14] Jun Zhang, Xiaomin Hu, X. Tan, J.H. Zhong and Q. Huang, "Implementation of an Ant Colony Optimization technique for job shop scheduling problem", 2006, The Institute of Measurement and Control 10.1191/0142331206tm165oa.

[15] Ahmad Rabbani Motlagh, "An Efficient Ant Colony Optimization Algorithm for Multi-objective Flow Shop Scheduling Problem", World Academy of Science, Engineering and Technology 75 2011.

[16] Benjamin Doerr, Frank Neumann, Dirk Sudholt and Carsten Witt, "On the Influence of Pheromone Updates in ACO Algorithms", Technical Report ISSN 1433-3325 January 2007.

[17] Christian Blum, "Review Ant colony optimization: Introduction and recent trends", available online at www.sciencedirect.com.

[18] Frederic Villeneuve, "A Method for Concept and Technology Exploration of Aerospace Architectures", School of Aerospace Engineering, Georgia Institute of Technology, USA, August, 2007.