ORIGINAL ARTICLE

OPTIMASI *PART TYPE SELECTION AND MACHINE LOADING PROBLEMS* PADA FMS MENGGUNAKAN METODE *PARTICLE SWARM OPTIMIZATION*

Wayan Firdaus Mahmudy

Program Studi Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer,
Universitas Brawijaya, Malang
wayanfm@ub.ac.id

Abstrak

Tulisan ini memaparkan pengembangan *particle swarm optimization* (PSO) untuk optimasi *part type selection and machine loading problems* dalam *Flexible Manufacturing System* (FMS). Kedua permasalahan tersebut sangat mempengaruhi produktivitas FMS dan dikelompokan dalam permasalahan *NP-Hard*. Representasi bilangan pecahan yang selalu menghasilkan solusi yang *feasible* digunakan oleh PSO untuk mengekplorasi daerah pencarian solusi secara efisien. PSO menghasilkan solusi yang memperbaiki kinerja FMS berdasarkan dua fungsi obyektif, yaitu: memaksimalkan *throughput* sistem dan menjaga keseimbangan beban mesin. Hasil dua nilai obyektif ini dibandingkan dengan solusi optimum yang dihasilkan oleh metode enumerasi *branch-and-bound*. Serangkaian analisis komputasi menunjukkan bahwa PSO dapat menghasilkan solusi yang mendekati optimum dalam waktu rata-rata kurang dari 1 menit.

Kata kunci : *particle swarm optimization, part type selection, machine loading, flexible manufacturing system*

1. Pendahuluan

Flexible Manufacturing System (FMS) merupakan sistem berbasis teknologi tinggi yang dikembangkan untuk menghasilkan beragam produk dalam volume kecil sampai sedang. Mesin dan peralatan (*tools*) yang ada bisa secara cepat dikonfigurasi ulang untuk menghasilkan produk baru sesuai dengan permintaan pasar yang dinamis. Proses produksi dilakukan secara otomatis dan dikendalikan oleh pusat komputer [1, 2]. Implementasi FMS merupakan investasi berbiaya tinggi. Sebuah perencanaan produksi yang bagus dibutuhkan untuk menaikkan utilisasi sumber daya sistem, memaksimalkan hasil *(throughput*), dan menekan biaya produksi. Kesemua hal tersebut dibutuhkan untuk mengembalikan modal investasi sesegera mungkin (*early return on investment*) [3, 4].

Dalam perencanaan produksi sebuah FMS, *part type selection* dan *machine loading* merupakan permasalahan yang berhubungan erat dan sangat mempengaruhi produktivitas dan efisiensi sistem [5, 6]. Permasalahan *part type selection* berkaitan dengan pengambilan keputusan *part type* (produk) mana saja yang harus segera diproduksi dari sejumlah *part type* yang ada dalam antrian pesanan. Hal ini harus dilakukan karena ada keterbatasan mesin, kapasitas magasin pada mesin, dan peralatan. Permasalahan *machine loading* berkaitan dengan alokasi operasi yang diperlukan untuk meproduksi *part type* dan pemasangan peralatan yang sesuai pada mesin. Hal ini harus dilakukan karena sifat fleksibel dari FMS yang memungkinan sebuah *part type* diproduksi melalui sejumlah alternatif urutan mesin [7, 8].

Pentingnya permasalahan *part type selection* dan *machine loading* ditunjukkan dalam banyak literatur yang membahas berbagai model FMS beserta metode optimasinya. Beberapa metode yang digunakan untuk optimasi perencanaan produksi sebuah FMS misalnya algoritma genetika [3, 4, 8-10], *particle swarm optimization* [5, 11], *ant colony optimization* [12], dan *immune algorithm* [13, 14].

Makalah ini mengajukan optimasi permasalahan *part type selection* dan *machine loading* secara simultan. Pendekatan ini terbukti menghasilkan solusi yang lebih baik yang ditunjukkan oleh hasil *(throughput*) yang lebih tinggi dan alokasi sumber daya (mesin dan peralatan) yang lebih efisien. *Particle swarm optimization* (PSO) dipilih sebagai metode untuk optimasi karena terbukti berhasil untuk

Mahmudy, WF 2014, 'Optimasi part type selection and machine loading problems pada FMS menggunakan metode particle swarm optimization', *Konferensi Nasional Sistem Informasi (KNSI)*, STMIK Dipanegara, Makassar, 27 Februari - 1 Maret, pp. 1718-1723.

menyelesaikan berbagai permasalahan kombinatorial kompleks [5, 15]. PSO juga memiliki parameter yang lebih sedikit dibanding teknik optimasi yang lain sehingga memudahkan dalam implementasi dan percobaan penentuan parameter input [16, 17]. Sebagai algoritma pencarian berbasis populasi, PSO terbukti efektif digunakan pada permasalahan dengan area pencarian yang sangat luas [18]. Kinerja PSO ini dibandingkan dengan keluaran solusi optimum menggunakan metode enumerasi *branch-and-bound*.

2. Permasalahan Part Type Selection dan Machine Loading

Penelitian ini dilakukan pada sebuah FMS yang mempunyai *m* mesin. Setiap mesin dilengkapi dengan *tool magazine* dengan kapasitas slot tertentu. Mesin-mesin ini bisa melakukan operasi yang berbeda jika dipasangi *tool* yang berbeda. Sejumlah tipe *tool* tersedia dan setiap *tool* menempati sejumlah slot jika ditempatkan pada *tool magazine* yang ada pada mesin.

Sistem bisa memproduksi sejumlah *part type* yang berbeda. Setiap *part type* mempunyai aturan produksi yang ditunjukkan oleh urutan operasi pada mesin. Contoh kebutuhan operasi dari 7 *part type* ditunjukkan pada Table 1. ***qty*** menunjukkan banyaknya part type (kuantitas) yang harus diproduksi. ***op*** menunjukkan operasi. ***mac*** menunjukan mesin yang digunakan. ***tools*** menunjukan paralatan yang harus dipasang pada mesin.

Tabel 1. Kebutuhan Operasi 7 Part Type

part type	qty	nilai Rp	op	mac	time	tools
1	20	5	1	2	20	2 3 5
			2	1	30	4 5
			3	2	30	3 4
				3	30	5
2	20	3	1	1	30	1 3
			2	2	20	3 4
			3	2	30	4 6 7
3	40	2	1	2	30	6 7 8
				3	40	8 9 10
			2	2	20	1 10
				3	40	2 10
			3	1	20	1 2
4	20	1	1	2	30	9 10
				3	20	9 10
			2	2	30	6 7
				1	40	6 7
			3	1	30	3 4
5	30	4	1	2	40	1 2 3
			2	1	40	7 8
				2	30	3 4
6	30	3	1	3	20	7 8
			2	2	50	9 10
			3	3	10	2
7	30	5	1	1	50	1 2 3
				2	40	7 9 10
			2	3	30	4 6

Tabel 1 menunjukkan bahwa untuk memproduksi part type 1 diperlukan 3 operasi. Operasi pertama bisa dilakukan pada mesin 2 dengan waktu 20 satuan. Operasi ini memerlukan *tools* 2, 3, dan 5. Operasi ke-2 dilakukan pada mesin 1. Operasi ke-3 bisa dilakukan pada dua alternative mesin, yaitu pada mesin 2 atau 3 dengan *tool* dan waktu operasi yang berbeda.

Sebagai ilustrasi, part type pada Tabel 1 akan diproduksi pada sebuah FMS yang mempunyai spesifikasi mesin yang tunjukkan pada Table 2.

Tabel 2. Spesifikasi Mesin

mesin	kapasitas tool slot	waktu tersedia
1	20	2500
2	15	2500
3	20	2500

Sepuluh tipe *tool* tersedia seperti ditunjukkan pada Tabel 3. ***qty*** menunjukan ketersediaan *tool* tipe tersebut. ***slot*** menunjukan banyaknya slot yang dibutuhkan jika *tool* tipe tersebut dipasang pada *tool magazine* yang ada pada mesin.

Tabel 3. Ketersediaan *Tool*

tool	1	2	3	4	5	6	7	8	9	10
qty	2	2	2	2	2	3	3	3	3	3
slot	3	3	4	4	5	5	4	4	3	3

Dari deskripsi yang telah diberikan, model matematika permasalahan *part type selection* dan *machine loading* bisa disusun sebagai berikut:

2.1. Indeks dan Parameter

$p = 1,\ldots,P$ part type
$o = 1,\ldots,O_p$ operasi dari part type p
$t = 1,\ldots,T$ tipe *tool*
$m = 1,\ldots,M$ mesin

MS_m = kapasitas tool slot mesin m
W_m = waktu yang tersedia (dialokasikan) untuk mesin m
TQ_t = banyaknya (kuantitas) tool tipe t
TS_t = banyaknya slot pada mesin yang dibutuhkan oleh *tool* tipe t
Q_p = banyaknya (kuantitas) part type p yang harus diproduksi
V_p = nilai (rupiah) part type p
MAC_{po} = himpunan mesin alternatif untuk operasi o part type p
$TM_{pomt} = \{1,0\}$: 1 jika tool tipe t diperlukan untuk operasi o part type p pada mesin m, 0 jika sebaliknya
T_{pom} = waktu pemrosesan operasi o part type p pada mesin m

Mahmudy, WF 2014, 'Optimasi part type selection and machine loading problems pada FMS menggunakan metode particle swarm optimization', *Konferensi Nasional Sistem Informasi (KNSI)*, STMIK Dipanegara, Makassar, 27 Februari - 1 Maret, pp. 1718-1723.

2.2. Variabel Keputusan

Ada dua keputusan yang harus diambil, yaitu: (1) part type mana saja yang terpilih untuk diproduksi; (2) mesin mana saja yang digunakan untuk memproses setiap operasi dari part type yang terpilih. Dua keputusan ini dapat dinyatakan sebagai berikut:

$X_p = \{1,0\}$: 1 jika part type p terpilih untuk segera diproduksi, 0 jika sebaliknya
$Y_{pom} = \{1,0\}$: 1 jika mesin m terpilih untuk memproses operasi o part type p, 0 jika sebaliknya

Sebagian akibat dari dua keputusan di atas, muncul variabel tak bebas yang menyatakan tool tipe apa saja yang harus dipasangkan pada tiap mesin sehingga proses produksi bisa dilakukan. Variabel ini bisa dinyatakan sebagai:

$Z_{mt} = \{1,0\}$: 1 jika tool tipe t dipasang pada mesin m, 0 jika sebaliknya

2.3. Fungsi Obyektif

Kinerja FMS dinilai berdasarkan dua fungsi obyektif, yaitu: memaksimalkan throughput sistem (*th*) yang dinyatakan pada Persamaan (1) dan menjaga keseimbangan beban mesin yang dinyatakan sebagai meminimumkan ketidakseimbangan beban kerja mesin (*unb*) seperti ditunjukkan pada Persamaan (2).

$$\text{Maksimalkan: } th = \sum_{p=1}^{P} X_p Q_p V_p \quad (1)$$

$$\text{Minimumkan: } unb = \sum_{m=1}^{M} |W_m - B_m| \quad (2)$$

$$B_m = \sum_{p=1}^{P} \sum_{o=1}^{O_p} Y_{pom} T_{pom} Q_p$$

2.4. Kendala

Beberapa kendala yang berkaitan dengan sumber daya sistem dinyatakan sebagai berikut:

- Semua operasi untuk semua part type yang terpilih harus dilakukan:

$$\sum_{o=1}^{O_p} \sum_{m=1}^{M} Y_{pom} = O_p X_p, \quad \text{untuk semua } p \quad (3)$$

- Setiap operasi harus dilakukan pada satu mesin yang tepilih dari sejumlah alternatif mesin:

$$\sum_{m \in MAC_{po}} Y_{pom} = X_p, \quad \text{untuk semua } p, o \quad (4)$$

- Jika sebuah mesin terpilih untuk sebuah operasi maka semua tipe tool yang diperlukan harus terpasang:

$$Z_{mt} = Y_{pom} TM_{pomt}, \quad \text{untuk semua } p, o, m, t \quad (5)$$

- Banyaknya tool yang dipasang pada mesin tidak melebihi ketersediaan tool tersebut:

$$\sum_{m=1}^{M} Z_{mt} \leq TQ_t, \text{ untuk semua } t \quad (6)$$

- Banyaknya tool slots digunakan pada mesin tidak melebihi kapasitas tool slot mesin tersebut:

$$\sum_{t=1}^{T} Z_{mt} TS_t \leq MS_m, \text{ untuk semua } m \quad (7)$$

3. Particle Swarm Optimization (PSO)

PSO menggunakan *particle* sebagai representasi dari solusi dari permasalahan yang akan dioptimasi. Sebagai algoritma meta-heuristic berbasis populasi maka sejumlah *np* particle berada dalam kelompok. Setiap particle disusun atas vektor posisi x_i dan vektor kecepatan v_i. Sebuah fungsi fitness pada Persamaan (8) digunakan untuk mengevaluasi seberapa bagus sebuah particle berdasarkan fungsi obyektif pada Persamaan (1) dan (2). α_1 dan α_2 merupakan parameter pembobot untuak dua fungsi obyektif. Sebuah particle dengan nilai F lebih besar dianggap sebagai calon solusi yang lebih baik.

$$F = \frac{th}{\sum_{p=1}^{P} Q_p V_p} \alpha_1 + \left(1 - \frac{unb}{\sum_{m=1}^{M} W_M}\right) \alpha_2 \quad (8)$$

3.2. Siklus PSO

Selama siklus PSO, setiap particle bergerak menjelajahi daerah pencarian solusi dengan kecepatan yang berubah secara dinamis berdasarkan posisi terbaik yang pernah dicapai dirinya sendiri dan posisi terbaik yang dicapai semua particle dalam kelompok seperti ditunjukkan pada Persamaan (9) [16, 19]. Posisi particle diupdate menggunakan Persamaan (10).

$$\acute{v}_i = wv_i + c_1 r_1 (xb_i - x_i) + c_2 r_2 (xgb_i - x_i), \quad i = 1, \ldots, n \quad (9)$$

$$\acute{x}_i = x_i + \acute{v}_i, \quad i = 1, .., n \quad (10)$$

n merupakan panjang particle. w merupakan vektor inersia dan dalam penelitian ini ditetapkan sebesar 0,4. xb_i adalah posisi terbaik yang pernah dicapai oleh particle pada interasi sebelumnya. xgb_i adalah posisi terbaik yang dicapai semua particle dalam kelompok. c_1 adalah sebuah konstanta yang disebut *self-recognition component* dan c_2 adalah sebuah konstanta yang disebut *social component*. Dalam penelitian ini, nilai c_1 dan c_2 ditentukan sebesar 1. r_1 and r_2 adalah bilangan acak pada interval [0,1].

Siklus pemutakhiran nilai posisi x_i dan vektor kecepatan v_i diulangi terus sampai kondisi berhenti

Mahmudy, WF 2014, 'Optimasi part type selection and machine loading problems pada FMS menggunakan metode particle swarm optimization', *Konferensi Nasional Sistem Informasi (KNSI)*, STMIK Dipanegara, Makassar, 27 Februari - 1 Maret, pp. 1718-1723.

tercapai. Pada interasi terakhir, particle terbaik diuraikan menjadi solusi optimum atau mendekati optimum. Siklus lengkap PSO ini ditunjukkan pada Gambar 1.

Langkah 0: Tentukan parameter PSO
Parameter: *np*, *w*, c_1, c_2

Langkah 1: Inisialisasi
Buat *np* secara acak

Langkah 2: Update
Update posisi tiap particle dan hitung nilai fitness-nya.
Update posisi terbaik tiap particle.
Update posisi terbaik keseluruhan particle (global).

Langkah 3: Cek kondisi berhenti.
Jika *lanjut* loncat ke Langkah 2, jika tidak Stop.

Gambar 1. Siklus PSO

3.2. Konversi Particle ke Solusi

Sebuah particle tersusun atas vektor bilangan pecahan dengan panjang sesuai banyaknya *part type* yang akan diproduksi. Mekanisme konversi yang diadopsi dari *Real-Coded Genetic Algorithm* (RCGA) [3] digunakan dalam penelitian ini. Sebagai contoh untuk permasalahan pada Table 1, particle dengan nilai posisi *x*=(778, 500, 307, 757, 490, 547, 490) dapat dikonversi menjadi solusi dengan part type yang terpilih adalah 3, 7 dan 5. Nilai dari part type (perkalian nilai/item dengan kuantitas) dan mesin yang digunakan untuk operasi disajikan pada Tabel 2.

Tabel 2. Part Type Terpilih

part type	nilai	mesin
3	80	3, 2, 1
7	150	1, 3
5	120	2, 1
throughput	350	

Beban kerja untuk tiap mesin disajikan pada Tabel 3. *m* merujuk pada mesin. *st* merupakan banyaknya slot terpakai. *tools* merupakan tipe tool yang terpasang pada mesin. Simbol yang lain sudah diuraikan pada deskripsi sistem.

Tabel 3. Beban Kerja Mesin

m	W_m	B_m	$\|W_m-B_m\|$	slot MS_m	*st*	*tools*
1	2500	3500	1000	20	18	1, 2, 3, 7, 8
2	2500	2000	500	15	13	1, 2, 3, 10
3	2500	2500	0	20	19	4, 6, 8, 9, 10
	unb		1500			

Dari Tabel 2 dan Tabel 3 didapatkan nilai fungsi obyektif yaitu throughput sistem (*th*) sebesar 350 dan ketidakseimbangan beban kerja mesin (*unb*) sebesar 1500.

4. Analisis Komputasi

Analisis komputasi dilakukan untuk mengevaluasi kinerja PSO terhadap solusi optimum yang dihasilkan metode *branch-and-bound*.

4.1. Desain Percobaan

Untuk mengevaluasi kinerja PSO, dua belas data uji dari [3, 4] yang tersedia pada ‘http://lecture.ub.ac.id/anggota/wayanfm/data_test/’ digunakan seperti ditunjukkan secara ringkas pada Table 4. Problem 1 sampai 4 mewakili data berukuran kecil, problem 5 sampai 8 mewakili data berukuran sedang, dan sisanya mewakili data berukuran besar. *P* menunjukkan banyaknya part type. *M* menunjukkan banyaknya mesin. *T* menunjukkan banyaknya tipe tool.

Data uji ini sudah dilengkapi dengan solusi optimum yang dihasilkan dengan metode *branch-and-bound*. Parameter data uji yang lain silahkan langsung merujuk [3, 4]. Penulis makalah ini menggarisbawahi bahwa meskipun metode *branch-and-bound* bisa digunakan untuk mencari solusi optimum, waktu komputasi yang dibutuhkan terlalu tinggi dan tidak mungkin digunakan pada perencanaan produksi harian.

Tabel 4. Data Uji

problem	*P*	*M*	*T*	W_m
1	8	4	20	4000
2	8	5	25	4000
3	10	4	20	4000
4	10	5	25	4000
5	16	4	20	7000
6	16	5	25	7000
7	18	4	20	7000
8	18	5	25	7000
9	24	4	20	10000
10	24	5	25	10000
11	26	4	20	10000
12	26	5	25	10000

Program komputer untuk PSO ditulis menggunakan Java. Percobaan dilakukan pada PC dengan prosesor AMD Quad Core yang berkerja pada kecepatan 2,80GHz. Percobaan dilakukan untuk membuktikan efektifitas PSO terhadap solusi optimum. Karena PSO bersifat stokastis maka hasil yang berbeda akan didapatkan setiap kali program dijalankan pada data uji yang sama. Untuk mendapatkan hasil dan kesimpulan yang valid maka

Mahmudy, WF 2014, 'Optimasi part type selection and machine loading problems pada FMS menggunakan metode particle swarm optimization', *Konferensi Nasional Sistem Informasi (KNSI)*, STMIK Dipanegara, Makassar, 27 Februari - 1 Maret, pp. 1718-1723.

Tabel 5. Hasil Percobaan

problem	Solusi Optimum			PSO					
	F	*th*	*unb*	*NOS*	*waktu*	*F*	*th*	*unb*	*DEV*
1	2,545	1.616	803	18	7,71	2,522	1.605.7	1.000.1	0,89
2	2,926	2.591	9.838	20	8,70	2,926	2.591.0	9.838.0	0,00
3	2,972	3.058	6.858	4	9,42	2,905	3.086.8	8.284.5	2,24
4	2,531	2.196	3.233	10	10,12	2,470	2.136.4	3.538.6	2,42
		Rata-rata			8,99				1,39
5	2,156	2.676	3.738	0	26,65	1,918	2.262.7	4.823.7	11,04
6	1,968	2.605	7.126	1	28,18	1,779	2.337.1	9.520.9	9,60
7	2,458	3.595	5.529	0	30,43	2,159	2.749.7	3.012.7	12,18
8	2,088	2.871	4.768	3	32,03	1,922	2.624.4	6.879.5	7,93
		Rata-rata			29,32				10,19
9	2,349	4.150	4.204	0	60,30	1,907	3.301.3	10.012.8	18,84
10	1,809	3.212	10.879	0	64,59	1,542	2.648.7	15.236.7	14,77
11	2,305	4.417	5.519	0	53,51	1,936	3.572.2	9.228.1	16,00
12	2,018	3.937	9.291	0	55,56	1,672	3.422.6	18.734.7	17,15
		Rata-rata			58,49				16,69

untuk setiap data uji percobaan diulang sebanyak 20 kali.

Uji coba dilakukan dengan menetapkan $\alpha_1 = 3$ dan $\alpha_2 = 1$. Serangkaian percobaan pendahuluan dilakukan untuk mendapatkan kombinasi nilai parameter yang tepat bagi PSO. Hasil percobaan pendahuluan disajikan sebagai berikut:

- Banyaknya particle sebesar 500, 1000, dan 1500 untuk kelompok data uji berukuran kecil, sedang, dan besar.
- Banyaknya iterasi sebesar 2000.

Dua parameter digunakan untuk mengevaluasi kinerja PSO. Yang pertama adalah banyaknya solusi optimum yang diperoleh (*number of optimum solutions*/NOS) untuk 20 kali percobaan per data uji. Parameter yang kedua adalah rata-rata deviasi solusi PSO (*FPSO*) terhadap solusi optimum (F_{opt}) seperti ditunjukkan pada Persamaan (11). *DEV* yang lebih kecil menunjukkan hasil yang lebih baik.

$$DEV = \frac{F_{opt} - \left(\sum_{i=1}^{20} FPSO_i \Big/ 20 \right)}{F_{opt}} 100\% \qquad (11)$$

4.2. Hasil dan Pembahasan

Hasil keseluruhan percobaan disajikan pada Tabel 5. Pada data uji berukuran kecil, PSO mampu memberikan hasil optimum pada mayoritas percobaan. Hal ini ditunjukan dengan nilai NOS yang relatif besar (mendekati 20). Bahkan pada problem 2, PSO memberikan hasil sempurna pada semua percobaan. Hasil yang baik ini juga ditunjukkan dengan rata-rata nilai *DEV* yang relatif kecil yaitu sebesar 1,39%.

Pada data berukuran sedang, PSO masih mempu memberikan hasil optimum pada beberapa percobaan. Hasil yang baik ini juga ditunjukkan dengan rata-rata nilai *DEV* yang relatif kecil yaitu sebesar 10,19%.

Pada data berukuran besar, meskipun PSO tidak mampu memberikan hasil optimum, rata-rata nilai *DEV* yang dicapai masih di bawah 17%. Hasil ini dicapai dalam waktu rata-rata kurang dari 1 menit.

5. Kesimpulan dan Saran

Optimasi permasalahan *part type selection* dan *machine loading* telah diselesaikan dengan *particle swarm optimization* (PSO). Serangkaian percobaan menunjukkan bahwa PSO mampu menghasilkan solusi optimum dan mendekati optimum dalam waktu relatif cepat, yaitu kurang dari 1 menit untuk data berukuran besar.

Penelitian ke depan akan memperhatikan integrasi perencanaan dan penjadwalan produksi pada FMS. Pada kasus ini, selain memaksimalkan *throughput* sistem dan menjaga keseimbangan beban mesin, PSO juga harus meminimumkan total keterlambatan (*tardiness*) dari semua part type. PSO yang lebih baik dikembangkan dangan melakukan hibridisasi dengan metode heuristik lain seperti *simulated annealing*, *tabu search*, dan *variable neighborhood search* (VNS).

6. Daftar Pustaka

[1] I. Badr, "An agent-based scheduling framework for flexible manufacturing systems," *International Journal of Computer, Information, and Systems Science, and Engineering*, vol. 2, pp. 123-129, 2008.

Mahmudy, WF 2014, 'Optimasi part type selection and machine loading problems pada FMS menggunakan metode particle swarm optimization', *Konferensi Nasional Sistem Informasi (KNSI)*, STMIK Dipanegara, Makassar, 27 Februari - 1 Maret, pp. 1718-1723.

[2] D. J. Parrish, *Flexible Manufacturing*. Oxford: Butterworth-Heinemann, 1993.

[3] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Solving part type selection and loading problem in flexible manufacturing system using real coded genetic algorithms – Part I: modeling," *World Academy of Science, Engineering and Technology*, vol. 69, pp. 699-705, 2012.

[4] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Solving part type selection and loading problem in flexible manufacturing system using real coded genetic algorithms – Part II: optimization," *World Academy of Science, Engineering and Technology*, vol. 69, pp. 706-710, 2012.

[5] S. Biswas and S. Mahapatra, "Modified particle swarm optimization for solving machine-loading problems in flexible manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 39, pp. 931-942, 2008.

[6] M. K. Tiwari, S. Kumar Jha, and R. Bardhan Anand, "Operation allocation and part type selection in e-manufacturing: An auction based heuristic supported by agent technology," *Robotics and Computer-Integrated Manufacturing*, vol. 26, pp. 312-324, 2010.

[7] K. E. Stecke, "Design, planning, scheduling, and control problems of flexible manufacturing systems," *Annals of Operations Research*, vol. 3, pp. 1-12, 1985.

[8] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Optimization of part type selection and loading problem with alternative production plans in flexible manufacturing system using hybrid genetic algorithms – Part 1: modelling and representation," in *5th International Conference on Knowledge and Smart Technology (KST)*, Chonburi, Thailand, 2013, pp. 75-80.

[9] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Optimization of part type selection and loading problem with alternative production plans in flexible manufacturing system using hybrid genetic algorithms – Part 2: genetic operators & results," in *5th International Conference on Knowledge and Smart Technology (KST)*, Chonburi, Thailand, 2013, pp. 81-85.

[10] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Hybrid genetic algorithms for multi-period part type selection and machine loading problems in flexible manufacturing system," in *IEEE International Conference on Computational Intelligence and Cybernetics*, Yogyakarta, Indonesia, 2013, pp. 126-130.

[11] S. G. Ponnambalam and L. S. Kiat, "Solving machine loading problem in flexible manufacturing systems using particle swarm optimization," *World Academy of Science, Engineering and Technology*, vol. 39, 2008.

[12] P. Udhayakumar and S. Kumanan, "Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm," *Int J Adv Manuf Technol*, vol. 50, pp. 1075-1084, 2010.

[13] A. Prakash, N. Khilwani, M. K. Tiwari, and Y. Cohen, "Modified immune algorithm for job selection and operation allocation problem in flexible manufacturing systems," *Adv. Eng. Softw.*, vol. 39, pp. 219-232, 2008.

[14] P. R. Dhall, S. S. Mahapatra, S. Datta, and A. Mishra, "An improved artificial immune system for solving loading problems in flexible manufacturing systems," presented at the Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on, 2010.

[15] F. Goksal, I. Karaoglan, and F. Altiparmak, "A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery," *Computers & Industrial Engineering*, 2012.

[16] R. C. Eberhart and J. Kennedy, "A new optimizer using particles swarm theory," in *Sixth Int Symposium on Micro Machine and Human Science*, 1995, pp. 39-43.

[17] H. Rania, C. Babak, W. Olivier de, and V. Gerhard, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, ed: American Institute of Aeronautics and Astronautics, 2005.

[18] W. F. Mahmudy, "Optimisation of Integrated Multi-Period Production Planning and Scheduling Problems in Flexible Manufacturing Systems (FMS) Using Hybrid Genetic Algorithms " Ph.D., School of Engineering, University of South Australia, 2013.

[19] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. San Mateo, CA, USA: Morgan Kaufmann, 2001.

Mahmudy, WF 2014, 'Optimasi part type selection and machine loading problems pada FMS menggunakan metode particle swarm optimization', *Konferensi Nasional Sistem Informasi (KNSI)*, STMIK Dipanegara, Makassar, 27 Februari - 1 Maret, pp. 1718-1723.