



Penyelesaian Penjadwalan Flexible Job Shop untuk Minimasi Due Windows dengan Algoritma Genetika

Ferdinan Rinaldo Tampubolon

Polliteknik Negeri Medan

Email: ferdinantampubolon@polmed.ac.id

Artikel info

Artikel history:

Diterima : 15-05-2021

Direvisi : 10-06-2021

Disetujui : 15-06-2021

Keywords:

due windows, flexible job shop; genetika

Abstract:

Flexible Job Shop (FJSP) is the development of a job shop where there is more than one machine that can process a job. This study aims to model and complete the FJSP with a due windows objective function where each job must be completed within a certain time span. The method used in this research is the Genetic Algorithm method. The result obtained is scheduling in the form of a Gantt Chart which has a minimum penalty value.

Abstrak:

Flexible Job Shop (FJSP) adalah pengembangan dari *job shop* dimana terdapat lebih dari satu mesin yang dapat memproses suatu pekerjaan. Penelitian bertujuan untuk memodelkan dan menyelesaikan FJSP dengan fungsi tujuan *due windows* dimana tiap-tiap pekerjaan harus selesai dalam suatu rentang waktu. Metode yang digunakan dalam penelitian ini adalah metode Algoritma Genetika. Hasil yang diperoleh adalah penjadwalan dalam bentuk *Gantt Chart* yang memiliki nilai penalti yang minimum.

Kata kunci:

due windows, flexible job shop; genetik

Corresponden author: Ferdinan Rinaldo Tampubolon

Email: ferdinantampubolon@polmed.ac.id
artikel dengan akses terbuka dibawah lisensi

CC BY SA 2021



Pendahuluan

Penjadwalan adalah suatu permasalahan dalam bidang Operasi Riset dimana terdapat sejumlah sumber daya yang terbatas untuk menyelesaikan banyak pekerjaan. Pengambil keputusan harus membuat prioritas untuk memilih pekerjaan yang harus diselesaikan terlebih dahulu berdasarkan tujuan yang akan dicapai (Saudagar, Kamboj, Mohan, Patil, & Powar, 2020).

Job Shop adalah salah satu jenis penjadwalan dimana sumber daya berupa mesin dan masing-masing pekerjaan mempunyai operasi-operasi yang harus diproses secara berurutan oleh satu buah mesin. *Flexible Job Shop* (FJSP) adalah pengembangan dari *Job Shop* dimana tiap operasi dapat diproses oleh beberapa mesin. Sehingga waktu pemrosesan dari suatu operasi bergantung pada mesin yang terpilih untuk menyelesaikan operasi tersebut. (Marzouki, Driss, & Ghédira, 2018). *Flexible Job Shop* memiliki kerumitan yang lebih tinggi dari *Job Shop*

karena yang harus diselesaikan tidak hanya menentukan pekerjaan mana yang harus diselesaikan terlebih dahulu akan tetapi juga mesin yang akan dipilih untuk menyelesaikan pekerjaan tersebut. (Ahmadi, Zandieh, Farrokh, & Emami, 2016)

Fungsi tujuan yang paling umum dari *Job Shop* adalah meminimumkan *makespan* yaitu penyelesaian keseluruhan pekerjaan. Pada kasus tertentu suatu pekerjaan mempunyai prioritas untuk diselesaikan terlebih dahulu. (Alzahrani, 2019). Fungsi tujuan *Job Shop* yang demikian adalah *tardiness*, yaitu apabila pekerjaan diselesaikan melebihi waktu yang diberikan maka akan diberikan penalti. Hal ini sesuai dengan keadaan sehari-hari dimana *customer* sewajarnya akan menuntut kompensasi apabila barang yang diantarkan terlambat. (Kim, Jun, Bang, Shin, & Choi, 2020). JIT (Just In Time Scheduling) adalah suatu konsep dimana suatu pekerjaan harus diselesaikan sesuai waktu yang ditentukan. Hal ini sesuai dengan realita dimana suatu barang yang diselesaikan terlalu cepat akan membuat barang menumpuk di gudang (*earliness*). (Idowu, Adamu, & Sawyerr, 2020). Pada JIT terdapat penalti apabila pekerjaan diselesaikan lebih cepat (*earliness*) ataupun lebih lama (*tardiness*) dari waktu yang ditentukan (*due date*). (Ahmadian, Salehipour, & Cheng, 2021). *Due date* yang berupa interval waktu disebut sebagai *due windows*. Suatu pekerjaan diharapkan dapat diselesaikan pada suatu rentang waktu. Penalti akan diberikan apabila pekerjaan diselesaikan tidak berada pada rentang waktu yang diberikan (Lin, 2020).

Pada penelitian Idowu, Lin, Ahmadian yang sudah dilaksanakan berbasis pekerjaan yang diharapkan dapat diselesaikan pada suatu rentang waktu. Penalti akan diberikan apabila pekerjaan diselesaikan tidak berada pada rentang waktu yang diberikan, konsep tersebut diterapkan pada kasus mesin tunggal (single machine), sedangkan pada penelitian ini konsep tersebut akan diterapkan pada kasus flexible job shop, dimana terdapat beberapa mesin yang dapat menyelesaikan beberapa buah operasi (multi purpose machine) Penelitian ini akan meminimumkan *due windows* pada penjadwalan FJSP. Model matematis akan dibentuk terlebih dahulu. Selanjutnya model tersebut akan diselesaikan dengan Algoritma Genetika, suatu algoritma yang paling populer untuk menyelesaikan FJSP (Amjad et al., 2018). Penggunaan *elite strategy* maupun *dispatching rule* pada insialisasi populasi akan menghasilkan penjadwalan FJSP dengan penalti yang minimum.

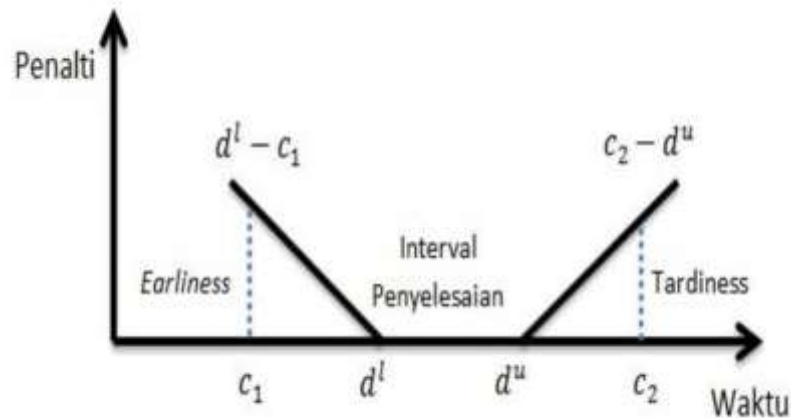
Metode Penelitian

Penelitian ini merupakan studi literatur, dimulai dengan menyusun model matematis dari permasalahan kemudian menyelesaikan permasalahan tersebut dengan metode Algoritma Genetika. Untuk melihat apakah metode yang digunakan sudah cukup baik dilakukan perbandingan hasil dari penelitian sebelumnya. Adapun langkah-langkah penelitian sebagai berikut:

Model Matematis

Langkah pertama dalam penelitian ini adalah menyusun pemodelan matematis terkait *Flexible Job Shop* dengan fungsi tujuan setiap pekerjaan selesai pada rentang waktu tertentu (*due windows*). Pada penelitian ini pekerjaan masing-masing memiliki interval waktu dimana pekerjaan diharapkan dapat diselesaikan pada interval tersebut $\{d_i^l, d_i^u\}$. Apabila pekerjaan diselesaikan diluar selang waktu tersebut maka akan diberikan penalti. Gambar 1

menunjukkan C_1 menunjukkan penyelesaian lebih cepat dari yang ditentukan, sementara C_2 menunjukkan penyelesaian yang dilakukan melewati batas waktu.



Gambar 1 Due Windows

Kendala-kendala dari penjadwalan *flexible job shop* harus memenuhi kriteria-kriteria dari *job shop*, menurut (Kuhpfahl, 2015) sebagai berikut:

1. Setiap mesin hanya dapat memproses satu pekerjaan dalam suatu waktu interval
2. Setiap operasi pada tiap pekerjaan harus dikerjakan secara berurutan
3. Setiap operasi hanya dapat diselesaikan oleh satu buah mesin tanpa *delay*
4. Operasi yang berbeda pada suatu pekerjaan tidak dapat dikerjakan bersamaan.

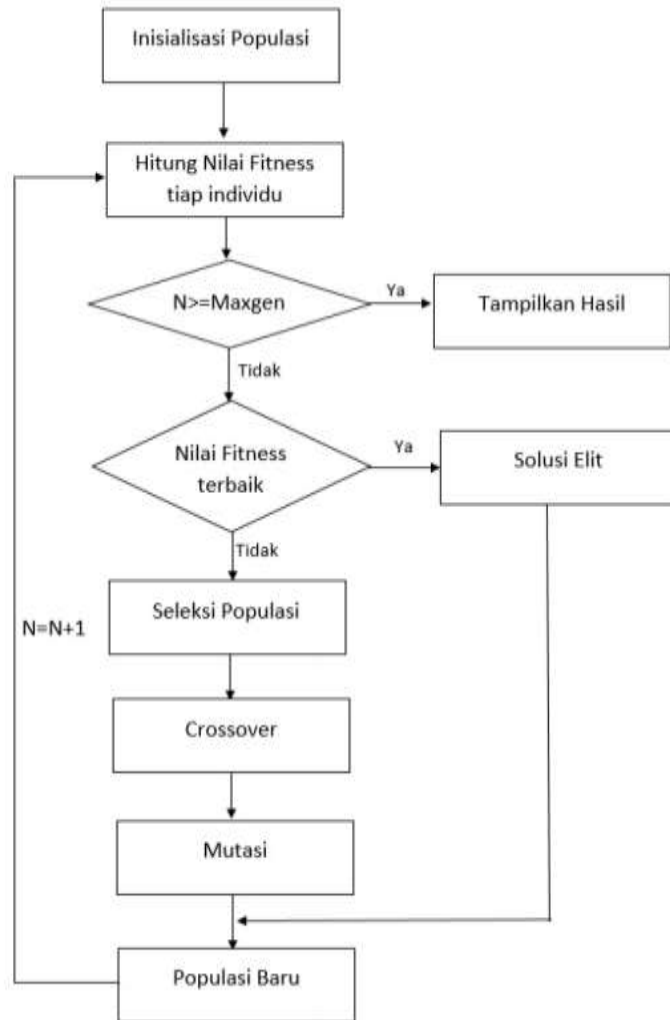
Penyelesaian Dengan Algoritma Genetika

Langkah-langkah penyelesaian Algoritma Genetika yaitu inialisasi populasi, nilai fitness, seleksi, persilangan dan mutasi. Inialisasi populasi dilakukan dengan membangkitkan beberapa kromosom urutan pekerjaan dan kromosom penugasan mesin. Bentuk kromosom seperti penelitian (Bierwirth, 1995). Kromosom-kromosom tersebut akan dimunculkan berdasarkan metode heuristik sederhana. Kromosom penugasan mesin berdasarkan (Kacem, Hammadi, & Borne, 2002). Kromosom urutan pekerjaan menggunakan *dispatching rule* (Michael, 2018) yaitu *Most Operation work remaining*, *Most operation remaining Earliest due data* dan *Random*.

Nilai fitness menghitung penalti *due windows* untuk tiap-tiap kromosom akan dihitung ($F(x)$). Semakin kecil nilai $F(x)$ maka akan semakin baik kromosom tersebut. Dengan demikian nilai *fitness* adalah $f(x) = \frac{1}{F(x)}$. Kromosom terbaik akan disimpan untuk dimunculkan pada iterasi selanjutnya. Hal ini dilakukan agar kromosom tersebut tidak mengalami perubahan-perubahan pada langkah persilangan maupun mutasi.

Proses seleksi dilakukan dengan metode *Roulette Wheel Selection* yaitu dengan membangkitkan sembarang bilangan acak r dan memilih kromosom yang memiliki nilai kumulatif individu lebih kecil dari r untuk tetap bertahan. Proses persilangan dilakukan dengan mempertukarkan gen antara dua kromosom yaitu kromosom pendonor dan reseptor. Kedua kromosom ini akan disebut sebagai kromosom induk. Kromosom pendonor akan memberikan beberapa gen kepada kromosom reseptor sehingga menghasilkan anak (*offspring*) Proses mutasi adalah perubahan gen pada suatu kromosom sehingga

menghasilkan kromosom baru. Karena penelitian ini merupakan *Flexible Job Shop* maka mutasi dilakukan secara dua tahap yaitu untuk urutan pekerjaan dan penugasan mesin seperti pada penelitian (Piroozfard, Wong, & Wong, 2018). Proses ini akan dilakukan secara berulang-ulang sampai maksimum iterasi (MaxGen) tercapai. Berikut adalah *flowchart* Algoritma Genetika



Gambar 2 *Flowchart* Algoritma Genetika

Pengumpulan data dan Perbandingan Hasil

Data yang digunakan adalah kasus *Flexible Job Shop* dari (L Yin, X Li, L Gao, C Lu, & Z Zhang, 2017). Karena penelitian ini memiliki fungsi tujuan yang berbeda maka untuk menguji keakuratan algoritma ini, dilakukan perbandingan hasil untuk fungsi tujuan *makespan*. Semakin kecil hasil untuk *makespan* yang maka semakin baik performa algoritam tersebut. Sementara untuk *due windows* dilakukan penambahan data untuk $\{d_i^l, d_i^u\}$ pada tiap-tiap pekerjaan secara acak. Software yang digunakan adalah Matlab 2014a. Berikut adalah *flowchart* dari Algoritma Genetika

Hasil dan Pembahasan

Model Matematis

Model matematis *Mixed Integer Programming* dengan fungsi tujuan meminumkan *due windows* adalah sebagai berikut:

Notasi yang digunakan

- i, g : indeks *job*, $i = 1.., n, g = 1 ... , n$
- j, h : indeks operasi $j = 1.., m, h = 1 ... , m$
- k : indeks operasi $k = 1.., o$
- p_{ijk} : waktu pemrosesan untuk operasi O_{ij} dengan mesin- k
- d_i^u : batas atas waktu penyelesaian pekerjaan ke- i
- d_i^l : batas bawah waktu penyelesaian pekerjaan ke- i
- T_i : *tardiness* pekerjaan ke- i , $i = 1.., n$
- E_i : *earliness* pekerjaan ke- i , $i = 1.., n$
- c_{ij} : waktu penyelesaian operasi O_{ij}
- c_k : waktu penyelesaian operasi terakhir pada mesin ke- k
- w_e : penalti untuk *earliness*
- w_t : penalti untuk *tardiness*

Variabel biner yaitu

$$x_{ijk} = \begin{cases} 1, & \text{operasi } O_{ij} \text{ diselesaikan dengan mesin } - k \\ 0, & \text{lainnya} \end{cases}$$

Fungsi tujuan dan kendala

$$\min D = w_e \sum_{i=1}^n E_i + w_t \sum_{i=1}^n T_i \quad (1)$$

Kendala

$$\sum_{k=1}^o x_{ijk} = 1 \quad \forall i, j \quad (2)$$

$$c_k \geq c_{ij} \cdot x_{ijk} \quad (3)$$

$$T_i = \max_{i=1}^n \{0, c_i - d_i^u\} \quad (4)$$

$$E_i = \max_{i=1}^n \{d_i^l - c_i, 0\} \quad (5)$$

$$c_i = \max_{i=1}^n \{c_{ij}\} \quad (6)$$

$$c_{ij} = s_{ij} + (p_{ijk} + x_{ijk}) \quad \forall i, j, k \quad (7)$$

$$c_{ij} \leq s_{i(j+1)} \quad \forall i, j \quad (8)$$

$$s_{ij} + p_{ijk} \leq s_{gh} + M(1 - y_{ijghk}) \quad \forall i, j, g, h, k \quad (9)$$

$$s_{gh} + p_{ghk} \leq s_{ij} + My_{ijghk} \quad \forall i, j, g, h, k \quad (10)$$

$$s_{ij} \geq 0, s_{gh} \geq 0, c_{ij} \geq 0 \quad (11)$$

Persamaan (1) menunjukkan fungsi tujuan meminimumkan due windows, kendala (2) memastikan tiap operasi diselesaikan oleh satu buah mesin, kendala (3) menunjukkan waktu pemrosesan operasi terakhir pada mesin, kendala (4) dan (5) menunjukkan perhitungan untuk *tardiness* dan *earliness*, dimana nilai keduanya harus bernilai positif, kendala (6) menunjukkan waktu penyelesaian pekerjaan yaitu waktu pemrosesan terbesar untuk pekerjaan tersebut, kendala (7) menunjukkan waktu penyelesaian operasi adalah penjumlahan waktu operasi mulai dikerjakan dan waktu pemrosesan, kendala (8) menunjukkan setiap operasi harus dikerjakan secara berurutan, kendala (9) dan (10) memastikan tiap mesin hanya dapat memroses suatu pekerjaan dalam suatu selang waktu

Penyelesaian dengan Algoritma Genetika

Data waktu pemrosesan mesin diperoleh dari (Lvjiang Yin, Xinyu Li, Liang Gao, Chao Lu, & Zhao Zhang, 2017) dengan penambahan data untuk rentang waktu keterlambatan *due windows*, D_l untuk *earliness* dan D_u untuk *tardiness*, maka persoalan yang akan diselesaikan seperti pada tabel 4.1

Tabel 1 Kasus 4 Job dan 5 Mesin

Job	Operasi	M_1	M_2	M_3	M_4	M_5	D_l	D_u
J_1	O_{11}	9	6	-	12	-	10	12
	O_{12}	-	-	6	-	7		
	O_{13}	-	2	4	-	-		
J_2	O_{21}	5	-	-	7	9	7	12
	O_{22}	-	4	6	8	-		
	O_{23}	-	-	4	-	8		
J_3	O_{31}	3	-	-	-	-	8	13
	O_{32}	-	3	-	5	-		
	O_{33}	8	-	10	-	13		
J_4	O_{41}	6	-	9	-	10	9	12
	O_{42}	-	3	-	-	-		
	O_{43}	2	-	-	4			

Untuk memastikan program sudah berjalan dengan baik, dilakukan pengujian dengan menggunakan penelitian (Lvjiang Yin et al., 2017) sebagai perbandingan untuk kasus single objective *makespan*. Parameter yang digunakan adalah, jumlah populasi =200, jumlah

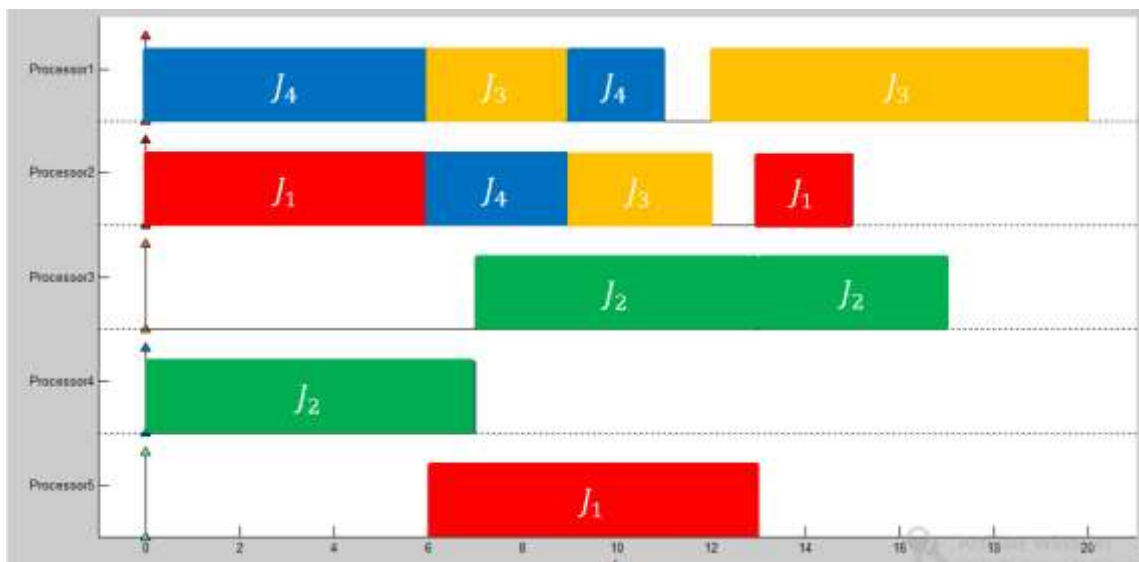
iterasi=100, probabilitas persilangan=0,9, probabilitas mutasi=0,8, dan probabilitas elit=0,1 Pada penelitian ini diperoleh hasil 17 yang mana sama dengan hasil pada penelitian tersebut.

Untuk penyelesaian *due windows* dilakukan pada beberapa kombinasi bobot untuk *earliness* (W_e) dan bobot untuk *tardiness* (W_t). Berikut hasil yang diperoleh

Tabel 2 Kasus 4 Job dan 5 Mesin

Bobot	Due Windows	Makespan
$W_e = 0,25; W_t = 0,75$	11,25	20
$W_e = 0,5; W_t = 0,5$	7,5	20
$W_e = 0,75; W_t = 0,25$	3,75	20

Dari tabel terlihat penalti terkecil diperoleh pada $W_e = 0,75; W_t = 0,25$. Hal ini terjadi karena penalti yang diberikan apabila pengerjaan terlambat lebih besar daripada pengerjaan yang terlalu cepat. Bentuk *Gantt Chart* untuk permasalahan tersebut adalah sebagai berikut.



Gambar 3 Gantt Chart

Jika dilihat pada diagram *Gantt Chart due windows* juga berkaitan erat dengan nilai makespan, semakin kecil *due windows* yang didapat maka semakin kecil nilai *makespan* yang diperoleh. Dari gambar terlihat hanya pekerjaan J_2 dan J_4 yang waktu penyelesaiannya berada pada selang waktu yang diberikan. J_2 diselesaikan pada waktu 11, sehingga berada pada rentang $\{7,12\}$, sementara J_4 diselesaikan pada waktu 11 berada pada rentang $\{9,12\}$

Kesimpulan

Due Windows menjadi alternatif untuk Penjadwalan Flexible Job Shop terkait dengan penyelesaian pekerjaan. Selang waktu yang diberikan memberikan toleransi pada penyelesaian pekerjaan sehingga lebih sesuai dengan kehidupan sehari-hari. Performa dari

Algoritma Genetika cukup baik dalam menyelesaikan kasus *Flexibel Job Shop*, terbukti hasil yang cukup baik terkait dengan hasil *makespan*. Penelitian ini dapat dikembangkan dengan melakukan penggabungan algoritma (*hybrid*) untuk hasil yang lebih baik.

Bibliografi

- Ahmadi, E., Zandieh, M., Farrokh, M., & Emami, S. M. (2016). [A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms](#). *Computers & Operations Research*, 73, 56-66.
- Ahmadian, M. M., Salehipour, A., & Cheng, T. (2021). [A meta-heuristic to solve the just-in-time job-shop scheduling problem](#). *European Journal of Operational Research*, 288(1), 14-29.
- Alzahrani, J. S. (2019). [Job Shop Scheduling Considering Makespan, Penalties Of Machine Idling, And Job Out Of Time](#). *International Journal of Research-Granthaalayah*, 7(1), 73-82.
- Amjad, M. K., Butt, S. I., Kousar, R., Ahmad, R., Agha, M. H., Faping, Z., . . . Asgher, U. (2018). [Recent research trends in genetic algorithm based flexible job shop scheduling problems](#). *Mathematical Problems in Engineering*, 2018.
- Bierwirth, C. (1995). [A generalized permutation approach to job shop scheduling with genetic algorithms](#). *Operations-Research-Spektrum*, 17(2), 87-92.
- Idowu, G. A., Adamu, M. O., & Sawyerr, B. S. (2020). [Hybrid Metaheuristic for Just In Time Scheduling in a Flow Shop with Distinct Time Windows](#). *International Journal of Mathematical Sciences and Optimization: Theory and Applications*, 2020(1), 741-756.
- Kacem, I., Hammadi, S., & Borne, P. (2002). [Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems](#). *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(1), 1-13.
- Kim, J.-G., Jun, H.-B., Bang, J.-Y., Shin, J.-H., & Choi, S.-H. (2020). [Minimizing Tardiness Penalty Costs in Job Shop Scheduling under Maximum Allowable Tardiness](#). *Processes*, 8(11), 1398.
- Kuhpfahl, J. (2015). [Job shop scheduling with consideration of due dates: Potentials of local search based solution techniques](#): Springer.
- Lin, S.-S. (2020). [Due-Window Assignment and Resource Allocation Scheduling with Truncated Learning Effect and Position-Dependent Weights](#). *Discrete Dynamics in Nature and Society*, 2020.
- Marzouki, B., Driss, O. B., & Ghédira, K. (2018). [Solving distributed and flexible job shop scheduling problem using a chemical reaction optimization metaheuristic](#). *Procedia Computer Science* 126, 1424-1433.
- Michael, L. P. (2018). [Scheduling: theory, algorithms, and systems](#): Springer.
- Piroozfard, H., Wong, K. Y., & Wong, W. P. (2018). [Minimizing total carbon footprint and](#)

total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. *Resources, Conservation and Recycling*, 128, 267-283.

Saudagar, S., Kamboj, A., Mohan, N., Patil, S., & Powar, N. (2020). [Resource Allocation and Task Scheduling with Skill Level and Time Bound Constraints](#). *International Journal of Industrial and Manufacturing Engineering*, 14(9), 732-738.

Yin, L., Li, X., Gao, L., Lu, C., & Zhang, Z. (2017). [A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem](#). *Sustainable Computing: Informatics and Systems* 13, 15-30.