

Alternatif Pemecahan Masalah *Open Shop Scheduling* dengan Pendekatan Algoritma Genetik dan Heuristik

Mochammad Zuliansyah¹⁾, Giva Andriana Mutiara²⁾

¹⁾Lab. NetPrep/OS, Program Studi Teknik Informatika, Fakultas Teknik,
Universitas ARS Internasional Bandung
e-mail: mzazka@yahoo.com

²⁾Universitas Langlangbuana, Bandung
e-mail: givamz@yahoo.com

Abstract

One of industrial problem is Open Shop Scheduling (OSS). This research try to propose open shop scheduling problem alternative solution with genetic and heuristic approach. First we try to make a genetic coding for this problem. Machine scheduling system analogue as individu, chromosom analogue as a machine (task), and gen analogue as the operation. Each individu consist of chromosoms, and each chromosom consist of gens. We used two methods which represented by coding by operation and code task-operation. After chromosom coding detected, fitness function can determine the function of giving chromosoms weight to choose parent chromosom and changing population. Fitness function design base on minimum sum of zero time on each machine with high efficiency, and lack of overlapping job for each machine.

Keywords: *open-shop scheduling, genetic algorithm, heuristic, fitness function*

1. Pendahuluan

Masalah penjadwalan mesin untuk jenis tugas-tugas dengan operasi-operasi yang tidak bersifat sekuensial (*Open-Shop Scheduling Problem*) merupakan masalah yang kompleks. OSSP muncul pada batasan bahwa sekumpulan operasi dapat dilayani oleh satu mesin. Efisiensi produksi dan proses manufaktur membutuhkan metode yang efektif untuk mengoptimasi berbagai aspek yang mempengaruhi proses penjadwalan mesin yang umumnya terfokus pada total waktu yang dibutuhkan untuk memproses semua operasi. Penelitian ini berupaya memberikan kontribusi positif pada OSSP dengan menggunakan algoritma Genetik dan Heuristik dengan tetap memberikan fleksibilitas dan kemudahan penggunaan dan memungkinkan munculnya pengembangan selanjutnya.

Penelitian ini menggunakan dua teknik representasi kromosom. Pertama dengan merepresentasikan operasi pada tiap tugas per mesin, dan yang kedua dengan menambah lebar kromosom. Penambahan lebar kromosom digunakan untuk menyimpan informasi induk dari suatu operasi yaitu tugas.

2. Batasan Masalah

Kemungkinan penyederhanaan OSSP adalah dengan memberi batasan bahwa untuk suatu operasi dapat diproses oleh mesin tertentu. Namun pada kenyataannya, sebuah operasi dapat diproses dengan berbagai kemungkinan, termasuk diproses oleh beberapa mesin. Faktor lain yang perlu diperhatikan adalah waktu batas akhir (*due date*) dan waktu setup mesin.

OSSP dibangun dari sejumlah m mesin dan t tugas, dengan tiap tugas merupakan kumpulan dari sejumlah o operasi. Operasi dinyatakan sebagai (a,b) dengan a merupakan identifikasi mesin yang memproses operasi tersebut dan b merupakan waktu yang dibutuhkan mesin untuk memproses operasi. Penjadwalan OSSP akan melakukan penentuan waktu awal dari tiap operasi untuk memenuhi batasan bahwa mesin hanya dapat memproses tepat satu operasi pada suatu saat.

Penelitian ini dilakukan berdasarkan batasan:

- Penelitian menggunakan kombinasi mesin-tugas 4X4, 5X5, 7x7, 10x10, 15x15, dan 20x20
- Representasi kromosom menggunakan 4 teknik, yaitu dengan pengkodean dasar, pengkodean langsung ganda, pengkodean heuristik, dan pengkodean heuristik dinamis
- Tiap tugas terdiri dari sejumlah operasi yang dapat dieksekusi secara acak / tidak terurut.
- Jumlah populasi 1000 individu.

3. Tujuan Penelitian

Tujuan penelitian ini adalah menemukan alternatif pemecahan OSSP dengan mengacu pada minimasi waktu kosong untuk semua mesin dalam menyelesaikan serangkaian tugas berdasarkan pendekatan algoritma genetik dan heuristik.

4. Dasar Teori

Algoritma genetik merupakan algoritma pencarian yang berdasar pada mekanisme seleksi dan proses genetika alami. Algoritma ini menggabungkan antara struktur string yang bertahan, struktur string yang belum diacak untuk menentukan cakupan pencarian dan dengan teknik pencarian cerdas yang dapat dilakukan oleh manusia. Pada setiap generasi, sebuah rangkaian populasi string yang merupakan gabungan antara string lama dan string baru hasil rekayasa genetik yang menggantikan posisi string *parent*. Pada saat pengacakan cakupan pencarian, algoritma genetik menggunakan data historis dan data empiris untuk menentukan arah pencarian yang lebih terarah pada string tujuan.

Pikiran dasar dalam penelitian seputar algoritma genetik adalah *robustness*, keseimbangan antara tingkat efisiensi dan efektifitas yang dibutuhkan untuk bertahan dalam berbagai cakupan kasus. Jika tingkat *robustness* dari sistem cerdas ditingkatkan, maka kemungkinan perekayasa ulang dapat diperkecil. Jika tingkat adaptasi dengan lingkungan kasus yang berbeda dapat ditingkatkan, maka sistem yang telah ada dapat melakukan fungsinya lebih lama dan lebih baik. Kemampuan untuk memperbaiki diri sendiri, menentukan arah pencarian sendiri dan reproduksi merupakan kemampuan yang dimiliki oleh sistem biologis, namun kemampuan tersebut sebaiknya juga dapat dilakukan oleh sistem cerdas.

Oleh karena itu arah pengkajian *robustness* mengacu pada sistem alami. Kemampuan beradaptasi dan bertahan hidup dari sistem biologi alami dapat digunakan sebagai dasar kajian dan acuan dalam membentuk sistem cerdas.

Kriteria penentuan optimisasi suatu metode pencarian dapat dipandang dari dua titik acuan yaitu melakukan pengkajian secara kreatif terhadap proses yang dilakukan (proses), dan optimalisasi pencapaian titik tujuan (destinasi). Pada kehidupan sehari-hari, kriteria optimasi pencarian dapat disederhanakan menjadi penentuan prioritas proses dan mengembangkan pengkajian terhadap proses untuk setiap prioritas secara kreatif.

Jika dibandingkan dengan metode pencarian berbasis kalkulus, enumeratif, dan acak, maka algoritma genetik mempunyai karakteristik sebagai berikut:

1. Algoritma genetik bekerja berdasar pada metode pengkodean untuk sekumpulan parameter yang berkaitan

2. Algoritma genetik melakukan pencarian dari sejumlah titik yang tergabung dalam satuan populasi, bukan pada titik pencarian tunggal
3. Algoritma genetik menggunakan fungsi obyektif (fungsi fitness) untuk menentukan membedakan antara satu item informasi dengan item yang lain
4. Algoritma genetik menggunakan aturan transisi dengan probabilitas, bukan dengan aturan deterministik.

Algoritma genetik membutuhkan sejumlah parameter alami pada masalah optimasi pencarian dan dikodekan dalam sejumlah terbatas string yang dibangun dari sejumlah terbatas karakter.

5. Representasi Kromosom

Elemen utama dari algoritma genetik adalah penentuan teknik pengkodean kromosom terhadap masalah yang akan dipecahkan. Mekanisme yang tepat akan berpengaruh pada penentuan fungsi fitness yang digunakan sebagai acuan pendekatan hasil. Setiap individu merupakan representasi sebuah bentuk penjadwalan mesin, sehingga individu dibentuk dari serangkaian terurut kromosom.

5.1 Pengkodean Dasar

Kromosom pada OSSP dibangun dari serangkaian g gen, dengan g merupakan kumulatif dari operasi dari setiap tugas. Setiap gen berkisar dari $\{1, 2, \dots, t\}$ dengan t merupakan tugas yang membutuhkan proses paling besar. Kromosom membangun penjadwalan mesin berdasarkan aturan sebagai berikut:

Rangkaian gen $abc\dots$ berarti:

- Pilih operasi yang belum dieksekusi dari tugas ke- a yang belum terselesaikan, dan tempatkan pada lokasi pertama yang kosong dan mempunyai renggang waktu yang cukup pada pembangunan jadwal.
- Pilih operasi yang belum dieksekusi dari tugas ke- b yang belum terselesaikan, dan tempatkan pada lokasi pertama yang kosong dan mempunyai renggang waktu yang cukup pada pembangunan jadwal.
- Pilih operasi yang belum dieksekusi dari tugas ke- c yang belum terselesaikan, dan tempatkan pada lokasi pertama yang kosong dan mempunyai renggang waktu yang cukup pada pembangunan jadwal.

Aturan serupa untuk gen berikutnya.

Pembangunan jadwal pada OSSP terjadi dengan melakukan kontrol pada *circular list* dari daftar tugas yang belum terselesaikan dan daftar operasi dari tugas tersebut. Sehingga notasi *tugas ke- a yang belum terselesaikan* ditentukan dari nilai modulo panjang *circular list*, untuk menentukan lokasi tugas tersebut secara tepat.

Proses pemilihan lokasi merupakan proses tersendiri yang membutuhkan metode representasi. Setiap Kromosom merepresentasikan ruang wilayah dari kemungkinan solusi. Sebagai contoh ruang wilayah, jika diketahui sebuah kromosom dinyatakan sebagai '1,2,1, ...' berarti operasi pertama yang terjadwalkan merupakan bagian dari tugas pertama, operasi kedua dari tugas ke-2 dan operasi ketiga dari tugas ke-3.

5.2 Pengkodean Langsung Ganda

Metode ini merepresentasi kromosom dengan panjang gen dua kali lipat dari metode pengkodean dasar. Informasi tambahan berupa keterangan tugas dari operasi tertentu. Rangkaian gen $abcd$ berarti pilih operasi ke- a yang belum terselesaikan dari operasi ke- b yang belum terselesaikan dan tempatkan pada lokasi pertama yang kosong dan mempunyai

renggang waktu yang cukup pada pembangunan jadwal, kemudian pilih operasi ke-c yang belum terselesaikan dari operasi ke-d yang belum terselesaikan dan tempatkan pada lokasi pertama yang kosong dan mempunyai renggang waktu yang cukup pada pembangunan jadwal.

5.3 Pengkodean dengan Heuristik

Metode pengkodean ini menggunakan teknik heuristik untuk menentukan operasi berikutnya yang akan dieksekusi. Rangkaian gen abcd berarti gunakan heuristik untuk menentukan operasi yang belum terselesaikan dari tugas ke-a yang belum terselesaikan dan tempatkan pada lokasi pertama yang kosong dan mempunyai renggang waktu yang cukup pada pembangunan jadwal, heuristik untuk menentukan operasi yang belum terselesaikan dari tugas ke-b yang belum terselesaikan dan tempatkan pada lokasi pertama yang kosong dan mempunyai renggang waktu yang cukup pada pembangunan jadwal, dan selanjutnya.

Teknik heuristik pada penelitian ini dilakukan dengan 8 cara, yaitu:

- BSR Pilih operasi dengan waktu pemrosesan terbesar, dan tentukan sebagai operasi awal dari operasi tersebut
- KCL Pilih operasi dengan waktu pemrosesan terkecil, dan tentukan sebagai operasi awal dari operasi tersebut
- EF-BSR Jika t merupakan waktu terawal suatu operasi dapat terjadwalkan, dan S merupakan kumpulan operasi yang dapat dijadwalkan pada t , maka lakukan BSR pada S
- EF-KCL Jika t merupakan waktu terawal suatu operasi dapat terjadwalkan, dan S merupakan kumpulan operasi yang dapat dijadwalkan pada t , maka lakukan KCL pada S
- EF-RAN Jika t merupakan waktu terawal suatu operasi dapat terjadwalkan, dan S merupakan kumpulan operasi yang dapat dijadwalkan pada t , maka pilih operasi pada S secara acak
- SG-BSR Jika G merupakan kumpulan operasi yang dapat dijadwalkan pada celah waktu penjadwalan, yang berarti operasi-operasi tersebut dapat dijadwalkan diantara dua operasi yang telah dijadwalkan sebelumnya pada mesin yang sama. Lakukan BSR pada G . Jika G kosong maka teknik heristik yang dilakukan sama dengan BSR.
- PJG Jika G merupakan kumpulan operasi yang dapat dijadwalkan pada celah waktu penjadwalan, yang berarti operasi-operasi tersebut dapat dijadwalkan diantara dua operasi yang telah dijadwalkan sebelumnya pada mesin yang sama. Pilih operasi dari G yang menyisakan waktu terbanyak pada celah waktu penjadwalan. Jika G kosong maka teknik heristik yang dilakukan sama dengan BSR.
- PDK Jika G merupakan kumpulan operasi yang dapat dijadwalkan pada celah waktu penjadwalan, yang berarti operasi-operasi tersebut dapat dijadwalkan diantara dua operasi yang telah dijadwalkan sebelumnya pada mesin yang sama. Pilih operasi dari G yang menyisakan waktu paling kecil pada celah waktu penjadwalan. Jika G kosong maka teknik heristik yang dilakukan sama dengan BSR.

5.4 Pengkodean dengan Heuristik Dinamis

Pengkodean dengan teknik heuristik dinamis merupakan pengembangan teknik heuristik. Rangkaian gen abcd berarti gunakan teknik heuristik ke-a untuk memilih operasi pada operasi ke-b yang belum terselesaikan dan dan tempatkan pada lokasi pertama yang kosong dan mempunyai renggang waktu yang cukup pada pembangunan jadwal, gunakan teknik heuristik ke-c untuk memilih operasi pada operasi ke-c yang belum terselesaikan dan dan tempatkan pada lokasi pertama yang kosong dan mempunyai renggang waktu yang cukup pada pembangunan jadwal, dan selanjutnya.

5.5 Penentuan Fungsi Fitness

Fungsi fitness didefinisikan sebagai fungsi kumulasi waktu celah pada penjadwalan. Fungsi akan mengembalikan nilai berupa jumlah kumulatif seluruh waktu celah pada keseluruhan jadwal mesin. Fungsi yang mengembalikan nilai lebih kecil, mengidentifikasi bentuk jadwal yang lebih baik daripada rangkaian jadwal yang menghasilkan waktu celah yang besar. Fungsi ini kemudian disebut dengan fungsi *celah-minimum*.

6. Analisis

Penelitian ini melakukan pengujian untuk setiap teknik representasi kromosom berdasarkan peta jumlah tugas dengan acuan jumlah mesin yang tersedia. Kombinasi jumlah tugas-mesin yang diuji adalah 4x4, 5x5, 7x7, 10x10, 15x15, dan 20x20. Pada setiap kasus, uji coba dilakukan berdasarkan teknik algoritma genetik dengan fungsi fitness menggunakan fungsi celah minimum yang telah dibahas pada sub bab 3.2. Pembentukan populasi awal dilakukan dengan teknik pemilihan acak untuk serangkaian kromosom berdasarkan distribusi uniform. Populasi yang dibentuk adalah sebanyak 1000 individu yang dibangun berdasarkan metode acak berdistribusi uniform. Pindah silang dilakukan berdasarkan nilai probabilitas pindah silang antara 30% hingga 80% dengan angka penambahan 0,05 %, sehingga muncul 1000 perbedaan angka probabilitas. Setelah proses pindah silang, maka dihasilkan dua kromosom anak yang kemudian dapat mengalami proses mutasi dengan probabilitas 50%.

6.1 Tugas-Mesin Kecil

Pada setiap kasus, hasil pengujian ditentukan berdasarkan hasil rata-rata dari setiap fungsi fitness terbaik dengan pengulangan sebanyak 10 kali percobaan. Tabel 1. menggambarkan hasil percobaan untuk ukuran matrik tugas-mesin kecil 4x4, 5x5, dan 7x7.

Tabel 1. Hasil pengujian untuk ukuran matrik tugas-mesin kecil

		Rata-rata			Nilai Terbaik		
		Tugas x Mesin			Tugas x Mesin		
Representasi Kromosom		4x4	5x5	7x7	4x4	5x5	7x7
Pengkodean Dasar		193	300	438	193	300	438
Pengkodean Langsung Ganda		194,4	308,5	454,1	193	302	441
Pengkodean dengan Heuristik	BSR	223	322,3	451,5	222	321	450
	KCL	199	316,7	450,3	199	313	449
	EF-BSR	211	312,2	449,8	211	305	443
	EF-KCL	193,4	303,9	449,7	193	301	445
	EF-RAN	195,6	305,6	448,9	195	301	436
	SG-BSR	196,2	304,2	449,8	194	301	444
	PJG	197,6	304,3	449,9	195	302	441
	PDK	197,8	304,7	449,9	194	301	443
Pengkodean dengan Heuristik Dinamis	BSR	197,3	312,6	451,2	197	311	449
	KCL	194,5	309,6	450,9	194	308	447
	EF-BSR	194,3	307,6	444,9	193	305	435
	EF-KCL	193	305	449,7	193	300	441
	EF-RAN	194,1	305,3	449,8	193	300	435
	SG-BSR	193,9	309,5	451,2	193	307	447
	PJG	195,8	308,9	452,1	195	304	449
	PDK	195,3	309,8	452,8	194	305	448

{waktu celah dalam satuan proses}

Berdasarkan tabel 1 dapat diambil kesimpulan bahwa metode pengkodean kromosom berdasarkan heuristik dinamis memberikan hasil yang terbaik. Pengujian heuristik dinamis EF-RAN memberikan hasil waktu celah terbaik pada matrik tugas-mesin 7x7, diikuti dengan EF-RAN pada heuristik statis. Secara umum pengkodean kromosom KCL memberikan hasil yang optimal pada matrik tugas-mesin berukuran kecil (4x4 dan 5x5) dan BSR memberikan hasil optimal pada matrik tugas-mesin berukuran sedang (7x7).

6.2 Tugas-Mesin Besar

Tabel 2 menggambarkan hasil percobaan untuk ukuran matrik tugas-mesin besar 10x10, 15x15, dan 20x20.

Tabel 2. Hasil pengujian untuk ukuran matrik tugas-mesin besar

		Rata-rata			Nilai Terbaik		
		Tugas x Mesin			Tugas x Mesin		
Representasi Kromosom		10x10	15x15	20x20	10x10	15x15	20x20
Pengkodean Dasar		645	937	1155	645	937	1155
Pengkodean Langsung Ganda		690,7	968,9	1244,5	668	951	1224
Pengkodean dengan Heuristik	BSR	745,2591	1077,117	1508,899	741,9171	1072,772	1503,886
	KCL	665,0518	1058,402	1504,889	665,0518	1046,036	1500,544
	EF-BSR	705,1555	1043,363	1503,218	705,1555	1019,301	1480,492
	EF-KCL	646,3368	1015,624	1502,883	645	1005,933	1487,176
	EF-RAN	653,6891	1021,306	1500,21	651,684	1005,933	1457,098
	SG-BSR	655,6943	1016,627	1503,218	648,342	1005,933	1483,834
	PJG	660,3731	1016,961	1503,552	651,684	1009,275	1473,808
	PDK	661,0415	1018,298	1503,552	648,342	1005,933	1480,492
Pengkodean dengan Heuristik Dinamis	BSR	659,3705	1044,7	1507,896	658,3679	1039,352	1500,544
	KCL	650,013	1034,674	1506,894	648,342	1029,326	1493,86
	EF-BSR	649,3446	1027,99	1486,842	645	1019,301	1453,757
	EF-KCL	645	1019,301	1502,883	645	1002,591	1473,808
	EF-RAN	648,6762	1020,303	1503,218	645	1002,591	1453,757
	SG-BSR	648,0078	1034,339	1507,896	645	1025,984	1493,86
	PJG	654,3575	1032,334	1510,904	651,684	1015,959	1500,544
	PDK	652,6865	1035,342	1513,244	648,342	1019,301	1497,202

{ waktu celah dalam satuan proses }

Pada kasus matrik tugas besar berukuran besar, hasil optimal diperoleh pada metode pengkodean kromosom dengan menggunakan teknik heuristik dinamis dan diikuti dengan teknik heuristik statis.

Berdasarkan percobaan-percobaan diatas, dapat diambil kesimpulan bahwa teknik hibrid yang merupakan gabungan antara algoritma genetik dan heuristik memberikan hasil waktu-celah yang lebih baik daripada hanya dengan menggunakan algoritma genetik.

7. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah:

- a. Waktu celah antar proses pada penjadwalan mesin terkecil didapat pada bentuk representasi kromosom dengan teknik heuristik dinamis pada keseluruhan kasus percobaan, dan diikuti dengan pengkodean kromosom dengan teknik heuristik statis.
- b. Pada bentuk kombinasi tugas-mesin yang relatif kecil seperti 4 x 4, dan 5 x 5, kinerja algoritma genetik menghasilkan waktu celah kumulatif penjadwalan yang relatif sama antara berbagai teknik representasi kromosom. Sehingga kombinasi bentuk representasi mempunyai kontribusi yang kecil terhadap hasil waktu celah kumulatif penjadwalan pada bentuk kombinasi tugas-mesin yang relatif kecil.
- c. Pada bentuk kombinasi tugas-mesin yang relatif besar seperti 7 x 7, 10 x 10, 15 x 15, dan 20 x 20, kinerja algoritma genetik dengan representasi kromosom berdasarkan teknik heuristik menghasilkan waktu celah kumulatif penjadwalan yang lebih baik daripada representasi kromosom tanpa heuristik. Sehingga kombinasi bentuk representasi khususnya dengan penambahan unsur heuristik, mempunyai kontribusi positif terhadap hasil waktu celah kumulatif penjadwalan pada bentuk kombinasi tugas-mesin yang relatif besar.

8. Saran

Penelitian ini membutuhkan pengembangan lebih lanjut untuk dapat diimplementasikan secara nyata. Pengembangan riset yang dapat dilakukan antara lain:

- a. Penambahan proses pemilihan jalur sekuensial antar operasi pada setiap tugas secara dinamis. Pendekatan ini memberikan dampak kedekatan dengan realitas yang lebih baik daripada bentuk rangkaian operasi per tugas pada OSSP.
- b. Pendekatan teknik algoritma genetik hybrid dan teknik riset operasi pada penyelesaian OSSP.

Daftar Pustaka

- Bagchi, Sugato., Uckun, Serdar., Miyabi, Yutaka., dan Kawamura, Kazuhiko., 1991, 'Exploring Problem-Specific Recombination Operators for Job-Shop Scheduling', dalam 'Proceedings of the Fourth International Conference on Genetics Algorithms, editor Belew, R.K., dan Booker, L.B., hal 10-17, San Mateo
- Reeves, Colin, 1994, 'Hybrid Genetic Algorithms for Bin-Packing and Related Problems', Technical Report, School of Mathematical and Information Sciences, Coventry University
- Fang, H., Ross, P., Corne, D., 1993, 'A Promising Genetic Algorithm Approach to Job-Shop Problems, Rescheduling, and Open-Shop Scheduling Problems', dalam 'Proceedings of the Fifth International Conference on Genetics Algorithms, editor Belew, R.K., dan Booker, L.B., hal 257-382, San Mateo
- Rich, E. dan Knight, K., *Artificial Intelligent*, McGraw Hill, 1991
- Taillard, E., 1993, 'Benchmarks for Basic Scheduling Problems', *European Journal of Operations Research*, edisi 64, hal 274-285.