



UNIVERSITAS  
GADJAH MADA

# Metode Optimasi

A genetic algorithm for scheduling open shops with conflict graphs to minimize makespan

**Nour El Houda Tellache dan Laoucine Kerbache (2023)**

---

Gusti Agung Rama Ayudhya  
20/459266/PA/19927

LOCALLY ROOTED,  
GLOBALLY RESPECTED

ugm.ac.id



UNIVERSITAS  
GADJAH MADA

# Pengenalan

# Open Shop Scheduling Problem (OSSP)

Masalah kombinatorial penjadwalan (*scheduling*),

- Himpunan  $n$  pekerjaan (*job*)  $J = \{J_j : j = 1, \dots, n\}$
- Himpunan  $m$  mesin  $M = \{M_i : i = 1, \dots, m\}$
- Setiap  $J_j \in J$  memiliki himpunan  $m$  operasi  $O_j = \{J_{ij} : i = 1, \dots, m\}$ , dengan  $J_{ij}$  adalah operasi job  $J_j$  yang diproses oleh mesin  $M_i$
- Operasi  $J_{ij}$  diproses selama  $p_{ij} \in \mathbb{N}$  satuan waktu secara ***non-preemptive***
- Setiap mesin hanya dapat memproses paling banyak satu operasi dalam satu waktu dan setiap job hanya dapat diproses oleh satu mesin dalam satu waktu.

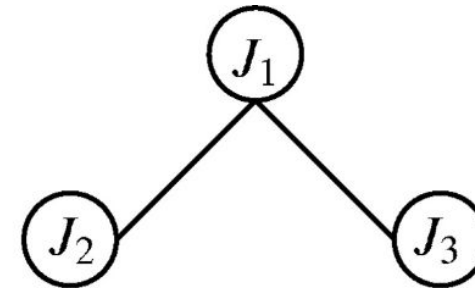
# Conflict Graph

- Graf sederhana tak berarah  $G = (V, E)$
- $V = J$
- Setiap **dua job** di  $V$  yang bertetangga (***adjacent***) pada  $G$  dikatakan ***conflict*** atau tidak boleh diproses dalam waktu bersamaan, termasuk jika dua job diproses oleh dua mesin berbeda
- Dua operasi  $J_{ij}$  dan  $J_{kl}$  dikatakan ***conflict*** jika
  - $J_j$  dan  $J_l$  ***conflict*** atau
  - $j = l$  (merupakan job yang sama) atau
  - $i = k$  (diproses oleh mesin yang sama)

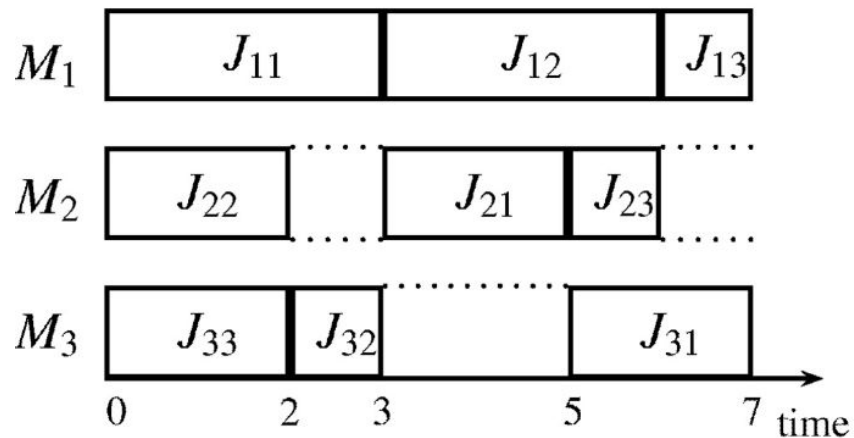
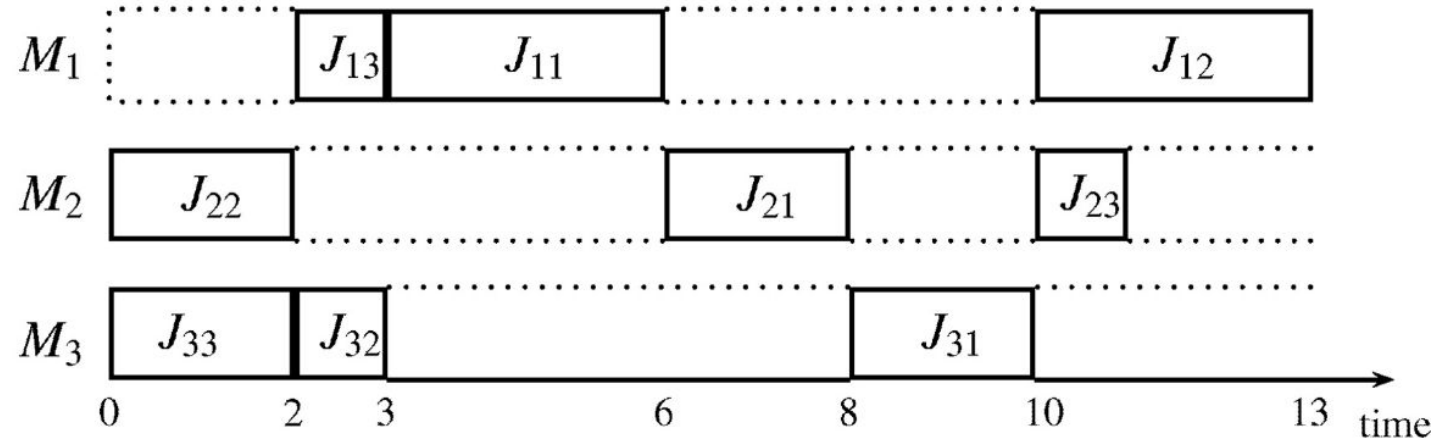
# Objektif

- Susunlah **jadwal** (tentukan waktu awal dan akhir pemrosesan setiap operasi) yang **meminimalkan *makespan*** dan **memenuhi syarat *no-conflict*** berdasarkan *conflict graph*
- ***Makespan*** adalah waktu yang dibutuhkan untuk menyelesaikan semua operasi
- ***No-conflict*** berarti tidak ada sebarang dua *job* yang *conflict*

|       | $M_1$ | $M_2$ | $M_3$ |
|-------|-------|-------|-------|
| $J_1$ | 3     | 2     | 2     |
| $J_2$ | 3     | 2     | 1     |
| $J_3$ | 1     | 1     | 2     |

Tabel  $p_{ij}$ 


Conflict graph


Jadwal yang **tidak feasible**

Jadwal yang **feasible**



UNIVERSITAS  
GADJAH MADA

# Formulasi Matematika

# Parameter Permasalahan

- Banyak job  $n \in \mathbb{N}^+$
- Banyak mesin  $m \in \mathbb{N}^+$
- Waktu pemrosesan operasi  $p_{ij} \in \mathbb{N}$  operasi  $J_{ij} \in O_j, j = 1, \dots, n$
- ( *Adjacency matrix*  $\mathbf{A}$  ) *conflict graph*  $\mathbf{G}$



# Variabel Keputusan

$C_{ij} \in \mathbb{N}$  : waktu selesai *job*  $J_j$  pada mesin  $M_i$

# Definisi Variabel

$x_{ijk}$  { 1, jika oleh mesin  $M_i$ , *job*  $J_j$  diproses sebelum *job*  $J_k$   
0, sebaliknya

$y_{ii'j}$  { 1, jika mesin  $M_i$  lebih dulu daripada  $M_{i'}$ , ketika memproses *job*  $J_j$   
0, sebaliknya

$a_{jk}$  { 1, jika  $J_j$  dan  $J_k$  *conflict*  
0, sebaliknya

$r_{iji'k}$  { 1, jika  $a_{jk} = 1$  dan operasi  $J_{ij}$  diproses sebelum operasi  $J_{i'k}$   
0, sebaliknya

# Fungsi Objektif dan Batasan

$$\min \quad C_{max} := \max \{C_{ij} : i = 1, \dots, m ; j = 1, \dots, n\} \quad (1)$$

$$\text{s.t.} \quad C_{max} \geq C_{ij} \quad i = 1, \dots, m ; j = 1, \dots, n \quad (2)$$

$$C_{ij} - M(1 - x_{ijk}) \leq C_{ik} - p_{ik} \quad i = 1, \dots, m ; 1 \leq k \neq j \leq n \quad (3)$$

$$x_{ijk} + x_{ikj} = 1 \quad i = 1, \dots, m ; 1 \leq k \neq j \leq n \quad (4)$$

$$C_{ij} - M(1 - y_{ii'j}) \leq C_{i'j} - p_{i'j} \quad 1 \leq i \neq i' \leq m ; j = 1, \dots, n \quad (5)$$

$$y_{ii'j} + y_{i'ij} = 1 \quad 1 \leq i \neq i' \leq m ; j = 1, \dots, n \quad (6)$$

$$C_{ij} - M(1 - r_{iji'k}) \leq C_{i'k} - p_{i'k} \quad \text{jika } a_{jk} = 1, 1 \leq i \neq i' \leq m ; 1 \leq k \neq j \leq n \quad (7)$$

$$r_{iji'k} + r_{i'kij} = 1 \quad \text{jika } a_{jk} = 1, 1 \leq i \neq i' \leq m ; 1 \leq k \neq j \leq n \quad (8)$$

# Fungsi Objektif dan Batasan

$$\text{s.t.} \quad C_{ij} \geq p_{ij} \quad i = 1, \dots, m ; j = 1, \dots, n \quad (9)$$

$$x_{ijk}, y_{ii'j}, r_{iji'k} \in \{0, 1\} \quad 1 \leq i \neq i' \leq m ; 1 \leq k \neq j \leq n \quad (10)$$

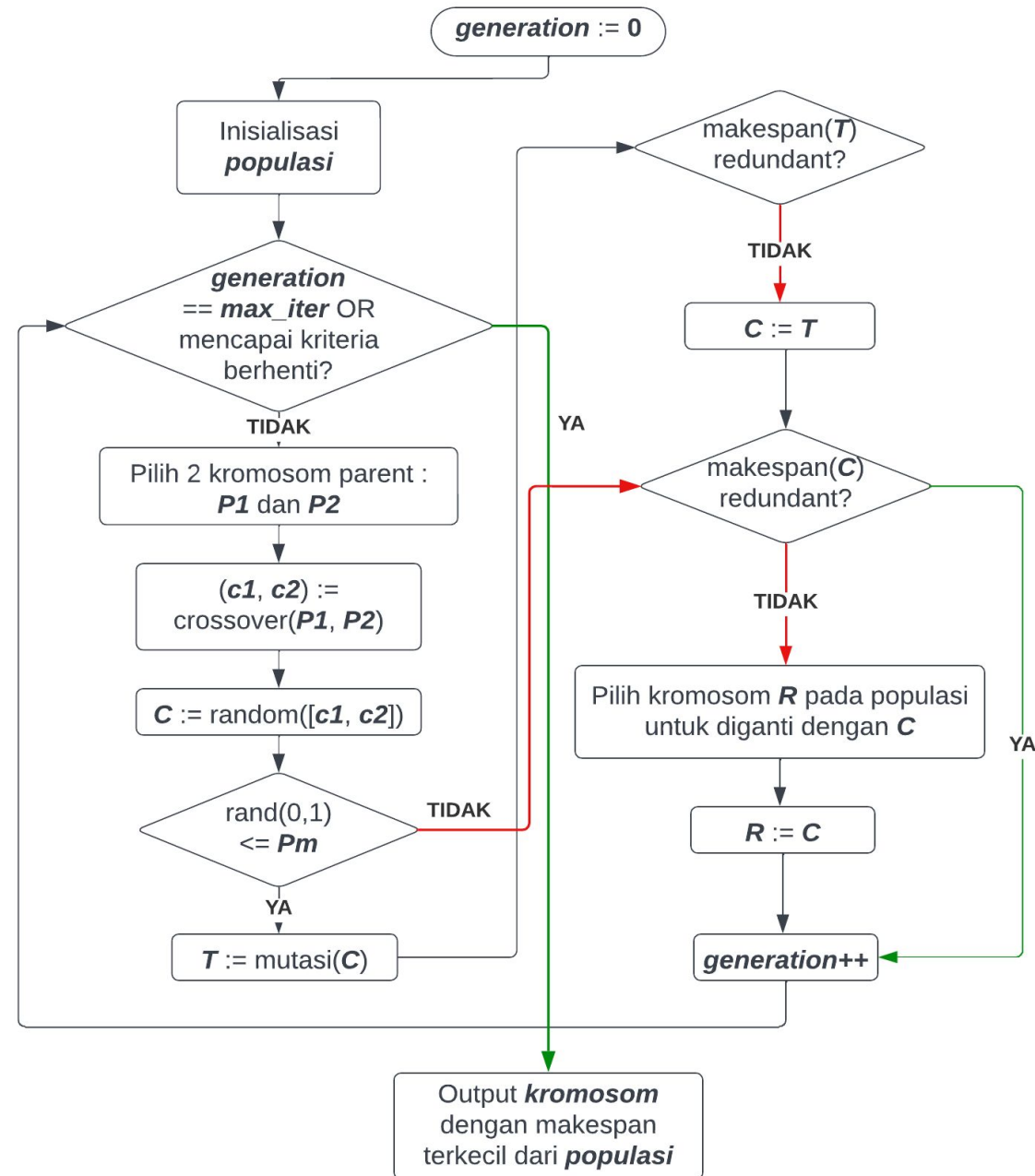


UNIVERSITAS  
GADJAH MADA

# Algoritma Genetika (GA)

# Skema Algoritma Genetika

- Representasi kromosom
- Fungsi *fitness*
- Populasi awal
- Operasi *crossover*
- Operasi mutasi
- Prosedur seleksi
- Prosedur *update* generasi
- Kriteria berhenti



# Representasi Kromosom

- Tiap kromosom merupakan **permutasi** dari  $\{J_{ij} : i = 1, \dots, m ; j = 1, \dots, n\}$
- **Urutan operasi** pada kromosom menentukan **urutan penjadwalan** operasi pada jadwal dengan memperhatikan syarat ***non-conflict***
- Jadwal dibentuk menggunakan satu dari tiga algoritma :
  1. ***Active schedule builder 1***
  2. ***Active schedule builder 2*** (Giffler dan Thompson, 1960)
  3. ***Non-delay schedule builder***



# Fungsi Fitness

- ***Fitness*** dari tiap kromosom **sama dengan *makespan*** dari jadwal yang dibentuk berdasarkan kromosom

# Algoritma 1

---

**Algorithm 1** Build active schedules

---

**Inputs:** permutation  $\pi$ , processing times  $(p_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$ , adjacency matrix  $A$  of  $G$

- 1: Initialize  $G(M_i)$  and  $G(J_j)$  to  $\{[0, +\infty[ \}$  for each  $M_i$  and  $J_j$ ;
- 2: **for**  $o = 1, \dots, |\pi|$  **do**
- 3:     Find interval  $[s, t] = [a, b] \cap [c, d]$  for  $J_{ij} = \pi(o)$  such that  $[a, b] \in G(M_i)$ ,  $[c, d] \in G(J_j)$ ,  $s$  is minimal, and  $t - s \geq p_{ij}$ ;
- 4:      $c_{ij} = s + p_{ij}$ ;
- 5:     Update  $G(M_i)$ ,  $G(J_j)$ , and  $G(J_k)$  for every  $k$  such that  $A[j][k] = 1$ ;

**Outputs:** completion times  $(c_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$  and the makespan.

---

# Algoritma 2

**Algorithm 2** Build active schedules by Giffler and Thompson mechanism (?)

**Inputs:** permutation  $\pi$ , processing times  $(p_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$ , adjacency matrix  $A$  of  $G$

- 1: Initialize the earliest starting times  $(s_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$  of all the operations to zero;
- 2: **while**  $\pi \neq \emptyset$  **do**
- 3:     Select an operation  $J_{i'j'}$  of  $\pi$  with the minimum earliest completion time  $s_{i'j'} + p_{i'j'}$ ;
- 4:     Select the first operation  $J_{ij}$  of  $\pi$  that is in conflict with  $J_{i'j'}$  (including  $J_{i'j'}$ ) such that  $s_{ij} < s_{i'j'} + p_{i'j'}$ ;
- 5:      $c_{ij} = s_{ij} + p_{ij}$ ;
- 6:      $\pi = \pi \setminus \{J_{ij}\}$ ;
- 7:     **for**  $J_{i''j''} \in \pi$  that are in conflict with  $J_{ij}$  **do**
- 8:         **if**  $s_{i''j''} < s_{ij} + p_{ij}$  **then**
- 9:              $s_{i''j''} = s_{ij} + p_{ij}$ ;

**Outputs:** completion times  $(c_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$  and the makespan.

# Algoritma 3

---

**Algorithm 3** Build non-delay schedules

---

**Inputs:** permutation  $\pi$ , processing times  $(p_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$ , adjacency matrix  $A$  of  $G$

- 1: Initialize the earliest starting times  $(s_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$  of all the operations to zero;
- 2: **while**  $\pi \neq \emptyset$  **do**
- 3:     Select the first operation  $J_{ij}$  of  $\pi$  with the minimum earliest starting time  $s_{ij}$ ;
- 4:      $c_{ij} = s_{ij} + p_{ij}$ ;
- 5:      $\pi = \pi \setminus \{J_{ij}\}$ ;
- 6:     **for**  $J_{i'j'} \in \pi$  that are in conflict with  $J_{ij}$  **do**
- 7:         **if**  $s_{i'j'} < s_{ij} + p_{ij}$  **then**
- 8:              $s_{i'j'} = s_{ij} + p_{ij}$ ;

**Outputs:** completion times  $(c_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$  and the makespan.

---

# Populasi awal

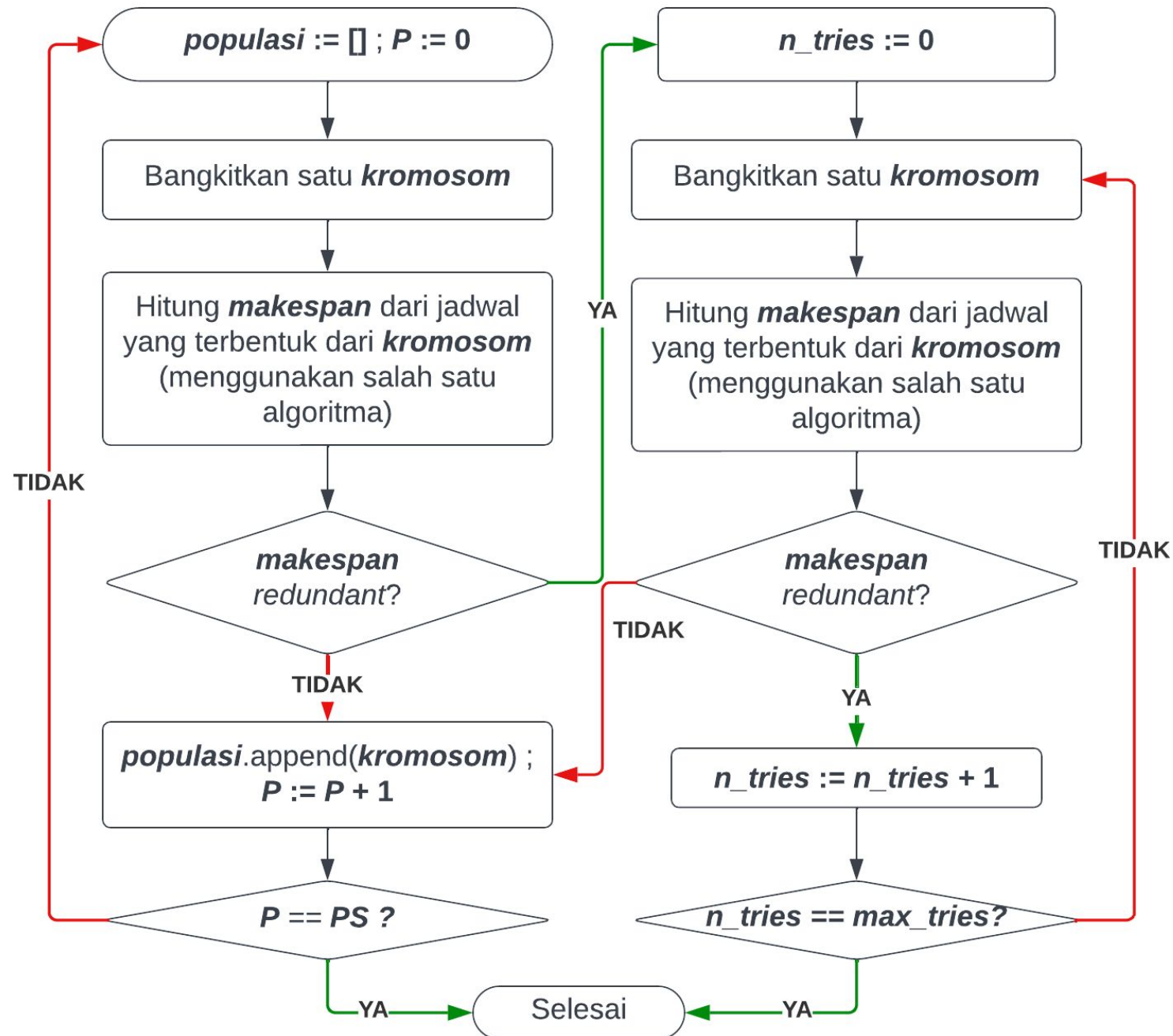
GA membangkitkan sebanyak **PS** kromosom berbeda,

1. secara **acak**, atau
2. dengan **mengurutkan operasi** berdasarkan 8 skala prioritas sebagai *seed*
  - a. urutan  $p_{ij}$  secara menaik atau menurun
  - b. urutan  $f_{ij}$  secara menaik atau menurun
  - c. urutan  $f_{ij} / p_{ij}$  secara menaik atau menurun
  - d. urutan  $g_{ij} / p_{ij}$  secara menaik atau menurun

$f_{ij}$  = banyak operasi yang conflict dengan operasi  $J_{ij}$

$g_{ij}$  = banyak operasi yang tidak conflict dengan operasi  $J_{ij}$

**PS** = ukuran populasi

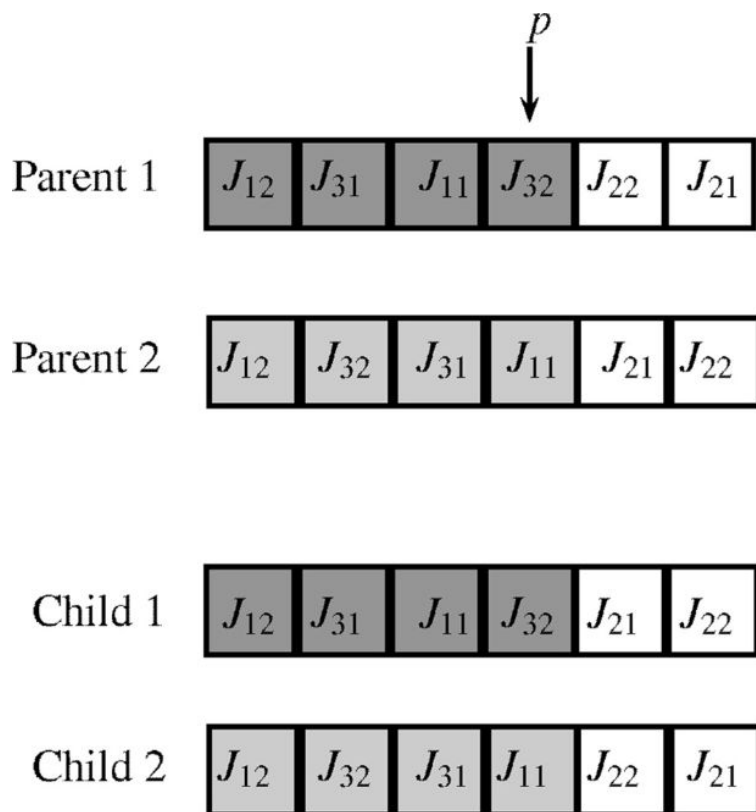


# Operasi *crossover*

Menggunakan salah satu dari tiga operasi ini :

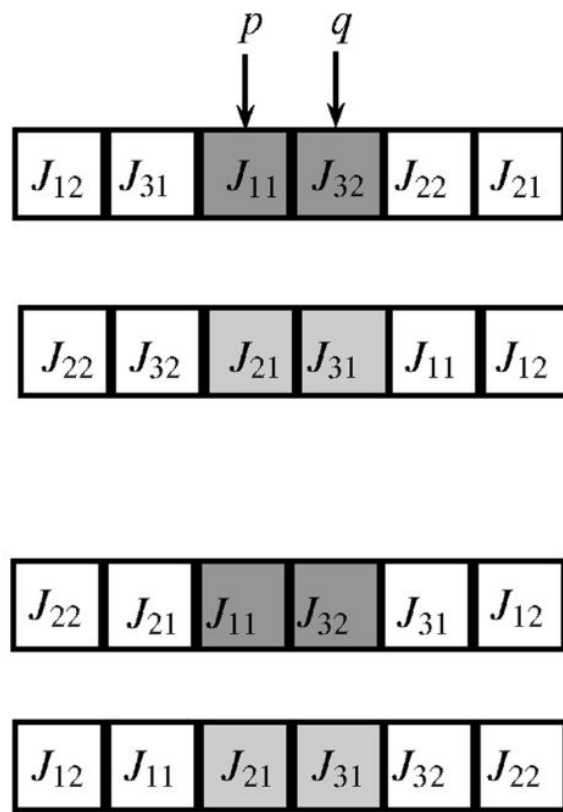
- ***One-point crossover (X1)***
- ***Order crossover (OX)***
- ***Linear order crossover (LOX)***





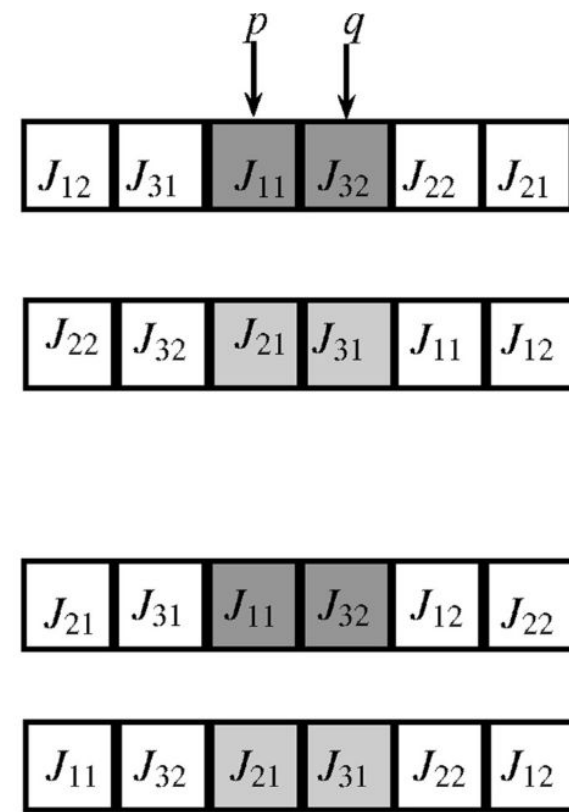
(a)

**X1**



(b)

**OX**



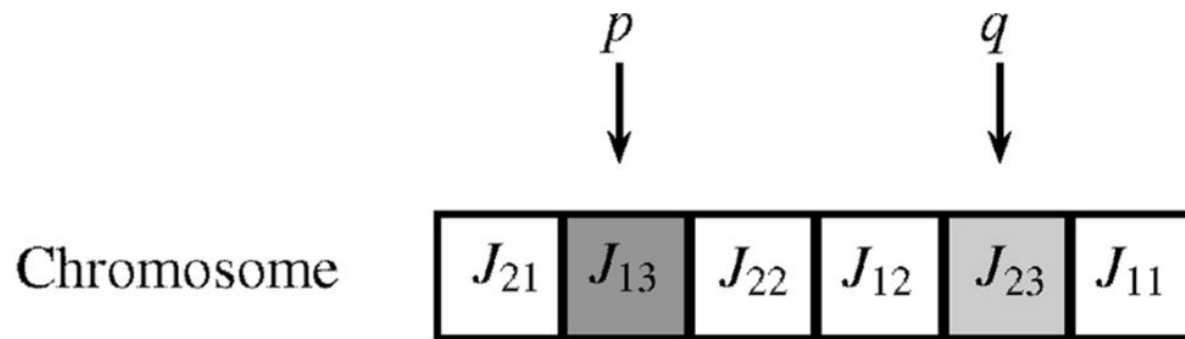
(c)

**LOX**

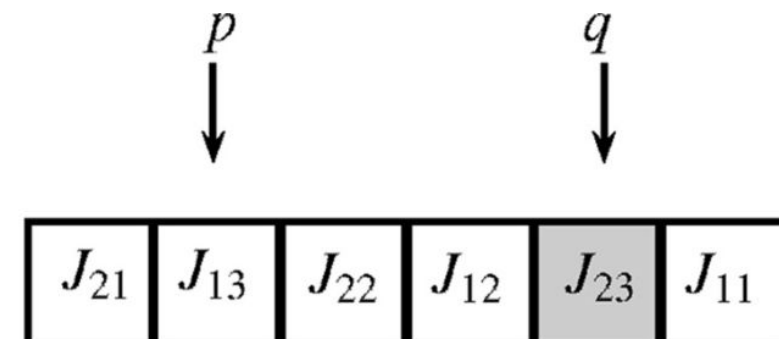


# Operasi mutasi

Menggunakan salah satu dari dua operasi ini : (a) **swap** atau (b) **move**



( a )



( b )

# Prosedur seleksi dan *update* generasi

- **Urutkan** populasi berdasarkan nilai ***fitness*** kromosom secara menurun
- Kromosom urutan ke- $k$  dipilih sebagai *parent* 1 dengan probabilitas  $2k/(PS(PS + 1))$
- Kromosom lainnya dipilih sebagai *parent* 2 dengan probabilitas  $1/(PS-1)$
- **Crossover** *parent* 1 dan *parent* 2 sehingga menghasilkan dua children
- **Pilih** salah satu *child* dari kedua *children* secara **acak**
- Lakukan **mutasi** terhadap *child* dengan probabilitas mutasi  $p_m$
- Pilih salah satu kromosom dari kromosom urutan **ke-1** hingga **ke- $\lfloor PS/2 \rfloor$**  secara **acak** untuk **digantikan** dengan ***child*** tadi

# Kriteria berhenti

- Iterasi sudah mencapai *max\_iter*
- Jadwal (kromosom) dengan *makespan* (*fitness*) **sama dengan batas bawah terbaik** (Tellache dan Kerbache, 2023)

# Batas bawah

- Iterasi sudah mencapai *max\_iter*
- Jadwal (kromosom) dengan *makespan* (*fitness*) **sama dengan batas bawah terbaik** (Tellache dan Kerbache, 2023)

# Referensi

- [1] N. E. Tellache dan L. Kerbache, “A genetic algorithm for scheduling open shops with conflict graphs to minimize the makespan,” *Computers & Operations Research*, vol. 156, p. 106247, 2023. doi:10.1016/j.cor.2023.106247
- [2] S. Sakai, Mitsunori Togasaki, and K. Yamazaki, “A note on greedy algorithms for the maximum weighted independent set problem,” *Discrete Applied Mathematics*, vol. 126, no. 2–3, pp. 313–322, Mar. 2003, doi:10.1016/s0166-218x(02)00205-6.



UNIVERSITAS  
GADJAH MADA

**TERIMA KASIH**

