International Journal of Quality research
UDK – 658.524
Short Scientific Paper (1.03)

# A New Approach to Job Shop-Scheduling Problem

***Ramezanali Mahdavinejad***
*University of Tehran, Iran*

***Abstract:*** *In this paper, single-processors jobshop scheduling problems are solved by a heuristic algorithm based on the hybrid of priority dispatching rules according to an ant colony optimization algorithm. The objective function is to minimize the makespan, i.e. total completion time, in which a simultanous presence of various kinds of ferons is allowed. The process of finding the best solution will be improved by using the suitable hybrid of priority dispatching rules. Ant colony optimization algorithm, not only promote the ability of this proposed algorithm, but also decreases the total working time because of decreasing in setup times and modifying the working production line. By solving some problems as samples (i.e. Fisher,s & Tomson,s problems), this algorithm is compared with the others. The results show that when the size of the problem becomes lorger, the deviation from lower limit increases, but its rate decreases with the size of the problems, so that it reaches to its limit.*
***Keywords:*** *Job shops scheduling, Priority dispatching rules, Makespan, Hybrid heuristic algorithm.*

## 1. INTRODUCTION

Solving jobshop-scheduling problem is a sequence of jobs to be processed on machines in alternative routes. The numerous variables and constraints are involved in job shop scheduling. Its complexity of the large solution space and multi-criteria objective functions, make the problem difficult. This problem is a class of NP-Hard ones that cannot be optimally solved for large-scale
problems in a reasonable amount of computational time. For this reason, great deals of researches develop and work on heuristic methods to find near-optimal solutions. In general to solve such problems, typical methods are introduced as: local searching algorithm (Brucker et all,1996), priority dispatching rules,expert systems,ant colony optimization, genetic algorithms (Shi,1997), branch-and-bound algorithm (Brucker et all,1994), shifting bottleneck algorithm (Demirkol et all,1997), tabu search, simulated annealing, neural network (Jain and Meeran,1998) and graph theory. In addition to the above methods, a hybrid method can be also proposed such as: local search and simulated annealing algorithms, fuzzy and priority dispatching rules algorithms, local search and genetic algorithms(Yamada and Nakano, 1996). Simulated annealing (SA) method is a well-known meta-heuristic method that is widely used in job shop scheduling problems. SA cannot find the exact solution for large-scale problems with high-speed rate. Therefore, to improve this weakness, this method can be combined with some other methods such as critical neighborhoods to produce active job shop scheduling [6] and genetic algorithms to speed up the process efficiently.

## 2. MATHEMATICAL MODEL

There are some assumptions in job shop scheduling listed bellow:

At the time beginning, all works and workstations are ready to start and processing jobs.

Machines are arranged according to the functional/process layout based on similar functions in the job shop. For example, turning machines are in one group, milling machines in another group and so on. In this research, each group is called a workplace and each machine in these groups is called workstation.

Each job is allowed to visit any workstation, which is determined via proposed algorithm. This assumption exists to the most real-world situation indeed. Note that the total time, which includes of setting and processing time on jobshop places, due to the ability differences in working stations, can be different from each other. Besides, the number of working places in production line and also the number of working stations in a working place are as input parameters.

Every job in the jobshop is considered as a non-natural ant.

If there are various kinds of jobs in the jobshop at the same time, then every job will remain its own fermon on its tracing line, so that it will pay attention only to its own fermon in selecting the moving trace. In this research, an algorithm inspired from ant colony optimization algorithm is used. So that, depending on the numerous jobs, various kinds of fermons are considered for non-natural ants instead of one special fermon.

Every job can pass through a special working place only once during its production sequences. Therefore, coming back to a special working place is forbidden. According to this assumption there is not any necessarily to create a special forbidden list for every non-natural ant. This list without the assumption mentioned is quite necessary to correct the action of the ant colony optimization algorithm.

The time used for any process on every working station is divided into two groups;one portion is the time needed for transportation, working station preparation and working setups on these working stations. This portion of time is called processing setup time. Another portion of the time is processing time, which is the time needed on the working station.

Works have not any predetermined priority to each other.

Setting up and processing times of each process on any working station is determined.

The sequence and the arrangement of the processes, which are being done on any work, are determined.

Interruption of a process when it is being done on a working station is forbidden, until the process is completed.

The variables and parameters of this model are as follows:

$N_i^j$ = Number of remaining process for job *j* at the stage of scheduling that this job can be assigned to station *i*.

$Y_{i,j}$ = Starting time for processing job *j* on stations *i*.

$\pi_{i,k}^j$ = Amount of feremon remained from job j on the route between station *i* to *k*.

$\tau_{i,k}^j$ = Severity of feremon remained from job j on the route between station *i* to *k*.

In order to better anderstanding of mathematical model, the problem is shown as a general graph in figure 1. In this graph, each node and arrow determines a process and its processing time, respectively. The direction of the arrows shows the transposition of the processes relative to each other. The node (i,j) shows a process in which the job j is being done on station i.

According to the above mentioned, the mathematical model of a single-jobshop scheduling process is as follows:

Minimize $C_{max}$

S.t.

$$Y_{kj} - Y_{ij} \geq P_{ij} \quad \forall (i,j) \rightarrow (k,j) \tag{1}$$

$$C_{\max} - Y_{ij} \geq P_{ij} \quad \forall (i,j) \tag{2}$$

$$\begin{cases} Y_{ij} - Y_{il} \geq P_{il} \\ \text{or} \\ Y_{il} - Y_{ij} \geq P_{ij} \end{cases} \quad \forall (i,l),(i,j) \qquad (3) \qquad (3)$$

$$P_{ij} > 0 \qquad \forall\,(i,\,j) \qquad (4)$$

$$Y_{ij} \geq 0 \qquad \forall(i,j) \qquad (5)$$

In above equations the parameters are defined as follows:

$C_{max}$:The total completion time , so that its optimization is the aim of this research.

$P_{ij}$ : Processing time of j job on stations i.

## 3. PROPOSED SOLVING METHOD

In this section, the priority algorithm will be discussed step by step. First of all, it

is necessary to give some more explanations about this algorithm. In this work, the scheduling jobs are assumed as unreal ants. These ants are done on general information, because localized information refers to the specification of the job at any point, so that, these specifications are a part of job and making any difference will be as revolution of the subject itself. Jjobshop scheduling with any algorithm with the aim of duration time optimization has to optimize the setting times on each working station. Because due to the variety and low rate productions in jobshop working, the general machines usually without any special accessories as fixtures are used. In this manner, working setup is easy. On the contrary, in mass production, the machines are designed for a special job with easy and rapid function. From this point of view, there is a noticeable difference between jobshop with other production process in the time percentage, which is needed to set up the workplace and machine itself. Therefore, optimization of scheduling jobshop and minimization of setting times is very important. So in this research

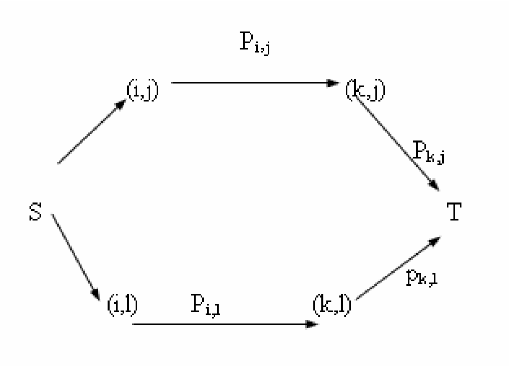

*Figure 1:Directed garhp for a job shop-scheduling problem*

making decision to find the traces themselves, which are work circumstance in the workshop indeed. These decisions are based on the selection of a rule, which is according to the action of two kinds of informations; The first one as general information is those, which are attending to experience of other ants during their tracing lines. Localized information is the second one, which is concerned to the decision of ant to travel from one working station to the next one. The first three sentences of the above equations (1-3) are the type of this kind of information. Optimization is multi fermons has been considered for various kinds of jobs in jobshop. In this manner, every job in jobshop is to be concerned to an unreal ant. A special unreal ant is also considered to a special job due to the fermon of an ant that is different from the others. Therefore, the fermons of two different ants pass through a special trace (line) can't be gathered together. An ant can follow its line only by the fermon of its own kind, which is remained and will not follow any traces with other kinds of fermons. Sensibility of unreal ants only to their own fermons is the trends by which, the setting times on working stations can be minimized. Supposed two kinds of jobs (1) and (2) can be done on a special working station. If you are going to do the job (2) on the working station that is doing job (1), then you need an extra time to prepare and set system up. But reprocessing of job (1) on this workstation will save a noticeable set up time. So it is better to process similar jobs on a special working station. In this research, as a classical optimization algorithm of ant colony, three steps; The calculation of a trace selection probability, Selection of a trace among the others and finally updating of fermons on all traces are considered. The sequences of proposed algorithm are as follows:

**First step:** Formation of schedulable processes.

**Second step:** Calculation of unreal processing times for all schedulable processes on their working stations. In this scheduling unreal time will be the start of processing time if the processing allocates to the considered station.

**Third step:** The allocation probability will be calculated for all scheduling processes. $\pi^{C}_{a,b}$ is defined as the fermon amount related to work (c) which is remained on the line from (a) to (b) working stations, so that:

$$\tau^{C}_{a,b} = \frac{\pi^{C}_{a,b}}{\sum_{k,i} \pi^{K}_{i,b}} \quad (6)$$

$\tau^{C}_{a,b}$ is fermon intensify from unreal ant (c) remained on line working stations from (a) to (b).

$\sum_{k,i} \pi^{K}_{i,b}$ is remained fermons of whole unreal ants on working stations from (a) to (b).

In this priority algorithm the following equation is used to calculate the allocated probability of processes:

$$p^{C}_{a,b} = \frac{\left(\frac{1}{T^{C}_{b}}\right)^{l} . \left(R^{C}_{b}\right)^{m} . \left(\frac{1}{1+S^{C}_{b}}\right)^{n} . \left(\tau^{C}_{a,b}\right)^{p}}{\sum \left(\frac{1}{T^{k}_{b}}\right)^{l} . \left(R^{k}_{b}\right)^{m} . \left(\frac{1}{1+S^{k}_{b}}\right)^{n} . \left(\tau^{K}_{i,b}\right)^{p}} \quad (7)$$

The parameters of this equation are as:

$P^{C}_{a,b}$ : Movement probability of work (c) from station (a) to (b).

$T^{C}_{b}$ : Processing time of work(c)on station (b).

$R^{C}_{b}$ : Remained relative working time of work (c) processing on work station (b).

$S^{C}_{b}$ : Assumed start-processing time of work (c) on station (b).

*l,m,n,p* : Constant parameters which can be any real and positive number.

These parameters evaluate the importance coefficient of processing times remained relative time, assumed start processing time and remaind intensify of fermons on each line.

K: scheduling processes that are going to be process on station (b).

$T^{C}_{b}$ Can be determined from following equation:

$$T^{C}_{b} = TP^{C}_{b} + \lambda^{C} . TS^{C}_{b} \quad (8)$$

The definition of the parameters in this equation is as follows:

$TP^{C}_{b}$ :Processing time of work(c)on station (b).

$TS^{C}_{b}$ :Setup time of work (c) on station (b).

$\lambda^{C}$ :Input coefficient between zero and one, which is used in any work. This coefficient is applied on times needed to set the piece up, when the working station had been processing the similar work, otherwise this parameter will be equal to one. $\lambda^{C}$ Denotes the saving times on working stations with repeated processing indeed. $R^{C}_{b}$ can also be calculated as:

$$R^{C}_{b} = \frac{N^{C}_{b}}{\sum N^{k}_{b}} \quad (9)$$

$N^{C}_{b}$ Is the remained processing number of work (c) before processing on station (b).

**Fourth step:** After calculation of allocating probability of each process of schedulable processes set in third step, these amounts will be ordered descending in this step.

**Fifth step:** The allocation of schedulable processes to working stations will be with the accordance of their highest priorities.

**Sixth step:** After allocation process in fifth step, allocating probability of all processes, which are in schedulable processes set and will allocate to allocated workstation in fifth step, become zero.

**Seventh step:** Starting and processing duration and also final production time will be calculated in this step.

**Eighth step:** Return to fourth step until the allocation probability of whole schedulable processes are zero.

**Ninth step:** Return to first step and making a new collection of schedulable process until this new collection is empty.

**Tenth step:** Record the longest completion time for scheduled processes as makespan. In this step a cycle will be completed.

**Eleventh step:** At the end of a cycle the fermons of various unreal ants on all traces must be up to date. As mentioned before, there may be more than one job on production line so in spite of their similarity; they may travel through various lines. Therefor, the question is that, for a special job, on which line the fermon will be strengthened or weakened?. To answer this question, it is noticeable that among similar lines, the fermon will be strengthened on the shortest one. Up dating values of fermons is done by following equation:

$$\pi_{i,j}^{C}[t+1]=\begin{cases}(1-\rho).\pi_{i,j}^{C}[t] & \text{(a)}\\ R+(1-\rho).\pi_{i,j}^{C}[t] & \text{(b)}\end{cases} \qquad (10)$$

The equation (10a) is valid only when the selected unreal c kind ant has not passed through line *i* to *j* otherwise the equation (10b) will be satisfied. In this equation input values of ρ and *R* are fermon evaporation rate (0-1) and real value greater than one respectively.

**Twelfth step:** Referring to the first step and starting the next optimization cycle until coming into the end conditions is satisfied. These conditions can be the optimization of time to solve the problem, the optimization number of cycles or percentage value optimization and so on.

**Thirteenth step:** Introducing the least amount of total time resulting from optimization cycles during production.

**Fourteenth step:** the end of algorithm.

The item $(1/T)$ in eq.7 denotes the use of preference distribution law for the shortest processing time. Using this law minimizes the handling time of pieces in jobshop and after that decreasing of total completion time. But this law has also a weakness. Suppose a special job with long process time is together with some other jobs short process times that are to be processed. In this manner, the job with longest processing time will always waiting at the end of line. According to this law to compensate this weakness, the term relative remained time *R* is considered in this equation. So with this term the arrangement of the jobs with long processing times will be according to their remained processing amounts and this is obviously a better solution of scheduling job shop problems. The term $(\frac{1}{1+S})$ is also considered the waste times on working stations. Therefore, the allocation preference in the same conditions will be to process with the earliest starting of unreal processing time. The term $\tau_{a,b}^{C}$ in each step of scheduling causes the selection of the highest density of fermon among working stations. This means that similar jobs will be processed on a special working station, so the setup times will be minimized by not processing of jobs on the other working stations.

## 4. DISCUSSION

The evaluation of single priority algorithm processor has been done by comparison of the results with four problems that three of them are from Fisher and Tamson. The results have also been compared with six famous methods of priority distribution rules, which will be mentioned here. EDD algorithm is the earliest delivery time rule. According to this rule, the job with the least delivery time will be allocated to a working station. MS algorithm is the least interruption rule in which schedule will be based upon the least interruption time of production. According to FCFS algorithm the processing arrangement will be based upon their arrival sequence on working stations. LPT is the longest processing time algorithm. In this algorithm the job will be allocated to working stations according to it highest processing time. On the contrary of LPT, SPT is based on the smallest processing time. CR denotes the critical rule algorithm in which job scheduling will be based on the remained to delivery times.

The general specifications of selected problems to priority algorithm evaluation are shown in Table 1.

The results of priority and other compared algorithm,s solving time are shown in figures 2 and 3 respectively
Among these algorithms, MS and LPT have the similar behavior. CR and SPT algorithm

show also better operations in increased dimensions (see 6*6*6 results in fig. 3).
predictable behavior. It also reaches to a constant amount at high values, which is the reason of its good operation at high dimensional problems.

These results are obtained when *l, m, n*=1.0 because the best answering of priority timed algorithm is when processing relative remained and starting unreal processing times are at the same rank of importance. Therefore, this can be convenient suggestion value to these parameters.

## 5. RESULTS AND SUGGESTIONS

Analysis and solved problem results show that dimensional expansion of lower bound increases solution times.

But the gain of this priority algorithm is that the deviation amounts decreases with dimensional increasing of problem size and reaches a constant amount.

Besides the behavior of priority algorithm against problems with various sizes is steady and predictable so that the operational quality and accuracy of this priority algorithm, which is very important, can be approximately predicted. This algorithm is also able to give optimized result to problems with $(6!)^6$ dimensional space and less than 3 to 5 percent of errors. But it must be mentioned that the solving time of problems in this algorithm varies exponentially with the size of problems.

*Table 1- Test problems required for solving a single-processor scheduling.*

| Test | Number of Jobs | Number of Machines | Number of Operations | Subject |
|---|---|---|---|---|
| 1 | 4 | 4 | 4 | - |
| 2 | 6 | 6 | 6 | FT6 |
| 4 | 10 | 10 | 10 | FT10 |
| 4 | 20 | 5 | 5 | FT20 |

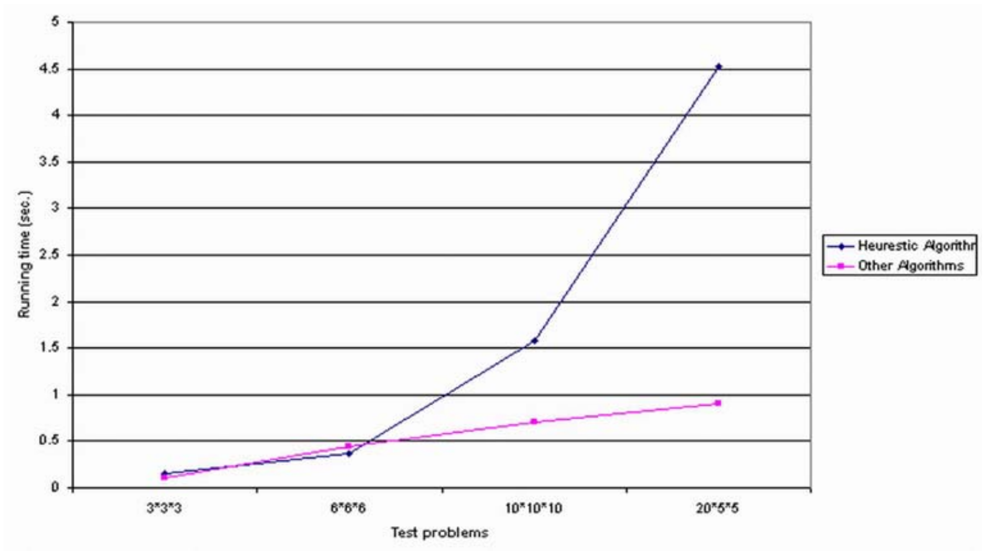

*Figure 2. Makespan comparison between proposed and other algorithms.*

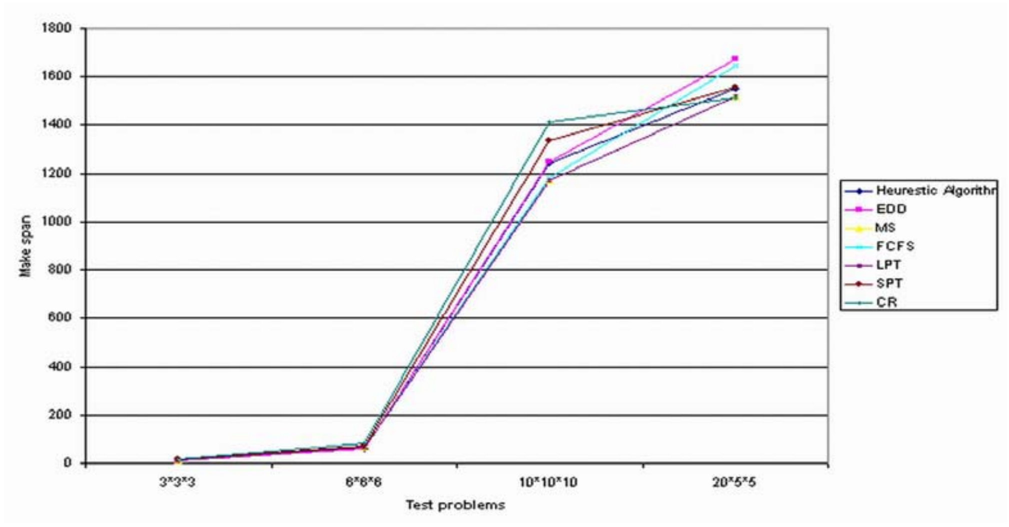

*Figure 3- Running time comparison between proposed and the other algorithms.*

REFERENCES

[1] Brucker,P.- Hurink,J. and Warner,F. (1996): “Improving local search heuristic for some scheduling problems part I”, Discrete Applied Mathematics 65/1-397-122.
[2] Brucker,P.- Jurisch,B. and Sievers,B. (1994): “A branch and bound algorithm for the job-shop scheduling problem”, Discrete Applied Mathematics 49, 109-127.
[3] Demirkol,E.-Mehta,S. and Uzsoy,R. (1997): “A computational study of shifting bottleneck procedures for shop scheduling problems”, Journal of Heuristic, 3/2, pp111-137.
[4] Jain,A.S. and Meeran,S. (1998): “Job-shop scheduling using neural networks” International Journal of production research 36/5, pp1249-1272.
[5] Shi,G. (1997): “A genetic algorithm applied to a classic job-shop scheduling problem”, International Journal of Systems Science 28/1, pp25-32.
[6] Yamada,T.and Nakano,R. (1996): “Scheduling by genetic local search with multi-step crossover”, PPSN’IV Fourth International Conference on parallel problem solving from nature, Berlin, Germany, pp 960-969.

Received: 02.09.2006 Accepted: 21.11.2006 Open for discussion: 1 Year