comparative study

by Tanti Octavia

Submission date: 05-Apr-2019 02:07PM (UTC+0700)
Submission ID: 1106399560
File name: comparative_study.pdf (326.48K)
Word count: 2607
Character count: 12810

Comparative Study of Multi-Item Batch Scheduling and the Hybrid of Proposed Ant Colony Algorithm -Tabu Search

Tanti Octavia[1] and Dessianne Thali[1]
[1)]Industrial Engineering Department, Petra Christian University
Surabaya - Indonesia
Email : tanti@peter.petra.ac.id

Abstract

This paper addresses the problem of scheduling jobs with multi batches. Multi-Item Batch Scheduling Model divides jobs into several batches so that the scheduling problem does not only determine the sequence of batches, but also determines the number of batches and batch sizes are used. The research attempts to compare the methods of multi-item batch scheduling to the combination of tabu search and proposed ant colony algorithm (PACO-TABU) to minimizing mean tardiness.

Multi-item batch scheduling model uses two methods to determine the size, sequence, and many batches are used. The method used is slack method and PACO-TABU algorithm. Experimental analyses are presented to compare the performance of the proposed PACO-TABU algorithm and multi-item batch scheduling in term of mean tardiness. The results show PACO-TABU algorithm has better performance than the method of slack in term of mean tardiness.

1. Introduction

In single machine batch scheduling problem, a set of jobs $J_1, J_2, J_3, \dots J_n$ can be partition into several parts, which is defined as batch. Each job requires a setup time and processing time. Some researches have been done in this single machine batch scheduling problem. Some researches used dynamic programming algorithm in order to solve a batch scheduling problem. Baptiste (2000) developed a dynamic programming algorithm for 1|p-batch, $b<n, r_j, p_j=p, C_j \leq d_j$|- which runs in $O(n^8)$ time. Shakhlevich (2010) provided $O(n^3 log\ n)$-time algorithm that minimizing the maximum lateness for parallel batch machine problem. Dati (2009) proposed a closed form solution in a batch scheduling problem on a 2-machine flow shop. Potts and Kovalyov (2000) developed a family scheduling considering the item availability model $1|s_f|V$ and the batch availability model $1|s_f;batch|V$ where s_f and V is setup time of each family F_f and objective function to be minimized. The completion time of a job J_j in a batch B of family F_f under schedule π is defined as

$$Cj(\pi) = SB(\pi) + s_f + \Sigma Pi,$$

Where SB(π) is the starting time of the batch B under π, and $\varphi j\ (\pi) = \{J_i\ \varepsilon\ B: i=j \text{ or } Ji \text{ is processed before } Jj\}$. In the batch availability model, the completion time of job J_j in a batch B of family F_f under a schedule π is defined as the time when the last job in this batch has finished processing. Shabtay (2007) presented polynomial time algorithms to find the job sequence into batches and resource allocation that minimizing the total completion time. In our paper, we challenge to study single machine batching scheduling problem using heuristic algorithm. We develop the combination of the ant colony algorithm and tabu search to minimize the mean tardiness. The tabu search will be applied as a local job search on the ant colony algorithm.

The paper is organized as follows. In Sect.2 we present tabu search and ant colony algorithm in scheduling problem. In sect.3, we present the proposed hybrid algorithm that minimizing the mean tardiness. Section 4 presents a numerical example and simulation result. The last section contains our concluding remarks.

2. Tabu Search

Tabu search is a meta-strategy for conducting known heuristics to help then avoid local optima. Thiesen (1998) stated four main elements in Tabu search approach are a local search strategy, a mechanism to the tabu list, a tabu tenure policy, and a mechanism to alter the search path. A local search is usually a simple greedy strategy which finds an improved solution in the immediate of a current solution. The tabu list is a mechanism to discourage a return to a recently visited solution. If a solution found by local search already exists in the tabu list then it is forbidden. In order to ensure the search does not exclude all neighbours quickly, tabu tenure formulate solutions are not held in the tabu list for some period of time.

In this paper, mean tardiness will be the objective function value (OFV) as criteria to determine whether or not the replacement position. Replacement is required if the OFV position that has been defined more optimal than the previous. Approach starts from the beginning of the operation sequence of jobs and search through the neighborhood with the smallest order of tardiness time. Job that exchanges and produces the best value in term of the mean tardiness will go into the tabu list. Tabu search has the ability to get out of local optimal tabu search but can not find the global optimum. Tabu search must have a total maximum limit of iterations and the size of the tabu list is determined solely by individuals who use this method.

3. Ant Colony Optimization (ACO)

This algorithm was instigated by the ant activity in search food. The ants deposit a chemical pheromone trail as they move about their environment, and they are also able to detect and follow pheromone trails that they may encounter. Dorigo (2002) describes the five requirements in ant colony approach. They are: 1. a heuristic function, that will guide the ants' search with problem specific information; 2. a pheromone trail definition, which states what information is to be stored in the pheromone trail. This allows the ant to share information about good solution; 3. the pheromone update rule, this defines the way in which good solutions are reinforced in the pheromone trail; 4. a fitness function which determines the quality of particular ant's solution; 5. a construction procedure that the ants follow as they build their solutions.

4. Multi-item batch scheduling

Batch scheduling problems have different characteristics with job scheduling, the execution of each job can be done in several parts, which is defined as a batch. As a result the number of job changes and job processing time distribution follow a batch job. This means that the scheduling problem becomes more complex, which is seeking the division of a batch job, the size of each batch, and the search for order execution of batch produced. These characteristics are fundamental differences lead to job scheduling rules can not be directly used for batch scheduling problems.

Research on multi-item batch scheduling with time dependent processing has been done by Sukoyo, et al (2010). The study uses actual flowtime criterion as a measure of performance of the model. Parameters used in the model of multi item batch scheduling as follows:

i, j, k : index of the order of batch (*i* = 1,2, ..., N)
g : job index (*g* = 1,2, ..., G)
G : the number of job
N : number of batches
Dg : the number of job requests *g* (unit)
d (*i*) : time of delivery (due date) for the *i*-th batch
s (*i*) : setup time for the *i*-th batch
L (*i*) *g* : the value of the *i*-th priority batch job contains the item *g*
Q (*i*) : the size of the batch for batch-*i*
t (*i*) : processing time per unit for the *i*-th batch
X (*i*) *g* : a binary variable (0 or 1) to express the *i*-th batch contains the item or other items *g*

Batch scheduling problem on a single machine for multi item with the increased processing time can be formulated in the following equation:

Minimum $\sum_{i=1}^{N}(\sum_{j=1}^{i}(Q_{[j]}t_{[j]} + s_{[j]}) - s_{[i]})Q_{[i]}$

Constraints:

- $\sum_{i=1}^{N}(Q_{[i]}t_{[i]} + s_{[i]}) - s_{[N]} \leq d$
- $B_{[i]} + Q_{[i]}t_{[i]} = d$
- $\sum_{i=1}^{N} x_{[i]g}Q_{[i]} = D_g$, for $g = 1,2,\dots,G$
- $\sum_{g=1}^{G} x_{[i]g} = 1$ for $i = 1,2,\dots,N$
- $t_{[i]} = \sum_{g=1}^{G} x_{[i]g}\ [T_{[g]} + \delta_{[g]}(\sum_{k=1}^{i-1}(Q_{[k]}t_{[k]} + s_{[k]})]$, for $i = 1,2,\dots,N$
- $s_{[i]} = \sum_{g=1}^{G} x_{[i]g}\ S_g$, for $i = 1,2,\dots,N$
- $Q_{[0]} = 0,\ t_{[0]} = 0$, and $s_{[0]} = 0, Q_{[i]} \geq 0$
- $N \geq G$, and $i = 1,2,\dots,N$
- $x_{[i]g}\begin{cases}1, \text{if } batch\ i \text{ contains item } g \\ 0, \text{otherwise}\end{cases}$

5. *Batch* Scheduling Model with *Multi Item* PACO-Tabu Algorithm

Rajendran and Ziegler (2004) developed the ACO algorithm with local job-search index (ACO) for minimizing makespan. Tanti, et al (2009) proposed the hybrid of ACO-Tabu search that also minimizing makespan.The initial solution in ACO-TABU is obtained from the proposed algoritm by Nawaz, Enscore, and Ham (NEH algorithm). The result of two proposed algorithms shows that there is no difference between the hybrid ACO-Tabu and proposed ACO algorithm by Rajendran and Ziegler in term of makespan.

The steps of batch scheduling model with multi item the algorithm PACO-Tabu Algorithm are in the following:

Step 1: Obtain the initial solution of tabu search from NEH algorithm. NEH algorithm starts by taking the two jobs that have second highest of processing time. The alternatives of two jobs sequencing are calculated in term of tardiness and smallest mean tardiness will be selected. A job that has the third largest processing time will be inserted into the prefer sequence previously and look for all possible sequences that were with. After that, repeating taking a job until all jobs is scheduled. If there are several sequences that have the same value of tardiness, then count earliest and selected sequences that produce the largest earliest.

Step 2: Find the best solution by applying tabu search. A tabu search approaches that have been determined in this study are in the following:

- Random method to be used is a neighborhood block insertion.
- Tabu list size is set at seven, if the tabu list exceeds seven then the first order on the taboo list will be excluded from the list.
- Maximum number of iterations 100.

Step 3: After running the tabu search algorithm, followed by a sequence ant build. The beginning of ant construction sequence is to calculate the value of τ_{ik} , then proceed with the calculation of T_{ik}. Calculation τ_{ik} using the following rules:

- If (| position of job i seed sequence-k | +1) $\leq n/4$ then τ_{ik} = (1/ZBest)
- If $n/4 \leq$ (| Position of *job i seed sequence* –k|+1) $\leq$ $n/2$ then τ_{ik} = (1 / (2 * Zbest))
- Otherwise τ_{ik} = (1 / (4 * Zbest))

where τ_{ik} is the trail intensity of the scheduled job i in position k

Sequencing batch process begins with calculation of the value $T_{ik} = \sum_{k=1}^{k} \tau_{ik}$ After that generate the uniform random numbers u [0,1]. The conditions must be met after a random value u is as follows:

- If $u \leq 0.4$, then the batch i at position k of the seed sequence will be selected.
- If $0.4 \leq u \leq 0.8$, then the entire batch of selected batches that have not been scheduled to have the value of T_{ik} is the greatest.
- Otherwise the batch i is selected from the entire batch that has not been scheduled for the position of k is based on the calculated probability in the following:

$$P_{ik} = \left(\frac{T_{ik}}{\sum_{l} T_{lk}} \right)$$

where l is the whole batch of some batches that have not been scheduled. After calculation of probability then uniform random numbers is generated. If the random result is greater than the largest value of the probability P_{ik}, then performed a random repetition. Selected Batch is a batch that has a probability range corresponding to the result of random numbers.

Step 4: Perform calculations using tabu search algorithm

Step 5: Conducting the process of updating trail intensity by following the rules as follows:

If the number of job ≤ 40 then

- If (| position of job i seed sequence - k |)≤ 1 then $\tau_{ik}^{new} = \rho.\ \tau_{ik}^{old} + (1/diffZ_{current})$
- Otherwise $\tau_{ik}^{new} = \rho.\ \tau_{ik}^{old}$

If the number of job> 40

- If (| position of job i seed sequence - k |)≤ 2 then $\tau_{ik}^{new} = \rho.\ \tau_{ik}^{old} + (1/diffZ_{current})$
  Otherwise $\tau_{ik}^{new} = \rho.\ \tau_{ik}^{old}$

Where:

$diff$ = If (| position of *job i* in the best order $-k$|+1)$^{1/2}$

ρ = evaporation coefficient, its value is between 0-1

After the process of *updating trail intensities* then step 3 to 5 is repeated up to 40 cycles.

Step 6: Perform swap scheme to swap the positions of each batch i at position j.

Step 7: Calculate the mean tardiness of each swap scheme and compare the results with the mean tardiness of the combination of another number and size of the batch.

Step 8: Set the size and number of batches that have the smallest value of mean tardiness.

Step 9: Conducting the process of solving a batch of up to two times the number of jobs and repeat step 1 to step 8.

6. Experimental result

In this paper, the aims are to propose ACO-Tabu considering multi-item batch scheduling and to compare it to slack time method in term of mean tardiness and computational time. Data are acquired from data collection of an electrical company. There are five sets of problem generated uniformly at random consisting 3, 4, 5, 6 and 7 jobs. Each set has 100 instances. Simulation runs using Visual Basic software. The simulation result is compared statistically using paired t-test. The results show that for 3 jobs and 7 jobs, there are different between with batch and without batch in term of mean tardiness significantly. However, in term of computational time slack method without batch is better than with batch significantly. The simulation result using slack method is shown in Table 1. As follows:

Table 1. Simulation Results using Slack Time Method

Number of Job	Without Batch		With batch	
	Mean tardiness	Computational Time (s)	Mean tardiness	Computational Time (s)
3 *Job*	12691.7	0.5	10310.2	0.81
4 *Job*	9357.78	0.84	8965.89	2.51
5 *Job*	33 507	1.4	30765.4	345.77
6 *Job*	33207.9	1.8	27750.2	1601.2
7 *Job*	24887.1	2.1	20669	8111.2

The simulation result using PACO-Tabu is shown in Table 2. The results show that there are not different significantly using $\alpha=5\%$ for number of jobs 5 and 6. However, the computational time using PACO-Tabu without batch is better than with batch for all number of jobs. The result also clearly shows that PACO-Tabu without batch gives the better solution than slack method without batch in term of mean tardiness. In contrast, slack method gives the better computational time than PACO-Tabu.

Table 2. Simulation Results using PACO-Tabu

Number of Job	Without batch		With *Batch*	
	Mean tardiness	*Computational Time*	Mean tardiness	*Computational Time*
3 *Job*	7341.46	0.68	7333.27	1.44
4 *Job*	7542.5	1.3	7535.68	4.3
5 *Job*	239 938	1.88	23994.4	580.1
6 *Job*	21244.7	2.51	21150.3	2451.1
7 *Job*	16280.2	3.2	16259	15 375

7. Conclusion

Based on the experimental result it can be concluded that the PACO-Tabu without batch is more effective than with batch in minimizing mean tardiness. On the contrary, computational time of PACO-Tabu is worst than slack method. Further experimentation with large number of jobs used for the flowshop scheduling dependent setup might be useful.

References

Baptiste, P. (2000). Batching identical jobs. *Mathematical Methods of Operation Research*, 52, 355-367.

Chandra, Agus Susanto, Octavia, Tanti, Sahputra, Iwan Halim. (2009). Tabu search sebagai local search pada algoritma ant colony untuk penjadwalan flowshop. *Jurnal Teknik Industri,* 11(2), 188 – 194.

Dati-Lika Ben, Mosheiov Gur, Oron Daniel. (2009). Batch Scheduling on Two-Machine Flowshop with Machine-Dependent Setup Times. *Advances in Operations Research*.

Dorigo, M. (2002). The ant colony optimization metaheuristic: algorithm, application, and advances. In Glover Kochenberger, G., editors Handbook of Metaheuristics, *Vol. 57 of International Series in Operations Research and Management Science*, p.251-285.

Potts, C.N., and Kovalyov, M.Y. (2000). Scheduling with batching: a review. *European Journal of Operation Research*, 37, 798-804.

Rajendran, C.R., *and* Ziegler, H. (2004). Ant-Colony algorithm for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 155 (2), 426-438.

Shabtay, Dvir, Steiner, George. (2007). Single machine batch scheduling to minimize total completion time and resource consumption costs. *Springer,* 10:255-261.

Sukoyo, Samadhi TMA, Iskandar, B., Halim A.H. (2010). Model penjadwalan batch multi item dengan dependent setup time. *Jurnal Teknik Industri*, 12 (2), 69-77.

Thiesen, A. (1998). Design and evaluation of tabu search algorithms for multiprocessor scheduling. *Journal of Heuristics*,4, 141-160.

comparative study

ORIGINALITY REPORT

0%	0%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes	On	Exclude matches	< 10%
Exclude bibliography	On		