# 1. Data de Problème:

## 1. Tableau de temps de processing

P[i][S[j]]: matrice des temps de processing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P | J0 | J1 | J2 | J3 | J4 | J5 |
| M0 | 7.0 | 9.0 | 5.0 | 9.0 | 8.0 | 4.0 |
| M1 | 3.0 | 8.0 | 4.0 | 5.0 | 4.0 | 9.0 |
| M2 | 5.0 | 5.0 | 8.0 | 7.0 | 6.0 | 7.0 |

**Table 1: Temps de processing**

## 2. Tableau de temps de préparation

TP[i][S[j1]][S[j2]]: Tenseur des temps de préparation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP0 | J0 | J1 | J2 | J3 | J4 | J5 |
| J0 | 3.0 | 2.0 | 3.0 | 2.0 | 3.0 | 2.0 |
| J1 | 2.0 | 2.0 | 2.0 | 3.0 | 2.0 | 3.0 |
| J2 | 4.0 | 2.0 | 2.0 | 3.0 | 2.0 | 4.0 |
| J3 | 3.0 | 3.0 | 3.0 | 5.0 | 4.0 | 3.0 |
| J4 | 3.0 | 2.0 | 3.0 | 2.0 | 4.0 | 3.0 |
| J5 | 2.0 | 4.0 | 3.0 | 4.0 | 3.0 | 2.0 |

**Table 2Temps de préparation sur la machine 0**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP1 | J0 | J1 | J2 | J3 | J4 | J5 |
| J0 | 4.0 | 3.0 | 3.0 | 2.0 | 3.0 | 4.0 |
| J1 | 2.0 | 3.0 | 2.0 | 4.0 | 2.0 | 3.0 |
| J2 | 2.0 | 2.0 | 2.0 | 5.0 | 3.0 | 2.0 |
| J3 | 3.0 | 2.0 | 3.0 | 3.0 | 5.0 | 4.0 |
| J4 | 2.0 | 4.0 | 3.0 | 4.0 | 5.0 | 2.0 |
| J5 | 3.0 | 2.0 | 3.0 | 4.0 | 5.0 | 3.0 |

**Table 3Temps de préparation sur la machine 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP2 | J0 | J1 | J2 | J3 | J4 | J5 |
| J0 | 5.0 | 4.0 | 3.0 | 3.0 | 2.0 | 3.0 |
| J1 | 3.0 | 2.0 | 2.0 | 4.0 | 2.0 | 4.0 |
| J2 | 2.0 | 2.0 | 3.0 | 2.0 | 3.0 | 3.0 |
| J3 | 3.0 | 2.0 | 3.0 | 3.0 | 2.0 | 3.0 |
| J4 | 3.0 | 2.0 | 3.0 | 3.0 | 4.0 | 4.0 |
| J5 | 2.0 | 2.0 | 3.0 | 2.0 | 3.0 | 3.0 |

**Table 4Temps de préparation sur la machine 2**

## 3. Liste des délais de chaque Job

d[S[j]]: List des délais de chaque job

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | J0 | J1 | J2 | J3 | J4 | J5 |
| d | 20.0 | 12.0 | 55.0 | 20.0 | 32.0 | 12.0 |

**Table 5: Délais de chaque job**

# 2. PFSP (Permutation Flow Shop Scheduling Problem)

## 1. Séquence par algorithme de CDS

après le teste de tous les k possibles de k = 1,...,2on a trouvé que le meilleur k qui nous donne Cmax le plus optimale est k=2

P[i][S[j]]: matrice des temps de processing trouvée par CDS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P\_CDS | J0 | J1 | J2 | J3 | J4 | J5 |
| M0 | 10.0 | 17.0 | 9.0 | 14.0 | 12.0 | 13.0 |
| M1 | 8.0 | 13.0 | 12.0 | 12.0 | 10.0 | 16.0 |

**Table 6: Temps de processing des deux machines fictives pou k optimale**

S[j]: Séquence trouvée par l'algotrithme de Jonhson

U[j]: Jobs tel que P[0][S[j]]<P[1][S[j]]:[2, 5]

V[j]: Jobs tel que P[0][S[j]]>=P[1][S[j]]:[0, 1, 3, 4]

U\_SPT[j]: U[j] avec Jobs ordonancer par SPT(short processing Time), temps de processing croissant:[2, 5]

V\_LPT[j]: V[j] avec Jobs ordonancer par LPT(long processing Time), temps de processing décroissant:[1, 3, 4, 0]

Alors la séquence trouvée par Jonhson est la suivante

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | J2 | J5 | J1 | J3 | J4 | J0 |

**Table 7: Séquence trouvée par l'algotrithme de Jonhson**

## 2. Etapes de résolution de PFSP (Cmax/TFT...)

C[i][S[j]]: matrice des dates de fin de chaque job dans chaque machine

Sans préparation et sans aucune condition   
équation 1: pour i = 0 et j = 0  
 C[0][J2] = P[0][J2] = 5.0  
équation 2: pour i = 0,...,2 et j = 0  
 C[1][J2] = C[0][J2] + P[1][J2] = 5.0 + 4.0 = 9.0  
 C[2][J2] = C[1][J2] + P[2][J2] = 9.0 + 8.0 = 17.0  
équation 3: pour i = 0 et j = 0,...,5  
 C[0][J5] = C[0][J2] + P[0][J5] = 5.0 + 4.0 = 9.0  
 C[0][J1] = C[0][J5] + P[0][J1] = 9.0 + 9.0 = 18.0  
 C[0][J3] = C[0][J1] + P[0][J3] = 18.0 + 9.0 = 27.0  
 C[0][J4] = C[0][J3] + P[0][J4] = 27.0 + 8.0 = 35.0  
 C[0][J0] = C[0][J4] + P[0][J0] = 35.0 + 7.0 = 42.0  
équation 4: pour i = 1,...,2 et j = 1,...,5  
 C[1][J5] = max( C[0][J5] , C[1][J2] ) + P[1][J5] = max(9.0,9.0) 9.0 = 9.0+9.0 = 18.0  
 C[1][J1] = max( C[0][J1] , C[1][J5] ) + P[1][J1] = max(18.0,18.0) 8.0 = 18.0+8.0 = 26.0  
 C[1][J3] = max( C[0][J3] , C[1][J1] ) + P[1][J3] = max(27.0,26.0) 5.0 = 27.0+5.0 = 32.0  
 C[1][J4] = max( C[0][J4] , C[1][J3] ) + P[1][J4] = max(35.0,32.0) 4.0 = 35.0+4.0 = 39.0  
 C[1][J0] = max( C[0][J0] , C[1][J4] ) + P[1][J0] = max(42.0,39.0) 3.0 = 42.0+3.0 = 45.0  
 C[2][J5] = max( C[1][J5] , C[2][J2] ) + P[2][J5] = max(18.0,17.0) 7.0 = 18.0+7.0 = 25.0  
 C[2][J1] = max( C[1][J1] , C[2][J5] ) + P[2][J1] = max(26.0,25.0) 5.0 = 26.0+5.0 = 31.0  
 C[2][J3] = max( C[1][J3] , C[2][J1] ) + P[2][J3] = max(32.0,31.0) 7.0 = 32.0+7.0 = 39.0  
 C[2][J4] = max( C[1][J4] , C[2][J3] ) + P[2][J4] = max(39.0,39.0) 6.0 = 39.0+6.0 = 45.0  
 C[2][J0] = max( C[1][J0] , C[2][J4] ) + P[2][J0] = max(45.0,45.0) 5.0 = 45.0+5.0 = 50.0  
  
Total flow time TFT = C[2][J2] + C[2][J5] + C[2][J1] + C[2][J3] + C[2][J4] + C[2][J0] =   
17.0 + 25.0 + 31.0 + 39.0 + 45.0 + 50.0 = 207.0  
  
Tardiness T:  
 T[J2] = max( C[2][J2] - d[J2] , 0 ) = max( 17.0 - 55.0 , 0 ) = 0  
 T[J5] = max( C[2][J5] - d[J5] , 0 ) = max( 25.0 - 12.0 , 0 ) = 13.0  
 T[J1] = max( C[2][J1] - d[J1] , 0 ) = max( 31.0 - 12.0 , 0 ) = 19.0  
 T[J3] = max( C[2][J3] - d[J3] , 0 ) = max( 39.0 - 20.0 , 0 ) = 19.0  
 T[J4] = max( C[2][J4] - d[J4] , 0 ) = max( 45.0 - 32.0 , 0 ) = 13.0  
 T[J0] = max( C[2][J0] - d[J0] , 0 ) = max( 50.0 - 20.0 , 0 ) = 30.0  
Tardiness TT = ∑ T[S[j]] : j = 0,...,5  
 TT = 94.0

## 3. Matrice des dates des fins et des débuts de chaque job sur chaque machine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C | J2 | J5 | J1 | J3 | J4 | J0 |
| M0 | 5.0 | 9.0 | 18.0 | 27.0 | 35.0 | 42.0 |
| M1 | 9.0 | 18.0 | 26.0 | 32.0 | 39.0 | 45.0 |
| M2 | 17.0 | 25.0 | 31.0 | 39.0 | 45.0 | 50.0 |

**Table 8: dates de fin de chaque job dans chaque machine PFSP**

F[i][S[j]]:dates de début de chaque job dans chaque machine PFSP   
pour trouver les dates de débuts de chaque job: F[i][S[j]] = C[i][S[j]] - P[i][S[j]]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| F | J2 | J5 | J1 | J3 | J4 | J0 |
| M0 | 0.0 | 5.0 | 9.0 | 18.0 | 27.0 | 35.0 |
| M1 | 5.0 | 9.0 | 18.0 | 27.0 | 35.0 | 42.0 |
| M2 | 9.0 | 18.0 | 26.0 | 32.0 | 39.0 | 45.0 |

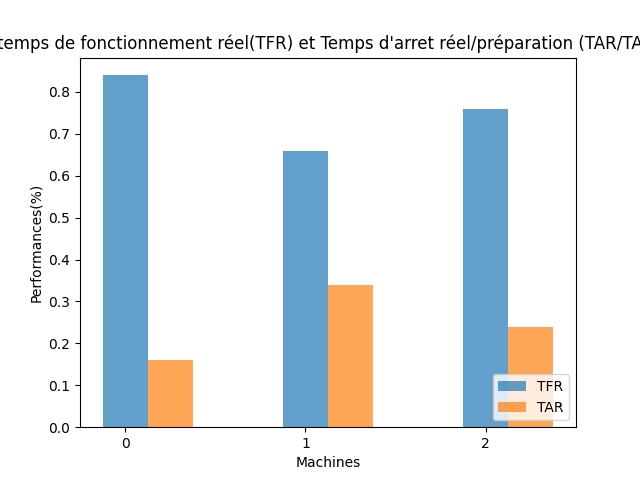
**Table 9: dates de début de chaque job dans chaque machine PFSP**

## 4. Performances des machines

TFR: (teaux de fonctionnement réel)  
TAR: (teaux d'arret réel)

|  |  |  |
| --- | --- | --- |
|  | TFR | TAR |
| M0 | 84.0% | 16.0% |
| M1 | 66.0% | 34.0% |
| M2 | 76.0% | 24.0% |

**Table 10: Pérformances des machines PFSP**



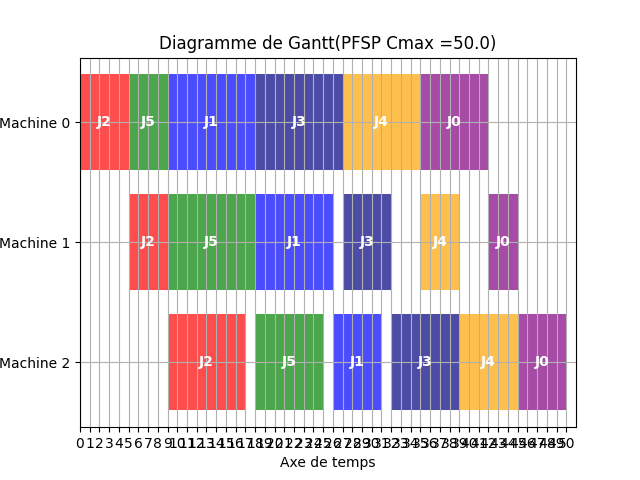
## 5. Temps d'attedre de chaque job entre chaque deux machines

TW: (waiting Time) temps d'attedre de chaque job entre chaque deux machines

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TW | J2 | J5 | J1 | J3 | J4 | J0 |
| M0/1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| M1/2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

**Table 11: Temps d'attendre de chaque job entre chaque deux machines PFSP**

## 6. Diagramme de Gantt,



# 3. PFSP-SDST (Permutation Flow Shop Scheduling Problem with Sequence Depending Setup Time)

## 1. Séquence par Test de tous les permutations possibles avec min(Cmax) et min(TT)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | J5 | J0 | J1 | J4 | J2 | J3 |

**Table 12: Séquence trouvée par l'algotrithme de Test de toutes les séquences**

## 2. Etapes de résolution de PFSP-SDST (Cmax/TFT...)

C[i][S[j]]: matrice des dates de fin de chaque job dans chaque machine   
M[i][S[j]][S[j]]: Tenseur des temps de préparation

Avec préparation et sans aucune condition   
équation 1: pour i = 0 et j = 0  
 C[0][J5] = P[0][J5] + M[0][J5][J5] = 4.0+2.0 = 6.0  
équation 2: pour i = 1,...,2 et j = 0  
 C[1][J5] = max( C[0][J5] , M[1][J5][J5] ) + P[1][J5] = max( 6.0 , 2.0) + 9.0 = 15.0  
 C[2][J5] = max( C[1][J5] , M[2][J5][J5] ) + P[2][J5] = max( 15.0 , 2.0) + 7.0 = 22.0  
équation 3: pour i = 0 et j = 1,...,5  
 C[0][J0] = C[0][J5] + P[0][J5] + M[0][J5][J0] = 6.0+7.0+2.0 = 15.0  
 C[0][J1] = C[0][J0] + P[0][J0] + M[0][J0][J1] = 15.0+9.0+2.0 = 26.0  
 C[0][J4] = C[0][J1] + P[0][J1] + M[0][J1][J4] = 26.0+8.0+2.0 = 36.0  
 C[0][J2] = C[0][J4] + P[0][J4] + M[0][J4][J2] = 36.0+5.0+3.0 = 44.0  
 C[0][J3] = C[0][J2] + P[0][J2] + M[0][J2][J3] = 44.0+9.0+3.0 = 56.0  
équation 4: pour i = 1,...,2j = 1,...,5  
 C[1][J0] = max( C[0][J0] , C[1][J5] + M[1][J5][J0] ) + P[1][J0] = max( 15.0 , 15.0+3.0) + 9.0 = 21.0  
 C[1][J1] = max( C[0][J1] , C[1][J0] + M[1][J0][J1] ) + P[1][J1] = max( 26.0 , 21.0+3.0) + 9.0 = 34.0  
 C[1][J4] = max( C[0][J4] , C[1][J1] + M[1][J1][J4] ) + P[1][J4] = max( 36.0 , 34.0+2.0) + 9.0 = 40.0  
 C[1][J2] = max( C[0][J2] , C[1][J4] + M[1][J4][J2] ) + P[1][J2] = max( 44.0 , 40.0+3.0) + 9.0 = 48.0  
 C[1][J3] = max( C[0][J3] , C[1][J2] + M[1][J2][J3] ) + P[1][J3] = max( 56.0 , 48.0+5.0) + 9.0 = 61.0  
 C[2][J0] = max( C[1][J0] , C[2][J5] + M[2][J5][J0] ) + P[2][J0] = max( 21.0 , 22.0+2.0) + 7.0 = 29.0  
 C[2][J1] = max( C[1][J1] , C[2][J0] + M[2][J0][J1] ) + P[2][J1] = max( 34.0 , 29.0+4.0) + 7.0 = 39.0  
 C[2][J4] = max( C[1][J4] , C[2][J1] + M[2][J1][J4] ) + P[2][J4] = max( 40.0 , 39.0+2.0) + 7.0 = 47.0  
 C[2][J2] = max( C[1][J2] , C[2][J4] + M[2][J4][J2] ) + P[2][J2] = max( 48.0 , 47.0+3.0) + 7.0 = 58.0  
 C[2][J3] = max( C[1][J3] , C[2][J2] + M[2][J2][J3] ) + P[2][J3] = max( 61.0 , 58.0+2.0) + 7.0 = 68.0  
  
Total flow time TFT = C[2][J5] + C[2][J0] + C[2][J1] + C[2][J4] + C[2][J2] + C[2][J3] =   
22.0 + 29.0 + 39.0 + 47.0 + 58.0 + 68.0 = 263.0  
  
Tardiness T:  
 T[J5] = max( C[2][J5] - d[J5] , 0 ) = max( 22.0 - 12.0 , 0 ) = 10.0  
 T[J0] = max( C[2][J0] - d[J0] , 0 ) = max( 29.0 - 20.0 , 0 ) = 9.0  
 T[J1] = max( C[2][J1] - d[J1] , 0 ) = max( 39.0 - 12.0 , 0 ) = 27.0  
 T[J4] = max( C[2][J4] - d[J4] , 0 ) = max( 47.0 - 32.0 , 0 ) = 15.0  
 T[J2] = max( C[2][J2] - d[J2] , 0 ) = max( 58.0 - 55.0 , 0 ) = 3.0  
 T[J3] = max( C[2][J3] - d[J3] , 0 ) = max( 68.0 - 20.0 , 0 ) = 48.0  
Tardiness TT = ∑ T[S[j]] : j = 0,...,5  
 TT = 112.0

## 3. Matrice des dates des fins et des débuts de chaque job sur chaque machine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C | J5 | J0 | J1 | J4 | J2 | J3 |
| M0 | 6.0 | 15.0 | 26.0 | 36.0 | 44.0 | 56.0 |
| M1 | 15.0 | 21.0 | 34.0 | 40.0 | 48.0 | 61.0 |
| M2 | 22.0 | 29.0 | 39.0 | 47.0 | 58.0 | 68.0 |

**Table 13: dates de fin de chaque job dans chaque machine PFSP-SDST**

F[i][S[j]]:dates de début de chaque job dans chaque machine PFSP-SDST   
pour trouver les dates de débuts de chaque job: F[i][S[j]] = C[i][S[j]] - P[i][S[j]]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| F | J5 | J0 | J1 | J4 | J2 | J3 |
| M0 | 2.0 | 8.0 | 17.0 | 28.0 | 39.0 | 47.0 |
| M1 | 6.0 | 18.0 | 26.0 | 36.0 | 44.0 | 56.0 |
| M2 | 15.0 | 24.0 | 34.0 | 41.0 | 50.0 | 61.0 |

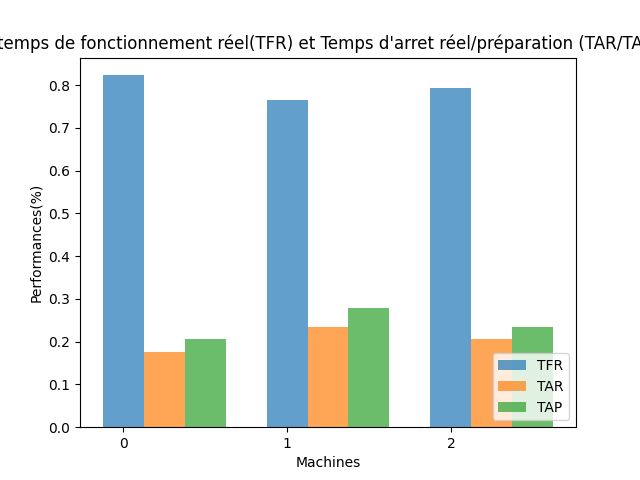
**Table 14: dates de début de chaque job dans chaque machine PFSP-SDST**

## 4. Performances des machines (machines non arretées en préparation)

TFR: (teaux de fonctionnement réel)  
TAR: (teaux d'arret réel)  
TAP: (teaux d'arret de préparation)

|  |  |  |  |
| --- | --- | --- | --- |
|  | TFR | TAR | TAP |
| M0 | 82.35% | 17.65% | 20.59% |
| M1 | 76.47% | 23.53% | 27.94% |
| M2 | 79.41% | 20.59% | 23.53% |

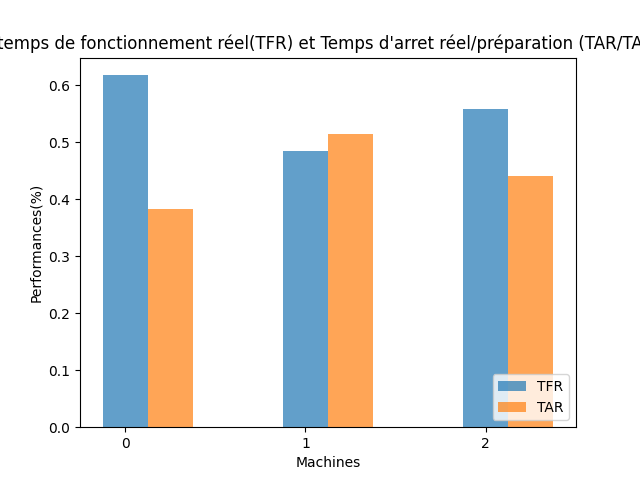
**Table 15: Pérformances des machines PFSP-SDST (machines non arretées en préparation)**



## 5. Performances des machines (machines arretées en préparation)

|  |  |  |
| --- | --- | --- |
|  | TFR | TAR |
| M0 | 82.35% | 17.65% |
| M1 | 76.47% | 23.53% |
| M2 | 79.41% | 20.59% |

**Table 16: Pérformances des machines PFSP-SDST (machines arretées en préparation)**



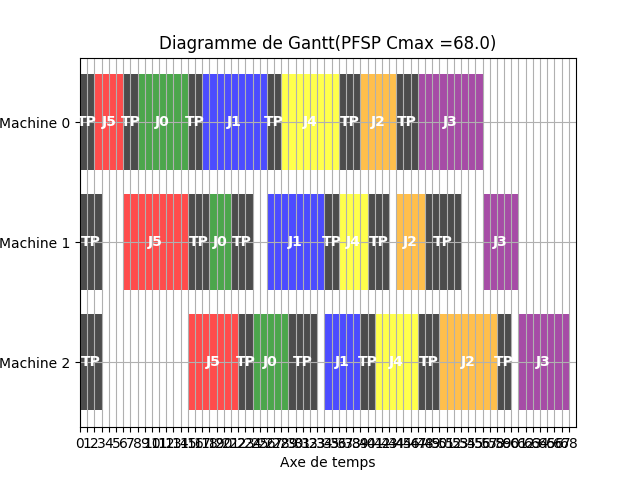
## 6. Temps d'attedre de chaque job entre chaque deux machines

TW: (waiting Time) temps d'attedre de chaque job entre chaque deux machines

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TW | J5 | J0 | J1 | J4 | J2 | J3 |
| M0/1 | 0.0 | 3.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| M1/2 | 0.0 | 3.0 | 0.0 | 1.0 | 2.0 | 0.0 |

**Table 17: Temps d'attendre de chaque job entre chaque deux machines PFSP-SDST**

## 7. Diagramme de Gantt,



# 4. BFSP (Blocking Flow Shop Scheduling Problem)

## 1. Séquence par Test de tous les permutations possibles avec min(Cmax) et min(TT)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | J5 | J1 | J3 | J4 | J0 | J2 |

**Table 18: Séquence trouvée par l'algotrithme de Test de toutes les séquences**

## 2. Etapes de résolution de BFSP (Cmax/TFT...)

D[i][S[j]]: matrice des dates de fin de chaque job avec son blocage dans chaque machine, aussi début des jobs dans la machine précédente   
pour la matrice D l'indexation des machines réel est de 1 à 3

Sans préparation et avec blocage   
 on Ajout une machine vertuel pour calculer la matrice D  
 indéxation des machines réeel de 1 à3  
équation 1: pour i = 0 et j = 0  
 D[0][J5] = 0  
équation 2: pour i = 1,...,2 et j = 0  
 D[1][J5] = D[0][J5] + P[0][J5] = 0.0 + 4.0 = 4.0  
 D[2][J5] = D[1][J5] + P[1][J5] = 4.0 + 9.0 = 13.0  
équation 5: pour i = 2 et j = 0  
 D[3][J5] = D[2][J5] + P[2][J5] = 13.0 + 7.0 = 20.0  
équation 3: pour i = 0 et j = 1  
 D[0][J1] = D[1][J5] = 4.0  
équation 4: pour i = 1 et j = 1  
 D[1][J1] = max( D[0][J1] + P[0][J13.0  
équation 4: pour i = 2 et j = 1  
 D[2][J1] = max( D[1][J1] + P[1][J21.0  
équation 5: pour i = 2 et j = 1  
 D[3][J1] = D[2][J1] + P[2][J1] = 21.0 + 5.0 = 26.0  
équation 3: pour i = 0 et j = 2  
 D[0][J3] = D[1][J1] = 13.0  
équation 4: pour i = 1 et j = 2  
 D[1][J3] = max( D[0][J3] + P[0][J22.0  
équation 4: pour i = 2 et j = 2  
 D[2][J3] = max( D[1][J3] + P[1][J27.0  
équation 5: pour i = 2 et j = 2  
 D[3][J3] = D[2][J3] + P[2][J3] = 27.0 + 7.0 = 34.0  
équation 3: pour i = 0 et j = 3  
 D[0][J4] = D[1][J3] = 22.0  
équation 4: pour i = 1 et j = 3  
 D[1][J4] = max( D[0][J4] + P[0][J30.0  
équation 4: pour i = 2 et j = 3  
 D[2][J4] = max( D[1][J4] + P[1][J34.0  
équation 5: pour i = 2 et j = 3  
 D[3][J4] = D[2][J4] + P[2][J4] = 34.0 + 6.0 = 40.0  
équation 3: pour i = 0 et j = 4  
 D[0][J0] = D[1][J4] = 30.0  
équation 4: pour i = 1 et j = 4  
 D[1][J0] = max( D[0][J0] + P[0][J37.0  
équation 4: pour i = 2 et j = 4  
 D[2][J0] = max( D[1][J0] + P[1][J40.0  
équation 5: pour i = 2 et j = 4  
 D[3][J0] = D[2][J0] + P[2][J0] = 40.0 + 5.0 = 45.0  
équation 3: pour i = 0 et j = 5  
 D[0][J2] = D[1][J0] = 37.0  
équation 4: pour i = 1 et j = 5  
 D[1][J2] = max( D[0][J2] + P[0][J42.0  
équation 4: pour i = 2 et j = 5  
 D[2][J2] = max( D[1][J2] + P[1][J46.0  
équation 5: pour i = 2 et j = 5  
 D[3][J2] = D[2][J2] + P[2][J2] = 46.0 + 8.0 = 54.0  
Calcul des temps de blocage  
 TB[0][J5] = D[1][J5] - D[0][J5] - P[0][J5] = 4.0 - 0.0 - 4.0 = 0.0  
 TB[0][J1] = D[1][J1] - D[0][J1] - P[0][J1] = 13.0 - 4.0 - 9.0 = 0.0  
 TB[0][J3] = D[1][J3] - D[0][J3] - P[0][J3] = 22.0 - 13.0 - 9.0 = 0.0  
 TB[0][J4] = D[1][J4] - D[0][J4] - P[0][J4] = 30.0 - 22.0 - 8.0 = 0.0  
 TB[0][J0] = D[1][J0] - D[0][J0] - P[0][J0] = 37.0 - 30.0 - 7.0 = 0.0  
 TB[0][J2] = D[1][J2] - D[0][J2] - P[0][J2] = 42.0 - 37.0 - 5.0 = 0.0  
 TB[1][J5] = D[2][J5] - D[1][J5] - P[1][J5] = 13.0 - 4.0 - 9.0 = 0.0  
 TB[1][J1] = D[2][J1] - D[1][J1] - P[1][J1] = 21.0 - 13.0 - 8.0 = 0.0  
 TB[1][J3] = D[2][J3] - D[1][J3] - P[1][J3] = 27.0 - 22.0 - 5.0 = 0.0  
 TB[1][J4] = D[2][J4] - D[1][J4] - P[1][J4] = 34.0 - 30.0 - 4.0 = 0.0  
 TB[1][J0] = D[2][J0] - D[1][J0] - P[1][J0] = 40.0 - 37.0 - 3.0 = 0.0  
 TB[1][J2] = D[2][J2] - D[1][J2] - P[1][J2] = 46.0 - 42.0 - 4.0 = 0.0  
 TB[2][J5] = D[3][J5] - D[2][J5] - P[2][J5] = 20.0 - 13.0 - 7.0 = 0.0  
 TB[2][J1] = D[3][J1] - D[2][J1] - P[2][J1] = 26.0 - 21.0 - 5.0 = 0.0  
 TB[2][J3] = D[3][J3] - D[2][J3] - P[2][J3] = 34.0 - 27.0 - 7.0 = 0.0  
 TB[2][J4] = D[3][J4] - D[2][J4] - P[2][J4] = 40.0 - 34.0 - 6.0 = 0.0  
 TB[2][J0] = D[3][J0] - D[2][J0] - P[2][J0] = 45.0 - 40.0 - 5.0 = 0.0  
 TB[2][J2] = D[3][J2] - D[2][J2] - P[2][J2] = 54.0 - 46.0 - 8.0 = 0.0  
  
Total flow time TFT = D[3][J5] + D[3][J1] + D[3][J3] + D[3][J4] + D[3][J0] + D[3][J2] =   
20.0 + 26.0 + 34.0 + 40.0 + 45.0 + 54.0 = 219.0  
  
Tardiness T:  
 T[J5] = max( D[3][J5] - d[J5] , 0 ) = max( 20.0 - 12.0 , 0 ) = 8.0  
 T[J1] = max( D[3][J1] - d[J1] , 0 ) = max( 26.0 - 12.0 , 0 ) = 14.0  
 T[J3] = max( D[3][J3] - d[J3] , 0 ) = max( 34.0 - 20.0 , 0 ) = 14.0  
 T[J4] = max( D[3][J4] - d[J4] , 0 ) = max( 40.0 - 32.0 , 0 ) = 8.0  
 T[J0] = max( D[3][J0] - d[J0] , 0 ) = max( 45.0 - 20.0 , 0 ) = 25.0  
 T[J2] = max( D[3][J2] - d[J2] , 0 ) = max( 54.0 - 55.0 , 0 ) = 0  
Tardiness TT = ∑ T[S[j]] : j = 0,...,5  
 TT = 69.0

## 3. Matrices D,C,F

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| D | J5 | J1 | J3 | J4 | J0 | J2 |
| M0 | 0.0 | 4.0 | 13.0 | 22.0 | 30.0 | 37.0 |
| M1 | 4.0 | 13.0 | 22.0 | 30.0 | 37.0 | 42.0 |
| M2 | 13.0 | 21.0 | 27.0 | 34.0 | 40.0 | 46.0 |
| M3 | 20.0 | 26.0 | 34.0 | 40.0 | 45.0 | 54.0 |

**Table 19: dates de fin de chaque job avec son blocage dans chaque machine BFSP**

C[i][S[j]]:dates de fin de chaque job dans chaque machine BFSP

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C | J5 | J1 | J3 | J4 | J0 | J2 |
| M0 | 4.0 | 13.0 | 22.0 | 30.0 | 37.0 | 42.0 |
| M1 | 13.0 | 21.0 | 27.0 | 34.0 | 40.0 | 46.0 |
| M2 | 20.0 | 26.0 | 34.0 | 40.0 | 45.0 | 54.0 |

**Table 20: dates de fin de chaque job dans chaque machine BFSP**

F[i][S[j]]:dates de début de chaque job dans chaque machine BFSP   
pour trouver les dates de débuts de chaque job: F[i][S[j]] = C[i][S[j]] - P[i][S[j]]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| F | J5 | J1 | J3 | J4 | J0 | J2 |
| M0 | 0.0 | 4.0 | 13.0 | 22.0 | 30.0 | 37.0 |
| M1 | 4.0 | 13.0 | 22.0 | 30.0 | 37.0 | 42.0 |
| M2 | 13.0 | 21.0 | 27.0 | 34.0 | 40.0 | 46.0 |

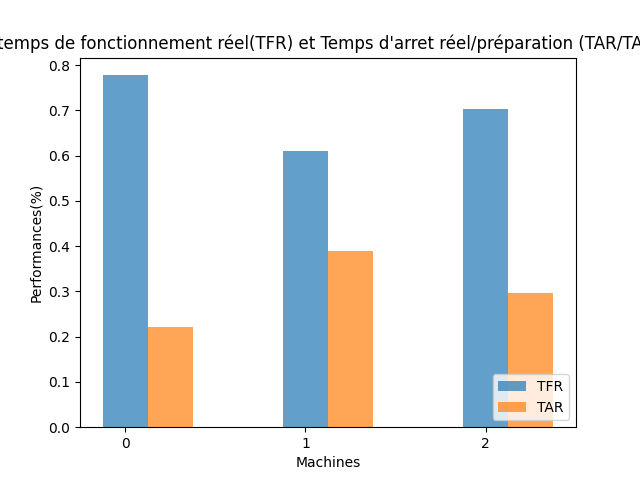
**Table 21: dates de début de chaque job dans chaque machine BFSP**

## 4. Performances des machines

TFR: (teaux de fonctionnement réel)  
TAR: (teaux d'arret réel)

|  |  |  |
| --- | --- | --- |
|  | TFR | TAR |
| M0 | 77.78% | 22.22% |
| M1 | 61.11% | 38.89% |
| M2 | 70.37% | 29.63% |

**Table 22: Pérformances des machines BFSP**



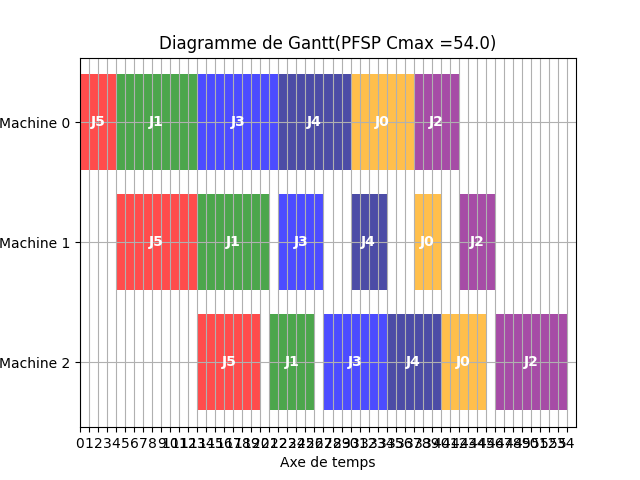
## 5. Temps de blocage de chaque job dans chaque machines

TB: (Blocking Time) temps de blocage de chaque job dans chaque machines

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TB | J5 | J1 | J3 | J4 | J0 | J2 |
| M0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| M1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| M2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

**Table 23: Temps de blocage de chaque job dans chaque machines BFSP**

## 6. Diagramme de Gantt,



# 5. BFSP-SDST (Blocking Flow Shop Scheduling Problem with Sequence Depending Setup Time)

## 1. Séquence par Test de tous les permutations possibles avec min(Cmax) et min(TT)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | J5 | J1 | J4 | J0 | J2 | J3 |

**Table 24: Séquence trouvée par l'algotrithme de Test de toutes les séquences**

## 2. Etapes de résolution de BFSP-SDST (Cmax/TFT...)

D[i][S[j]]: matrice des dates de fin de chaque job avec son blocage dans chaque machine, aussi début des jobs dans la machine précédente   
pour la matrice D l'indexation des machines réel est de 1 à 3 et l'indexation des jobs réel de 1 à 6

Avec préparation et avec blocage   
 on Ajout une machine vertuel et un job virtuel pour calculer la matrice D  
 indéxation des machines réeel de 1 à 3 ; et des jobs de 1 à 6  
équation 1: pour i = 0,...,3 et j = 0  
 D[i][J0] = 0  
équation 2: pour i = 0 et j = 1  
 D[0][J6] = D[1][J0] + M[1][J6][J6] = 0.0 + 2.0 = 2.0  
équation 4: pour i = 1 et j = 1  
 D[1][J6] = max( D[0][J6] + P[1][J6] , D[2][J0] + M[1][J6][J6] ) = max( 2.0 + 4.0 , 0.0 + 3.0 ) = 6.0  
équation 4: pour i = 2 et j = 1  
 D[2][J6] = max( D[1][J6] + P[2][J6] , D[3][J0] + M[2][J6][J6] ) = max( 6.0 + 9.0 , 0.0 + 3.0 ) = 15.0  
équation 6: pour i = 3 et j = 1  
 D[3][J6] = D[2][J6] + P[3][J6] = 15.0 + 7.0 = 22.0  
équation 3: pour i = 0 et j = 2  
 D[0][J2] = D[1][J6] + M[1][J6][J2] = 6.0 + 4.0 = 10.0  
équation 5: pour i = 1 et j = 2  
 D[1][J2] = max( D[0][J2] + P[1][J2] , D[2][6] + M[1][6][2] ) = max( 10.0 + 9.0 , 15.0 + 2.0 ) = 19.0équation 5: pour i = 2 et j = 2  
 D[2][J2] = max( D[1][J2] + P[2][J2] , D[3][6] + M[2][6][2] ) = max( 19.0 + 8.0 , 22.0 + 2.0 ) = 27.0équation 6: pour i = 3 et j = 2  
 D[3][J2] = D[2][J2] + P[3][J2] = 27.0 + 5.0 = 32.0  
équation 3: pour i = 0 et j = 3  
 D[0][J5] = D[1][J2] + M[1][J2][J5] = 19.0 + 2.0 = 21.0  
équation 5: pour i = 1 et j = 3  
 D[1][J5] = max( D[0][J5] + P[1][J5] , D[2][2] + M[1][2][5] ) = max( 21.0 + 8.0 , 27.0 + 2.0 ) = 29.0équation 5: pour i = 2 et j = 3  
 D[2][J5] = max( D[1][J5] + P[2][J5] , D[3][2] + M[2][2][5] ) = max( 29.0 + 4.0 , 32.0 + 2.0 ) = 34.0équation 6: pour i = 3 et j = 3  
 D[3][J5] = D[2][J5] + P[3][J5] = 34.0 + 6.0 = 40.0  
équation 3: pour i = 0 et j = 4  
 D[0][J1] = D[1][J5] + M[1][J5][J1] = 29.0 + 3.0 = 32.0  
équation 5: pour i = 1 et j = 4  
 D[1][J1] = max( D[0][J1] + P[1][J1] , D[2][5] + M[1][5][1] ) = max( 32.0 + 7.0 , 34.0 + 2.0 ) = 39.0équation 5: pour i = 2 et j = 4  
 D[2][J1] = max( D[1][J1] + P[2][J1] , D[3][5] + M[2][5][1] ) = max( 39.0 + 3.0 , 40.0 + 3.0 ) = 43.0équation 6: pour i = 3 et j = 4  
 D[3][J1] = D[2][J1] + P[3][J1] = 43.0 + 5.0 = 48.0  
équation 3: pour i = 0 et j = 5  
 D[0][J3] = D[1][J1] + M[1][J1][J3] = 39.0 + 3.0 = 42.0  
équation 5: pour i = 1 et j = 5  
 D[1][J3] = max( D[0][J3] + P[1][J3] , D[2][1] + M[1][1][3] ) = max( 42.0 + 5.0 , 43.0 + 3.0 ) = 47.0équation 5: pour i = 2 et j = 5  
 D[2][J3] = max( D[1][J3] + P[2][J3] , D[3][1] + M[2][1][3] ) = max( 47.0 + 4.0 , 48.0 + 3.0 ) = 51.0équation 6: pour i = 3 et j = 5  
 D[3][J3] = D[2][J3] + P[3][J3] = 51.0 + 8.0 = 59.0  
équation 3: pour i = 0 et j = 6  
 D[0][J4] = D[1][J3] + M[1][J3][J4] = 47.0 + 3.0 = 50.0  
équation 5: pour i = 1 et j = 6  
 D[1][J4] = max( D[0][J4] + P[1][J4] , D[2][3] + M[1][3][4] ) = max( 50.0 + 9.0 , 51.0 + 5.0 ) = 59.0équation 5: pour i = 2 et j = 6  
 D[2][J4] = max( D[1][J4] + P[2][J4] , D[3][3] + M[2][3][4] ) = max( 59.0 + 5.0 , 59.0 + 2.0 ) = 64.0équation 6: pour i = 3 et j = 6  
 D[3][J4] = D[2][J4] + P[3][J4] = 64.0 + 7.0 = 71.0  
 Indexation normale pour TB et C  
Calcul de la matrice C  
 C[0][J5] = D[1][J0] + P[0][J5] = 2.0 + 4.0 = 6.0  
 C[0][J1] = D[1][J6] + P[0][J1] = 10.0 + 9.0 = 19.0  
 C[0][J4] = D[1][J2] + P[0][J4] = 21.0 + 8.0 = 29.0  
 C[0][J0] = D[1][J5] + P[0][J0] = 32.0 + 7.0 = 39.0  
 C[0][J2] = D[1][J1] + P[0][J2] = 42.0 + 5.0 = 47.0  
 C[0][J3] = D[1][J3] + P[0][J3] = 50.0 + 9.0 = 59.0  
 C[1][J5] = D[2][J0] + P[1][J5] = 6.0 + 9.0 = 15.0  
 C[1][J1] = D[2][J6] + P[1][J1] = 19.0 + 8.0 = 27.0  
 C[1][J4] = D[2][J2] + P[1][J4] = 29.0 + 4.0 = 33.0  
 C[1][J0] = D[2][J5] + P[1][J0] = 39.0 + 3.0 = 42.0  
 C[1][J2] = D[2][J1] + P[1][J2] = 47.0 + 4.0 = 51.0  
 C[1][J3] = D[2][J3] + P[1][J3] = 59.0 + 5.0 = 64.0  
 C[2][J5] = D[3][J0] + P[2][J5] = 15.0 + 7.0 = 22.0  
 C[2][J1] = D[3][J6] + P[2][J1] = 27.0 + 5.0 = 32.0  
 C[2][J4] = D[3][J2] + P[2][J4] = 34.0 + 6.0 = 40.0  
 C[2][J0] = D[3][J5] + P[2][J0] = 43.0 + 5.0 = 48.0  
 C[2][J2] = D[3][J1] + P[2][J2] = 51.0 + 8.0 = 59.0  
 C[2][J3] = D[3][J3] + P[2][J3] = 64.0 + 7.0 = 71.0  
Calcul des temps de blocage  
 TB[0][J5] = D[1][J0] - C[0][J5] = 6.0 - 6.0 = 0.0  
 TB[0][J1] = D[1][J6] - C[0][J1] = 19.0 - 19.0 = 0.0  
 TB[0][J4] = D[1][J2] - C[0][J4] = 29.0 - 29.0 = 0.0  
 TB[0][J0] = D[1][J5] - C[0][J0] = 39.0 - 39.0 = 0.0  
 TB[0][J2] = D[1][J1] - C[0][J2] = 47.0 - 47.0 = 0.0  
 TB[0][J3] = D[1][J3] - C[0][J3] = 59.0 - 59.0 = 0.0  
 TB[1][J5] = D[2][J0] - C[1][J5] = 15.0 - 15.0 = 0.0  
 TB[1][J1] = D[2][J6] - C[1][J1] = 27.0 - 27.0 = 0.0  
 TB[1][J4] = D[2][J2] - C[1][J4] = 34.0 - 33.0 = 1.0  
 TB[1][J0] = D[2][J5] - C[1][J0] = 43.0 - 42.0 = 1.0  
 TB[1][J2] = D[2][J1] - C[1][J2] = 51.0 - 51.0 = 0.0  
 TB[1][J3] = D[2][J3] - C[1][J3] = 64.0 - 64.0 = 0.0  
 TB[2][J5] = D[3][J0] - C[2][J5] = 22.0 - 22.0 = 0.0  
 TB[2][J1] = D[3][J6] - C[2][J1] = 32.0 - 32.0 = 0.0  
 TB[2][J4] = D[3][J2] - C[2][J4] = 40.0 - 40.0 = 0.0  
 TB[2][J0] = D[3][J5] - C[2][J0] = 48.0 - 48.0 = 0.0  
 TB[2][J2] = D[3][J1] - C[2][J2] = 59.0 - 59.0 = 0.0  
 TB[2][J3] = D[3][J3] - C[2][J3] = 71.0 - 71.0 = 0.0  
  
Total flow time TFT = D[4][J0] + D[4][J6] + D[4][J2] + D[4][J5] + D[4][J1] + D[4][J3] + D[4][J4] =   
0.0 + 22.0 + 32.0 + 40.0 + 48.0 + 59.0 + 71.0 = 272.0  
  
Tardiness T:  
 T[J0] = max( D[4][J0] - d[J-1] , 0 ) = max( 0.0 - 12.0 , 0 ) = 0  
 T[J6] = max( D[4][J6] - d[J5] , 0 ) = max( 22.0 - 12.0 , 0 ) = 10.0  
 T[J2] = max( D[4][J2] - d[J1] , 0 ) = max( 32.0 - 12.0 , 0 ) = 20.0  
 T[J5] = max( D[4][J5] - d[J4] , 0 ) = max( 40.0 - 32.0 , 0 ) = 8.0  
 T[J1] = max( D[4][J1] - d[J0] , 0 ) = max( 48.0 - 20.0 , 0 ) = 28.0  
 T[J3] = max( D[4][J3] - d[J2] , 0 ) = max( 59.0 - 55.0 , 0 ) = 4.0  
 T[J4] = max( D[4][J4] - d[J3] , 0 ) = max( 71.0 - 20.0 , 0 ) = 51.0  
Tardiness TT = ∑ T[S[j]] : j = 0,...,6  
 TT = 121.0

## 3. Matrices D,C,F

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| D | J0 | J6 | J2 | J5 | J1 | J3 | J4 |
| M0 | 0.0 | 2.0 | 10.0 | 21.0 | 32.0 | 42.0 | 50.0 |
| M1 | 0.0 | 6.0 | 19.0 | 29.0 | 39.0 | 47.0 | 59.0 |
| M2 | 0.0 | 15.0 | 27.0 | 34.0 | 43.0 | 51.0 | 64.0 |
| M3 | 0.0 | 22.0 | 32.0 | 40.0 | 48.0 | 59.0 | 71.0 |

**Table 25: dates de fin de chaque job avec son blocage dans chaque machine BFSP-SDST**

C[i][S[j]]:dates de fin de chaque job dans chaque machine BFSP-SDST

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C | J5 | J1 | J4 | J0 | J2 | J3 |
| M0 | 6.0 | 19.0 | 29.0 | 39.0 | 47.0 | 59.0 |
| M1 | 15.0 | 27.0 | 33.0 | 42.0 | 51.0 | 64.0 |
| M2 | 22.0 | 32.0 | 40.0 | 48.0 | 59.0 | 71.0 |

**Table 26: dates de fin de chaque job dans chaque machine BFSP-SDST**

F[i][S[j]]:dates de début de chaque job dans chaque machine BFSP-SDST   
pour trouver les dates de débuts de chaque job: F[i][S[j]] = C[i][S[j]] - P[i][S[j]]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| F | J5 | J1 | J4 | J0 | J2 | J3 |
| M0 | 2.0 | 10.0 | 21.0 | 32.0 | 42.0 | 50.0 |
| M1 | 6.0 | 19.0 | 29.0 | 39.0 | 47.0 | 59.0 |
| M2 | 15.0 | 27.0 | 34.0 | 43.0 | 51.0 | 64.0 |

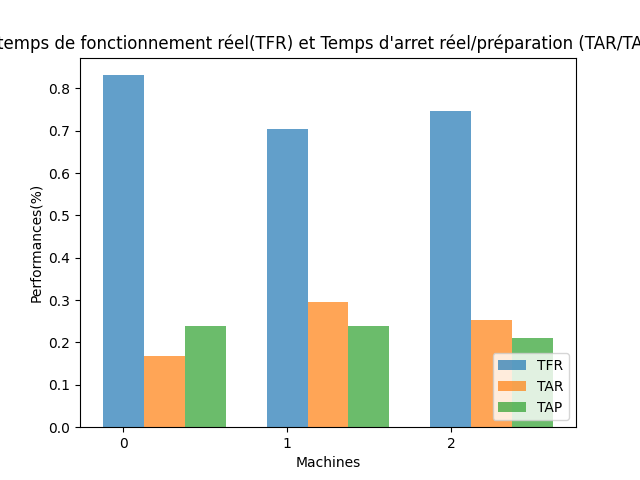
**Table 27: dates de début de chaque job dans chaque machine BFSP-SDST**

## 4. Performances des machines (machines non arretées en préparation)

TFR: (teaux de fonctionnement réel)  
TAR: (teaux d'arret réel)  
TAP: (teaux d'arret de préparation)

|  |  |  |  |
| --- | --- | --- | --- |
|  | TFR | TAR | TAP |
| M0 | 83.1% | 16.9% | 23.94% |
| M1 | 70.42% | 29.58% | 23.94% |
| M2 | 74.65% | 25.35% | 21.13% |

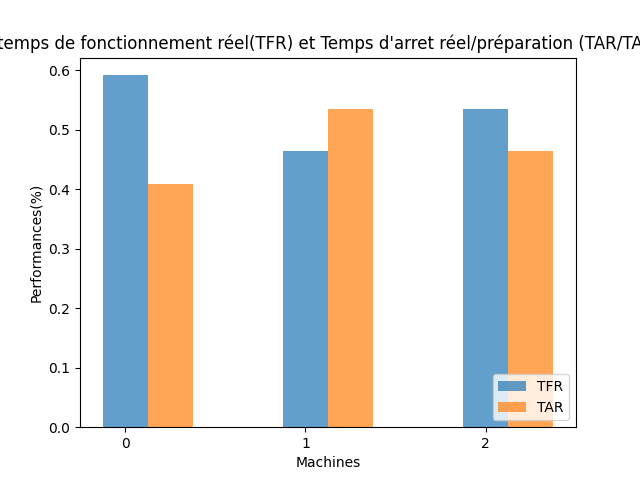
**Table 28: Pérformances des machines BFSP-SDST (machines non arretées en préparation)**



## 5. Performances des machines (machines arretées en préparation)

|  |  |  |
| --- | --- | --- |
|  | TFR | TAR |
| M0 | 83.1% | 16.9% |
| M1 | 70.42% | 29.58% |
| M2 | 74.65% | 25.35% |

**Table 29: Pérformances des machines BFSP-SDST (machines arretées en préparation)**



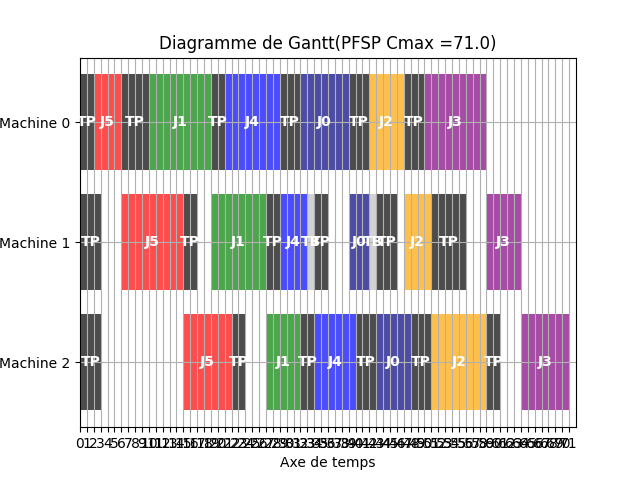
## 6. Temps de blocage de chaque job dans chaque machines

TB: (Blocking Time) temps de blocage de chaque job dans chaque machines

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TB | J5 | J1 | J4 | J0 | J2 | J3 |
| M0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| M1 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| M2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

**Table 30: Temps de blocage de chaque job dans chaque machines BFSP-SDST**

## 7. Diagramme de Gantt,



# 6. NIPFSP (No-Idel Permutation Flow Shop Scheduling Problem)

## 1. Séquence par Test de tous les permutations possibles avec min(Cmax) et min(TT)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | J0 | J4 | J5 | J1 | J3 | J2 |

**Table 31: Séquence trouvée par l'algotrithme de Test de toutes les séquences**

## 2. Etapes de résolution de NIPFSP (Cmax/TFT...)

C[i][S[j]]: matrice des dates de fin de chaque job dans chaque machine

Sans préparation et dans arrets  
équation 1 : pour i = 0 :   
 L[0] = 0  
équation 3 et 4 : pour i = 0 et j = 0,...,5:   
 C[0][J0] = L[0] + P[0][J0] = 0.0 + 7.0 = 7.0  
 C[0][J4] = C[0][J0] + P[0][J4] = 7.0 + 8.0 = 15.0  
 C[0][J5] = C[0][J4] + P[0][J5] = 15.0 + 4.0 = 19.0  
 C[0][J1] = C[0][J5] + P[0][J1] = 19.0 + 9.0 = 28.0  
 C[0][J3] = C[0][J1] + P[0][J3] = 28.0 + 9.0 = 37.0  
 C[0][J2] = C[0][J3] + P[0][J2] = 37.0 + 5.0 = 42.0  
équation 2: pour i=1  
 maxs[0] = P[0][J0] + P[0][J4] + P[0][J5] + P[0][J1] + P[0][J3] + P[0][J2] - P[1][J0] - P[1][J4] - P[1][J5] - P[1][J1] - P[1][J3] = 7.0 + 8.0 + 4.0 + 9.0 + 9.0 + 5.0 - 3.0 - 4.0 - 9.0 - 8.0 - 5.0 = 13.0  
 maxs[1] = P[0][J0] + P[0][J4] + P[0][J5] + P[0][J1] + P[0][J3] - P[1][J0] - P[1][J4] - P[1][J5] - P[1][J1] = 7.0 + 8.0 + 4.0 + 9.0 + 9.0 - 3.0 - 4.0 - 9.0 - 8.0 = 13.0  
 maxs[2] = P[0][J0] + P[0][J4] + P[0][J5] + P[0][J1] - P[1][J0] - P[1][J4] - P[1][J5] = 7.0 + 8.0 + 4.0 + 9.0 - 3.0 - 4.0 - 9.0 = 12.0  
 maxs[3] = P[0][J0] + P[0][J4] + P[0][J5] - P[1][J0] - P[1][J4] = 7.0 + 8.0 + 4.0 - 3.0 - 4.0 = 12.0  
 maxs[4] = P[0][J0] + P[0][J4] - P[1][J0] = 7.0 + 8.0 - 3.0 = 12.0  
 maxs[5] = P[0][J0] - = 7.0 - = 7.0  
 L[1] = L[0] + max( maxs[0] , maxs[1] , maxs[2] , maxs[3] , maxs[4] , maxs[5] ) = 0.0 + max( 13.0 , 13.0 , 12.0 , 12.0 , 12.0 , 7.0 ) = 0.0 + 13.0 = 13.0  
équation 3 et 4 : pour i = 1 et j = 0,...,5:   
 C[1][J0] = L[1] + P[1][J0] = 13.0 + 3.0 = 16.0  
 C[1][J4] = C[1][J0] + P[1][J4] = 16.0 + 4.0 = 20.0  
 C[1][J5] = C[1][J4] + P[1][J5] = 20.0 + 9.0 = 29.0  
 C[1][J1] = C[1][J5] + P[1][J1] = 29.0 + 8.0 = 37.0  
 C[1][J3] = C[1][J1] + P[1][J3] = 37.0 + 5.0 = 42.0  
 C[1][J2] = C[1][J3] + P[1][J2] = 42.0 + 4.0 = 46.0  
équation 2: pour i=2  
 maxs[0] = P[1][J0] + P[1][J4] + P[1][J5] + P[1][J1] + P[1][J3] + P[1][J2] - P[2][J0] - P[2][J4] - P[2][J5] - P[2][J1] - P[2][J3] = 3.0 + 4.0 + 9.0 + 8.0 + 5.0 + 4.0 - 5.0 - 6.0 - 7.0 - 5.0 - 7.0 = 3.0  
 maxs[1] = P[1][J0] + P[1][J4] + P[1][J5] + P[1][J1] + P[1][J3] - P[2][J0] - P[2][J4] - P[2][J5] - P[2][J1] = 3.0 + 4.0 + 9.0 + 8.0 + 5.0 - 5.0 - 6.0 - 7.0 - 5.0 = 6.0  
 maxs[2] = P[1][J0] + P[1][J4] + P[1][J5] + P[1][J1] - P[2][J0] - P[2][J4] - P[2][J5] = 3.0 + 4.0 + 9.0 + 8.0 - 5.0 - 6.0 - 7.0 = 6.0  
 maxs[3] = P[1][J0] + P[1][J4] + P[1][J5] - P[2][J0] - P[2][J4] = 3.0 + 4.0 + 9.0 - 5.0 - 6.0 = 5.0  
 maxs[4] = P[1][J0] + P[1][J4] - P[2][J0] = 3.0 + 4.0 - 5.0 = 2.0  
 maxs[5] = P[1][J0] - = 3.0 - = 3.0  
 L[2] = L[1] + max( maxs[0] , maxs[1] , maxs[2] , maxs[3] , maxs[4] , maxs[5] ) = 13.0 + max( 3.0 , 6.0 , 6.0 , 5.0 , 2.0 , 3.0 ) = 13.0 + 6.0 = 19.0  
équation 3 et 4 : pour i = 2 et j = 0,...,5:   
 C[2][J0] = L[2] + P[2][J0] = 19.0 + 5.0 = 24.0  
 C[2][J4] = C[2][J0] + P[2][J4] = 24.0 + 6.0 = 30.0  
 C[2][J5] = C[2][J4] + P[2][J5] = 30.0 + 7.0 = 37.0  
 C[2][J1] = C[2][J5] + P[2][J1] = 37.0 + 5.0 = 42.0  
 C[2][J3] = C[2][J1] + P[2][J3] = 42.0 + 7.0 = 49.0  
 C[2][J2] = C[2][J3] + P[2][J2] = 49.0 + 8.0 = 57.0  
Calcul des temps d'attendre :  
 TA[0/1][J0] = C[1][J0] - P[1][J0] - C[i][J0] = 16.0 - 3.0 - 7.0 = 6.0  
 TA[0/1][J4] = C[1][J4] - P[1][J4] - C[i][J4] = 20.0 - 8.0 - 15.0 = 1.0  
 TA[0/1][J5] = C[1][J5] - P[1][J5] - C[i][J5] = 29.0 - 4.0 - 19.0 = 1.0  
 TA[0/1][J1] = C[1][J1] - P[1][J1] - C[i][J1] = 37.0 - 5.0 - 28.0 = 1.0  
 TA[0/1][J3] = C[1][J3] - P[1][J3] - C[i][J3] = 42.0 - 4.0 - 37.0 = 0.0  
 TA[0/1][J2] = C[1][J2] - P[1][J2] - C[i][J2] = 46.0 - 9.0 - 42.0 = 0.0  
 TA[1/2][J0] = C[2][J0] - P[2][J0] - C[i][J0] = 24.0 - 5.0 - 16.0 = 3.0  
 TA[1/2][J4] = C[2][J4] - P[2][J4] - C[i][J4] = 30.0 - 5.0 - 20.0 = 4.0  
 TA[1/2][J5] = C[2][J5] - P[2][J5] - C[i][J5] = 37.0 - 8.0 - 29.0 = 1.0  
 TA[1/2][J1] = C[2][J1] - P[2][J1] - C[i][J1] = 42.0 - 7.0 - 37.0 = 0.0  
 TA[1/2][J3] = C[2][J3] - P[2][J3] - C[i][J3] = 49.0 - 6.0 - 42.0 = 0.0  
 TA[1/2][J2] = C[2][J2] - P[2][J2] - C[i][J2] = 57.0 - 7.0 - 46.0 = 3.0  
  
Total flow time TFT = C[2][J0] + C[2][J4] + C[2][J5] + C[2][J1] + C[2][J3] + C[2][J2] =   
24.0 + 30.0 + 37.0 + 42.0 + 49.0 + 57.0 = 239.0  
  
Tardiness T:  
 T[J0] = max( C[2][J0] - d[J0] , 0 ) = max( 24.0 - 20.0 , 0 ) = 4.0  
 T[J4] = max( C[2][J4] - d[J4] , 0 ) = max( 30.0 - 32.0 , 0 ) = 0  
 T[J5] = max( C[2][J5] - d[J5] , 0 ) = max( 37.0 - 12.0 , 0 ) = 25.0  
 T[J1] = max( C[2][J1] - d[J1] , 0 ) = max( 42.0 - 12.0 , 0 ) = 30.0  
 T[J3] = max( C[2][J3] - d[J3] , 0 ) = max( 49.0 - 20.0 , 0 ) = 29.0  
 T[J2] = max( C[2][J2] - d[J2] , 0 ) = max( 57.0 - 55.0 , 0 ) = 2.0  
Tardiness TT = ∑ T[S[j]] : j = 0,...,5  
 TT = 90.0

## 3. Matrice des dates des fins et des débuts de chaque job sur chaque machine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C | J0 | J4 | J5 | J1 | J3 | J2 |
| M0 | 7.0 | 15.0 | 19.0 | 28.0 | 37.0 | 42.0 |
| M1 | 16.0 | 20.0 | 29.0 | 37.0 | 42.0 | 46.0 |
| M2 | 24.0 | 30.0 | 37.0 | 42.0 | 49.0 | 57.0 |

**Table 32: dates de fin de chaque job dans chaque machine NIPFSP**

F[i][S[j]]:dates de début de chaque job dans chaque machine NIPFSP   
pour trouver les dates de débuts de chaque job: F[i][S[j]] = C[i][S[j]] - P[i][S[j]]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| F | J0 | J4 | J5 | J1 | J3 | J2 |
| M0 | 0.0 | 7.0 | 15.0 | 19.0 | 28.0 | 37.0 |
| M1 | 13.0 | 16.0 | 20.0 | 29.0 | 37.0 | 42.0 |
| M2 | 19.0 | 24.0 | 30.0 | 37.0 | 42.0 | 49.0 |

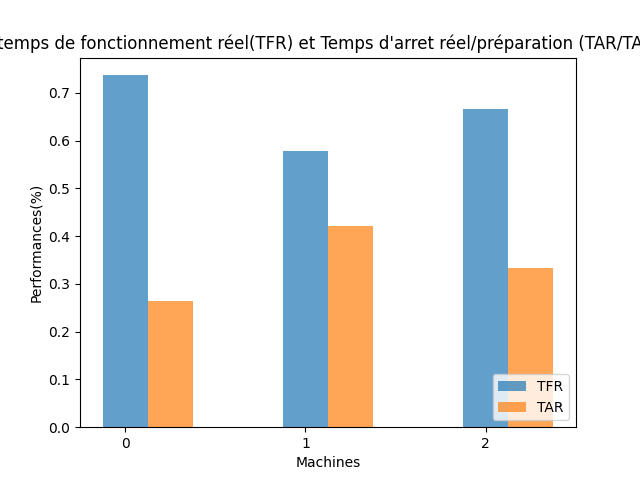
**Table 33: dates de début de chaque job dans chaque machine NIPFSP**

## 4. Performances des machines

TFR: (teaux de fonctionnement réel)  
TAR: (teaux d'arret réel)

|  |  |  |
| --- | --- | --- |
|  | TFR | TAR |
| M0 | 73.68% | 26.32% |
| M1 | 57.89% | 42.11% |
| M2 | 66.67% | 33.33% |

**Table 34: Pérformances des machines NIPFSP**



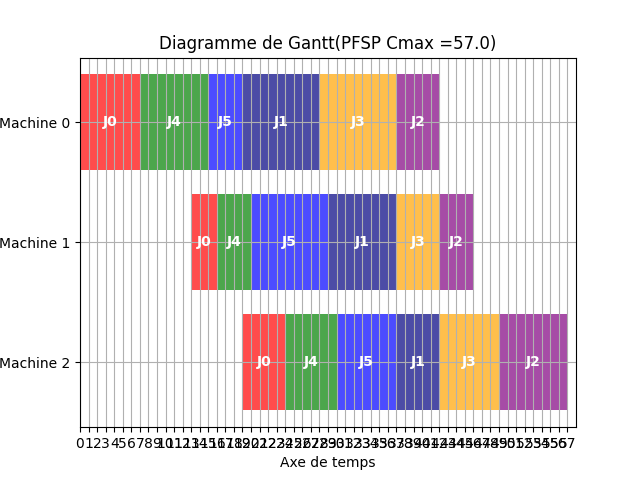
## 5. Temps d'attedre de chaque job entre chaque deux machines

TW: (waiting Time) temps d'attedre de chaque job entre chaque deux machines

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TW | J0 | J4 | J5 | J1 | J3 | J2 |
| M0/1 | 6.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| M1/2 | 3.0 | 4.0 | 1.0 | 0.0 | 0.0 | 3.0 |

**Table 35: Temps d'attendre de chaque job entre chaque deux machines NIPFSP**

## 6. Diagramme de Gantt,



# 7. NWPFSP (No-Wait Permutation Flow Shop Scheduling Problem)

## 1. Séquence par Test de tous les permutations possibles avec min(Cmax) et min(TT)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | J5 | J1 | J3 | J4 | J0 | J2 |

**Table 36: Séquence trouvée par l'algotrithme de Test de toutes les séquences**

## 2. Etapes de résolution de NWPFSP (Cmax/TFT...)

C[i][S[j]]: matrice des dates de fin de chaque job dans chaque machine

Sans préparation et dans attendre  
équation 1 : Calcul de la matrice des délais D entre chaque deux job tel que j1 = 0,...,5 j2 = 0,...,5 :  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J0] - P[0][J0] = 3.0 - 7.0 = -4.0  
 maxs[2] = P[1][J0] + P[2][J0] - P[0][J0] - P[1][J0] = 3.0 + 5.0 - 7.0 - 3.0 = 8.0 - 10.0 = -2.0  
 maxs = [0, -4.0, -2.0]  
 D[J0][J0] = P[0][J0] + max(max(maxs),0) = 7.0 + 0 = 7.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J0] - P[0][J1] = 3.0 - 9.0 = -6.0  
 maxs[2] = P[1][J0] + P[2][J0] - P[0][J1] - P[1][J1] = 3.0 + 5.0 - 9.0 - 8.0 = 8.0 - 17.0 = -9.0  
 maxs = [0, -6.0, -9.0]  
 D[J0][J1] = P[0][J0] + max(max(maxs),0) = 7.0 + 0 = 7.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J0] - P[0][J2] = 3.0 - 5.0 = -2.0  
 maxs[2] = P[1][J0] + P[2][J0] - P[0][J2] - P[1][J2] = 3.0 + 5.0 - 5.0 - 4.0 = 8.0 - 9.0 = -1.0  
 maxs = [0, -2.0, -1.0]  
 D[J0][J2] = P[0][J0] + max(max(maxs),0) = 7.0 + 0 = 7.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J0] - P[0][J3] = 3.0 - 9.0 = -6.0  
 maxs[2] = P[1][J0] + P[2][J0] - P[0][J3] - P[1][J3] = 3.0 + 5.0 - 9.0 - 5.0 = 8.0 - 14.0 = -6.0  
 maxs = [0, -6.0, -6.0]  
 D[J0][J3] = P[0][J0] + max(max(maxs),0) = 7.0 + 0 = 7.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J0] - P[0][J4] = 3.0 - 8.0 = -5.0  
 maxs[2] = P[1][J0] + P[2][J0] - P[0][J4] - P[1][J4] = 3.0 + 5.0 - 8.0 - 4.0 = 8.0 - 12.0 = -4.0  
 maxs = [0, -5.0, -4.0]  
 D[J0][J4] = P[0][J0] + max(max(maxs),0) = 7.0 + 0 = 7.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J0] - P[0][J5] = 3.0 - 4.0 = -1.0  
 maxs[2] = P[1][J0] + P[2][J0] - P[0][J5] - P[1][J5] = 3.0 + 5.0 - 4.0 - 9.0 = 8.0 - 13.0 = -5.0  
 maxs = [0, -1.0, -5.0]  
 D[J0][J5] = P[0][J0] + max(max(maxs),0) = 7.0 + 0 = 7.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J1] - P[0][J0] = 8.0 - 7.0 = 1.0  
 maxs[2] = P[1][J1] + P[2][J1] - P[0][J0] - P[1][J0] = 8.0 + 5.0 - 7.0 - 3.0 = 13.0 - 10.0 = 3.0  
 maxs = [0, 1.0, 3.0]  
 D[J1][J0] = P[0][J1] + max(max(maxs),0) = 9.0 + 3.0 = 12.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J1] - P[0][J1] = 8.0 - 9.0 = -1.0  
 maxs[2] = P[1][J1] + P[2][J1] - P[0][J1] - P[1][J1] = 8.0 + 5.0 - 9.0 - 8.0 = 13.0 - 17.0 = -4.0  
 maxs = [0, -1.0, -4.0]  
 D[J1][J1] = P[0][J1] + max(max(maxs),0) = 9.0 + 0 = 9.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J1] - P[0][J2] = 8.0 - 5.0 = 3.0  
 maxs[2] = P[1][J1] + P[2][J1] - P[0][J2] - P[1][J2] = 8.0 + 5.0 - 5.0 - 4.0 = 13.0 - 9.0 = 4.0  
 maxs = [0, 3.0, 4.0]  
 D[J1][J2] = P[0][J1] + max(max(maxs),0) = 9.0 + 4.0 = 13.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J1] - P[0][J3] = 8.0 - 9.0 = -1.0  
 maxs[2] = P[1][J1] + P[2][J1] - P[0][J3] - P[1][J3] = 8.0 + 5.0 - 9.0 - 5.0 = 13.0 - 14.0 = -1.0  
 maxs = [0, -1.0, -1.0]  
 D[J1][J3] = P[0][J1] + max(max(maxs),0) = 9.0 + 0 = 9.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J1] - P[0][J4] = 8.0 - 8.0 = 0.0  
 maxs[2] = P[1][J1] + P[2][J1] - P[0][J4] - P[1][J4] = 8.0 + 5.0 - 8.0 - 4.0 = 13.0 - 12.0 = 1.0  
 maxs = [0, 0.0, 1.0]  
 D[J1][J4] = P[0][J1] + max(max(maxs),0) = 9.0 + 1.0 = 10.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J1] - P[0][J5] = 8.0 - 4.0 = 4.0  
 maxs[2] = P[1][J1] + P[2][J1] - P[0][J5] - P[1][J5] = 8.0 + 5.0 - 4.0 - 9.0 = 13.0 - 13.0 = 0.0  
 maxs = [0, 4.0, 0.0]  
 D[J1][J5] = P[0][J1] + max(max(maxs),0) = 9.0 + 4.0 = 13.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J2] - P[0][J0] = 4.0 - 7.0 = -3.0  
 maxs[2] = P[1][J2] + P[2][J2] - P[0][J0] - P[1][J0] = 4.0 + 8.0 - 7.0 - 3.0 = 12.0 - 10.0 = 2.0  
 maxs = [0, -3.0, 2.0]  
 D[J2][J0] = P[0][J2] + max(max(maxs),0) = 5.0 + 2.0 = 7.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J2] - P[0][J1] = 4.0 - 9.0 = -5.0  
 maxs[2] = P[1][J2] + P[2][J2] - P[0][J1] - P[1][J1] = 4.0 + 8.0 - 9.0 - 8.0 = 12.0 - 17.0 = -5.0  
 maxs = [0, -5.0, -5.0]  
 D[J2][J1] = P[0][J2] + max(max(maxs),0) = 5.0 + 0 = 5.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J2] - P[0][J2] = 4.0 - 5.0 = -1.0  
 maxs[2] = P[1][J2] + P[2][J2] - P[0][J2] - P[1][J2] = 4.0 + 8.0 - 5.0 - 4.0 = 12.0 - 9.0 = 3.0  
 maxs = [0, -1.0, 3.0]  
 D[J2][J2] = P[0][J2] + max(max(maxs),0) = 5.0 + 3.0 = 8.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J2] - P[0][J3] = 4.0 - 9.0 = -5.0  
 maxs[2] = P[1][J2] + P[2][J2] - P[0][J3] - P[1][J3] = 4.0 + 8.0 - 9.0 - 5.0 = 12.0 - 14.0 = -2.0  
 maxs = [0, -5.0, -2.0]  
 D[J2][J3] = P[0][J2] + max(max(maxs),0) = 5.0 + 0 = 5.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J2] - P[0][J4] = 4.0 - 8.0 = -4.0  
 maxs[2] = P[1][J2] + P[2][J2] - P[0][J4] - P[1][J4] = 4.0 + 8.0 - 8.0 - 4.0 = 12.0 - 12.0 = 0.0  
 maxs = [0, -4.0, 0.0]  
 D[J2][J4] = P[0][J2] + max(max(maxs),0) = 5.0 + 0 = 5.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J2] - P[0][J5] = 4.0 - 4.0 = 0.0  
 maxs[2] = P[1][J2] + P[2][J2] - P[0][J5] - P[1][J5] = 4.0 + 8.0 - 4.0 - 9.0 = 12.0 - 13.0 = -1.0  
 maxs = [0, 0.0, -1.0]  
 D[J2][J5] = P[0][J2] + max(max(maxs),0) = 5.0 + 0 = 5.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J3] - P[0][J0] = 5.0 - 7.0 = -2.0  
 maxs[2] = P[1][J3] + P[2][J3] - P[0][J0] - P[1][J0] = 5.0 + 7.0 - 7.0 - 3.0 = 12.0 - 10.0 = 2.0  
 maxs = [0, -2.0, 2.0]  
 D[J3][J0] = P[0][J3] + max(max(maxs),0) = 9.0 + 2.0 = 11.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J3] - P[0][J1] = 5.0 - 9.0 = -4.0  
 maxs[2] = P[1][J3] + P[2][J3] - P[0][J1] - P[1][J1] = 5.0 + 7.0 - 9.0 - 8.0 = 12.0 - 17.0 = -5.0  
 maxs = [0, -4.0, -5.0]  
 D[J3][J1] = P[0][J3] + max(max(maxs),0) = 9.0 + 0 = 9.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J3] - P[0][J2] = 5.0 - 5.0 = 0.0  
 maxs[2] = P[1][J3] + P[2][J3] - P[0][J2] - P[1][J2] = 5.0 + 7.0 - 5.0 - 4.0 = 12.0 - 9.0 = 3.0  
 maxs = [0, 0.0, 3.0]  
 D[J3][J2] = P[0][J3] + max(max(maxs),0) = 9.0 + 3.0 = 12.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J3] - P[0][J3] = 5.0 - 9.0 = -4.0  
 maxs[2] = P[1][J3] + P[2][J3] - P[0][J3] - P[1][J3] = 5.0 + 7.0 - 9.0 - 5.0 = 12.0 - 14.0 = -2.0  
 maxs = [0, -4.0, -2.0]  
 D[J3][J3] = P[0][J3] + max(max(maxs),0) = 9.0 + 0 = 9.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J3] - P[0][J4] = 5.0 - 8.0 = -3.0  
 maxs[2] = P[1][J3] + P[2][J3] - P[0][J4] - P[1][J4] = 5.0 + 7.0 - 8.0 - 4.0 = 12.0 - 12.0 = 0.0  
 maxs = [0, -3.0, 0.0]  
 D[J3][J4] = P[0][J3] + max(max(maxs),0) = 9.0 + 0 = 9.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J3] - P[0][J5] = 5.0 - 4.0 = 1.0  
 maxs[2] = P[1][J3] + P[2][J3] - P[0][J5] - P[1][J5] = 5.0 + 7.0 - 4.0 - 9.0 = 12.0 - 13.0 = -1.0  
 maxs = [0, 1.0, -1.0]  
 D[J3][J5] = P[0][J3] + max(max(maxs),0) = 9.0 + 1.0 = 10.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J4] - P[0][J0] = 4.0 - 7.0 = -3.0  
 maxs[2] = P[1][J4] + P[2][J4] - P[0][J0] - P[1][J0] = 4.0 + 6.0 - 7.0 - 3.0 = 10.0 - 10.0 = 0.0  
 maxs = [0, -3.0, 0.0]  
 D[J4][J0] = P[0][J4] + max(max(maxs),0) = 8.0 + 0 = 8.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J4] - P[0][J1] = 4.0 - 9.0 = -5.0  
 maxs[2] = P[1][J4] + P[2][J4] - P[0][J1] - P[1][J1] = 4.0 + 6.0 - 9.0 - 8.0 = 10.0 - 17.0 = -7.0  
 maxs = [0, -5.0, -7.0]  
 D[J4][J1] = P[0][J4] + max(max(maxs),0) = 8.0 + 0 = 8.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J4] - P[0][J2] = 4.0 - 5.0 = -1.0  
 maxs[2] = P[1][J4] + P[2][J4] - P[0][J2] - P[1][J2] = 4.0 + 6.0 - 5.0 - 4.0 = 10.0 - 9.0 = 1.0  
 maxs = [0, -1.0, 1.0]  
 D[J4][J2] = P[0][J4] + max(max(maxs),0) = 8.0 + 1.0 = 9.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J4] - P[0][J3] = 4.0 - 9.0 = -5.0  
 maxs[2] = P[1][J4] + P[2][J4] - P[0][J3] - P[1][J3] = 4.0 + 6.0 - 9.0 - 5.0 = 10.0 - 14.0 = -4.0  
 maxs = [0, -5.0, -4.0]  
 D[J4][J3] = P[0][J4] + max(max(maxs),0) = 8.0 + 0 = 8.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J4] - P[0][J4] = 4.0 - 8.0 = -4.0  
 maxs[2] = P[1][J4] + P[2][J4] - P[0][J4] - P[1][J4] = 4.0 + 6.0 - 8.0 - 4.0 = 10.0 - 12.0 = -2.0  
 maxs = [0, -4.0, -2.0]  
 D[J4][J4] = P[0][J4] + max(max(maxs),0) = 8.0 + 0 = 8.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J4] - P[0][J5] = 4.0 - 4.0 = 0.0  
 maxs[2] = P[1][J4] + P[2][J4] - P[0][J5] - P[1][J5] = 4.0 + 6.0 - 4.0 - 9.0 = 10.0 - 13.0 = -3.0  
 maxs = [0, 0.0, -3.0]  
 D[J4][J5] = P[0][J4] + max(max(maxs),0) = 8.0 + 0 = 8.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J5] - P[0][J0] = 9.0 - 7.0 = 2.0  
 maxs[2] = P[1][J5] + P[2][J5] - P[0][J0] - P[1][J0] = 9.0 + 7.0 - 7.0 - 3.0 = 16.0 - 10.0 = 6.0  
 maxs = [0, 2.0, 6.0]  
 D[J5][J0] = P[0][J5] + max(max(maxs),0) = 4.0 + 6.0 = 10.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J5] - P[0][J1] = 9.0 - 9.0 = 0.0  
 maxs[2] = P[1][J5] + P[2][J5] - P[0][J1] - P[1][J1] = 9.0 + 7.0 - 9.0 - 8.0 = 16.0 - 17.0 = -1.0  
 maxs = [0, 0.0, -1.0]  
 D[J5][J1] = P[0][J5] + max(max(maxs),0) = 4.0 + 0 = 4.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J5] - P[0][J2] = 9.0 - 5.0 = 4.0  
 maxs[2] = P[1][J5] + P[2][J5] - P[0][J2] - P[1][J2] = 9.0 + 7.0 - 5.0 - 4.0 = 16.0 - 9.0 = 7.0  
 maxs = [0, 4.0, 7.0]  
 D[J5][J2] = P[0][J5] + max(max(maxs),0) = 4.0 + 7.0 = 11.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J5] - P[0][J3] = 9.0 - 9.0 = 0.0  
 maxs[2] = P[1][J5] + P[2][J5] - P[0][J3] - P[1][J3] = 9.0 + 7.0 - 9.0 - 5.0 = 16.0 - 14.0 = 2.0  
 maxs = [0, 0.0, 2.0]  
 D[J5][J3] = P[0][J5] + max(max(maxs),0) = 4.0 + 2.0 = 6.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J5] - P[0][J4] = 9.0 - 8.0 = 1.0  
 maxs[2] = P[1][J5] + P[2][J5] - P[0][J4] - P[1][J4] = 9.0 + 7.0 - 8.0 - 4.0 = 16.0 - 12.0 = 4.0  
 maxs = [0, 1.0, 4.0]  
 D[J5][J4] = P[0][J5] + max(max(maxs),0) = 4.0 + 4.0 = 8.0  
 maxs[0] = 0 - 0 = 0  
 maxs[1] = P[1][J5] - P[0][J5] = 9.0 - 4.0 = 5.0  
 maxs[2] = P[1][J5] + P[2][J5] - P[0][J5] - P[1][J5] = 9.0 + 7.0 - 4.0 - 9.0 = 16.0 - 13.0 = 3.0  
 maxs = [0, 5.0, 3.0]  
 D[J5][J5] = P[0][J5] + max(max(maxs),0) = 4.0 + 5.0 = 9.0  
équation 2 : Calcul de la dernière ligne de la matrice C pour i = 2 et j = 0,...,5 :  
C[2][J5] = P[0][J5] + P[1][J5] + P[2][J5] = 4.0 + 9.0 + 7.0 = 20.0 = 20.0  
C[2][J1] = D[J5][J1] + P[0][J1] + P[1][J1] + P[2][J1] = 4.0 + 9.0 + 8.0 + 5.0 = 4.0 + 22.0 = 26.0  
C[2][J3] = D[J5][J1] + D[J1][J3] + P[0][J3] + P[1][J3] + P[2][J3] = 4.0 + 9.0 + 9.0 + 5.0 + 7.0 = 13.0 + 21.0 = 34.0  
C[2][J4] = D[J5][J1] + D[J1][J3] + D[J3][J4] + P[0][J4] + P[1][J4] + P[2][J4] = 4.0 + 9.0 + 9.0 + 8.0 + 4.0 + 6.0 = 22.0 + 18.0 = 40.0  
C[2][J0] = D[J5][J1] + D[J1][J3] + D[J3][J4] + D[J4][J0] + P[0][J0] + P[1][J0] + P[2][J0] = 4.0 + 9.0 + 9.0 + 8.0 + 7.0 + 3.0 + 5.0 = 30.0 + 15.0 = 45.0  
C[2][J2] = D[J5][J1] + D[J1][J3] + D[J3][J4] + D[J4][J0] + D[J0][J2] + P[0][J2] + P[1][J2] + P[2][J2] = 4.0 + 9.0 + 9.0 + 8.0 + 7.0 + 5.0 + 4.0 + 8.0 = 37.0 + 17.0 = 54.0  
équation 3 : Calcul de la matrice C pour i = 1,...,0 et j = 0,...,5 :  
C[1][J5] = C[2][J5] - P[2][J5] = 20.0 - 7.0 = 13.0  
C[1][J1] = C[2][J1] - P[2][J1] = 26.0 - 5.0 = 21.0  
C[1][J3] = C[2][J3] - P[2][J3] = 34.0 - 7.0 = 27.0  
C[1][J4] = C[2][J4] - P[2][J4] = 40.0 - 6.0 = 34.0  
C[1][J0] = C[2][J0] - P[2][J0] = 45.0 - 5.0 = 40.0  
C[1][J2] = C[2][J2] - P[2][J2] = 54.0 - 8.0 = 46.0  
C[0][J5] = C[1][J5] - P[1][J5] = 13.0 - 9.0 = 4.0  
C[0][J1] = C[1][J1] - P[1][J1] = 21.0 - 8.0 = 13.0  
C[0][J3] = C[1][J3] - P[1][J3] = 27.0 - 5.0 = 22.0  
C[0][J4] = C[1][J4] - P[1][J4] = 34.0 - 4.0 = 30.0  
C[0][J0] = C[1][J0] - P[1][J0] = 40.0 - 3.0 = 37.0  
C[0][J2] = C[1][J2] - P[1][J2] = 46.0 - 4.0 = 42.0  
  
Total flow time TFT = C[2][J5] + C[2][J1] + C[2][J3] + C[2][J4] + C[2][J0] + C[2][J2] =   
20.0 + 26.0 + 34.0 + 40.0 + 45.0 + 54.0 = 219.0  
  
Tardiness T:  
 T[J5] = max( C[2][J5] - d[J5] , 0 ) = max( 20.0 - 12.0 , 0 ) = 8.0  
 T[J1] = max( C[2][J1] - d[J1] , 0 ) = max( 26.0 - 12.0 , 0 ) = 14.0  
 T[J3] = max( C[2][J3] - d[J3] , 0 ) = max( 34.0 - 20.0 , 0 ) = 14.0  
 T[J4] = max( C[2][J4] - d[J4] , 0 ) = max( 40.0 - 32.0 , 0 ) = 8.0  
 T[J0] = max( C[2][J0] - d[J0] , 0 ) = max( 45.0 - 20.0 , 0 ) = 25.0  
 T[J2] = max( C[2][J2] - d[J2] , 0 ) = max( 54.0 - 55.0 , 0 ) = 0  
Tardiness TT = ∑ T[S[j]] : j = 0,...,5  
 TT = 69.0

## 3. Matrice des dates des fins et des débuts de chaque job sur chaque machine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C | J5 | J1 | J3 | J4 | J0 | J2 |
| M0 | 4.0 | 13.0 | 22.0 | 30.0 | 37.0 | 42.0 |
| M1 | 13.0 | 21.0 | 27.0 | 34.0 | 40.0 | 46.0 |
| M2 | 20.0 | 26.0 | 34.0 | 40.0 | 45.0 | 54.0 |

**Table 37: dates de fin de chaque job dans chaque machine NWPFSP**

F[i][S[j]]:dates de début de chaque job dans chaque machine NWPFSP   
pour trouver les dates de débuts de chaque job: F[i][S[j]] = C[i][S[j]] - P[i][S[j]]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| F | J5 | J1 | J3 | J4 | J0 | J2 |
| M0 | 0.0 | 4.0 | 13.0 | 22.0 | 30.0 | 37.0 |
| M1 | 4.0 | 13.0 | 22.0 | 30.0 | 37.0 | 42.0 |
| M2 | 13.0 | 21.0 | 27.0 | 34.0 | 40.0 | 46.0 |

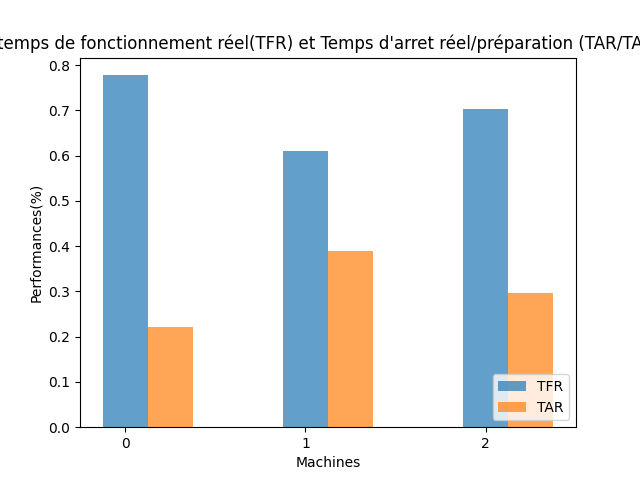
**Table 38: dates de début de chaque job dans chaque machine NWPFSP**

## 4. Performances des machines

TFR: (teaux de fonctionnement réel)  
TAR: (teaux d'arret réel)

|  |  |  |
| --- | --- | --- |
|  | TFR | TAR |
| M0 | 77.78% | 22.22% |
| M1 | 61.11% | 38.89% |
| M2 | 70.37% | 29.63% |

**Table 39: Pérformances des machines NWPFSP**



## 5. Diagramme de Gantt,

