

Teknik Relaksasi Lagrange untuk Penjadwalan Pekerjaan Majemuk dengan Penggunaan Sumberdaya Simultan

Suprayogi¹, Valentine²

Abstract: This paper discusses the multiple jobs scheduling problem with simultaneous resources. The problem involves one or more jobs with each job consists of a set of operations. Each operation is performed by more than one resources simultaneously. Number of units of each resource used for performing an operation is one or more units. The problem deals with determining a schedule of operations minimizing total weighted tardiness. In this paper, solution techniques based on Lagrangian relaxation are proposed. In general, the Lagrangian relaxation technique consists of three parts run iteratively, i.e., (1) solving individual job problems, (2) obtaining a feasible solution, and (3) solving a Lagrangian dual problem. For solving the individual job problems, two approaches are applied, i.e., enumeration and dynamic programming. In this paper, the Lagrangian relaxation technique using the enumeration and dynamic programming approaches are called RL1 and RL2, respectively. The solution techniques proposed are examined using a set of hypothetical instances. Numerical experiments are carried out to compare the performance of RL1, RL2, and two others solution techniques (optimal and genetic algorithm techniques). Numerical experiments show that RL2 is more efficient compared to RL1. In terms of the solution quality, it is shown that RL2 gives same results compared to the optimal technique and genetic algorithm. However, both RL2 and genetic algorithm can handle larger problems efficiently.

Keywords: Scheduling problem; Multiple jobs; Simultaneous resource; Lagrangian relaxation.

Pendahuluan

Makalah ini membahas masalah penjadwalan pekerjaan majemuk dengan sumberdaya simultan. Masalah penjadwalan melibatkan lebih dari satu pekerjaan dengan tiap pekerjaan terdiri atas satu himpunan operasi. Tiap operasi memerlukan pengerjaan secara simultan oleh lebih dari satu jenis sumberdaya. Jumlah unit dari tiap sumberdaya yang digunakan untuk pengerjaan suatu operasi adalah satu atau lebih. Masalah terkait dengan penentuan jadwal tiap operasi untuk tiap pekerjaan yang meminimumkan total *tardiness* tertimbang (*total weighted tardiness*).

Masalah penjadwalan pekerjaan yang dibahas dalam makalah ini termasuk dalam masalah penjadwalan pekerjaan dengan penggunaan sumberdaya simultan. Beberapa literatur yang membahas masalah penjadwalan ini antara lain adalah Dobson dan Karmarkar [1], Dobson dan Khosla [2], Suprayogi dan Mardiono [3], Luh *et al.* [4], Toha dan Halim [5], Suprayogi dan Toha [6], Suprayogi dan Partono [7, 8], Suprayogi *et al.* [9], Wu dan Chien [10] dan Wu *et al.* [11].

Suprayogi *et al.* [9] telah membahas masalah ini dan merumuskannya ke dalam model pemrograman linier bilangan bulat. Algoritma genetika telah diusulkan untuk memecahkannya.

Karena sifat dari probabilitiknya, algoritma genetika perlu dijalankan dalam sejumlah replikasi untuk mencari solusi terbaik. Dalam makalah ini, teknik pemecahan berbasis relaksasi Lagrange diusulkan sebagai teknik pemecahan. Teknik relaksasi Lagrange merupakan teknik yang efektif untuk memecahkan masalah optimisasi kombinatorial yang sulit (*hard combinatorial optimization problem*) seperti diungkapkan oleh Kaskavelis dan Caramanis [12, 13].

Masalah Penjadwalan Pekerjaan

Definisi masalah

Definisi masalah penjadwalan pekerjaan yang dibahas dalam makalah ini diambil dari definisi masalah yang terdapat dalam Suprayogi *et al.* [9]. Definisi masalah penjadwalan diuraikan berikut ini.

Misal terdapat suatu himpunan \mathcal{N} terdiri atas n pekerjaan yang dinyatakan dengan $\mathcal{N} = \{1, 2, \dots, n\}$. Tiap pekerjaan $i \in \mathcal{N}$, terdiri atas suatu himpunan \mathcal{N}_i yang terdiri atas n_i operasi yang dinyatakan

¹ Kelompok Keahlian Sistem Industri dan Tekno Ekonomi, Fakultas Teknologi Industri, Institut Teknologi Bandung, Bandung, Indonesia. Email: yogi@mail.ti.itb.ac.id

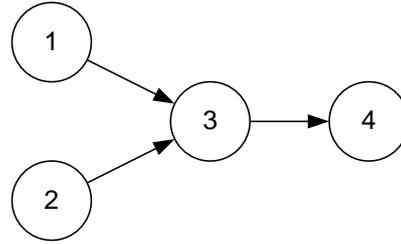
² Program Studi Sarjana Teknik Industri, Fakultas Teknologi Industri, Institut Teknologi Bandung, Bandung, Indonesia. Email: valentine.kosasih@gmail.com

dengan $\mathcal{N}_i = \{1, 2, \dots, n_i\}$. Untuk tiap pekerjaan $i \in \mathcal{N}$, himpunan operasi yang mendahului operasi $j \in \mathcal{N}_i$ dinyatakan dengan \mathcal{P}_{ij} . Tiap pekerjaan $i \in \mathcal{N}$ memiliki saat tenggat (*due date*) yang dinyatakan dengan d_i dan tingkat kepentingan dinyatakan dengan w_i . Misal \mathcal{H} menyatakan himpunan sumberdaya dan $\mathcal{H}_{ij} \subseteq \mathcal{H}$ menyatakan himpunan jenis sumberdaya yang digunakan untuk mengerjakan operasi $j \in \mathcal{N}_i$ untuk pekerjaan $i \in \mathcal{N}$. Waktu pengerjaan tiap operasi $j \in \mathcal{N}_i$ untuk pekerjaan $i \in \mathcal{N}$ dinyatakan dengan t_{ij} ($t_{ij} \geq 0$ dan bilangan bulat). Horizon waktu didiskretisasi yang terdiri atas t slot waktu dengan himpunan slot waktu dinyatakan dengan \mathcal{K} , yaitu: $\mathcal{K} = \{1, 2, \dots, t\}$. Jumlah sumberdaya $h \in \mathcal{H}_{ij}$ yang digunakan operasi $j \in \mathcal{N}_i$ untuk pekerjaan $i \in \mathcal{N}$ dinyatakan dengan q_{ijh} ($q_{ijh} \geq 0$ dan bilangan bulat). Ketersediaan sumberdaya $h \in \mathcal{H}$ pada slot waktu $k \in \mathcal{K}$ direpresentasikan dengan notasi m_{hk} ($m_{hk} \geq 0$ dan bilangan bulat). Masalah penjadwalan terkait dengan penentuan jadwal pengerjaan tiap operasi $j \in \mathcal{N}_i$ untuk tiap pekerjaan $i \in \mathcal{N}$ pada tiap sumberdaya $h \in \mathcal{H}_{ij}$, yaitu saat mulai B_{ijh} ($B_{ijh} \geq 0$ dan bilangan bulat) dan saat selesai C_{ijh} ($C_{ijh} \geq 0$ dan bilangan bulat) yang meminimumkan total *tardiness* tertimbang dengan memperhatikan pembatas ketergantungan operasi dan ketersediaan sumberdaya.

Asumsi-Asumsi

Asumsi-asumsi yang dalam masalah penjadwalan pekerjaan yang dibahas adalah sebagai berikut: (1) setiap pekerjaan $i \in \mathcal{N}$ adalah saling independen, (2) pengerjaan tiap operasi $j \in \mathcal{N}_i$ untuk tiap pekerjaan $i \in \mathcal{N}$ tidak dapat diinterupsi, (3) untuk tiap operasi $j \in \mathcal{N}_i$ dalam tiap pekerjaan $i \in \mathcal{N}$ yang menggunakan sumberdaya $h \in \mathcal{H}_{ij}$, lama waktu pengerjaannya adalah sama sebesar t_{ij} , (4) setiap unit dari tiap jenis sumberdaya $h \in \mathcal{H}$ adalah identik, (5) setiap pekerjaan $i \in \mathcal{N}$ digambarkan dalam bentuk jaringan dengan operasi pada simpul dan hubungan ketergantungan antar operasi membentuk suatu jaringan berarah tak bersiklus dengan setiap pekerjaan hanya memiliki satu operasi terakhir, (6) tiap pekerjaan siap dijadwalkan pada saat $k = 1$, (7) horizon waktu \mathcal{K} adalah cukup panjang sehingga semua pekerjaan dapat dijadwalkan.

Gambar 1 menunjukkan contoh jaringan dari suatu pekerjaan. Untuk contoh pekerjaan ini, terdapat empat operasi. Operasi 1 dan 2 adalah operasi-operasi pertama yang merupakan operasi tanpa pendahulu. Operasi 4 adalah operasi terakhir.



Gambar 1. Contoh jaringan dari suatu pekerjaan

Model Pemrograman Bilangan Bulat

Perumusan model pemrograman bilangan bulat untuk masalah penjadwalan pekerjaan memerlukan pendefinisian variabel-variabel keputusan tambahan sebagai berikut. Variabel-variabel keputusan binier X_{ijhk} dengan $X_{ijhk} = 1$ menunjukkan bahwa operasi $j \in \mathcal{N}_i$ untuk pekerjaan $i \in \mathcal{N}$ menggunakan sumberdaya $h \in \mathcal{H}_{ij}$ pada slot waktu $k \in \mathcal{K}$ dan $X_{ijhk} = 0$ untuk kondisi sebaliknya. Variabel-variabel keputusan tak negatif B_{ijh} dan C_{ijh} yang masing-masing menunjukkan saat mulai dan saat selesai pengerjaan operasi $j \in \mathcal{N}_i$ untuk pekerjaan $i \in \mathcal{N}$ dengan sumberdaya $h \in \mathcal{H}_{ij}$. Variabel-variabel keputusan C_{ij} yang menunjukkan saat selesai pengerjaan operasi $j \in \mathcal{N}_i$ untuk pekerjaan $i \in \mathcal{N}$ dengan C_{i,n_i} menunjukkan saat selesai pengerjaan operasi terakhir untuk pekerjaan $i \in \mathcal{N}$. Variabel-variabel keputusan C_i merepresentasikan saat selesai pengerjaan pekerjaan $i \in \mathcal{N}$. Model pemrograman bilangan bulat untuk masalah penjadwalan pekerjaan dengan horison waktu diskret dalam suatu himpunan slot waktu $\mathcal{K} = \{1, 2, \dots, t\}$ dinyatakan sebagai berikut (disebut model \mathbb{P}):

Meminimumkan

$$J = \sum_{i \in \mathcal{N}} w_i \max\{C_i - d_i, 0\} \tag{1}$$

dengan pembatas-pembatas:

$$C_i = C_{i,n_i}; \forall i \in \mathcal{N} \tag{2}$$

$$C_{ij} \geq C_{il} + t_{ij}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall l \in \mathcal{P}_{ij} \tag{3}$$

$$C_{ij} \geq t_{ij}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i \tag{4}$$

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} q_{ijh} X_{ijhk} \leq m_{hk}; \forall h \in \mathcal{H}_{ij}, \forall k \in \mathcal{K} \tag{5}$$

$$B_{ij} = C_{ij} - t_{ij} + 1; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i \tag{6}$$

$$B_{ijh} = B_{ij}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \tag{7}$$

$$C_{ijh} = C_{ij}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \tag{8}$$

$$X_{ijhk} = \begin{cases} 1 & ; B_{ijh} \leq k \leq B_{ijh} + t_{ij} - 1 \\ 0 & ; k \text{ yang lain} \end{cases}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \tag{9}$$

$$C_i \geq 0; \forall i \in \mathcal{N} \tag{10}$$

$$B_{ij}, C_{ij} \geq 0; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i \tag{11}$$

$$B_{ijh}, C_{ijh} \geq 0; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \tag{12}$$

$$X_{ijhk} \in \{0, 1\}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij}, \forall k \in \mathcal{K} \tag{13}$$

Persamaan (1) menunjukkan fungsi tujuan yang diminimumkan, yaitu jumlah dari *tardiness* tertimbang. Tiap persamaan (2) mendefinisikan saat selesai tiap pekerjaan yang bersesuaian dengan saat selesai operasi terakhir dari pekerjaan bersangkutan. Tiap pertidaksamaan (3) merepresentasikan pembatas hubungan ketergantungan untuk tiap operasi dari suatu pekerjaan. Tiap pertidaksamaan (4) menjamin bahwa saat selesai untuk tiap operasi dari tiap pekerjaan lebih besar atau sama dengan waktu pengerjaan. Tiap pertidaksamaan (5) merupakan pembatas ketersediaan tiap sumberdaya pada tiap slot waktu. Pembatas ini menjamin bahwa jumlah sumberdaya $h \in \mathcal{H}_{ij}$ yang digunakan untuk pengerjaan operasi $j \in \mathcal{N}_i$ untuk pekerjaan $i \in \mathcal{N}$ pada slot waktu $k \in \mathcal{K}$ tidak melebihi ketersediaan dari tiap jenis sumberdaya. Tiap persamaan (6) menunjukkan hubungan antara saat mulai dan saat selesai dari tiap operasi untuk masing-masing pekerjaan. Tiap persamaan (7) mendefinisikan saat mulai pengerjaan untuk operasi dari pekerjaan tertentu adalah sama dengan saat mulai pada tiap jenis sumberdaya yang digunakan. Tiap persamaan (8) mendefinisikan saat selesai untuk operasi dari pekerjaan tertentu adalah sama dengan saat selesai pada tiap jenis sumberdaya yang digunakan. Tiap persamaan (9) menunjukkan hubungan antara variabel-variabel keputusan X_{ijhk} dan B_{ijh} , yaitu $X_{ijhk} = 1$ untuk $B_{ijh} \leq k \leq B_{ijh} + t_{ij} - 1$ dan $X_{ijhk} = 0$ untuk k yang lain. Pembatas-pembatas untuk nilai variabel keputusan ditunjukkan pada pembatas-pembatas (10)-(13).

Dari model \mathbb{P} , jika B_{ij} diketahui maka C_{ij} dapat ditentukan berdasarkan persamaan (6). Selanjutnya, berdasarkan persamaan (2), $C_i = C_{i,n_i}$. Dengan demikian, model \mathbb{P} dapat disederhanakan menjadi:

Meminimumkan

$$J = \sum_{i \in \mathcal{N}} w_i \max\{C_{i,n_i} - d_i, 0\} \quad (14)$$

dengan pembatas-pembatas:

$$B_{ij} \geq B_{il} + t_{ij}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall l \in \mathcal{P}_{ij} \quad (15)$$

$$B_{ij} \geq 1; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i \quad (16)$$

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} q_{ijh} X_{ijhk} \leq m_{hk}; \forall h \in \mathcal{H}_{ij}, \forall k \in \mathcal{K} \quad (17)$$

$$B_{ijh} = B_{ij}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \quad (18)$$

$$X_{ijhk} = \begin{cases} 1 & ; B_{ijh} \leq k \leq B_{ijh} + t_{ij} - 1 \\ 0 & ; k \text{ yang lain} \end{cases} \quad (19)$$

$$; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \quad (19)$$

$$C_{i,n_i} \geq 0; \forall i \in \mathcal{N} \quad (20)$$

$$B_{ij} \geq 0; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i \quad (21)$$

$$B_{ijh} \geq 0; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \quad (22)$$

$$X_{ijhk} \in \{0, 1\}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij}, \forall k \in \mathcal{K} \quad (23)$$

Model \mathbb{P} merupakan rumusan model pemrograman bilangan bulat. Berdasarkan persamaan (14), model

ini merupakan model tak linier. Selain itu, model ini juga masih bersifat implisit yang ditunjukkan pada persamaan (19). Model pemrograman linier bilangan bulat yang dinyatakan secara eksplisit telah dirumuskan oleh Suprayogi *et al* [9]. Model pemrograman linier bilangan bulat yang dikembangkan merupakan model penjadwalan dengan horison waktu diskret dan merupakan pengembangan lanjutan dari model-model yang dikembangkan oleh Suprayogi dan Toha [6] dan Suprayogi dan Partono [7, 8]. Semua model yang dikembangkan tersebut mengacu pada model yang dikembangkan oleh Pritsker *et al.* [23] dan Morton and Pentico [24].

Model pemrograman linier bilangan bulat yang dirumuskan oleh Suprayogi *et al* [9] dapat dipecahkan untuk menghasilkan solusi optimal global dengan menggunakan piranti lunak komersial yang tersedia. Namun, pemecahan model pemrograman linier bilangan bulat ini memiliki kelemahan dalam hal keterbatasannya untuk memecahkan masalah yang kecil (yaitu, jumlah pekerjaan dan operasi yang sedikit dan horison waktu yang pendek) sehingga sulit diterapkan untuk kasus nyata. Untuk itu, teknik relaksasi Lagrange diusulkan sebagai teknik pemecahan masalah.

Model Relaksasi Lagrange

Misal diberikan pengali Lagrange $\pi_{hk} \geq 0$ tiap $h \in \mathcal{H}$ dan tiap $k \in \mathcal{K}$. Bentuk terelaksasi dari model \mathbb{P} adalah sebagai berikut (disebut model \mathbb{R}):

Meminimumkan

$$J' = \sum_{i \in \mathcal{N}} w_i \max\{C_{i,n_i} - d_i, 0\} + \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \pi_{hk} \left(\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} q_{ijh} X_{ijhk} - m_{hk} \right) \quad (24)$$

dengan pembatas-pembatas:

$$B_{ij} \geq B_{il} + t_{ij}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall l \in \mathcal{P}_{ij} \quad (25)$$

$$B_{ij} \geq 1; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i \quad (26)$$

$$B_{ijh} = B_{ij}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \quad (27)$$

$$X_{ijhk} = \begin{cases} 1 & ; B_{ijh} \leq k \leq B_{ijh} + t_{ij} - 1 \\ 0 & ; k \text{ yang lain} \end{cases} \quad (28)$$

$$; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \quad (28)$$

$$C_{i,n_i} \geq 0; \forall i \in \mathcal{N} \quad (29)$$

$$B_{ij} \geq 0; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i \quad (30)$$

$$B_{ijh} \geq 0; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij} \quad (31)$$

$$X_{ijhk} \in \{0, 1\}; \forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i, \forall h \in \mathcal{H}_{ij}, \forall k \in \mathcal{K} \quad (32)$$

Dengan hanya melibatkan fungsi tujuan, bentuk dual dari model \mathbb{R} (dinyatakan sebagai model Lagrange dual \mathbb{D}) adalah:

Memaksimumkan

$$L = - \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \pi_{hk} m_{hk} + \min_{\{B_{ij}\}} \sum_{i \in \mathcal{N}} \left(w_i \max\{C_{i,n_i} - d_i, 0\} + \sum_{j \in \mathcal{N}_i} \sum_{h \in \mathcal{H}_{ij}} \sum_{k \in \mathcal{K}} q_{ijh} \pi_{hk} X_{ijhk} \right) \quad (33)$$

dengan pembatas-pembatas: (25)-(32) dan $\pi_{hk} \geq 0$ ($\forall h \in \mathcal{H}, \forall k \in \mathcal{K}$).

Untuk tiap $i \in \mathcal{N}$, $j \in \mathcal{N}_i$ dan $h \in \mathcal{H}_{ij}$, $X_{ijhk} = 1$ jika $B_{ij} \leq k \leq B_{ij} + t_{ij} - 1$ dan $X_{ijhk} = 0$ untuk k yang lain. Oleh karena itu, $\pi_{hk} X_{ijhk} = \pi_{hk}$ untuk tiap $i \in \mathcal{N}$, $j \in \mathcal{N}_i$, $h \in \mathcal{H}_{ij}$ dan $B_{ij} \leq k \leq B_{ij} + t_{ij} - 1$. Dengan demikian model \mathbb{D} dapat dinyatakan kembali dengan:

Memaksimumkan

$$L = - \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \pi_{hk} m_{hk} + \min_{\{B_{ij}\}} \sum_{i \in \mathcal{N}} \left(w_i \max\{C_{i,n_i} - d_i, 0\} + \sum_{j \in \mathcal{N}_i} \sum_{h \in \mathcal{H}_{ij}} \sum_{k \in B_{ij}^{B_{ij}+t_{ij}-1}} \pi_{hk} q_{ijh} \right) \quad (34)$$

dengan pembatas-pembatas: (25)-(32) dan $\pi_{hk} \geq 0$ ($\forall h \in \mathcal{H}, \forall k \in \mathcal{K}$).

Karena pekerjaan-pekerjaan adalah saling independen, berdasarkan Luh *et al.* [15] dan Hoitomt *et al.* [18], minimum dari penjumlahan adalah jumlah dari minima. Dengan demikian, model \mathbb{D} dapat dinyatakan dengan:

Memaksimumkan

$$L = - \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \pi_{hk} m_{hk} + \sum_{i \in \mathcal{N}} \min_{\{B_{ij}\}} \left(w_i \max\{C_{i,n_i} - d_i, 0\} + \sum_{j \in \mathcal{N}_i} \sum_{h \in \mathcal{H}_{ij}} \sum_{k \in B_{ij}^{B_{ij}+t_{ij}-1}} \pi_{hk} q_{ijh} \right) \quad (35)$$

dengan pembatas-pembatas: (25)-(32) dan $\pi_{hk} \geq 0$ ($\forall h \in \mathcal{H}, \forall k \in \mathcal{K}$).

Komponen kedua dari ruas kanan dalam persamaan (35) pada dasarnya merupakan permasalahan untuk tiap pekerjaan individual $i \in \mathcal{N}$ yang dapat dinyatakan sebagai berikut (disebut Model \mathbb{R}_i):

Meminimumkan

$$L_i = w_i \max_{B_{ij}^{B_{ij}+t_{ij}-1}} \{C_{i,n_i} - d_i, 0\} + \sum_{j \in \mathcal{N}_i} \sum_{h \in \mathcal{H}_{ij}} \sum_{k \in B_{ij}} \pi_{hk} q_{ijh} \quad (36)$$

dengan pembatas-pembatas (25)-(32) dan $\pi_{hk} \geq 0$ ($\forall h \in \mathcal{H}, \forall k \in \mathcal{K}$).

Misal untuk tiap pekerjaan $i \in \mathcal{N}$, nilai minimum L_i dinyatakan dengan L_i^* . Model \mathbb{D} dapat dinyatakan kembali dengan:

Memaksimumkan

$$L = \sum_{i \in \mathcal{N}} L_i^* - \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \pi_{hk} m_{hk} \quad (37)$$

dengan pembatas-pembatas: (25)-(32) dan $\pi_{hk} \geq 0$ ($\forall h \in \mathcal{H}, \forall k \in \mathcal{K}$).

Teknik Pemecahan

Garis Besar Teknik pemecahan

Secara garis besar, langkah-langkah teknik pemecahan adalah sebagai berikut:

Langkah 0 (*Inisialisasi*)

Lakukan inisialisasi $\pi_{hk} \geq 0$ untuk tiap $h \in H$ dan $k \in K$. Tetapkan pencacah iterasi $n = 1$. Lanjutkan ke langkah 1.

Langkah 1 (*Pemecahan masalah pekerjaan individual*)

Untuk tiap $i \in \mathcal{N}$, pecahkan model \mathbb{R}_i untuk memperoleh $\{B_{ij}\}$ yang meminimumkan L_i . Misal L_i yang minimum dinyatakan dengan L_i^* dan $\{B_{ij}\}$ yang berasosiasi dengan L_i minimum dinyatakan $\{B_{ij}^*\}$. Lanjutkan ke langkah 2.

Langkah 2 (*Penentuan solusi layak*)

Berdasarkan $\{B_{ij}^*\}$, tentukan $\{B_{ij}\}$ yang layak yang dinyatakan dengan $\{\bar{B}_{ij}\}$ dengan nilai fungsi tujuannya dinyatakan dengan \bar{J} . Lanjutkan ke langkah 3.

Langkah 3 (*Penentuan solusi layak terbaik*)

Mulai dari iterasi $n' = 1$ hingga $n' = n$, tentukan solusi layak terbaik saat ini. Misal solusi layak terbaik saat ini dinyatakan dengan $\{\bar{B}_{ij}^*\}$ dengan nilai fungsi tujuan layak terbaik dinyatakan dengan \bar{J}^* . Lanjutkan ke langkah 4.

Langkah 4 (*Pemecahan masalah dual*)

Berdasarkan $\{B_{ij}^*\}$, pecahkan model \mathbb{D} untuk menghasilkan $\pi_{hk} \geq 0$ yang baru untuk tiap $h \in H$ dan $k \in K$. Lanjutkan ke langkah 5.

Langkah 5 (*Pemeriksaan kriteria henti*)

Jika kriteria henti belum terpenuhi, tetapkan $n = n + 1$ dan kembali ke langkah 1. Jika kriteria henti terpenuhi, tetapkan $\{\bar{B}_{ij}^*\}$ sebagai solusi untuk masalah penjadwalan.

Pemecahan Masalah Pekerjaan Individu

Pemecahan masalah pekerjaan individu terkait dengan pemecahan masalah penjadwalan pekerjaan individual $i \in \mathcal{N}$ (model \mathbb{R}_i) untuk memperoleh B_{ij}

($\forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i$) yang meminimumkan L_i ($\forall i \in \mathcal{N}$) dengan memperhatikan pembatas-pembatas (25)-(32). Dalam makalah ini, masalah penjadwalan pekerjaan individu dapat dipecahkan dengan metode enumerasi dan metode pemrograman dinamik.

Metode Enumerasi

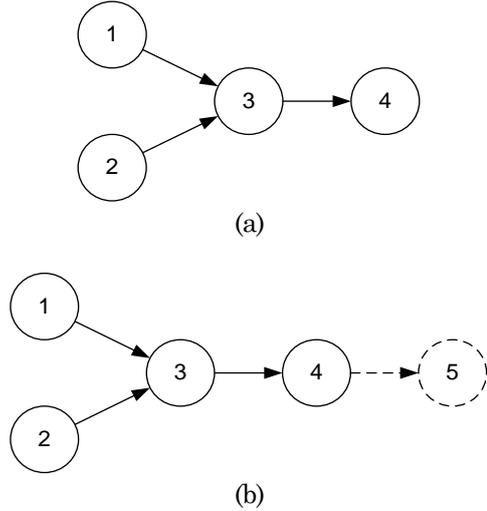
Prinsip dasar metode enumerasi adalah sebagai berikut. Untuk tiap pekerjaan $i \in \mathcal{N}$, misal himpunan operasi yang mendahului operasi $j \in \mathcal{N}_i$ dinyatakan dengan \mathcal{P}_{ij} . Misal himpunan operasi yang mengikuti operasi $j \in \mathcal{N}_i$ yang membentuk hubungan lurus (linier) dinyatakan dengan \mathcal{Q}_{ij} .

Untuk operasi $j \in \mathcal{N}_i$ dengan $\mathcal{P}_{ij} = \emptyset$ (yaitu, operasi tanpa pendahulu), tiap B_{ij} dienumerasi pada $1 \leq k \leq K - \sum_{l \in \mathcal{Q}_{ij}} t_{il} - t_{ij} + 1$ hingga k . Untuk operasi $j \in \mathcal{N}_i$ dengan $\mathcal{P}_{ij} \neq \emptyset$ dan $j \neq n_i$ (yaitu operasi dengan pendahulu dan bukan merupakan operasi terakhir), maka B_{ij} dienumerasi mulai pada $\max_{j \in \mathcal{P}_{ij}} \{B_{ij} + t_{ij} - 1\} \leq k \leq K - \sum_{l \in \mathcal{Q}_{ij}} t_{il} - t_{ij} + 1$. Untuk operasi $j \in \mathcal{N}_i$ dengan $\mathcal{P}_{ij} \neq \emptyset$ dan $j = n_i$ (yaitu operasi dengan pendahulu dan merupakan operasi terakhir), maka B_{ij} dienumerasi mulai pada $\max_{j \in \mathcal{P}_{ij}} \{B_{ij} + t_{ij} - 1\} \leq k \leq K - t_{ij} + 1$. Tetapkan tiap B_{ij} yang memberikan L_i (persamaan (24) minimum sebagai B_{ij}^* .

Metode Pemrograman Dinamis Maju

Metode pemrograman dinamis maju untuk pemecahan masalah penjadwalan pekerjaan individu memerlukan pendefinisian beberapa hal sebagai berikut.

Tahap. Misal \mathcal{P}_{ij} menyatakan himpunan pendahulu dari operasi $j \in \mathcal{N}_i$. Misal operasi $j \in \mathcal{N}_i$ didahului oleh operasi $l \in \mathcal{N}_i$ dan $l \in \mathcal{P}_{ij}$. Tiap tahap ditandai dengan notasi (i, j, i, l) . Misal himpunan operasi \mathcal{N}_i terdiri atas n_i operasi. Dalam metode pemrograman dinamis maju, satu operasi semu ($n_i + 1$) ditambahkan setelah operasi terakhir dengan waktu pengerjaan $t_{i, n_i + 1} = 0$. Dengan demikian himpunan operasi dinyatakan dengan $\mathcal{N}'_i = \mathcal{N}_i \cup \{n_i + 1\}$ dan \mathcal{P}_{ij} diperbarui menjadi \mathcal{P}'_{ij} dengan pengambahan bahwa $j = n_i$ mendahului $j = n_i + 1$. Perhitungan dalam metode pemrograman dinamis maju dimulai untuk tahap $j \in \mathcal{N}'_i$ dengan $\mathcal{P}'_{ij} \neq \emptyset$ (yaitu operasi yang memiliki operasi pendahulu) dan diakhir untuk tahap $j = n_i + 1$. Gambar 2 menunjukkan ilustrasi penambahan operasi semu.



Gambar 2. Ilustrasi penambahan operasi semu

Status. Pada tiap tahap (i, j, i, l) , status dinyatakan dengan slot waktu yang mungkin dari saat mulai operasi $j \in \mathcal{N}'_i$ yang didahului oleh operasi $l \in \mathcal{N}'_i$. Batas bawah dan atas dari B_{ij} untuk $j \in \mathcal{N}'_i$ dengan $\mathcal{P}'_{ij} \neq \emptyset$ adalah $\max_{l \in \mathcal{P}'_{ij}} \{B_{il} + t_{il} - 1\} \leq k \leq K - \sum_{q \in \mathcal{Q}_{ij}} t_{iq} - t_{ij} + 1$.

Variabel keputusan. Pada tiap tahap (i, j, i, l) , variabel keputusan dinyatakan menyatakan slot waktu dari saat mulai operasi $l \in \mathcal{N}'_i$ yang mendahului operasi $j \in \mathcal{N}'_i$, yaitu $x_{ijl} = B_{il}$.

Fungsi Transisi. Fungsi transisi dari tahap dinyatakan dengan:

$$B_{il} = B_{ij}; \forall j \in \mathcal{N}'_i, \forall l \in \mathcal{P}'_{ij}, \mathcal{P}'_{ij} \neq \emptyset \tag{38}$$

Fungsi Akumulasi. Pada tiap tahap (i, j, i, l) , untuk $j \in \mathcal{N}'_i$ dengan $\mathcal{P}'_{ij} \neq \emptyset$ dan $j \neq n_i + 1$, fungsi akumulasi yang merupakan persamaan rekursif dinyatakan dengan:

$$f_{ijl}(B_{ij}, B_{il}) = \sum_{h \in \mathcal{H}_{il}} \sum_{k \in B_{il}}^{\max\{B_{il}, k\}} \pi_{hk} q_{ilh} + \sum_{l \in \mathcal{P}_{ij}} f_{ijl}^*(S_{ijl}) \tag{39}$$

Pada tiap tahap (i, j, i, l) , untuk $j \in \mathcal{N}'_i$ dengan $\mathcal{P}'_{ij} \neq \emptyset$ dan $j = n_i + 1$, fungsi akumulasi yang merupakan persamaan rekursif dinyatakan dengan:

$$f_{ijl}(B_{ij}, B_{il}) = w_i \max\{C_{i, n_i} - d_i, 0\} + \sum_{h \in \mathcal{H}_{il}} \sum_{k \in B_{il}}^{\max\{B_{il}, k\}} \pi_{hk} q_{ijh} + \sum_{l \in \mathcal{P}_{ij}} f_{ijl}^*(S_{ijl}) \tag{40}$$

Pada tiap tahap (i, j, i, l) , nilai $f_{iju}^*(S_{iju})$ dinyatakan dengan:

$$f_{iju}^*(S_{iju}) = \begin{cases} \min_{B_{iu}} f_{iju}(B_{ij}, B_{iu}) & ; \forall j \in \mathcal{N}'_i, \mathcal{P}'_{ij} \neq \emptyset \\ 0 & ; \forall j \in \mathcal{N}'_i, \mathcal{P}'_{ij} = \emptyset \end{cases} \quad (41)$$

Langkah-langkah teknik pemrograman dinamis untuk tiap $j, l \in \mathcal{N}'_i$ adalah sebagai berikut

- Langkah 1 Mulai untuk tahap (i, j, i, l) . Jika $j \in \mathcal{N}'_i$ dengan $\mathcal{P}'_{ij} \neq \emptyset$ dan $j \neq n_i + 1$, lanjutkan ke langkah 2. Jika $j \in \mathcal{N}'_i$ dengan $\mathcal{P}'_{ij} = \emptyset$ dan $j = n_i + 1$, lanjutkan ke langkah 3.
- Langkah 2 Hitung $f_{iju}(B_{ij}, B_{iu})$ dengan menggunakan persamaan-persamaan (39) dan (41). Lanjutkan ke langkah 4.
- Langkah 3 Hitung $f_{iju}(B_{ij}, B_{iu})$ dengan menggunakan persamaan-persamaan (40) dan (41). Lanjutkan ke langkah 4.
- Langkah 4 Tetapkan $j = j + 1$. Jika $j \neq n_i + 1$, kembali ke langkah 1. Sebaliknya, lanjutkan ke langkah 5.
- Langkah 5 Lakukan penelusuran balik mulai dari $j = n_i + 1$ menggunakan hubungan transisi pada persamaan (38). Berhenti.

Penentuan Solusi Layak

Solusi yang diperoleh dengan pemecahan masalah pekerjaan individual (baik menggunakan metode enumerasi maupun pemrograman dinamis) dapat berupa solusi yang tak layak, yaitu pembatas-pembatas ketersediaan sumberdaya adalah dilanggar. Untuk menghasilkan solusi yang layak, algoritma *list scheduling* diterapkan. Prinsip dasar dari algoritma ini adalah sebagai berikut.

Misal terdapat himpunan $\{B_{ij}^*\}$ yang diperoleh dari pemecahan pekerjaan individu. Selanjutnya B_{ij}^* disusun dalam urutan menaik. Jika terdapat dua atau lebih B_{ij}^* yang sama, urutannya didasarkan atas nilai inkrementalnya. Nilai inkremental ini menunjukkan besarnya *tardiness* tertimbang jika operasi ini ditunda dalam satu slot waktu. Jika sumberdaya yang ada tersedia (memiliki kapasitas minimal sebesar kebutuhan setiap operasi) dan memperhatikan pembatas ketergantungan operasi dan ketersediaan mesin pada sisa waktu proses dari operasi tersebut maka setiap operasi dapat dijadwalkan pada setiap sumberdaya sesuai urutan. Algoritma akan menentukan operasi baru yang akan dimulai pada slot waktu tersebut dan operasi mana yang akan ditunda selama satu unit slot waktu apabila

terdapat pelanggaran ketersediaan sumberdaya pada waktu k . Apabila kapasitas sebuah sumberdaya tidak memenuhi lagi untuk waktu k maka operasi yang membutuhkan sumberdaya tersebut akan ditunda selama satu unit slot waktu. Operasi lainnya yang mengikuti operasi tersebut akan diperiksa, apakah terjadi pelanggaran terhadap pembatas ketergantungan operasi. Jika terjadi pelanggaran, maka operasi tersebut akan ditunda selama satu unit slot waktu. Proses ini akan berulang sampai semua operasi terjadwal pada sumberdaya secara layak. Langkah-langkah dari algoritma *list scheduling* adalah sebagai berikut.

Misal suatu himpunan U merupakan himpunan semua B_{ij}^* dengan B_{ij}^* diurutkan menaik. Jika terdapat satu atau lebih operasi memiliki B_{ij}^* yang sama, maka urutannya didasarkan pada nilai inkremental yang didefinisikan sebagai berikut:

$$f_{ij} = w_i((T_i + 1) - T_i) \quad (42)$$

Misal U merupakan himpunan semua (i, j) yang sudah dijadwalkan. Tetapkan S_{ij} adalah himpunan operasi yang mengikuti operasi (i, j) . Misal n adalah indeks untuk melacak ketersediaan sumberdaya. Langkah-langkah algoritma *list scheduling* adalah sebagai berikut:

- Langkah 1 Ambil $(i, j) \in U$ yang berada pada urutan pertama. Tetapkan $B = B_{ij}^*$. Lanjutkan ke langkah 2.
- Langkah 2 Tentukan l sedemikian hingga $m_{hl} \geq q_{ijh}$ untuk $h \in \mathcal{H}_{ij}$. Tetapkan $n = l$. Lanjutkan ke langkah 3.
- Langkah 3 Jika $m_{hl} \geq q_{ijh}$ untuk $n \leq l \leq n + t_{ij} - 1$ dan $h \in \mathcal{H}_{ij}$, lanjutkan ke langkah 4. Sebaliknya, lanjutkan ke langkah 5.
- Langkah 4 Jika pembatas ketergantungan operasi tidak dilanggar, tetapkan $\bar{B}_{ij} = n$, perbarui $m_{hn} = m_{hn} - q_{ijh}$ untuk $l \leq n \leq l + t_{ij} - 1$ dan $h \in \mathcal{H}_{ij}$, perbarui $U = U \setminus (i, j)$ dan lanjutkan ke langkah 7. Jika tidak lanjutkan ke langkah 5.
- Langkah 5 Tetapkan $n = n + 1$. Jika $n > B$, lanjutkan ke langkah 6. Sebaliknya, kembali ke langkah 3.
- Langkah 6 Untuk $(i, j) \in U$ dengan $B_{ij}^* < n$, tetapkan $B_{ij}^* = n$. Untuk semua $j \in S_{ij}$ yang melanggar pembatas ketergantungan operasi, tetapkan $B_{ij}^* = B_{ij}^* + 1$ dan buat daftar U yang baru. Kembali ke langkah 1.

Langkah 7 Jika $U = \emptyset$, maka berhenti. Jika tidak kembali langkah 1.

Pemecahan Masalah Lagrange Dual

Pemecahan masalah Lagrange dual pada dasarnya bertujuan untuk menghasilkan $\pi_{hk} (\forall h \in \mathcal{H}, \forall k \in \mathcal{K})$ yang baru dengan diberikan B_{ij}^* ($\forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i$). Pemecahan masalah Lagrange dual menggunakan metode subgradien. Misal $\pi_{hk}^{(n)}$ menunjukkan nilai π_{hk} pada iterasi ke- n . Tiap $\pi_{hk}^{(n)}$ diperbarui dengan persamaan berikut:

$$\pi_{hk}^{(n+1)} = \max \left\{ 0, \pi_{hk}^{(n)} + \alpha_h^{(n)} g \left(\pi_{hk}^{(n)} \right) \right\}; \forall h \in \mathcal{H}, \forall k \in \mathcal{K} \quad (43)$$

Pada iterasi n , fungsi $g(\pi_{hk}^{(n)})$ adalah subgradien L terhadap $\pi_{hk}^{(n)}$, yaitu:

$$g(\pi_{hk}^{(n)}) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} q_{ijh} X_{ijkh} - m_{hk}; \forall h \in \mathcal{H}, \forall k \in \mathcal{K} \quad (44)$$

Besaran $\alpha_h^{(n)}$ adalah ukuran langkah pada iterasi ke- n untuk tiap jenis sumberdaya $h \in \mathcal{H}$ yang diberikan dengan persamaan:

$$\alpha_h^n = \lambda \frac{\bar{L}^{(n)} - L^{(n)}}{\|g(\pi_{hk}^{(n)})\|^2}; \forall h \in \mathcal{H} \quad (45)$$

dengan $\bar{L}^{(n)}$ dan $L^{(n)}$ masing-masing menyatakan taksiran dari fungsi tujuan dual optimal dan nilai fungsi tujuan Lagrange dual pada iterasi ke- n . Nilai $L^{(n)}$ dihitung dengan menggunakan persamaan (35) atau (37). Nilai taksiran dari fungsi tujuan dual optimal ditetapkan sebagai nilai fungsi tujuan dari solusi layak terbaik sepanjang atau hingga iterasi ke- n . Misal nilai fungsi tujuan layak terbaik dinyatakan dengan \bar{J}^* . Dengan demikian, $\bar{L}^{(n)} = \bar{J}^*$.

Dalam makalah ini, dengan mengacu pada Fisher [25], nilai parameter λ ditetapkan sama dengan dua pada saat iterasi dimulai dan nilainya diturunkan setengahnya jika $L^{(n)}$ gagal untuk naik dalam beberapa iterasi (dalam hal ini digunakan kriteria tiga iterasi).

Kriteria Henti

Prosedur pemecahan berhenti jika memenuhi salah satu kriteria henti sebagai berikut: sebagai berikut:

(1) $|\bar{L}^{(n)} - L^{(n)}| < 0,1$; (2) $L^{(n-p)} - 0,01 \leq L^{(n)} \leq L^{(n-p)} + 0,01$ untuk $p = 1, 2, \dots, 20$; dan (2) $n = 1000$. Kriteria henti (1) digunakan untuk menjamin

bahwa selisih antara nilai fungsi tujuan dari solusi layak terbaik sepanjang atau hingga iterasi ke- n dan nilai fungsi tujuan dual adalah kecil. Kriteria henti (2) digunakan untuk memeriksa kekonvergenan, yaitu nilai fungsi tujuan dual naik dan turun naik pada kisaran nilai tertentu yang cukup kecil (dalam hal ini, pada kisaran $L^{(n)} \pm 0,01$ dalam 20 iterasi sebelumnya). Sementara kriteria henti (3) merupakan kriteria henti untuk jumlah iterasi.

Percobaan Numerik

Percobaan numerik dilakukan menggunakan contoh data dari Suprayogi *et al.* [9]. Tabel 1 menunjukkan karakteristik dari tiap contoh data. Teknik relaksasi Lagrange yang diusulkan dijalankan menggunakan aplikasi piranti lunak yang dibuat dengan bahasa pemrograman Visual Basic 2008 pada komputer dengan prosesor Intel Core Duo 2.00 GB RAM. Terdapat dua teknik berbasis relaksasi Lagrange yang diterapkan dalam percobaan yang masing-masing disebut dengan RL1 dan RL2. RL1 adalah teknik pemecahan berbasis relaksasi Lagrange dengan pemecahan masalah pekerjaan individu menggunakan metode enumerasi. Sementara itu RL2 menunjukkan teknik berbasis relaksasi Lagrange dengan pemecahan masalah pekerjaan individu menggunakan metode pemrograman dinamis.

Hasil pemecahan dengan teknik relaksasi Lagrange dibandingkan dengan hasil-hasil yang diperoleh menggunakan pemecahan model pemrograman linier bilangan bulat (teknik optimal) dan teknik algoritma yang keduanya diambil dari Suprayogi *et al.* [9]. Pemecahan model pemrograman linier bilangan bulat tercampur menggunakan bantuan aplikasi piranti lunak komersial LINGO 8.0. Sementara itu, teknik algoritma genetika dijalankan menggunakan aplikasi piranti lunak yang dibuat dengan bahasa pemrograman Visual Basic 6.0. Suprayogi *et al.* [9] menjalankan kedua aplikasi pada komputer dengan spesifikasi prosesor Pentium M 4, 1.7 GHz, RAM 256-share 32 MB. Hasil perbandingan ditunjukkan pada Tabel 2.

Sampai dengan dihentikan secara paksa, RL1 hanya dapat memecahkan 19 contoh data (contoh-contoh data 1-19). Sementara itu, RL2 dapat menghasilkan solusi untuk semua contoh data (contoh-contoh data 1-25). Berdasarkan contoh-contoh data 1-19, kualitas solusi (nilai fungsi tujuan yang diperoleh) baik dengan RL1 dan RL 2 adalah sama. Namun, waktu komputasi RL2 (rata-rata adalah 1,89 detik) lebih cepat dibandingkan dengan RL1 (rata-rata adalah 28228,95 detik).

Analisis statistik yang menggunakan uji t untuk rerata dua sampel berpasangan dilakukan untuk

menguji kualitas solusi antara RL2 dengan teknik optimal dan algoritma genetika. Ringkasan hasil pengujian terhadap rerata nilai fungsi tujuan ditunjukkan pada Tabel 3. Dengan uji dua-arah, pada tingkat signifikansi 5%, rerata nilai fungsi tujuan RL2 memberikan hasil yang tidak berbeda secara signifikan baik dengan teknik optimal maupun dengan algoritma genetika. Walaupun analisis perbandingan terhadap waktu komputasi tidak dilakukan karena spesifikasi komputer yang digunakan untuk menjalankan aplikasi-aplikasi piranti lunak adalah berbeda, namun percobaan numerik menunjukkan bahwa RL2 dan algoritma genetika mampu menangani masalah penjadwalan dengan ukuran yang besar.

Tabel 1. Contoh data yang digunakan dalam percobaan

Contoh	Jumlah pekerjaan	Jumlah operasi untuk tiap pekerjaan	Jumlah unit untuk tiap sumber-daya	Panjang horison perencanaan	Struktur pekerjaan
1	3	4, 4, 4	3	25	Pohon
2	4	3, 3, 3, 3	2	25	Linier
3	5	3, 3, 3, 3	2	17	Linier
4	2	4, 4	2	20	Pohon
5	3	3, 3, 3	3	25	Pohon
6	2	5, 4	2	17	Pohon
7	3	2, 3, 4	2	20	Pohon
8	4	2, 3, 3, 4	2	30	Pohon
9	3	4, 4, 4	2	30	Pohon
10	2	5, 5	2	25	Pohon
11	3	4, 4, 4	3	30	Pohon
12	2	5, 5	2	18	Pohon
13	3	4, 4, 4	3	30	Pohon
14	3	4, 4, 4	3	20	Pohon
15	3	3, 3, 3	2	25	Pohon
16	6	5 (untuk tiap pekerjaan)	5	80	Linier
17	6	5 (untuk tiap pekerjaan)	5	70	Pohon
18	8	5 (untuk tiap pekerjaan)	5	80	Linier
19	8	5 (untuk tiap pekerjaan)	5	80	Pohon
20	10	5 (untuk tiap pekerjaan)	5	90	Linier
21	10	5 (untuk tiap pekerjaan)	5	90	Pohon
22	15	5 (untuk tiap pekerjaan)	5	90	Linier
23	15	5 (untuk tiap pekerjaan)	5	110	Pohon
24	20	5 (untuk tiap pekerjaan)	5	150	Linier
25	20	5 (untuk tiap pekerjaan)	5	120	Pohon

Table 2. Perbandingan hasil percobaan numerik

Contoh data	Teknik optimal (Pemecahan MPLBBC)		Algoritma genetika (Rata-rata dari 20 replikasi)		RL1		RL2	
	NFT	WK	NFT	WK	NFT	WK	NFT	WK
1	68	76	69,1	1,3	72	106	72	0,5
2	98	10	98,0	1,2	99	30	99	0,5
3	48	80	49,6	1,1	48	38	48	0,7
4	62	1	62,0	1,0	62	36	62	0,7
5	52	3	53,4	1,1	52	720	52	0,9
6	6	4	8,1	1,0	10	590	10	0,8
7	61	2	62,8	1,0	61	260	61	1,0
8	68	2	68,0	1,0	68	49	68	1,0
9	74	19	75,5	1,1	77	194	77	1,2
10	43	2	43,0	1,0	43	168	43	1,1
11	83	15	89,2	1,5	109	196	109	1,1
12	12	3	12,0	1,0	12	144	12	1,0
13	79	31	80,1	1,2	106	226	106	1,6
14	20	18	20,0	1,0	20	114	20	0,5
15	69	126	69,0	1,0	69	122	69	1,0
16	-	-	448,4	8,1	527	1031	527	1,3
17	-	-	316,4	8,9	323	73486	323	5,0
18	-	-	731,2	20,5	717	37788	717	7,0
19	-	-	701,3	22,5	674	80954	674	9,0
20	-	-	795,7	33,5	-	-	842	15,0
21	-	-	927,7	42,6	-	-	1073	12,0
22	-	-	2495,4	112,9	-	-	2593	29,0
23	-	-	2970,8	137,1	-	-	2793	37,0
24	-	-	6796,6	358,0	-	-	7132	69,0
25	-	-	4766,2	308,0	-	-	4312	75,0

NFT: Nilai fungsi tujuan (total *tardiness* tertimbang), WK: Waktu komputasi (dalam detik)

Tabel 3. Ringkasan hasil pengujian uji *t* dua-sampel berpasangan (uji dua-arah) untuk rerata nilai fungsi tujuan

	Teknik optimal dan RL2 (Contoh-contoh data 1-15)		Algoritma genetika dan RL 2 (Contoh-contoh data 1-25)	
	Teknik optimal	RL2	Algoritma genetika	RL2
Rata-rata	56,20	60,53	872,38	875,76
Simpangan baku	26,52	31,07	1680,99	1684,35
Nilai <i>t</i> yang dihitung	-1,8395		-0,1324	
Derajat kebebasan	14		24	
Nilai <i>P</i>	0,0871		0,8958	

Simpulan

Makalah ini telah mengusulkan teknik relaksasi Lagrange untuk memecahkan masalah penjadwalan pekerjaan majemuk dengan sumberdaya simultan. Masalah penjadwalan melibatkan lebih dari satu pekerjaan dengan tiap pekerjaan terdiri atas satu himpunan operasi. Tiap operasi memerlukan pengerjaan secara simultan oleh lebih dari satu jenis sumberdaya dengan jumlah unit dari tiap sumberdaya yang digunakan untuk pengerjaan suatu

operasi adalah satu atau lebih. Masalah terkait dengan penentuan jadwal dari tiap operasi dari tiap pekerjaan yang meminimumkan total *tardiness* tertimbang.

Terdapat dua teknik relaksasi Lagrange yang diusulkan sebagai teknik pemecahan yang masing-masing disebut dengan RL1 dan RL2. RL1 adalah teknik relaksasi Lagrange dengan pemecahan masalah pekerjaan individu menggunakan metode enumerasi. Sementara itu RL2 menunjukkan teknik relaksasi Lagrange dengan pemecahan masalah pekerjaan individu menggunakan metode pemrograman dinamis.

Berdasarkan percobaan numerik, RL1 dan RL2 keduanya menghasilkan solusi dengan nilai fungsi tujuan yang sama. Namun, RL2 adalah lebih efisien dibandingkan dengan RL1. Hasil percobaan numerik menunjukkan bahwa RL2 memberikan kualitas solusi yang tidak berbeda dibandingkan dengan teknik optimal dan algoritma genetika. Namun demikian, baik RL1 dan algoritma genetika, keduanya memiliki kemampuan untuk menangani masalah dengan ukuran yang lebih besar. Keunggulan RL2 dibandingkan dengan algoritma genetika adalah bahwa RL2 cukup dijalankan satu kali dibandingkan dengan algoritma genetika. Karena sifat probabilistiknya, algoritma genetika perlu dijalankan dalam sejumlah replikasi untuk mencari solusi yang terbaik.

Daftar Pustaka

1. Dobson, G., and Karmarkar, U.S., Simultaneous Resource Scheduling to Minimize Weighted Flow Times, *Operations Research*, 37, 1989, pp. 592-600.
2. Dobson, G., and Khosla, I., Simultaneous Resource Scheduling with Batching to Minimize Weighted Flow Times, *IEE Transactions*, 27, 1995, pp. 587-598.
3. Suprayogi dan Mardiono, N.M.T, Pengembangan Algoritma bagi Masalah Penjadwalan Banyak Job Operasi Tunggal Banyak Sumber Paralel Simultan, *Jurnal Teknik dan Manajemen Industri*, 17, 1997, pp. 39-49.
4. Luh, P.B., Liu, F., and Moser, B., Scheduling of Design Projects with Uncertain Number of Iterations, *European Journal of Operational Research*, 113, 1999, pp. 575-592.
5. Toha, I.S., dan Halim, A.H., Algoritma penjadwalan produksi berbasis jaringan untuk sumberdaya tunggal dan simultan, *Jurnal Teknik dan Manajemen Industri*, 19 (2), 1999, pp. 10-20.
6. Suprayogi dan Toha, I.S., Model Pemrograman Linier Bilangan Bulat untuk Masalah Penjadwalan Sumber-Majemuk Paralel Simultan, *Jurnal Teknik dan Manajemen Industri*, 22 (1), 2002, pp. 27-38.
7. Suprayogi dan Partono, S., Pendekatan Pemecahan Berbasis Relaksasi Lagrange untuk Masalah Penjadwalan Job-Majemuk, Operasi-Majemuk Sumber-Majemuk Paralel Simultan, *Prosiding Seminar Sistem Produksi VII 2005*, Surabaya, 2005, pp. 907-920
8. Suprayogi and Partono, S., Lagrangian Relaxation Technique for Solving Simultaneous Multiresource Scheduling Problems, *Proceedings of Asia Pacific Conference on Manufacturing Systems (APCOMS) 2007, Bali*, 2007.
9. Suprayogi, Siregar, A., and Aji, T., Solving Multiproject Constrained Scheduling Problems with Simultaneous Multiresource using Genetic Algorithm, *Proceedings of Asia Pacific Conference on Manufacturing Systems (APCOMS) 2007, Bali*, 2007.
10. Wu, J.-Z., and Chien, C.-F., Modeling semiconductor testing job scheduling and dynamic testing machine configuration, *Expert Systems with Applications*, 35, 2008, pp. 485–496.
11. Wu, J.-Z., Hao, X.-C., Chien, C.-F., Gen, M., A Novel Bi-Vector Encoding Genetic Algorithm for the Simultaneous Multiple Resources Scheduling Problem, *Journal of Intelligent Manufacturing*, 23, 2012, pp. 2255-2270.
12. Kaskavelis, C.A., and Caramanis. M.C., A Lagrangian Relaxation Based Algorithm for Scheduling Multiple-Part Production-System: Industrial Implementation Experience, *Proceedings of the 1994 Japan-U.S.A. Symposium on Flexible Automation*, Kobe, Japan, July 11-18, 1994, pp.173-180.
13. Kaskavelis, C.A., and Caramanis. M.C., Application of a Lagrangian Relaxation Scheduling Based Algorithm to A Semiconductor Testing Facility, *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, Troy, NY, October 10-12., 1994, pp. 106-112.
14. Kaskavelis. C.A., and Caramanis, M.C., Efficient Lagrangian Relaxation Algorithms for Industry Size Job-shop Scheduling Problems, *IEE Transactions*, 30, 1998, pp. 1085-1097.
15. Luh, P.B., Hoitomt, D.J., Max, E., and Pattipati, K.R., Schedule Generation and Reconfiguration for Parallel Machines, *IEEE Transactions on Robotics and Automation*, 6 (6), 1990, pp. 687-696.
16. Hoitomt, D.J., Luh, P.B., and Pattipati, K.R., A Practical Approach to Job Shop Scheduling Problems, *IEEE Transactions on Robotics dan Automation*, 9 (1), 1993, pp. 1-13.
17. Luh, P.B., and Hoitomt, D.J., Scheduling of Manufacturing Systems using the Lagrangian Relaxation Technique, *IEEE Transaction on Automatic Control*, 38 (7), 1993, pp. 1066-1079.

18. Luh, P.B., Wang, J.H., Wang, J.L., Tomastik, R.N., and Howes, T.D., Near Optimal Scheduling of Manufacturing Systems with Presence of Batch Machines and Setup Requirements. *CIRP Annals – Manufacturing Technology*, 46 (1), 1997, pp. 397–402.
19. Chen, T.R. and Hsia, T.C., Scheduling for IC Sort and Test Facilities with Precedence Constraints via Lagrangian Relaxation , *Journal of Manufacturing Systems*, 16 (2), 1997, 117-128.
20. Luh, P.B., Zhou, X., and Tomastik, R.N., An Effective Method to Reduce Inventory in Job Shops, *IEEE Transactions on Robotics and Automation*, 16 (4), 2000, pp. 420-424.
21. Zhang, Y., Luh, P.B., Yoneda, K., Kano, T., and Kyoya, Y., Mixed-Model Assembly Line Scheduling using the Lagrangian Relaxation Technique, *Proceedings of the 1997 IEEE International Conference on Control Applications*, Hartford, CT, U.S.A., October 5-7, 1997, pp. 429-434.
22. Zhang, Y., Luh, P.B., Narimatsu, K., Moriya, T., Shimada, T., and Fang, L., A Macro-Level Scheduling Method Using Lagrangian Relaxation, *IEEE Transactions on Robotics and Automation*, 17 (1), 2001, pp. 70-79.
23. Morton, T.E., and Pentico, D.W., *Heuristic Scheduling System: with Applications to Production Systems and Project Management*, John Wiley and Sons, New York, U.S.A., 1993.
24. Pritsker, A.A.B., Watters, L.J., and Wolfe, P.M., Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach, *Management Science* 16, 1969, pp. 93-108.
25. Fisher, M.L., An Application Oriented Guide to Lagrangian Relaxation, *Interfaces* 15, 1985, pp. 10-21.