

Algoritma LPT-Branch and Bound Pada Penjadwalan Flexible Flowshop Untuk Meminimasi Makespan

Dana Marsetiya Utama

Jurusan Teknik Industri, Fakultas Teknik

Universitas Muhammadiyah Malang

E-mail Address : dana@umm.ac.id

Diterima : 26 Mei 2018 ; Disetujui: 15 Juni 2018

ABSTRAK

Penelitian ini membahas masalah penjadwalan flow shop untuk meminimasi makespan. Tujuan penelitian ini adalah mengembangkan algoritma LPT dan Branch And Bound (LPT- Branch And Bound) untuk meminimasi makespan. Metode yang dipakai adalah Longest Processing Time (LPT) dan Branch And Bound. Tahapan penyelesaian di bagi menjadi 3 bagian. Untuk membuktikan algoritma usulan, dilakukan percobaan numerik dengan membandingkan algoritma LPT-LN. Hasil percobaan numerik menunjukkan bahwa algoritma usulan LPT-Branch and Bound lebih efisien dibandingkan algoritma LPT-LN.

Kata Kunci : Flexible flowshop, LPT, Branch And Bound, Makespan

ABSTRACT

This research discussed the problem of flow shop scheduling to minimize the makespan. The purpose of this article is to develop the LPT and Branch And Bound (LPT-Branch And Bound) algorithms to minimize the makespan. The proposed method is Longest Processing Time (LPT) and Branch And Bound. Stage settlement is divided into 3 parts. To proved the proposed algorithm, a numerical experiment was conducted by comparing the LPT-LN algorithm. The result of the numerical experiment shows that LPT-Branch And Bound's proposed algorithm is more efficient than the LPT-LN algorithm.

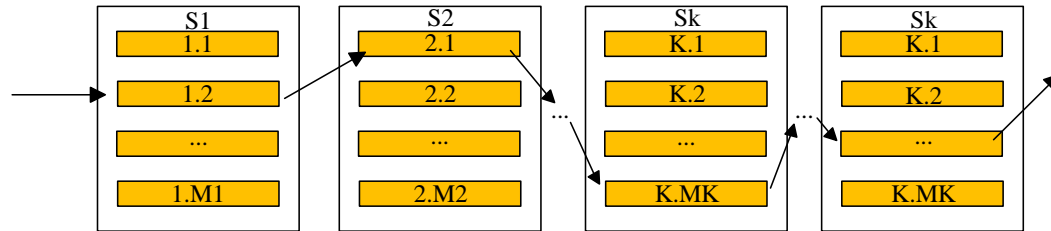
Keywords: Flexible Flowshop, LPT, Branch And Bound, Makespan

PENDAHULUAN

Penjadwalan merupakan pengalokasian *resources* atau mesin-mesin dalam melaksanakan sekumpulan tugas dalam jangka waktu tertentu [1]. Penjadwalan melibatkan pengerjaan sejumlah komponen yang sering disebut *Job*. *Job* merupakan elemen dasar yang bisa disebut aktivitas atau operasi. Tiap operasi ini membutuhkan alokasi sumber daya selama periode tertentu yang disebut waktu proses. Alokasi sumber daya yang sebelumnya disebut adalah seperti mesin, waktu tunggu, transportasi [2]. Tujuan melakukan penjadwalan adalah pesanan selesai tepat waktu dan meminimasi total biaya [3]. Umumnya, terdapat 2 klasifikasi umum diantaranya *flow shop* dan *job shop*. Aturan-aturan prioritas *Shortest Processing Time* (SPT), *Longest Processing Time* (LPT), *Earliest Due date* (EDD)[4] sering dipakai dalam penjadwalan *job shop* dan *flow shop*. Algoritma heuristik aktif dan *non delay* bagus digunakan di lingkungan *job shop* umumnya[5].

Penjadwalan *Flexible flowshop* merupakan generalisasi dari permasalahan *pure flowshop* dengan adanya beberapa mesin paralel setiap *stage* atau tahap dari suatu proses (gambar 1). Karakteristik Penjadwalan *Flexible flowshop* adalah apabila terdapat *m* mesin yang disusun secara seri dengan beberapa *stage*, yang didalam *stage* terdapat jumlah mesin identik yang disusun secara paralel. Masing-masing *job* akan diproses melewati *stage* 1

kemudian *stage 2* dan sampai *stage n*. Pada masing-masing *stage*, *job* akan diproses oleh salah satu mesin identik [6].



Gambar 1 Skema *Flexible Flowshop* [6].

Kriteria dalam penjadwalan umumnya menggunakan kriteria meminimasi makespan. makespan adalah total waktu yang dibutuhkan untuk menyelesaikan seluruh *job* [7]. Beberapa penelitian yang sudah dilakukan pada kasus *flexible flowshop* menunjukkan bahwa minimasi makespan menjadi masalah yang perlu untuk di optimalkan. Sebelumnya telah dilakukan penjadwalan *flexible flowshop* dengan menggunakan kombinasi LPT dan algoritma, pada kasus 2 *stage* untuk meminimasi makespan [8]. Penggunaan kombinasi metode LPT dan LN untuk meminimasi *makespan* pada kasus *n stage* [9]. Berdasarkan penelitian *pure flowshop*, algoritma *Branch and Bound* memberikan solusi makspan yang terbaik dibandingkan dengan algoritma lainnya [10].

Beberapa penelitian untuk meminimasi makespan telah banyak dilakukan. Sayangnya, penelitian tentang minimasi *makespan* kebanyakan masih dipakai untuk masalah *pure flowshop* dan minimasi *makespan*. Pada kasus *flexible flowshop* umumnya belum menghasilkan minimasi makespan yang optimal. Dari permasalahan diatas, penelitian ini bertujuan untuk mendapatkan algoritma baru yang dimodifikasi dari Algoritma LPT [11] dan Algoritma *Branch and bound* untuk mendapatkan makespan optimal untuk kasus *flexible flowshop*. Algoritma *Branch and Bound* dipilih karena memberikan solusi makespan yang terbaik dibandingkan dengan algoritma lainnya [10]. Algoritma LPT dan Algoritma *Branch and bound* diharapkan menjadi alternatif metode penjadwalan untuk kasus *flexible flowshop* [11]. Algoritma baru yang kami usulkan mempunyai keuntungan dapat meminimasi makespan dari seluruh *job* sehingga waktu penyelesaian *job* semakin cepat.

METODE

Umumnya algoritma *branch and bound* telah digunakan dalam kasus *pure flow shop problem*. Sedangkan penelitian ini termasuk dalam *flexible flow shop problem*. Peneliti telah mengembangkan algoritma *branch and bound* yang simple untuk *flexible flow shop problem*. Langkah-langkah penjadwalan *flexible flow shop problem* yang telah diusulkan terdapat 3 bagian antara lain:

1. Bagian 1
 - Membentuk lini *flow shop*
 - a. Langkah 1
 - Membentuk pasangan kelompok mesin (lini), masing-masing berisi satu mesin dari masing-masing mesin utama. F_1, F_2, \dots, F_p .
 - b) Langkah 2
 - Memberikan inialisasi waktu mulai pada setiap lini *flow shop* f_1, f_2, \dots, f_p adalah nol
 2. Bagian 2
 - Menetapkan job pada tiap lini *flow shop*.
 - c) Langkah 3
 - Gunakan metode LPT untuk menetapkan *job* untuk masing-masing kelompok mesin (*flow shop*).

Untuk setiap *job* J_j , $1 \leq j \leq n$, untuk menemukan total waktu *job* $tt_j = t_{1j} + t_{2j} + \dots + t_{mj}$

- d) Langkah 4
Urutkan *job* berdasarkan waktu pemrosesan terbesar ke waktu terkecil dari total waktu tiap *job* tt_j
- e) Langkah 5
Menempatkan *job* berdasarkan langkah 4 ke tiap-tiap lini *flowshop*. Penempatan berdasarkan beban terkecil di tiap lini *flow shop*.
- 3. Bagian 3 :
Mengurutkan penjadwalan dengan algoritma *Branch and Bound* yang telah dimodifikasi dari algoritma y [10]. Adapun Langkah-langkah utama dalam algoritma penjadwalan *Branch And Bound* dengan *Flexible Flowshop* adalah sebagai berikut:
 - f) Langkah 6
Membuat cabang dari alternatif jadwal parsial yang mungkin di tiap lini *flow shop*.
 - g) Langkah 7
Menghitung bi untuk tiap cabang dengan *job* $j = 1,2,..n$ dan mesin $i = 1,2,..m$. berdasarkan persamaan 1.
 - h) Langkah 8
Menghitung B_j tiap cabang berdasarkan persamaan 2.
 - i) Langkah 9
Menghitung M tiap cabang berdasarkan persamaan 3.
 - j) Langkah 10
Memilih cabang dari jadwal parsial yang memiliki M terkecil.
 - k) Langkah 11
Ulangi langkah 6,7,8,9 dan 10 sampai semua *job* di tiap lini terjadwal.

$$b_i = q_i + \sum_{J \in \sigma'} t_{ji} + \text{Min}_{J \in \sigma'} \left\{ \sum_{i=i+1}^m t_{ji} \right\} \quad (1)$$

$$B_j = \text{Max}_{i=1}^m \{b_i\} \quad (2)$$

$$M = \text{Min}_{J=1}^n \{B_j\} \quad (3)$$

Dimana :

bi = lower bound, adalah hasil perhitungan dari nilai batas bawah pada mesin i.

qi = jumlah *completion time* yang terakhir pada mesin i.

σ' = job yang belum dijadwalkan

tji = waktu proses *job* j pada mesin i.

tt_j = total waktu *job* j

t_{mj} = waktu proses mesin i pada pada *job* j.

m = banyaknya tahapan pengerjaan di mesin.

n = banyaknya *job*

$B_j = \max$ lower bound dari mesin i untuk $job j$.

$M = \min$ lower bound dari $job j$

HASIL DAN PEMBAHASAN

Contoh numerik

Contoh numerik telah dilakukan menggunakan kasus 4 job dengan 3 stage (mesin) dan tiap mesin (stage) memiliki 2 mesin identik. Data waktu proses tiap job dan mesin dapat dilihat tabel 1.

Tabel 1 Data waktu proses tiap job dan mesin

Mesin	Job			
	J1	J2	J3	J4
M1	3	2	1	4
M2	1	2	3	5
M3	2	5	4	2

Langkah-langkah penyelesaian Algoritma usulan LPT- *Branch and Bound* adalah

1. Bagian 1

a. Langkah 1

Membentuk lini flow shop mesin. Karena terdapat 2 mesin identik di tiap stage maka terdapat 2 lini *flow shop*.

Lini flow shop kita notasikan dengan $F1$ dan $F2$

$F1 = M1, M2, M3$ dan $F2 = M1, M2, M3$

b. Langkah 2

Inisialisasi waktu penyelesaian setiap *flowshop* $f1$ dan $f2$ nol.

2. Bagian 2

c. Langkah 3

Melakukan perhitungan total waktu tiap-tiap *job* seperti ditunjukkan pada tabel 2.

Tabel 2 Data total waktu proses tiap job

Mesin	Job			
	J1	J2	J3	J4
M1	3	2	1	4
M2	1	2	3	5
M3	2	5	4	2
Total Waktu Proses	6	9	8	11

d. Langkah 4

Urutkan waktu proses berdasarkan tabel 2 dari waktu proses terbesar ke terkecil (tabel 3).

Tabel 3 Urutan total waktu proses dari terbesar ke terkecil tiap *job*

Job	Total Waktu Proses
4	11
2	9
3	8
1	6

e. Langkah 5

Mencari waktu proses minimum F_i pada tiap lini *flowshop*. Karena semua waktu awal pemrosesan sama dengan 'no!', kemudian masukkan pada tabel alokasi penetapan *job* pada tiap-tiap *flowshop*, jumlah waktu yang sudah di masukkan dalam tabel, pilih total waktu terkecil untuk memasukkan *job* selanjutnya yang belum masuk dalam tabel hingga daftar pekerjaan kosong. Hasil alokasi *job* tiap *flow shop* dilihat pada tabel 4.

Tabel 4 Hasil alokasi *job* tiap *flow shop*

<i>Flowshop</i>	<i>Allocated jobs</i>	<i>Total processing time</i>
F1	J4 J1	17
F2	J2 J3	17

f. Langkah 6

Membuat cabang dari alternatif jadwal parsial tiap lini *flow shop*. Untuk *flow shop* F1 jadwal parsialnya adalah J4-X dan J1-X. Sedangkan Untuk *flow shop* F2 jadwal parsialnya adalah J2-X dan J3-X.

g. Langkah 7

Disini memberikan contoh perhitungan b_i berdasarkan persamaan 1 untuk jadwal parsial J4-X. nilai b_1 diperoleh dari nilai q_1 waktu penyelesaian pengerjaan *job* 4 di mesin 1. Nilai q_1 diperoleh 4. Dengan tahapan yang sama diperoleh nilai q_2 dan q_3 sebesar 9 dan 11. Kemudian menghitung waktu pengerjaan pekerjaan yang belum di jadwalkan pada mesin 1. Dan mencari nilai minimal dari jumlah waktu proses di mesin ke 2 dan 3 untuk masing-masing *job* yang belum di jadwalkan. Sehingga nilai b_1, b_2 dan b_3 adalah 10, 12 dan 13.

h. Langkah 8

Menghitung B_j pada jadwal parsial J4-X berdasarkan persamaan 2 adalah 13. Nilai B_j untuk jadwal parsial J1-X adalah 14.

i. Langkah 9

Berdasarkan jadwal parsial J4-X dan J1-X maka nilai M cabang (berdasarkan persamaan 3) ini adalah nilai jadwal parsial J4-X karena memiliki nilai yang kecil.

j. Langkah 10 dan 11

Ulangi langkah 6,7,8,9 dan 10 sampai semua *job* di tiap lini terjadwal. Rekapitulasi perhitungan tiap-tiap *flow shop* dapat dilihat pada tabel 5.

Tabel 5. Rekapitulasi perhitungan tiap-tiap *flow shop*

<i>Flow shop</i>	<i>Partial sequence</i>	q_1, q_2, q_3	b_1, b_2, b_3	B
F1	4x	4, 9, 11	10, 12, 13	13
	1x	3, 4, 6	14, 11, 8	14
F2	2x	2, 4, 9	10, 11, 13	13
	3x	1, 4, 8	10, 11, 13	13

Dari tabel 5 menunjukkan bahwa makespan pada *flow shop* F1 sebesar 13 dan *flow shop* F2 sebesar 13. Makespan untuk kasus ini adalah 13.

Percobaan Numerik

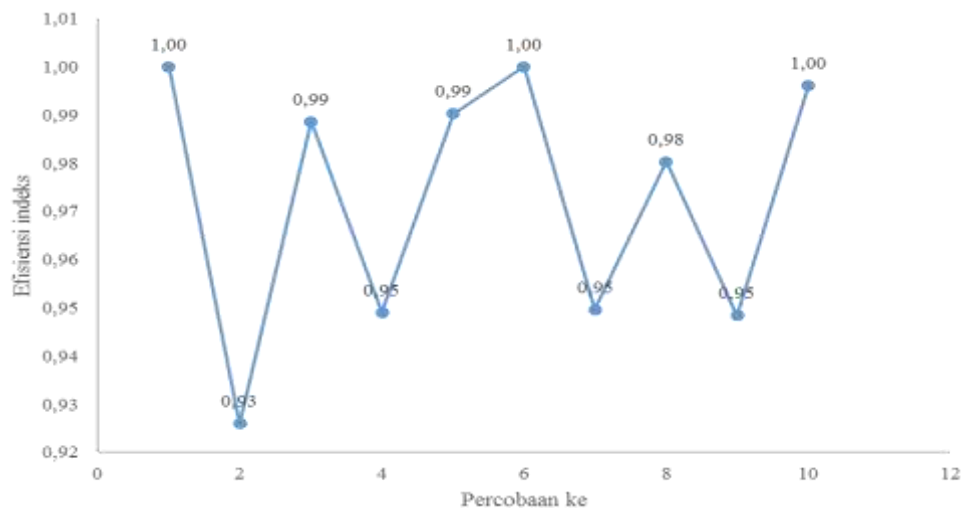
Percobaan numerik penjadwalan *flexible flow shop* algoritma usulan LPT- *Branch and Bound* (LPT-BB) juga dibandingkan dengan algoritma LPT-LN. Perbandingan metode LPT- *Branch and Bound* dan LPT-LN telah dilakukan untuk 10 variasi masalah penjadwalan. Dengan waktu proses berdistribusi *uniform* (1,100). Hasil percobaan numerik dapat dilihat pada tabel 6.

Tabel 6 Hasil perbandingan metode LPT- *Branch and Bound* dan LPT-LN

Percobaan	Jumlah job	Stage/ Mesin	Jumlah Lini Flow shop	Makespan	
				LPT-BB	LPT- LN
1	4	3	2	13	13
2	10	5	2	513	554
3	20	5	2	692	700
4	10	10	2	913	962
5	20	10	2	1014	1024
6	8	3	4	14	14
7	20	5	4	528	556
8	40	5	4	698	712
9	20	10	4	918	968
10	40	10	4	1020	1024

Dari Tabel 6 hasil perbandingan metode metode LPT- *Branch and Bound* dan LPT-LN menunjukkan bahwa metode yang telah diusulkan memberikan hasil total biaya yang makespan terkecil di beberapa percobaan numerik. Selain itu, setelah dilakukan pengukuran perhitungan *efficiency index* (EI). Perhitungan EI untuk melihat efisiensi metode terhadap metode lain yang sudah ada. Perhitungan EI seperti pada persamaan 4. Jika nilai EI < 1, maka algoritma usulan LPT- *Branch and Bound* memiliki kinerja yang lebih bagus. Jika nilai EI =1, maka algoritma usulan LPT- *Branch and Bound* memiliki kinerja yang sama bagusnya dengan algoritma LPT-LN. Jika nilai EI > 1, maka algoritma usulan LPT- *Branch and Bound* memiliki kinerja yang buruk dibandingkan algoritma LPT-LN. Berdasarkan gambar 1, nilai rata-rata EI sebesar 0,97. Hasil ini menggambarkan metode LPT- *Branch and Bound* yang diusulkan menghasilkan solusi yang lebih efektif dan efisien.

$$EI = \frac{\text{Makespan LPT_BB}}{\text{makespan LPT_LN}} \tag{4}$$



Gambar 2. Nilai EI terhadap tiap-tiap percobaan

KESIMPULAN

Dalam makalah ini, penulis membahas masalah penjadwalan *flexible flowshop* untuk meminimasi makespan. Penulis mengusulkan algoritma baru LPT- *Branch and Bound* untuk masalah tersebut. Algoritma usulan secara umum terdiri dari 3 bagian. Algoritma usulan kemudian dibandingkan dengan algoritma LPT-LN untuk mengetahui performansi dari algoritma. Percobaan numerik telah dilakukan dalam penelitian ini. Hasil percobaan numerik menunjukkan bahwa algoritma usulan LPT-*Branch And Bound* lebih efisien dibandingkan algoritma LPT-LN. Saran untuk penelitian selanjutnya adalah menggunakan algoritma usulan LPT-*Branch And Bound* sebagai solusi awal di algoritma metaheuristik.

DAFTAR PUSTAKA

- [1] K. Baker and D. Trietsch, *Principles of Sequencing and Scheduling*. John Wiley & Sons, 2013.
- [2] R. Ginting, *Penjadwalan Mesin*. Yogyakarta: Graha Ilmu, 2009.
- [3] D. Sipper and R. Bulfin, *Production: Planning, Control, and Integration*. 1997.
- [4] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Springer Science & Business Media, 2012.
- [5] S. Harto, A. K. Garside, A., and D. Utama, "Penjadwalan Produksi Menggunakan Algoritma Jadwal Non Delay Untuk Meminimalkan Makespan Studi Kasus di CV," *Bima Mebel. Spektrum Ind.*, vol. 14, 2016.
- [6] G. Kulcsar and F. Erdélyi, "Modeling and Solving of the Extended Flexible Flow Shop Scheduling Problem," *Prod. Syst. Inf. Eng.*, vol. 3, pp. 257–271, 2005.
- [7] K. Baker, *Introduction to Sequencing and Scheduling*. John Wiley & Sons, 1974.
- [8] P. Gilmore and R. Gomory, "Sequencing a one state-variable machine: A solvable case of the traveling salesman problem," *Oper. Res.*, vol. 12, pp. 655–679, 1964.
- [9] R. Logendran, N. Nudtasomboon, and N., "Minimizing the Makespan of a Group Scheduling Problem: a New Heuristic," *Int. J. Prod. Econ.*, vol. 22, pp. 217–230, 1991.
- [10] Z. Lomnicki, *A "Branch-and-Bound" Algorithm For the Exact Solution of the Three-Machine Scheduling Problem*. 1965.
- [11] C. Sriskandarajah and S. Sethi, "Scheduling Algorithms For Flexible Flowshops: Worst and Average Case Performance," *Eur. J. Oper. Res.*, vol. 43, pp. 143–160, 1989.

Algoritma LPT-*Branch and Bound* Pada Penjadwalan *Flexible Flowshop* Untuk Meminimasi Makespan / Dana Marsetiya Utama

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2018 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licences/by/4.0/>)