

The background is a solid red color. A large, stylized arch made of small red dots spans the top half of the image, framing the central text.

# HUST

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



ĐẠI HỌC  
BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY  
OF SCIENCE AND TECHNOLOGY

# JOB SHOP SCHEDULING PROBLEM

ONE LOVE. ONE FUTURE.  
HÀ NỘI, 9/2023

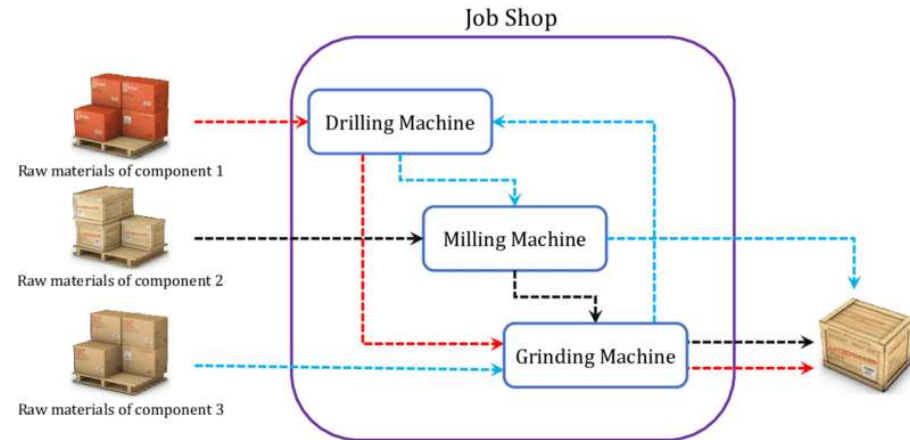
# 1. MÔ TẢ BÀI TOÁN

Cho tập hợp  $J$  bao gồm  $n_j$  công việc:  $J = \{ j_1, j_2, \dots, j_{n_j} \}$ . Mỗi công việc bao gồm  $n_i$  thao tác:  $j_{n_j} = \{ O_1, O_2, \dots, O_{n_i} \}$ . Và tập hợp:  $M = \{ m_1, m_2, \dots, m_m \}$  gồm  $m$  máy khác nhau.

➤ *Ràng buộc:*

- Mỗi máy chỉ xử lý được một thao tác trong cùng một thời gian
- Trong mỗi công việc thì các thao tác phải được xử lý lần lượt

➤ *Yêu cầu bài toán:* Sắp xếp các thao tác lên các máy sao cho đảm bảo các điều kiện ràng buộc và tối ưu hóa được các thông số sản xuất (ví dụ: Thời gian hoàn thành hết các công việc là nhỏ nhất, thời gian rảnh nhỏ nhất, ....)



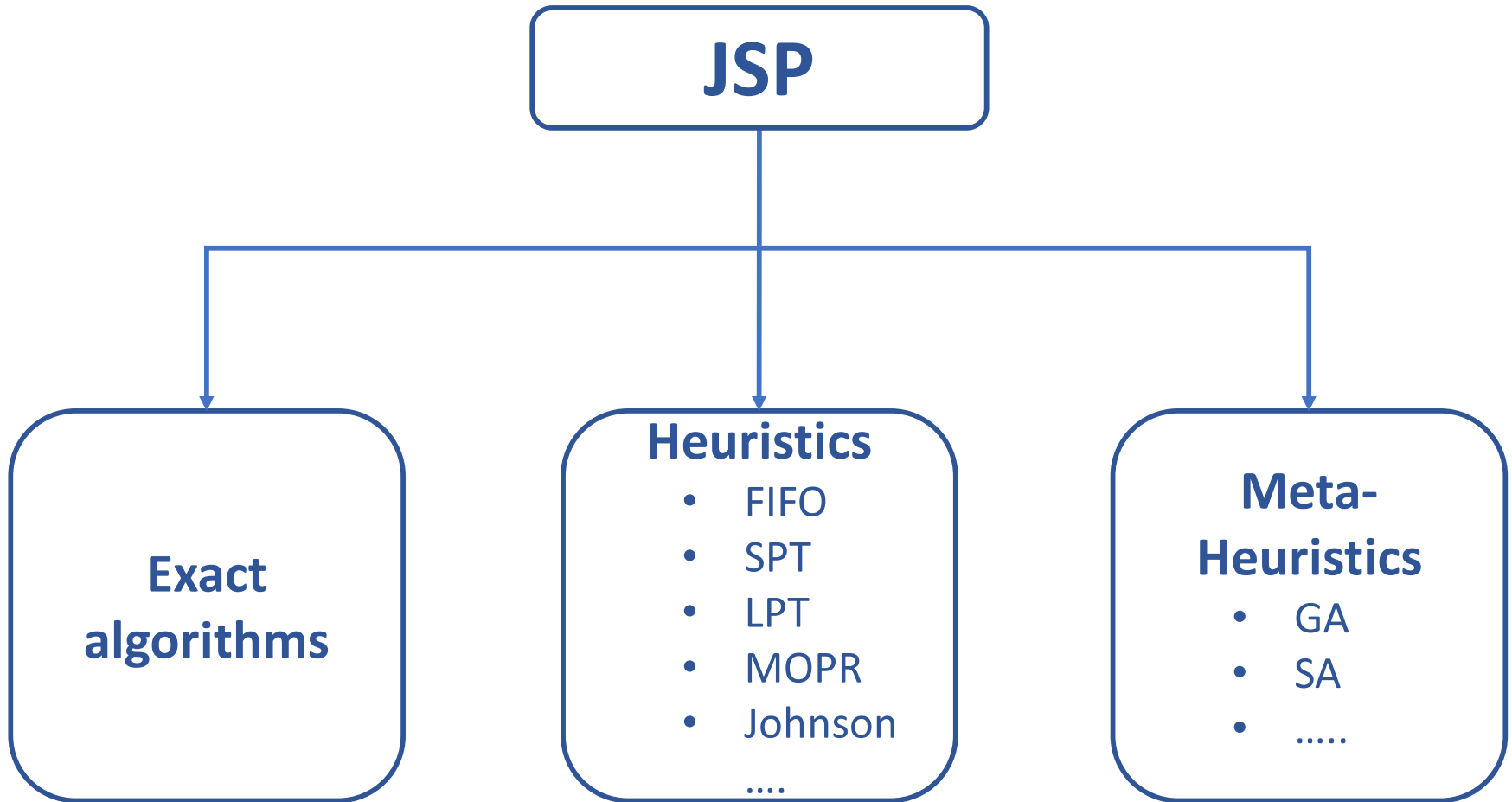
# 1. MÔ TẢ BÀI TOÁN

➤ *Yêu cầu bài toán: Một số các điều kiện phổ biến trong lập kế hoạch sản xuất*

Notation	Description	Meaning	Interpretation
$C_{\max}$	$\max_j (C_j)$	makespan or maximum completion time	cost of a schedule depends on how long the entire set of jobs has finished processing
$T_{\max}$	$\max_j (T_j)$	maximum tardiness	maximum difference between the completion time and the due date of a single job
$T_t$	$\sum T_j$	total tardiness	positive difference between the completion time and the due date of all jobs and there is no reward for early jobs and only penalties incurred for late jobs
$\bar{T}$	$(\sum T_j)/n$	mean tardiness	average difference between the completion time and the due date of a single job
$L_{\max}$	$\max_j (L_j)$	maximum lateness	check how well the due dates are respected, and there is a positive reward for completing a job early
$I_t$	$\sum I_j$	total idle time	difference between running time and processing time of all machines
$F_t$	$\sum F_j$	total flow time	time that all jobs spent in the shop
$\bar{F}$	$(\sum F_j)/n$	mean flow time	average time a single job spent in the shop
$W_{\max}$	$\max_j (W_j)$	maximum workload	maximum working time among all machines
$W_t$	$\sum W_j$	total workload	total working time on all machines
$O_t$	$\sum O_j$	total operation cost	cost value of all operations
$E_t$	$\sum E_j$	total energy consumption	energy consumption of the whole production process

# 1. MÔ TẢ BÀI TOÁN

## ➤ Phân loại các phương pháp

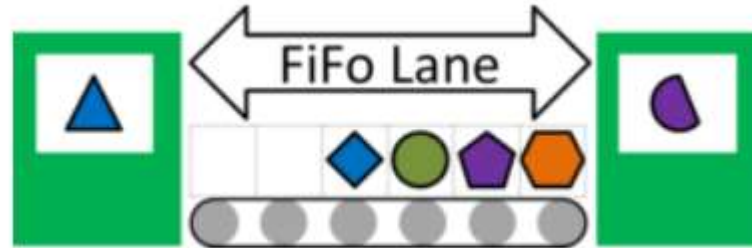




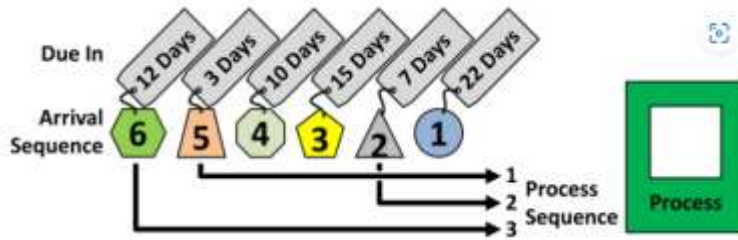
# 1. MÔ TẢ BÀI TOÁN

## Heuristics

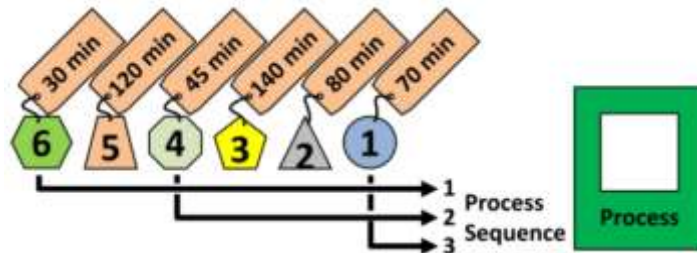
### ➤ FIFO



### ➤ EDD

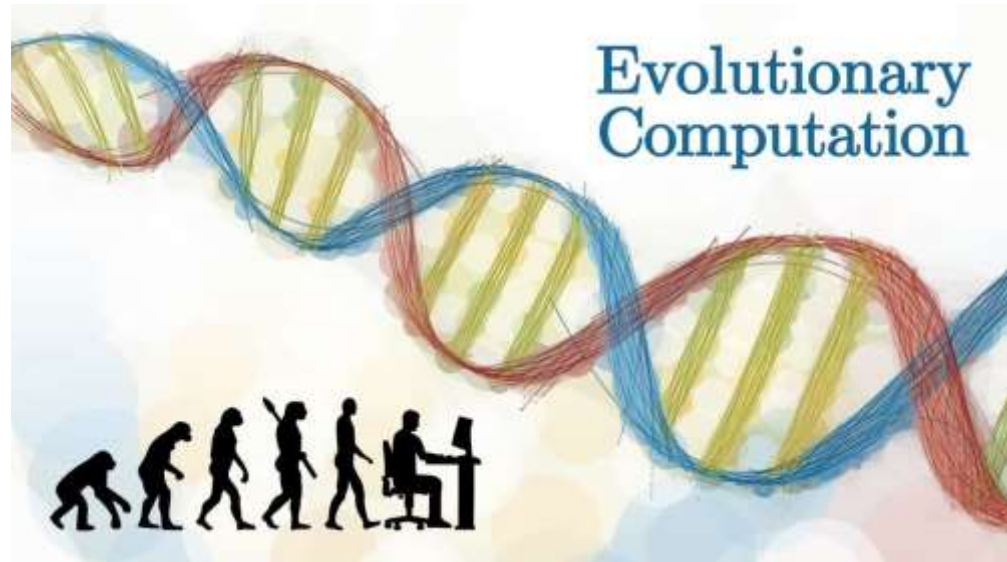
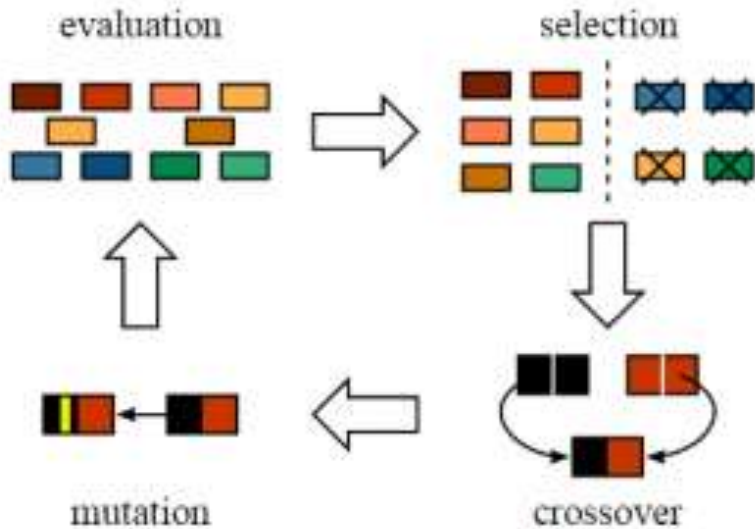


### ➤ SPT



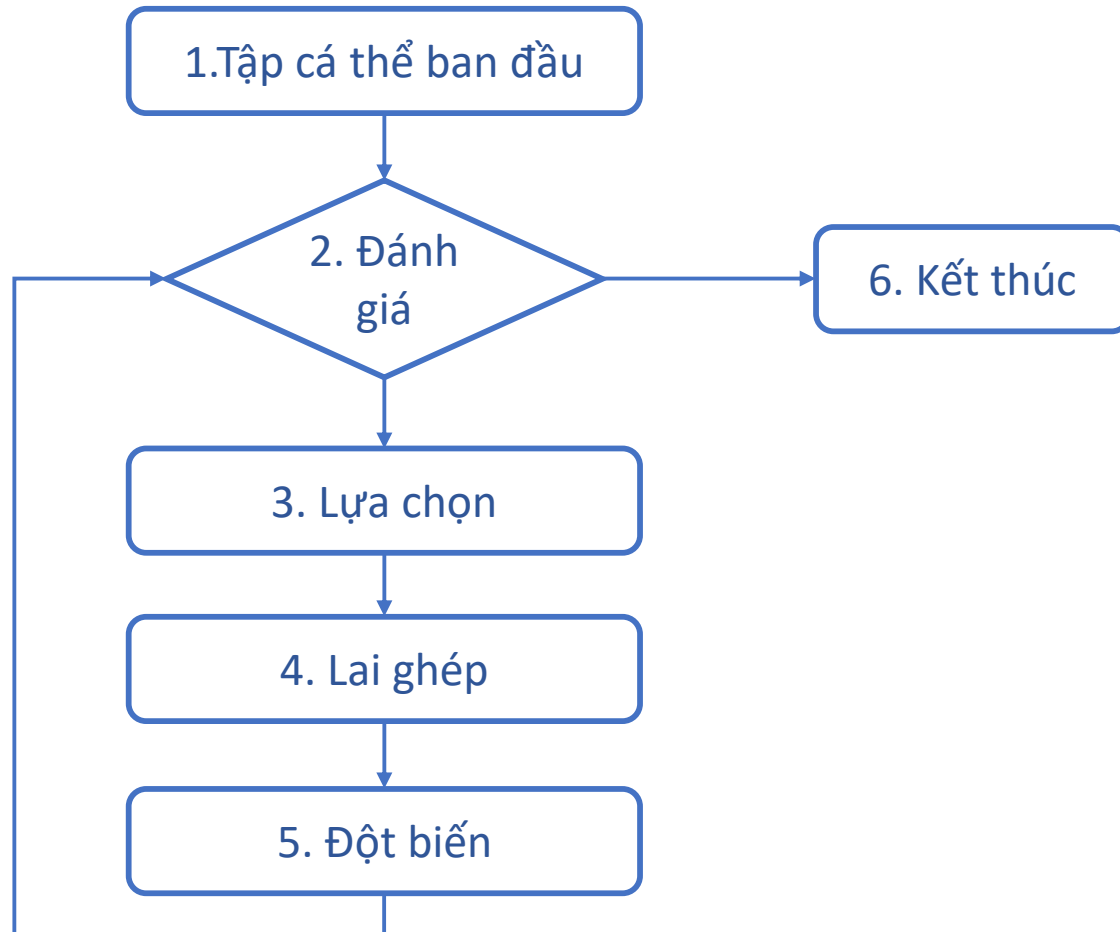
# 1. MÔ TẢ BÀI TOÁN

## Meta-Heuristics | Genetic Algorithm



## 2. THUẬT TOÁN DI TRUYỀN

### ➤ Sơ đồ thuật toán – Genetic Algorithms





### 3. VÍ DỤ

Cho ví dụ sau:

Có 3 công việc cần được xử lý trên 4 máy khác nhau. Mỗi công việc bao gồm một số các thao tác và mỗi thao tác có thể xử lý trên cả 4 máy. Thời gian xử lý trên các máy đã cho như bảng bên cạnh.

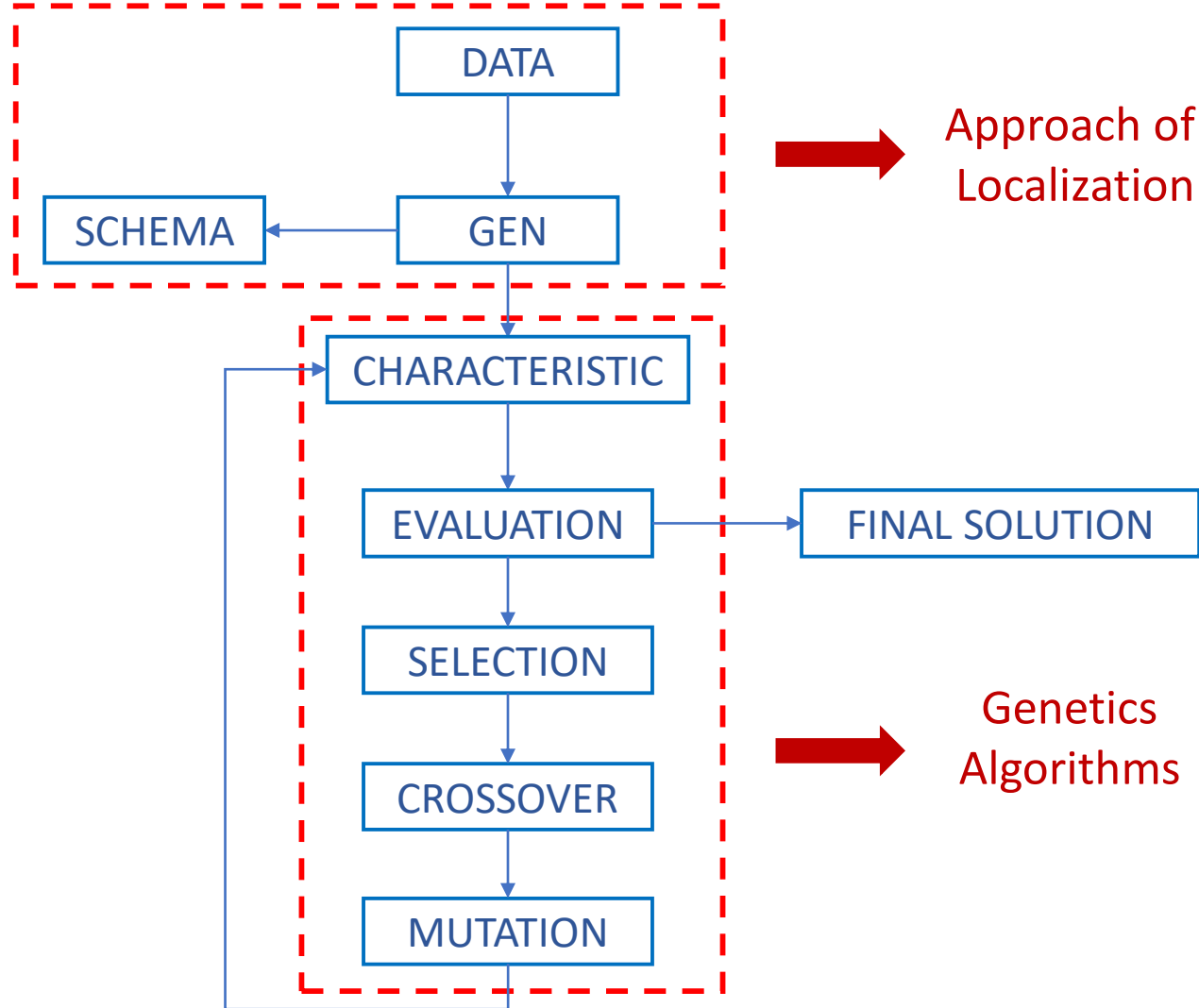
Hãy sắp xếp các thao tác trên các máy sao cho tổng thời gian hoàn thành tất cả các công việc – Makespans là nhỏ nhất?

TABLE I  
TABLE D

		M1	M2	M3	M4
J1	O 1,1	1	3	4	1
	O 2,1	3	8	2	1
	O 3,1	3	5	4	7
J2	O 1,2	4	1	1	4
	O 2,2	2	3	9	3
	O 3,2	9	1	2	2
J3	O 1,3	8	6	3	5
	O 2,3	4	5	8	1

*Imed Kacem, Slim Hammadi: "Approach by Localization and Multiobjective Evolutionary Optimization for Flexible Job-Shop Scheduling Problems", 2002*

## 4. CGAs



## Approach of Localization

Starting from a table D presenting the processing times possibilities on the various machines, create a new table D' whose size is the same one as the table D;  
 create a table S whose size is the same one as the table D (S is going to represent chosen assignments);  
 initialize all elements of S to 0 ( $S_{i,j,k} = 0$ );  
 recopy D in D';  
**FOR** ( $j=1; j \leq N$ )  
     **FOR** ( $i=1; i \leq n_j$ )  
         •  $Min = +\infty$ ;  
         •  $Position = 1$ ;  
         • **FOR** ( $k=1; k \leq M$ )  
             **IF** ( $d'_{i,j,k} < Min$ ) **Then** { $Min = d'_{i,j,k}$ ;  $Position = k$ ;}  
             **End IF**  
         **End FOR**  
         •  $S_{i,j,Position} = 1$  (assignment of  $O_{i,j}$  to the machine  $M_{Position}$ );  
         // updating of D':  
         • **FOR** ( $i'=i+1; i' \leq n_j$ )  
              $d'_{i',j,Position} = d'_{i,j,Position} + d_{i,j,Position}$ ;  
             **End FOR**  
         • **FOR** ( $j'=j+1; j' \leq N$ )  
             **FOR** ( $i'=1; i' \leq n_{j'}$ )  
                  $d'_{i',j',Position} = d'_{i',j,Position} + d_{i,j,Position}$ ;  
                 **End FOR**  
         **End FOR**  
     **End FOR**  
**End FOR**

TABLE I  
TABLE D

		M1	M2	M3	M4
J1	O1,1	1	3	4	1
	O2,1	3	8	2	1
	O3,1	3	5	4	7
J2	O1,2	4	1	1	4
	O2,2	2	3	9	3
	O3,2	9	1	2	2
J3	O1,3	8	6	3	5
	O2,3	4	5	8	1

## Approach of Localization

TABLE IV  
ASSIGNMENT S1

		M1	M2	M3	M4
J1	O 1,1	1	0	0	0
	O 2,1	0	0	0	1
	O 3,1	1	0	0	0
J2	O 1,2	0	1	0	0
	O 2,2	0	1	0	0
	O 3,2	0	0	1	0
J3	O 1,3	0	0	1	0
	O 2,3	0	0	0	1



TABLE XVII  
A SCHEDULE GIVEN BY THE AL

	Ope 1	Ope 2	Ope 3
J1	4, 0, 1	4, 1, 2	1, 3, 6
J2	2, 0, 1	1, 1, 3	2, 3, 4
J3	3, 0, 3	4, 3, 4	*****

Machines workloads ( $W_k$ ):  $\{W_1 = 5, W_2 = 2, W_3 = 3, W_4 = 3\}$ .

The sum of workloads of machines  $W = \sum W_k = 13$ .

The workload of the most loaded machine =  $\text{Max}(W_k) = 5$ .

The makespan =  $C_{\max} = 6$ .

### Beginning Scheduling Algorithm

initialize the vector of machines availabilities  $\text{Dispo\_Machine}[k]=0$  for each machine  $M_k$  ( $k \leq M$ );  
initialize the vector of jobs availabilities  $\text{Dispo\_Job}[j]=0$  for each job  $j$  ( $j \leq N$ );

**FOR** ( $i=1, i \leq \text{Max}_j(n_j)$ )

- construct the set  $E_i$  of operations to schedule from  $S$ :

$E_i = \{O_{i,j} / S_{i,j,k}=1, 1 \leq j \leq N\}$ ;

- classify the operations of  $E_i$  according to the chosen priority rule;

**FOR** ( $j=1 ; 1 \leq j \leq N$ )

- calculate starting times by following the same order given by the classification of  $E_i$  according to the formula:  
 $t_{i,j} = \text{Max}(\text{Dispo\_Machine}[k], \text{Dispo\_Job}[j])$  such that  $S_{i,j,k}=1$ ;
- updating of the vector of machine availabilities:  
 $\text{Dispo\_Machine}[k] = t_{i,j} + d_{i,j,k}$ ;
- updating of the vector of job availabilities:  
 $\text{Dispo\_Job}[j] = t_{i,j} + d_{i,j,k}$ ;

**End FOR**

**End FOR**

**End Scheduling Algorithm**

## Approach of Localization

TABLE I  
TABLE D

		M1	M2	M3	M4
J1	O1,1	1	3	4	1
	O2,1	3	8	2	1
	O3,1	3	5	4	7
J2	O1,2	4	1	1	4
	O2,2	2	3	9	3
	O3,2	9	1	2	2
J3	O1,3	8	6	3	5
	O2,3	4	5	8	1

TABLE II  
TABLE D' FOR  $j = 1$  AND  $i = 1$

		M1	M2	M3	M4
J1	O1,1	1	3	4	1
	O2,1	4	8	2	1
	O3,1	4	5	4	7
J2	O1,2	5	1	1	4
	O2,2	3	3	9	3
	O3,2	10	1	2	2
J3	O1,3	9	6	3	5
	O2,3	5	5	8	1

TABLE III  
TABLE D' FOR  $j = 1$  AND  $i = 2$

		M1	M2	M3	M4
J1	O1,1	1	3	4	1
	O2,1	4	8	2	1
	O3,1	4	5	4	8
J2	O1,2	5	1	1	5
	O2,2	3	3	9	4
	O3,2	10	1	2	3
J3	O1,3	9	6	3	6
	O2,3	5	5	8	2

TABLE IV  
ASSIGNMENT S1

		M1	M2	M3	M4
J1	O1,1	1	0	0	0
	O2,1	0	0	0	1
	O3,1	1	0	0	0
J2	O1,2	0	1	0	0
	O2,2	0	1	0	0
	O3,2	0	0	1	0
J3	O1,3	0	0	1	0
	O2,3	0	0	0	0

$i = 1$

$i = 2$

## Crossover

Random A

Parent 1

		M1	M2	M3	M4
J1	O 1,1	0	0	0	1
	O 1,2	0	0	0	1
	O 1,3	1	0	0	0
J2	O 2,1	0	1	0	0
	O 2,2	1	0	0	0
	O 2,3	0	1	0	0
J3	O 3,1	0	0	1	0
	O 3,2	0	0	0	1

A

E1

		M1	M2	M3	M4
J1	O 1,1	1	0	0	0
	O 1,2	0	0	0	1
	O 1,3	1	0	0	0
J2	O 2,1	0	1	0	0
	O 2,2	1	0	0	0
	O 2,3	0	1	0	0
J3	O 3,1	0	0	1	0
	O 3,2	0	0	0	1

Parent 2

		M1	M2	M3	M4
J1	O 1,1	1	0	0	0
	O 1,2	0	0	0	1
	O 1,3	1	0	0	0
J2	O 2,1	0	0	1	0
	O 2,2	1	0	0	0
	O 2,3	0	1	0	0
J3	O 3,1	0	0	1	0
	O 3,2	0	0	0	1

A

E2

		M1	M2	M3	M4
J1	O 1,1	0	0	0	1
	O 1,2	0	0	0	1
	O 1,3	1	0	0	0
J2	O 2,1	0	0	1	0
	O 2,2	1	0	0	0
	O 2,3	0	1	0	0
J3	O 3,1	0	0	1	0
	O 3,2	0	0	0	1



## Mutation

TABLE I  
TABLE D

		M1	M2	M3	M4
J1	O 1,1	1	3	4	1
	O 2,1	3	8	2	1
	O 3,1	3	5	4	7
J2	O 1,2	4	1	1	4
	O 2,2	2	3	9	3
	O 3,2	9	1	2	2
J3	O 1,3	8	6	3	5
	O 2,3	4	5	8	1

		M1	M2	M3	M4
J1	O 1,1	0	0	0	0, 1
	O 2,1	0	0	3, 5	0
	O 3,1	5, 8	0	0	0
J2	O 1,2	0	0, 1	0	0
	O 2,2	0	1, 4	0	0
	O 3,2	0	0	0	4, 6
J3	O 1,3	0	0	0, 3	0
	O 2,3	0	0	0	3, 4

Mutation



		M1	M2	M3	M4
J1	O 1,1	0	0	0	0, 1
	O 2,1	0	0	0	1, 2
	O 3,1	2, 5	0	0	0
J2	O 1,2	0	0, 1	0	0
	O 2,2	0	1, 4	0	0
	O 3,2	0	0	0	4, 6
J3	O 1,3	0	0	0, 3	0
	O 2,3	0	0	0	3, 4

## 4. CGAs

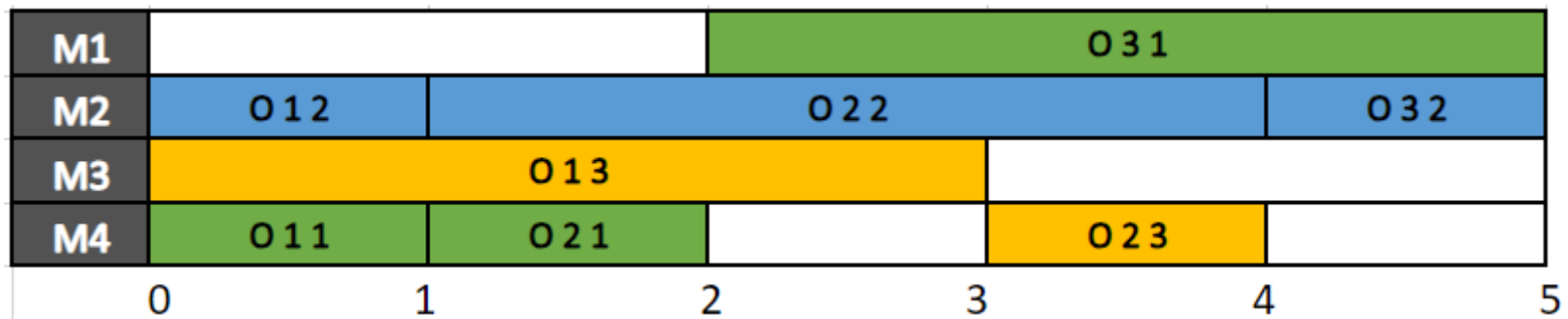
### Dữ liệu

TABLE I  
TABLE D

		M1	M2	M3	M4
J1	O1,1	1	3	4	1
	O2,1	3	8	2	1
	O3,1	3	5	4	7
J2	O1,2	4	1	1	4
	O2,2	2	3	9	3
	O3,2	9	1	2	2
J3	O1,3	8	6	3	5
	O2,3	4	5	8	1

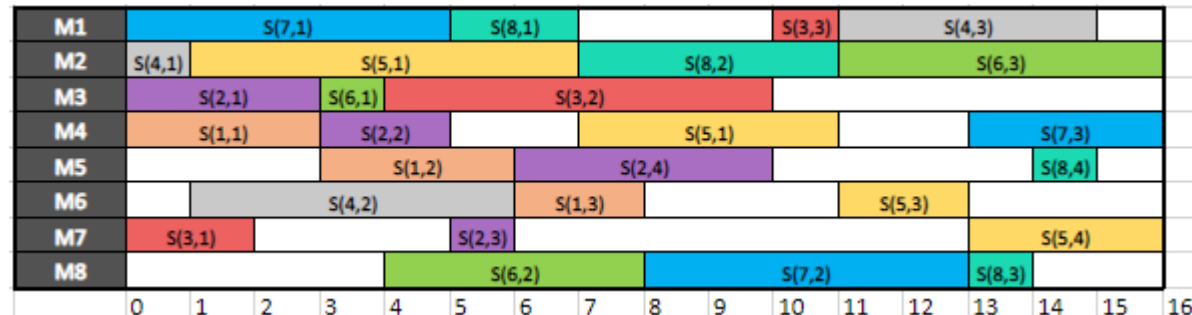
### Kết quả

```
J[1][1]: [0, 0, 0, (0, 1)]
J[1][2]: [0, 0, 0, (1, 2)]
J[1][3]: [(2, 5), 0, 0, 0]
J[2][1]: [0, (0, 1), 0, 0]
J[2][2]: [0, (1, 4), 0, 0]
J[2][3]: [0, (4, 5), 0, 0]
J[3][1]: [0, 0, (0, 3), 0]
J[3][2]: [0, 0, 0, (3, 4)]
J[3][3]: [0, 0, 0, 0]
```



# 4. CGAs

Job	Operation	M1	M2	M3	M4	M5	M6	M7	M8
1	S[1][1]	0	0	0	(0,3)	0	0	0	0
	S[1][2]	0	0	0	0	(3,6)	0	0	0
	S[1][3]	0	0	0	0	0	(6,8)	0	0
2	S[2][1]	0	0	(0,3)	0	0	0	0	0
	S[2][2]	0	0	0	(3,5)	0	0	0	0
	S[2][3]	0	0	0	0	0	0	(5,6)	0
	S[2][4]	0	0	0	0	(6,10)	0	0	0
3	S[3][1]	0	0	0	0	0	0	(0,2)	0
	S[3][2]	0	0	(4,10)	0	0	0	0	0
	S[3][3]	(10,11)	0	0	0	0	0	0	0
4	S[4][1]	0	(0,1)	0	0	0	0	0	0
	S[4][2]	0	0	0	0	0	(1,6)	0	0
	S[4][3]	(11,15)	0	0	0	0	0	0	0
5	S[5][1]	0	(1,7)	0	0	0	0	0	0
	S[5][2]	0	0	0	(7,11)	0	0	0	0
	S[5][3]	0	0	0	0	0	(11,13)	0	0
	S[5][4]	0	0	0	0	0	0	(13,16)	0
6	S[6][1]	0	0	(3,4)	0	0	0	0	0
	S[6][2]	0	0	0	0	0	0	0	(4,8)
	S[6][3]	0	(11,16)	0	0	0	0	0	0
7	S[7][1]	(0,5)	0	0	0	0	0	0	0
	S[7][2]	0	0	0	0	0	0	0	(8,13)
	S[7][3]	0	0	0	(13,16)	0	0	0	0
8	S[8][1]	(5,7)	0	0	0	0	0	0	0
	S[8][2]	0	(7,11)	0	0	0	0	0	0
	S[8][3]	0	0	0	0	0	0	0	(13,14)
	S[8][4]	0	0	0	0	(14,15)	0	0	0



		M1	M2	M3	M4	M5	M6	M7	M8
J1	O1,1	5	3	5	3	3	X	10	9
	O2,1	10	X	5	8	3	9	9	6
	O3,1	X	10	X	5	6	2	4	5
J2	O1,2	5	7	3	9	8	X	9	X
	O2,2	X	8	5	2	6	7	10	9
	O3,2	X	10	X	5	6	4	1	7
J3	O3,4	10	8	9	6	4	7	X	X
	O1,3	10	X	X	7	6	5	2	4
	O2,3	X	10	6	4	8	9	10	X
J4	O3,3	1	4	5	6	X	10	X	7
	O1,4	3	1	6	5	9	7	8	4
	O2,4	12	11	7	8	10	5	6	9
J5	O3,4	4	6	2	10	3	9	5	7
	O1,5	3	6	7	8	9	X	10	X
	O2,5	10	X	7	4	9	8	6	X
	O3,5	X	9	8	7	4	2	7	X
J6	O4,5	11	9	X	6	7	5	3	6
	O1,6	6	7	1	4	6	9	X	10
	O2,6	11	X	9	9	9	7	6	4
J7	O3,6	10	5	9	10	11	X	10	X
	O1,7	5	4	2	6	7	X	10	X
	O2,7	X	9	X	9	11	9	10	5
J8	O3,7	X	8	9	3	8	6	X	10
	O1,8	2	8	5	9	X	4	X	10
	O2,8	7	4	7	8	9	X	10	X
	O3,8	9	9	X	8	5	6	7	1
	O4,8	9	X	3	7	1	5	8	X

Time Complete Job:	[8, 10, 11, 15, 16, 16, 16, 15]
Workloads per Machine:	[12, 16, 10, 12, 8, 9, 6, 10]
Makespan-Cmax:	16
Total Workloads-Wmax:	83



# 5. ỨNG DỤNG OR-Tools

OR-Tools là một phần mềm mã nguồn mở nhằm *tối ưu hóa tổ hợp*.

Một số ví dụ:

- Định tuyến xe
- Lên lịch
- Đóng gói
- .....

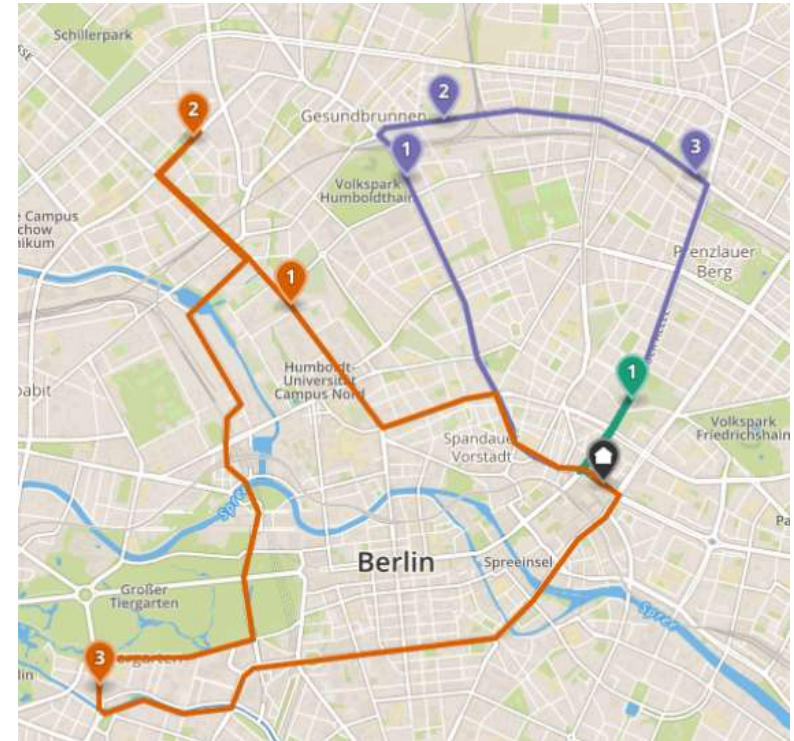


Google OR-Tools

# 5. ỨNG DỤNG OR-Tools

OR-Tools bao gồm các trình giải quyết cho:

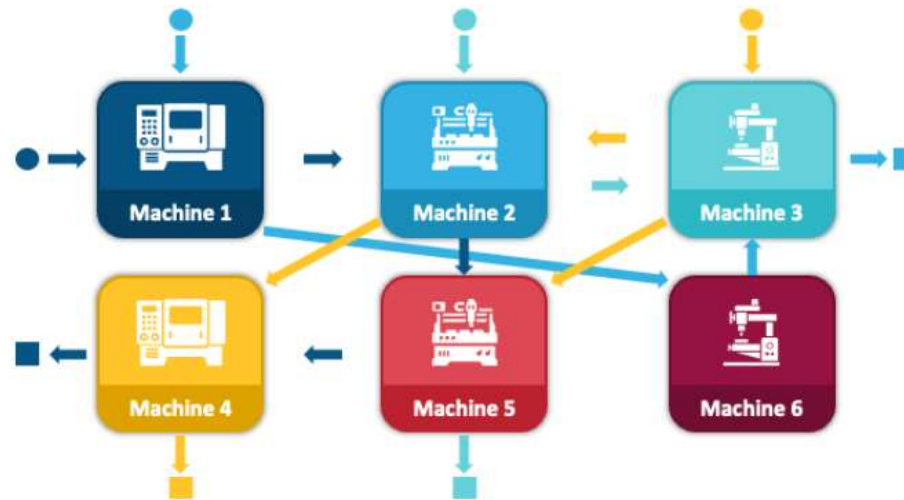
1. *Tối ưu hóa tuyến tính*
2. *Tối ưu hóa điều kiện ràng buộc*
3. *Tối ưu hóa hỗn hợp số nguyên*
4. *Chuyển nhượng*
5. *Lên lịch*
6. *Đóng gói*
7. *Định tuyến*
8. *Luồng mạng*



Để giải quyết một vấn đề tối ưu hóa thì bước đầu tiên đó là đi xác định *mục tiêu* và các *hạn chế, ràng buộc*.

# 5. ỨNG DỤNG OR-Tools

## 5.1\_OR – Tools trong bài toán lên lịch



**Bài toán lập lịch** là việc phân bổ nguồn tài nguyên cho tập hợp các nhiệm vụ trong thời điểm cụ thể. Một ví dụ quan trọng đó là **Job shop problem**, bao gồm nhiều công việc trên một số máy. Mỗi công việc bao gồm chuỗi nhiệm vụ và phải được xử lý tuần tự và mỗi nhiệm vụ phải được xử lý trên một máy cụ thể. Yêu cầu vấn đề là xây dựng một lịch trình để hoàn thành tất cả các công việc trong thời gian ngắn nhất có thể



## 5.1\_OR – Tools trong bài toán lên lịch





**HUST**

**THANK YOU !**