Local Search Methods to Find Approximate Solution for The Sum of Two Criteria with Unequal Ready Times in Machine Scheduling Problem

Tahani Jabbar Khraibet¹, Waleed Khalid Jaber², Lafta Abed Dawood³

¹Department of Mathematics, Thi-Qar Directorates of Education, Ministry of Education
tahani@utq.edu.iq
²Department of Mathematics, Computer Science and Mathematics, University of Thi-Qar, Iraq
waleedkhalidj@utq.edu.iq.
³Department of Mathematics, Thi-Qar Directorates of Education, Ministry of Education
laftaalsalihi@yahoo.com

Abstract
This research concentrates on the study of SA / TS which is regarded as two of the approaches of modern artificial intelligence, to solve the problems of scheduling by single machine to minimize bi-criteria \( \frac{1}{r_j} / \sum_{j=1}^{n} w_j (1 - e^{-r c_j}) + T_{max} \). This problem settled up to 30000 jobs.

Key words
Single machine scheduling, bi-criteria, Simulated Annealing, Tabu Search.

1. Introduction

We depend on the scheduling in separate jobs by a single machine with dates release dates to minimize the multiple objective problem. Our goal is to gain a schedule that minimize sum of discounted total weighted completion time and maximum tardiness, by using (Local search) method. This problem is denoted by \( \frac{1}{r_j} / \sum_{j=1}^{n} w_j (1 - e^{-r c_j}) + T_{max} \), and we actually believe that the researcher couldn’t deal with this problem yet.

It is minimizing of total discounted weighted completion time which it expand from the minimization of total weighted completion time \( \sum_{j=1}^{n} W_j c_j \). Rothkopf (1966) [1], Rothkopf and Smith (1984) [2] consider such time as \( 1 / \sum_{j=1}^{n} w_j (1 - e^{-r c_j}) \). Their
study was more inclusive cost function and discounted average of \( r \), \( 0 < r < 1 \) of the problem can be settled as P-type optimally in polynomial time by the Weighted Discounted Shorter Processing Time (WDSPT). The basis that schedules the jobs in non-decreasing order of rate: \( \frac{w_j e^{-rp_j}}{1-e^{-rp_j}} \) [3]. Wang et.a (2006) take the issue \( F_m / / \sum_j w_j (1-e^{-rc_j}) \) is NP-hard [4]. The problem \( 1 / / T_{\text{max}} \) is solution in \( O( n \log n ) \) time by Jackson’s [5] earliest due date (EDD) rule. Pinedo (2012) [3] showed that the optimal schedule for \( 1/r_j / T_{\text{max}} \) is strongly NP-hard.

A local search algorithm begins with an original (primary) solution and then tries constantly to add superior solutions through search neighborhoods. The algorithm traverses the arrangement space so that each solution visited is a neighbor of the previously one. A solution is called a local minimum with respect to a neighborhood function, if all its neighbors, the local search provides approach of high- quality solutions to NP- hard problems in a reasonable amount of time. This paper consists of these sections; Section 1 is formulation of the problem. Section 2 local search heuristics (SA and TS methods. Section 3 computational experience.

2. Formulation Problem

The actual difficulty of scheduling jobs in a schedule on a single machine to decrease the general that cost can be shown as: A set \( N=\{1,2,3,\ldots,n\} \) of \( n \) separate jobs which must be put in a schedule on a single machine for the purpose of reducing a certain criterion. This examination concerns the one machine scheduling problem which has a function of a different impartial feature resembled by \( 1/r_j / \sum_j w_j (1-e^{-rc_j}) + T_{\text{max}} \). The preemption in this problem is disallowed. There is no prior relation can be found between jobs and only one job \( j \) can be dealt with in a time. Each job \( j, j \in N \) has an integer processing time \( p_j \), a release date \( r_j \) and typically it ought to be finished at its due date, \( d_j \). Discounted total weighted completion time \( \sum_j w_j (1-e^{-rc_j}) \) and maximum tardiness \( T_{\text{max}} \), can be respectively defined as: \( \sum_j w_j (1-e^{-rc_j}) = \sum_j w_j (1-e^{-r\sum_j p_j}) \) and \( T_{\text{max}} = \max\{c_j - d_j, 0\} \), \( j = 1, \ldots, n \).

For a given sequence \( \sigma \) of the jobs, \( \sum_j w_j (1-e^{-rc_j}) \) and \( T_{\text{max}} \) are given by:
\[
\sum_{j=1}^{n} w_{\sigma(j)}(1 - e^{-rC_{\sigma(j)}}) = \sum_{j=1}^{n} w_{\sigma(j)}(1 - e^{-r\sum_{i=1}^{j} \mu_{\sigma(j)}}), j = 1, \ldots, n \tag{1}
\]

\[
T_{\text{max}(\sigma)} = \max\{C_{\sigma(j)} - d_{\sigma(j)}, 0\}, j = 1, \ldots, n \tag{2}
\]

The multiple objective function (MOF) \( \left( \sum_{j=1}^{n} w_j(1 - e^{-rC_j}) + T_{\text{max}} \right) \) is strongly NP-hard because the problem 1//\( \sum_{j=1}^{n} w_j(1 - e^{-rC_j}) \) is P-type [2] and the problem 1/\( r_j/T_{\text{max}} \) is strongly NP-hard [3]. The problem is to find a sequence \( \sigma \) that minimizes the total cost \( N \), this problem denoted by (P) could be indicated as follows:

\[
N = \min_{\sigma \in \mathcal{S}} \{Z(\sigma)\} = \min_{\sigma \in \mathcal{S}} \left( \sum_{j=1}^{n} w_j(1 - e^{-rC_j}) + T_{\text{max}} \right)
\]

s.t 
\[
\begin{align*}
C_{\sigma(j)} & \geq p_{\sigma(j)} & j = 1,2,\ldots,n \\
C_{\sigma(j)} & \geq r_{\sigma(j)} + p_{\sigma(j)} & j = 1,2,\ldots,n \\
C_{\sigma(j)} & \geq C_{\sigma(j-1)} + p_{\sigma(j)} & j = 1,2,\ldots,n \\
0 & < r < 1 & j = 1,2,\ldots,n \\
T_{\sigma(j)} & \geq C_{\sigma(j)} - d_{\sigma(j)} & j = 1,2,\ldots,n \\
r_{\sigma(j)} & \geq 0, w_{\sigma(j)} > 0, d_{\sigma(j)} \geq 0, p_{\sigma(j)} > 0 & j = 1,2,\ldots,n \\
\end{align*}
\]

while \( \sigma(j) \), resembles the occupation of job \( j \) in the ordering \( \sigma \) and \( \delta \) denotes the group of all schedules under accounting.

3. Simulated Annealing (SA) Method

Simulated annealing (SA) method is a device for getting the best or nearly the best settlement for problems of correspondent improvement, or problems which have separate features. It was supposed by Kirkpatrick et al. [6] and was fortunately applied to divide circles, places, and focusing in the real form of complete circles. Annealing is a process in where a physical item is heated to a high degree and progressively cooled down until the solid case is cleaned. SA shows its theory in the solid annealing process which is the same as settling the problem of incorporation improvement. Beginning from an initial sequence \( p \), a neighbor \( p' \) is gene averaged (generally arbitrarily) in a certain. At that point the distinction \( \Delta = F(P') - F(P) \), in the values of the objective function \( F \) is resolved. When \( \Delta \leq 0 \), sequence \( p' \) is recognized as new solution to continue the next iteration. For the circumstance \( \Delta > 0 \), sequence \( p' \) is recognized as new \( \exp(-\Delta/t_k) \), where \( t_k \) is a parameter known as the temperature, which depends on the iteration \( k \). The rule that defines \( t_k \) is called a
cooling schedule. Negeman (2001) [7] takes the parameter \( t_k \) decreases in \( k \); he sets \( t_1 \) equal to some positive value and \( t_k = \alpha^k t_1 \) \((0 < \alpha < 1)\). So that the probability to accept a worse solution is decreasing over time. A probable stopping criterion is a cycle of definitive neighborhood close to zero.

The following structure gives the outline of (SA) [8].

1. **Initialization:**
   - select (or generate randomly) initial solution (p)
   - put (p) as a current solution (s)
   - put (cost (s)) as a current value (f)
   - select initial temperature \( T > 0 \)

2. **While** termination criterion is not satisfied do
3. Generate (p’) neighborhood for (s).
4. Compute cost (p’).
5. \( \Delta = \text{cost(p’)} - \text{f} \)
6. If \( \Delta \leq 0 \)
7. \( s = p' \) (s ← p’ ) (accept the move).
8. \( f = \text{cost(p’)} \) (f ← cost (p’)) (change the current value).
9. Else if \( \exp(-\Delta/t_k) \geq \alpha \) \((0 < \alpha < 1)\).
10. \( s = p' \) (s ← p’ ) (accept the move)
11. end if .
12. end while.

4. **Tabu Search (TS) Method**

Tabu search, found by Fred W. Glover in 1986 [9], and designed in 1989, [10] is a metaheuristic study method involving local search devices used for mathematical items of TS, as its title reveals is its use of supple optimization. One of the main memory (tabu list) to tabu particular moves for a certain amount of time. In each repetition of TS, a move can be immediately spotted to the tabu list when it move is selected to move the study from the actual solution to its near solution. Tabu study improves the realization of these devices by involving memory forms that prescribes the emergent settlements or user-provided sets of basis [10], if an internal settlement
is clearly visited during a particular short-term passed or if it violates a rule, it is marked as "tabu" (disallowed). As a result the algorithm does not account for that probability repeatedly. The nearby size resembles the amount of possible settlements every repetition of the study process.

The following structure gives the outline of (TS)[8]:

1. Initialization:
   select (or generate randomly) initial solution (p)
   put (p) as a current solution (s)
   put (cost (s)) as a current value (f)
   generate a tabu list which contained only (s)

2. While termination criterion is not satisfied do
3. Generate (p') neighborhood for (s) (by insert or swap)
4. Compute cost (p')
   If  cost(p') < f
5.      s = p'   (s ← p' ) (accept the move)
6.      f = cost (p') (f ← cost (p' )) (change the current value)
7.      tabu list= tabu list + p'
8.     else if   cost(p') = f
9.        if   p'  \notin  tabu list
10.     s = p'   (s ← p' ) (accept the move)
11.    tabu list= tabu list + p'
12.   End
13.    Solution = s

5. Computational Results

The local search algorithms in this paper (Simulated Annealing and Tabu Search.) are coded in MATLAB 8.4.0.150421 (R2014b) and fulfilled on Intel (R) i7-4500UCPU @ 1.80 GHz,2.40Hz with RAM 8.00 GB (2045GB usable) individual computer. In our computation, we use the condition that: if the solution of an example with "n" jobs for any algorithm does not appear after (300) seconds i.e. (5 minutes ) from its run; then this example is unsolved and this algorithm is active until the problem of size "n". These criteria were used by Stoppler and Bierwrith [11].
6. Comparative Effective of Local Search Algorithms

Table (1) indicates the values of every local study algorithms and the number of times that everyone reach the typical value, where:

SA = the value got by Simulated Annealing.

TS = the value found by Tabu Search.

Av. Time = the average time algorithm for (10) examples.

Table (1): Shows comparison of the average the SA and TS for

\[ n \in \{5, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\} \]

| \( n \) | \( \text{Av. of SA} \) | \( \text{Av. of time} \) | \( \text{Av. of TS} \) | \( \text{Av. of time} \) |
|------|----------------|----------------|----------------|----------------|
| 5    | 581.41981      | 0.9118         | 570.0622       | 0.3284         |
| 10   | 1220.0817      | 0.9236         | 1159.3936      | 0.4149         |
| 20   | 2159.4633      | 0.9428         | 1898.5255      | 0.4217         |
| 30   | 3458.7013      | 0.9683         | 3140.122       | 0.4469         |
| 40   | 5344.9889      | 1.0281         | 5114.314       | 0.4581         |
| 50   | 7687.673       | 1.0178         | 7432.57        | 0.4816         |
| 100  | 19645.18       | 1.1259         | 18479.79       | 0.5783         |
| 200  | 45603.02       | 1.3348         | 43649.5        | 0.7642         |
| 300  | 73269.18       | 1.6732         | 68649.5        | 1.3742         |
| 400  | 120295.8       | 2.5042         | 116320.8       | 1.8137         |
| 500  | 162520.5       | 3.1256         | 155059.67      | 2.7066         |
| 1000 | 399585.8       | 5.6082         | 370324.8       | 5.1668         |

Table (2): Shows comparison of the average the SA and TS for

\[ n \in \{5000, 10000, 20000, 30000\} \]

| \( n \) | \( \text{Av. of SA} \) | \( \text{Av. of time} \) | \( \text{Av. of TS} \) | \( \text{Av. of time} \) |
|------|----------------|----------------|----------------|----------------|
| 5000 | 2616502.2      | 25.2192        | 2533502.3      | 27.6343        |
| 10000 | 5750837.4     | 73.6983        | 5450127.6      | 94.9584        |
| 20000 | 11684304.4    | 113.6768       | 11444124.6     | 134.6570       |
| 30000 | 17641925.8    | 257.9364       | 17271045.2     | 283.8264       |
Table (3): Shows activity of the local search methods

| Algorithm | Active until (maximum no. of jobs) |
|-----------|-----------------------------------|
| SA        | 30000                             |
| TS        | 30000                             |

**References**

[1] M. H. Rothkopf, “Scheduling Independent Tasks on Parallel Processors,” Management Science., vol. 12, pp. 437–447, 1966.

[2] M. H. Rothkopf and S.A. Smith, “There are no Undiscovered Priority Index Sequencing Rules for Minimizing Total Delay Costs,” Operations Research., vol. 32, pp. 451–456, 1984.

[3] M. L. Pinedo, “Scheduling theory algorithms and systems,” Springer Science Business Media, LLC., New York, 2012.

[4] W. Ji-Bo, S. Feng, J. Bo, and W. Li-Yan, “Permutation flow shop scheduling with dominant machines to minimize discounted total weighted completion time,” Applied Mathematics and Computation 182 947–954, 2006.

[5] J. R. Jackson, “Scheduling a Production Line to Minimize Maximum Tardiness,” Res. Report 43 Management science, Res. Project. University of California, Los Angeles, CA, 1955.

[6] A. Kirkpatrick, C.D. Gelatt Jr, and M. P. Vechi, “Optimization by simulated annealing,” Science., 220, 671-680,1983.

[7] E. G. Negenman, “Local Search Algorithms for the Multiprocessor Flow Shop Scheduling Problem,” European Journal of Operational Research., 128. 147-158, 2001.

[8] S. Tariq and J. Hussein, “Exact and Local Search Methods For Three Machine Flow Shop with Transportation Times,” AL- Mustansiriya J Sci., vol. 21, no. 7, 2010.

[9] W. Glover, “Future Paths for integer Programming and Links to Artificial Intelligence,” Computers and Operations Research., 13 (5): 533-549,1986, doi:10.1016/0305-0548(86)90048-1.

[10] W. Glover, “Tabu search – Part 2,” ORSA Journal on computing, 2(1): 4-32,1989 doi:10.1287/ijoc.2.1.4.190-206.

[11] S. Stöppler and S. Brierwirth, “The Application of parallel Genetic algorithm to the n/m/p/Cmax flow shop problem,” University of Bremen <C13,f@dhbrrz 41.bet.net>.