BVG Trust

J. Sci. Trans. Environ. Technov. 2014, 8(2) : 75-78

Scientific Transactions in Environment and Technovation

Minimization of makes pan in multiple parallel flow line scheduling

J.V.Balasubramanian and [1]Tejas Krishna Sathyamurthi"

Dept. of Management Studies, SITAMS, Chittoor

[1] 7th Grade, New Hampshire, USA.

"(Research carried out along with the Author during Summer 2014)

Abstract

This paper presents a scheduling problem for parallel flow line with sequence-dependent setup times, using simulated annealing (SA). The problem accounts for allotting work parts of " l " jobs into " m " parallel flow line, where an optimal schedule is defined as one that gives the optimal makes pan. Each machine has its own processing times according to the characteristics of the machine as well as job types. The set of jobs has to be scheduled, without preemption, on identical parallel flow lines. Each line is able to treat only one job at a time. Setup times are machine independent but job sequence dependent. SA, a meta-heuristic, is employed in this study to determine a scheduling policy so as to minimize makes pan. The experimental analysis shows that the proposed SA method significantly outperforms a neighborhood search method in terms of makes pan.

Keywords : meta-heuristic, parallel flow line, simulated annealing

INTRODUCTION

Parallel flow line scheduling can be briefly characterized as the activity of assigning a number of jobs to a number of performing lines carrying machines such that certain performance demands like time or cost effectiveness are fulfilled. This activity, which is best viewed as an optimization or constraint satisfaction task, is ubiquitous in everyday life. As the above characterization indicates, scheduling plays a particularly important and critical role in industrial contexts. Scheduling is a highly complex activity that is known to be NP-hard problem even for much static. Parallel flow line setups have more than one flow line, which are added or installed at different period of time. In such circumstances the process is normally same but the processing times and time to switch over from one variety of jobs to another variety is varying. It is reasonable to assume that new installation would be efficient with more features and hence avails minimum time for processing. At the same time, it is to be kept in mind that the old ones are still in usable form and need to be utilized for order fulfillment. Hence parallel flow line scheduling problems essentially require allocation of job variety to a specific flow line and sequencing of varieties allotted within each flow line.

The decision to be taken for such scheduling problem is a difficult one. Genetic algorithm (GA) and tabu search algorithm (TS) have been applied for such combinatorial optimization problems, in view of its characteristics such as near optimization, high speed and easy realization. Now it is tried with simulate annealing approach.

Maged M. Dessouky(1998) considers the problem of scheduling 'n' identical jobs with unequal ready times on 'm' parallel uniform machines to minimize the maximum lateness and develops a branch-and-bound procedure that optimally solves the problem and introduces six simple single-pass heuristic procedures that approximate the optimal solution. Guo-Hui Lin, (1998) et al discuss parallel machine scheduling problem, in which the conventional weighting function technique have been used. Moon-Won Park and Yeong-Dae Kim (1997) consider a multiple parallel machine scheduling orders with objective of minimizing holding cost and suggest two local search heuristics, simulated annealing algorithm and tabu search algorithms. Yong He (2000) investigates on-line parallel machine scheduling problems and shows the optimality of the classical list scheduling (LS) algorithm. Dong-Won Kim et al. (2000) addresses the problem of unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. They propose four search heuristics to address the problem, namely (i) the earliest weighted due date, (ii) the shortest weighted processing time, (iii) the two-level batch scheduling heuristic, and (iv) the simulated annealing method. Michael X. Weng et al.(2001) address the problem of scheduling a set of independent jobs on unrelated parallel machines with job sequence dependent setup times so as to minimize a weighted mean completion time. They propose seven heuristic algorithms and test by simulation. Bank and Werner. (2001) schedule the jobs assigned to each machine such that the weighted sum of linear earliness and tardiness penalties is minimal. Dong-Won Kima, et al[8] presents a scheduling problem for unrelated parallel machines with sequence-dependent setup times, using simulated annealing (SA). Zhiyi Tan and Yong He (2002) consider ordinal algorithms for parallel

"Corresponding Author :
e-mail : dr.jvbalu@gmail.com

machine scheduling with non-simultaneous machine available times with the objectives of minimizing the latest job completion time and minimizing the latest machine completion time. Liu Min and Wu Cheng (1999) minimize the makes pan in identical machine scheduling problem. From the literature survey it is decided that simulated annealing can solve parallel flow problems and performance of it can be measured. In the next section, problem environment that is being dealt is defined with the assumptions and notations made in it. The general scheme of SA, its structure and methodology are briefly described in third section. Simulated annealing algorithm and its parameter with procedure are put forth in the fourth section. In section five, the experimental studies on the defined problem through simulated annealing algorithm are illustrated.

Statement of the Problem

In the parallel flow line-scheduling problems, the number of flow lines each having similar set of machines but with different capabilities are considered. It is assumed that the job requires processing in every machine of the line and the processing time differs from line to line. The reason for variation of processing time is due to the varying capability and functionality of machines of each line. The setup times of jobs are varied for every sequence of the jobs in every line but the setup time is same in all lines**.** Also, each flow line is composed of identical machines having the same functional performance but different functional capabilities. Each job has a deterministic processing time. No preemption is considered in this case, since no interruption of jobs between the flow lines occurs. The jobs are processed in any one of the lines and each line can process utmost one job at a time.

The following assumptions are made in solving the problem.

- All the jobs are dependent sequences as arranged in the flow line.
- Any job could be processed in any line.
- The individual job processing time is deterministic and included in transportation time between two workstations.
- Breakdown of machine is not allowed.
- The jobs once assigned to particular line should flow in the same line.
- The sequence is dependent with setup time and there is prescribed priority between individual jobs.

The following notations shall be used to define the problem.

i-job identifier {i= 1,2,3…n}
l-line identifier {l= 1,2,3…j}
k-machine identifier {k= 1,2,3…m}
t_{ilk}-process time for job i at line l in machine k
q-batch quantity of job i
T_{cil}-total processing times for job i in line l.
s_{i-i} – set up time for job to job

The objective function of the problem is to determine which job is to be allocated in which line and in what sequence so as to minimize make span.

$$\text{Min} \sum_{i=1} T_{cil} = \sum_{l=1}^{n} \sum_{k=1}^{j} \sum^{m} t_{ilk} + (q_i - l) \max (t_{ilk}) + s_{i-i}$$

General Scheme of SA

Simulated Annealing was proposed by Kirkpatrick et al. (1983), and independently by Cerny[12]. Starting from an initial solution, SA generates a new solution S′ in the neighborhood of the current solution S, Then, the change in the cost function value, $\Delta = C(S') - C(S)$, is calculated, where C (.) denotes the cost function value of solution •. In minimization problems, if ″ < 0, transition to the new solution is accepted (this transition is called a downhill move). If $\Delta > 0$, transition to the new solution is accepted with a specified probability usually obtained by the function exp ($-\Delta$ / T), where T is a control parameter called the temperature. By allowing uphill moves (transitions that increase objective function values) like this, SA can escape from a local minimum in its search for the global minimum. SA repeats this process L times at a temperature, where L is a control parameter called the epoch length. The parameter T is gradually decreased by a cooling function as SA progresses, until a given stopping condition is satisfied. Schematic procedure is as follows.

1. An initial configuration is configured.
2. Starting off at an initial configuration, a sequence of iterations is generated. Random selection of a configuration from the neighborhood of the current configuration leads to each iteration.
3. The corresponding change is calculated in terms of cost function.
4. The neighborhood is defined by the choice of a generation mechanism, i.e. a "prescription" to generate a transition from one configuration into another by a small perturbation.
5. If the cost function change is negative, the transition is unconditionally accepted; if the cost function increases, then the transition is accepted with a probability based upon the Boltzmann distribution, which is a constant, and the temperature is a control parameter.
6. From a sufficiently high starting value this temperature is gradually lowered throughout the algorithm to a "freezing'' temperature, where no further changes occur.

7. The temperature decrement is done through stages. At each stage the temperature is kept constant until thermal quasi-equilibrium is reached.

8. The whole of parameters determining the temperature decrement (initial temperature, stop criterion, temperature decrement between successive stages, number of transitions for each temperature value) is called the cooling schedule.

A typical procedure of SA is given below.

Step 1. Generate an initial solution S.

Step 2. Select a value for the initial temperature, $T_1 > O$ Set the epoch count k = 1.

Step 3. Repeat the following L times (L is called the *epoch length).*

i)Generate a neighborhood solution *S'* of S.

ii)Let Δ= C(S') - *C(S).*

iii) If Δ < 0, let S be S' (downhill move).

iv) If Δ >_ 0, let S be S' with probability, exp (Δ /T_k) (uphill move).

Step 4. If a given stopping condition is satisfied, stop.

Otherwise, let $T_{k+1} = F(T_k)$ and k = k + 1, and go to step 3. (T_k and F denote the temperature at the k-th epoch and the cooling function, respectively.)

The initial sequence obtained from randomly generated method. The pair wise interchange mechanism is explained with an example,

Let the initial sequence be {2 1 2 / 2 3 5 1 4}

Set {a} = {2 1 2 / 2 3 5 1 4}

Swap the jobs which is in the position 1 & 2 after slash in {a} {2 1 2 / 3 2 5 1 4}

Swap the jobs which is in the position 1 & 3 after slash in {a} {2 1 2 /5 3 2 1 4}

Swap the jobs which is in the position 1 & 4 after slash in {a} {2 1 2 / 1 3 5 2 4}

Swap the jobs which is in the position 1 & 5 after slash in {a} {2 1 2 / 4 3 5 1 2}

Swap the jobs which is in the position 2 & 3 after slash in {a} {2 1 2 / 2 5 3 1 4}

Swap the jobs which is in the position 2 & 4 after slash in {a} {2 1 2 / 2 1 5 3 4}

Swap the jobs which is in the position 2 & 5 after slash in {a} {2 1 2 / 2 4 5 1 3}

Swap the jobs which is in the position 3 & 4 after slash in {a} {2 1 2 / 2 3 1 5 4}

Swap the jobs which is in the position 3 & 5 after slash in {a} {2 1 2 / 2 3 4 1 5}

Swap the jobs which is in the position 4 & 5 after slash in {a} {2 1 2 / 2 3 5 4 1}

Swap the jobs which is in the position 1 & 2 before slash in {a} {1 2 2 / 2 3 5 1 4}

Swap the jobs which is in the position 1 & 3 before slash in {a} {2 1 2 / 2 3 5 1 4}

Swap the jobs which is in the position 2 & 3 before slash in {a} {2 2 1 / 2 3 5 1 4}

EXPERIMENTAL STUDIES

The computational experiments are conducted with an illustrative problem. Thereafter processing time for three jobs those processed in three lines, each line containing three machines are given in Table – 1. The setup time for the particular sequence in all lines is same. The sequence setup time matrix for the sample problem is shown in Table - 2. Various numbers of jobs have to be processed for all lines. The quantities of jobs to be processed are shown in Table – 3. SA program have to run and got the results is shown in the following Table – 4.

Table – 1 : Processing Time

		M1	M2	M3
J1	L1	2	7	3
	L2	4	5	6
	L3	5	4	6
J2	L1	6	6	5
	L2	4	2	2
	L3	3	4	3
J3	L1	1	7	5
	L2	2	8	6
	L3	4	2	8
J4	L1	3	6	4
	L2	5	3	7
	L3	3	7	6
J5	L1	2	4	6
	L2	4	6	3
	L3	7	1	6

Table – 2 : Sequence Setup Time Matrix For all lines

j	1	2	3	4	*5*
1	—	5	6	***2***	***1***
2	4	—	7	***7***	***4***
3	2	5	—	***3***	***6***
4	***3***	***6***	***7***	—	***3***
5	***5***	***2***	***1***	***4***	—

Table – 3

Quantity of job to be processed

JOB	QUANTITY
J1	12
J2	15
J3	16
J4	19
J5	13

Table – 4

Result

Lines	Sequence	Makes pan
L1	5 – 3	203
L2	1 – 2	
L3	4	

CONCLUSION

This paper develops a performance measure of simulated annealing that optimally solves the problem of scheduling identical jobs in parallel flow lines to minimize the maximum makes pan of the jobs. The performance of the proposed algorithm was not affected by any increase in the number of machines, increase in the number of lines, and increase in the number of jobs but the optimal solution obtained from more number of generations. From numerical computational result of the algorithm, the simulated annealing is very powerful and capable of solving the problem very viable and reliable, since it is not too complicated and may be easily implemented through personal computer, and may provide satisfactory solutions.

REFERENCES

Bank, J. and Werner, F., 2001. Heuristic Algorithms for Unrelated Parallel Machine Scheduling with a Common Due Date, Release Dates, and Linear Earliness and Tardiness Penalties, *Mathematical and Computer Modelling*, 33 : 363-383.

Cerny, V. 1985. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm, *Journal of Optimization Theory and Applications*, 45:41-51.

Dong-Won Kim, Kyong-Hee Kim, Wooseung Jang and Frank Chen, F. 2002. Unrelated parallel machine scheduling with setup times using simulated annealing, *Robotics and Computer Integrated Manufacturing*, 18 : 223–231.

Dong-Won Kim, Dong-Gil Na and Frank Chen, F. 2003. Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective, *Robotics and Computer Integrated Manufacturing*, 19 : 173–181.

Guo-Hui Lin, En-Yu Yao and Yong He. 1998. Parallel machine scheduling to maximize the minimum load with nonsimultaneous machine available times, *Operations Research Letters*, 22 : 75 – 81.

Kirkpatrick, S, Gelatt, C.D, Jr. and Vecchi, M.P.1983. Optimization by Simulated Annealing, Science, 220:671-680.

Liu Min and Wu Cheng. 1999. A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines, *Artificial Intelligence in Engineering*, 13 : 399–403.

Maged M. Dessouky. 1998. Scheduling Identical Jobs with Unequal Ready Times on Uniform Parallel Machines to Minimize the Maximum Lateness, *Computers Industrial Engineering*, 34 : 793 – 806.

Moon-Won Park and Yeong-Dae Kim. 1997. Search Heuristics for a Parallel Machine Scheduling Problem with Ready Times and Due Dates , *Computers Industrial Engineering*, 33 : 793-796.

Michael X. Weng, John Lu and Haiying Ren. 2001. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective, *International Journal of Production Economics*, 70 : 215 – 226.

Yong He. 2000. The Optimal On-Line Parallel Machine Scheduling, *Computers and Mathematics with Applications*, 39 : 117-121.

Zhiyi Tan and Yong He. 2002. Ordinal Algorithms for Parallel Machine Scheduling with Nonsimultaneous Machine Available Times, *Computers and Mathematics with Applications*, 43 : 1521-1528.