



Model Integer Programming untuk penugasan pekerjaan dengan waktu kedatangan dan keberangkatan yang berbeda

Andre Sugiyono¹, Dedy Kurniadi²

¹Islam Sultan Agung

Jl.Raya Kaligawe Km.4 Semarang (+6224 6583584), e-mail: andre@unissula.ac.id

¹Islam Sultan Agung

Jl.Raya Kaligawe Km.4 Semarang (+6224 6583584), e-mail: ddy.kurniadi@unissula.ac.id

ARTICLE INFO

History of the article :

Received 11 November 2022

Received in revised form 17 Januari 2023

Accepted 17 Januari 2023

Available online 30 Januari 2023

Keywords:

tugas, lokasi, pekerjaan

* Correspondence:

Telepon:

-

E-mail:

andre@unissula.ac.id

ABSTRACT

Makalah ini membahas masalah alokasi peti kemas ke lokasi galangan terminal kapal dimana peti kemas memiliki waktu kedatangan dan keberangkatan yang berbeda untuk meminimalkan total biaya penanganan peti kemas. Diasumsikan bahwa jika lokasi galangan yang dipilih lebih dekat ke titik dimana peti kemas diturunkan dari kapal, biaya keseluruhan akan berkurang. Masalah ini dapat didefinisikan sebagai masalah alokasi mesin pekerjaan di mana biaya pemrosesan pekerjaan pada setiap mesin berbeda tetapi tidak tergantung pada waktu pemrosesan dan pekerjaan yang tidak dapat diakhiri memiliki waktu kedatangan dan keberangkatan yang berbeda. Masalah ini dirumuskan dalam solusi eksak sebagai model pemrograman integer campuran. Hasil komputasi menunjukkan bahwa heuristik ketiga yang diusulkan efisien dalam memecahkan masalah lokasi dalam biaya yang berbeda, jumlah pekerjaan, dan waktu komputasi.

1. INTRODUCTION

Masalah penugasan pekerjaan dengan waktu kedatangan dan keberangkatan yang berbeda muncul ketika kita telah melihat masalah operasi terminal peti kemas. Pengoperasian container yard terdiri dari urutan operasi yang menangani material dari yard ke kapal untuk bongkar muat kapal. Operasi dapat dilakukan dengan cara berikut (Steenken, et.al, 2004): (a) operasi pembongkaran, Quay Cranes (QC) membongkar peti kemas dari kapal ke truk halaman internal; (b) truk halaman mendistribusikan kontainer ke terminal yang memilikinya dan menumpuknya di halaman; (c) truk halaman kosong melakukan perjalanan kembali ke QC untuk menerima pengiriman kontainer pelepas berikutnya dan (d) truk eksternal mengambil kontainer dan mengeluarkannya dari terminal.

Untuk operasi pemuatan, prosedur dilakukan dalam arah yang berbeda. Diperlukan strategi untuk mengoptimalkan operasi ini.

Ketika kapal peti kemas memanggil ke pelabuhan, peti kemas di atas kapal harus dibongkar dan disimpan di halaman sampai diangkat lebih lanjut dengan truk. Setelah peti kemas diturunkan dari kapal dan ditempatkan di truk, peti kemas dipindahkan ke halaman Bongkar Muat. Kontainer harus disimpan sedemikian rupa untuk meminimalkan jarak penanganan; perlu menempatkan peti kemas di tempat penyimpanan dan mengeluarkannya ketika truk eksternal datang untuk mengambil, sehingga peti kemas memiliki atribut sebagai waktu kedatangan dan waktu keberangkatan. Oleh karena itu, permasalahan yang diteliti adalah mencari jadwal yang layak untuk peti kemas untuk ditempatkan di area slot di halaman. Yang pekarangan memiliki biaya penanganan yang berbeda. Di sisi lain untuk meminimalkan total biaya area yang dihasilkan untuk kontainer yang dialokasikan.

Kami menganggap satu set kontainer sebagai satu set pekerjaan dan area slot sebagai mesin dalam hal masalah penjadwalan. Untuk setiap pekerjaan waktu kedatangan dan waktu keberangkatan diberikan. Masalah penataan peti kemas adalah mencari jadwal yang layak untuk peti kemas untuk ditempatkan di lokasi di lapangan peti kemas. Dimana suatu lokasi memiliki biaya jarak yang berbeda. Di sisi lain adalah untuk meminimalkan total biaya area untuk kontainer yang dialokasikan. Dalam hal ini setiap lokasi hanya dapat menyimpan satu pekerjaan pada satu waktu dan waktu penyimpanan setiap pekerjaan tidak dapat didahulukan. Waktu penyimpanan semua lokasi adalah sama, namun biaya yang terkait dengan penyimpanan setiap pekerjaan tidak tergantung pada pekerjaan yang diproses.

Kami menganggap penugasan pekerjaan dengan waktu kedatangan dan keberangkatan yang berbeda analog dengan n pekerjaan dan m terminologi mesin. (1) Asumsikan bahwa kita memiliki n pekerjaan (kontainer) dengan waktu kedatangan dan keberangkatan untuk diproses pada m mesin. (2) Setiap mesin hanya dapat memproses satu pekerjaan pada satu waktu dan pemrosesan setiap pekerjaan tidak dapat didahulukan. (3) Waktu pemrosesan semua mesin adalah sama, namun biaya yang terkait dengan proses setiap pekerjaan tidak tergantung pada pekerjaan yang diproses.

RESEARCH METHODS

Kami mendefinisikan tugas pekerjaan dalam dua definisi khusus yaitu:

Definisi 1: pekerjaan yang tumpang tindih

Kami mendefinisikan dua pekerjaan dengan waktu kedatangan (a_i, d_i) dan keberangkatan (a_j, d_j) dan sebagai pekerjaan yang tumpang tindih jika:

$$a_j \leq d_i \text{ atau } d_j \geq a_i$$

Dengan asumsi tanpa kehilangan pekerjaan i harus tiba sebelum pekerjaan j

Definisi 2: layak ditugaskan

Kami mendefinisikan mesin telah layak ditugaskan pekerjaan jika tidak ada pasangan pekerjaan yang tumpang tindih yang ditugaskan ke mesin itu.

Setelah kami menyajikan definisi masalah diatas, selanjutnya kami menyajikan solusi yang tepat untuk masalah minimalisasi tersebut menggunakan Pemrograman Integer Campuran (MIP) dan berbagai algoritma heuristik berdasarkan definisi 1 dan 2 diperkenalkan untuk menentukan solusi yang layak.

Mixed Integer Programming Model

Jenis alokasi pekerjaan dapat dirumuskan sebagai program matematika. Salah satu program matematika yang sering digunakan dalam masalah penjadwalan adalah program integer. Program integer (IP) mengacu pada masalah optimasi di mana variabel harus bilangan bulat. Jika hanya subset dari variabel yang diperlukan untuk menjadi bilangan bulat dan sisanya diperbolehkan untuk

menjadi nyata, maka masalah ini disebut sebagai program integer campuran (MIP). Pada bagian ini, perumusan MIP diberikan untuk menyelesaikan alokasi pekerjaan.

Misalkan ada n pekerjaan dan m mesin, Misalkan a_j dan d_j masing-masing adalah waktu kedatangan dan keberangkatan dari pekerjaan j . maka biarkan jika pekerjaan j ditugaskan ke mesin p dan o sebaliknya. Biarkan c_p menjadi biaya pemrosesan pekerjaan pada mesin p . Maka masalah meminimalkan biaya pemrosesan dapat dirumuskan sebagai IP sebagai berikut:

$$\min \sum_{j=1}^n \sum_{p=1}^m c_p x_{jp} \quad (1)$$

Bergantung pada :

$$\sum_{p=1}^m x_{jp} = 1, \forall j \quad (2)$$

$$d_j \leq a_k + Mu_{jk}, \text{Ml arg e}, \forall j \langle k \quad (3)$$

$$x_{jp} + x_{kp} \leq 2 - u_{jk}, \forall p, \forall j \langle k \quad (4)$$

$$x_{jp}, u_{jk} \in \{0,1\} \quad (5)$$

Kami berasumsi bahwa pekerjaan diurutkan menurut jumlah yang meningkat pada waktu kedatangan mereka. Kami juga berasumsi bahwa pekerjaan tiba sebelum keberangkatan mereka $a_j \langle d_j, \quad \forall_j = 1, \dots, j$ Pertimbangkan sepasang pekerjaan $i, j \in N$ dan $S_{ij} = (i, j \in N : a_j \geq a_i; d_i \leq d_j)$.

Dimana fungsi tujuan (1) menyatakan bahwa kita mencari jadwal yang layak untuk subset pekerjaan dengan biaya total minimum.

Hasil 1. Setiap solusi yang memenuhi kendala (3)-(4) dapat diinterpretasikan sebagai jadwal non-preemptive yang layak.

Validasi. Batasan (2) menunjukkan bahwa setiap pekerjaan perlu ditugaskan ke tepat satu mesin. Batasan (3) memastikan bahwa tidak ada tumpang tindih antara dua pekerjaan (j dan k). Sehingga pasangan pekerjaan j dan pekerjaan k layak ditugaskan jika waktu keberangkatan pekerjaan j kurang dari atau sama dengan waktu kedatangan pekerjaan k . Batasan (4) menentukan dua pekerjaan j dan k dapat ditugaskan ke mesin yang sama.

RESULTS

Di sini kami mendemonstrasikan contoh sederhana untuk solusi eksak dengan menggunakan pemrograman integer campuran. Misalkan terdapat 4 job dan 2 mesin kosong dengan waktu kedatangan dan waktu keberangkatan yang berbeda dapat dilihat pada tabel 1. Untuk atribut biaya pada tabel 2 kami memberikan biaya untuk mesin pertama adalah yang terkecil karena merupakan mesin terdekat. Kami mengembangkan matriks konflik berdasarkan dua definisi yang kami jelaskan di atas. Pada tabel 3 kita dapat melihat bahwa Pekerjaan pertama ($J1$) tumpang tindih dengan pekerjaan kedua ($J2$) dan pekerjaan ketiga ($J3$). Ini berarti bahwa mereka tidak dapat ditugaskan ke mesin yang sama.

Table 1. Simple Case 4 jobs 2 machines

Jobs	a_j	d_j
1	1	4
2	1	2
3	2	5
4	5	7

Table 2. Cost of machines

	M ₁	M ₂
Cost	1	2

Table 3. Conflict matrix

$d_j - a_{j+1}$	J_2	J_3	J_4
J_1	3	2	-1
J_2	-	0	-3
J_3		-	0
J_4			-

DISCUSSION

Pendekatan solusi yang dijelaskan dalam makalah ini diimplementasikan dengan Python. Kami menggunakan PULP dan GUROBI Ver. 4.5.2 untuk menyelesaikan semua formulasi MIP. Deskripsi MIP berikut ini:

```

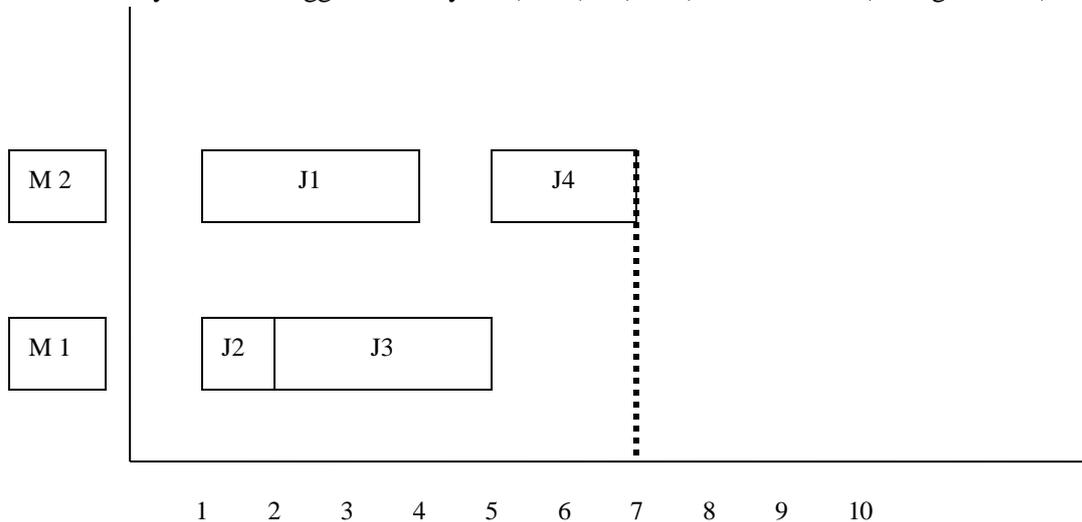
TITLE 4 jobs 2 machines
! Objective function
MAXIMIZE
x1 +x2 +x3 +x4 +2 x5 +2 x6 +2 x7 +2 x8
! Constraints
SUBJECT TO
    x1 +x2 <= 1
    x1 +x3 <= 1
    x1 +x4 <= 1
    2 x5 +2 x6 <= 1
    2 x6 +2 x7 <= 1
    2 x7 +2 x8 <= 1
    x1 +2 x5 = 1
    x2 +2 x6 = 1
    x3 +2 x7 = 1
    x4 +2 x8 = 1
END
! Integer definitions
GIN x1
GIN x2
    
```

Result

Table 4 Result from MIP

Variables	MILP Feasible	result
	4	4
x2	1	1
x3	1	1
x4	0	0
x5	0.5	0.5
x7	0	0
x1	0	0
x6	0	0
x8	0.5	0.5

Dari hasil yang ditunjukkan pada tabel 4 bahwa hanya variabel x_2 dan x_3 yang layak untuk ditugaskan ke mesin pertama, dan variabel x_1 dan x_4 tidak layak untuk ditugaskan ke mesin pertama dan akan ditugaskan ke mesin 2. Jika biaya mesin dikaitkan dengan proses setiap job yang tidak bergantung pada job yang diproses, yaitu M1 memiliki biaya mesin = 1 dan M2 memiliki biaya = 2, sehingga total biaya = $(2 \times 1) + (2 \times 2) = 6$ unit cost (lihat gambar 3).



Gambar 3 *Gantt chart* untuk penjadwalan awal.

Hasil dari menjalankan berbagai heuristik tersebut, memberikan solusi optimal untuk mesin simple case 4 job 2 .

CONCLUSIONS AND RECOMMENDATIONS

Kami telah menghadirkan berbagai solusi untuk penugasan pekerjaan dengan masalah waktu kedatangan dan keberangkatan yang berbeda. Dalam penelitian ini, evaluasi eksperimental menjadi perhatian, membuktikan kinerja mereka.

Penelitian ini membahas masalah penugasan pekerjaan dengan waktu kedatangan dan keberangkatan yang berbeda dengan identitas urutan operasi untuk pekerjaan yang tumpang tindih. Solusi tersebut menunjukkan bahwa hasil yang sangat baik diperoleh dari MIP sebagai solusi yang tepat meskipun memberikan waktu komputasi yang lebih lama. Berbagai heuristik juga kompetitif dalam hasilnya.

REFERENCES

- [1] Billionet, A., M. C. Costa, A. Sutter. 1992. An efficient algorithm for a task allocation problem. *Journal Assoc. Comput. Machinery* 39 502-518.
- [2] Chen, W., C. Lin. 2000. A hybrid heuristic to solve a task allocation problem. *Eur. Journal. Operations research.* 27 287-303.
- [3] Dutta, A., G. Koehler, A. Whinston. 1982. On optimal allocation in a distributed processing environment. *Management Sci.* 28 839-853.
- [4] Ernst, A. T., H. Jiang, M. Krishnamoorthy, B. Owens, D. Sier. 2004. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research* 127 21-144.
- [5] Ernst, A. T., H. Jiang, M. Krishnamoorthy. 2006. Exact solutions to task allocation problems. *Management Sci.* 52 1634-1646.

-
- [6] Hamam, Y., K. S. Hindi 2000. Assignment of program modules to processors: A simulated annealing approach. *Eur. J. Oper. Res.* 122 509-513.
 - [7] Kolen, A.W.J, J.K. Lenstra, C.H. Papadimitriou, F.C.R. Spijksma. 2007. Interval scheduling: a survey. *Naval research logistics.* 54(5) 530-543.
 - [8] Kopidakis, Y., M. Laman, V. Zissimopoulos. 1997. On the task assignment problem: two new efficient heuristic algorithms. *Journal parallel Distributed Comput.* 42 21-29.
 - [9] Lo, V. M. 1988. Heuristic algorithms for task assignment in distributed systems. *IEEE Trans. Comput.* 37 1384-1397.
 - [10] Ma, P. Y. R., E. Y. S. Lee, M. T. Tsuchiya. 1982. A task allocation model for distributed computing systems. *IEEE Trans. Comput.* C-31 41-47.
 - [11] Milis, I. 1995. Task assignment in distributed systems using network flow methods. *Computer Sci*, Vol. 1120. Springer-Verlag, London, UK, 396-405.
 - [12] Sadykov, R., L. A. Wosley. 2006. Integer programming and constraint programming in solving a multimachine assignment scheduling problem with deadlines and release dates. *INFORMS Journal on computing* 18 209-217.
 - [13] Sarje, A. K., G. Sagar. 1991. Heuristic model for task allocation in distributed computer systems. *IEE Proc.* E-138 313-318.
 - [14] Steenken, D., S. VoB, R. Stahlback. 2004. Container terminal operation and operations research. *OR Spectrum* 26 3-49.