



**HAL**  
open science

# L'Optimisation énergétique de l'ordonnancement des systèmes de production

Minh Hung Ho

► **To cite this version:**

Minh Hung Ho. L'Optimisation énergétique de l'ordonnancement des systèmes de production. Energie électrique. Université de Technologie de Troyes, 2020. Français. NNT: 2020TROY0007. tel-03808645

**HAL Id: tel-03808645**

**<https://theses.hal.science/tel-03808645>**

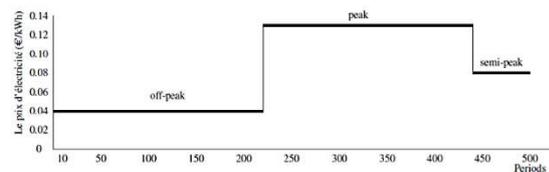
Submitted on 10 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

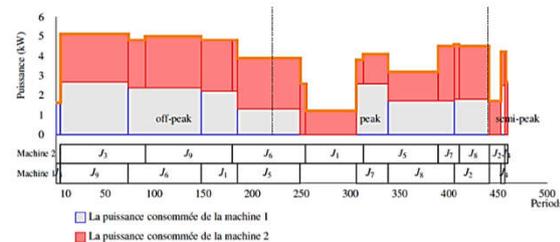
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Minh Hung HO

# L'optimisation énergétique de l'ordonnancement des systèmes de production



(a) La tarification d'électricité ToU



(b) L'évolution de la consommation énergétique du système

Figure : L'évolution de la consommation d'électricité du système sous la tarification d'électricité ToU

**Champ disciplinaire :**  
Sciences pour l'Ingénieur

---

---

# THESE

*pour l'obtention du grade de*

## DOCTEUR

de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

EN SCIENCES POUR L'INGENIEUR

**Spécialité : OPTIMISATION ET SURETE DES SYSTEMES**

*présentée et soutenue par*

**Minh Hung HO**

*le 23 septembre 2020*

---

---

**L'optimisation énergétique de l'ordonnancement  
des systèmes de production**

---

---

## JURY

M. Lionel AMODEO	PROFESSEUR DES UNIVERSITES	Président
M. Alexandre DOLGUI	PROFESSEUR IMT NANTES - HDR	Rapporteur
Mme Nathalie SAUER	PROFESSEURE DES UNIVERSITES	Rapporteure
Mme Valeria BORODIN	MAITRE ASSISTANTE ENSM SAINT-ETIENNE	Examinatrice
M. Frédéric DUGARDIN	MAITRE DE CONFERENCES	Directeur de thèse
M. Faicel HNAIEN	MAITRE DE CONFERENCES - HDR	Directeur de thèse

## Remerciement

J'exprime mes remerciements les plus sincères et mon profond respect à mes directeurs de thèse : Dr. Faïcel HNAÏEN et Dr. Frédéric DUGARDIN, de m'avoir encadré pour la réalisation de ces travaux. Qu'ils trouvent ici l'expression de ma profonde gratitude et ma haute reconnaissance pour la collaboration active et fructueuse que nous avons eu. Grâce à leur patience, leur disponibilité, ainsi qu'à leurs encouragements, ce mémoire a pu être concrétisé et mené à terme.

Le travail dans ce mémoire a été effectué dans l'équipe Logistique et Optimisation des Systèmes Industriels (LOSI) de l'Université de Technologie de Troyes.

J'adresse toute ma gratitude aux membres du jury qui font l'honneur de participer à l'examen de cette thèse : Pr. Nathalie SAUER, Professeur à l'Université Lorraine et Pr. Alexandre DOLGUI, Professeur à l'École des Mines en leur qualité de rapporteurs. Je remercie également Pr. Lionel AMODEO, Professeur à l'Université de Technologie de Troyes, Dr. Valeria BORODIN, maître assistant à l'École des Mines de Saint-Étienne en leur qualité d'examineurs.

Je tiens à remercier mes chers amis et collègues, Guillaume, Paulin, Miguel, Mariam, Abbas, Alice, Cyril et Trinh doctorants à l'UTT pour leurs soutiens et leurs partages, qui m'ont apporté des réflexions et des inspirations. Je remercie mon comité de suivi individuel Dr. Yassine OUAZENE, Pr. Nacima LABADIE et Dr. Laurence DIEULLE pour leur suivi et leur attention. Je remercie également Dr. Taha ARBAOUI, Dr. Mohammadmohsen AGHELINEDJAD et Dr. Hasan Murat AFSAR ainsi que les autres membres au LOSI pour leurs encouragements, leurs présences et leurs dévouements exemplaires.

Je tiens à remercier Madame Pascale DENIS, Madame Isabelle LECLERCQ, Madame Véronique BANSE pour leur aide dans les démarches administratives pendant mes séjours à l'UTT. J'aimerais adresser mes plus profonds remerciements à mes parents, qui m'ont toujours encouragé et soutenu tout au long de mon cursus. Ils ont su m'offrir les meilleures chances pour réussir et pour poursuivre ma carrière. Enfin, c'est avec grand plaisir que je salue tous les amis qui m'ont offert leur soutien moral pendant cette expérience précieuse de ma vie.

# Table des Matières

<b>Remerciement</b>	<b>1</b>
<b>Listes des tableaux</b>	<b>4</b>
<b>Listes des figures</b>	<b>6</b>
<b>Nomenclature</b>	<b>9</b>
<b>1 Introduction</b>	<b>12</b>
1.1 Contexte général . . . . .	12
1.2 Les notions préliminaires . . . . .	16
1.2.1 Introduction générale aux problèmes d’ordonnancement . . . . .	16
1.2.1.1 Définition . . . . .	16
1.2.1.2 Les éléments fondamentaux et leurs notations . . . . .	17
1.2.1.3 Quelques notions sur la complexité des problèmes et des algorithmes . . . . .	20
1.2.1.4 Approches de résolution . . . . .	21
1.2.2 “Demand Side Management” et les tarifications d’électricité . . . . .	24
1.2.2.1 “Demand Side Management” . . . . .	24
1.2.2.2 Les tarifications d’électricité . . . . .	26
1.3 Problématique de cette étude . . . . .	28
<b>2 L’état de l’art</b>	<b>31</b>
2.1 L’efficacité énergétique (“Energy Efficiency”) . . . . .	31
2.1.1 La consommation énergétique liée à l’état de fonctionnement de machine . . . . .	31
2.1.2 La stratégie “Turn off/ Turn on” de la machine . . . . .	32
2.1.3 Le choix de vitesse de traitement . . . . .	32
2.1.4 Le choix de machine de traitement . . . . .	34
2.1.5 L’émission de $CO_2$ . . . . .	35
2.1.6 La consommation énergétique liée à la fiabilité du système . . . . .	36
2.1.7 La consommation énergétique différente entre des tâches . . . . .	36
2.1.8 Les contraintes énergétiques . . . . .	37
2.2 La réponse à la demande (“Demand Response”) . . . . .	37
2.2.1 “Time of Use” . . . . .	38
2.2.2 “Critical Peak Pricing” . . . . .	39
2.2.3 “Real Time Pricing” . . . . .	40
2.3 Conclusion . . . . .	41
<b>3 La minimisation de coût d’électricité de la solution optimale de flow shop à deux machines sous ToU.</b>	<b>45</b>
3.1 La description du problème . . . . .	45
3.2 La modélisation mathématique . . . . .	48

3.3	L'approche de résolution . . . . .	50
3.3.1	L'ensemble des solutions de $F2 perm C_{max}$ . . . . .	50
3.3.1.1	La règle de Johnson . . . . .	50
3.3.1.2	L'ensemble des solutions optimales équivalentes . . . . .	51
3.3.1.3	L'algorithme de [Hadda et al. 2018] . . . . .	52
3.3.1.4	Extension de l'ensemble des solutions optimales équivalentes . . . . .	57
3.3.2	Le problème $F2, ToU sequence, Pu_{ij} TEC$ . . . . .	66
3.3.3	Le problème $F2, ToU perm, C_{max}^* TEC$ . . . . .	70
3.3.4	La borne inférieure . . . . .	72
3.4	Les tests numériques . . . . .	73
3.4.1	La génération des données . . . . .	73
3.4.2	Les résultats numériques . . . . .	74
3.5	Conclusion . . . . .	76
<b>4</b>	<b>Optimisation du coût d'électricité d'un flow shop sous ToU liée aux états de fonctionnement des machines</b> . . . . .	<b>86</b>
4.1	La problématique . . . . .	86
4.2	La règle de Johnson et la programmation dynamique . . . . .	90
4.2.1	La règle de Johnson . . . . .	90
4.2.2	La programmation dynamique [Wang et al. (2018)] . . . . .	91
4.3	Les propositions fondées sur la règle de Johnson . . . . .	92
4.3.1	Deux intervalles de prix . . . . .	92
4.3.2	Plusieurs intervalles de prix . . . . .	96
4.4	Modélisation mathématique . . . . .	106
4.4.1	Modèle mathématique de [Wang et al. (2018)] . . . . .	106
4.4.2	La transformation de la fonction objectif . . . . .	108
4.4.3	Nouveau modèle mathématique ( $M1$ ) . . . . .	109
4.4.4	L'amélioration de modèle mathématique par la règle de Johnson . . . . .	110
4.5	La "Logic-Based Benders Decomposition" . . . . .	111
4.5.1	Introduction . . . . .	112
4.5.2	Le problème maître d'affectation des tâches aux intervalles de prix . . . . .	113
4.5.3	Le problème esclave de date de début optimale et la coupe de Benders . . . . .	115
4.6	Les tests numériques . . . . .	117
4.6.1	La génération des données . . . . .	117
4.6.2	Les résultats numériques . . . . .	118
4.7	Conclusion . . . . .	119
<b>5</b>	<b>Conclusion et perspectives</b> . . . . .	<b>123</b>
<b>6</b>	<b>Bibliographie</b> . . . . .	<b>126</b>

## Listes des tableaux

1	Les aspects énergétiques dans les études d’ordonnancement sur une seule machine. . . . .	42
2	Les aspects énergétiques dans les études d’ordonnancement d’atelier de type flow shop. . . . .	43
3	Les aspects énergétiques dans les études d’ordonnancement d’atelier de machines parallèles et job shop. . . . .	44
4	Le temps de traitement et la puissance requise pour chaque tâche .	46
5	La comparaison des solutions optimales équivalentes $F2 perm C_{max}$	47
6	Notations et paramètres . . . . .	49
7	Variables . . . . .	49
8	Le temps de traitement de l’exemple 1 . . . . .	51
9	Le temps de traitement de l’exemple 2 . . . . .	60
10	Le temps de traitement de l’exemple 3 . . . . .	61
11	Le temps de traitement de l’exemple 4 . . . . .	65
12	La moyenne, max, min et l’écart-type des résultats des heuristiques pour toutes les instances . . . . .	76
13	La comparaison entre les heuristiques basées sur Johnson avec la meilleure solution trouvée “BKS” pour les instances de 5, 10 et 15 tâches . . . . .	77
14	La comparaison entre les heuristiques basées sur Hadda avec la meilleure solution trouvée “BKS” pour les instances de 5, 10 et 15 tâches . . . . .	78
15	La comparaison entre <i>HFF</i> , <i>Cplex</i> et <i>ILSDP</i> avec la meilleure solution trouvée “BKS” pour les instances de 5, 10 et 15 tâches . . .	79
16	La comparaison entre les heuristiques basées sur Johnson avec la meilleure solution trouvée “BKS” pour les instances de 20 tâches .	80
17	La comparaison entre les heuristiques basées sur Hadda avec la meilleure solution trouvée “BKS” pour les instances de 20 tâches.	81
18	La comparaison entre <i>HFF</i> , <i>Cplex</i> et <i>ILSDP</i> avec la meilleure solution trouvée “BKS” pour les instances de 20 tâches. . . . .	82
19	La comparaison entre les heuristiques, <i>Cplex</i> et <i>ILSDP</i> [Wang et al. (2018)] avec la borne inférieure pour les instances de 10 et 15 tâches.	83
20	La comparaison entre les heuristiques, <i>Cplex</i> et <i>ILSDP</i> [Wang et al. (2018)] avec la borne inférieure pour les instances de 20 tâches. .	84
21	Le temps de traitement et la puissance consommée liée aux fonctionnements des machines. . . . .	87
22	La comparaison du coût d’électricité total entre $\pi_1$ et $\pi_2$ . . . . .	87
23	Le bilan des tests numériques . . . . .	119
24	La comparaison en terme de temps de calcul entre les approches de résolution pour les instances de 6, 8 tâches. . . . .	120

25	La comparaison en terme de l'écart avec la "meilleure solution trouvée" entre les approches de résolution pour les instances de 20,30 tâches. . . . .	121
26	L'écart d'optimalité donné par <i>CPLEX</i> de <i>LBD</i> pour les instances de 20,30 tâches . . . . .	122

## Listes des figures

1	L'évolution mondiale de la consommation énergétique, de l'émission de dioxyde de carbone et de la température de 1850 à 2018 . . . . .	13
	(a) La consommation d'énergie mondiale de 1850 à 2018 (Ritchie and Roser (2020b)) . . . . .	13
	(b) L'émission mondiale de dioxyde de carbone de 1850 à 2018 (Ritchie and Roser (2020a)) . . . . .	13
	(c) Anomalie de la température moyenne mondiale par rapport à la température moyenne de période 1961-1990 (° C) de 1850 à 2018 (Ritchie and Roser (2020a)) . . . . .	13
2	La répartition par secteur de la consommation énergétique mondiale et de l'émission de carbone. . . . .	14
	(a) La consommation énergétique selon secteur (Outlook (2019)) . . . . .	14
	(b) L'émission mondiale de $CO_2$ par secteur en 2017 (IEA (2019)) . . . . .	14
3	Émissions de $CO_2$ en grammes par kilowatt-heure générées par la production d'électricité en France le 23/10/2019. . . . .	16
4	Les tarifications d'électricité de la réponse à la demande. . . . .	27
	(a) Le tarif ToU de "Pacific Gas & Electric Company" (Wang and Li (2016)) . . . . .	27
	(b) Le tarif CPP de "Pacific Gas & Electric Company" (Wang and Li (2016)) . . . . .	27
	(c) Le tarif RTP de "ComEd power company" à Chicago, Etats-Unis le 15/09/2019 (ComED (2019)) . . . . .	27
5	Exemple de l'atelier de type flow shop à deux machines. . . . .	28
6	L'évolution de la consommation d'électricité du système sous la tarification ToU. . . . .	29
	(a) La tarification d'électricité ToU . . . . .	29
	(b) L'évolution de la consommation énergétique du système . . . . .	29
7	La comparaison entre la solution 1 et la solution 2 . . . . .	47
	(a) La tarification d'électricité de ToU . . . . .	47
	(b) L'évolution de la consommation énergétique de la solution 1 et la solution 2 . . . . .	47
8	Les solutions équivalentes de l'ordre partiel total $J_1 \prec \{J_2, J_3\} \prec J_4 \prec \{J_5, J_6\} \prec J_7$ . . . . .	53
9	Les "ground job", les groupes de tâches et les temps d'inactivité. . . . .	54
10	L'ordre partiel total $SPO$ de l'exemple 2 (M1, M2 sont les machines 1 et 2 respectivement). . . . .	60
11	Les séquences $\pi_1$ et $\pi_2$ de l'exemple 3 . . . . .	61
	(a) La séquence $\pi_1$ . . . . .	61
	(b) La séquence $\pi_2$ . . . . .	61
12	L'exemple de l'algorithme "Change2" . . . . .	65

	(a)	La séquence $\pi_z^1$ avec le “crossover job” $J_8$ . . . . .	65
	(b)	La transformation des tâches de $J_z^1$ . . . . .	65
	(c)	Le nouveau ordre partiel total $SPO^*$ avec $\bar{J}$ et $J'$ . . . . .	65
	(d)	La nouvelle séquence obtenue avec l’algorithme “Change2” . . . . .	65
13		La décomposition de $(F2, ToU sequence, TW, Pu_{ij} TEC)$ en deux problèmes $(1, ToU sequence, TW, Pu_{ij} TEC)$ . . . . .	68
	(a)	Les tâches et les fenêtres de temps de leurs dates de début . . . . .	68
	(b)	Le problème $1, ToU sequence, TW, Pu_{ij} TEC$ sur la machine 2 . . . . .	68
	(c)	Le problème $1, ToU sequence, TW, Pu_{ij} TEC$ sur la machine 1 . . . . .	68
14		L’exemple de détermination de date de début optimale des tâches $J_1, J_2$ et $J_3$ . . . . .	70
	(a)	Les fenêtres de temps pour les tâches $J_1, J_2$ et $J_3$ . . . . .	70
	(b)	La modélisation par graphe pour la détermination de la date de début optimale de $J_1, J_2, J_3$ . . . . .	70
15		L’évolution de la consommation d’électricité de la solution $\pi_1$ et $\pi_2$ sous la tarification ToU. . . . .	88
	(a)	La tarification d’électricité ToU . . . . .	88
	(b)	L’évolution de la consommation énergétique de la séquence $\pi_1 = J_5 \prec J_2 \prec J_3 \prec J_1 \prec J_4 \prec J_6$ . . . . .	88
	(c)	L’évolution de la consommation énergétique de la séquence $\pi_2 = J_5 \prec J_3 \prec J_2 \prec J_1 \prec J_4 \prec J_6$ . . . . .	88
16		La permutation entre $J_k$ et $J_j$ . . . . .	96
	(a)	La séquence de Johnson . . . . .	96
	(b)	La séquence $\pi$ . . . . .	96
17		La comparaison entre la séquence $\pi^*$ et la séquence de $\pi$ . . . . .	99
	(a)	La séquence $\pi^*$ . . . . .	99
	(b)	La séquence $\pi$ qui suit la règle de Johnson et retarde la date de début des tâches . . . . .	99
18		Comparaison entre la séquence $\pi^*$ et la séquence de $\pi$ . . . . .	102
	(a)	La séquence $\pi^*$ . . . . .	102
	(b)	La séquence $\pi^*$ avec le retardement de traitement des tâches . . . . .	102
	(c)	La séquence $\pi$ qui suit la règle de Johnson et retarde la date de début des tâches . . . . .	102
19		La comparaison entre $\pi^*$ et la séquence de $\pi$ . . . . .	105
	(a)	La séquence $\pi^*$ . . . . .	105
	(b)	La séquence $\pi^*$ qui retarde la date de début des tâches . . . . .	105
	(c)	La séquence $\pi$ qui respecte la règle de Johnson et retarde la date de début des tâches . . . . .	105
20		La séquence $\pi$ et la réorganisation $\pi$ selon la règle de Johnson . . . . .	106
	(a)	La séquence $\pi$ . . . . .	106
	(b)	Réorganisation de la séquence $\pi$ selon la règle de Johnson . . . . .	106

21	La présentation de variable $\alpha_{ijp}$ pour la tâche $J_1$ sur la machine 1	108
22	La “Logic-based Benders Decomposition” pour le problème étudié $F2 Pu_{ij} = Pu_{ij'}, Pu_{idle}, ToU TEC$ . . . . .	116

## Nomenclature

$x_{jj'}$	Variable binaire, = 1 si la tâche $j$ est placée avant la tâche $j'$
$\alpha_{ijp}$	Variable entière, la quantité de traitement de la tâche $j$ sur la machine $i$ pendant l'intervalle de prix $p$
$\beta_{ijp}$	Variable binaire, = 1 si la tâche $j$ est traitée sur la machine $i$ pendant l'intervalle de prix $p$ , et = 0 sinon
$\gamma_{jp}$	Variable binaire, = 1 si la tâche $j$ commence sur la machine 2 pendant l'intervalle de prix $p$ , et = 0 sinon
$\omega_{jj'}^p$	Variable binaire, = 1 si la tâche $j$ et la tâche $j'$ sont commencent leurs traitement sur la machine 2 pendant le même intervalle de prix $p$ , et = 0 sinon
$\bar{E}_t$	Le niveau souhaité de la consommation énergétique du système pendant la période $t$
$\pi$	Une séquence de tâches, où $\pi_j$ indique la tâche à la $j^{eme}$ position dans $\pi$
$\sum C_j$	La somme des dates d'achèvement
$\sum CT_j$	Le coût total lié au retard
$\sum NPE_i$	La consommation énergétique des opérations hors usinage
$\sum PE_i$	La consommation énergétique totale
$\sum T_j$	Le retard total
$\sum w_j C_j$	La somme pondérée des dates d'achèvement
$\sum w_j T_j$	La somme pondérée des retards
$\sum Q_j$	La qualité totale des tâches
$\theta_p$	Le groupe de tâches qui est associé à "ground job" $g_p$
$b_i$	La quantité d'énergie consommée par unité de temps lorsque la machine $i$ est en marche
$C_{max}^*$	La date d'achèvement maximale de la règle de Johnson
$C_{ij}$	La date d'achèvement de tâche $j$ sur la machine $i$
$C_j$	La date de fin d'exécution de la tâche $j$ sur la dernière machine
$C_{max}$	La date d'achèvement maximale

$c_t$	Le prix d'électricité de la période de temps $t$
$CE$	L'émission de carbone
$CPP$	La tarification d'électricité de type "Critical Peak Pricing"
$d_i$	La quantité d'énergie consommée par unité de temps lorsque la machine $i$ est en idle
$d_j$	La date d'échéance de la tâche $j$
$d_p$	La durée de l'intervalle de prix $p$
$DR$	La réponse à la demande
$DSM$	"Demand Side Management"
$E_{it}$	La consommation énergétique de la machine $i$ pendant la période $t$
$E_j$	L'avance par rapport au délai souhaité de la tâche $j$
$e_p$	La date de fin de l'intervalle de prix $p$
$EC$	La conservation de l'énergie
$EE$	L'efficacité énergétique
$F_i$	La date d'arrêt de la machine $i$
$F_j$	Le temps de présence de la tâche $j$ dans l'atelier
$fc_p$	Le prix d'électricité de l'intervalle de prix $p$
$g_{it}$	Variable binaire, = 1 si la machine $i$ est en marche pendant la période $t$ , et = 0 sinon
$g_p$	La tâche dite "ground job" de la séquence
$J$	L'ensemble des tâches
$J_z^1$	L'ensemble des tâches qui précèdent $J_z$
$J_z^2$	L'ensemble des tâches qui succèdent $J_z$
$J_z$	La tâche dite "crossover job" de la séquence
$K$	Le nombre d'intervalles de prix de l'horizon de travail
$L_j$	L'écart par rapport au délai souhaité de la tâche $j$
$LC$	Le coût de main d'oeuvre
$M$	L'ensemble des machines

$m$	Le nombre total de machines
$n$	Le nombre total de tâches
$p_{ij}$	Le temps de traitement de tâche $j$ sur la machine $i$
$Pu_{ij}$	La puissance énergétique requise pour traiter la tâche $j$ sur la machine $i$
$Pu_{max}$	La puissance consommée maximale
$r_j$	La date disponible de la tâche $j$
$RTP$	La tarification d'électricité de type "Real Time Pricing"
$S_{ij}$	La date de début de la tâche $j$ sur la machine $i$
$SPO$	L'ordre partiel total
$St_p$	La date de début de l'intervalle de prix $p$
$T$	Le nombre de période de temps de l'horizon de travail
$T_j$	Le retard par rapport au délai souhaité de la tâche $j$
$TAS$	La disponibilité totale du système
$TC$	Le coût de production total
$TEC$	Le coût d'électricité total
$ToU$	La tarification d'électricité de type "Time of Use"
$TW_j$	La fenêtre des dates de début possibles de la tâche $j$
$w_j$	Le poids d'importance de la tâche $j$
$WL_i$	La charge de travail sur la machine $i$
$z_{ijt}$	Variable binaire, = 1 si la tâche $j$ est traitée sur la machine $i$ pendant la période $t$

# 1 Introduction

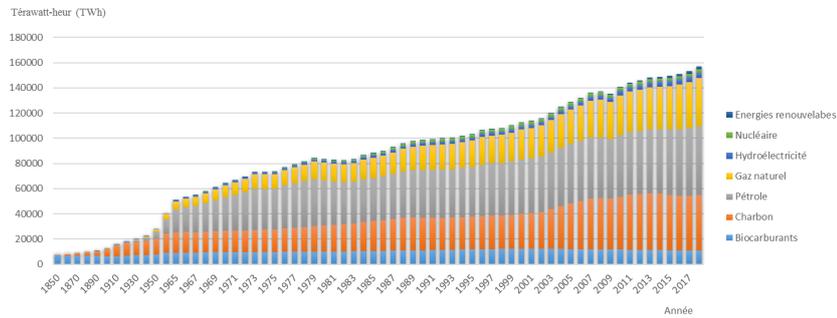
## 1.1 Contexte général

A la fin du *XVIII<sup>ème</sup>* siècle, l'apparition de la machine à vapeur contribue largement à la première révolution industrielle. Cela marque l'aube de l'industrialisation qui deviendra ensuite la clé de meilleures conditions de travail, d'un meilleur niveau de vie, de l'augmentation des richesses, etc. Mais toute médaille a son revers, les industries consomment rapidement les énergies fossiles et augmentent considérablement les quantités de dioxyde de carbone et d'autres gaz à effet de serre dans l'atmosphère. C'est à partir de la fin de *XIX<sup>ème</sup>*, à l'avènement de l'ère industrielle, que l'on voit apparaître le réchauffement climatique.

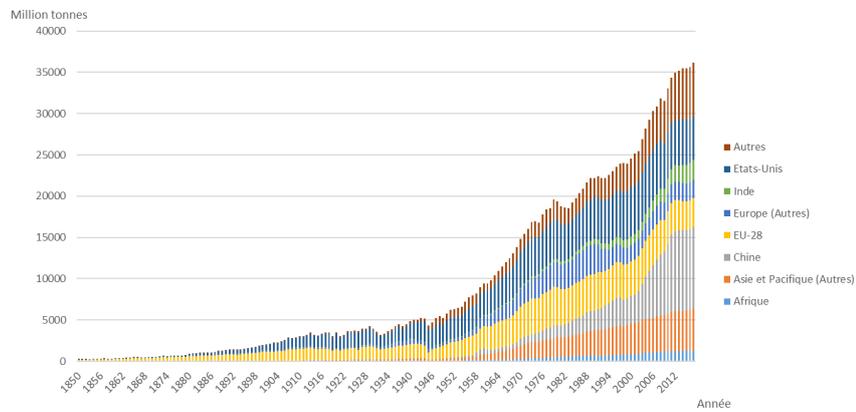
De 1850 à 2018, la consommation énergétique mondiale et l'émission mondiale de dioxyde de carbone n'ont cessé d'augmenter. La consommation énergétique mondiale a été multipliée par 20 en moins de 170 ans (Fig.(1a)). Elle a progressé de 7791 TWh en 1850 (soit 669 MTep) à 157603 TWh en 2018 (soit 13551 MTep) [Ritchie and Roser (2020b)]. En outre, l'émission de dioxyde de carbone a considérablement augmenté (Fig.(1b)). En 2018, elle est estimée à 37 milliard de tonnes, une multiplication par 188 depuis 1850 [Ritchie and Roser (2020a)], avec comme conséquence une augmentation de la température moyenne de 0.7°C par rapport à la référence sur la période 1961 – 1990 (29 ans). Avant cela, de 1850 – 1960 (110 ans) on ne constate qu'une augmentation de 0.4°C. Alors, globalement, on relève une augmentation moyenne de la température de 1.1 °C depuis 1850 (Fig.(1c)).

L'augmentation de la température moyenne du globe devrait bouleverser le climat et provoquer de graves conséquences sociales et écologiques. Depuis la conférence de l'Organisation des Nations Unies sur l'environnement à Stockholm en 1972, plusieurs conférences scientifiques et le protocole de Kyoto ont été mise en place pour affronter cette préoccupation. Et pour la première fois, 195 pays ont conclu le premier universel, légalement contraignant accord climatique mondial lors de la conférence de Paris sur le climat (COP21) en 2015. L'accord de Paris vise à réduire les émissions de gaz à effet de serre et à maintenir l'augmentation de la température moyenne mondiale en dessous de 2°C par rapport aux niveaux préindustriels [COP21 (2016)]. Ce défi est considérable et nécessite un changement social et technologique profond, surtout dans la réduction de la consommation énergétique et de l'émission de dioxyde de carbone.

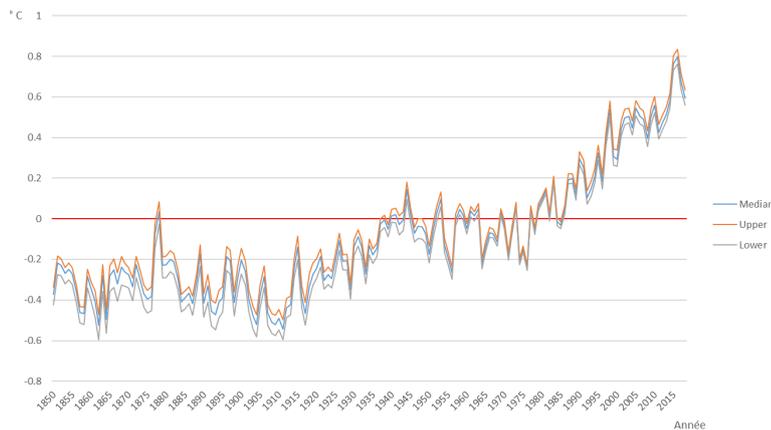
Le secteur industriel est l'un des plus grands consommateurs d'énergie mondiale et de source de l'émission de  $CO_2$ . La Fig.(2a) représente la consommation énergétique par secteurs et sa prévision jusqu'à 2040 en unité de "toe" (tonne d'équivalent pétrole en anglais). Nous trouvons que le secteur industriel, y compris les industries sans combustion de carburant, représente plus de la moitié de la consommation d'énergie mondiale. La Fig.(2b) montre l'émission de  $CO_2$  selon secteurs en 2017. Nous apercevons que le secteur industriel est responsable de 24% des émissions de dioxyde de carbone ( $CO_2$ ). En outre, la production d'électricité constitue la plus grande source d'émission, représentant 42% du to-



(a) La consommation d'énergie mondiale de 1850 à 2018 (Ritchie and Roser (2020b))



(b) L'émission mondiale de dioxyde de carbone de 1850 à 2018 (Ritchie and Roser (2020a))



(c) Anomalie de la température moyenne mondiale par rapport à la température moyenne de période 1961-1990 ( $^{\circ}$  C) de 1850 à 2018 (Ritchie and Roser (2020a))

Figure 1 – L'évolution mondiale de la consommation énergétique, de l'émission de dioxyde de carbone et de la température de 1850 à 2018

tal mondial. Sachant que le secteur industriel utilise presque 42% de l'électricité générée en 2017 [IEA (2019)], l'optimisation de la consommation d'électricité dans le secteur industriel joue un rôle indiscutable dans les mécanismes de réduction des émissions de  $CO_2$ .

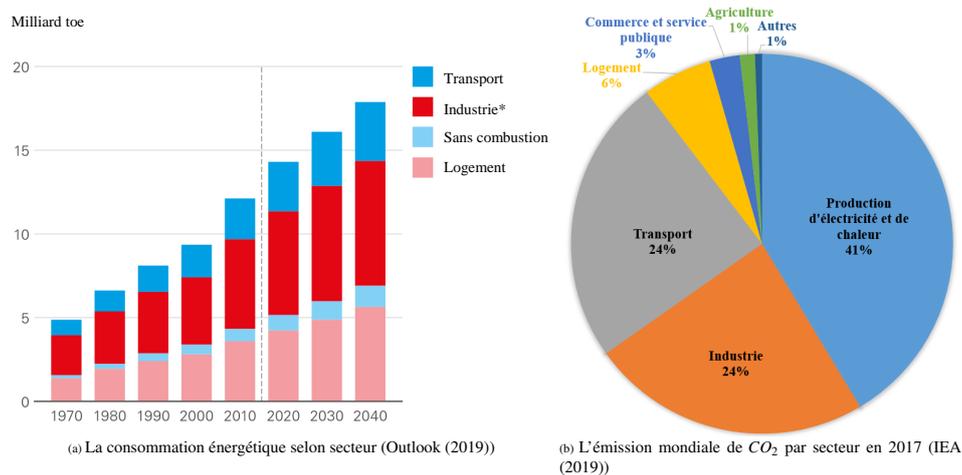


Figure 2 – La répartition par secteur de la consommation énergétique mondiale et de l'émission de carbone.

Pendant ces dernières années, l'optimisation d'énergie dans le secteur industriel, notamment dans le secteur manufacturier, a attiré de plus en plus d'attention. Ces études peuvent être classées en trois grandes perspectives : au niveau des produits, au niveau de la machine et au niveau du système de fabrication [Dai et al. (2013)]. Au niveau du produit, l'approche est fondée sur la reconception du produit pour la réduction de l'énergie intrinsèque qui est la quantité d'énergie requise au cours de son cycle de vie. Cette approche nécessite un logiciel de simulation performant pour l'analyse et l'évaluation de l'énergie intrinsèque du produit. Il en résulte un investissement en capital important qui n'est pas facilement applicable dans la plupart des entreprises manufacturières, en particulier dans les petites et moyennes entreprises.

De plus, le développement ou la conception d'une machine plus économe en énergie est une approche importante au niveau de la machine. De nombreuses études ont montré que la plupart de l'énergie est consommée par l'équipement et les fonctions auxiliaires de la machine telles que : système de refroidissement, pompe hydraulique, système de contrôle, etc. Les auteurs de [Gutowski et al. 2005] ont montré que 85.2 % d'énergie est consommée par les opérations hors usinage dans le cas de Toyota Motor. En étudiant la consommation d'énergie de la machine et de ses accessoires, [Drake et al. 2006] ont souligné que l'énergie requise pour l'opération de coupe ne prend que 19 % de la consommation totale d'énergie d'une fraiseuse CNC.

Au niveau du système de fabrication, la planification et l'ordonnancement sont

des voies très prometteuses pour les entreprises afin d'optimiser la consommation d'énergie. Cette approche ne nécessite pas d'investissements importants dans la reconception de la machine ni du produit, mais elle a un impact significatif sur les performances et l'optimisation énergétique du système de production. Avec régulation de la consommation d'électricité via le programme de réponse à la demande (DR), cette approche peut apporter des améliorations non négligeables sans investissements importants pour les entreprises manufacturières.

De plus, avec l'émergence du Smart Grid, la maîtrise de la demande en énergie ("Demand Side Management" (DSM) en anglais) est essentielle pour améliorer l'efficacité énergétique et optimiser la répartition de la puissance. La DSM comprend trois concepts : la réponse à la demande (DR), l'efficacité énergétique (EE) et la conservation de l'énergie (EC) [Dabur et al. (2012)]. Du point de vue industriel, l'efficacité énergétique (EE) et la réponse à la demande (DR) se traduiraient par une amélioration significative de l'utilisation de l'électricité dans la plupart des entreprises manufacturières. L'EE cherche à réduire la consommation d'énergie sans diminuer la productivité. Entre temps, la DR fait référence à l'évolution des modes de consommation des clients en réponse à la fluctuation des prix de l'électricité au fil du temps [Gong et al. (2016)].

La DR peut réduire les coûts et également les émissions de carbone associés à la consommation d'électricité pour les consommateurs industriels [Summerbell et al. (2017)]. Avec la présence de plus en plus importante des énergies renouvelables (éolienne, solaire, etc), l'impact carbone de la génération d'électricité varie significativement entre différents modes de production. Par exemple, pour générer un kilowatt-heure (kWh) d'électricité, une centrale à charbon émet 950 g de  $CO_2$  contre entre 60 et 150 g de  $CO_2$  des panneaux photovoltaïques [RTE (2019)]. L'intensité carbone de l'électricité produite varie également au fil du temps (Fig.(3)). Si la DR est bien appliquée, on déplacera une partie de la consommation d'énergie vers des heures ou des jours où la consommation globale est plus faible. Pendant ces périodes de faible consommation, les centrales électriques utilisent les modes de production dont l'intensité de carbone est plus faible (éolienne, solaire, etc).

En outre, depuis le début du millénaire, dans les pays industriels, les prix de l'électricité ne cessent d'augmenter. Ceci est le résultat des taxes et des impôts pour soutenir l'intégration des énergies renouvelables et l'abandon de l'électricité à faible coût produite par l'énergie nucléaire. Par conséquent, cela entraîne une baisse de la compétitivité par rapport aux pays où les prix de l'électricité augmentent moins rapidement. Il est alors indispensable d'étudier comment améliorer l'efficacité d'un système de production en termes de consommation d'électricité afin de réduire les coûts de production et l'impact environnemental.

Dans ce contexte, nous menons une étude de l'optimisation énergétique de l'ordonnancement dans un système de fabrication sous DR. A notre connaissance, l'une des tarification d'électricité les plus courantes de la DR est le tarif " Time of use " [Sharma et al. 2015]. Ce tarif prévoit des prix variables pour l'électricité d'une période à l'autre. C'est pourquoi il encourage les utilisateurs à gérer leur

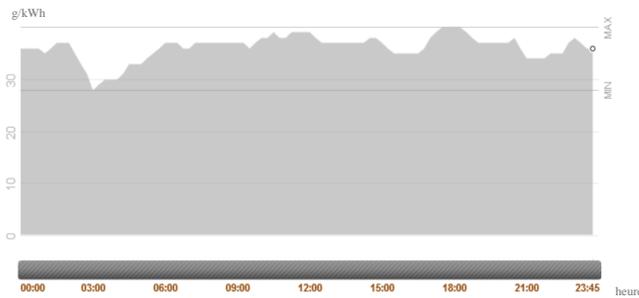


Figure 3 – Émissions de CO2 en grammes par kilowatt-heure générées par la production d'électricité en France le 23/10/2019.

consommation d'énergie pour réduire au maximum le coût de l'électricité. Comme l'ordonnancement d'atelier de type flow shop est un des problèmes les plus connus dans le domaine de l'ordonnancement, nos études se focalisent sur l'optimisation de coût d'électricité dans le problème d'ordonnancement de flow shop à deux machines.

## 1.2 Les notions préliminaires

Cette partie consiste à rappeler quelques notions de bases qui seront utiles lors de notre étude. D'abord nous rappelons quelques éléments relatifs aux problèmes d'ordonnancement. Nous commençons par la définition d'un problème d'ordonnancement. Ensuite nous rappelons des éléments fondamentaux d'un problème d'ordonnancement et les approches de résolutions. Enfin, nous rappelons également quelques concepts sur la "Demand Side Management" avec les tarifications d'électricité.

### 1.2.1 Introduction générale aux problèmes d'ordonnancement

#### 1.2.1.1 Définition

Dans une entreprise, nous avons trois types de décisions à prendre régulièrement pour assurer son bon fonctionnement. Les décisions au niveau stratégique agissent sur le long terme et gèrent l'organisation, les produits, les services et les gestions globales de l'entreprise. Au niveau tactique, les décisions agissent sur le moyen terme et s'appuient sur la prévision des demandes et la planification de la production. A court terme, nous trouvons les décisions opérationnelles qui sont prises de façon continue. Les décisions opérationnelles mettent l'accent sur l'organisation de la production à court terme.

L'ordonnancement des ordres de fabrication fait partie des activités principales à court terme dans l'industrie. Cette activité consiste à programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates

d'exécution [Chretienne et al. (1997)]. Le terme réalisation varie selon le champs d'application de l'ordonnancement considéré. En production, la réalisation concerne les opérations à effectuer sur des machines. Nous avons quatre notions fondamentales dans un problème d'ordonnancement de production : les tâches (jobs), les ressources, les contraintes et les objectifs :

- Une tâche est constituée d'une ou plusieurs opérations qui doivent être exécutées. Chaque tâche est localisée temporellement par une date de début et une date de fin.
- Une ressource est une machine ou un humain qui intervient dans l'exécution d'une tâche.
- Les contraintes sont les limites temporelles, technologiques ou de ressources à respecter lors de l'exécution des tâches.
- Les objectifs correspondent à des exigences qualitatives et quantitatives à optimiser en termes de temps, de ressources, de coûts permettant d'évaluer la qualité de l'ordonnancement établi.

Alors, un problème d'ordonnancement consiste à choisir pour chaque opération une date de début en respectant les contraintes du problème pour optimiser un ou plusieurs objectifs donnés. Une solution, encore appelé ordonnancement, est dite admissible ou faisable si toutes les contraintes du problème sont respectées.

### 1.2.1.2 Les éléments fondamentaux et leurs notations

Comme nous avons précisé auparavant, un problème d'ordonnancement est défini par quatre éléments : l'ensemble des tâches à réaliser, l'ensemble des machines disponibles pour les traiter, les contraintes à respecter relatives aux opérations et aux machines et ainsi que les fonctions objectifs à optimiser. Normalement, nous considérons un ensemble  $J = \{J_1, J_2, \dots, J_n\}$  de  $n$  tâches à traiter sur  $m$  machines  $\{m_1, m_2, \dots, m_m\}$  pour un problème d'ordonnancement. Nous utilisons principalement l'indice  $j$  pour désigner la tâche et l'indice  $i$  pour désigner la machine. Les notations suivantes sont fréquemment utilisées dans la majorité des problèmes.

Pour chaque tâche, nous avons souvent les données qui la caractérisent comme ci-dessous :

- Le temps d'exécution de la tâche  $j$  sur la machine  $i$  :  $p_{ij}$  (le "processing time" en anglais).
- La date de disponibilité de la tâche  $j$  :  $r_j$  (le "release date" en anglais). C'est la date d'arrivée de la tâche  $j$  dans le système, on peut également la considérer comme sa date de début au plus tôt.
- La date d'échéance de la tâche  $j$  :  $d_j$  ("due date" en anglais). C'est la date de fin préférentielle de la tâche  $j$ .

- Le poids d'importance relative de la tâche  $j$  :  $w_j$  ("weight" en anglais). Cela représente le facteur de priorité de la tâche  $j$  par rapport aux autres.

En fonction du nombre d'opérations nécessaires à l'exécution des tâches, nous pouvons distinguer les problèmes d'ordonnancement en deux catégories :

- Les problèmes où chaque tâche nécessite une seule opération : les problèmes d'ordonnancement de machine unique, de machines dédiées ou machines parallèles.
- Les problèmes où chaque tâche se compose de plusieurs opérations. Nous les appelons les problèmes d'atelier. Nous avons trois types d'atelier courant, "Open Shop", "Flow shop" et "Job Shop".

Pour identifier les différents problèmes d'ordonnancement, [Graham et al. (1979)] ont proposé une notation à trois champs distincts :  $\alpha|\beta|\gamma$ .

- Le champ  $\alpha$  désigne les caractéristiques de la ressource ou, autrement dit, l'environnement machine.
- Le champ  $\beta$  renseigne sur les contraintes liées aux opérations.
- Le champ  $\gamma$  indique le ou les critères à optimiser.

Généralement, l'environnement machine se compose de deux informations : la nature de l'environnement machine (Open Shop, Flow shop, etc) et le nombre de machines (pool de ressources). Le champ  $\alpha$  dans la notation de [Graham et al. (1979)] est composé alors de deux sous champs  $\alpha_1$  et  $\alpha_2$ .  $\alpha_1$  renseigne sur la nature de l'environnement machine, tandis que  $\alpha_2$  renseigne sur le nombre de machines.

Nous avons les différents environnements machine ci-dessous :

- Machine unique : le cas de base des problèmes d'ordonnancement dans lequel une seule machine traite les tâches. Dans la réalité, il existe très peu de systèmes de production à une seule machine. En revanche, leur étude permet de trouver des idées pour la résolution des problèmes plus compliqués. Dans l'étude de machine unique, le champ  $\alpha_1 = \emptyset$  ou absent.
- Machines parallèles : il existe  $m$  machines en parallèles. Dans ce type d'environnement, chaque tâche est composée d'une opération et peut être exécutée sur une des machines. En fonction de la relation générale entre les performances des machines, nous pouvons distinguer trois types de machines parallèles. Premièrement, si le temps de traitement d'une tâche est la même sur toutes les machines, les machines sont donc identiques,  $\alpha_1 = P$ . Deuxièmement, si les performances des machines sont proportionnelles, alors les machines sont uniformes,  $\alpha_1 = Q$ . Troisièmement, s'il n'y a pas de relation générale entre les performances des machines, alors les machines sont non reliées,  $\alpha_1 = R$ .

- Les ateliers à cheminement unique ou "flow shop" : un processus de fabrication en ligne formé par  $m$  machines en série. Les tâches à traiter visitent l'ensemble des machines dans le même ordre. Les temps de traitement sur chaque machine pour différentes tâches peuvent différer. Dans l'étude de flow shop,  $\alpha_1 = F$ .
- Les ateliers à cheminement multiples ou "job-shop": il dispose de  $m$  machines et que les ordres de visite des machines de chaque tâche sont différents. Dans l'étude de "job shop",  $\alpha_1 = J$ .
- Les ateliers à cheminement libre ou "open shop" : il dispose  $m$  machines, l'ordre de passage sur les machines est libre pour toutes les tâches. Il n'y a aucune relation de précédence entre les opérations d'une même tâche. Dans l'étude de "open shop",  $\alpha_1 = O$ .

Le second champ  $\beta$  renseigne les contraintes liées aux opérations qui spécifient les caractéristiques des tâches. Il existe plusieurs types de contraintes dans le problème d'ordonnancement. Nous ne citons que les plus courantes :

- Dates de disponibilité de tâche  $j$  :  $r_j$ . Ces contraintes signifient que les dates de disponibilité des tâches sont distinctes.
- La préemption est autorisée :  $pmtn$ . Si la préemption est autorisée, une machine peut interrompre à tout moment le traitement d'une opération pour en exécuter une autre. L'opération interrompue peut être reprise à tout moment.
- La précédence entre des tâches à respecter :  $prec$ . La précédence entre deux tâches  $j_1$  et  $j_2$  signifie que la tâche  $j_2$  ne peut commencer son exécution avant la fin d'exécution de la tâche  $j_1$ .
- "No-wait" contraintes :  $nwt$ . Les contraintes "No-wait" signifient que les opérations doivent s'enchaîner sans attente devant la machine.
- Les temps de traitement égaux :  $p_j = p$ . Si le champ  $\beta$  a la valeur  $p_j = p$ , toutes les opérations ont le même temps de traitement.

Le dernier champ  $\gamma$  renseigne sur les critères d'optimisation. Les critères d'optimisation permettent de qualifier des solutions. Il existe plusieurs critères pour évaluer la qualité des solutions. Les critères les plus courants sont la date de fin d'exécution d'une tâche, le temps de présence dans le système, l'écart par rapport au délai souhaité, le retard ou encore l'avance. Les notations pour ces critères sont données ci-dessous :

- La date de fin d'exécution de la tâche  $j$  sur la dernière machine :  $C_j$ .
- Le temps de présence de la tâche  $j$  dans l'atelier ("Flow time") :  $F_j = C_j - r_j$ .
- L'écart par rapport au délai souhaité ("Lateness") :  $L_j = C_j - d_j$ .

- Le retard (“Tardiness”) mesure le retard positif d’une tâche :  $T_j = \max\{0, L_j\} = \max\{0, C_j - d_j\}$ .
- L’avance (“Earliness”) est calculée si une tâche se termine avant le délai  $d_j$ , sinon elle est nulle :  $E_j = \max\{0, -L_j\}$ .

Un critère peut être régulier ou irrégulier : un critère est dit régulier si quand au moins un  $C_j$  varie, alors le critère varie également. Par exemple, les critères présentés auparavant sont tous réguliers.

### 1.2.1.3 Quelques notions sur la complexité des problèmes et des algorithmes

Les problèmes d’ordonnancement sont soit des problèmes de décisions, soit des problèmes d’optimisation. Les problèmes de décision ont comme solution "oui" ou "non". Un problème d’optimisation consiste en une étude monocritère dans laquelle on cherche à optimiser un critère, ou bien une étude multi-objectif dans laquelle on cherche le meilleur compromis entre différents critères. En fait, il est possible d’associer à tout problème d’optimisation un problème de décision. Par exemple, pour le problème d’optimisation  $F2||C_{max}$ , on peut l’associer au problème de décision : existe-t-il un ordonnancement tel que  $C_{max} \leq b$  ? La valeur  $b$  étant connue.

L’étude de la complexité a pour but d’apporter des informations sur la difficulté théorique à résoudre un problème. Cette étude permet de classer les problèmes selon leur difficulté théorique. Généralement, la complexité d’un problème consiste à évaluer la complexité algorithmique du meilleur algorithme résolvant le problème de façon optimale [Yalaoui et al. (2012)].

La complexité algorithmique est évaluée sur deux facteurs : le temps de calcul et la taille de la mémoire. Pour une instance de taille  $n$ , le temps de calcul d’un algorithme est évalué par le nombre d’opérations élémentaires  $T(n)$  nécessaires à l’exécution de l’algorithme. La taille de la mémoire d’un algorithme est la taille nécessaire pour stocker les différentes structures de données. La complexité algorithmique permet de mesurer les performances d’un algorithme et de le comparer avec d’autres algorithmes qui réalisent les mêmes fonctionnalités.

La complexité algorithmique peut avoir une des caractéristiques ci-dessous :

- Algorithme polynomial : s’il existe un polynôme  $Q(n)$  qui majore  $T(n)$  ( $T(n) \leq Q(n)$ ), alors on dit que l’algorithme est de complexité polynomiale  $O(Q(n))$ . De plus, si  $Q(n)$  est borné supérieurement par un polynôme d’ordre  $k$ , on dira alors que l’algorithme est polynomial d’ordre  $k$ , ( $T(n) = O(n^k)$ ).
- Algorithme pseudo-polynomial : s’il n’est pas polynomial mais s’il existe un polynôme  $Q(n, P)$  qui dépendent non seulement de  $n$  mais aussi d’une autre grandeur  $P$  caractéristique et que  $Q(n, P)$  majore  $T(n)$  ( $T(n) \leq Q(n, P)$ ).

- Algorithme exponentiel : s'il n'est ni polynomial, ni pseudo-polynomial, par exemple  $T(n) = 2^n$ .

En se fondant sur la complexité algorithmique, nous pouvons classer les problèmes en deux catégories principales : les problèmes polynomiaux (la classe  $P$ ) et les problèmes NP-difficiles (la classe  $NP$ ). Un problème est dit polynomial s'il existe un algorithme polynomial qui peut le résoudre de manière exacte. Les problèmes de décision que l'on ne peut pas résoudre optimalement par un algorithme polynomial sont dit NP-complets. Le problème d'optimisation dont le problème de décision est NP-complet sera alors qualifié de NP-difficile.

#### 1.2.1.4 Approches de résolution

Après avoir classé les problèmes d'ordonnement selon leur classe de complexité, nous nous intéressons aux méthodes de résolution. Dans la littérature, il existe plusieurs méthodes de résolution correspondant aux classes de complexité. Pour des problèmes de classe polynomiale, nous trouvons des algorithmes polynomiaux pour les résoudre de manière optimale. Par contre, pour des problèmes NP-difficiles, le temps de calcul augmente de manière considérable avec la taille du problème (exponentiellement). Nous avons deux alternatives possibles pour résoudre des problèmes NP-difficiles. Les méthodes exactes qui permettent de garantir l'optimalité de la solution du problème, mais ne peuvent traiter que des instances de tailles réduites en un temps de calcul raisonnable. Les méthodes approchées qui permettent de donner des solutions pour des instances de tailles beaucoup plus larges mais l'optimalité de la solution n'est pas garantie.

Les méthodes exactes les plus courantes pour résoudre les études des problèmes d'optimisation sont présentées ci-dessous :

- La programmation mathématique est une des premières étapes à réaliser avant de développer ou d'appliquer une méthode d'optimisation. Il s'agit de modéliser mathématiquement le problème sous forme d'une fonction de variables soumises à un ensemble de contraintes. Ces dernières sont sous formes d'équations ou d'inéquations qui délimitent l'espace des solutions à explorer. Dans le cas où les variables sont entières et les fonctions sont linéaires, nous avons une programmation linéaire en nombre entiers (PLNE). Si les variables sont à la fois entières et réelles, le modèle devient une programmation linéaire mixte en nombres entiers (PLNE mixte). Pour les PLNE, l'algorithme du simplexe est très efficace pour les résoudre dans la plupart des cas [Dantzig (1998)]. Généralement, il existe différents solveurs permettant d'obtenir la solution optimale du PLNE et du PLNE mixte comme CPLEX, Xpress, GAMS, etc.
- La programmation dynamique a été introduite la première fois par Bellman en 1957 [Bellman (1966)]. Cette méthode est appelée également optimisation récursive, son principe général est applicable à de nombreux problèmes

d'optimisation qui possèdent une certaine propriété dite de décomposabilité. Cela signifie qu'un problème de dimension  $n$  peut être décomposé en  $n$  sous-problèmes de dimension 1 à résoudre de façon séquentielle. A partir du problème initial de dimension  $n$ , la méthode génère  $n$  sous-problèmes ( $n$  étape de décision). L'utilisation de cette méthode nécessite une équation réursive qui permet d'estimer la valeur optimale du critère à une étape en fonction de sa valeur à l'étape précédente. Par récurrence, on peut déterminer la trajectoire optimale par le processus de décision consécutif. Selon le problème, ces méthodes peuvent avoir une complexité polynomiale ou exponentielle.

- La procédure par séparation et évaluation (PSE) est fondée sur une recherche arborescente qui énumère des sous-ensembles de solutions sous forme de noeuds. Comme l'énumération complète de toutes les solutions réalisables pour un problème d'optimisation n'est pas possible, alors la PSE n'explore que certains sous-ensembles de solutions dans lesquels la solution optimale peut se trouver. La PSE nécessite des propriétés de dominance, des bornes inférieures et supérieures pour pouvoir éliminer des sous-ensembles de mauvaises solutions. Pour évaluer chaque sous-ensemble, il faut calculer une borne inférieure et une borne supérieure du critère. L'application de la borne inférieure et de la borne supérieure permet d'avoir une idée des solutions qui se trouvent dans un sous-ensemble. Par exemple, on cherche à minimiser un critère donné, si la borne inférieure d'un noeud est supérieure à la meilleure borne supérieure trouvée jusqu'ici, alors il est certain qu'il n'existe pas de solutions meilleures dans ce sous-ensemble. Le branchement au niveau de ce noeud est coupé.
- La génération de colonnes repose sur la méthode de décomposition de Dantzig et Wolf [Dantzig and Wolfe (1960)]. En effet, les problèmes d'optimisation possèdent souvent un grand nombre de variables (colonnes). Mais dans la solution optimale, la plupart des variables sont très souvent nulles, ça veut dire qu'on prend un petit sous-ensemble de variables pour résoudre le problème. L'idée principale de cette méthode consiste à décomposer le modèle d'origine en plusieurs sous-problèmes plus faciles à résoudre. L'algorithme résout en premier temps le problème original (P) avec un sous-ensemble de colonnes (variables) de taille réduite. Puis, l'algorithme ajoute itérativement les nouvelles colonnes susceptibles d'améliorer la solution courante, jusqu'à atteindre l'optimalité.
- La décomposition de Benders utilise une stratégie de résolution de problèmes qui peut traiter des modèles de programmation comprenant plusieurs entiers à grande échelle [Benders (1962)]. Elle partitionne les variables d'un problème en deux vecteurs  $x$  et  $y$ . Premièrement, elle fixe les valeurs  $y$  en résolvant à l'optimalité un problème maître (MP) qui est un modèle linéaire mixtes en nombre entiers. Le MP est une relaxation du modèle global. En-

suite, les valeurs de  $y$  sont ensuite utilisées pour définir un sous-problème (SP) qui ne contient que  $x$ . La solution de SP peut révéler que les valeurs de  $y$  est infaisable, alors, dans ce cas, la solution sera utilisée pour générer une coupe Benders. La coupe Benders éliminera les valeurs inacceptables de  $y$  de l'ensemble de solutions. Puis, nous obtenons autre ensemble de valeurs de  $y$  en résolvant à nouveau le MP contenant ainsi toutes les coupes Benders générées. Nous résolvons de manière itérative le MP à l'optimalité et en utilisant la solution pour générer les sous-problèmes jusqu'à ce que le MP et le SP convergent.

Par rapport aux méthodes exactes qui se trouvent limitées lorsque la taille du problème augmente, des méthodes approchées sont capables de donner des solutions de qualité acceptable mais sans garantir de performance avec des temps de calcul beaucoup moins importants. En milieu industriel où le facteur temps est important, la rapidité de résolution des méthodes approchées les rend très utiles. Cependant ces méthodes approchées nécessitent d'être évaluées par rapport à la qualité de solution fournie. Deux facteurs importants sont à évaluer afin d'évaluer la qualité de ces méthodes : la déviation moyenne du critère par rapport à sa valeur optimale avec une base de plusieurs jeux d'instances et le temps de calcul en fonction de la taille du problème.

Les méthodes approchées sont également appelées les heuristiques. En se fondant sur le mécanisme de construction de la solution, nous pouvons distinguer des heuristiques en trois types. Premièrement, nous avons les heuristiques de construction qui génèrent une solution au problème sans remise en cause des décisions. Deuxièmement, nous avons les heuristiques de voisinage qui tentent d'améliorer des solutions initiales dans leur voisinage par des techniques de recherche locale. Troisièmement, nous avons les heuristiques à base de population de solutions qui explorent un espace de recherche de solutions très vaste.

Les métaheuristiques sont des cadres généraux des heuristiques. Les mécanismes de métaheuristiques sont génériques, elles ne nécessitent pas de connaissances particulières sur le problème optimisé pour fonctionner. Nous présentons ensuite les métaheuristiques les plus étudiées dans la littérature :

- Les méthodes de recherche locale ou métaheuristiques à base de voisinage qui commencent par une solution de départ. Pour améliorer le critère, elles passent d'une solution à une autre voisine par déplacements successifs. Dans le mécanisme de passage d'une solution à une autre, nous avons des méthodes de descente. Dans ces méthodes, une solution voisine doit améliorer le critère et s'il n'existe plus de meilleure solution dans le voisinage, elles s'arrêtent. Cependant, afin de pouvoir s'échapper des minima locaux, nous avons autres approches qui autorisent la dégradation du critère dans le but de pouvoir obtenir un optimum global tel que la méthode du recuit simulé [Kirkpatrick et al. (1983)], la méthode de recherche taboue [Glover (1989)] et [Glover (1990)], etc.

- Les métaheuristiques à base de populations qui sont connues également sous l'appellation d'algorithmes évolutionnistes. Ces méthodes recherchent dans une population de solutions plutôt que sur une solution unique. Les méthodes les plus connues sont les algorithmes génétiques (AG) développés à la fin des années 1950s par Holland [Dréo et al. (2003)]. Les AG sont inspirés par l'évolution biologique des espèces. Dans ces algorithmes, une population d'individus ou de solutions "parents" passent une séquence de recombinaison (opérateurs de croisement et mutation) pour former de nouveaux individus "enfants" ayant une empreinte génétique nouvelle héritée des parents. Ensuite, un schéma de sélection est mis en œuvre par une fonction d'évaluation "fitness". La fonction "fitness" mesure la force d'adaptation de chaque individu de la population pour que la prochaine génération ne contienne que les "meilleurs" individus. Après un certain nombre de générations, les programmes convergent vers le meilleur individu, ce qui représente la solution proposée par l'AG.

A part les heuristiques et métaheuristiques qui sont citées ci-dessus, il existe toutefois d'autres méthodes dans la littérature telles que SPT, EDD [Yalaoui et al. (2012)], la recherche à voisinage variable, la méthode "iterated local search" [Sevaux (2004)], l'algorithme de colonies de fourmis [Dorigo (1992)].

## **1.2.2 "Demand Side Management" et les tarifications d'électricité**

### **1.2.2.1 "Demand Side Management"**

Depuis son apparition dans les années 1970s, la "Demand Side Management" (DSM) joue un rôle crucial dans la réalisation d'une énergie durable qui vise à optimiser l'utilisation de l'énergie et à réduire les émissions. La DSM améliore la flexibilité dans le fonctionnement du système électrique et facilite la transition à faible émission de carbone dans la production d'électricité.

Si nous remontons le fil du temps jusqu'aux années 1970s, le choc pétrolier à l'époque a provoqué des changements profonds dans l'industrie de l'électricité [Gellings (1996)]. Face à la croissance de la demande en électricité, il existe deux stratégies différentes dans le fonctionnement du système électrique pour répondre à la demande de charge totale prévue : l'expansion de la production d'électricité et l'utilisation des mesures de gestion. La première vise à fournir une quantité d'énergie supplémentaire pour répondre à la demande de charge croissante par la modernisation ou le développement physique du système. La deuxième stratégie vise à maintenir le niveau de demande de charge d'électricité au fil du temps à travers la mise en œuvre de mesures managériales appropriées. Le concept de "Demand Side Management" (DSM) est né en se fondant sur le principe de la deuxième stratégie. En 1980, le terme "Demand Side Management" a été inventé par Clark W. Gellings [Meyabadi and Deihimi (2017)] :

La "Demand Side Management" est la planification et la mise en œuvre des activités de compagnie d'électricité conçues pour influencer l'utilisation de

l'électricité par les clients de manière à produire les changements souhaités dans la production d'électricité, c'est-à-dire dans le schéma temporel et l'ampleur de la charge d'électricité. Les activités de compagnie d'électricité dans la DSM comprennent la gestion de la charge, les nouvelles utilisations, la conservation stratégique, l'électrification, la génération de clients et les ajustements des parts de marché.

Les objectifs de la DSM sont principalement [Summerbell et al. (2017)] :

- Réduire la demande aux heures de pointe pour diminuer le besoin d'unités de production utilisées uniquement pour satisfaire la pointe.
- Encourager les consommateurs à utiliser plus d'électricité aux heures creuses au lieu d'autres sources d'énergie.
- Equilibrer la charge entre les heures de pointes et les heures creuses en déplaçant la demande de pointe hors des heures de pointe.

Pour atteindre ces objectifs, nous avons trois différentes stratégies de la DSM qui comprennent [Palensky and Dietrich (2011)] :

- L'efficacité énergétique (EE) qui réduit la demande d'électricité en rendant l'utilisation énergétique plus efficace. L'EE vise à minimiser la consommation en énergie pour un service rendu identique.
- La réponse à la demande (DR) fait varier le prix d'électricité afin d'inciter les consommateurs à basculer la consommation hors des heures de pointe.
- La réserve de l'inertie des machines tournantes de l'électricité hydraulique et thermique ("Spinning reserve") est la capacité d'augmentation du couple appliqué au rotor de la turbine des générateurs qui permet de fournir de la puissance supplémentaire en cas de besoin immédiate.

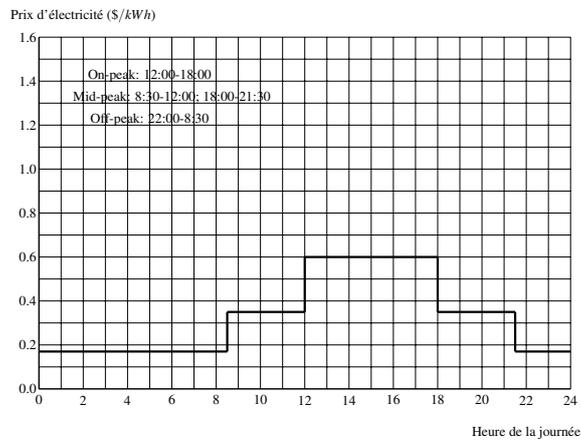
Dans la stratégie de la réponse à la demande, nous pouvons distinguer deux types de programmes : le pilotage par le prix ("price driven") et le pilotage par l'évènement ("event driven") [Goldman (2010)]. Dans la stratégie de pilotage par le prix, le coût de l'électricité varie en fonction du temps qui incite les industriels à planifier et organiser leurs activités de manière à minimiser le coût de l'électricité. Quant à la stratégie de pilotage par l'évènement, il existe des événements qui impactent la génération d'électricité, on les nomme des événements déclencheurs : des changements climatiques, adaptation à des perturbations économiques, dégradation de la fiabilité des équipements, ect. Si les industriels arrivent à réduire leurs consommations énergétique pour répondre à ces événements déclencheurs, des récompenses seront accordées.

Dans notre étude, nous mettons en avant deux stratégies de DSM : l'efficacité énergétique et la réponse à la demande. Nous présentons les tarifications les plus courantes de la réponse à la demande dans la section qui suit. Les tarifications les plus courantes de DR sont : le "Time-of-Use" (ToU), le "Critical Peak Pricing" (CPP) et le "Real Time Pricing" (RTP).

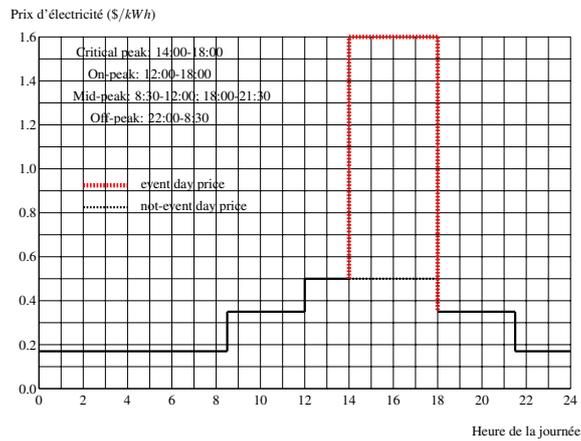
### 1.2.2.2 Les tarifications d'électricité :

Nous présentons ici les principales notions de tarifications : le "Time-of-Use" (ToU), le "Critical Peak Pricing" (CPP) et le "Real Time Pricing" (RTP) ci-dessous : [Newsham and Bowker (2010)].

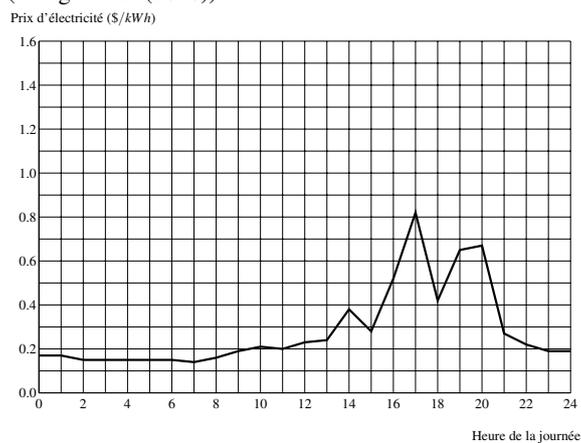
- "Time of Use" (ToU) : Dans cette tarification, la journée est divisée en plusieurs blocs de temps (heures). Le prix de l'électricité par unité de consommation varie selon les différents blocs de temps, mais pas à l'intérieur des blocs. Généralement, le ToU a trois types de blocs de temps : "peak", "semi-peak" et "off-peak". Le prix d'électricité est le plus élevé pour le bloc en période "peak". Le prix d'électricité est le moins élevé pour le bloc en période "off-peak". La différence offre aux utilisateurs la possibilité de minimiser le coût de l'électricité en attribuant des charges d'électricité à des périodes moins chères et disponibles. La Fig.(4a) représente un exemple de ToU de la "Pacific Gas & Electric Company".
- "Critical Peak Pricing" (CPP) : Les réseaux d'électricité parfois peuvent se trouver dans un instable état car ils nécessitent plus de puissance que sa capacité. Nous appelons ces périodes des "critical peak". Le concept de CPP est similaire à celui de ToU, sauf qu'il n'est appliqué que sur un nombre relativement restreint de jours "d'événement" où des périodes "critical peak" pourraient apparaître. Ces journées sont généralement annoncées par la compagnie d'électricité à l'avance, sur la base de leur prévision d'une demande particulièrement élevée. Le rapport entre le prix de "peak" et le prix de "off-peak" est plus élevé pendant les jours "d'événement" du CPP que dans un programme ToU. La Fig.(4b) représente un exemple de tarif CPP de "Pacific Gas & Electric Company".
- "Real Time Pricing" (RTP) : La tarification RTP permet d'adapter à la situation actuelle du réseau en considérant l'interaction dynamique entre l'offre et la demande. En plus, le tarif RTP facilite l'intégration des ressources d'énergies renouvelables dans le réseau. Avec l'apparition des "Smart Grid", la communication entre le fournisseur et les utilisateurs est renforcée par "Smart Meters". L'information de prix et de puissance consommée sont communiqués en temps réel. Le prix d'électricité qui varie toutes les heures ou toutes les unités de temps peut être communiqué aux clients un jour à l'avance ou une heure à l'avance. La Fig.(4c) représente un exemple de tarif RTP de "ComEd power company" à Chicago le 15/09/2019.



(a) Le tarif ToU de “Pacific Gas & Electric Company” (Wang and Li (2016))



(b) Le tarif CPP de “Pacific Gas & Electric Company” (Wang and Li (2016))



(c) Le tarif RTP de “ComEd power company” à Chicago, Etats-Unis le 15/09/2019 (ComED (2019))

Figure 4 – Les tarifications d’électricité de la réponse à la demande.

### 1.3 Problématique de cette étude

Dans cette partie, nous présentons la problématique de notre étude. Nous étudions le problème d'ordonnancement d'atelier de type flow shop à deux machines sous la tarification d'électricité de type ToU. Nous présentons tout d'abord la configuration de système de flow shop à deux machines. Ensuite, nous y intégrons l'aspect énergétique pour former la problématique étudiée.

Tout d'abord, l'atelier de type flow shop à deux machines est composé d'un ensemble  $M = 1, 2$  de machines disponibles qui traitent un ensemble de tâches  $J = \{1, 2, \dots, n\}$ . Chaque tâche  $J_j \in J$  a un temps de traitement  $p_{ij}$  sur la machine  $i \in M$ . Les tâches doivent être traitées de manière non préemptive d'abord sur la machine 1, puis sur la machine 2. À tout moment, chaque machine ne peut traiter qu'une seule tâche à la fois, et chaque tâche ne peut être traitée que sur une seule machine au maximum.

L'ordre de passage des tâches sur la machine 1 et la machine 2 sont similaires. Cet ordre est représenté par la relation de précédence entre les tâches ( $\prec$ ). Pour deux tâches quelconque  $J_j, J_{j'} \in J$ , on dit " $J_j \prec J_{j'}$ " si  $C_{ij} \leq S_{ij'}, i \in M$ , avec  $S_{ij}$  soit la date de début de tâche  $J_j$  sur la machine  $i$  et  $C_{ij}$  soit la date d'achèvement de tâche  $J_j$  sur la machine  $i$ .

La Fig.(5) présente un exemple de l'atelier de type flow shop à deux machines. Nous avons quatre tâches  $J_3 \prec J_1 \prec J_4 \prec J_2$  à traiter sur la machine 1 et la machine 2. L'ordre de passage des tâches sur la machine 1 et sur la machine 2 sont similaires.

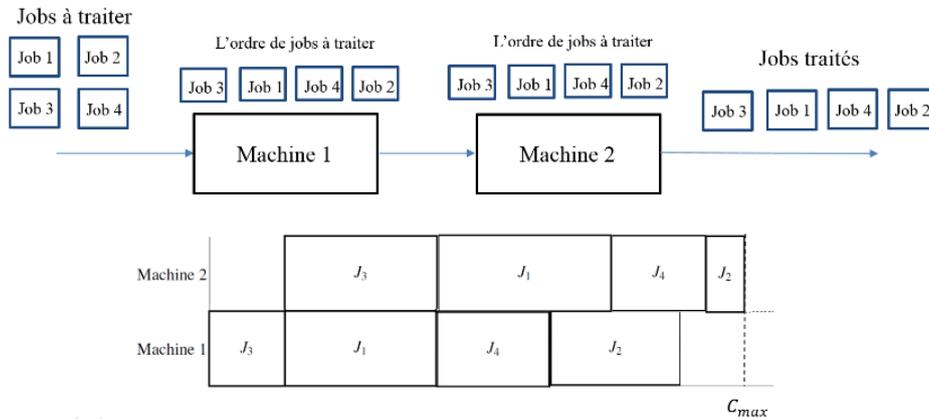
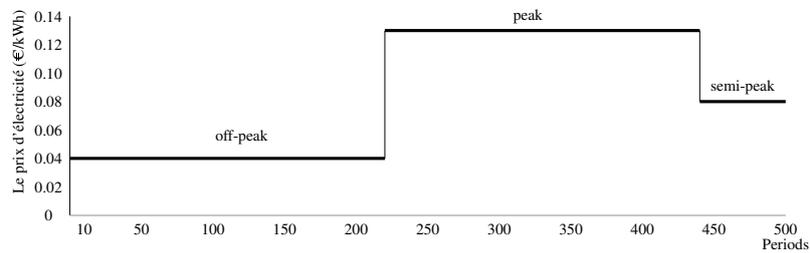


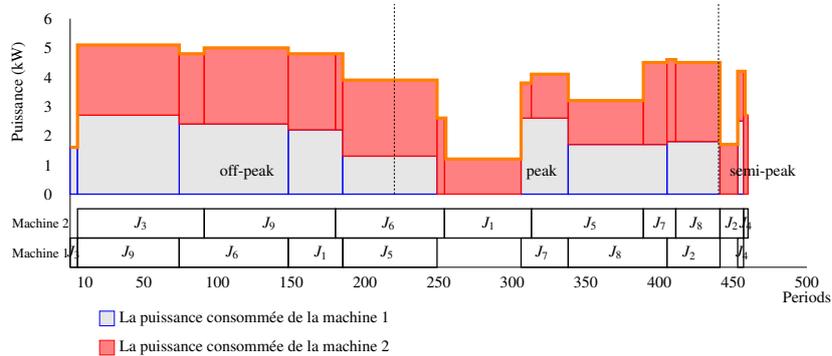
Figure 5 – Exemple de l'atelier de type flow shop à deux machines.

Dans cette thèse, l'atelier de type flow shop fonctionne sous le contexte de tarification d'électricité de ToU. Nos travaux portent sur la minimisation du coût d'électricité total de fonctionnement de ce système de flow shop à deux machines ( $F2|ToU|TEC$ ). Le coût d'électricité total est la multiplication entre le prix unitaire

d'électricité et la consommation d'électricité. Comme dans ToU, le prix unitaire d'électricité fluctue beaucoup d'une période à l'autre. C'est pourquoi nous nous intéressons à l'évolution de la consommation d'électricité du système au long de l'horizon de travail. La consommation d'électricité du système dépend de la caractéristique énergétique des tâches à traiter et de l'état de fonctionnement des machines. Nous cherchons donc à séquencer les tâches à traiter, nous déterminons également leurs dates de début et ainsi l'état de fonctionnement des machines pour que le système consomme moins d'énergie pendant les périodes de prix chers et inversement. La Fig.(6) donne un exemple de l'évolution de la consommation énergétique du système de flow shop à deux machines sous la tarification d'électricité ToU.



(a) La tarification d'électricité ToU



(b) L'évolution de la consommation énergétique du système

Figure 6 – L'évolution de la consommation d'électricité du système sous la tarification ToU.

L'étude de [Fang et al. (2016)] a démontré que le problème  $1|ToU|TEC$  est NP-difficile. Comme  $1|ToU|TEC$  est une version particulière de  $F2|ToU|TEC$  quand le temps de traitement de tous les tâches sur la machine 2 sont égaux à zéro, alors  $F2|ToU|TEC$  est NP-difficile. Nous traitons deux sous problèmes de  $F2|ToU|TEC$ . Le premier problème minimise le coût d'électricité total de la solution optimale de  $F2||C_{max}$  dont la consommation énergétique dépend de la caractéristique énergétique des tâches. Le deuxième problème minimise le coût d'électricité total de flow shop à deux machines avec la prise en compte de plusieurs états de fonctionnement de machines dont la consommation énergétique dépend de

l'état de fonctionnement de machine.

Ce manuscrit est constitué de cinq chapitres. Le premier chapitre présente le contexte, la tendance indiscutable de l'optimisation de la consommation énergétique dans l'industrie. Ce chapitre présente également les notions préliminaires du problème d'ordonnement et les tarifications d'électricité nécessaires pour la compréhension de notre étude.

Le second chapitre est consacré à l'analyse de l'existant et une synthèse bibliographique sur les différentes techniques d'optimisation d'énergie dans l'industrie et de l'ordonnement des systèmes de fabrications. Nous présentons les études de deux principaux axes de "Demand Side Management" : l'efficacité énergétique et la réponse à la demande.

Dans le troisième chapitre, nous étudions le problème de minimisation du coût d'électricité total de la solution optimale de problème d'ordonnement de flow shop à deux machines sous ToU. Cette étude repose sur la théorie des solutions équivalentes de  $F2||C_{max}$  qui sera appliquée pour la première fois dans l'optimisation de coût d'électricité. Des expérimentations numériques permettant d'évaluer les résultats de ces méthodes heuristiques sont présentées et des conclusions clôturent ce chapitre.

Puis, le quatrième chapitre est consacré au problème de minimisation de coût d'électricité de flow shop à deux machines avec la prise en compte de plusieurs états de fonctionnement des machines. Tout d'abord, nous présentons les cas optimaux pour certaines structures spécifiques de prix *ToU*. Deuxièmement, nous développons une proposition fondée sur la règle de Johnson pour déterminer la séquence optimale pour les structures de prix générales de *ToU*. Troisièmement, nous proposons une nouvelle programmation linéaire en nombres entiers mixtes intégrant la proposition. Ensuite, nous concevons une méthode exacte fondée sur la «décomposition de Benders» pour résoudre le problème. Enfin, les tests numériques montrent que l'approche proposée améliore d'une manière significative la qualité des résultats existants dans la littérature.

Nous clôturons ce manuscrit par une conclusion générale qui présente le bilan des travaux réalisés au cours de cette thèse. Des perspectives liées à nos travaux de recherche sont également mises en évidence.

## 2 L'état de l'art

Dans ce chapitre, nous présentons des études existantes dans la littérature sur les différentes techniques d'optimisation d'énergie dans l'ordonnement des systèmes de fabrications. L'une des premières études en ordonnancement qui prend en compte la consommation énergétique dans la fabrication est l'étude de [Subai et al. (2006)]. Dans ce travail, les auteurs ont étudié un problème d'ordonnement des mouvements du robot dans une ligne de traitement de surface. Ils ont considéré le coût de la consommation d'énergie, les critères environnementaux avec autres critères d'ordonnement pour caractériser une fonction de coût global non linéaire.

Depuis [Subai et al. (2006)], les chercheurs ont développé plusieurs méthodes d'optimisation énergétique dans le système de production. La "Demand Side Management" (DSM) a été développée en réponse à cette tendance. Dans cette section, nous présentons des études relatives à deux axes principaux de DSM pour l'optimisation énergétique dans un problème d'ordonnement : l'efficacité énergétique et la réponse à la demande.

### 2.1 L'efficacité énergétique ("Energy Efficiency")

L'efficacité énergétique (EE) regroupe des études qui mettent en avant la transparence de l'utilisation d'énergie des machines. Ensuite, l'EE cherche des cadres pour réduire la consommation d'énergie sans diminuer les performances de production [Gong et al. (2016)]. Dans cette partie, nous parlons des méthodes et des techniques dans la littérature concernant l'EE.

#### 2.1.1 La consommation énergétique liée à l'état de fonctionnement de machine

L'étude de [Mouzon et al. (2007)] distingue la consommation énergétique d'une machine en deux types. Le premier type est la consommation énergétique liée au traitement des tâches (PE). Le deuxième type est la consommation énergétique non liée au traitement des tâches (NPE), par exemple, une machine peut consommer une quantité significative d'énergie quand elle est en idle. Dans plusieurs ateliers, les machines restent en idle environ 16% du temps d'ouverture. Etablir un plan d'ordonnement efficace qui réduit le temps de non activité des machines peut réduire considérablement la consommation énergétique de type NPE.

[Liu et al. (2014b)] se sont concentrés sur l'environnement de job shop qui est largement utilisé dans l'industrie manufacturière. Ils ont proposé un modèle pour le problème d'ordonnement bi-objectif qui minimise la consommation énergétique totale et le retard pondéré total. La consommation énergétique totale est minimisée en réduisant la consommation énergétique des machines en idle qui est liée fortement au plan d'ordonnement établi. Ils ont développé un algorithme génétique multi-objectif pour obtenir le front de Pareto.

### 2.1.2 La stratégie “Turn off/ Turn on” de la machine

En étudiant le comportement énergétique de la machine, la stratégie “Turn off /Turn on” de la machine est apparue pour réduire la consommation d’énergie dans l’ordonnancement de la production. Les auteurs [Mouzon et al. (2007)] ont trouvé qu’on pourrait économiser une quantité importante d’énergie de type NPE en arrêtant la machine si elle restait en idle pendant une longue période. Mais il faut prendre conscience que chaque fois qu’une machine est démarrée, elle consomme également une quantité importante d’énergie [Drake et al. (2006)]. Le principe de cette technique est que si une machine reste en idle pendant une période assez longue, jusqu’à ce que la consommation énergétique en idle devienne supérieure à celle de son redémarrage, l’arrêt et le redémarrage de cette machine est plus avantageuse pour minimiser la consommation énergétique.

Dans de nombreux ateliers, les machines non-goulet restent souvent en idle. Cette technique permet de déterminer si nous laissons ces machines tourner à vide, ou bien si nous les arrêtons pendant un certain temps pour minimiser la consommation énergétique. Les auteurs [Mouzon et al. (2007)] ont proposé plusieurs règles d’ordonnancement pour établir un plan efficace de “Turn off/ Turn on” de la machine fondé sur la prédiction des arrivées des tâches afin de réduire la consommation énergétique. Les auteurs [Yildirim and Mouzon (2011)] ont présenté un modèle mathématique pour minimiser à la fois le makespan et la consommation énergétique totale. Cette dernière peut être réduite en arrêtant la machine plutôt que de la laisser en idle. Il a proposé un algorithme génétique multi-objectif avec les règles de dominance.

Les auteurs [Dai et al. (2013)] ont étudié la stratégie de “Turn off/ Turn on” de la machine dans le problème d’ordonnancement d’atelier de type flow shop. Ils ont utilisé l’algorithme de recuit simulé en conjonction avec l’algorithme génétique pour minimiser le makespan et la consommation d’énergie du système. Les auteurs [Liu et al. (2016)] ont étudié la même stratégie dans le problème d’ordonnancement d’un atelier de type job shop dans le but de minimiser à la fois la consommation totale d’énergie et le retard total pondéré. Les auteurs [May et al. (2015)] ont examiné l’interaction entre le makespan et le coût total de l’électricité dans un problème d’ordonnancement d’atelier de type job shop. Ils ont proposé un nouvel algorithme génétique qui a été construit en fusionnant et en améliorant deux des algorithmes génétiques les plus connus dans la littérature, NSGA-II et SPEA-II.

### 2.1.3 Le choix de vitesse de traitement

Dans certains ateliers, les machines et les appareils ne peuvent pas être éteints pendant le processus de fabrication, mais ils peuvent traiter des tâches à différents niveaux de vitesse. Chaque niveau de vitesse consomme une quantité d’énergie différente. De manière générale, plus la vitesse de traitement est rapide, plus la consommation énergétique de la machine est grande. Alors le changement de vitesse de traitement de tâches devient une technique pour minimiser la consommation

énergétique.

Le travail [Fang et al. (2013)] est l'une des premières études sur le choix de la vitesse de traitement dans le problème d'ordonnancement de système de fabrication. Ils ont étudié le problème de l'ordonnancement des ateliers de type flow shop de permutation avec une restriction de la consommation électrique maximale. Ils ont considéré les cas où les vitesses sont discrètes et les cas où les vitesses sont continues. Dans les cas de vitesses continues, la consommation d'énergie est une fonction exponentielle des vitesses. Ils ont montré que ces problèmes sont équivalents à des cas particuliers du problème de voyageur de commerce asymétriques. Les auteurs [Fang and Lin (2013)] ont étudié un problème d'ordonnancement des machines parallèles où les vitesses de la machine peuvent être ajustées. Ils ont minimisé une somme pondérée de la pénalité totale des retards de tâche et du coût total de l'énergie. Ils ont proposé une méta-heuristique de l'optimisation par essaim de particules et deux heuristiques fondées sur la règle EDD (earliest due date) et la règle WSPT (Weighted Shortest Processing Time).

Les auteurs [Mansouri et al. (2016)] ont considéré un problème d'ordonnancement d'un atelier de type flow shop de permutation à deux machines avec des temps de réglages dépendants de la séquence et où les machines ont une vitesse variable. Ils ont proposé un modèle mathématique intégrant le choix de vitesse des machines pour étudier le compromis entre le makespan et la consommation énergétique sous le contexte de temps de traitement variable en fonction de la vitesse. Ils ont également développé une heuristique constructive pour résoudre les problèmes de grandes tailles dans un délai acceptable. Les auteurs [Zhang and Chiong (2016)] ont étudié le problème d'ordonnancement des ateliers de type job shop en minimisant deux objectifs : le retard pondéré total et la consommation énergétique totale. Dans leur étude, la vitesse de traitement de chaque machine peut être sélectionnée à partir d'un ensemble fini et discret. Deux stratégies d'amélioration spécifiques au problème sont proposées. La première améliore le retard pondéré total sous des vitesses de machine fixes. La seconde minimise la consommation totale d'énergie selon un plan d'ordonnancement fixé. Ils ont proposé un algorithme génétique intégrant ces stratégies de recherche locales pour atteindre un meilleur équilibre entre exploitation et exploration.

Les auteurs [Zhang et al. (2019)] ont résolu un problème d'ordonnancement écologique d'un atelier de type flow shop hybride dont des vitesses de traitement de machine sont variables. Ils ont minimisé le makespan et la consommation énergétique totale par un algorithme de colonie d'abeilles artificielles discrètes et multi-objectifs. Dans cette étude, la consommation totale d'énergie comprend la consommation d'énergie lorsque la machine est à l'état de traitement, à l'état de réglage et en idle. Les auteurs [Liu et al. (2017)] ont considéré un problème d'ordonnancement d'atelier de type flow shop. Le système compose d'une série d'étapes de traitement et d'une étape finale de contrôle de la qualité. À chaque étape de traitement, la vitesse de traitement de la machine est choisie à partir d'un ensemble discret. Une vitesse plus élevée entraîne un temps de traitement plus court mais un risque plus élevé de dégradation de la qualité du produit. A l'étape

du contrôle de la qualité, les produits de mauvaise qualité seront réintégrés dans le système pour le retraitement. Dans cette étude, la consommation énergétique est associée à la qualité du produit, à la vitesse de machine et à l'état de fonctionnement de machine. Ils ont proposé une approche de résolution en trois étapes pour résoudre le problème d'ordonnancement éco-énergétique proposé avec la minimisation de la consommation énergétique totale.

En plus du choix de vitesse de traitement de la machine, nous avons également dans la littérature des études sur le choix de vitesse de tapis de transport dans l'atelier. La logique est similaire à celle de vitesse de traitement de la machine. Plus la vitesse de tapis est élevée, plus la consommation énergétique est grande.

Les auteurs [Jiang and Wang (2019)] ont pris en compte le temps de réglage et de transport dans un problème d'ordonnancement bi-objectif écoénergétique de l'atelier de type flow shop. Dans cette étude, le temps de réglage dépend de la séquence et le temps de transport est contrôlable en augmentant ou en diminuant la vitesse du tapis roulant. Ils ont proposé un algorithme évolutif multi-objectif amélioré fondé sur une stratégie de décomposition. L'approche a décomposé le problème en plusieurs sous-problèmes afin de minimiser le makespan et la consommation énergétique totale.

Les auteurs [Lu et al. (2017)] ont étudié un problème d'ordonnancement de l'atelier de type flow shop. Ils ont considéré un exemple réel d'une entreprise de fabrication, dans lequel ils ont considéré le temps de réglage dépendant de la séquence et un temps de transport contrôlable. Ils ont proposé un nouveau modèle mathématique multi-objectif prenant en compte à la fois le makespan et la consommation énergétique totale.

#### **2.1.4 Le choix de machine de traitement**

L'autre technique pour minimiser la consommation énergétique est le choix de machine de traitement alternative ("Process Route Selection"). Cette technique ne fonctionne que dans les ateliers ayant des machines alternatives avec des caractéristiques énergétiques différentes pour le traitement d'une même tâche, par exemple les ateliers de machines en parallèles, les flow shop hybrides [Liu et al. (2014b)].

Les auteurs [Ji et al. (2013)] ont étudié un problème d'ordonnancement sur des machines parallèles uniformes où l'objectif est de minimiser la consommation de ressources de telle sorte que le makespan ne dépasse pas un certain niveau. Ils ont démontré que le problème est NP-difficile. Ensuite, ils ont proposé une heuristique et une optimisation par essaim de particules pour résoudre le problème. Les auteurs [Jiang et al. (2014)] ont considéré un problème d'ordonnancement multi-objectif dans un job shop flexible dans lequel ils ont optimisé simultanément le makespan, le coût de production, la consommation énergétique, et la qualité de produits. Ils ont amélioré un NSGA-II pour éviter une convergence prématurée vers une solution optimale locale en déterminant le taux de mutation en fonction de ressemblance entre chromosomes.

Les auteurs [Jia et al. (2019)] ont étudié un problème d'ordonnancement multi-

objectif des machines de traitement par lots en parallèle. Ils ont proposé un algorithme d'optimisation des colonies de fourmis pour minimiser le makespan et la consommation énergétique totale. Les auteurs [Liang et al. (2019)] ont considéré un problème de planification et d'ordonnancement de la production à capacité finie avec le temps de réglage dépendant de la séquence pour une ligne de production continue. Ils ont minimisé les coûts totaux qui comprennent les coûts de stockage, les coûts de réglage et les coûts énergétiques de la production. Ils ont développé un modèle de programmation linéaire en nombres mixtes et une heuristique pour résoudre le problème.

### 2.1.5 L'émission de $CO_2$

Dans ces études, nous minimisons l'émission de dioxyde de carbone ou l'empreinte carbone liée à la consommation énergétique dans le système de production. La quantité de dioxyde de carbone transmise est convertie à partir de la consommation énergétique par un facteur d'émission (en équivalent  $kg$  en  $CO_2$  pour le  $kWh$  électrique).

Les auteurs [Fang et al. (2011)] ont présenté un nouveau modèle mathématique du problème d'ordonnancement de l'atelier de type flow shop qui prend en compte la puissance consommée maximale, la consommation d'énergie et l'empreinte carbone. Dans leur étude, la vitesse de machine est une variable qui peut varier pour réduire la puissance consommée maximale à un instant et la consommation énergétique totale. Les auteurs [Liu and Huang (2014)] ont étudié l'empreinte carbone dans le problème d'ordonnancement dans un processus de traitement par lot. Ils ont considéré deux problèmes multi-objectif. Le premier est l'ordonnancement sur une machine de traitement par lot qui minimise simultanément le retard total pondéré et l'empreinte carbone. Le second est l'ordonnancement d'un système de flow shop hybride composé d'une machine de traitement par lot et ensuite deux machines en parallèle. La vitesse de traitement de deux machines en parallèle peut varier pour économiser la consommation énergétique. Le problème vise à minimiser simultanément le retard total pondéré, l'empreinte carbone et la puissance consommée maximale. Ils ont proposé un algorithme génétique multi-objectif adaptatif et un NSGA-II pour construire le front de Pareto de ces deux problèmes.

Les auteurs [Liu et al. (2014a)] ont considéré le problème d'ordonnancement sur une seule machine avec la minimisation simultanément du temps d'achèvement total et de l'émission de dioxyde de carbone liée à la consommation énergétique. Dans leur étude, la date de disponibilité des tâches est donnée, ils ont appliqué également la stratégie "Turn off/Turn on" machine pour réduire l'émission de  $CO_2$  liée à la consommation énergétique des machines en idle. Ils ont proposé un algorithme génétique multi-objectif (NSGA-II) pour la résolution. Les auteurs [Liu (2014)] ont développé un " $\epsilon$ -archived genetic algorithm" pour examiner deux problèmes d'ordonnancement de machine de traitement par lot avec l'objectif de minimiser simultanément le retard pondéré total et l'émission de  $CO_2$  liée à la consommation énergétique.

Les auteurs [Li et al. (2015)] ont étudié le compromis entre le makespan, la consommation énergétique maximale et l’empreinte carbone associée à l’utilisation énergétique dans le problème d’ordonnement de l’atelier de type flow shop. Les auteurs [Lei and Guo (2015)] ont étudié un problème d’ordonnement d’atelier de type job shop avec la prise en compte de contraintes de ressources. Les ressources considérées sont la main d’œuvre hétérogène. Le temps de traitement d’une tâche est variable entre les opérateurs et est représentée par un intervalle. Une méthode d’optimisation lexicographique est appliquée pour minimiser d’abord le makespan et ensuite l’empreinte carbone.

### **2.1.6 La consommation énergétique liée à la fiabilité du système**

Dans certains ateliers, les opérations de maintenance sont également une source de consommation énergétique. Dans la littérature, ils existent des études qui relient l’aspect de la consommation énergétique avec la fiabilité du système dans les problèmes d’ordonnement.

Les auteurs [Mokhtari and Hasani (2017)] ont étudié un problème d’ordonnement dans un atelier de job shop flexible avec trois fonctions objectifs : minimiser le temps d’achèvement total, maximiser la fiabilité totale du système et minimiser le coût énergétique total. Le coût énergétique total comprend les coûts énergétiques pour des opérations de production et ceux des opérations de maintenance. Dans leur étude, il existe un ensemble d’opérations de maintenance préventive. Chaque opération a un taux de réparation et un coût énergétique spécifique. Ils ont proposé un algorithme génétique combiné à un algorithme de recuit simulé pour résoudre le problème.

Les auteurs [Chen et al. (2019)] ont étudié un problème d’ordonnement sur une seule machine avec des contraintes de fiabilité de la machine. Dans leur étude, ils ont proposé un modèle de programmation mathématique qui montrait la relation entre la fiabilité et la consommation d’énergie de la machine. Ils ont ensuite présenté un algorithme d’optimisation des colonies de fourmis hybridés avec des règles d’Emmons modifiées pour minimiser les coûts de retard et les coûts énergétiques.

### **2.1.7 La consommation énergétique différente entre des tâches**

Dans ces études, la consommation énergétique dépend des caractéristiques des tâches. Sur la même machine, la consommation énergétique de traitement d’une tâche est différente de celle de traitement d’une autre.

Les auteurs [Tang et al. (2016)] ont étudié le problème d’ordonnement multi-objectif de l’atelier de type flow shop flexible qui minimise simultanément la consommation énergétique totale et le makespan. Dans leur étude, chaque tâche exige une vitesse de traitement spécifique qui entraîne une différence dans la consommation énergétique de la machine entre les tâches. Ils ont considéré le cas de l’ordonnement dynamique dans lequel des nouvelles tâches peuvent arriver à

n'importe quel moment. Ils ont proposé l'optimisation multi-objectifs par essais particuliers pour établir un front de Pareto.

Les auteurs [Rager et al. (2015)] ont présenté un problème d'ordonnancement dans un atelier de machines parallèles avec minimisation de la somme des écarts au carré de la consommation énergétique actuelle avec un niveau souhaité. Dans leur étude, le traitement de chaque opération nécessite une quantité énergétique différente. Ils ont développé un algorithme génétique et deux algorithmes mémétiques pour résoudre le problème.

### **2.1.8 Les contraintes énergétiques**

Dans la littérature, il y a des travaux qui s'intéressent aux contraintes énergétiques dans des problèmes d'ordonnancement. Dans ces études, les auteurs ont considéré une contrainte de capacité totale de l'électricité consommée par la machine à chaque période ou sur l'horizon.

Les auteurs [Bruzzone et al. (2012)] ont étudié un problème d'ordonnancement d'atelier de type flow shop flexible qui prend en compte la minimisation de la somme pondérée entre le retard et le makespan sans violation de la puissance énergétique maximale à chaque période. Les auteurs [Rozycki and Weglarz (2012)] ont considéré un problème d'ordonnancement préemptif de l'atelier des machines parallèles où le makespan est borné. Chaque tâche est décrite par une fonction continue et strictement concave qui relie le taux de vitesse de traitement à la puissance énergétique allouée à chaque instant. Dans leur étude, il faut respecter la limite de la puissance énergétique à chaque instant et respecter également la limite de la consommation énergétique totale de l'horizon de travail.

Les auteurs [Dupty et al. (2016)] ont examiné divers problèmes d'ordonnancement d'atelier à machines parallèle soumis à des contraintes de ressources qui imposent la restriction selon laquelle toutes les machines ne peuvent pas être fonctionnées en même temps. Il peut s'agir de contraintes de la puissance d'énergie maximale. Dans le contexte où il y a une contrainte de puissance, ils ont montré que le problème de la minimisation de makespan dans lequel les tâches peuvent être divisées en plusieurs parties est NP-difficile. Le même problème mais avec la minimisation de la consommation énergétique avec des contraintes de puissance maximale est également NP-difficile. L'ordonnancement avec des contraintes portant sur la limite de la consommation énergétique totale avec des tâches qui peuvent être divisées en plusieurs parties peut être trouvé en temps polynomial, et les problèmes avec les tâches qui ne peuvent pas être divisées en plusieurs parties sont NP-difficiles.

## **2.2 La réponse à la demande (“Demand Response”)**

L'évolution rapide des technologies de comptage, de communication et de contrôle motive de plus en plus de fournisseurs d'électricité à mettre en application une tarification d'électricité variable. Cette tendance permet d'équilibrer l'offre

et la demande d'électricité afin d'améliorer la fiabilité et l'efficacité des réseaux électriques [Braithwait et al. (2007)]. Nous présentons ici la variation de prix d'électricité en fonction du temps dans le cadre de la production industrielle. La dernière provoque des impacts significatifs dans le coût de production total de n'importe quel système de fabrication.

La réponse à la demande pour l'optimisation énergétique tient compte du prix volatile de l'électricité. Il encourage le client à faire passer sa consommation d'électricité de la période de pointe à la période creuse pour bénéficier d'un prix inférieur. Par conséquent, les industriels planifient et organisent leurs activités de manière à minimiser le coût de l'électricité. "Time of Use", "Critical Peak Pricing" et "Real Time Pricing" sont les exemples les plus courants pour ce type de programme. Après avoir examiné certaines des études dont les objectifs sont la réduction de la consommation d'énergie auparavant, nous considérons dans cette partie les études de l'ordonnement de système de fabrication vis-à-vis la variation de prix de l'électricité .

### 2.2.1 "Time of Use"

Les auteurs [Fang et al. (2016)] ont examiné le problème d'ordonnement sur une seule machine afin de minimiser le coût total de l'électricité en considérant la tarification ToU. Ils ont pris en compte deux cas : le premier cas avec la machine à vitesse uniforme et le deuxième cas avec la machine à une vitesse variable. Ils ont montré que, dans les deux cas, le problème est NP-difficile, mais la version autorisant la préemption est polynomiale. D'autre part, pour le problème à vitesse uniforme, si toutes les tâches ont le même temps de traitement et que les prix de l'électricité sont de structure pyramidale, le problème peut être résolu en temps polynomial.

Les auteurs [Aghelinejad et al. (2018b)] ont étudié les problèmes d'ordonnement sur une seule machine où la machine a plusieurs états de fonctionnement et chaque état consomme une quantité d'énergie différente. Ils ont prouvé que lorsque la séquence des tâches est prédéterminée, le cas de vitesse uniforme est polynomial et le cas de vitesse variable est pseudo-polynomial. Les auteurs [Aghelinejad et al. (2018a)] ont proposé deux modèles mathématiques pour le problème d'ordonnement non préemptif sur une seule machine avec l'objectif de minimisation des coûts énergétiques totaux. Le premier modèle traite un problème d'ordonnement au niveau de la machine où la séquence des tâches est prédéterminée. Le deuxième modèle se concentre sur la généralisation du précédent où l'hypothèse de séquence fixe prédéterminée initiale est levée. Ils ont proposé une heuristique et un algorithme génétique pour fournir de bonnes solutions dans un temps de calcul raisonnable.

Les auteurs [Chen et al. (2018)] ont minimisé la somme pondérée des temps d'achèvement et du coût de l'électricité pour un problème d'ordonnement préemptif sur une seule machine sous le tarif d'électricité ToU. Ils ont résolu la version non pondérée à l'optimalité par des algorithmes polynomiaux. La version

pondérée est NP-difficile et ils ont proposé un schéma d'approximation du temps polynomial pour la version pondérée. Ils ont également démontré que le problème de minimisation de la somme entre le makespan et le coût total de l'électricité sur des machines parallèles non liées peut être résolu en temps polynomial. Les auteurs [Chen and Zhang (2019)] ont étudié les problèmes d'ordonnancement sur une seule machine sous différents tarifs ToU. Ils ont minimisé le coût énergétique total tout en garantissant un niveau minimal de performances de production. Dans leur étude, un ordonnancement réalisable doit satisfaire à l'un des critères de production réguliers tels que : la date limite, le retard borné, le temps de présence borné et le temps d'achèvement total borné. Ils ont établi la traçabilité informatique du problème. Dans la structure générale des coûts de type ToU, le problème est NP-difficile. Cependant, il existe des structures spéciales de coûts de ToU pour lesquelles des algorithmes efficaces existent pour rendre le problème polynomial.

Concernant l'ordonnancement dans les ateliers, les études sous la tarification d'électricité ToU sont plutôt clairsemées. Les auteurs [Ding et al. (2015)] ont considéré un problème d'ordonnancement des machines parallèles non liées. Ils ont minimisé le coût total de l'électricité sous différentes structures de ToU. Pour résoudre ce problème, ils ont proposé une nouvelle programmation linéaire en nombres mixtes fondée sur des intervalles de temps. Ensuite, ils ont utilisé la décomposition de Dantzig-Wolfe pour réduire le problème et ils ont présenté une approche de résolution fondée sur la génération de colonnes. Ils ont également analysé le compromis entre le makespan et le coût total de l'électricité via les frontières de Pareto obtenues.

Les auteurs [Wang et al. (2018)] ont minimisé le coût total de l'électricité pour un problème d'ordonnancement de l'atelier de type flow shop de permutation à deux machines sous ToU. Premièrement, ils ont modélisé le problème par une programmation linéaire en nombres entiers mixtes. Deuxièmement, ils ont développé une programmation dynamique pour trouver un ordonnancement optimal en temps polynomial lorsque la séquence de tâches est prédéterminée. Ils ont présenté deux algorithmes heuristiques fondés sur la règle de Johnson et la programmation dynamique. Ils ont ensuite proposé une méthode de recherche locale itérative pour résoudre le problème. Les auteurs [Zheng et al. (2019)] ont étudié un problème d'ordonnancement d'atelier de type flow shop de permutation avec blocage sous le tarif ToU et les machines à vitesse variable. Ils ont proposé un algorithme de colonie de fourmis hybride multi-objectifs afin de trouver un compromis entre les coûts énergétiques totaux et le makespan.

### **2.2.2 “Critical Peak Pricing”**

Les auteurs [Modos et al. (2017)] ont étudié un problème d'ordonnancement sur une seule machine avec la minimisation du retard total. Dans leur étude, la consommation énergétique totale est limitée dans les intervalles de temps spécifiés. Si le fabricant ne peut pas respecter les limites de consommation énergétique totale pendant ces intervalles de temps spécifiés, il reçoit une pénalité. Ils ont estimé que les

incertitudes survenant lors de l'exécution d'ordonnancement pouvaient entraîner des retards inattendus et des violations des limites de consommation d'énergie. Ils ont proposé deux algorithmes exacts, un algorithme par séparation et évaluation avec une décomposition de Benders, et une heuristique de recherche taboue pour concevoir des ordonnancements robustes de production qui satisfont aux limites de consommation d'énergie vis-à-vis de scénarios inattendus.

Les auteurs [Bego et al. (2014)] ont présenté une méthodologie pour l'application du programme "Critical Peak Pricing" dans un système de fabrication qui se compose de plusieurs machines et de stocks intermédiaires en série. Ils ont développé une formulation non linéaire en variables mixtes pour établir le modèle mathématique pour l'objectif la minimisation du coût total d'électricité ainsi que le coût potentiel des pénalités en raison de la non-réalisation de la cible de production. Le modèle non linéaire développé est ensuite résolu par le solveur commercial LINGO.

### 2.2.3 "Real Time Pricing"

Les auteurs [Shrouf et al. (2014)] ont considéré un problème d'ordonnancement sur une seule machine avec la minimisation des coûts de consommation d'énergie sous des prix d'énergie variables pendant une journée. Dans leur étude, ils ont supposé que la séquence des tâches était prédéterminée. Ils visaient à déterminer les dates de début pour le traitement des tâches, le temps d'inactivité, les états de fonctionnement de la machine, etc. Ils ont proposé un algorithme génétique pour résoudre le problème. Ils ont également développé analytiquement des solutions optimales pour évaluer la performance de l'algorithme génétique proposé.

Les auteurs [Gong et al. (2017)] ont examiné un problème d'ordonnancement sur une seule machine qui prend en compte le coût d'électricité et le coût de main-d'œuvre sous le tarif d'électricité RTP. Généralement, le prix de main-d'œuvre est plus élevé pendant les périodes où le prix de l'électricité baisse. Ils ont examiné une étude de cas pour un processus de moulage par soufflage pour un fabricant de bouteilles en plastique. Bien que le coût de l'énergie augmente de 9% lorsqu'ils optimisent conjointement le coût de l'énergie et de la main-d'œuvre, ils pourraient réduire considérablement de 12% le coût total.

Les auteurs [Gong et al. (2019)] ont étudié un problème d'ordonnancement multi-objectif de l'atelier de type job shop flexible, qui minimise cinq objectifs : makespan, coût total de l'énergie, coût total de main-d'œuvre, charge de travail maximale et charge de travail totale. Dans la configuration de l'atelier, ils ont pris en compte la recirculation des tâches et le temps de réglage de la machine dépendant de la séquence. Ils ont adapté l'algorithme NSGA-III pour examiner les relations entre ces cinq objectifs sous RTP et ToU. Ils ont démontré que NSGA-III est plus efficace et efficient que NSGA-II pour résoudre un problème d'ordonnancement multi-objectifs de l'atelier de type job shop flexible avec prise en compte de l'énergie et du coût de la main-d'œuvre.

Pour comparer la différence entre les études dans la littérature, les tableaux

(1), (2) et (3) marquent les aspects énergétiques considérés dans chaque article. Dans chaque tableau (1), (2) et (3), nous trouvons les références dont l'objectif à minimiser, l'environnement machine et l'aspect énergétique sont résumés. Nous retrouvons les aspects énergétiques de l'efficacité énergétique (EE) et la réponse à la demande (DR).

- “Turn off/Turn on” représente la stratégie “Turn off/Turn on” de la machine.
- “Vitesse” signifie le choix de vitesse de traitement.
- “Machine” signifie le choix de machine de traitement.
- “CO<sub>2</sub>” représente l'émission de dioxyde de carbone.
- “Fiabilité” représente la consommation énergétique liée à la fiabilité du système.
- “Jobs” représente que la consommation énergétique est différente entre des tâches.
- “Etat machine” signifie la consommation énergétique liée aux états de fonctionnement de machine.
- “Contraintes” représente les contraintes énergétiques.

### 2.3 Conclusion

Nous avons présenté dans ce chapitre un état de l'art non exhaustif de la prise en compte de l'aspect énergétique dans les différentes études de l'ordonnancement des systèmes de fabrication industriel. Une classification des stratégies employées pour l'optimisation énergétique des problèmes d'ordonnancement a été fondée sur deux axes principaux : l'efficacité énergétique et la réponse à la demande. Cette classification permet de situer nos travaux par rapport à l'existant dans la littérature. Parmi ces études, à notre connaissance, il existe très peu de publications qui considèrent la tarification d'électricité dans l'ordonnancement d'atelier.

Dans cette thèse, nous nous intéressons à l'étude d'optimisation énergétique des problèmes d'ordonnancement d'atelier de type flow shop sous la tarification d'électricité. Nous proposons des méthodes exactes et approchées pour enrichir et améliorer des résultats existants dans la littérature. Nous proposons une approche qui applique pour la première fois la théorie de “group sequence” des solutions équivalentes de l'ordonnancement de flow shop aux études avec l'aspect énergétique. Nous présentons également une approche exacte de la décomposition de Benders qui est rarement utilisée dans les études de l'ordonnancement éco-énergétique de l'atelier.

Tableau 1 – Les aspects énergétiques dans les études d’ordonnement sur une seule machine.

N°	Références	Objective	Environnement	L'aspect énergétique																		
				EE							DR											
				Turn On/ Turn Off	Vitesse	Machine	CO <sub>2</sub>	Fiabilité	Jobs	Etat Machine	Contraintes	ToU	CPP	RTP								
1	[Mouzon et al. (2007)]	$\min \sum P E_i$ $\min C_{max}$	Une seule machine	x																		
2	[Albers and Fujiwara (2007)]	$\min \sum P E_i$ $\min \sum F_j$	Une seule machine		x																	
3	[Yildirim and Mouzon (2011)]	$\min \sum P E_i$ $\min C_{max}$	Une seule machine	x																		
4	[Antoniadis and Huang (2013)]	$\min \sum P E_i$	Une seule machine		x																	
5	[Antoniadis et al. (2014)]	$\min \sum P E_i$ $\min \sum C_j$	Une seule machine	x	x																	
6	[Liu et al. (2014a)]	$\min CE$	Une seule machine	x					x													
7	[Liu (2014)]	$\min \sum w_j T_j$ $\min CE$	Une seule machine						x													
8	[Shrouf et al. (2014)]	$\min TEC$	Une seule machine	x																	x	
9	[Bampis et al. (2015)]	$\min \sum P E_i$	Une seule machine		x																	
10	[Fang et al. (2016)]	$\min TEC$	Machines parallèles							x											x	
11	[Che et al. (2016)]	$\min TEC$	Une seule machine								x										x	
12	[Mouzon and Yildirim (2008)]	$\min \sum P E_i$ $\min \sum T_j$	Une seule machine	x																		
13	[Modos et al. (2017)]	$\min \sum T_j$	Une seule machine																			
14	[Che et al. (2017)]	$\min \sum P E_i$ $\min T_{max}$	Une seule machine	x																	x	
15	[Gong et al. (2017)]	$\min TEC$ $\min LC$	Une seule machine																			
16	[Aghelnejad et al. (2018b)]	$\min TEC$	Une seule machine	x																	x	
17	[Aghelnejad et al. (2018a)]	$\min TEC$	Une seule machine	x																	x	
18	[Chen and Zhang (2019)]	$\min TEC$	Une seule machine																		x	
19	[Chen et al. (2019)]	$\min \sum T_j$ $\min TEC$	Une seule machine																		x	
20	[Liu and Huang (2014)]	$\min \sum w_j T_j$ $\min CE$ $\min P_{lmax}$	Une seule machine Flow shop hybride																		x	
21	[Chen et al. (2018)]	$\min(\sum w_j C_j + TEC)$	Une seule machine Machines parallèles																			x

Tableau 2 – Les aspects énergétiques dans les études d’ordonnement d’atelier de type flow shop.

N°	Références	Objective	Environnement	L'aspect énergétique													
				EE						DR							
				Turn On/ Turn Off	Vitesse	Machine	CO <sub>2</sub>	Fiabilité	Jobs	Etat Machine	Contraintes	ToU	CPP	RTP			
1	[Fang et al. (2011)]	$\min P_{i_{max}}$ $\min \sum PE_i$ $\min CE$	Flow shop	x			x										
2	[Bruzzone et al. (2012)]	$\min(\sum w_j T_j + C_{max})$	Flow shop flexible			x											
3	[Mashaei and Lennartson (2012)]	$\min \sum PE_i$	Flow shop flexible	x		x											
4	[Dai et al. (2013)]	$\min \sum PE_i$ $\min C_{max}$	Flow shop	x													
5	[Fang et al. (2013)]	$\min C_{max}$	Flow shop		x								x				
6	[Li et al. (2015)]	$\min C_{max}$ $\min \sum PE_i$ $\min CE$	Flow shop														
7	[Hadera et al. (2015)]	$\min \sum w_j C_j$ $\min TEC$	Flow shop flexible			x											x
8	[Tang et al. (2016)]	$C_{max}$ $\min \sum PE_i$	Flow shop flexible														
9	[Mansouri et al. (2016)]	$\min \sum PE_i$ $\min C_{max}$	Flowshop		x						x						
10	[Liu et al. (2017)]	$\min \sum PE_i$	Flowshop														
11	[Lu et al. (2017)]	$\min C_{max}$ $\min \sum PE_i$	Flow shop		x												
12	[Wang et al. (2018)]	$\min TEC$	Flowshop														x
13	[Pileroot et al. (2018)]	$\min TEC$	Flow shop														x
14	[Zhang et al. (2019)]	$\min \sum PE_i$ $\min C_{max}$	Flowshop hybride		x												
15	[Jiang and Wang (2019)]	$\min C_{max}$ $\min \sum PE_i$	Flow shop		x												
16	[Zheng et al. (2019)]	$\min C_{max}$ $\min TEC$	Flowshop		x												x
17	[Bego et al. (2014)]	$\min TEC$ $\min TC$	Ligne de production														x
18	[Liang et al. (2019)]	$\min TC$	Ligne de production													x	



### 3 La minimisation de coût d'électricité de la solution optimale de flow shop à deux machines sous ToU.

Dans ce chapitre, nous présentons l'optimisation conjointe du makespan et du coût de l'électricité dans le problème d'ordonnancement de flow shop à deux machines sous la tarification d'électricité. Nous améliorons l'aspect financier de la solution optimale de  $(F2|perm|C_{max})$  en minimisant le coût de l'électricité sans augmenter le makespan. Après la description du problème, nous présentons dans un premier temps, la théorie de solutions optimales équivalentes. Cette théorie permet de générer un ensemble de plusieurs solutions optimales de  $(F2|perm|C_{max})$ . Nous exploitons cet ensemble pour construire un plan de production de coût d'électricité minimal avec le makespan optimal. Les tests numériques montrent que notre approche proposée améliore considérablement le coût de l'électricité sans augmenter le makespan qui peut contribuer au développement durable de l'industrie de fabrication.

Ce chapitre est organisé de la façon suivante. Dans la première section, nous décrivons le problème. Ensuite, nous formulons le modèle mathématique. Nous proposons des approches de résolutions dans la section 3. La performance des approches de résolution est montrée par les tests numériques dans la section 4. Enfin, la section 5 dresse la conclusion sur la contribution de cette étude.

#### 3.1 La description du problème

Cette étude aborde le problème d'ordonnancement d'un atelier de type flow shop à deux machines sous ToU avec une optimisation conjointe du makespan et du coût de l'électricité, notée  $(F2, ToU|perm, C_{max}^*|TEC)$ . Dans cette étude, nous considérons la valeur optimale du makespan obtenue par la règle de Johnson comme la date limite. Toute solution faisable de ce problème doit respecter le makespan optimal  $C_{max}^*$ . La consommation énergétique dans cette étude dépend de la caractéristique des tâches. Sur une même machine, la consommation énergétique de traitement d'une tâche est différente de celle de traitement d'une autre. Déplacer les tâches qui consomment plus d'énergie vers des périodes de prix bas peut minimiser le coût total de l'énergie, et inversement. Le coût de l'électricité est donc minimisé en deux étapes. Lors de la première étape, nous générons un ensemble  $E$  de permutations  $\pi$  dont le makespan est optimal,  $(\forall \pi \in E / C_{max}^\pi \leq C_{max}^*)$ . Ensuite, dans la deuxième étape, nous déterminons la date de début optimale de traitement des tâches dans chaque permutation  $\pi$ .

Le problème d'ordonnancement de flow shop à deux machines pour minimiser le makespan est d'abord étudié par [Johnson 1954]. Il a proposé un algorithme polynomial pour résoudre le problème de manière optimale. Cependant, afin de construire la séquence, il a considéré uniquement le temps de traitement des tâches, qui n'est pas toujours corrélé avec la puissance requise pour chaque tâche. De plus, l'algorithme de Johnson produit une solution unique parmi un grand ensemble de solutions optimales. En considérant le cas où la relation entre le temps de

traitement et la demande d'énergie est arbitraire, la consommation énergétique sur l'horizon de travail de chaque solution peut être différente malgré des makespans équivalents. Dans le cadre du tarif ToU, le prix de l'électricité varie d'une période à l'autre. En conséquence, les solutions optimales équivalentes ont des coûts d'électricité différents malgré le même makespan. C'est la raison pour laquelle le manager peut mieux adapter son plan de production à la variation du prix de l'électricité avec un large ensemble de solutions équivalentes.

Le tableau (5) montre les avantages offerts par l'ensemble de solutions optimales équivalentes en termes de makespan. La première solution est trouvée par l'algorithme de Johnson. Toutes les solutions ont le même makespan égal à 460. Le temps de traitement de chaque tâche dans l'exemple est donné dans le tableau (4a).

La consommation d'énergie sur l'horizon de travail sera différente entre les solutions car la demande d'énergie de chaque tâche est différente et sans rapport avec son temps de traitement. Dans le cadre de ToU, la différence de coût de l'électricité entre les différentes solutions peut être donc importante. Par exemple, la demande d'énergie de chaque tâche est représentée dans le tableau (4b) et le tarif de l'électricité basé sur [Sharma et al. 2015] est illustré dans la Fig.(7a).

Tableau 4 – Le temps de traitement et la puissance requise pour chaque tâche

(a) Le temps de traitement de chaque tâche

	Tâche 1	Tâche 2	Tâche 3	Tâche 4	Tâche 5	Tâche 6	Tâche 7	Tâche 8	Tâche 9
Machine 1	37	36	1	4	64	74	32	67	73
Machine 2	59	16	90	3	76	74	22	30	89

(b) La puissance requise pour le traitement de chaque tâche

	Tâche 1	Tâche 2	Tâche 3	Tâche 4	Tâche 5	Tâche 6	Tâche 7	Tâche 8	Tâche 9
Machine 1	2.2	1.8	1.6	2.5	1.3	2.4	2.6	1.7	2.7
Machine 2	1.2	1.7	2.4	2.7	1.5	2.6	2.8	2.7	2.6

En comparant la première solution avec la seconde, nous pouvons trouver une différence de 12,36 % en termes de coût d'électricité par rapport à la seconde solution bien qu'ils aient le même makespan. La comparaison peut être trouvée dans le tableau (5), où  $TEC$  est le coût total de l'électricité. L'écart de 12,36 % est calculé comme suit :  $Gap_i = \frac{TEC_i - TEC_2}{TEC_2}$ . Le coût de l'électricité de chaque solution est calculé par :

$$TEC = \sum_{t \in T} Ener_t \times c_t$$

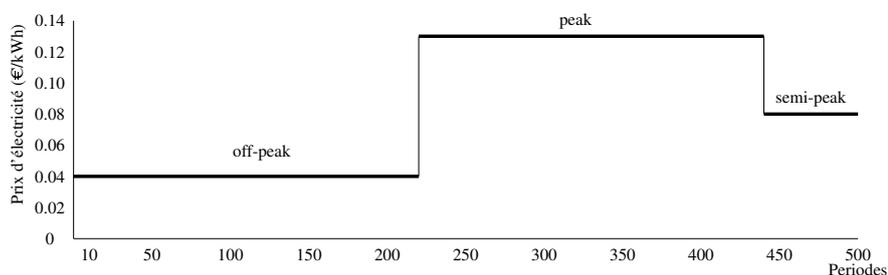
Dont :

- $Ener_t$ : consommation d'énergie du système pendant la période  $t$ ,  $t = 1 \dots T$ .
- $c_t$ : prix de l'électricité à la période  $t$ .

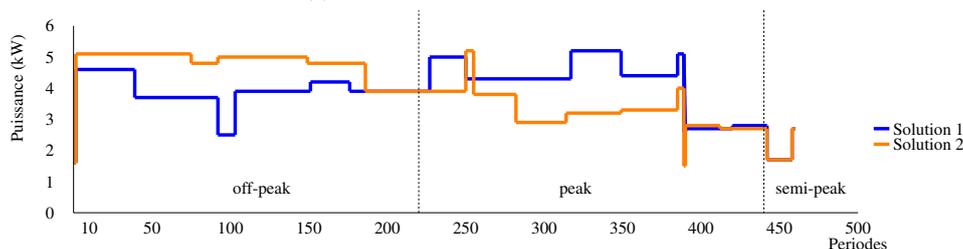
Tableau 5 – La comparaison des solutions optimales équivalentes  $F2|perm|C_{max}$

N°	Sequence	Makespan	TEC	Gap
1	3-1-5-9-6-8-7-2-4	460	156.616	12.36%
<b>2</b>	<b>3-9-6-1-5-7-8-2-4</b>	<b>460</b>	<b>139.391</b>	<b>0.00%</b>
3	3-6-9-1-5-8-7-2-4	460	139.501	0.08%
4	3-9-5-1-6-8-7-2-4	460	145.582	4.44%
5	3-6-1-9-5-8-7-2-4	460	145.725	4.54%
6	3-9-5-1-6-7-8-2-4	460	146.042	4.77%
...	...	...	...	...

La Fig.(7b) représente la consommation d'énergie de la première et de la deuxième solution sur l'horizon de travail. Ce dernier est divisé en trois périodes de prix : les périodes "off-peak" où le prix de l'électricité est le plus bas, les périodes "peak" où le prix de l'électricité est le plus élevé et les périodes "semi-peak" où le prix de l'électricité est moyen. Il est facile de constater que dans la solution 1, une quantité importante d'énergie est utilisée dans les périodes "peak" tandis que dans la solution 2, l'énergie est utilisée principalement dans les périodes "off-peak". Dans la deuxième solution, les tâches dont la puissance requise est élevée sont traités en périodes "off-peak" où l'électricité est moins chère que les autres périodes.



(a) La tarification d'électricité de ToU



(b) L'évolution de la consommation énergétique de la solution 1 et la solution 2

Figure 7 – La comparaison entre la solution 1 et la solution 2

### 3.2 La modélisation mathématique

Le problème d'ordonnancement  $(F2, ToU | perm, C_{max}^* | TEC)$ , peut être modélisé comme suit. Nous avons un ensemble de tâches  $J = \{1, 2, \dots, n\}$  à traiter sur  $M = \{1, 2\}$  machines à partir de l'instant zéro. Les tâches doivent être traitées de manière non préemptive d'abord sur la machine 1, puis sur la machine 2. Le temps de traitement de tâche  $j \in J$  sur la machine  $i \in M$  est  $p_{ij}$ . À tout moment, chaque machine ne peut traiter qu'une seule tâche au maximum, et chaque tâche ne peut être traitée que sur une seule machine à la fois. Le traitement de tâche  $j \in J$  sur la machine  $i \in M$  consomme une puissance énergétique  $Pu_{ij}$  par unité de temps qui entraîne des coûts énergétiques correspondants.

Le prix d'électricité varie au fil de l'horizon de travail selon le schéma de tarification de type ToU. L'horizon de travail est composé de  $T$  périodes de temps. Ces périodes peuvent être classées en  $K$  intervalles de prix, chaque intervalle de prix  $p \in K$  est caractérisée par la date de début  $St_p$ , la durée  $d_p$  et le prix d'électricité  $fc_p$ . Les périodes  $t \in T$  qui se trouvent dans un intervalle de prix  $p$  ont le prix d'électricité  $c_t = fc_p, \forall St_p \leq t \leq St_p + d_p$ . En une journée, nous avons trois types de périodes : "off-peak", "peak" et "semi-peak". Pour une période donnée, les coûts d'électricité correspondent à la consommation d'énergie multipliée par les prix d'électricité au cours de cette période.

Nous rappelons que le problème  $(F2 | perm | C_{max})$  peut être résolu de manière optimale par la règle de Johnson [Johnson 1954]. Cette étude a pour l'objectif d'améliorer l'aspect financier des solutions optimales de  $(F2 | perm | C_{max})$  en assurant le même niveau de productivité ou autrement dit, le même makespan. Par conséquent, la valeur  $C_{max}^*$  de la solution de Johnson sera considérée comme une contrainte et l'objectif est la minimisation du coût de l'électricité.

Nous représentons un modèle mathématique linéaire mixte en nombre entiers pour le problème  $(F2, ToU | perm, C_{max}^* | TEC)$ . Le modèle est fondé sur le modèle de Manne pour le problème d'ordonnancement d'atelier de flow shop [Gupta and Stafford Jr 2006]. Nous intégrons les variables qui déterminent l'état de fonctionnement de chaque machine à chaque période. Ces variables nous permettent de tracer la consommation d'énergie au cours de chaque période. La définition des notation du problème, des variables est donnée comme suit dans le tableau (6) et le tableau (7).

Tableau 6 – Notations et paramètres

Notations

$M$	l'ensemble de machines.
$i$	l'indice de machine, $i = 1, 2$ .
$J$	l'ensemble de tâches.
$j, j'$	l'indice de tâches, $j, j' = 1, \dots, n$ .
$T$	l'horizon de travail.
$t, h$	l'indice de périodes de temps, $t, h = 1, \dots, T$ .
$p_{ij}$	le temps de traitement de tâche $j$ sur la machine $i$ .
$Pu_{ij}$	la puissance requise pour traiter la tâche $j$ sur la machine $i$ .
$c_t$	le prix d'électricité à période $t$ .
$V, L$	très grande valeur.

Tableau 7 – Variables

Variables

$x_{jj'}$	variable binaire, 1 si la tâche $j$ est placée avant la tâche $j'$ , 0 si non
$C_{ij}$	la date d'achèvement de tâche $j$ sur la machine $i$ .
$z_{ijt}$	variable binaire, 1 si la tâche $j$ est traitée sur la machine $i$ pendant la période $t$ , $t = 1 \dots T$ , 0 si non.
$E_{it}$	la consommation énergétique de la machine $i$ pendant la période $t$ , $t = 1 \dots T$ .
$Ener_t$	la consommation énergétique du système pendant la période $t$ , $t = 1 \dots T$ .
$C_{max}$	la date d'achèvement maximale, le makespan.
$TEC$	le coût d'électricité total.

Le modèle mathématique:

$$\text{Min } TEC = \sum_{t \in T} Ener_t \times c_t \quad (1)$$

$$C_{max} \geq C_{ij} \quad \forall i \in M, \forall j \in J \quad (2)$$

$$C_{1j} \geq p_{1j} \quad \forall j \in J \quad (3)$$

$$C_{(i+1)j} - C_{ij} \geq p_{(i+1)j} \quad \forall j \in J, 1 \leq i \leq M-1 \quad (4)$$

$$C_{ij} - C_{ij'} + V \times x_{jj'} \geq p_{ij} \quad \forall i \in M, 1 \leq j < j' \leq J \quad (5)$$

$$C_{ij'} - C_{ij} + V \times (1 - x_{jj'}) \geq p_{ij'} \quad \forall i \in M, 1 \leq j < j' \leq J \quad (6)$$

$$C_{max} \leq C_{max}^* \quad (7)$$

$$z_{ijt} \times t \leq C_{ij} \quad \forall i \in M, \forall j \in J, \forall t = 1 \dots T \quad (8)$$

$$C_{ij} - p_{ij} \leq (1 - z_{ijt}) \times L + t - 1 \quad \forall i \in M, \forall j \in J, \forall t = 1 \dots T \quad (9)$$

$$\sum_{t=1 \dots T} z_{ijt} = p_{ij} \quad \forall i \in M, \forall j \in J \quad (10)$$

$$Ener_t = \sum_{i \in M} \sum_{j \in J} z_{ijt} \times Pu_{ij} \quad \forall t \in T \quad (11)$$

$$x_{jj'} \in \{0, 1\} \quad \forall j, j' \in J \quad (12)$$

$$z_{ijt} \in \{0, 1\} \quad \forall t \in T, \forall j \in J, \forall i \in M \quad (13)$$

$$C_{ij}, C_{max} \geq 0 \quad \forall j \in J, \forall i \in M \quad (14)$$

$$Ener_t \geq 0 \quad \forall t \in T \quad (15)$$

L'objectif (1) est de minimiser le coût total de l'électricité. Les contraintes (2) définissent le makespan qui est la date d'achèvement maximale de toutes les tâches sur toutes les machines. Les contraintes (3) garantissent que la date de fin de chaque tâche sur la première machine n'est pas plus tôt que son temps de traitement sur la première machine. Les contraintes (4) garantissent que la date de fin de chaque tâche sur la deuxième machine n'est pas antérieure à sa date de fin sur la première machine ajouté à son temps de traitement sur la première machine. Les contraintes (5) et (6) assurent que la tâche  $j$  soit précède, soit succède la tâche  $j'$ . Les contraintes (7) assurent que la date d'achèvement maximale de la solution équivaut au makespan optimal. Les contraintes (8), (9) et (10) déterminent l'état de fonctionnement de la machine. Si la machine  $i$  traite la tâche  $j$  pendant la période  $t$ , elle doit être limitée par la date de fin (8), la date de début (9) et doit satisfaire le temps de traitement de tâche  $j$  (10). Les contraintes (11) calculent la consommation énergétique du système à la période  $t$  en fonction de l'état de fonctionnement de la machine et la puissance requise. Les contraintes (12), (13), (14) et (15) sont la définition des variables.

### 3.3 L'approche de résolution

Nous présentons dans cette section l'approche de résolution pour le problème  $(F2, ToU|perm, C_{max}^*|TEC)$ . L'approche de résolution se compose de deux étapes principales. En première étape, nous établissons l'ensemble des solutions optimales de problème  $(F2|perm|C_{max})$  car il fait partie de l'enveloppe de solution de notre problème. La recherche de l'ensemble des solutions optimales est fondée sur la théorie des solutions optimales équivalentes de [Hadda et al. 2018]. Ensuite, nous donnons les propositions pour étendre l'ensemble des solutions optimales. En deuxième étape, pour chaque solution trouvée en étape précédente, nous y appliquons une procédure qui détermine la date de début optimale de traitement des tâches pour construire un plan de production de coût d'électricité minimal du problème  $(F2, ToU|perm, C_{max}^*|TEC)$ .

#### 3.3.1 L'ensemble des solutions de $F2|perm|C_{max}$

##### 3.3.1.1 La règle de Johnson

Il est évident que la solution optimale de  $(F2|perm|C_{max})$  est une solution faisable de  $(F2, ToU|perm, C_{max}^*|TEC)$ . La règle de Johnson [Johnson 1954] permet de résoudre le problème  $(F2|perm|C_{max})$  de manière optimale et polynomiale. Alors,

avant d'étudier l'approche de résolution de  $(F2, ToU|perm, C_{max}^*|TEC)$ , nous rappelons la règle de Johnson ci-dessous.

Pour toutes instances de  $(F2|perm|C_{max})$ , prenant en compte uniquement le temps de traitement, nous pouvons séparer les tâches en deux ensembles  $\bar{J} = \{J_j \in J | p_{1j} < p_{2j}\}$  et  $\underline{J} = \{J_j \in J | p_{2j} < p_{1j}\}$ . Les tâches dont  $p_{1j} = p_{2j}$  peuvent être placées arbitrairement soit dans  $\bar{J}$  soit dans  $\underline{J}$ . L'algorithme de Johnson permet de trouver la solution optimale de  $(F2|perm|C_{max})$  en  $O(n \log n)$  étapes.

1. Trier  $\bar{J}$  dans l'ordre non-décroissant de temps de traitement sur la machine 1
2. Trier  $\underline{J}$  dans l'ordre non-croissant de temps de traitement sur la machine 2.
3. La séquence optimale est constituée de  $\bar{J}$  suivi de  $\underline{J}$  triés.

Nous prenons un exemple de quatre tâches (tableau (8)) pour illustrer l'algorithme de Johnson. Dans un premier temps, nous trions les tâches pour obtenir deux ensembles  $\bar{J} = \{J_1, J_3\}$  et  $\underline{J} = \{J_2, J_4\}$ . Ensuite nous trions  $\bar{J}$  dans l'ordre non-décroissant de temps de traitement sur la machine 1, nous obtenons  $J_3 \prec J_1$ . Nous trions  $\underline{J}$  dans l'ordre non-croissant de temps de traitement sur la machine 2, nous obtenons  $J_4 \prec J_2$ . Alors, la solution optimale de  $(F2|perm|C_{max})$  est constituée de  $\bar{J}$  suivi de  $\underline{J}$ , et nous obtenons  $J_3 \prec J_1 \prec J_4 \prec J_2$ .

Tableau 8 – Le temps de traitement de l'exemple 1

	Tâche 1	Tâche 2	Tâche 3	Tâche 4
Machine 1	8	7	4	6
Machine 2	9	2	8	5

### 3.3.1.2 L'ensemble des solutions optimales équivalentes

La solution de Johnson n'est pas la seule solution optimale pour le problème  $(F2|perm|C_{max})$ . L'auteur [Kamburowski (1997)] a montré qu'il peut y avoir d'autres solutions optimales en plus de celle trouvée par la règle de Johnson. Les auteurs Billaut et Lopez [Billaut and Lopez (1998)] ont proposé une approche fondée sur l'énumération pour construire l'ensemble des solutions optimales à partir de séquence de Johnson. Cette approche ne peut être utilisée que pour les instances de petites tailles (moins d'une douzaine de tâches) en raison de la complexité élevée de l'algorithme proposé.

Au lieu d'utiliser une énumération, les auteurs de [Briand et al. 2006] ont proposé deux structures d'intervalle spéciales, une pour  $\bar{J}$  et l'autre pour  $\underline{J}$ . Chaque tâche est associée à un intervalle fondée sur l'ordre relatif des temps de traitement. Ils ont ensuite utilisé l'algèbre d'Allen pour analyser les relations entre les intervalles afin de fournir un ordre partiel des tâches. L'ordre partiel donne la possibilité de positionner un certain nombre de tâches sans augmenter la valeur du makespan.

Les auteurs de [Hadda et al. 2018] ont proposé une approche pour construire un "group sequence" pour trouver un large ensemble des solutions optimales. Un

“group sequence” est une séquence de groupes de tâches permutables où toute permutation des tâches à l’intérieur du groupe n’augmente pas la date d’achèvement initiale. En fonction des temps de traitement, l’approche recherche la permutation qui compose des “group sequence” et la relation de précédence entre eux. L’ordre des tâches dans chaque “group sequence” est arbitraire, ce qui conduit à un grand nombre de solutions équivalentes. Ils ont montré également que leur approche dominait celle donnée dans [Briand et al. 2006] concernant le nombre de solutions équivalentes.

Dans les sections qui suivent, nous présentons l’algorithme proposé par [Hadda et al. 2018] car cela joue un rôle important dans notre approche de résolution. Ensuite, nous améliorons cette approche initiale par de nouvelles propositions ainsi la recherche locale qui permettent d’élargir l’ensemble des solutions optimales équivalentes trouvées.

### 3.3.1.3 L’algorithme de [Hadda et al. 2018]

L’algorithme proposé par [Hadda et al. 2018] consiste à repartir des tâches en “group sequence” et ensuite à déterminer l’ordre partiel total entre eux. [Cheng et al. (2002)] ont défini un ordre partiel  $P$  par une paire  $P = (X, \prec_p)$ ,  $X$  étant un ensemble d’éléments et la relation  $\prec_p$  sur  $X \times X$  est réflexive, antisymétrique et transitive. Un ordre partiel total est fondé sur un ordre partiel  $P = (X, \prec_p)$  dans lequel, pour tout couple  $(u, v) \in X \times X$ , la relation  $u \prec_p v$  ou  $v \prec_p u$  est vérifiée. Dans le contexte de notre étude, l’ensemble  $X$  correspond à l’ensemble de tâche  $J$  et la relation  $\prec_p$  est la relation de précédence entre deux tâches ou deux ensembles. Pour deux ensembles de tâches  $G$  et  $H$ , nous disons que  $G \prec H$  ( $G \cap H = \emptyset$ ) si les tâches de  $H$  ne peuvent commencer qu’après avoir terminé toutes les tâches de  $G$ .

Le “group sequence” se compose des “ground job” et des groupes de tâches associés et succédés à chaque “ground job”. L’ordre dans chaque groupe de tâches est arbitraire. La permutation entre deux tâches quelconques dans un groupe n’impacte pas la valeur de date d’achèvement du groupe. Nous prenons un exemple dans la Fig.(8). Nous avons un ordre partiel total de trois “ground job”  $J_1, J_4, J_7$ , le groupe de tâches  $\{J_2, J_3\}$  associé à  $J_1$  et le groupe  $\{J_5, J_6\}$  associé à  $J_4$ . L’ordre partiel total est représenté comme  $J_1 \prec \{J_2, J_3\} \prec J_4 \prec \{J_5, J_6\} \prec J_7$ . La permutation des tâches à l’intérieur de chaque groupe nous donne trois autres solutions équivalentes en terme de makespan.

- N°1 :  $J_1 \prec J_3 \prec J_2 \prec J_4 \prec J_5 \prec J_6 \prec J_7$
- N°2 :  $J_1 \prec J_2 \prec J_3 \prec J_4 \prec J_5 \prec J_6 \prec J_7$
- N°3 :  $J_1 \prec J_3 \prec J_2 \prec J_4 \prec J_6 \prec J_5 \prec J_7$

Avant d’entrer dans les détails de l’algorithme de [Hadda et al. 2018], nous rappelons que pour n’importe quelle instance de  $(F2|perm|C_{max})$ , les tâches peuvent être classées soit dans l’ensemble  $\bar{J}$  soit dans l’ensemble  $\underline{J}$  selon le temps de traitement. En plus, le problème  $(F2|perm|C_{max})$  est réversible. Ça signifie que

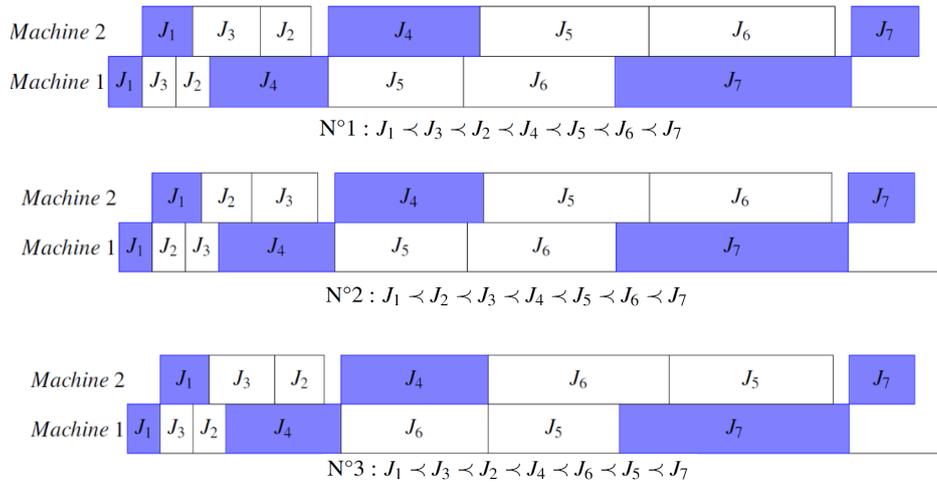


Figure 8 – Les solutions équivalentes de l'ordre partiel total  $J_1 \prec \{J_2, J_3\} \prec J_4 \prec \{J_5, J_6\} \prec J_7$

le problème ne change pas si nous inversons le rôle des machines et inversons le chemin de passage des tâches en même temps (la machine 1 devient la machine 2 et inversement). La réversibilité du problème donne un résultat très intéressant. Les propositions que nous développons pour l'ensemble  $\bar{J}$  restent valables pour l'ensemble  $\underline{J}$  en inversant la configuration.

Pour une séquence  $\pi$  quelconque, nous supposons que toutes les tâches de  $\pi$  commencent au plus tôt. Nous avons les définitions et les notations nécessaires ci-dessous.

- $\pi = \langle \pi_1, \pi_2, \dots, \pi_n \rangle$  : une séquence de tâches, où  $\pi_j$  indique la tâche à la  $j^{\text{eme}}$  position dans  $\pi$ .
- $S_{ij}(\pi)$  et  $C_{ij}(\pi)$  : la date de début et la date de fin de la tâche  $\pi_j$  sur la machine  $i$ ,  $J \in 1, \dots, n$ ,  $i \in \{1, 2\}$  de la séquence  $\pi$ . Pour simplifier la notation, nous enlevons  $\pi$  quand il n'y a pas de risque de confusion.
- $C_{max}(\pi)$  : le makespan de la séquence  $\pi$ .
- $\pi^*$  : la séquence optimale
- $C_{max}^*$  : la valeur optimale du makespan.
- Pour un ensemble de tâches  $G \subseteq J$ ,  $a(G)$  et  $b(G)$  sont définies comme fonctions :  $a(G) = \sum_{J_j \in G} p_{1j}$  et  $b(G) = \sum_{J_j \in G} p_{2j}$  (si  $G = \emptyset$  alors  $a(G) = b(G) = 0$ )

Pour une séquence  $\pi$ , nous avons le makespan de  $\pi$  ci-dessous :

$$C_{max}(\pi) = \max_{1 \leq u \leq n} \left\{ \sum_{i=1}^u p_{1\pi_i} + \sum_{i=u}^n p_{2\pi_i} \right\}$$

Dans n'importe quelle séquence donnée, nous pouvons toujours déterminer une tâche qui s'appelle "crossover job". Une tâche est appelée "crossover job"  $J_z = \pi_x$  si elle satisfait la condition :  $C_{max}(\pi) = \sum_{i=1}^x p_{1\pi_i} + \sum_{i=x}^n p_{2\pi_i}$ . Cela signifie que le traitement de cette tâche sur la machine 2 commence tout de suite après la fin de son traitement sur la machine 1 ( $S_{2z} = C_{1z}$ ) et que la machine 2 n'a plus de temps libre après  $S_{2z}$ . Autrement dit, le "crossover job" est précédé par le dernier temps d'inactivité sur la machine 2.

Soit  $\bar{\pi}$  une sous-séquence des tâches de l'ensemble  $\bar{J}$ , une tâche  $\bar{\pi}_w$  est appelée "ground job" si elle est précédée par le temps d'inactivité sur la machine 2, ce qui signifie qu'elle satisfait  $S_{2\bar{\pi}_w} = C_{1\bar{\pi}_w}$  et  $C_{2\bar{\pi}_{w-1}} \neq S_{2\bar{\pi}_w}$  (avec  $C_{2\bar{\pi}_0} = 0$ ). Soit  $l$  le nombre de "ground job" de  $\bar{\pi}$ , nous notons les "ground job" par  $g_p, p \in \{1, \dots, l\}$ . Nous notons également  $p_{1p}^g$  et  $p_{2p}^g$  le temps de traitement de "ground job"  $g_p$  sur la machine 1 et la machine 2 respectivement.

Nous notons  $\theta_p, p \in \{1, \dots, l-1\}$  le groupe de tâches qui est associé à "ground job"  $g_p$ . Ces tâches seront séquencées entre  $g_p$  et  $g_{p+1}$ . Elles succèdent au "ground job"  $g_p$  et précèdent le  $g_{p+1}$ . Le groupe de tâche  $\theta_p$  peut être vide pour certains  $p \in \{1, \dots, l\}$ . En utilisant ces notations définies, la sous-séquence  $\bar{\pi}$  de  $\bar{J}$  peut être sous la forme  $\bar{\pi} = g_1 \prec \theta_1 \prec g_2 \prec \theta_2 \dots \prec g_l \prec \theta_l$ . Nous associons à chaque "ground job"  $g_p = \bar{\pi}_w, p \in \{1, \dots, l\}$  la quantité  $\delta_p = S_{2\bar{\pi}_w} - C_{2\bar{\pi}_{w-1}}$ .  $\delta_p$  correspond à la dernière période de temps d'activité sur la machine 2 qui précède le traitement de  $g_p$  sur cette machine.

Comme le problème  $(F2|perm|C_{max})$  est réversible, nous pouvons également définir les  $l'$  "ground job"  $g'_p$ , leurs groupes de tâches associés  $\theta'_p$  et les temps d'inactivité  $\delta'_p$  pour la sous-séquence  $\underline{\pi}$  de  $\underline{J}$ . Ces éléments de la sous-séquence  $\underline{\pi}$  sont définis en inversant la configuration du problème : la machine 1 est considérée comme la machine 2 et inversement, les tâches de  $\underline{\pi}$  commencent au plus tard possible.

Alors, après avoir déterminé les éléments de  $\bar{\pi}$  et de  $\underline{\pi}$ , n'importe quelle séquence  $\pi$  peut se décomposer de cette façon (Fig.(9)) :

$$\pi = g_1 \prec \theta_1 \prec \dots \prec g_l \prec \theta_l \prec \theta'_l \prec g'_l \prec \dots \prec \theta'_1 \prec g'_1$$

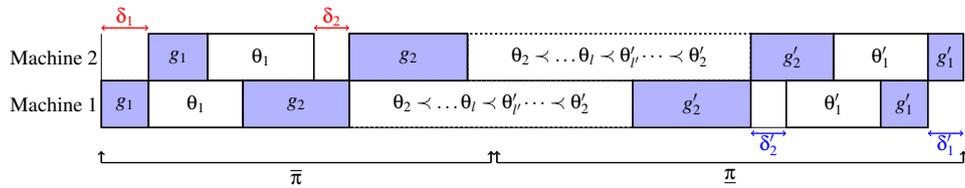


Figure 9 – Les "ground job", les groupes de tâches et les temps d'inactivité.

Nous établissons séparément des sous-séquences pour l'ensemble de tâches de  $\bar{J}$  et  $\underline{J}$ . La séquence complète est constituée de  $\bar{J}$  suivi de  $\underline{J}$ . Nous présentons les résultats pour l'ensemble de  $\bar{J}$ . Ces résultats sont toujours valables pour  $\underline{J}$  en inversant la configuration du problème.

**Lemma 1.** *Pour une instance de  $(F2|perm|C_{max})$  dont  $\underline{J} = \emptyset$  et les dates disponibles des machines 1 et 2 sont  $s_1$  et  $s_2$  respectivement, si  $p_{1j} \leq s_2 - s_1, \forall j \in \bar{J}$ , alors toutes les solutions ont les mêmes dates d'achèvement qui valent  $(s_1 + a(\bar{J}))$  sur la machine 1 et  $(s_2 + b(\bar{J}))$  sur la machine 2. [Hadda et al. 2018]*

En se fondant sur le lemme (1), [Hadda et al. 2018] ont proposé l'algorithme *OG* qui permet de repartir des tâches en "ground job" et de construire un ordre partiel total nommé  $O^{OG}$  de la solution optimale de  $(F2|perm|C_{max})$ . Si  $n_p$  est le nombre de groupes de tâches associés à "ground job"  $g_p$ , l'algorithme spécifie la composition et l'ordre dans lequel le groupe de tâches  $\theta_p^q (p \in \{1, \dots, l\}, q \in \{1, \dots, n_p\})$  devrait apparaître. La séquence des tâches dans chaque groupe  $\theta_p^q$  est arbitraire, ce qui offre une flexibilité sur le positionnement des tâches sans risquer d'augmenter la valeur du makespan.

Pour un ensemble de  $\bar{J}$ , l'algorithme *OG* commence en choisissant la première tâche de "ground job"  $g_1$ . Ce "ground job" est la tâche dont le temps de traitement sur la machine 1 est le plus petit parmi celles de  $\bar{J}$ . Ce choix permet de minimiser le temps d'inactivité sur la machine 2. Comme nous traitons les tâches à partir de l'instant 0, la date de disponibilité des machines après avoir traité  $g_1$  est  $s_1 = C_{1g_1}$  et  $s_2 = C_{1g_1} + p_{21}^g$ . Ensuite, l'algorithme (1) construit un ensemble  $I$  des tâches  $J_j$  qui satisfont la condition du lemme (1) ( $p_{1j} \leq s_2 - s_1 = C_{1g_1} + p_{21}^g - C_{1g_1} = p_{21}^g$ ). Comme des tâches de  $I$  satisfont la propriété du lemme (1), alors elles peuvent être placées arbitrairement après  $g_1$  sans entraîner de temps d'inactivité sur les deux machines. Nous prenons le groupe de tâches  $\theta_1^1$  des tâches de  $I$  associé à "ground job"  $g_1$  ( $g_1 \prec \theta_1^1$ ). Soit  $R$ , l'ensemble des tâches qui succèdent le "ground job", nous construisons à nouveau un ensemble  $I'$  des tâches qui satisfont la condition  $p_{1j} \leq p_{21}^g + b(R) - a(R)$ . Nous prenons ensuite le groupe de tâches  $\theta_1^2$  des tâches de  $I'$ . ( $g_1 \prec \theta_1^1 \prec \theta_1^2$ ).

Le processus continue itérativement jusqu'à ce que l'ensemble  $I$  soit vide, ce qui signifie qu'il n'y a plus de tâche qui satisfait la condition du lemme (1). L'algorithme *OG* choisit une tâche dont le temps de traitement sur la machine 1 est le plus petit comme le "ground job" suivant et recommence le processus pour déterminer les groupes de tâches associé à ce nouveau "ground job". L'algorithme *OG* arrête quand toutes les tâches de  $\bar{J}$  sont placées. Alors, l'algorithme *OG* détermine la composition et l'ordre de "ground job"  $g_p$  et leurs groupes de tâches  $\theta_p^q (p \in \{1, \dots, l\}, q \in \{1, \dots, n_p\})$ , l'ordre de tâches dans chaque groupe  $\theta_p^q$  est arbitraire. Les détails de l'algorithme *OG* sont donnés dans l'algorithme (1).

**Algorithme 1** : L'algorithme *OG* proposé par (Hadda et al. (2018))

```

Choisir  $J_z \in \bar{J}$  avec  $p_{1z} = \min_{J_j \in \bar{J}} p_{1j}$ ;
 $p \leftarrow 1, q \leftarrow 0, g_1 \leftarrow J_z, S \leftarrow \bar{J} \setminus g_1, Q \leftarrow \emptyset, R \leftarrow \emptyset, O^{OG} \leftarrow g_1$ ;
while  $S \neq \emptyset$  do
    Soit  $I \leftarrow J_j \in S | p_{1j} \leq p_{2z} + b(R) - a(R)$ ;
    if  $I \neq \emptyset$  then
         $q \leftarrow q + 1$ ;
        Soit  $\theta_p^q$  un sous-ensemble arbitraire non vide de  $I$ ;
         $R \leftarrow R \prec \theta_p^q$ ;
         $O^{OG} \leftarrow O^{OG} \prec \theta_p^q$ ;
         $S \leftarrow S \setminus \theta_p^q$ ;
    end
    else
         $Q \leftarrow Q \prec J_z \prec R$ ;
         $n_p \leftarrow q$ ;
         $p \leftarrow p + 1$ ;
         $q \leftarrow 0$ ;
        Choisir  $J_z \in S$  avec  $p_{1z} = \min_{J_j \in S} p_{1j}$ ;
         $g_p \leftarrow J_z$ ;
         $R \leftarrow \emptyset$ ;
         $O^{OG} \leftarrow O^{OG} \prec g_p; S \leftarrow S \setminus g_p$ ;
    end
end
 $l \leftarrow p$ ;
 $n_l \leftarrow q$ ;

```

Pour l'ensemble de  $\bar{J}$ , l'algorithme *OG* génère un ordre partiel total sous la forme :

$$O^{OG} = g_1 \prec \theta_1^1 \prec \theta_1^2 \dots \prec \theta_1^{n_1} \prec g_2 \prec \theta_2^1 \dots \prec \theta_2^{n_2} \dots \prec g_l \prec \theta_l^1 \dots \prec \theta_l^{n_l}$$

Le nombre de solutions équivalentes générées à partir de  $O^{OG}$  peut être calculé par la formule :  $\prod_{p=1}^l \prod_{q=1}^{n_p} |\theta_p^q|!$ , où  $|\theta_p^q|$  est la cardinalité de groupe  $\theta_p^q$ . Pour une sous-séquence  $\bar{\pi}$  qui suit le  $O^{OG}$ , nous avons :  $C_{max}(\bar{\pi}) = a(Q) + p_{1z} + p_{2z} + b(R)$ , avec  $Q$  et  $R$  sont des ensembles de tâches qui précèdent et succèdent respectivement le "crossover job" de  $\bar{\pi}$  (dans  $\bar{\pi}$  le "crossover job" est le dernier "ground job").

Pour l'ensemble de  $\underline{J}$ , l'algorithme *OG* permet également de générer un ordre partiel total de  $O^{OG'}$  en inversant la configuration des machines, la machine 1 devient la machine 2 et inversement. Alors pour une instance de  $(F2|perm|C_{max})$  avec deux ensembles  $\bar{J}$  et  $\underline{J}$ , nous pouvons toujours obtenir  $O^{OG}$  pour  $\bar{\pi}$  et  $O^{OG'}$  pour  $\underline{\pi}$  avec l'algorithme *OG*.

**Théorème 1.** *Pour toutes les solutions  $\pi = \bar{\pi} \prec \underline{\pi}$  qui suivent l'ordre partiel total SPO :  $O^{OG} \prec O^{OG'}$  sont les solutions optimale de  $F2|perm|C_{max}$ . [Hadda et al. (2018)].*

Selon le théorème (1), l'ordre partiel total pour la solution optimale du problème de  $(F2|perm|C_{max})$  est  $SPO : O^{OG} \prec O^{OG'}$ . Une séquence  $\pi = \bar{\pi} \prec \underline{\pi}$  qui suit SPO a un makespan optimal de :

$$C_{max}(\pi) = a(Q) + p_{1z} + \max\{p_{2z} + b(R) + b(R'), a(R) + a(R') + p_{1z'}\} + p_{2z'} + b(Q')$$

Notons  $J_{z'}, Q', R'$  sont respectivement le “crossover job”, l'ensemble des tâches qui précèdent  $J_{z'}$ , l'ensemble des tâches qui succèdent  $J_{z'}$  de  $O^{OG'}$ .

Nous avons également le temps d'inactivité qui apparaît soit sur la machine 1 soit sur la machine 2 quand nous enchaînons  $\bar{\pi}$  et  $\underline{\pi}$  :

$$\Delta = |p_{2z} + b(R) + b(R') - (a(R) + a(R') + p_{1z'})|$$

### 3.3.1.4 Extension de l'ensemble des solutions optimales équivalentes

Comme nous avons présenté dans les sections précédentes, l'algorithme  $OG$  permet de construire un ensemble très large des solutions optimales équivalentes pour le problème  $(F2|perm|C_{max})$ . Cependant, l'ordre partiel total ( $SPO$ ) généré par l'algorithme  $OG$  présentent deux caractéristiques qui limitent l'espace de solutions. Premièrement, le premier et le dernier “ground job” de  $SPO$  sont fixés. Deuxièmement, le  $SPO$  de  $OG$  a une forme unique  $SPO = \bar{\pi} \prec \underline{\pi}$ . Nous présentons des propositions pour étendre l'ensemble de solutions optimales équivalentes en se fondant sur ces deux caractéristiques.

Tout d'abord, nous présentons quelques notations et définitions qui sont nécessaires pour nos propositions. Ensuite, nous présentons deux approches pour étendre l'ensemble de solutions de  $OG$  : le changement de premier “ground job” et le changement des tâches entre  $\bar{J}$  et  $\underline{J}$ .

Pour un SPO généré par l'algorithme  $OG$ , on note  $J_z$  est le “crossover job”,  $J_z^1$  l'ensemble des tâches qui précèdent  $J_z$ ,  $J_z^2$  l'ensemble des tâches qui succèdent  $J_z$ . Alors, en se fondant sur le positionnement de  $J_z$ , SPO peut être exprimé sous la forme :  $SPO = J_z^1 \prec J_z \prec J_z^2$ . Nous présentons les études pour l'ensemble  $J_z^1$ , les résultats restent valable pour l'ensemble  $J_z^2$  en inversant la configuration grâce à la réversibilité du problème  $F2|perm|C_{max}$ .

Pour l'ensemble  $J_z^1$ , soit  $\tau_u$  le temps d'inactivité total sur la machine 2 entre le “ground job”  $g_u \in J_z^1$  et le “crossover job”  $J_z$  :

$$\tau_u = \begin{cases} \sum_{p=u+1}^x \delta_p, si J_z \in \bar{J}. \\ \Delta + \sum_{p=u+1}^l \delta_p, sinon \end{cases}$$

Avec,  $J_z = g_x$  : le “crossover job” est le  $x^{eme}$  “ground job” de SPO.

#### Changement du premier “ground job” :

L'algorithme  $OG$  ne permute que l'ordre des tâches à l'intérieur de chaque groupe. Il ne change ni la date de début ni la date de fin de chaque “ground job”  $g_p$  ni le temps d'inactivité  $\delta_p$  qui précède  $g_p$  sur la machine 2. Le “crossover job”

est un “ground job” particulier, sa date de début sur la machine 2 ne dépend donc que de sa date de fin sur la machine 1. Si nous changeons l’ordre des tâches qui précèdent le “crossover job” à conditions que le changement n’impacte pas la date de début sur machine 2 du “crossover job”, alors sa date de fin et celles des tâches qui le succèdent restent inchangées. Par conséquent, le makespan n’est pas affecté. L’idée est d’effectuer les permutations de tâches qui modifient la quantité  $\delta_p$  sans affecter le “crossover job”.

**Théorème 2.** *Pour une séquence donnée par l’algorithme OG,  $J_z = g_x$  est un “crossover job”. Soit  $J_w \in \bar{J}$  une tâche qui précède  $J_z$ ,  $g_v$  est son “ground job” associé. S’il existe  $g_s (s \leq v)$  un “ground job” avec  $(p_{1w} - p_{1s}^g \leq \tau_s)$ . Alors toutes les solutions de l’algorithme OG en insérant  $J_w$ :*

- *Juste avant un “ground job” quelconque  $g_u, u \in \{s, \dots, v\}$ .*
- *Ou dans n’importe quel groupe  $\theta_u^q, u \in \{s, \dots, v\}, q \in \{1, \dots, n_u\}$*

*sont optimales. [Hadda et al. (2018)].*

Pour construire une sous-séquence  $\bar{\pi}$ , l’algorithme OG choisit toujours pour le premier “ground job” la tâche dont le temps de traitement sur la machine 1 est le plus petit. Ce choix permet de réduire le temps d’inactivité de la machine 2. Le théorème (2) nous fournit une condition pour modifier cette caractéristique de SPO généré par OG afin de pouvoir étendre l’ensemble des solutions optimales équivalentes.

A partir d’un SPO, nous pouvons calculer la quantité  $\tau_j$  de la tâche  $J_j \in J_z^1$ . Si la tâche  $J_j \in J_z^1 \cap \bar{J}$ , alors nous calculons la différence entre son temps de traitement sur la machine 1 avec celui de premier “ground job”  $g_1$  :  $(p_{1j} - p_{11}^g)$ . Si  $(p_{1j} - p_{11}^g \leq \tau_1)$ , alors nous ajoutons cette tâche dans un ensemble  $U$ . L’ensemble  $U$  contient des tâches qui peuvent être positionnées précédemment à  $g_1$  pour former une nouvelle sous-séquence  $\bar{\pi}'$  sans augmenter la valeur du makespan. Ensuite, nous appliquons l’algorithme ESG pour obtenir de nouveaux SPO dont le premier “ground job” appartient à l’ensemble  $U$ .

**Algorithme 2 :** L'algorithme *ESG* proposé par (Hadda et al. (2018))

Identifier à partir de sous-séquence  $\bar{\pi}$  les “ground job”  $g_p$  et les groupes de tâches  $\theta_p, p \in \{1, \dots, l\}$  associés;

```

for  $p = 1$  to  $l$  do
     $q \leftarrow 1, S \leftarrow \theta_p, Q \leftarrow \emptyset;$ 
    while  $S \neq \emptyset$  do
        Soit  $I \leftarrow \{J_j \in S | p_{1j} \leq p_{2p}^g + b(Q) - a(Q)\};$ 
        Soit  $\theta_p^q$  un sous-ensemble non vide de  $I;$ 
         $Q \leftarrow Q \prec \theta_p^q;$ 
         $S \leftarrow S \setminus \theta_p^q;$ 
         $q \leftarrow q + 1;$ 
    end
     $n_p \leftarrow q;$ 
end
 $n_l \leftarrow q;$ 

```

L'algorithme *ESG* permet de décomposer une sous-séquence  $\bar{\pi}$  donnée en “ground job” et groupe de tâches associées. Ensuite, l'algorithme construit un *SPO* qui donne un ensemble de solutions ayant le même makespan de  $C_{max}(\bar{\pi})$ . Comme le premier job de  $\bar{\pi}$  est une tâche  $J_j \in U$ , alors le premier “ground job” de *SPO* généré par *ESG* est une tâche de  $U$  qui fait la différence avec *SPO* généré par l'algorithme *OG*.

Nous formalisons l'extension de l'ensemble des solutions optimales équivalentes par changement de premier “ground job” de *SPO* par un algorithme nommé “Change” ci-dessous. L'algorithme “Change” traite tout d'abord la sous-séquence  $\bar{\pi}$ . Pour traiter la sous-séquence  $\underline{\pi}$ , nous appliquons également l'algorithme “Change” en inversant la configuration. L'application de l'algorithme “Change” sur  $\underline{\pi}$  permet de changer le dernier “ground job” de *SPO*.

**Algorithme 3 :** L'algorithme de changement de premier “ground job”

**Data :** L'ordre partiel total *SPO*,  $\bar{\pi}$ , “crossover job”  $J_z = g_x$ .

**Result :** Nouveaux *SPO'*, nouvelles  $\bar{\pi}'$ .

$U \leftarrow \emptyset;$

Calculer la quantité  $\tau_1$  de “ground job”  $g_1 : \tau_1 = \begin{cases} \sum_{p=2}^x \delta_p, \text{ si } J_z \in \bar{J}. \\ \Delta + \sum_{p=2}^l \delta_p, \text{ sinon} \end{cases}$

Pour chaque tâche  $J_j \in J_z^1 \cap \bar{J}$  :

**if**  $p_{1j} - p_{11}^g \leq \tau_1$  **then**  
 |  $U \leftarrow J_j$

**end**

Pour chaque tâche  $J_j \in U$  :

Positionner  $J_j$  avant  $g_1$  de *SPO* pour obtenir nouvelle séquence  $\pi'$  ;

Appliquer l'algorithme *ESG* à  $\pi'$  pour obtenir nouveau *SPO'*, nouvelle  $\bar{\pi}'$ ;

Nous prenons un exemple de 12 tâches (tableau (9)) pour illustrer l’algorithme ESG et l’extension de l’ensemble des solutions optimales équivalentes. Nous avons deux ensembles  $\bar{J} = \{J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8, J_9, J_{10}\}$  et  $\underline{J} = \{J_{11}, J_{12}\}$ . En appliquant l’algorithme *OG* sur  $\bar{J}$ , nous obtenons  $\bar{\pi} = J_1 \prec \{J_2, J_3\} \prec \{J_4, J_5\} \prec J_6 \prec \{J_7, J_8\} \prec \{J_9, J_{10}\}$ . En appliquant l’algorithme *OG'* sur  $\underline{J}$ , nous obtenons  $\underline{J} = J_{11} \prec J_{12}$ . Alors l’ordre partiel total est  $SPO = \bar{\pi} \prec \underline{\pi} = J_1 \prec \{J_2, J_3\} \prec \{J_4, J_5\} \prec J_6 \prec \{J_7, J_8\} \prec \{J_9, J_{10}\} \prec J_{11} \prec J_{12}$ , ce qui donne 16 solutions optimales équivalentes.

La Fig.(10) représente l’ordre partiel total *SPO* avec  $\delta_1 = 2, \delta_2 = 2$  et  $\Delta = 9$  et  $J_{11}$  est le “crossover job”. Pour la tâche  $J_7$ , nous avons  $p_{17} - p_{11}^g = p_{17} - p_{11} = 10$ . La valeur  $\tau_1$  associée à “ground job”  $g_1 : \tau_1 = \delta_1 + \Delta = 11$ . Comme  $p_{17} - p_{11}^g \leq \tau_1$  cela satisfait le théorème (2), alors nous pouvons positionner  $J_7$  avant  $J_1$  sans augmenter la valeur du makespan. Nous obtenons une nouvelle sous-séquence  $\bar{\pi}' = J_7 \prec J_1 \prec J_2 \prec J_3 \prec J_4 \prec J_5 \prec J_6 \prec J_8 \prec J_9 \prec J_{10}$ . En appliquant *ESG* sur  $\bar{\pi}'$ , nous obtenons un nouvel ordre partiel  $J_7 \prec \{J_1, J_2, J_3, J_4, J_5, J_6, J_8\} \prec \{J_9, J_{10}\}$  qui donne  $7! \times 2! = 10080$  solutions optimales équivalentes.

Tableau 9 – Le temps de traitement de l’exemple 2

Tâche	1	2	3	4	5	6	7	8	9	10	11	12
<b>Machine 1</b>	1	2	2	4	5	10	11	11	15	15	27	3
<b>Machine 2</b>	2	3	4	6	6	12	12	13	16	17	4	2

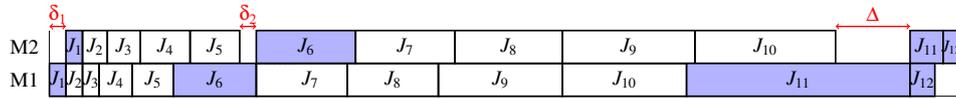


Figure 10 – L’ordre partiel total *SPO* de l’exemple 2 (M1, M2 sont les machines 1 et 2 respectivement).

### Le changement des tâches entre $\bar{J}$ et $\underline{J}$ :

La solution de Johnson [Johnson (1954)] et les solutions optimales équivalentes trouvées par [Hadda et al. 2018] pour le problème  $(F_2|perm|C_{max})$  sont sous la forme  $\pi = \bar{\pi} \prec \underline{\pi}$ . Cela signifie que dans les solutions optimales trouvées, les tâches de l’ensemble  $\bar{J}$  précèdent toujours les tâches de l’ensemble  $\underline{J}$ . Alors, une quantité considérable de solutions optimales ne seront pas étudiées dans leurs algorithmes.

Par exemple dans Fig.(11), nous avons un ensemble  $J = \{J_1, J_2, J_3\}$ . Les temps de traitement de ces tâches sont donnés dans le tableau (10). Nous classons ces tâches en deux sous-ensembles  $\bar{J} = \{J_1\}$  et  $\underline{J} = \{J_2, J_3\}$  Nous comparons ici deux séquences équivalentes  $\pi_1$  et  $\pi_2$ . La séquence  $\pi_1$  est illustrée dans Fig.(11a), la séquence  $\pi_2$  est décrite dans Fig.(11b). La séquence  $\pi_1$  est de la forme  $\bar{\pi} \prec \underline{\pi} = J_1 \prec J_3 \prec J_2$  et a un makespan de 328. Pendant que dans  $\pi_2$ , la tâche  $J_3 \in \underline{J}$  précède la tâche  $J_1 \in \bar{J}$ ,  $\pi_2$  a le même makespan de celui de la séquence  $\pi_1$ .

Tableau 10 – Le temps de traitement de l'exemple 3

Tâche	1	2	3
Machine 1	108	166	30
Machine 2	158	24	26

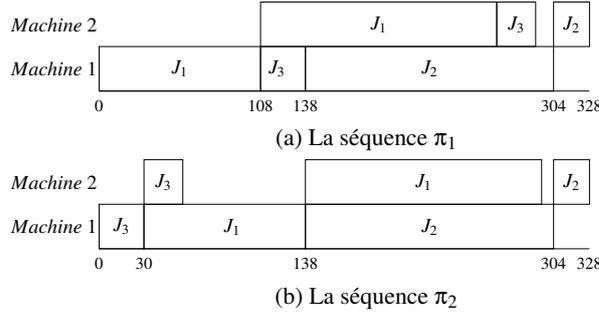


Figure 11 – Les séquences  $\pi_1$  et  $\pi_2$  de l'exemple 3

Pour un ordre partiel total  $SPO$  généré par l'algorithme  $OG$ ,  $SPO = J_z^1 \prec J_z \prec J_z^2$ . Pour rappel,  $J_z$  est le "crossover job",  $J_z^1$  et  $J_z^2$  contiennent des tâches qui précèdent  $J_z$  et succèdent  $J_z$  respectivement. Si  $J_z^1$  ne contient que des tâches venant soit de  $\bar{J}$  soit de  $\underline{J}$ , alors nous examinons le changement de premier "ground job" précisé auparavant. Si  $J_z^1$  contient des tâches venant de  $\bar{J}$  et de  $\underline{J}$ , nous divisons  $J_z^1$  en deux sous-ensembles  $\bar{J}_z^1$  et  $\underline{J}_z^1$ , ( $\forall J_j \in \bar{J}_z^1, J_j \in \bar{J}$ ) et ( $\forall J_j \in \underline{J}_z^1, J_j \in \underline{J}$ ). Dans ce cas, nous avons une condition qui permet de transformer des tâches de  $\underline{J}_z^1$  et de placer ces tâches transformées en  $\bar{J}_z^1$  sans impacter le makespan. Cela permet de trouver d'autres solutions optimales qui ne suivent pas la forme  $\pi = \bar{\pi} \prec \underline{\pi}$ .

Soit  $\pi_z^1$  une séquence de  $J_z^1$  dont  $\bar{J}_z^1 \neq \emptyset$  et  $\underline{J}_z^1 \neq \emptyset$ ,  $n_{\pi_z^1}$  est le nombre de tâches dans  $\pi_z^1$  et  $\pi_z^1(j)$  signifie la tâche à la  $j^{eme}$  position dans  $\pi_z^1$ . Supposons que toutes les tâches de  $\pi_z^1$  commencent le plus tôt possible. Nous pouvons calculer le temps d'inactivité sur la machine 2 ( $\delta_j^1$ ) qui se situe entre deux tâches  $\pi_z^1(j)$  et  $\pi_z^1(j+1)$  :

$$\delta_j^1 = S_{2\pi_z^1(j+1)} - C_{2\pi_z^1(j)}, \forall j \neq n_{\pi_z^1}$$

Pour la dernière tâche de  $\pi_z^1$ ,  $\delta_{n_{\pi_z^1}}^1 = S_{2z} - C_{2\pi_z^1(n_{\pi_z^1})}$ , avec  $S_{2z}$  la date de début sur la machine 2 de "crossover job"  $J_z$ .

Nous définissons la marge ( $\mu_j$ ) de tâche  $\pi_z^1(j)$ . La marge ( $\mu_j$ ) d'une tâche est la quantité de temps maximum dont nous pouvons retarder le traitement sur la machine 2 sans impacter le makespan. Nous avons :

$$\mu_j = \begin{cases} \delta_{n_{\pi_z^1}}^1, & \text{si } j = n_{\pi_z^1} \\ \sum_{j'} \delta_{j'}^1, & \text{sinon} \end{cases}$$

**Proposition 1.**

Soit  $\pi_z^1$ , une séquence de  $J_z^1$  dont  $\overline{J_z^1} \neq \emptyset$  et  $\underline{J_z^1} \neq \emptyset$ . Pour une tâche  $J_j \in \underline{J_z^1}$ , si  $(p_{1j} - p_{2j} \leq \mu_j)$ , alors nous pouvons remplacer  $J_j$  par une tâche  $J_k$  dont  $p_{1k} = p_{2k} = p_{1j}$  sans augmenter  $C_{max}(\pi_z^1)$ .

*Démonstration.* Comme  $p_{1\pi_z^1(1)}$  équivaut à temps d'inactivité de la machine 2 avant de traiter la première tâche, donc  $(p_{1\pi_z^1(1)} + \sum_{j=1}^{n_{\pi_z^1}} \delta_j)$  équivaut à temps d'inactivité total de la machine 2. Alors, la valeur de makespan de  $\pi_z^1$  peut être exprimée par :

$$C_{max}(\pi_z^1) = p_{1\pi_z^1(1)} + \sum_{j=1}^{n_{\pi_z^1}} p_{2j} + \sum_{j=1}^{n_{\pi_z^1}} \delta_j \quad (16)$$

Pour une tâche  $J_j \in \underline{J_z^1}$ , nous pouvons réécrire l'équation (16) par :

$$C_{max}(\pi_z^1) = p_{1\pi_z^1(1)} + \sum_{q=1}^{j-1} p_{2q} + p_{2j} + \sum_{q=j+1}^{n_{\pi_z^1}} p_{2q} + \sum_{q=1}^{j-1} \delta_q + \mu_j. \quad (17)$$

Nous avons  $(p_{1j} - p_{2j} \leq \mu_j)$ , alors  $\mu_j = p_{1j} - p_{2j} + a$ , avec  $a \geq 0$ . L'équation (17) devient :

$$\begin{aligned} C_{max}(\pi_z^1) &= p_{1\pi_z^1(1)} + \sum_{q=1}^{j-1} p_{2q} + p_{2j} + \sum_{q=j+1}^{n_{\pi_z^1}} p_{2q} + \sum_{q=1}^{j-1} \delta_q + p_{1j} - p_{2j} + a. \\ &= p_{1\pi_z^1(1)} + \sum_{q=1}^{j-1} p_{2q} + \sum_{q=j+1}^{n_{\pi_z^1}} p_{2q} + \sum_{q=1}^{j-1} \delta_q + p_{1j} + a. \\ &= p_{1\pi_z^1(1)} + \sum_{q=1}^{j-1} p_{2q} + \sum_{q=j+1}^{n_{\pi_z^1}} p_{2q} + \sum_{q=1}^{j-1} \delta_q + p_{2k} + a. \end{aligned}$$

Alors, si  $(p_{1j} - p_{2j} \leq \mu_j)$ , alors nous pouvons remplacer  $J_j$  par une tâche  $J_k$  dont  $p_{1k} = p_{2k} = p_{1j}$  sans augmenter  $C_{max}(\pi_z^1)$ .  $\square$

Selon la proposition (1), si une tâche  $J_j \in \underline{J_z^1}$  et  $(p_{1j} - p_{2j} \leq \mu_j)$ , alors nous pouvons la remplacer par une tâche  $J_k$  dont  $p_{1k} = p_{2k} = p_{1j}$  sans augmenter  $C_{max}(\pi_z^1)$ . Comme  $p_{1k} = p_{2k}$  alors  $J_k$  peut être classée arbitrairement soit dans  $\overline{J}$  soit dans  $\underline{J}$ . Nous classons  $J_k$  à  $\overline{J}$  dans le but de faire la différence avec  $J_j$  qui était dans  $\underline{J}$ . Alors nous ajoutons  $J_k$  à  $\overline{J}$  qui formera l'ensemble  $\overline{J'}$  et nous enlevons  $J_j$  de  $\underline{J}$  pour former l'ensemble  $\underline{J'}$ .

Nous appliquons l'algorithme  $OG$  à  $\overline{J'}$  et l'algorithme  $OG'$  à  $\underline{J'}$ . Nous obtenons un nouvel ordre partiel total  $SPO^*$  qui a la même valeur de makespan de  $C_{max}(\pi_z^1)$ . A la fin, pour chaque solution obtenue par  $SPO^*$ , nous remplaçons  $J_k$  par la tâche  $J_j$ . Le remplacement de  $J_k$  par  $J_j$  n'augmente pas la valeur de makespan car  $(p_{2j} \leq$

$p_{1j} = p_{2k}$ ). Alors, la tâche  $J_j$  qui était dans  $\underline{J}_z^1$  appartient à  $\overline{J}_z^1$  dans la nouvelle séquence.

Nous proposons l'algorithme "Change2" qui exploite la proposition (1) pour étendre l'ensemble des solutions optimales équivalentes. L'algorithme "Change2" traite l'ensemble  $J_z^1$  dont  $\overline{J}_z^1 \neq \emptyset$  et  $\underline{J}_z^1 \neq \emptyset$ , nous notons  $n_{\pi_z^1}^2$  le nombre de tâches dans  $\underline{J}_z^1$ . Pour l'ensemble  $J_z^2$  dont  $\overline{J}_z^2 \neq \emptyset$  et  $\underline{J}_z^2 \neq \emptyset$ , nous appliquons l'algorithme "Change2" en inversant la configuration du problème.

**Algorithme 4 :** L'algorithme de changement des tâches entre  $\overline{J}$  et  $\underline{J}$

**Data :**  $J_z^1, \overline{J}_z^1 \neq \emptyset, \underline{J}_z^1 \neq \emptyset, \pi_z^1, n_{\pi_z^1}, n_{\pi_z^1}^2$

**Result :**  $SPO^*$

$\overline{U} \leftarrow 0 ; \underline{U} \leftarrow 0 ;$

Obtenir  $S_{2j}, C_{2j}$  quand les tâches  $J_j$  de  $J_z^1$  commence au plus tôt possible ;

**for**  $j = 1$  to  $n_{\pi_z^1}$  **do**

    Calculer le temps d'inactivité  $\delta_j^1$  :

$$\delta_j^1 = \begin{cases} S_{2\pi_z^1(j+1)} - C_{2\pi_z^1(j)}, \forall j \neq n_{\pi_z^1}. \\ S_{2z} - C_{2\pi_z^1(n_{\pi_z^1})}, si j = n_{\pi_z^1}. \end{cases}$$

**end**

**for**  $j = 1$  to  $n_{\pi_z^1}$  **do**

    Calculer la marge  $\mu_j$  :

$$\mu_j = \begin{cases} \delta_{n_{\pi_z^1}}^1, si j = n_{\pi_z^1}. \\ \sum_{j=1}^{n_{\pi_z^1}} \delta_j^1, sinon \end{cases}$$

**end**

**for**  $j = n_{\pi_z^1}$  to  $n_{\pi_z^1} - n_{\pi_z^1}^2$  **do**

**if**  $(p_{1j} - p_{2j} \leq \mu_j)$  **then**

        Créer une tâche  $J_k$  dont  $p_{1k} = p_{2k} = p_{1j}$  ;

$\overline{U} \leftarrow J_k ; \underline{U} \leftarrow J_j ;$

**for**  $q = j$  to  $n_{\pi_z^1} - n_{\pi_z^1}^2$  **do**

$\mu_q = \mu_q - (p_{1j} - p_{2j}) ;$

**end**

**end**

**end**

$\overline{J}' \leftarrow \overline{U} ;$

$\underline{J}' \setminus \underline{U} ;$

Appliquer l'algorithme  $OG$  sur  $\overline{J}'$  et l'algorithme  $OG'$  sur  $\underline{J}'$  ;

Pour  $SPO^*$  obtenu, remplacer les tâches  $J_k$  de  $\overline{U}$  par la tâche  $J_j$  associée ;

L'algorithme (4) détermine tout d'abord la date de début et la date de fin des

tâches de la séquence  $\pi_z^1$  quand ces tâches commencent au plus tôt. Les deux ensembles  $\bar{U}$  et  $\underline{U}$  servent à contenir des tâches que nous ajoutons à  $\bar{J}_z^1$  ou enlevons de  $\underline{J}_z^1$ . Ensuite, l'algorithme calcule le temps d'inactivité  $\delta_j^1$  de tâche  $J_j \in \pi_z^1$ . Le temps d'inactivité  $\delta_j^1$  est utilisé pour calculer la marge  $\mu_j$ . En commençant par la dernière tâche  $J_{n_{\pi_z^1}}$  de  $\pi_z^1$ , nous examinons par récurrence chaque tâche  $J_j \in \underline{J}_z^1$ . Si la tâche  $J_j$  satisfait la condition de la proposition (1), alors nous ajoutons une tâche  $J_k$  dont  $p_{1k} = p_{2k} = p_{1j}$  à l'ensemble  $\bar{U}$  et nous incluons  $J_j$  à l'ensemble  $\underline{U}$ . Puis, nous mettons à jour la marge des tâches  $J_q \in \underline{J}_z^1$  qui précèdent  $J_j$ ,  $\mu_q = \mu_q - (p_{1j} - p_{2j})$ . Nous recommençons la procédure jusqu'à ce que toutes les tâches de  $\underline{J}_z^1$  soit examinées. Nous appliquons l'algorithme *OG* à  $\bar{J}'$  et l'algorithme *OG'* à  $\underline{J}'$ . Nous obtenons un nouvel ordre partiel total *SPO\** qui a la même valeur de makespan de  $C_{max}(\pi_z^1)$ . A la fin, pour chaque solution obtenue par *SPO\**, nous remplaçons  $J_k$  par la tâche  $J_j$  correspondante.

Nous prenons l'exemple 4 (tableau (11)) pour illustrer l'algorithme "Change2". A partir des données de temps de traitement dans le tableau (11), nous avons un ordre partiel total *SPO* =  $J_1 \prec J_2 \prec \{J_3\} \prec J_4 \prec \{J_5, J_6, J_7\} \prec J_8$ . Avec une séquence  $\pi = J_1 \prec J_2 \prec J_3 \prec J_4 \prec J_5 \prec J_6 \prec J_7 \prec J_8$  (Fig.(12a)), nous avons le "crossover job"  $J_8$ . Les tâches  $J_1, J_2, J_4$  sont les "ground job". La sous-séquence  $\pi_z^1 = J_1 \prec J_2 \prec J_3 \prec J_4 \prec J_5 \prec J_6 \prec J_7$  dont  $\bar{J}_z^1 = \{J_1, J_2, J_3, J_4\}$ ,  $\underline{J}_z^1 = \{J_5, J_6, J_7\}$ . Les temps d'inactivité  $\delta_j^1$  associés à chaque tâche sont :  $\delta_1^1 = 1$ ,  $\delta_3^1 = 2$ ,  $\delta_5^1 = 2$ ,  $\delta_6^1 = 5$  et  $\delta_7^1 = 5$ .

A partir des temps d'inactivité  $\delta_j^1$ , nous calculons la marge associée à chaque tâche. Nous nous intéressons aux tâches  $J_5, J_6$  et  $J_7$  qui appartiennent à  $\underline{J}_z^1$  :  $\mu_7 = \delta_7^1 = 5$ ,  $\mu_6 = \delta_6^1 + \delta_7^1 = 10$  et  $\mu_5 = \delta_5^1 + \delta_6^1 + \delta_7^1 = 12$ . Nous examinons par récurrence chaque tâche de  $\underline{J}_z^1$  à partir de la dernière tâche de  $\pi_z^1$  :  $J_7$ . Nous avons :  $p_{17} - p_{27} = 7 \geq \mu_7 = 5$ , donc la condition de proposition (1) n'est pas satisfaite. Ensuite, pour la tâche  $J_6$ , nous avons :  $p_{16} - p_{26} = 4 \leq \mu_6 = 10$ . La condition de proposition (1) est satisfaite, alors nous remplaçons  $J_6$  par  $J'_6$  dont le temps de traitement sur les deux machines sont égaux. Nous mettons à jour la marge des tâches qui précèdent  $J_6$  :  $\mu_5 = 12 - (p_{16} - p_{26}) = 8$ . Nous avons  $p_{15} - p_{25} = 5 \leq \mu_5 = 8$ , alors nous remplaçons  $J_5$  par  $J'_5$ . Les remplacements des tâches sont illustrés dans la Fig.(12b).

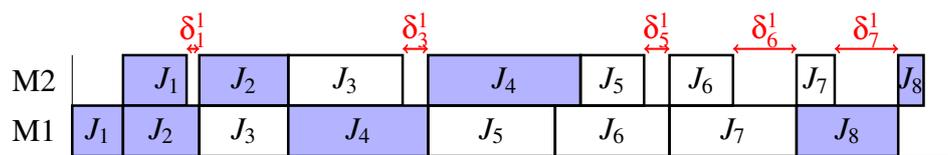
Après avoir remplacé les tâches, nous obtenons deux nouveaux ensembles  $\bar{J}' = \bar{J}_z^1 \cup \{J'_6, J'_5\}$  et  $\underline{J}' = \underline{J}_z^1 \setminus \{J_6, J_5\}$ . En appliquant l'algorithme *OG* sur ces deux ensembles, nous obtenons un nouvel ordre partiel total *SPO\** (Fig.(12c)). A la fin, nous remplaçons les tâches  $J'_6, J'_5$  par  $J_6$  et  $J_5$  respectivement pour obtenir la solution finale. Dans la Fig.(12d), nous avons la solution  $\pi' = J_1 \prec J_2 \prec J_3 \prec J_6 \prec J_5 \prec J_4 \prec J_7 \prec J_8$ . Les tâches  $J_6$  et  $J_5$  qui appartiennent à  $\underline{J}_z^1$  précèdent la tâche  $J_4 \in \bar{J}_z^1$  et n'impacte pas le makespan.

### La recherche locale

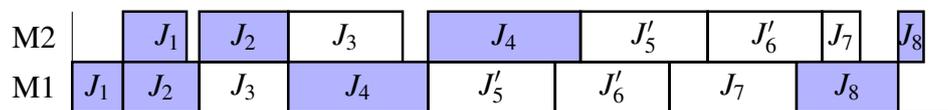
Nous proposons une recherche locale qui nous permet de trouver autres so-

Tableau 11 – Le temps de traitement de l'exemple 4

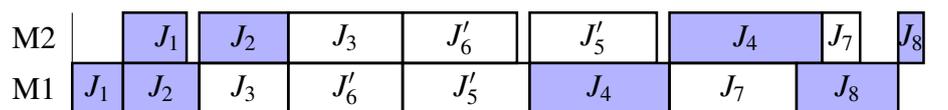
Tâche	1	2	3	4	5	6	7	8
<b>Machine 1</b>	4	6	7	11	10	9	10	8
<b>Machine 2</b>	5	7	9	12	5	5	3	2



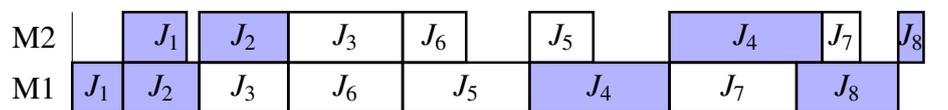
(a) La séquence  $\pi_z^1$  avec le "crossover job"  $J_8$



(b) La transformation des tâches de  $J_z^1$



(c) Le nouveau ordre partiel total  $SPO^*$  avec  $J'_7$  et  $J'_6$



(d) La nouvelle séquence obtenue avec l'algorithme "Change2"

Figure 12 – L'exemple de l'algorithme "Change2"

lutions optimales équivalentes à partir d'une solution optimale donnée. Pour une séquence  $\pi = \langle \pi_1 \prec \pi_2 \prec \dots \prec \pi_n \rangle$ ,  $\pi_j$  signifie la tâche à la  $j^{eme}$  position dans  $\pi$ . Pour chaque tâche de la position  $\pi_1$  à  $\pi_n$ , nous permutons cette tâche avec une autre tâche qui la succède. Si la nouvelle solution obtenue a la même valeur de makespan, alors nous l'ajoutons dans l'ensemble des solutions optimales équivalentes.

La recherche locale est donnée dans l'algorithme (5). Nous pouvons appliquer cette recherche locale soit sur la solution de Johnson soit sur les solutions données par l'algorithme *OG* de [Hadda et al. (2018)]. Elle permet de trouver autres solutions optimales qui ne respectent pas la forme  $\pi = \bar{\pi} \prec \underline{\pi}$  en  $O(\frac{n(n-1)}{2})$  étapes.

**Algorithme 5 :** La recherche locale pour des solutions optimales équivalentes

```

Pour une sequence  $\pi = \langle \pi_1 \prec \pi_2 \prec \dots \prec \pi_n \rangle$  de règle de Johnson ou de
l'algorithme de Hadda;
for  $i = 1$  to  $i = n$  do
    for  $j = i$  to  $j = n$  do
        Permuter la position entre  $\pi_i$  et  $\pi_j$  pour obtenir la nouvelle solution
         $\pi'$ ;
        Evaluer la valeur de makespan de  $\pi'$ ;
        if  $C_{max}(\pi')$  est optimal then
            Ajouter  $\pi'$  dans l'ensemble des solutions optimales
            équivalentes;
        end
    end
end

```

### 3.3.2 Le problème $F2, ToU|sequence, Pu_{ij}|TEC$ .

Après avoir obtenu l'ensemble des solutions optimales équivalentes à l'étape précédente, pour chaque solution obtenue, nous avons besoins de déterminer la date de début optimale de ses tâches. Par conséquent, nous étudions le problème d'ordonnancement écoénergétique de flow shop à deux machines avec une séquence fixée et une consommation énergétique différente entre les tâches. Ce problème peut être noté par  $(F2, ToU|sequence, Pu_{ij}|TEC)$ , dont  $Pu_{ij}$  représente la puissance requise pour traiter la tâche  $j$  sur la machine  $i$ . Nous montrons dans les sections suivantes que ce problème peut être résolu de manière optimale en temps polynomial.

**Théorème 3.** *Le problème  $(F2, ToU|sequence, Pu_{ij}|TEC)$  peut être décomposé en deux problèmes d'ordonnancement sur une seule machine avec une séquence fixe, une fenêtre de temps donnée ( $TW$ ) pour la minimisation du coût total de l'électricité ( $TEC$ ) sous la tarification d'électricité  $ToU$ , notée  $1, ToU|sequence, TW, Pu_{ij}|TEC$ , un sur la machine 1 et un sur la machine 2, et il peuvent être résolu de manière optimale en temps polynomial ( $O(T^3)$ ).*

*Démonstration.*

Pour une séquence donnée  $\pi$ , nous pouvons toujours déterminer le “crossover job”  $J_z$ , l’ensemble  $J_z^1$  des tâches qui précèdent  $J_z$  et l’ensemble  $J_z^2$  des tâches qui succèdent  $J_z$ . Selon la définition de “crossover job”, nous avons :

$$J_z = \pi_x | C_{max}(\pi) = \sum_{i=1}^x p_{1\pi_i} + \sum_{i=x}^n p_{2\pi_i}$$

Nous pouvons montrer facilement que :  $\sum_{i=1}^{x-1} p_{2\pi_i} \leq \sum_{i=2}^x p_{1\pi_i}$  et  $\sum_{i=x+1}^n p_{1\pi_i} \leq \sum_{i=x}^{n-1} p_{2\pi_i}$ . Alors, chaque tâche  $J_j \in J_z^1$  a une fenêtre de temps pour sa date de début sur la machine 2. La fenêtre de temps de chaque tâche  $J_j = \pi_j \in J_z^1$  est déterminée par sa date de début au plus tôt ( $EST_{2j}$ ) sur la machine 2 :  $EST_{2j} = \max(C_{1\pi_j}, C_{2\pi_{j-1}})$  et par sa date de début au plus tard ( $LST_{2j}$ ) sur la machine 2 :  $LST_{2j} = S_{2\pi_x} - \sum_{i=j+1}^{x-1} p_{2j}$ . Respectivement, chaque tâche  $J_j \in J_z^2$  a une fenêtre de temps pour sa date de début sur la machine 1. La fenêtre de temps de chaque tâche  $J_j = \pi_j \in J_z^2$  est déterminée par sa date de début au plus tôt ( $EST_{1j}$ ) sur la machine 1 :  $EST_{1j} = C_{1\pi_{j-1}}$  et par sa date de début au plus tard ( $LST_{1j}$ ) sur la machine 1 :  $LST_{1j} = C_{1\pi_n} - \sum_{i=j+1}^n p_{1i}$ . Les fenêtres de temps des tâches d’un même ensemble peuvent se chevaucher. Par récurrence, en déterminant la date de début de la tâche  $\pi_j$ , la fenêtre de temps de tâche  $\pi_{j+1}$  peut être modifiée pour garantir la faisabilité du planning.

Le choix de date de début des tâches d’un ensemble n’affecte pas la fenêtre de temps des tâches de l’autre ensemble. Par conséquent, nous étudions deux problèmes d’ordonnement sur une seule machine avec une séquence fixe, une fenêtre de temps ( $TW$ ) pour la minimisation totale des coûts d’électricité ( $TEC$ ) sous le tarif ToU.

Nous prenons l’exemple d’une séquence de 6 tâches  $J_1 \prec J_2 \prec J_3 \prec J_6 \prec J_5 \prec J_4$  dans dans la Fig.(13a). La tâche  $J_6$  est le “crossover job”. Les dates de début possibles pour chaque tâche sont représentées par la fenêtre de temps  $TW$  ( $TW$  en rouge représente les dates de début possibles sur la machine 2,  $TW$  en bleu représente les dates de début possibles sur la machine 1). Alors, nous étudions le problème  $(1, ToU | sequence, TW, Pu_{ij} | TEC)$  sur la machine 2 pour déterminer la date de début optimale des tâches  $J_z^1 = \{J_1, J_2, J_3\}$  (Fig.(13b). Sur la machine 1, nous étudions le problème  $1, ToU | sequence, TW, Pu_{ij} | TEC$  pour déterminer la date de début optimale des tâches  $J_z^2 = \{J_5, J_4\}$  (Fig.(13c).

Le problème similaire a été étudié par [Aghelinejad et al. 2018b]. Dans leur étude, ils ont étudié les problèmes d’ordonnement sur une seule machine, où la machine a plusieurs états de fonctionnement : en marche, en idle ou en arrêt. Chaque état de fonctionnement consomme une quantité d’énergie différente. En outre, la puissance nécessaire pour le traitement d’une tâche est différente de celle d’une autre. La séquence des tâches est fixée, et la tarification d’électricité est ToU. Ils ont considéré des cas où la vitesse de la machine est uniforme et des cas où la vitesse de la machine varie. Ces problèmes sont résolus par une approche de programmation dynamique. Ils ont montré que le problème de vitesse uniforme est

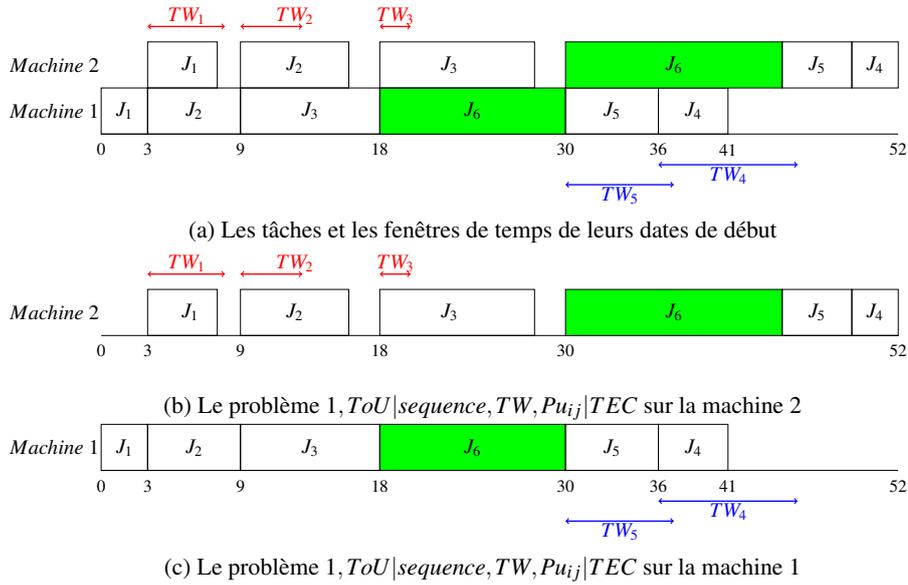


Figure 13 – La décomposition de  $(F2, ToU|sequence, TW, Pu_{ij}|TEC)$  en deux problèmes  $(1, ToU|sequence, TW, Pu_{ij}|TEC)$

polynomial de degré 3 ( $O(T^3)$ ) et que le problème de vitesse variable est pseudo-polynomial.

Sur la base de leur étude sur le cas de vitesse uniforme, notre problème est similaire à leur problème où la machine n'a que deux états de fonctionnement : marche ou arrêt. Une approche de programmation dynamique qui est prouvée polynomiale, est développée pour résoudre notre problème

Alors, le problème  $F2, ToU|sequence, Pu_{ij}|TEC$  peut être décomposé en deux problèmes d'ordonnancement sur une seule machine avec une séquence fixe, une fenêtre de temps donné ( $TW$ ) pour la minimisation du coût total de l'électricité ( $TEC$ ) sous la tarification d'électricité ToU, noté  $1, ToU|sequence, TW, Pu_{ij}|TEC$ , un sur la machine 1 et un sur la machine 2, et peut être résolu de manière optimale en temps polynomial ( $O(T^3)$ ).  $\square$

Selon le théorème (3), le problème  $F2, ToU|sequence, Pu_{ij}|TEC$  peut être décomposé en deux problèmes d'ordonnancement,  $1, ToU|sequence, TW, Pu_{ij}|TEC$ , sur la machine 1 et sur la machine 2, respectivement. Le problème est réversible, alors nous présentons l'approche de résolution de  $(1, ToU|sequence, TW, Pu_{ij}|TEC)$  sur la machine 2. La même approche peut être appliquée sur la machine 1 en inversant la configuration.

Dans notre approche, le problème avec  $T$  périodes et  $n$  tâches est modélisé par un graphe de  $V$  nœuds,  $E$  arcs et  $T + 1$  niveaux de décisions. Chaque niveau de décision  $t$  correspond à une période de temps (de période 1 à période  $T$ ). Chaque niveau  $t$  est associé à un ensemble de nœuds ( $N_t$ ) de date de début possible de la dernière tâche qui a été traitée jusqu'à la période  $t$ . Chaque nœud  $v \in V$  est identifié

par la tâche  $J_j$  et sa date de début  $t : (J_j, t)$ . Par exemple, le nœud  $v_1 = \{J_1, 8\}$  représente le traitement de la tâche  $J_1$  qui commence à la période  $t = 8$ .

Avec une séquence donnée, nous avons donc un nombre fini de tâches et leurs fenêtres de temps associées sont donc connues. Alors le nombre total des nœuds pour  $n$  tâches est prédéterminé. Nous avons besoins de deux nœuds supplémentaires, un nœud  $I$  qui représente l'état initial où aucune tâche n'a été traitée, un nœud  $F$  pour représenter l'état final où toutes les tâches sont traitées. Au final, le nombre total de nœuds est :  $|V| = \sum_{j=1}^n |TW_j| + 2$  dont  $|TW_j|$  : est le nombre de période  $t$  dans la fenêtre de temps  $TW_j$ .

L'arc  $e \in E : e_{(J_j, t) - (J_k, t')}$  avec  $J_j, J_k \in J$  et  $t, t' \in T$  représente la transition entre la tâche  $J_j$  commençant à période  $t$  et la tâche  $J_k$  commençant à période  $t'$ . La valeur de chaque arc  $e_{(J_j, t) - (J_k, t')}$  représente le coût d'électricité de traitement de tâche  $J_j$  si elle commence à période  $t$ .

$$TEC_{(J_j, t) - (J_k, t')} = \sum_t^{t+p_{2j}} Pu_{2j} \times c_t$$

Où :

$p_{2j}$  : le temps de traitement de tâche  $j$  sur la machine 2.

$Pu_{2j}$  : la puissance requise pour le traitement de tâche  $j$  sur la machine 2.

$c_t$  : le prix d'électricité à la période  $t$ .

Tout d'abord, nous avons  $|TW_{\pi_1}|$  arcs qui relie l'état initial avec le traitement de la première tâche ( $\pi_1$ ) de la séquence. Ensuite, nous avons  $|TW_{\pi_n}|$  qui relie le traitement de la dernière tâche ( $\pi_n$ ) de la séquence avec l'état final. Nous avons également la relation de précédence entre des nœuds qui est définie par la séquence. Pour une tâche  $J_k$  qui succède la tâche  $J_j$  dans la séquence  $\pi$ ,  $(J_k, t')$  est précédé par tous les nœuds  $(J_j, t)$  dont  $(t \leq t' - p_{2j})$ . Alors, entre  $(J_j, t)$  et  $(J_k, t')$ , le nombre d'arcs est :  $\min((t' - p_{2j} - t + 1), |TW_j|)$ . Le nombre d'arcs entre des nœuds dans le graphe est :  $\sum_{j=2}^n \sum_{t=EST_j}^{LST_j} \sum_{t'=EST_{j-1}}^{t'} \min((t - p_{2j} - t'), |TW_j|)$ . En comptant l'état initial et l'état final, nous avons le nombre total des arcs dans le graphe :

$$|E| = \sum_{j=2}^n \sum_{t=EST_j}^{LST_j} \sum_{t'=EST_{j-1}}^{t'} \min((t - p_{2j} - t'), |TW_j|) + |TW_{\pi_1}| + |TW_{\pi_n}|$$

Dans notre graphe, une solution réalisable du problème est représentée par un chemin qui part du nœud  $I$  au niveau 1 jusqu'au nœud  $F$  au niveau  $T$ . Le chemin le plus court qui commence au nœud  $I$  et se termine au nœud  $F$  représente la solution optimale du problème qui vise à minimiser le coût total d'électricité. Il est bien connu que lorsque toutes les valeurs des arcs du graphe sont positives, l'algorithme de Dijkstra est l'un des algorithmes les plus efficaces pour trouver le chemin le plus court d'un graphe comme la solution optimale. Nous considérons que le coût  $C(J_j, t)$  associé au nœud  $(J_j, t)$  indique le coût minimum pour la date de début de la tâche  $J_j$  à la période  $t$ . La relation de récurrence utilisée pour évaluer le coût de chaque nœud est la suivante :

$$C(I, 0) = 0$$

$$C(J_j, t) = \min_{(J', t') \in A_{(J_j, t)}} \left[ C(J', t') + TEC_{(J_j, t) - (J_k, t')} \right]$$

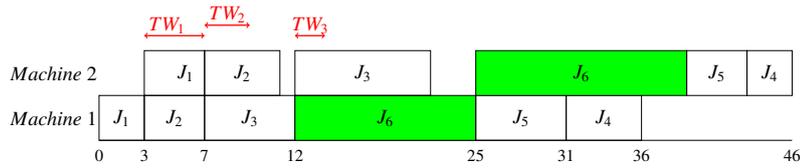
Dont :

$A_{J_j, t}$  : l'ensemble des noeuds reliés à  $(J_j, t)$ .

Nous prenons un exemple pour illustrer notre modélisation, Fig.(14). Nous avons une séquence de 6 tâches  $\pi = J_1 \prec J_2 \prec J_3 \prec J_6 \prec J_5 \prec J_4$ . Dans cet exemple, nous déterminons la date de début optimale sur la machine 2 pour les tâches  $J_1, J_2, J_3$ .

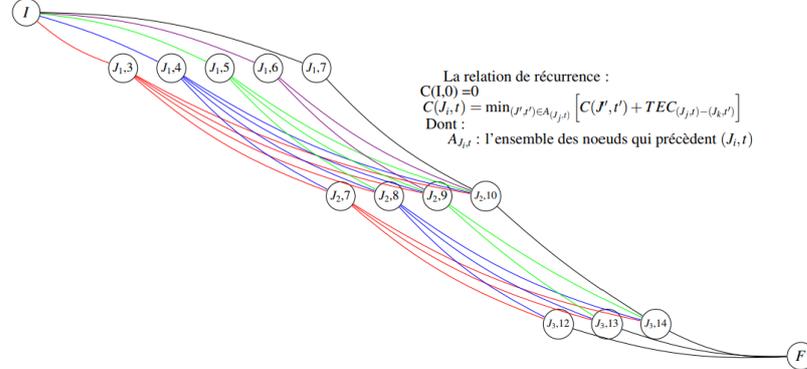
- $J_1$  a  $EST_1 = 3$  et  $LST_1 = 7$  donc  $TW_1 = [3; 7]$ .
- $J_2$  a  $EST_2 = 7$  et  $LST_2 = 10$  donc  $TW_2 = [7; 10]$ .
- $J_3$  a  $EST_3 = 12$  et  $LST_3 = 14$  donc  $TW_3 = [12; 14]$ .

Les fenêtres de temps pour les tâches  $J_1, J_2, J_3$  sont représentées dans la Fig.(14a). La modélisation en graphe est représentée dans la Fig.(14b).



(a) Les fenêtres de temps pour les tâches  $J_1, J_2$  et  $J_3$

Periodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	25
Prix d'électricité	0.04	0.04	0.04	0.04	0.04	0.13	0.13	0.13	0.13	0.04	0.04	0.04	0.08	0.08	...	0.13



(b) La modélisation par graphe pour la détermination de la date de début optimale de  $J_1, J_2, J_3$

Figure 14 – L'exemple de détermination de date de début optimale des tâches  $J_1, J_2$  et  $J_3$

### 3.3.3 Le problème $F2, ToU | perm, C_{max}^* | TEC$ .

Nous avons présenté dans les sections précédentes la recherche des solutions optimales équivalentes de problème  $F2 | perm | C_{max}$  et l'approche de résolution pour

le problème  $F2, ToU|sequence, Pu_{ij}|TEC$ . Nous présentons dans cette section l'approche de résolution pour le problème  $F2, ToU|perm, C_{max}^*|TEC$  qui reprend l'ensemble des éléments présentés auparavant.

L'approche de résolution se compose de deux phases. La première phase est la recherche des solutions optimales équivalentes. Dans cette phase, nous appliquons les méthodes présentées dans la partie (3.3.1) pour construire un ensemble des solutions équivalentes. Ensuite, la deuxième phase consiste à déterminer la date de début optimale des tâches dans chaque solution trouvée lors de la première phase. Dans cette phase, nous appliquons l'approche présentée dans la partie (3.3.2) pour construire le l'ordonnancement. A la fin, nous prenons l'ordonnancement dont le coût d'électricité est minimum pour notre solution. Nous présentons notre heuristique *HFF* dans l'algorithme (6).

**Algorithme 6 :** Heuristique *HFF*

**Data :**  $J, p_{ij}, Pu_{ij}, c_t$

**Result :**  $\pi^*, TEC_{\pi^*}$

1. Classer les tâches dans les deux sous-ensembles  $\bar{J}$  et  $\underline{J}$  ;
2. Appliquer l'algorithme "OG" de [Hadda et al. (2018)] pour obtenir l'ordre partiel total *SPO*;
3. Déterminer le "crossover"  $J_z$  et  $J_z^1 \prec J_z \prec J_z^2$  pour *SPO*;
4. Générer l'ensemble *E* des solutions optimales équivalentes à partir de *SPO*;
5. Appliquer l'algorithme (4) sur  $J_z^1$  et  $J_z^1$  pour le changement des tâches entre  $\bar{J}$  et  $\underline{J}$  et obtenir nouveau *SPO\**;
6. Ajouter des solutions générées par *SPO\** à l'ensemble *E*;
7. Appliquer l'algorithme (3) sur *SPO* pour obtenir les nouveaux *SPO'* en changement le premier et le dernier "ground job".
8. Ajouter des solutions générées par *SPO'* à l'ensemble *E*;
9. Pour chaque *SPO'* obtenu à l'étape 7, reprendre les étape 3,4,5 et 6;
10. Pour chaque solution de *E*, appliquer la recherche locale (5) qui étend l'ensemble de solution *E*;
11. Pour chaque solution optimale équivalente de *E*, appliquer la programmation dynamique présentée dans la partie (3.3.2) pour construire le plan de production.
12. Prendre la solution  $\pi^* \in E$  dont  $TEC_{\pi^*}$  est minimum pour la solution finale.

A part de l'heuristique *HFF*, nous proposons 6 autres heuristiques qui sont fondées sur les résultats de Johnson [Johnson (1954)] et de Hadda [Hadda et al. (2018)] :

- Heuristique *Johnson<sup>LS</sup>* : l'application de la recherche locale (5) sur la solution de Johnson.
- Heuristique *Johnson<sup>ST</sup>* : l'application de la programmation dynamique de la partie (3.3.2) sur la solution de Johnson pour déterminer la date de début optimale des tâches.
- Heuristique *Johnson<sup>LS+ST</sup>* : l'application de la recherche locale (5) sur la solution de Johnson et ensuite la détermination de la date de début optimale des tâches par la programmation dynamique.
- Heuristique *Hadda<sup>LS</sup>* : l'application de la recherche local (5) sur l'ensemble des solutions générées par l'algorithme "OG" de [Hadda et al. (2018)].
- Heuristique *Hadda<sup>ST</sup>* : l'application de la programmation dynamique de la partie (3.3.2) sur les solutions de l'algorithme "OG" de [Hadda et al. (2018)] pour déterminer la date de début optimale des tâches..
- Heuristique *Hadda<sup>LS+ST</sup>* : l'application de la recherche locale (5) sur les solutions de l'algorithme "OG" de [Hadda et al. (2018)] et ensuite la détermination de la date de début optimale des tâches par la programmation dynamique.

### 3.3.4 La borne inférieure

Dans cette section, nous proposons une borne inférieure pour estimer la qualité des solutions fournies par nos heuristiques. La borne inférieure est obtenue en relaxant certaines contraintes du problème.

Premièrement, nous prenons la machine 1 et la machine 2 pour deux machines parallèles non reliées. La charge de travail de chaque machine ne change pas. Elle est définie par le temps de traitement des tâches sur cette machine ( $WL_i = \sum_{j \in J} p_{ij}, \forall i \in M$ ).

Deuxièmement, la préemption est autorisée. Le traitement d'une tâche peut être interrompu à tout moment et peut se poursuivre ultérieurement sans frais supplémentaires.

Le problème relaxé est similaire au problème  $(R|pmtn|C_{max} + TEC)$ , qui peut être résolu en temps polynomial selon l'étude de [Chen et al. (2018)].

Soit  $Z = \max(WL_1, WL_2)$  dont  $WL_i$  est la charge de travail de la machine  $i$ . Si  $C_{max}^*$  est connu, alors la solution optimale de  $(R|pmtn|C_{max}^* + TEC)$  utilise  $Z$  périodes qui se trouvent avant  $C_{max}^*$  dont les prix d'électricité sont les moins chers. Cette proposition est démontrée dans l'étude de [Chen et al. (2018)]. Alors la borne inférieure de notre problème  $(F2, ToU|perm, C_{max}^*|TEC)$  est calculée de la façon suivante :

1. De période 1 à  $C_{max}^*$ , trier les périodes de temps dans l'ordre non-décroissante de leurs prix d'électricité. Nous obtenons un horizon de travail  $T'$  dont les prix d'électricité sont non-décroissants pendant une fenêtre de temps allant de la première période à  $C_{max}^*$ .
2. Pour chaque machine  $i$ , trier les tâches dans l'ordre non-croissant de la puissance requise. En traitant les tâches selon l'ordre trié dans l'horizon de travail  $T'$ , nous obtenons le coût d'électricité total  $TEC'_i$  pour chaque machine  $i$ .
3. La borne inférieure est alors :  $LB = \sum_{i \in M} TEC'_i$ .

### 3.4 Les tests numériques

Dans cette section, nous menons des tests numériques. Nous comparons nos heuristiques avec une approche de résolution dans la littérature : “Iterated local search dynamic programming” (ILSDP) de [Wang et al. (2018)]. ILSDP emploie une recherche locale itérative pour résoudre un problème d'ordonnement de flow shop à deux machines de permutation avec la minimisation de coût d'électricité pendant un horizon de travail  $T$ . Ensuite, ils appliquent la programmation dynamique à la solution finale de la recherche locale itérative. En assurant que ILSDP donne une solution  $\pi$  réalisable pour  $(F2, ToU | perm, C_{max}^* | TEC)$  ( $C_{max}(\pi) = C_{max}^*$ ), l'horizon de travail est alors limité  $T = C_{max}^*$ .

Les heuristiques et ILSDP sont implémentées sur “Microsoft Visual Studio Community 2017-Visual C++ 2017 version 15.5.2”. Le modèle mathématique est implémenté sur “Microsoft Visual Studio Community 2017-Visual C++ 2017 version 15.5.2” couplé avec “Cplex Studio IDE 12.7.1”. La solution de Johnson étant évidemment une solution réalisable de notre problème, alors elle est prise comme solution initiale pour le solveur Cplex. Les tests numériques sont menés sur un ordinateur “Intel Core i5-7200U CPU @2.5 GHz 2.5 GHz” de 6 GB de mémoire et un système d'exploitation de *Windows 7*.

#### 3.4.1 La génération des données

Une instance de  $(F2, ToU | perm, C_{max}^* | TEC)$  se compose de deux types de données. Le premier type contient des données de tâches : le temps de traitement et la puissance requise pour chaque tâche sur chaque machine. La configuration de chaque instance est déterminée par le nombre de tâche à traiter  $n$ . Nous générons des instances pour  $n = 5, 10, 15, 20$  tâches. Les temps de traitement sont générés en se fondant sur des instances de Taillard pour l'ordonnement de flow shop [Taillard 1993]. Si  $n \leq 20$ , les temps de traitement sont pris aléatoirement à partir de l'ensemble d'une instance de 20 tâches et 5 machines de [Taillard 1993]. Si  $n = 20$ , les temps de traitement sont pris de deux premières machines dans chaque instance de 20 tâches de [Taillard 1993]. La puissance requise pour le traitement

de chaque tâche est générée aléatoirement entre [1;3] à la manière de l'étude de [Pilerood et al. 2018].

Le deuxième type de données est la tarification d'électricité. Nous étudions le tarif de type "Time of Use" (*ToU*). Tout d'abord, nous définissons l'horizon de travail :  $T = 0.8 \times \sum_{i \in M} \sum_{j \in J} p_{ij}$ . Ensuite, l'horizon  $T$  est divisé en trois types de période de prix : "off-peak", "peak" et "semi-peak". Le prix d'électricité de chaque type de période est de 0.04 euro/kWh, 0.13 euro/kWh et 0.08 euro/kWh respectivement, basé sur l'étude de [Sharma et al. 2015]. La durée pour chaque type de périodes de prix est donnée comme ci-dessous :

$$c_t = \begin{cases} 0.04, & 1 \leq t \leq \lceil T/4 \rceil \\ 0.13, & \lceil T/4 \rceil + 1 \leq t \leq \lceil T/2 \rceil \\ 0.04, & \lceil T/2 \rceil + 1 \leq t \leq \lceil 3T/4 \rceil \\ 0.08, & \lceil 3T/4 \rceil + 1 \leq t \leq T \end{cases}$$

### 3.4.2 Les résultats numériques

Nous comparons nos heuristiques proposées avec ILSDP en termes d'écart entre leur *TEC* avec la "meilleure solution trouvée" et également avec la borne inférieure. Nous avons deux types d'écart. Le premier type d'écart est entre le *TEC* de chaque approche de solution avec la "meilleure solution trouvée" (BKS).  $TEC^{BKS}$  représente la meilleure solution trouvée parmi toutes les approches proposées. Le  $Gap_i$  représente l'écart entre la solution de l'approche  $i$  avec la meilleure solution :  $Gap_i = \frac{TEC_i - TEC^{BKS}}{TEC^{BKS}}$ . Le deuxième type d'écart est la comparaison avec la borne inférieure. Le " $Gap_i^{LB}$ " représente l'écart entre la solution de l'approche  $i$  avec la borne inférieure  $LB$  :  $Gap_i^{LB} = \frac{TEC_i - LB}{LB}$ . Le tableau (12) donne: la moyenne, la valeur maximale, la valeur minimale et l'écart-type ( $\sigma$ ) des deux types d'écart pour toutes les instances, le tableau (12a) pour la comparaison avec la "meilleure solution trouvée" et le tableau (12b) pour la comparaison avec la borne inférieure.

Les résultats et les comparaisons entre ces heuristiques avec la "meilleure solution trouvée" sont présentés dans les tableaux (13), (14) et (15) pour les instances de 5, 10 et 15 tâches. Les tableaux (16), (17) et (18) représentent les résultats pour les instances de 20 tâches. Dans chaque tableau, nous retrouvons le coût d'électricité total de la solution trouvée, le temps d'exécution en secondes de chaque approche. La résolution de *Cplex* est limitée par un temps d'exécution maximum de 3600 secondes.

Les résultats et les comparaisons entre ces heuristiques avec la borne inférieure sont présentés dans le tableau (19) pour les instances de 10 et 15 tâches. Le tableau (20) représente les résultats pour les instances de 20 tâches. Pour les instances de 5 tâches, *Cplex* donne la solution optimale, alors nous ne comparons pas ces instances avec la borne inférieure.

D'après les tableaux (12a), (13), (14), (15), (16), (17) et (18), nous avons des observations ci-dessous :

- Partant d'une solution de Johnson, nous avons réduit considérablement son coût total d'électricité en mettant en œuvre la recherche locale et la détermination de la date de début optimale. L'écart moyen par rapport à la "meilleure solution trouvée" est diminué de 13.5% (la solution de Johnson) à 3.8% ( $Johnson^{LS+ST}$ ) (tableau (12a)). Bien qu'une amélioration ait été observée, nous trouvons toujours un écart maximal de 12.3% de ( $Johnson^{LS+ST}$ ) par rapport à la "meilleure solution trouvée" pour certaines instances. Par conséquent, pour les tarifs  $ToU$ , la solution de Johnson n'est pas prometteuse pour un gain financier.
- L'heuristique  $HFF$  donne la meilleure solution parmi toutes les approches proposées et dans toutes les instances. Les solutions de  $HFF$  sont donc prises pour les "meilleures solutions trouvées" (tableau (15) et tableau (18)).
- $ILSDP$  donne les solutions de bonne qualité pour les instances de petite taille. En ce qui concerne les instances de 5 tâches, l'écart entre  $HFF$  et  $ILSDP$  n'est pas significatif (0.01%). Cependant, les performances de  $ILSDP$  se détériorent au fur et à mesure que le nombre de tâche augmente (tableau (15) et tableau (18)). Pour les instances de 10, 15 et 20 tâches, les solutions de  $ILSDP$  ont un écart respectivement de 5.59%, 7.64% et 8.84% par rapport à celle de  $HFF$ .
- L'heuristique  $Hadda^{LS+ST}$  donne également des solutions de bonne qualité qui ont en moyenne un écart de 0.6% par rapport à la "meilleure solution trouvée" pour toutes les instances (tableau (12a)). Bien que pour certaines instances, l'écart maximal entre  $Hadda^{LS+ST}$  et  $HFF$  soit à 3.8%,  $Hadda^{LS+ST}$  est aussi très recommandé grâce à son temps d'exécution : 145.13 secondes en moyenne pour les instances de 20 tâches (tableau (17)).

D'après les tableaux (12b), (19) et (20), nous observons :

- $HFF$  donne de bonnes performances dans toutes les instances. L'écart entre ses solutions et la borne inférieure est en moyenne de 5.2%. D'autre part, plus la taille des instances augmente, plus l'écart entre la solution de  $HFF$  avec la borne inférieure diminue (6.7% pour 10 tâches, 5.5% pour 15 tâches et 4.7% pour 20 tâches). Ainsi, lorsque l'exigence de la qualité de solution est élevée,  $HFF$  est une approche très recommandée.
- En plus de  $HFF$ , nous avons également l'heuristique  $Hadda^{LS+ST}$  dont l'écart avec la borne inférieure est en moyenne de 6%. (7.6% pour 10 tâches, 6.0% pour 15 tâches et 5.5% pour 20 tâches).  $Hadda^{LS+ST}$  est aussi une très bonne approche en terme de qualité de solution.
- Nous avons dans le tableau (12b) trois autres heuristiques  $Johnson^{LS+ST}$ ,  $Hadda^{LS}$  et  $Hadda^{ST}$  dont la performance est meilleure que celle de  $ILSDP$  pour le problème  $F2, ToU | perm, C_{max}^* | TEC$ . De plus, ces heuristiques ont

un grand avantage sur le temps d'exécution. Lorsque la limitation de temps d'exécution est importante, elles peuvent être utilisés pour trouver des solutions réalisables de bonne qualité pour les instances de grandes tailles.

Tableau 12 – La moyenne, max, min et l'écart-type des résultats des heuristiques pour toutes les instances

(a) Ecart avec la meilleure solution trouvée "BKS"

Gap (%)	Johnson	Johnson <sup>LS</sup>	Johnson <sup>ST</sup>	Johnson <sup>LS+ST</sup>	Hadda	Hadda <sup>LS</sup>	Hadda <sup>ST</sup>	Hadda <sup>LS+ST</sup>	HFF	Cplex	ILSDP
<b>Moyenne</b>	13.5%	8.4%	7.9%	3.8%	8.6%	4.7%	3.0%	0.6%	<b>0.0%</b>	7.6%	6.6%
<b>Max</b>	35.5%	25.5%	20.3%	12.3%	27.8%	16.1%	9.0%	3.8%	<b>0.0%</b>	20.3%	21.0%
<b>Min</b>	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	<b>0.0%</b>	0.0%	0.0%
$\sigma$	7.9%	5.8%	4.5%	2.8%	6.7%	4.1%	2.4%	0.9%	<b>0.0%</b>	5.0%	5.3%

(b) Ecart avec la borne inférieure

Gap (%)	Johnson	Johnson <sup>LS</sup>	Johnson <sup>ST</sup>	Johnson <sup>LS+ST</sup>	Hadda	Hadda <sup>LS</sup>	Hadda <sup>ST</sup>	Hadda <sup>LS+ST</sup>	HFF	Cplex	ILSDP
<b>Moyenne</b>	21.8%	15.7%	14.8%	9.9%	15.8%	11.0%	8.9%	6.0%	<b>5.2%</b>	14.9%	13.6%
<b>Max</b>	44.0%	29.8%	29.4%	20.8%	35.8%	21.9%	20.4%	13.1%	<b>10.4%</b>	29.4%	23.2%
<b>Min</b>	9.9%	7.4%	2.4%	2.4%	6.3%	5.1%	2.3%	1.8%	<b>1.8%</b>	2.4%	4.7%
$\sigma$	6.8%	5.1%	4.6%	3.1%	6.6%	4.2%	3.2%	2.1%	<b>1.8%</b>	4.8%	4.7%

### 3.5 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur le problème d'ordonnancement d'un atelier de flow shop à deux machines sous ToU avec une optimisation conjointe du makespan et du coût de l'électricité, notée  $(F2, ToU | perm, C_{max}^* | TEC)$ . Nous avons exploité et étendu l'ensemble des solutions optimales équivalentes de [Hadda et al. 2018] avec des nouvelles propositions et la recherche locale. Nous avons démontré que sous le makespan optimal et la séquence prédéterminée, le problème d'ordonnancement écoénergétique de flow shop à deux machines peut être résolu optimalement dans  $O(T^3)$  étapes qui est meilleure que  $O(T^4)$  étapes de la programmation dynamique de [Wang et al. (2018)]. Nous avons proposé trois heuristiques reprenant la règle de Johnson, trois heuristiques reposant sur l'algorithme *OG* de [Hadda et al. 2018] et une heuristique *HFF*. Nous avons mené des tests numériques pour évaluer la performance de nos heuristiques. Nous avons comparé nos heuristiques avec l'approche *ILSDP* de [Wang et al. (2018)] et avec une borne inférieure.

Notre étude sur l'extension de l'ensemble des solutions équivalentes offre plus de solutions au manager non seulement dans la minimisation de coût d'électricité mais aussi dans l'approche d'ordonnancement robuste. En cas de perturbation, une solution peut être remplacée par une autre sans impact significatif car elles ont toutes la même valeur de makespan. Cinq parmi les sept heuristiques proposées ont amélioré la solution de Johnson et *ILSDP* en termes de coût d'électricité total. Le manager peut utiliser n'importe lequel de ces heuristiques *Johnson<sup>LS+ST</sup>*, *Hadda<sup>LS</sup>*, *Hadda<sup>ST</sup>*, *Hadda<sup>LS+ST</sup>* et *HFF* pour diminuer le coût total d'électricité

Tableau 13 – La comparaison entre les heuristiques basées sur Johnson avec la meilleure solution trouvée “BKS” pour les instances de 5, 10 et 15 tâches

Number of job	Instance	Johnson			Johnson <sup>S</sup>			Johnson <sup>S1</sup>			Johnson <sup>S+S1</sup>			Cmax	TEC <sup>BKS</sup>
		TEC <sub>1</sub>	Gap <sub>1</sub>	Time	TEC <sub>2</sub>	Gap <sub>2</sub>	Time	TEC <sub>3</sub>	Gap <sub>3</sub>	Time	TEC <sub>4</sub>	Gap <sub>4</sub>	Time		
5	1	76.6799	0.00%	1	76.6799	0.00%	1	76.6799	0.00%	1	76.6799	0.00%	1	284	76.6799
	2	127.646	1.29%	1	126.694	0.54%	1	126.996	0.78%	1	126.996	0.78%	1	368	126.019
	3	102.282	6.12%	1	100.527	4.30%	1	100.982	4.77%	1	98.8378	2.54%	1	362	96.387
	4	44.14	0.00%	1	44.14	0.00%	1	44.14	0.00%	1	44.14	0.00%	1	236	44.14
	5	25.866	0.00%	1	25.866	0.00%	1	25.866	0.00%	1	25.866	0.00%	1	182	25.866
	6	48.266	0.00%	1	48.266	0.00%	1	48.266	0.00%	1	48.266	0.00%	1	238	48.266
	7	84.023	8.46%	1	77.4708	0.00%	1	84.023	8.46%	1	77.4708	0.00%	1	318	77.4708
	8	43.089	5.02%	1	41.3249	0.72%	1	43.089	5.02%	1	41.3249	0.72%	1	262	41.0279
	9	123.285	3.32%	1	122.295	2.49%	1	119.485	0.13%	1	119.485	0.13%	1	442	119.325
	10	51.361	0.00%	1	51.361	0.00%	1	51.361	0.00%	1	51.361	0.00%	1	263	51.361
	Max	-	8.46%	1	-	4.30%	1	-	8.46%	1	-	2.54%	1	-	-
	Moyenne	-	2.42%	1	-	0.80%	1	-	1.92%	1	-	0.42%	1	-	-
	Min	-	0.00%	1	-	0.00%	1	-	0.00%	1	-	0.00%	1	-	-
10	1	128.382	8.73%	1	121.416	2.83%	1	127.392	7.89%	1	120.426	1.99%	1	574	118.074
	2	122.459	13.02%	1	120.173	10.91%	1	112.848	4.15%	1	110.373	1.86%	1	523	108.356
	3	168.745	18.13%	1	156.001	9.21%	1	157.15	10.01%	1	146.721	2.71%	1	634	142.846
	4	198.591	7.49%	1	189.429	2.53%	1	197.523	6.91%	1	188.505	2.03%	1	656	184.761
	5	107.617	19.83%	1	101.119	12.59%	2	89.932	0.14%	1	89.932	0.14%	2	544	89.81
	6	110.356	18.30%	2	104.893	12.44%	3	99.355	6.51%	2	94.738	1.56%	3	549	93.2839
	7	119.554	20.05%	3	110.176	10.64%	3	114.352	14.83%	3	104.974	5.41%	3	408	99.5827
	8	120.623	9.07%	3	117.032	5.83%	3	117.707	6.44%	3	115.331	4.29%	3	501	110.588
	9	123.176	23.29%	3	118.478	18.59%	3	106.597	6.70%	3	101.899	1.99%	4	521	99.9071
	10	129.803	35.53%	4	110.864	15.76%	4	106.97	11.69%	4	100.895	5.35%	4	520	95.7742
	Max	-	35.53%	4	-	18.59%	4	-	14.83%	4	-	5.41%	4	-	-
	Moyenne	-	17.34%	2	-	10.13%	2.2	-	7.53%	2	-	2.73%	2.3	-	-
	Min	-	7.49%	1	-	2.53%	1	-	0.14%	1	-	0.14%	1	-	-
15	1	225.372	12.15%	1	212.493	5.74%	1	221.638	10.29%	1	208.759	3.88%	1	809	200.957
	2	173.076	16.40%	1	163.678	10.08%	1	159.764	7.45%	1	152.881	2.82%	3	770	148.692
	3	224.162	11.62%	1	209.033	4.09%	1	223.699	11.39%	1	208.571	3.86%	5	828	200.823
	4	258.845	14.21%	1	244.075	7.69%	1	253.295	11.76%	1	238.524	5.24%	7	974	226.64
	5	227.819	22.61%	1	212.968	14.62%	1	195.77	5.36%	1	191.816	3.24%	10	845	185.802
	6	162.234	17.91%	1	151.852	10.37%	1	145.476	5.73%	1	141.848	3.09%	6	755	137.59
	7	156.987	22.65%	1	147.304	15.09%	1	136.14	6.36%	1	130.995	2.84%	9	725	127.995
	8	221.353	10.21%	1	214.432	6.76%	1	221.057	10.06%	1	208.599	3.86%	4	807	200.848
	9	208.269	15.15%	1	197.928	9.43%	1	198.465	9.73%	1	188.124	4.01%	6	792	180.87
	10	183.56	12.23%	1	177.612	8.60%	1	173.164	5.88%	1	167.739	2.56%	1	766	163.55
	Max	-	22.65%	1	-	15.09%	1	-	11.76%	1	-	5.24%	10	-	-
	Moyenne	-	15.51%	1	-	9.25%	1	-	8.40%	1	-	3.49%	5.2	-	-
	Min	-	10.21%	1	-	4.09%	1	-	5.36%	1	-	2.34%	1	-	-

Tableau 14 – La comparaison entre les heuristiques basées sur Hadda avec la meilleure solution trouvée “BKS” pour les instances de 5, 10 et 15 tâches

Number of job	Instance	Hadda		Hadda <sup>S</sup>		Hadda <sup>S†</sup>		Hadda <sup>S+ST</sup>		Cmax	TEC <sup>BKS</sup>				
		TEC <sub>5</sub>	Gaps	Time	TEC <sub>6</sub>	Gaps	Time	TEC <sub>7</sub>	Gaps			Time	TEC <sub>8</sub>	Gaps	Time
5	1	76.6799	0.00%	1	76.6799	0.00%	1	76.6799	0.00%	1	76.6799	0.00%	1	284	76.6799
	2	126.694	0.54%	1	126.694	0.54%	1	126.019	0.00%	1	126.019	0.00%	1	368	126.019
	3	100.527	4.30%	1	97.6871	1.35%	1	99.227	2.95%	1	96.3871	0.00%	1	362	96.387
	4	44.14	0.00%	1	44.14	0.00%	1	44.14	0.00%	1	44.14	0.00%	1	236	44.14
	5	25.866	0.00%	1	25.866	0.00%	1	25.866	0.00%	1	25.866	0.00%	1	182	25.866
	6	48.266	0.00%	1	48.266	0.00%	1	48.266	0.00%	1	48.266	0.00%	1	238	48.266
	7	77.4708	0.00%	1	77.4708	0.00%	1	77.4708	0.00%	1	77.4708	0.00%	1	318	77.4708
	8	42.063	2.52%	1	41.3249	0.72%	1	42.063	2.52%	1	41.3249	0.72%	1	262	41.0279
	9	123.285	3.32%	1	122.295	2.49%	1	119.485	0.13%	1	119.485	0.13%	1	442	119.325
	10	51.361	0.00%	1	51.361	0.00%	1	51.361	0.00%	1	51.361	0.00%	1	263	51.361
	Max	-	4.30%	1	-	2.49%	1	-	2.95%	1	-	0.72%	1	-	-
	Moyenne	-	1.07%	1	-	0.51%	1	-	0.56%	1	-	0.09%	1	-	-
	Min	-	0.00%	1	-	0.00%	1	-	0.00%	1	-	0.00%	1	-	-
10	1	120.7	2.22%	1	120.7	2.22%	1	118.774	0.59%	1	118.774	0.59%	7	574	118.074
	2	122.594	13.14%	1	119.471	10.26%	1	112.658	3.97%	1	110.093	1.60%	1	523	108.356
	3	151.958	6.38%	1	151.958	6.38%	1	145.696	2.00%	1	142.846	0.00%	1	634	142.846
	4	190.653	3.19%	1	188.998	2.29%	1	189.405	2.51%	2	187.893	1.70%	1	656	184.761
	5	96.7	7.67%	3	96.7	7.67%	1	89.81	0.00%	1	89.81	0.00%	1	544	89.81
	6	97.323	4.33%	4	97.323	4.33%	1	93.7161	0.46%	1	93.7161	0.46%	1	549	93.2839
	7	113.758	14.23%	4	107.53	7.98%	1	108.556	9.01%	1	101.995	2.42%	1	408	99.5827
	8	120.605	9.06%	4	112.352	1.60%	1	117.581	6.32%	1	110.588	0.00%	1	501	110.588
	9	116.972	17.08%	5	115.976	16.08%	1	99.9071	0.00%	1	99.9071	0.00%	2	521	99.9071
	10	122.36	27.76%	5	101.731	6.22%	1	103.946	8.53%	1	97.9953	2.32%	1	520	95.7742
	Max	-	27.76%	5	-	16.08%	1	-	9.01%	2	-	2.42%	7	-	-
	Moyenne	-	10.51%	2.9	-	6.50%	1	-	3.34%	1.1	-	0.91%	1.7	-	-
	Min	-	2.22%	1	-	1.60%	1	-	0.00%	1	-	0.00%	1	-	-
15	1	208.676	3.84%	2	205.601	2.31%	6	203.98	1.50%	1	201.577	0.31%	37	809	200.957
	2	165.277	11.15%	1	158.777	6.78%	2	149.96	0.85%	1	148.922	0.15%	20	770	148.692
	3	205.114	2.14%	1	202.851	1.01%	1	203.76	1.46%	1	201.993	0.58%	10	828	200.823
	4	234.757	3.58%	4	232.902	2.76%	13	228.96	1.02%	2	226.64	0.00%	120	974	226.64
	5	227.679	22.54%	4	208.708	12.33%	0	188.85	1.64%	1	188.803	1.62%	1	845	185.802
	6	164.397	19.48%	1	146.21	6.26%	0	137.59	0.00%	1	137.59	0.00%	1	755	137.59
	7	157.303	22.90%	1	145.522	13.69%	1	135.15	5.59%	1	130.63	2.06%	9	725	127.995
	8	216.425	7.76%	3	210.476	4.79%	5	214.61	6.85%	1	200.86	0.01%	37	807	200.848
	9	194.909	7.76%	3	187.878	3.87%	1	180.97	0.06%	1	180.87	0.00%	5	792	180.87
	10	179.737	9.90%	3	174.81	6.88%	1	164.95	0.86%	1	163.55	0.00%	10	766	163.55
	Max	-	22.90%	4	-	13.69%	13	-	6.85%	2	-	2.06%	120	-	-
	Moyenne	-	11.10%	2.3	-	6.07%	3	-	1.98%	1.1	-	0.47%	25	-	-
	Min	-	2.14%	1	-	1.01%	0	-	0.00%	1	-	0.00%	1	-	-

Tableau 15 – La comparaison entre *HFF*, *Cplex* et *ILSDP* avec la meilleure solution trouvée “BKS” pour les instances de 5, 10 et 15 tâches

Number of job	Instance	HFF			Cplex			ILSDP			Cmax	TEC <sup>BKS</sup>
		TEC <sub>9</sub>	Gap <sub>9</sub>	Time	TEC <sub>10</sub>	Gap <sub>10</sub>	Time	TEC <sub>11</sub>	Gap <sub>11</sub>	Time		
5	1	76.6799	0.00%	1	76.6799	0.00%	23	76.68	0.00%	3	284	76.6799
	2	126.019	0.00%	1	126.019	0.00%	47	126.019	0.00%	3	368	126.019
	3	96.387	0.00%	1	96.387	0.00%	97	96.3871	0.00%	0	362	96.387
	4	44.14	0.00%	1	44.14	0.00%	8	44.14	0.00%	1	236	44.14
	5	25.866	0.00%	1	25.866	0.00%	10	25.866	0.00%	1	182	25.866
	6	48.266	0.00%	1	48.266	0.00%	12	48.266	0.00%	0	238	48.266
	7	77.4708	0.00%	1	77.471	0.00%	65	77.471	0.00%	0	318	77.4708
	8	41.0279	0.00%	1	41.028	0.00%	123	41.0279	0.00%	0	262	41.0279
	9	119.325	0.00%	1	119.325	0.00%	268	119.485	0.13%	1	442	119.325
	10	51.361	0.00%	1	51.361	0.00%	1	51.361	0.00%	0	263	51.361
	Max	-	0.00%	1	-	0.00%	268	-	0.13%	3	-	-
	Moyenne	-	0.00%	1	-	0.00%	65.4	-	0.01%	0.9	-	-
	Min	-	0.00%	1	-	0.00%	1	-	0.00%	0	-	-
10	1	118.074	0.00%	8	127.392	7.89%	3600	119.291	1.03%	4	574	118.074
	2	108.356	0.00%	8	112.713	4.02%	3600	110.1	1.61%	2	523	108.356
	3	142.846	0.00%	7	157.15	10.01%	3600	159.587	11.72%	1	634	142.846
	4	184.761	0.00%	2	197.523	6.91%	3600	197.271	6.77%	1	656	184.761
	5	89.81	0.00%	2	89.932	0.14%	3600	93.325	3.91%	2	544	89.81
	6	93.2839	0.00%	9	99.299	6.45%	3600	102.358	9.73%	3	549	93.2839
	7	99.5827	0.00%	2	114.055	14.53%	3600	101.617	2.04%	2	408	99.5827
	8	110.588	0.00%	1	117.707	6.44%	3600	117.505	6.25%	3	501	110.588
	9	99.9071	0.00%	3	106.594	6.69%	3600	99.956	0.05%	2	521	99.9071
	10	95.7742	0.00%	8	100.895	5.35%	3600	108.056	12.82%	2	520	95.7742
	Max	-	0.00%	9	-	14.53%	3600	-	12.82%	4	-	-
	Moyenne	-	0.00%	5	-	6.84%	3600	-	5.59%	2.2	-	-
	Min	-	0.00%	1	-	0.14%	3600	-	0.05%	1	-	-
15	1	200.957	0.00%	178	221.638	10.29%	3600	202.026	0.53%	6	809	200.957
	2	148.692	0.00%	155	159.715	7.41%	3600	157.089	5.65%	2	770	148.692
	3	200.823	0.00%	10	223.7	11.39%	3600	223.466	11.28%	1	828	200.823
	4	226.64	0.00%	160	253.295	11.76%	3600	242.674	7.07%	1	974	226.64
	5	185.802	0.00%	120	195.67	5.31%	3600	204.783	10.22%	2	845	185.802
	6	137.59	0.00%	10	145.476	5.73%	3600	152.705	10.99%	2	755	137.59
	7	127.995	0.00%	102	135.92	6.19%	3600	134.31	4.93%	1	725	127.995
	8	200.848	0.00%	180	220.948	10.01%	3600	220.781	9.92%	1	807	200.848
	9	180.87	0.00%	10	198.465	9.73%	3600	198.269	9.62%	1	792	180.87
	10	163.55	0.00%	20	173.164	5.88%	3600	173.657	6.18%	1	766	163.55
	Max	-	0.00%	180	-	11.76%	3600	-	11.28%	6	-	-
	Moyenne	-	0.00%	94.5	-	8.37%	3600	-	7.64%	1.8	-	-
	Min	-	0.00%	10	-	5.31%	3600	-	0.53%	1	-	-

Tableau 16 – La comparaison entre les heuristiques basées sur Johnson avec la meilleure solution trouvée “BKS” pour les instances de 20 tâches

Number of job	Instance	Johnson			Johnson <sup>LS</sup>			Johnson <sup>ST</sup>			Johnson <sup>LS+ST</sup>			C <sub>max</sub>	TE <sup>BKS</sup>
		TEC <sub>1</sub>	Gap <sub>1</sub>	Time	TEC <sub>2</sub>	Gap <sub>2</sub>	Time	TEC <sub>3</sub>	Gap <sub>3</sub>	Time	TEC <sub>4</sub>	Gap <sub>4</sub>	Time		
	1	296.294	16.08%	1	282.623	10.72%	1	287.559	12.66%	1	273.887	7.30%	9	1124	255.252
	2	247.156	10.66%	1	236.257	5.78%	1	247.058	10.62%	1	236.159	5.74%	2	1018	223.338
	3	259.767	12.30%	1	249.686	7.94%	1	253.959	9.79%	1	242.169	4.69%	9	1002	231.315
	4	284.191	10.00%	1	275.415	6.60%	1	281.265	8.87%	1	273.489	5.86%	2	1186	258.353
	5	255.578	20.89%	1	244.166	15.49%	1	228.157	7.92%	1	216.656	2.48%	7	1109	211.422
	6	207.742	21.12%	1	195.952	14.25%	1	177.311	3.38%	1	173.972	1.43%	4	1006	171.514
	7	222.706	23.18%	1	216.001	19.47%	1	201.232	11.31%	1	196.687	8.79%	5	938	180.792
	8	239.17	8.78%	1	228.792	4.06%	1	232.81	5.88%	1	226.256	2.90%	7	1042	219.873
	9	250.971	19.25%	1	239.433	13.77%	1	230.429	9.49%	1	220.421	4.73%	2	1048	210.459
	10	242.498	17.24%	1	228.359	10.41%	1	236.699	14.44%	1	223.852	8.23%	9	990	206.835
	11	292.83	10.41%	1	275.011	3.69%	1	291.318	9.84%	1	273.499	3.12%	6	1111	265.226
	12	272.087	12.98%	1	259.726	7.85%	1	259.576	7.78%	2	250.405	3.98%	9	1163	240.831
	13	266.825	13.23%	1	253.73	7.67%	3	260.263	10.44%	8	247.311	4.95%	8	1045	235.652
	14	236.29	8.06%	1	228.2	4.36%	1	231.593	5.91%	5	223.502	2.21%	6	877	218.669
	15	229.219	22.79%	1	218.734	17.17%	1	213.009	14.11%	1	201.894	8.15%	2	862	186.675
	16	211.027	20.08%	1	200.227	13.94%	2	177.624	1.07%	2	175.886	0.09%	4	988	175.735
	17	254.507	3.69%	2	248.861	1.39%	2	256.415	4.46%	3	247.827	0.97%	10	987	245.457
	18	282.609	33.63%	1	265.348	25.47%	1	243.001	14.90%	2	227.338	7.50%	2	1082	211.484
	19	217.559	8.97%	2	213.626	7.00%	2	212.644	6.50%	2	208.711	4.53%	6	836	199.657
	20	308.751	8.50%	3	301.901	6.09%	5	308.749	8.50%	6	300.883	5.73%	9	1110	284.567
	21	285.261	14.91%	1	268.079	7.99%	1	272.636	9.83%	2	261.636	5.39%	2	1180	248.245
	22	215.677	10.76%	1	207.424	6.52%	1	214.057	9.93%	2	205.804	5.69%	9	877	194.725
	23	285.533	13.62%	1	272.646	8.49%	2	279.057	11.04%	6	269.539	7.25%	7	1023	251.307
	24	270.536	13.79%	3	261.949	10.18%	4	266.636	12.15%	7	258.698	8.81%	10	1034	237.743
	25	319.499	9.75%	1	304.659	4.65%	1	309.794	6.41%	2	297.359	2.14%	2	1214	291.128
	26	260.955	21.13%	1	248.385	15.30%	1	238.527	10.72%	2	227.754	5.72%	5	984	215.432
	27	226.907	25.97%	1	214.11	18.87%	4	202.882	12.63%	4	193.418	7.38%	6	1023	180.128
	28	237.391	15.79%	2	222.604	8.58%	2	235.124	14.69%	2	221.155	7.87%	2	993	205.017
	29	270.252	21.23%	1	252.477	13.26%	3	268.214	20.32%	3	250.439	12.34%	7	999	222.926
	30	289.793	11.07%	2	278.552	6.77%	4	281.496	7.89%	4	271.694	4.14%	10	1110	260.901
	Max	-	33.63%	3	-	25.47%	5	-	20.32%	8	-	12.34%	10	-	-
	Moyenne	-	15.33%	1.27	-	10.12%	1.73	-	9.78%	2.53	-	5.34%	5.93	-	-
	Min	-	3.69%	1	-	1.39%	1	-	1.07%	1	-	0.09%	2	-	-

Tableau 17 – La comparaison entre les heuristiques basées sur Hadda avec la meilleure solution trouvée “BKS” pour les instances de 20 tâches.

Number of job	Instance	Hadda			Hadda <sup>S</sup>			Hadda <sup>S+ST</sup>			C <sub>max</sub>	TEC <sup>BKS</sup>			
		TEC <sub>5</sub>	Gaps	Time	TEC <sub>6</sub>	Gap <sub>6</sub>	Time	TEC <sub>7</sub>	Gap <sub>7</sub>	Time			TEC <sub>8</sub>	Gap <sub>8</sub>	Time
	1	280.774	10.00%	8	271.517	6.37%	27	266.969	4.59%	1	262.55	2.86%	173	255.252	1124
	2	225.827	1.11%	4	223.338	0.00%	22	224.833	0.67%	1	223.338	0.00%	51	223.338	1018
	3	245.96	6.33%	1	239.885	3.70%	4	239.259	3.43%	2	231.315	0.00%	17	231.315	1002
	4	278.769	7.90%	12	265.66	2.83%	144	276.909	7.18%	2	262.137	1.46%	408	258.353	1186
	5	249.402	17.96%	2	231.091	9.30%	39	217.512	2.88%	2	212.924	0.71%	493	211.422	1109
	6	197.9	15.38%	2	187.507	9.32%	3	176.524	2.92%	1	171.514	0.00%	36	171.514	1006
	7	214.741	18.78%	2	208.612	15.39%	62	191.979	6.19%	2	187.742	3.84%	424	180.792	938
	8	236.001	7.34%	11	224.605	2.15%	161	230.195	4.69%	4	223.854	1.81%	564	219.873	1042
	9	236.675	12.46%	2	226.415	7.58%	4	218.111	3.64%	1	210.459	0.00%	44	210.459	1048
	10	222.469	7.56%	2	214.638	3.77%	16	213.102	3.03%	1	207.784	0.46%	64	206.835	990
	11	287.682	8.47%	1	269.82	1.73%	6	282.93	6.68%	1	265.226	0.00%	17	265.226	1111
	12	260.813	8.30%	1	250.571	4.04%	4	251.478	4.42%	1	241.638	0.34%	23	240.831	1163
	13	255.996	8.63%	11	244.875	3.91%	131	249.652	5.94%	3	240.568	2.09%	389	235.652	1045
	14	228.884	4.67%	1	221.513	1.30%	7	225.32	3.04%	1	218.669	0.00%	19	218.669	877
	15	214.971	15.16%	3	203.992	9.28%	62	196.567	5.30%	3	188.8	1.14%	278	186.675	862
	16	198.79	13.12%	2	190.627	8.47%	27	178.13	1.36%	4	175.735	0.00%	389	175.735	988
	17	261.947	6.72%	1	246.054	0.24%	10	259.343	5.66%	1	246.054	0.24%	29	245.457	987
	18	260.675	23.26%	1	240.252	13.60%	7	218.936	3.52%	1	211.484	0.00%	120	211.484	1082
	19	211.997	6.18%	1	205.309	2.83%	7	206.867	3.61%	1	202.529	1.44%	20	199.657	836
	20	295.285	3.77%	3	285.576	0.35%	55	294.821	3.60%	1	284.567	0.00%	130	284.567	1110
	21	267.591	7.79%	1	261.269	5.25%	4	254.214	2.40%	1	248.245	0.00%	39	248.245	1180
	22	204.517	5.03%	1	199.415	2.41%	7	201.736	3.60%	1	194.725	0.00%	20	194.725	877
	23	271.066	7.86%	1	259.421	3.23%	8	260.987	3.85%	1	255.593	1.71%	47	251.307	1023
	24	249.735	5.04%	1	245.854	3.41%	7	245.513	3.27%	1	239.791	0.86%	27	237.743	1034
	25	307.616	5.66%	1	297.995	2.36%	4	300.663	3.28%	1	291.128	0.00%	17	291.128	1214
	26	243.418	12.99%	2	232.766	8.05%	26	221.076	2.62%	1	215.432	0.00%	158	215.432	984
	27	214.662	19.17%	1	196.536	9.11%	4	189.654	5.29%	1	180.128	0.00%	54	180.128	1023
	28	214.453	4.60%	1	210.907	2.87%	14	211.876	3.35%	1	205.981	0.47%	42	205.017	993
	29	244.943	9.88%	4	229.643	3.01%	51	236.157	5.94%	1	222.926	0.00%	173	222.926	999
	30	280.487	7.51%	2	272.03	4.27%	26	274.216	5.10%	1	268.382	2.87%	89	260.901	1110
	Max	-	23.26%	12	-	15.39%	161	-	7.18%	4	-	3.84%	564	-	-
	Moyenne	-	9.62%	2.87	-	5.01%	31.63	-	4.04%	1.43	-	0.74%	145.13	-	-
	Min	-	1.11%	1	-	0.00%	3	-	0.67%	1	-	0.00%	17	-	-

20

Tableau 18 – La comparaison entre *HFF*, *Cplex* et *ILSDP* avec la meilleure solution trouvée “BKS” pour les instances de 20 tâches.

Number of job	Instance	HFF		Cplex		ILSDP			C <sub>max</sub>	TEC <sup>BKS</sup>		
		TEC <sub>9</sub>	Gap <sub>9</sub>	Time	TEC <sub>10</sub>	Gap <sub>10</sub>	Time	TEC <sub>11</sub>			Gap <sub>11</sub>	Time
20	1	255.252	0.00%	658	287.559	12.66%	3600	289.5	13.42%	10	1124	255.252
	2	223.338	0.00%	234	247.058	10.62%	3600	236.79	6.02%	1	1018	223.338
	3	231.315	0.00%	316	253.959	9.79%	3600	253.877	9.75%	3	1002	231.315
	4	258.353	0.00%	857	281.265	8.87%	3600	277.113	7.26%	1	1186	258.353
	5	211.422	0.00%	1256	231.844	9.66%	3600	246.011	16.36%	2	1109	211.422
	6	171.514	0.00%	106	177.311	3.38%	3600	194.047	13.14%	3	1006	171.514
	7	180.792	0.00%	868	201.232	11.31%	3600	187.742	3.84%	2	938	180.792
	8	219.873	0.00%	1025	232.81	5.88%	3600	232.561	5.77%	2	1042	219.873
	9	210.459	0.00%	104	250.971	19.25%	3600	245.043	16.43%	2	1048	210.459
	10	206.835	0.00%	688	236.519	14.35%	3600	231.113	11.74%	2	990	206.835
	11	265.226	0.00%	123	291.318	9.84%	3600	283.348	6.83%	1	1111	265.226
	12	240.831	0.00%	270	259.576	7.78%	3600	257.589	6.96%	1	1163	240.831
	13	235.652	0.00%	905	260.262	10.44%	3600	260.002	10.33%	1	1045	235.652
	14	218.669	0.00%	109	231.593	5.91%	3600	226.859	3.75%	1	877	218.669
	15	186.675	0.00%	1556	213.009	14.11%	3600	198.42	6.29%	1	862	186.675
	16	175.735	0.00%	907	177.624	1.07%	3600	183.521	4.43%	3	988	175.735
	17	245.457	0.00%	62	256.415	4.46%	3600	246.049	0.24%	1	987	245.457
	18	211.484	0.00%	1646	243.91	15.33%	3600	251.18	18.77%	2	1082	211.484
	19	199.657	0.00%	411	212.645	6.51%	3600	203.197	1.77%	1	836	199.657
	20	284.567	0.00%	205	308.749	8.50%	3600	308.733	8.49%	1	1110	284.567
	21	248.245	0.00%	511	272.636	9.83%	3600	272.389	9.73%	2	1180	248.245
	22	194.725	0.00%	102	214.057	9.93%	3600	203.77	4.65%	1	877	194.725
	23	251.307	0.00%	629	279.057	11.04%	3600	271.318	7.96%	1	1023	251.307
	24	237.743	0.00%	407	266.636	12.15%	3600	261.203	9.87%	1	1034	237.743
	25	291.128	0.00%	308	309.795	6.41%	3600	308.596	6.00%	1	1214	291.128
	26	215.432	0.00%	627	238.527	10.72%	3600	234.727	8.96%	2	984	215.432
	27	180.128	0.00%	104	202.883	12.63%	3600	217.927	20.98%	2	1023	180.128
	28	205.017	0.00%	561	235.124	14.69%	3600	235.05	14.65%	1	993	205.017
	29	222.926	0.00%	250	268.214	20.32%	3600	237.561	6.56%	1	999	222.926
	30	260.901	0.00%	447	281.496	7.89%	3600	272.187	4.33%	1	1110	260.901
	Max	-	0.00%	1646	-	20.32%	3600	-	20.98%	10	-	-
	Moyenne	-	0.00%	541.73	-	10.18%	3600.00	-	8.84%	1.80	-	-
	Min	-	0.00%	62	-	1.07%	3600	-	0.24%	1	-	-

Tableau 19 – La comparaison entre les heuristiques, *Cplex* et *ILSDP* [Wang et al. (2018)] avec la borne inférieure pour les instances de 10 et 15 tâches.

Number of job	Instance	Johnson <sup>ST</sup>		Johnson <sup>ST</sup> +LS		Hadda		Hadda <sup>LS</sup>		Hadda <sup>ST</sup>		Hadda <sup>LS+ST</sup>		HFF		Cplex		ILSDP		Cmax	LB	
		Gap <sup>LB</sup> <sub>1</sub>	Gap <sup>LB</sup> <sub>2</sub>	Gap <sup>LB</sup> <sub>3</sub>	Gap <sup>LB</sup> <sub>4</sub>	Gap <sup>LB</sup> <sub>5</sub>	Gap <sup>LB</sup> <sub>6</sub>	Gap <sup>LB</sup> <sub>7</sub>	Gap <sup>LB</sup> <sub>8</sub>	Gap <sup>LB</sup> <sub>9</sub>	Gap <sup>LB</sup> <sub>10</sub>	Gap <sup>LB</sup> <sub>11</sub>	Gap <sup>LB</sup> <sub>12</sub>	Gap <sup>LB</sup> <sub>13</sub>	Gap <sup>LB</sup> <sub>14</sub>	Gap <sup>LB</sup> <sub>15</sub>	Gap <sup>LB</sup> <sub>16</sub>	Gap <sup>LB</sup> <sub>17</sub>	Gap <sup>LB</sup> <sub>18</sub>			Gap <sup>LB</sup> <sub>19</sub>
10	1	18.9%	12.4%	18.0%	11.5%	11.8%	11.8%	11.8%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	9.4%	18.0%	18.0%	10.5%	10.5%	574	107.974	
	2	21.2%	18.9%	11.6%	9.2%	21.3%	18.2%	18.2%	18.2%	11.5%	8.9%	8.9%	8.9%	7.2%	7.2%	11.5%	8.9%	8.9%	523	101.075		
	3	23.1%	13.8%	14.6%	7.0%	10.8%	10.8%	10.8%	10.8%	6.2%	4.2%	4.2%	4.2%	4.2%	4.2%	14.6%	16.4%	16.4%	634	137.129		
	4	16.6%	11.2%	16.0%	10.7%	12.0%	11.0%	11.0%	11.0%	11.2%	10.3%	10.3%	10.3%	8.5%	8.5%	16.0%	15.9%	15.9%	656	170.279		
	5	22.5%	15.1%	2.4%	2.4%	10.1%	10.1%	10.1%	10.1%	2.3%	2.3%	2.3%	2.3%	2.3%	2.3%	2.4%	2.4%	6.3%	6.3%	544	87.8278	
	6	26.3%	20.1%	13.7%	8.4%	11.4%	11.4%	11.4%	11.4%	7.3%	7.3%	7.3%	7.3%	6.8%	6.8%	13.7%	17.2%	17.2%	549	87.3649		
	7	32.5%	22.2%	26.8%	16.4%	26.1%	19.2%	19.2%	19.2%	20.4%	13.1%	13.1%	13.1%	10.4%	10.4%	26.5%	12.7%	12.7%	408	90.1962		
	8	16.3%	12.8%	13.4%	11.2%	16.2%	8.3%	8.3%	8.3%	13.3%	6.6%	6.6%	6.6%	6.6%	6.6%	13.4%	13.2%	13.2%	501	103.759		
	9	29.5%	24.5%	12.0%	7.1%	22.9%	21.9%	21.9%	21.9%	5.0%	5.0%	5.0%	5.0%	5.0%	5.0%	12.0%	5.1%	5.1%	521	95.1396		
	10	44.0%	23.0%	18.7%	11.9%	35.8%	12.9%	12.9%	12.9%	15.3%	8.7%	8.7%	8.7%	8.7%	6.3%	11.9%	19.9%	19.9%	520	90.1315		
	Max	44.0%	24.5%	26.8%	16.4%	35.8%	21.9%	21.9%	20.4%	13.1%	13.1%	13.1%	13.1%	10.4%	26.5%	14.0%	12.6%	12.6%	-	-	-	
	Moyenne	25.1%	17.4%	14.7%	9.6%	17.8%	13.6%	13.6%	10.2%	7.6%	7.6%	7.6%	7.6%	6.7%	14.0%	2.4%	2.4%	5.1%	5.1%	-	-	-
	Min	16.3%	11.2%	2.4%	2.4%	10.1%	8.3%	8.3%	2.3%	2.3%	2.3%	2.3%	2.3%	2.3%	2.4%	2.4%	2.4%	5.1%	5.1%	-	-	-
15	1	19.4%	12.6%	17.4%	10.6%	10.5%	8.9%	8.9%	8.1%	8.1%	6.8%	6.8%	6.8%	6.5%	17.4%	17.4%	7.0%	7.0%	809	188.769		
	2	24.5%	17.7%	14.9%	10.0%	18.9%	14.2%	14.2%	7.9%	7.9%	7.1%	7.1%	7.1%	7.0%	14.9%	14.9%	13.0%	13.0%	770	139.005		
	3	17.8%	9.8%	17.5%	9.6%	7.8%	6.6%	6.6%	6.6%	7.1%	6.1%	6.1%	6.1%	5.5%	17.5%	17.5%	17.4%	17.4%	828	190.339		
	4	19.5%	12.7%	16.9%	10.1%	8.4%	7.5%	7.5%	7.5%	5.7%	4.6%	4.6%	4.6%	4.6%	16.9%	16.9%	12.0%	12.0%	974	216.591		
	5	31.9%	23.3%	13.3%	11.0%	31.8%	20.8%	20.8%	20.8%	9.3%	9.3%	9.3%	9.3%	7.6%	13.3%	13.3%	18.6%	18.6%	845	172.732		
	6	22.5%	14.7%	9.9%	7.1%	24.2%	10.4%	10.4%	10.4%	3.9%	3.9%	3.9%	3.9%	3.9%	9.9%	9.9%	15.3%	15.3%	755	132.391		
	7	27.9%	20.0%	10.9%	6.7%	28.2%	18.6%	18.6%	18.6%	10.1%	6.5%	6.5%	6.5%	4.3%	10.8%	10.8%	9.5%	9.5%	725	122.712		
	8	16.0%	12.4%	15.8%	9.3%	13.4%	10.3%	10.3%	10.3%	12.4%	5.2%	5.2%	5.2%	5.2%	15.8%	15.8%	15.7%	15.7%	807	190.858		
	9	21.9%	15.9%	16.2%	10.1%	14.1%	10.0%	10.0%	10.0%	5.9%	5.9%	5.9%	5.9%	5.9%	16.2%	16.2%	16.1%	16.1%	792	170.829		
	10	16.8%	13.0%	10.1%	6.7%	14.3%	11.2%	11.2%	11.2%	4.9%	4.9%	4.9%	4.9%	4.0%	10.1%	10.1%	10.5%	10.5%	766	157.217		
	Max	31.9%	23.3%	17.5%	11.0%	31.8%	20.8%	20.8%	12.4%	9.3%	9.3%	9.3%	9.3%	7.6%	17.5%	17.5%	18.6%	18.6%	-	-	-	
	Moyenne	21.8%	15.2%	14.3%	9.1%	17.2%	11.9%	11.9%	7.5%	6.0%	6.0%	6.0%	6.0%	5.5%	14.3%	14.3%	13.5%	13.5%	-	-	-	
	Min	16.0%	9.8%	9.9%	6.7%	7.8%	6.6%	6.6%	3.9%	3.9%	3.9%	3.9%	3.9%	3.9%	9.9%	9.9%	7.0%	7.0%	-	-	-	

Tableau 20 – La comparaison entre les heuristiques, *Cplex* et *ILSDP* [Wang et al. (2018)] avec la borne inférieure pour les instances de 20 tâches.

Number of job	Instance	Johnson <sup>LS</sup>		Johnson <sup>ST</sup>		Johnson <sup>ST+LS</sup>		Hadda <sup>LS</sup>		Hadda <sup>ST</sup>		Hadda <sup>LS+ST</sup>		Cplex		ILSDP		Cmax	LB
		Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>	Gap <sup>LB</sup>		
	<b>1</b>	21.2%	15.6%	17.6%	12.0%	14.8%	11.0%	9.2%	7.4%	5.2%	17.6%	18.4%	1124	18.4%	17.6%	18.4%	1124	244.563	
	<b>2</b>	16.4%	11.2%	16.3%	11.2%	6.3%	5.2%	5.9%	5.2%	7.9%	7.4%	7.4%	1018	11.5%	5.2%	11.5%	1018	212.399	
	<b>3</b>	16.8%	12.3%	14.2%	8.9%	10.6%	7.9%	7.6%	4.0%	4.0%	4.0%	4.0%	1002	14.2%	4.0%	14.2%	1002	222.32	
	<b>4</b>	17.2%	13.6%	16.0%	12.8%	15.0%	9.6%	14.2%	8.1%	9.6%	8.1%	8.1%	1186	16.0%	8.1%	16.0%	1186	242.418	
	<b>5</b>	25.4%	19.8%	11.9%	6.3%	22.4%	13.4%	6.7%	4.5%	6.7%	4.5%	4.5%	1109	13.8%	4.5%	13.8%	1109	203.811	
	<b>6</b>	27.1%	19.9%	8.5%	6.5%	21.1%	14.7%	8.0%	5.0%	8.0%	5.0%	5.0%	1006	18.8%	5.0%	18.8%	1006	163.407	
	<b>7</b>	26.1%	22.3%	13.9%	11.3%	21.6%	18.1%	8.7%	6.3%	8.7%	6.3%	6.3%	938	13.9%	6.3%	13.9%	938	176.643	
	<b>8</b>	13.5%	8.6%	10.5%	7.4%	12.0%	6.6%	9.2%	6.2%	9.2%	6.2%	6.2%	1042	10.5%	6.2%	10.5%	1042	210.742	
	<b>9</b>	26.2%	20.4%	15.8%	10.8%	19.0%	13.8%	9.7%	5.8%	9.7%	5.8%	5.8%	1048	26.2%	5.8%	26.2%	1048	198.915	
	<b>10</b>	22.0%	14.9%	19.1%	12.6%	11.9%	8.0%	7.2%	4.5%	7.2%	4.5%	4.5%	990	19.0%	4.5%	19.0%	990	198.805	
	<b>11</b>	16.4%	9.4%	15.8%	8.8%	14.4%	7.3%	12.5%	5.5%	7.3%	5.5%	5.5%	1111	15.8%	5.5%	15.8%	1111	251.487	
	<b>12</b>	20.5%	15.0%	14.9%	10.9%	15.5%	10.9%	11.3%	7.0%	11.3%	7.0%	7.0%	1163	14.9%	7.0%	14.9%	1163	225.864	
	<b>13</b>	16.6%	10.9%	13.8%	8.1%	11.9%	7.0%	9.1%	5.2%	9.1%	5.2%	5.2%	1045	13.8%	5.2%	13.8%	1045	228.774	
	<b>14</b>	14.2%	10.3%	11.9%	8.0%	10.6%	7.1%	8.9%	5.7%	8.9%	5.7%	5.7%	877	11.9%	5.7%	11.9%	877	206.879	
	<b>15</b>	31.1%	25.1%	21.8%	15.4%	18.1%	16.6%	12.4%	8.0%	12.4%	8.0%	8.0%	862	21.8%	8.0%	21.8%	862	174.894	
	<b>16</b>	25.4%	19.0%	5.6%	4.5%	18.1%	13.3%	5.9%	4.4%	5.9%	4.4%	4.4%	988	5.6%	4.4%	5.6%	988	168.255	
	<b>17</b>	9.9%	7.4%	10.7%	7.0%	13.1%	6.2%	11.9%	6.2%	11.9%	6.2%	6.2%	987	10.7%	6.2%	10.7%	987	231.676	
	<b>18</b>	38.3%	29.8%	18.9%	11.2%	27.5%	17.5%	7.1%	3.5%	7.1%	3.5%	3.5%	1082	19.3%	3.5%	19.3%	1082	204.393	
	<b>19</b>	12.1%	10.1%	9.6%	7.6%	9.3%	5.8%	6.6%	4.4%	6.6%	4.4%	4.4%	836	9.6%	4.4%	9.6%	836	194.042	
	<b>20</b>	13.6%	11.1%	13.6%	10.8%	8.7%	5.1%	8.5%	3.0%	8.5%	3.0%	3.0%	1110	13.6%	3.0%	13.6%	1110	271.673	
	<b>21</b>	18.3%	11.2%	13.1%	8.5%	11.0%	8.4%	5.5%	4.7%	5.5%	4.7%	4.7%	877	13.1%	4.7%	13.1%	877	186.761	
	<b>22</b>	15.5%	11.1%	14.6%	10.2%	9.5%	6.8%	8.0%	4.3%	8.0%	4.3%	4.3%	1023	14.6%	4.3%	14.6%	1023	236.542	
	<b>23</b>	20.7%	15.3%	18.0%	13.9%	14.6%	9.7%	10.3%	8.1%	9.7%	8.1%	8.1%	1034	18.0%	8.1%	18.0%	1034	227.512	
	<b>24</b>	18.9%	15.1%	17.2%	13.7%	9.8%	8.1%	7.9%	5.4%	7.9%	5.4%	5.4%	1214	17.2%	5.4%	17.2%	1214	274.899	
	<b>25</b>	16.2%	10.8%	12.7%	8.2%	11.9%	8.4%	9.4%	4.1%	9.4%	4.1%	4.1%	984	12.7%	4.1%	12.7%	984	206.927	
	<b>26</b>	26.1%	20.0%	15.3%	10.1%	17.6%	12.5%	6.8%	1.8%	6.8%	1.8%	1.8%	1023	15.3%	1.8%	15.3%	1023	176.881	
	<b>27</b>	28.3%	21.0%	14.7%	9.3%	21.4%	11.1%	7.2%	6.0%	7.2%	6.0%	6.0%	993	14.7%	6.0%	14.7%	993	194.252	
	<b>28</b>	22.2%	14.6%	21.0%	13.8%	10.4%	8.6%	9.1%	7.6%	9.1%	7.6%	7.6%	999	21.0%	7.6%	21.0%	999	207.254	
	<b>29</b>	30.4%	21.8%	29.4%	20.8%	18.2%	10.8%	13.9%	8.4%	13.9%	8.4%	8.4%	1110	29.4%	8.4%	29.4%	1110	253.046	
	<b>30</b>	14.5%	10.1%	11.2%	7.4%	10.8%	7.5%	8.4%	6.1%	8.4%	6.1%	6.1%	-	11.2%	6.1%	11.2%	-	-	
	<b>Max</b>	38.3%	29.8%	29.4%	20.8%	27.5%	18.1%	14.2%	8.1%	14.2%	8.1%	8.1%	-	29.4%	8.1%	29.4%	-	-	
	<b>Moyenne</b>	20.7%	15.3%	14.9%	10.3%	14.7%	9.9%	8.9%	5.5%	9.9%	5.5%	5.5%	-	15.3%	5.5%	15.3%	-	-	
	<b>Min</b>	9.9%	7.4%	5.6%	4.5%	6.3%	5.1%	5.5%	1.8%	5.5%	1.8%	1.8%	-	5.6%	1.8%	5.6%	-	-	

d'un ordonnancement sans augmenter le makespan. En particulier,  $Hadda^{LS+ST}$  et  $HFF$  fournissent des solutions de très bonne qualité dans un temps d'exécution réduit.

Concernant l'approche de résolution, nos heuristiques peuvent être appliquées à tout type de politique de prix de l'électricité telle que Critical-Peak-Pricing, Real-Time-Pricing sans aucune difficulté en raison de la linéarité de la fonction objectif. Il existe également des politiques de tarification de l'électricité dont les fonctions de coût ne sont pas linéaires. Dans ces cas, le coût de l'électricité peut être estimé par une fonction par morceaux [Haouassi et al. 2016], une fonction exponentielle de la consommation d'énergie [Zhang et al. 2017] qui est utilisée pour éviter efficacement le pic de consommation. Nous pouvons étendre notre étude en linéarisant les fonctions objectives non linéaires mais cette approche peut prendre beaucoup de temps pour la résolution. Des travaux supplémentaires sont nécessaires pour trouver des approches efficaces pour ces politiques de tarification.

Après avoir étudié le problème écoénergétique de flow shop à deux machines avec la consommation énergétique dépendant de la caractéristique des tâches. Nous nous intéressons, dans le chapitre suivant, à l'étude d'ordonnancement écoénergétique de flow shop à deux machines dont la consommation énergétique est liée à l'état de fonctionnement des machines.

## 4 Optimisation du coût d'électricité d'un flow shop sous ToU liée aux états de fonctionnement des machines

Dans le chapitre précédent, nous avons étudié l'optimisation du coût d'électricité de la solution optimale de flow shop à deux machines sous *ToU*. La consommation énergétique dans l'étude précédente dépend de caractéristiques des tâches. Dans ce chapitre, nous considérons la minimisation du coût d'électricité de flow shop à deux machines sous *ToU* dont la consommation énergétique est liée aux états de fonctionnement des machines : en marche et en idle.

Tout d'abord, nous donnons les cas optimaux pour certaines structures spécifiques de prix d'électricité de *ToU* qui ne comprennent que deux intervalles de prix. Le cas où le prix de l'électricité diminue, la règle de Johnson et la programmation dynamique donnent une solution optimale. Le cas où le prix de l'électricité augmente, nous fournissons une condition d'optimalité pour la règle de Johnson. Deuxièmement, nous développons une propriété fondée sur la règle de Johnson qui détermine la séquence optimale pour les structures de prix générales de *ToU*. Troisièmement, nous proposons un nouveau modèle mathématique linéaire en nombres entiers mixtes intégrant cette propriété. Ensuite, nous concevons une méthode exacte fondée sur la décomposition de Benders pour résoudre le problème. Enfin, les tests numériques montrent que notre approche proposée améliore significativement la qualité des résultats existants dans la littérature.

Ce chapitre est organisé de la façon suivante. Dans la première section, nous présentons la problématique du problème. Puis, nous rappelons la règle de Johnson et la programmation dynamique proposée par [Wang et al. (2018)] dans la section 2. Dans la section 3, nous proposons les propriétés fondées sur la règle de Johnson qui renforcent l'approche de résolution. Nous présentons le modèle mathématique dans la section 4, y compris le modèle de la littérature et notre modèle proposé. Ensuite, nous proposons la méthode de résolution exacte fondée sur la décomposition de Benders dans la section 5. Dans la section 6, nous menons des tests numériques pour évaluer la performance de nos approches de résolutions. Enfin, nous concluons ce chapitre par les contributions de notre étude.

### 4.1 La problématique

Dans cette étude, nous étudions un problème d'ordonnancement de flow shop de permutation à deux machines avec la prise en compte de coût d'électricité. [Wang et al. (2018)] ont étudié ce problème et l'ont nommé TMPFSEC. En utilisant la notation de Graham [Graham et al. (1979)], nous pouvons référencer le problème comme  $F2|Pu_{ij} = Pu_{ij}, Pu_{idle}, ToU|TEC$ , dont  $Pu_{ij}$  représente la puissance requise pour le traitement de la tâche  $j$  sur la machine  $i$ ,  $Pu_{idle}$  représente la puissance consommée quand la machine est en idle.

Nous prenons un exemple de six tâches pour mieux illustrer notre étude (tableau(21)). Le temps de traitement de chaque tâche est donné dans le tableau (21a). Le tableau (21b) donne la puissance consommée quand les machines sont

en marche et en idle. Nous comparons le coût d'électricité entre deux plans de production  $\pi_1 = J_5 \prec J_2 \prec J_3 \prec J_1 \prec J_4 \prec J_6$  et  $\pi_2 = J_5 \prec J_3 \prec J_2 \prec J_1 \prec J_4 \prec J_6$ . La tarification d'électricité *ToU* est donnée dans la Fig.(15a). L'évolution de la consommation énergétique de  $\pi_1$  et  $\pi_2$  sont données dans les Fig.(15b) et Fig.(15c) respectivement. Le tableau (22) compare le coût d'électricité total entre deux plans de production. L'écart entre deux solution est calculé comme suit :

$$Gap_i = \frac{TEC_i - TEC_1}{TEC_1}$$

Nous trouvons un écart significatif de 29.21% entre la solution  $\pi_1$  et  $\pi_2$ . Alors, il existe une grande possibilité de gain financier pour le problème d'ordonnement de flow shop à deux machines sous *ToU* avec la consommation énergétique liée aux états de fonctionnement des machines : en marche et en idle.

Tableau 21 – Le temps de traitement et la puissance consommée liée aux fonctionnements des machines.

(a) Le temps de traitement de chaque tâche

	Tâche 1	Tâche 2	Tâche 3	Tâche 4	Tâche 5	Tâche 6
<b>Machine 1</b>	8	6	2	6	1	10
<b>Machine 2</b>	9	10	8	5	7	5

(b) La puissance consommée liée aux fonctionnements des machines

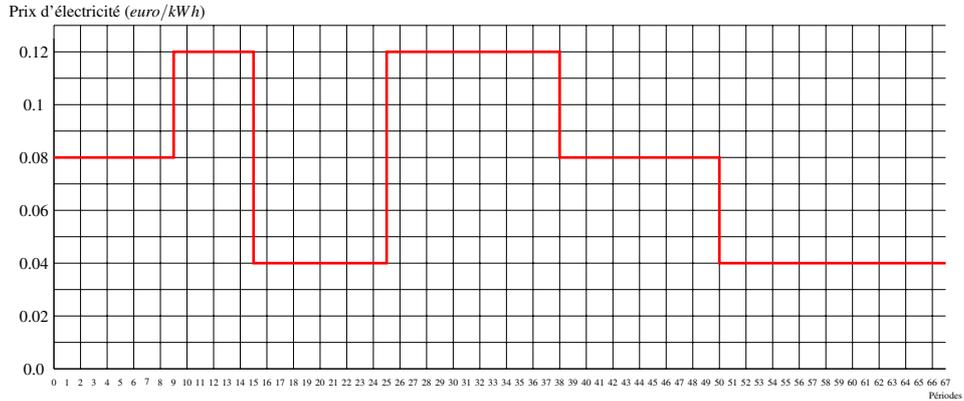
	En marche	En idle
<b>Machine 1</b>	3	1
<b>Machine 2</b>	6	1

Tableau 22 – La comparaison du coût d'électricité total entre  $\pi_1$  et  $\pi_2$ .

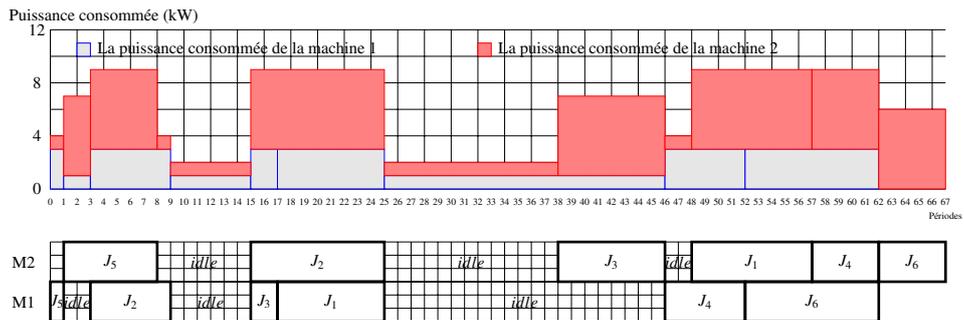
N°	Séquence	Coût d'électricité (TEC)	Gap
<b>1</b>	$J_5 \prec J_2 \prec J_3 \prec J_1 \prec J_4 \prec J_6$	<b>25.6</b>	<b>0.00%</b>
<b>2</b>	$J_5 \prec J_3 \prec J_2 \prec J_1 \prec J_4 \prec J_6$	33.08	29.21%

Une instance de  $F2|Pu_{ij} = Pu_{ij'}, Pu_{idle}, ToU|TEC, Pu_{ij}$  consiste à traiter un ensemble de  $J = \{1, 2, \dots, n\}$  de  $n$  tâches sur un ensemble  $M = \{1, 2\}$  machines. Les machines sont disponibles à partir de l'instant zéro. Chaque tâche  $j \in J$  a une quantité de temps de traitement  $p_{ij}$  sur la machine  $M$ . Les tâches doivent être traitées de manière non préemptive tout d'abord sur la machine 1, puis sur la machine 2. À tout moment, chaque machine ne peut traiter qu'une seule tâche au maximum, et chaque tâche ne peut être traitée que sur une seule machine à la fois.

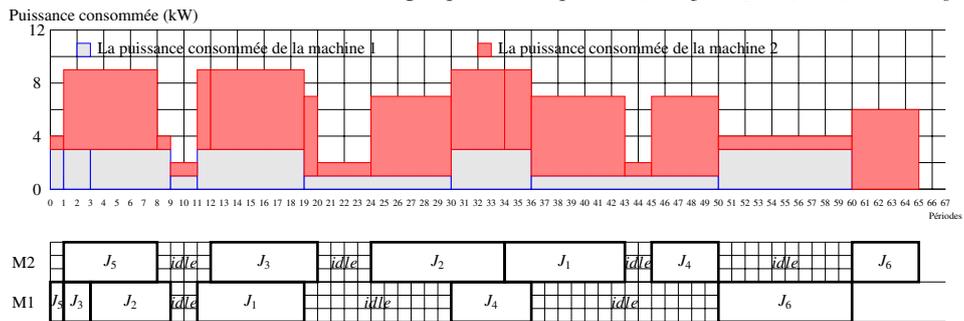
Les machines qui démarrent à l'instant zéro signifie qu'elles sont soit en marche soit en idle à partir de l'instant initial. Nous ne pouvons qu'arrêter



(a) La tarification d'électricité ToU



(b) L'évolution de la consommation énergétique de la séquence  $\pi_1 = J_5 \prec J_2 \prec J_3 \prec J_1 \prec J_4 \prec J_6$



(c) L'évolution de la consommation énergétique de la séquence  $\pi_2 = J_5 \prec J_3 \prec J_2 \prec J_1 \prec J_4 \prec J_6$

Figure 15 – L'évolution de la consommation d'électricité de la solution  $\pi_1$  et  $\pi_2$  sous la tarification ToU.

une machine si et seulement si toutes les tâches sur cette machine ont été traitées. De plus, la machine  $i$  consomme  $b_i$  comme quantité d'énergie par unité de temps lorsqu'elle traite les tâches, elle consomme  $d_i$  comme quantité d'énergie par unité de temps lorsqu'elle est en idle.

La tarification d'électricité dans cette étude est de tarif *ToU*. Nous rappelons ici le mécanisme de tarif *ToU*. L'horizon de travail est composé de  $T$  périodes de temps qui est réparties en  $K$  intervalles de prix, chaque intervalle de prix  $p \in K$  est caractérisé par la date de début  $St_p$ , la durée  $d_p$  et le prix d'électricité  $fc_p$ . Généralement, le prix d'électricité dans un intervalle correspond à un des trois niveaux de prix : "off-peak", "peak" et "semi-peak". Les périodes  $t \in T$  qui se trouvent dans un intervalle de prix  $p$  ont le prix d'électricité  $c_t = fc_p, \forall St_p \leq t \leq St_p + d_p$ .

Nous rappelons la notation des données et des variables qui ont été présentées dans le chapitre précédent, nous présentons également la notation des nouvelles données et nouvelles variables comme ci-dessous :

Les données :

- $M$  : l'ensemble de machines, parcouru par l'indice  $i = 1, 2$
- $J$  : l'ensemble de tâches, parcouru par l'indice  $j, j' = 1, \dots, n$ .
- $T$  : l'horizon de travail, parcouru par l'indice  $t = 1, \dots, T$ .
- $K$  : l'ensemble d'intervalle de prix, parcouru par l'indice  $p$
- $b_i$  : la quantité d'énergie consommée par unité de temps lorsque la machine  $i$  traite les tâches.
- $d_i$  : la quantité d'énergie consommée par unité de temps lorsque la machine  $i$  est en idle.
- $c_t$  : le prix d'électricité de la période  $t$ .
- $fc_k$  : le prix d'électricité de l'intervalle de prix  $k$ .
- $p_{ij}$  : le temps de traitement de tâche  $j$  sur la machine  $i$ .
- $St_k$  : la date de début de l'intervalle de prix  $k$ .
- $e_k$  : la date de fin de l'intervalle de prix  $k$ .
- $V$  : une très grande valeur.

Les variables:

- $S_{ij}$  : la date de début de traitement de tâche  $j$  sur la machine  $i$ .
- $C_{ij}$  : la date d'achèvement de traitement de tâche  $j$  sur la machine  $i$ .

- $F_i$  : la date d'arrêt de la machine  $i$ .
- $g_{it} = 1$  si la machine  $i$  est en marche pendant la période  $t$ , et  $= 0$  sinon.
- $z_{ijt} = 1$  si la machine  $i$  traite la tâche  $j$  pendant la période de temps  $t$ , et  $= 0$  sinon.
- $\alpha_{ijp}$  : la quantité de traitement de la tâche  $j$  sur la machine  $i$  pendant l'intervalle de prix  $p$ .
- $\beta_{ijp} = 1$  si la tâche  $j$  est traitée sur la machine  $i$  pendant l'intervalle de prix  $p$ , et  $= 0$  sinon.
- $x_{jj'} = 1$  si la tâche  $j$  précède la tâche  $j'$ , et  $= 0$  sinon.
- $\gamma_{jp} = 1$  si la tâche  $j$  a commencé à être traitée sur la machine 2 pendant l'intervalle de prix  $p$ , et  $= 0$  sinon.
- $\omega_{jj'}^p = 1$  si la tâche  $j$  et la tâche  $j'$  ont commencé leurs traitements sur la machine 2 pendant le même intervalle de prix  $p$ , et  $= 0$  sinon.
- $TEC$  : le coût d'électricité total.

## 4.2 La règle de Johnson et la programmation dynamique

Avant d'aborder le problème, nous rappelons brièvement deux résultats de la littérature qui conviennent à notre approche de résolution. Premièrement, la règle de Johnson [Johnson (1954)] peut trouver la solution optimale en  $O(n \log n)$  étapes pour le problème  $F2|perm|C_{max}$ . Deuxièmement, la programmation dynamique (DP) proposée par [Wang et al. (2018)] résout de manière optimale  $F2|Pu_{ij} = Pu_{ij}, Pu_{idle}, ToU|TEC$  lorsque la séquence des tâches est prédéterminée.

### 4.2.1 La règle de Johnson

Pour toutes les instances de  $(F2|perm|C_{max})$ , en fonction du temps de traitement, nous distinguons les tâches en deux ensembles  $\bar{J} = \{J_j \in J | p_{1j} < p_{2j}\}$  et  $\underline{J} = \{J_j \in J | p_{2j} < p_{1j}\}$ . Les tâches dont  $p_{1j} = p_{2j}$  peuvent être placées arbitrairement soit dans  $\bar{J}$  soit dans  $\underline{J}$ . Pour obtenir la séquence de Johnson, nous appliquons le processus ci-dessous :

1. Trier  $\bar{J}$  dans l'ordre non-décroissant des temps de traitement sur la machine 1.
2. Trier  $\underline{J}$  dans l'ordre non-croissant des temps de traitement sur la machine 2.
3. La séquence optimale est constituée de  $\bar{J}$  suivi de  $\underline{J}$  après avoir les triés.

#### 4.2.2 La programmation dynamique [Wang et al. (2018)]

Nous rappelons brièvement la programmation dynamique proposée par [Wang et al. (2018)]. Soit  $\pi$  une séquence de tâches donnée, la DP recherche la date de début optimale de chaque tâche pour  $F2|Pu_{ij} = Pu_{ij'}, Pu_{idle}, ToU|TEC$ . Soit  $\underline{C}_{i,j}$  and  $\bar{C}_{i,j}$  respectivement les bornes inférieure et supérieure pour la date de fin de la tâche à la  $j^{eme}$  position dans  $\pi$  ( $\pi(j)$ ) sur la machine  $i$ . L'horizon de travail  $T$  et la séquence  $\pi$  étant donnés, nous pouvons calculer  $\underline{C}_{i,j}$  et  $\bar{C}_{i,j}$  pour toutes les tâches  $\pi(j) \in J$  par les équations ci-dessous :

$$\underline{C}_{1,j} = \sum_{l=1}^j p_{1,l}, \forall j \in J; \quad (18)$$

$$\underline{C}_{2,1} = \sum_{i=1}^2 p_{i,1}; \quad (19)$$

$$\underline{C}_{2,j} = \max\{\underline{C}_{2,j-1}, \underline{C}_{1,j}\} + p_{2,j}; \forall j \in J \setminus \{1\}; \quad (20)$$

$$\bar{C}_{2,j} = T - \sum_{l=j+1}^n p_{2,l}; \forall j \in J; \quad (21)$$

$$\bar{C}_{1,j} = \min\{\bar{C}_{1,j+1} - p_{1,j+1}, \bar{C}_{2,j}\} - p_{2,j}; \forall j \in J \setminus \{n\}; \quad (22)$$

$$\bar{C}_{1,n} = T - p_{2,n}; \forall j \in J \setminus \{n\}; \quad (23)$$

Une date de fin réalisable de la tâche  $\pi(j)$  doit satisfaire deux conditions :

1.  $\underline{C}_{1,j} \leq C_{1,j} \leq \bar{C}_{1,j}$ .
2.  $\max\{C_{1,j} + p_{2,j}, \underline{C}_{2,j}\} \leq C_{2,j} \leq \bar{C}_{2,j}$ .

La DP énumère l'ensemble des dates de fin réalisables pour les tâches sur deux machines en  $O(T^2)$  étapes. Puis, la DP détermine la date d'achèvement optimale de la tâche  $\pi(j)$  à partir de l'ensemble des dates d'achèvement réalisables, ce qui a le coût total d'électricité minimal. La DP procède de manière récurrente de la première position à la dernière position de la séquence  $\pi$  en  $O(nT^4)$  étapes pour obtenir le coût d'électricité total minimum. Le détail de DP et la relation de récurrence sont donnés ci-dessous :

Nous définissons tout d'abord :

- $B_{it} = b_i \sum_{k=1}^t c_k$  : le coût d'électricité de la machine  $i$  lorsqu'elle est en marche de la période  $k = 1$  à la période  $t$ .
- $D_{it} = d_i \sum_{k=1}^t c_k$  : le coût d'électricité de la machine  $i$  lorsqu'elle est en idle de la période  $k = 1$  à la période  $t$ .

Alors, nous avons  $B_{it'} - B_{it}$  le coût d'électricité de la machine  $i$  lorsqu'elle est en marche dans un intervalle de temps  $[t, t']$ . Nous avons le même résultat pour le coût d'électricité lorsque la machine est en idle.

Soit  $\Omega(j, \phi, \psi)$  l'ensemble de date d'achèvement réalisable de la tâche à la  $(j-1)^{eme}$  position sur deux machines lorsque la tâche à la  $j^{eme}$  position a  $C_{1,j} = \phi$  et  $C_{2,j} = \psi$ . Alors, nous avons :

$$\Omega(j, \phi, \psi) = \{(\phi', \psi') | \underline{C}_{1,j-1} \leq \phi' \leq \min\{\bar{C}_{1,j-1}, \phi - p_{1,j}\}, \\ \max\{\phi' + p_{2,j-1}, \underline{C}_{2,j-1}\} \leq \psi' \leq \min\{\bar{C}_{2,j-1}, \psi - p_{2,j}\}\}.$$

Soit  $TEC(j, \phi, \psi)$  le coût d'électricité minimum de  $\Omega(j, \phi, \psi)$ . Si  $\Omega(j, \phi, \psi) = \emptyset$ , nous avons  $TEC(j, \phi, \psi) = +\infty$ . Nous pouvons calculer  $TEC(j, \phi, \psi)$  grâce à la relation de récurrence suivante :

1.

$$TEC(0, 0, 0) = 0.$$

2.

$$TEC(j, \phi, \psi) = \min_{\phi', \psi'} \{B_{1\phi} - B_{1,\phi-p_{1,j}} + B_{2,\psi-p_{2,j}} + D_{1,\phi-p_{1,j}} - D_{1,\phi'} \\ + D_{2,\psi-p_{2,j}} - D_{2,\psi'} + TEC(j-1, \phi', \psi') | (\phi', \psi') \in \Omega(j, \phi, \psi)\}$$

Alors, le coût d'électricité total optimal d'une séquence  $\pi$  donnée est obtenu par l'équation ci-dessous :

$$TEC(\pi) = \min_{\phi', \psi'} \{TEC(n, \phi, \psi) | \underline{C}_{1,j} \leq C_{1,j} \leq \bar{C}_{1,j}, \max\{C_{1,j} + p_{2,j}, \underline{C}_{2,j}\} \leq C_{2,j} \leq \bar{C}_{2,j}\}$$

### 4.3 Les propositions fondées sur la règle de Johnson

Dans cette section, nous développons certaines propriétés fondées sur la règle de Johnson pour améliorer l'approche de résolution. Tout d'abord, nous présentons des configurations spécifiques de tarification  $ToU$  à deux intervalles de prix. Nous montrons que dans ces configurations, la règle de Johnson [Johnson (1954)] avec la DP donne la solution optimale. Ensuite, nous développons une propriété fondée sur la règle de Johnson pour la configuration générale de  $ToU$ .

#### 4.3.1 Deux intervalles de prix

Nous supposons que l'horizon de travail se compose de deux intervalles de prix  $p$  et  $(p+1)$  dont les prix de l'électricité associés sont respectivement  $fc_p$  et  $fc_{p+1}$ .  $p$  précède  $p+1$ .

Soit  $\pi_{Johnson}$  la séquence de Johnson,  $\delta_1$  le temps d'inactivité positif le plus petit sur la machine 2 lorsque toutes les tâches de  $\pi_{Johnson}$  commencent au plus tôt possible. Pour chaque instance donnée, nous pouvons calculer :  $A_j = \frac{p_{1j} - p_{2j}}{\min(\delta_1, p_{2j})}, \forall j \in \mathcal{J}$ . Nous rappelons que  $b_2$  et  $d_2$  sont les puissances énergétiques consommées par unité de temps lorsque la machine 2 est en marche et en idle respectivement.

**Propriété 1.** Lorsque le problème  $F2|Pu_{ij} = Pu_{ij}', Pu_{idle}, ToU|TEC$  n'a que deux intervalles de prix, l'application de DP sur la séquence de Johnson donne la solution optimale dans les cas suivants :

1.  $fc_p \geq fc_{p+1}$ .
2.  $fc_p < fc_{p+1}$  et  $\frac{b_2 \times (fc_{p+1} - fc_p)}{d_2 \times fc_p} \leq A_j, \forall j \in J$ .

*Démonstration.*

**Le prix d'électricité est décroissant ( $fc_p > fc_{p+1}$ ) :**

Dans cette section, l'horizon de travail se compose de deux intervalles de prix  $p$  et  $(p+1)$ . L'intervalle  $p$  correspond à la période "peak" avec le prix de l'électricité associé  $fc_p$ , l'intervalle  $(p+1)$  correspond à la période "off-peak" avec le prix de l'électricité associé  $fc_{p+1}$ ,  $fc_{p+1} < fc_p$ .

Soit  $\pi$  une solution optimale qui ne suit pas la règle de Johnson. Soit  $C_{2\pi}$  et  $C_{1\pi}$  la date d'achèvement sur la machine 2 et sur la machine 1 respectivement de la séquence  $\pi$ . Soit  $C_{2Johnson}$  et  $C_{1Johnson}$  la date d'achèvement sur la machine 2 et sur la machine 1 respectivement de la séquence de Johnson  $\pi_{Johnson}$ . Comme nous le savons,  $C_{1Johnson} \leq C_{1\pi}$  et  $C_{2Johnson} \leq C_{2\pi}$  pour n'importe quelle séquence  $\pi$ . Nous pouvons donc toujours retarder certaines tâches de la séquence de Johnson pour obtenir  $C_{2Johnson}^* = C_{2\pi}$  et  $C_{1Johnson}^* = C_{1\pi}$ . ( $C_{2Johnson}^*$  et  $C_{1Johnson}^*$  sont les dates d'achèvement sur la machine 2 et sur la machine 1 respectivement de la séquence de Johnson après avoir été retardée).

Soit  $\Delta_1$  et  $\Delta_2$  le temps idle sur la machine 1 et sur la machine 2 respectivement. Nous avons  $\Delta_1 = C_{1\pi} - \sum_{j \in J} p_{1j}$  et  $\Delta_2 = C_{2\pi} - \sum_{j \in J} p_{2j}$ . Toutes les séquences ayant les mêmes  $C_{2\pi}$  et  $C_{1\pi}$ , on obtient les mêmes  $\Delta_1$  et  $\Delta_2$ . Si  $fc_p = fc_{p+1}$ , alors nous avons  $TEC_{Johnson} = TEC_{\pi} = (\sum_{j \in J} p_{1j} \times b_1 + \sum_{j \in J} p_{2j} \times b_2) \times fc_p + (\Delta_1 \times d_1 + \Delta_2 \times d_2) \times fc_p$ . Si  $fc_p \neq fc_{p+1}$ , la différence se pose sur la charge de traitement de tâches dans chaque intervalle de prix. Entre deux solutions, une solution dont la charge de traitement de tâches dans la période  $p$  est supérieure à celle de l'autre solution ont également une quantité de temps idle dans la période  $(p+1)$  plus importante. Alors, la différence en terme de coût total d'électricité entre  $\pi$  et la séquence de Johnson  $\pi_{Johnson}$  peut être estimée comme suit :

$$TEC_{Johnson} - TEC_{\pi} = \gamma_1 \times b_2 \times (fc_p - fc_{p+1}) - \gamma_2 \times d_2 \times (fc_p - fc_{p+1})$$

Dont :

- $\gamma_1$  est la différence en termes de charge de traitement de tâches dans les périodes "off-peak" entre la séquence  $\pi$  et la solution de Johnson  $\pi_{Johnson}$ .
- $\gamma_2$  est la différence en termes de temps idle dans les périodes "off-peak" entre la solution de Johnson  $\pi_{Johnson}$  et la séquence  $\pi$ .

La séquence de Johnson nous permet de retarder certaines tâches pour obtenir  $\gamma_1 = 0$ . Cela signifie que la charge de traitement de tâches en période "peak" et en

période “off-peak” sont les mêmes entre la séquence de Johnson et la séquence  $\pi$ . Puisque les prix de l’électricité sont décroissants, cela conduit à  $TEC_{Johnson} - TEC_{\pi} = -\gamma_2 \times d_2 \times (fc_p - fc_{p+1}) \leq 0$ .

Comme  $TEC_{Johnson} - TEC_{\pi} \leq 0$ , alors la règle de Johnson avec la DP donne la solution optimale en termes de coût d’électricité total.

**Le prix d’électricité est croissant ( $fc_p < fc_{p+1}$ ) :**

Dans cette section, l’horizon de travail se compose de deux intervalles de prix  $p$  et  $(p+1)$ . L’intervalle  $p$  correspond à la période “off-peak” avec le prix de l’électricité associé  $fc_p$ , l’intervalle  $(p+1)$  correspond à la période “peak” avec le prix de l’électricité associé  $fc_{p+1}$ ,  $fc_p < fc_{p+1}$ .

Nous comparons le coût d’électricité total entre la séquence de Johnson et une séquence  $\pi$  ( $TEC_{Johnson}$  et  $TEC_{\pi}$  respectivement). La séquence  $\pi$  est obtenue en échangeant deux tâches  $J_k$  et  $J_j$  dans la séquence Johnson. Nous considérons deux cas possibles comme suit :

- 1<sup>er</sup> cas :  $J_k, J_j \in \bar{J}$  ou  $J_k, J_j \in \underline{J}$ . Nous avons  $TEC_{Johnson} \leq TEC_{\pi}$ . La règle de Johnson donne la séquence optimale car tout échange entre  $J_k$  et  $J_j$  conduit à plus de temps idle ou à exploiter moins les périodes dont le prix d’électricité est moins cher.
- 2<sup>eme</sup> cas,  $J_k \in \bar{J}$  et  $J_j \in \underline{J}$ .
  - Si  $p_{1J_j} > p_{1J_k}$ , alors la solution de Johnson donne le meilleur résultat car une permutation quelconque entre  $J_k$  et  $J_j$  entraîne plus de temps idle dans les périodes “off-peak”.
  - Si  $p_{1J_j} \leq p_{1J_k}$ , nous trouvons dans les Fig.(16a) et Fig.(16b) un exemple de permutation entre  $J_k$  et  $J_j$  de la séquence  $\pi$ . Nous notons que la période d’inactivité immédiatement avant  $J_k$  sur la machine 1 et sur la machine 2 sont respectivement  $\Delta_1$  et  $\Delta_2$ . Soit  $\delta_1$  le “temps d’inactivité” sur la machine 2 de la séquence de Johnson et  $\delta_2$  le “temps d’inactivité” sur la machine 2 de la séquence  $\pi$ .

Nous avons :

$$\delta_1 = \Delta_1 + p_{1J_k} - \Delta_2$$

Pour estimer  $\delta_2$ , soit  $J_i$  la tâche qui succédait directement la tâche  $J_k$  dans la séquence de Johnson. Nous considérons deux configurations.

Si  $J_i \in \bar{J}$ , selon la règle de Johnson, nous avons :  $p_{1J_i} \geq p_{1J_k}$ . Dans ce cas, nous avons :

$$\begin{aligned} \delta_2 &= \Delta_1 + p_{1J_j} + p_{1J_i} - (\Delta_2 + p_{2J_j}) \\ &\geq \Delta_1 + p_{1J_j} + p_{1J_k} - (\Delta_2 + p_{2J_j}) \end{aligned}$$

Si  $J_i \in \underline{J}$ ,  $J_i$  précède  $J_j$ , alors  $p_{2J_k} \geq p_{2J_j}$  qui est conforme à la règle de Johnson. Dans ce cas, si  $p_{1J_i} \leq p_{1J_k}$  alors la permutation entre  $J_i$  et  $J_k$  conduit à moins de “ temps d’inactivité ” que la permutation entre  $J_k$

et  $J_j$ . Alors, il vaut mieux étudier la permutation entre  $J_i$  et  $J_k$ . Nous échangeons  $J_i$  avec  $J_k$ . Avec un changement d'indice de  $i$  à  $j$ , nous avons :

$$\delta_2 = \Delta_1 + p_{1J_j} + p_{1J_k} - (\Delta_2 + p_{2J_j})$$

Alors dans tous les cas, nous avons toujours :

$$\delta_2 \geq \Delta_1 + p_{1J_j} + p_{1J_k} - (\Delta_2 + p_{2J_j})$$

Nous avons :

$$\begin{aligned} \delta_2 - \delta_1 &\geq \Delta_1 + p_{1J_j} + p_{1J_k} - (\Delta_2 + p_{2J_j}) - (\Delta_1 + p_{1J_k} - \Delta_2) \\ &\geq p_{1J_j} - p_{2J_j} \end{aligned}$$

Soit  $TEC_\pi$  et  $TEC_{Johnson}$  le coût total d'électricité de la séquence  $\pi$  et de la séquence de *Johnson* respectivement. Soit  $G$  le meilleur gain en permutant  $J_j$  et  $J_k$ :

$$G = \min(\delta_1, p_{2J_j}) \times b_2 \times (fc_{p+1} - fc_p)$$

Dont  $b_2$  est la puissance consommée par unité de temps de la machine 2 lorsqu'elle est en marche.

La valeur  $G$  représente le gain en coût d'électricité en traitant une quantité de charge de tâche égale à  $\min(\delta_1, p_{2J_j})$  pendant les périodes "off-peak" au lieu de les traiter pendant les périodes "peak".

Soit  $P$  la perte minimale, c'est-à-dire l'augmentation minimale du coût de l'électricité par rapport à la séquence de *Johnson* :

$$\begin{aligned} P &\geq (\delta_2 - \delta_1) \times d_2 \times fc_p \\ &\geq (p_{1J_j} - p_{2J_j}) \times d_2 \times fc_p \end{aligned}$$

Dont  $d_2$  est la puissance consommée par unité de temps de la machine 2 lorsqu'elle est en idle.

La valeur  $P$  représente l'augmentation du coût de l'électricité. Cette augmentation est en fonction de la quantité de temps idle différente en permutant  $J_j$  et  $J_k$ . De plus, nous supposons que ces temps idle augmentés sont situés sur un intervalle de période "off-peak".

Nous avons :

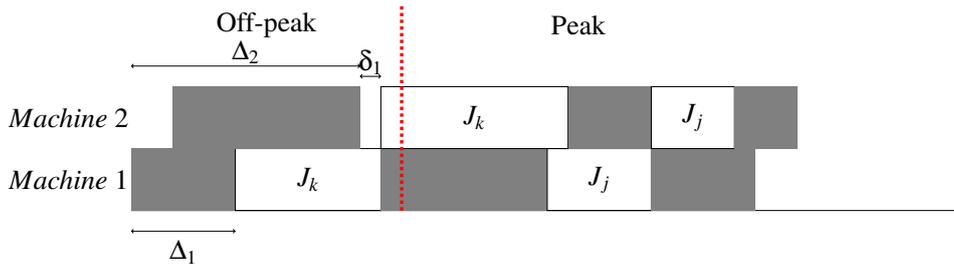
$$\begin{aligned} TEC_{Johnson} - TEC_\pi &\leq G - P \\ &\leq \min(\delta_1, p_{2J_j}) \times b_2 \times (fc_{p+1} - fc_p) \\ &\quad - (p_{1J_j} - p_{2J_j}) \times d_2 \times fc_p \end{aligned}$$

Soit  $A = G - P$ . Si  $A \leq 0$ , alors la solution de Johnson domine autres séquences en termes de coût d'électricité total.

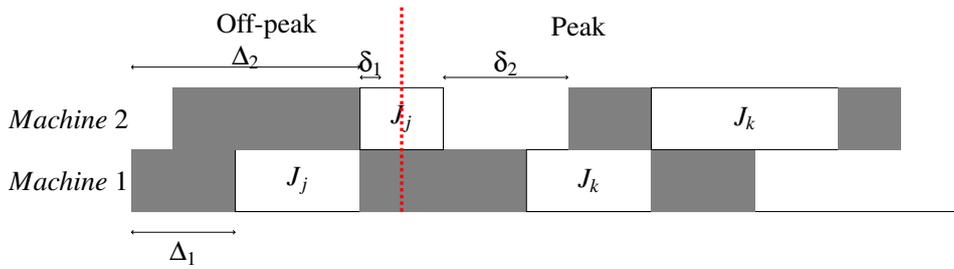
$$\begin{aligned} & \min(\delta_1, p_{2J_j}) \times b_2 \times (fc_{p+1} - fc_p) - (p_{1J_j} - p_{2J_j}) \times d_2 \times fc_p \leq 0 \\ \Leftrightarrow & \min(\delta_1, p_{2J_j}) \times b_2 \times (fc_{p+1} - fc_p) \leq (p_{1J_j} - p_{2J_j}) \times d_2 \times fc_p \\ \Leftrightarrow & \frac{b_2 \times (fc_{p+1} - fc_p)}{d_2 \times fc_p} \leq \frac{p_{1J_j} - p_{2J_j}}{\min(\delta_1, p_{2J_j})} \end{aligned}$$

Nous notons la tâche  $J_j$  par  $j$  pour simplifier la notation. Alors, dans le cas de deux intervalles de prix et les prix de l'électricité augmentent, la règle de Johnson donne la solution optimale si :

$$\frac{b_2 \times (fc_{p+1} - fc_p)}{d_2 \times fc_p} \leq \frac{p_{1j} - p_{2j}}{\min(\delta_1, p_{2j})}, \forall j \in \underline{J}$$



(a) La séquence de Johnson



(b) La séquence  $\pi$

Figure 16 – La permutation entre  $J_k$  et  $J_j$

□

#### 4.3.2 Plusieurs intervalles de prix

Pour les cas plus généraux, où la propriété (1) n'est pas valide, nous présentons une propriété qui sera prise en compte dans notre approche. Dans le tarif de l'électricité

$ToU$ , nous supposons que chaque intervalle de prix est relativement long pour exécuter plusieurs tâches. Pour chaque intervalle de prix  $p$  avec sa date de début  $St_p$  et sa date de fin  $e_p$ , nous pouvons déterminer un ensemble de tâches dont les dates de début sur la machine 2 sont exécutées pendant cet intervalle de prix. Nous prouvons qu'en ordonnant ces tâches selon la règle de Johnson, nous pourrions obtenir une solution aussi efficace que n'importe quelle autre séquence en terme de coût d'électricité.

**Propriété 2.** Soit  $E_p \subset J$  un ensemble des tâches dont les dates de début sur la machine 2 sont exécutées pendant l'intervalle de prix  $p$  ( $\forall j \in E_p, St_p \leq S_{2j} \leq e_p$ ). Pour tout séquence de tâches appartenant à  $E_p$ , réorganiser ces tâches en suivant la règle de Johnson ne détériore pas la solution de  $F2|Pu_{ij} = Pu_{ij}, Pu_{idle}, ToU|TEC$ .

*Démonstration.*

### Deux intervalles de prix $p$ et $p + 1$

Nous étudions dans cette partie une séquence  $\pi^*$  de l'ensemble des tâches  $E_p \subset J$  et  $E_{p+1} \subset J$  dont la date de début sur la machine 2 se trouve dans une même période de prix soit dans  $p$  soit dans  $p + 1$  respectivement.

Nous montrons que pour une séquence  $\pi^*$  quelconque, le réarrangement des tâches de  $E_p$  et des tâches de  $E_{p+1}$  selon la règle de Johnson n'augmente pas le coût d'électricité de la séquence  $\pi^*$ .

### L'ensemble $E_p$

Nous avons :

- $E_p$  : l'ensemble des tâches de la séquence  $\pi$  dont la date de début sur la machine 2 se trouve dans l'intervalle de prix  $p$ .
- $S_{i\pi}$  : la date de début sur la machine  $i$  de la séquence  $\pi$ .
- $C_{i\pi}$  : la date de fin sur la machine  $i$  de la séquence  $\pi$ .
- $D_{i\pi}^p$  : le temps d'idle total dans l'intervalle de prix  $p$  de la machine  $i$  de séquence  $\pi$ .
- $P_{i\pi}^p$  : la charge totale de traitement des tâches sur la machine  $i$  dans l'intervalle de prix  $p$  de la séquence  $\pi$ .

Supposons qu'il existe une séquence  $\pi^*$  optimale pour l'ensemble de tâches  $E_p$ .

Comme toutes les tâches  $J_j$  de  $E_p$  commencent sur la machine 2 pendant l'intervalle de prix  $p$ , nous avons :

$$D_{1\pi^*}^{(p+1)} = 0. \quad (24)$$

$$P_{1\pi^*}^p = \sum_{j \in E_p} p_{1j}. \quad (25)$$

$$D_{1\pi^*}^p + P_{1\pi^*}^p = C_{1\pi^*}. \quad (26)$$

$$P_{1\pi^*}^{(p+1)} = 0. \quad (27)$$

$$D_{2\pi^*}^{(p+1)} = 0. \quad (28)$$

$$P_{2\pi^*}^{(p+1)} = \max(0, C_{2\pi^*} - e_p). \quad (29)$$

$$P_{2\pi^*}^p + P_{2\pi^*}^{(p+1)} = \sum_{j \in E_p} p_{2j}. \quad (30)$$

$$St_p + D_{2\pi^*}^p + P_{2\pi^*}^p + P_{2\pi^*}^{(p+1)} = C_{2\pi^*}. \quad (31)$$

Toutes les tâches  $J_j$  de  $E_p$  commencent leurs traitements sur la machine 2 pendant l'intervalle de prix  $p$ , cela signifie que tous les traitements sur la machine 1 doivent être terminés avant la date de fin de l'intervalle de prix  $p$ . Les équations (24), (25) et (27) représentent ce résultat.

La dernière tâche de la séquence  $\pi^*$  commence sur la machine 2 à l'intervalle de prix  $p$  alors cette tâche peut être terminée soit dans l'intervalle de prix  $p$  ( $P_{2\pi^*}^{(p+1)} = 0$ ) soit dans l'intervalle de prix  $p+1$  ( $P_{2\pi^*}^{(p+1)} = C_{2\pi^*} - e_p$ ). Dans tous les cas, nous avons toujours :  $D_{2\pi^*}^{(p+1)} = 0$ . Les équations (28) et (29) représentent ce résultat.

Nous avons le coût d'électricité total de la séquence  $\pi^*$ :

$$\begin{aligned} TEC_{\pi^*} &= \sum_{i \in M} D_{i\pi^*}^p \times d_i \times fc_p + \sum_{i \in M} P_{i\pi^*}^p \times b_i \times fc_p \\ &+ \sum_{i \in M} D_{i\pi^*}^{(p+1)} \times d_i \times fc_{p+1} + \sum_{i \in M} P_{i\pi^*}^{(p+1)} \times b_i \times fc_{p+1} \end{aligned}$$

Nous considérons maintenant la séquence  $\pi$  de  $E_p$  dont l'ordre des tâches suivent la règle de Johnson.

Les équations (24)-(31) sont toujours valables pour la séquence  $\pi$ . En plus, comme  $\pi$  suivent la règle de Johnson, nous avons :

$$C_{1\pi} \leq C_{1\pi^*}. \quad (32)$$

$$C_{2\pi} \leq C_{2\pi^*}. \quad (33)$$

A partir de l'équation (33), en retardant le traitement des tâches sur la machine 2, nous pouvons toujours obtenir :  $C_{2\pi} = C_{2\pi^*}$ . Alors, à partir de l'équation (29), nous avons :  $P_{2\pi}^{(p+1)} = P_{2\pi^*}^{(p+1)} = \max(0, C_{2\pi^*} - e_p)$ . Avec les équations (30) et (31), nous obtenons également :  $D_{2\pi}^p = D_{2\pi^*}^p$  et  $P_{2\pi}^p = P_{2\pi^*}^p$ .

Les équations (25), (26) et (32), entraînent que :  $P_{1\pi^*}^p = P_{1\pi}^p = \sum_{j \in E_p} p_{1j}$  et  $D_{1\pi}^p \leq D_{1\pi^*}^p$ .

Alors avec n'importe quelle séquence  $\pi$  qui suivent la règle de Johnson, nous pouvons obtenir :

$$\begin{aligned} TEC_{\pi} &= \sum_{i \in M} D_{i\pi}^p \times d_i \times fc_p + \sum_{i \in M} P_{i\pi}^p \times b_i \times fc_p \\ &+ \sum_{i \in M} D_{i\pi}^{(p+1)} \times d_i \times fc_{p+1} + \sum_{i \in M} P_{i\pi}^{(p+1)} \times b_i \times fc_{p+1}. \\ &\leq TEC_{\pi^*}. \end{aligned}$$

Nous considérons un exemple illustré dans la Fig.(17). Nous constatons que le temps d'attente et le temps de traitement de jobs répartis dans chaque période de prix  $p$  et  $p+1$  ne changent pas entre  $\pi^*$  et  $\pi$ , donc  $TEC_{\pi} = TEC_{\pi^*}$ .

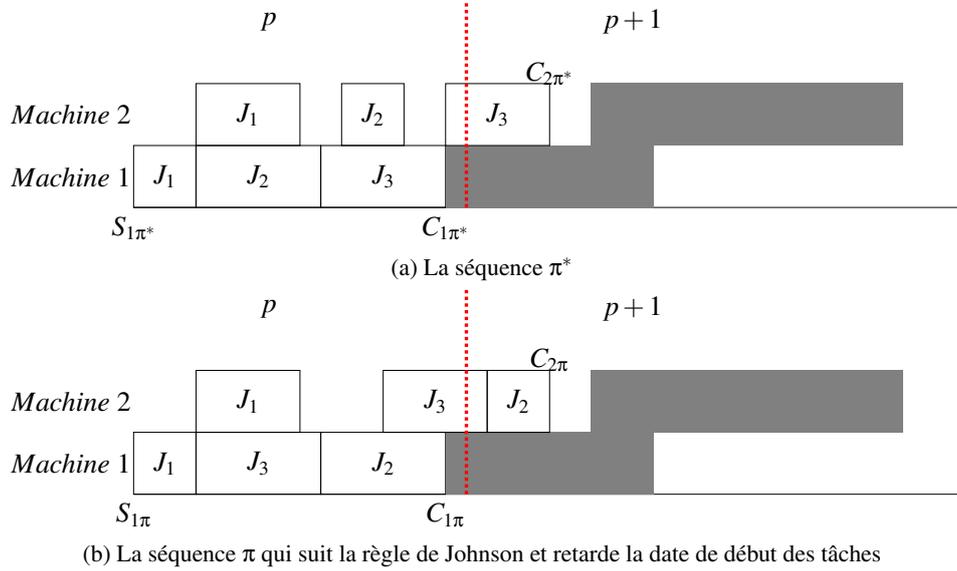


Figure 17 – La comparaison entre la séquence  $\pi^*$  et la séquence de  $\pi$

### L'ensemble $E_{p+1}$

Soit  $\pi^*$  la séquence optimale pour l'ensemble  $E_{p+1}$ . Comme toutes les tâches  $E_{p+1}$  commencent sur la machine 2 pendant l'intervalle de prix  $p+1$ , nous avons :

$$D_{2\pi^*}^p = 0. \quad (34)$$

$$P_{2\pi^*}^p = 0. \quad (35)$$

$$P_{2\pi^*}^{(p+1)} = \sum_{j \in E_{p+1}} p_{2j}. \quad (36)$$

$$P_{1\pi^*}^p = \max(0, e_p - S_{1\pi^*} - D_{1\pi^*}^p). \quad (37)$$

$$P_{1\pi^*}^p + P_{1\pi^*}^{(p+1)} = \sum_{j \in E_{p+1}} p_{1j}. \quad (38)$$

$$S_{1\pi^*} + D_{1\pi^*}^p + D_{1\pi^*}^{(p+1)} + P_{1\pi^*}^p + P_{1\pi^*}^{(p+1)} = C_{1\pi^*}. \quad (39)$$

Toutes les tâches de  $E_{p+1}$  commencent sur la machine 2 pendant l'intervalle de prix  $p+1$ , alors la machine 2 ne traite aucune tâche de  $E_{p+1}$  pendant l'intervalle de prix  $p$ . Ces résultats sont représentés par les équations (34), (35) et (36).

Toutes les tâches de  $E_{p+1}$  commencent sur la machine 2 pendant l'intervalle de prix  $p+1$ , alors sur la machine 1, nous avons: soit toutes les tâches commencent dans l'intervalle de prix  $p+1$  ( $P_{1\pi^*}^p = 0$ ) soit au moins une tâche commence dans l'intervalle de prix  $p$  ( $P_{1\pi^*}^p = e_p - S_{1\pi^*} - D_{1\pi^*}^p$ ).

Les équations (38) et (39) assurent la charge de traitement de tâches sur la machine 1.

Dans l'équation (37), si  $D_{1\pi^*}^p \neq 0$ , alors au moins une tâche a commencé et termine dans l'intervalle de prix  $p$ . Retarder ces tâches pour que  $D_{1\pi^*}^p = 0$  n'impacte pas ni  $C_{1\pi^*}$  ni  $C_{2\pi^*}$ . Nous posons  $S_{1\pi^*}^* = S_{1\pi^*} + D_{\pi^*1}^p$ .

Alors, nous avons :

$$D_{2\pi^*}^p = 0. \quad (40)$$

$$P_{2\pi^*}^p = 0. \quad (41)$$

$$P_{2\pi^*}^{(p+1)} = \sum_{j \in E_{p+1}} p_{2j}. \quad (42)$$

$$P_{1\pi^*}^p = \max(0, e_p - S_{1\pi^*}^*). \quad (43)$$

$$D_{1\pi^*}^p = A. \quad (44)$$

Où  $A$  est une constante.

$$P_{1\pi^*}^p + P_{1\pi^*}^{(p+1)} = \sum_{j \in E_{p+1}} p_{1j}. \quad (45)$$

$$S_{1\pi^*}^* + D_{1\pi^*}^{(p+1)} + P_{1\pi^*}^p + P_{1\pi^*}^{(p+1)} = C_{1\pi^*}. \quad (46)$$

Soit  $\pi$  la séquence dont l'ordre des tâches suit la règle de Johnson et nous prenons  $S_{1\pi^*}^*$  la date de début sur la machine 1 de cette séquence.

Les équations (40)-(46) sont toujours valables pour cette séquence  $\pi$ . Comme  $\pi$  suit la règle de Johnson, nous avons en plus :

$$C_{1\pi} \leq C_{1\pi^*}. \quad (47)$$

$$C_{2\pi} \leq C_{2\pi^*}. \quad (48)$$

A partir de l'équation (47), en retardant le traitement des tâches sur la machine 1, nous pouvons toujours obtenir  $C_{1\pi} = C_{1\pi^*}$  et  $S_{1\pi}^* = S_{1\pi^*}^*$ .

Alors, nous avons :

$$D_{2\pi}^p = D_{2\pi^*}^p = 0. \quad (49)$$

$$P_{2\pi}^p = P_{2\pi^*}^p = 0. \quad (50)$$

$$P_{2\pi}^{(p+1)} = P_{2\pi^*}^{(p+1)} = \sum_{j \in E_{p+1}} p_{2j}. \quad (51)$$

$$D_{2\pi}^{(p+1)} \leq D_{2\pi^*}^{(p+1)}. \quad (52)$$

$$P_{1\pi}^p = P_{1\pi^*}^p = \max(0, e_p - S_{1\pi^*}^*). \quad (53)$$

$$D_{1\pi}^p = D_{1\pi^*}^p = A. \quad (54)$$

Où A est une constante.

$$P_{1\pi^*}^p + P_{1\pi^*}^{(p+1)} = \sum_{j \in E_{p+1}} p_{1j}. \quad (55)$$

$$S_{1\pi^*}^* + D_{1\pi^*}^{(p+1)} + P_{1\pi^*}^p + P_{1\pi^*}^{(p+1)} = C_{1\pi^*}. \quad (56)$$

Donc avec la séquence de  $\pi$  qui suit la règle de Johnson, nous pouvons obtenir :

$$\begin{aligned} TEC_{\pi} &= \sum_{i \in M} D_{i\pi}^p \times d_i \times fc_p + \sum_{i \in M} P_{i\pi}^p \times b_i \times fc_p \\ &+ \sum_{i \in M} D_{i\pi}^{(p+1)} \times d_i \times fc_{p+1} + \sum_{i \in M} P_{i\pi}^{(p+1)} \times b_i \times fc_{p+1}. \\ &\leq TEC_{\pi^*}. \end{aligned}$$

Nous considérons un exemple illustré dans la Fig.(18). Nous constatons que le temps d'attente et le temps de traitement de jobs répartis dans chaque période de prix  $p$  et  $p+1$  ne changent pas entre  $\pi^*$  et  $\pi$ , donc  $TEC_{\pi} = TEC_{\pi^*}$

### Trois intervalle de prix ( $p-1$ ), $p$ et ( $p+1$ )

Nous étudions dans cette partie une séquence  $\pi^*$  d'un ensemble  $E_p \subset J$  des tâches dont la date de début sur la machine 2 se trouve dans un même intervalle de prix  $p$ .

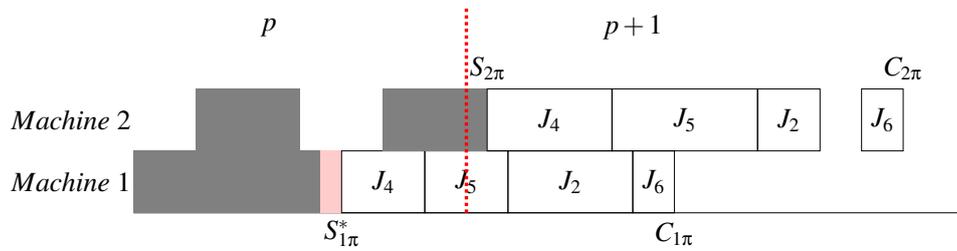
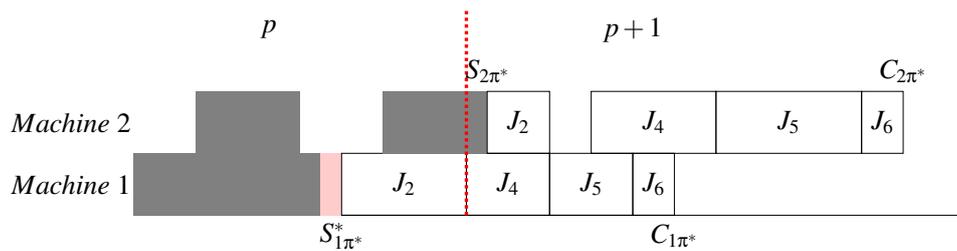
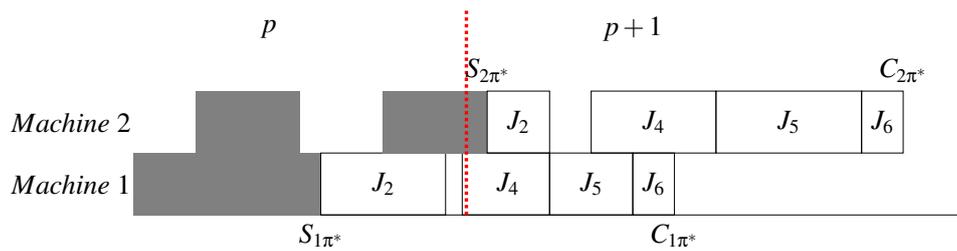


Figure 18 – Comparaison entre la séquence  $\pi^*$  et la séquence de  $\pi$

La première tâche de  $\pi^*$  commence sur la machine 1 à l'intervalle de prix  $(p - 1)$  et la dernière tâche de  $\pi^*$  termine à l'intervalle de prix  $(p + 1)$

Nous montrons que pour une séquence  $\pi^*$  quelconque, la réorganisation des tâches selon la règle de Johnson n'augmente pas le coût d'électricité de la séquence  $\pi^*$ .

En se fondant sur l'étude de deux intervalles de prix  $p$  et  $p + 1$  développé auparavant, nous avons la répartition de temps de traitement et le temps d'idle dans chaque période de prix ci-dessous.

$$D_{2\pi^*}^{(p-1)} = 0. \quad (57)$$

$$D_{2\pi^*}^{(p+1)} = 0. \quad (58)$$

$$P_{2\pi^*}^{(p-1)} = 0. \quad (59)$$

$$P_{2\pi^*}^{(p+1)} = \max(0, C_{2\pi^*} - e_p). \quad (60)$$

$$P_{2\pi^*}^p + P_{2\pi^*}^{(p+1)} = \sum_{j \in E_p} p_{2j}. \quad (61)$$

$$St_p + D_{2\pi^*}^p + P_{2\pi^*}^p + P_{2\pi^*}^{(p+1)} = C_{2\pi^*}. \quad (62)$$

$$D_{1\pi^*}^{(p+1)} = 0. \quad (63)$$

$$D_{1\pi^*}^{(p-1)} = A. \quad (64)$$

Où A est une constante.

$$P_{1\pi^*}^{(p+1)} = 0. \quad (65)$$

$$P_{1\pi^*}^{(p-1)} = \max(0, e_{p-1} - S_{1\pi^*}^*). \quad (66)$$

$$P_{1\pi^*}^{(p-1)} + P_{1\pi^*}^p = \sum_{j \in E_p} p_{1j}. \quad (67)$$

$$S_{1\pi^*} + D_{1\pi^*}^{(p-1)} + D_{1\pi^*}^p + P_{1\pi^*}^{(p-1)} + P_{1\pi^*}^p = C_{1\pi^*}. \quad (68)$$

Nous avons le coût d'électricité total pour la séquence  $\pi^*$  :

$$\begin{aligned} TEC_{\pi^*} &= \sum_{i \in M} D_{i\pi^*}^{(p-1)} \times d_i \times fc_{(p-1)} + \sum_{i \in M} P_{i\pi^*}^{(p-1)} \times b_i \times fc_{(p-1)} \\ &+ \sum_{i \in M} D_{i\pi^*}^p \times d_i \times fc_p + \sum_{i \in M} P_{i\pi^*}^p \times b_i \times fc_p. \\ &+ \sum_{i \in M} D_{i\pi^*}^{(p+1)} \times d_i \times fc_{(p+1)} + \sum_{i \in M} P_{i\pi^*}^{(p+1)} \times b_i \times fc_{(p+1)}. \end{aligned}$$

Soit  $\pi$  une séquence de  $E_p$  dont les tâches suivent la règle de Johnson. Nous prenons  $S_{1\pi^*}^*$  pour la date de début sur la machine 1 de la séquence  $\pi$ .

Les équations (57)-(68) sont toujours valables pour cette séquence  $\pi$ . Comme  $\pi$  suit la règle de Johnson, nous avons également :

$$C_{1\pi} \leq C_{1\pi^*}. \quad (69)$$

$$C_{2\pi} \leq C_{2\pi^*}. \quad (70)$$

D'après les équations (69) and (70), suivant  $\pi$ , nous constatons qu'avec la séquence de  $\pi$ , nous pouvons retarder les tâches pour que  $C_{1\pi} = C_{1\pi^*}$  and  $C_{2\pi} = C_{2\pi^*}$ . Dans ce cas, nous obtenons :

$$D_{2\pi}^{(p-1)} = D_{2\pi^*}^{(p-1)} = 0. \quad (71)$$

$$D_{2\pi}^{(p+1)} = D_{2\pi^*}^{(p+1)} = 0. \quad (72)$$

$$P_{2\pi}^{(p-1)} = P_{2\pi^*}^{(p-1)} = 0. \quad (73)$$

$$P_{2\pi}^{(p+1)} = P_{2\pi^*}^{(p+1)} = \max(0, C_{2\pi^*} - e_p). \quad (74)$$

$$P_{2\pi}^p = P_{2\pi^*}^p = \sum_{j \in S} p_{2j} - \max(0, C_{2\pi^*} - e_p). \quad (75)$$

$$D_{2\pi}^p = D_{2\pi^*}^p = C_{2\pi^*} - P_{2\pi^*}^{(p+1)} - P_{2\pi^*}^p - St_p. \quad (76)$$

$$D_{1\pi}^{(p+1)} = D_{1\pi^*}^{(p+1)} = 0. \quad (77)$$

$$D_{1\pi}^{(p-1)} = D_{1\pi^*}^{(p-1)} = A. \quad (78)$$

Où A est une constante.

$$P_{1\pi}^{(p+1)} = P_{1\pi^*}^{(p+1)} = 0. \quad (79)$$

$$P_{1\pi}^{(p-1)} = P_{1\pi^*}^{(p-1)} = \max(0, e_p - S_{1\pi^*}^*). \quad (80)$$

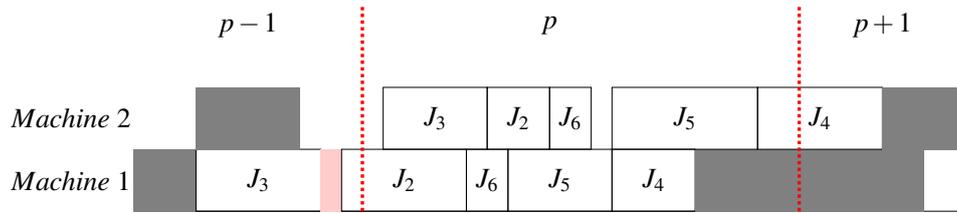
$$P_{1\pi}^p = P_{1\pi^*}^p = \sum_{j \in J} p_{1j} - \max(0, e_p - S_{1\pi^*}^*). \quad (81)$$

$$D_{1\pi}^p = D_{1\pi^*}^p = C_{1\pi^*} - S_{1\pi^*}^* - D_{1\pi^*}^{(p-1)} - P_{1\pi^*}^{(p-1)} - P_{1\pi^*}^p. \quad (82)$$

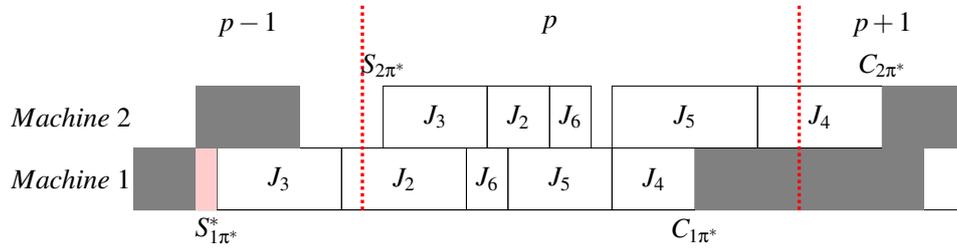
Avec les équations (71)-(82), nous obtenons alors :  $TEC_{\pi} = TEC_{\pi^*}$ .

Alors, pour toutes les séquences  $\pi^*$  dont la date de début des tâches sur la machine 2 se trouvent dans le même intervalle de prix, il existe toujours une séquence  $\pi$  respectant la règle de Johnson et possédant le même coût d'électricité total.

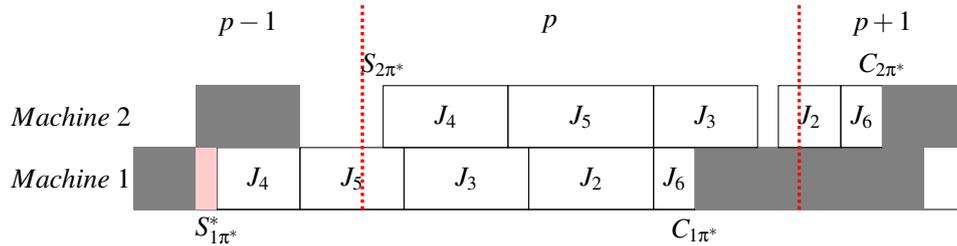
Nous considérons un exemple illustré dans la Fig.(19). Nous constatons que le temps d'attente et le temps de traitement de jobs répartis dans chaque période de prix  $p-1$ ,  $p$  et  $p+1$  ne changent pas entre  $\pi^*$  and  $\pi$ , donc  $TEC_{\pi} = TEC_{\pi^*}$



(a) La séquence  $\pi^*$



(b) La séquence  $\pi^*$  qui retarde la date de début des tâches



(c) La séquence  $\pi$  qui respecte la règle de Johnson et retarde la date de début des tâches

Figure 19 – La comparaison entre  $\pi^*$  et la séquence de  $\pi$

□

La propriété (2) est illustrée par un exemple dans la Fig.(20). Nous avons ici trois intervalles de prix  $(p - 1)$ ,  $p$  et  $(p + 1)$ . Nous avons un ensemble de tâches  $E_p = \{J_1, J_2, J_3, J_4\}$  dont la date de début sur la machine 2 est affectée à l'intervalle de prix  $p$ . Nous avons une séquence  $\pi = J_1 \prec J_2 \prec J_4 \prec J_3$ . Après avoir réorganisé  $\pi$  selon la règle de Johnson, nous obtenons alors :  $\pi' = J_3 \prec J_4 \prec J_1 \prec J_2$ . Comme indiqué par la propriété (2), nous avons :  $TEC_{\pi'} \leq TEC_{\pi}$ .

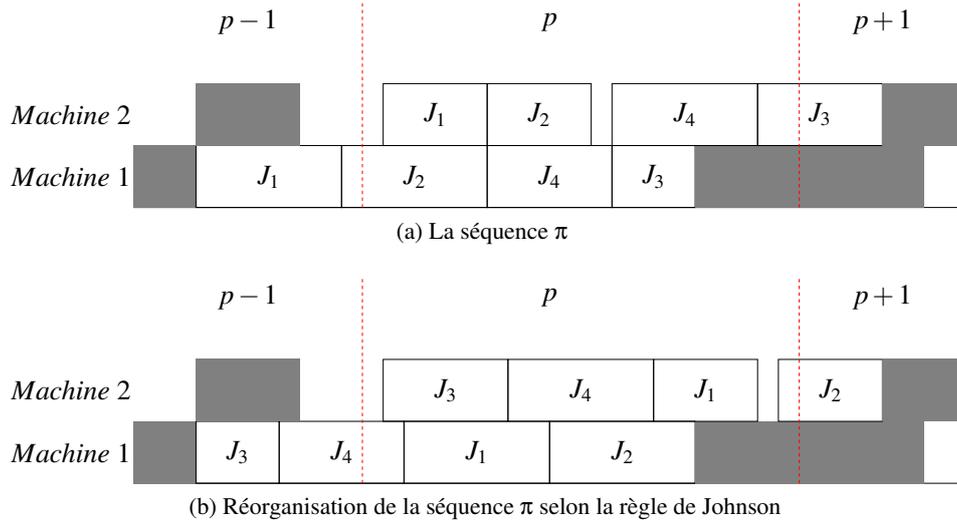


Figure 20 – La séquence  $\pi$  et la réorganisation  $\pi$  selon la règle de Johnson

La propriété (2) détermine la séquence des tâches dans chaque intervalle de prix. Cela signifie que l'objectif de notre problème étudié revient à définir l'ensemble des tâches affectées à chaque intervalle de prix, puis l'ordre des tâches à l'intérieur de chaque ensemble suit la règle de Johnson qui est indiquée par la propriété (2). Dans les sections qui suivent, nous présentons un nouveau modèle mathématique et des approches de résolution qui exploitent la propriété (2) pour améliorer la performance.

## 4.4 Modélisation mathématique

### 4.4.1 Modèle mathématique de [Wang et al. (2018)]

Nous présentons le modèle mathématique proposé par [Wang et al. (2018)]. Ensuite, nous introduisons la fonction objectif et les contraintes qui sont nécessaires pour formuler notre nouveau modèle mathématique.

Le modèle mathématique de [Wang et al. (2018)] est présenté ci-dessous :

La fonction objectif :

$$\min TEC = \sum_{i \in M} \sum_{j \in J} \sum_{t \in T} c_t \times b_i \times z_{ijt} + \sum_{i \in M} \sum_{t \in T} c_t \times d_i \times y_{it}. \quad (83)$$

Sous contraintes :

$$\sum_{j \in J} z_{ijt} \leq 1, \forall i \in M, t \in T. \quad (84)$$

$$\sum_{j \in J} z_{ijt} + y_{it} = g_{it}, \forall i \in M, t \in T. \quad (85)$$

$$\sum_{t \in T} z_{ijt} = p_{ij}, \forall i \in M, j \in J. \quad (86)$$

$$S_{ij} \leq t + V \times (1 - z_{ijt}), \forall i \in M, j \in J, t \in T. \quad (87)$$

$$C_{ij} \geq t + 1 - V \times (1 - z_{ijt}), \forall i \in M, j \in J, t \in T. \quad (88)$$

$$C_{ij} = S_{ij} + p_{ij}, \forall i \in M, j \in J. \quad (89)$$

$$C_{ij} \leq S_{ij'} + V \times (1 - x_{jj'}), \forall i \in M, \forall j, j' \in J : j \neq j'. \quad (90)$$

$$C_{ij'} \leq S_{ij} + V \times x_{jj'}, \forall i \in M, \forall j, j' \in J : j \neq j'. \quad (91)$$

$$x_{jj'} + x_{j'j} = 1, \forall j, j' \in J, j \neq j'. \quad (92)$$

$$C_{1j} \leq S_{2j}, \forall j \in J. \quad (93)$$

$$F_i \geq C_{ij}, \forall i \in M, j \in J. \quad (94)$$

$$F_i \leq t + V \times g_{it}, \forall i \in M, t \in T. \quad (95)$$

$$F_i \geq t + 1 - V \times (1 - g_{it}), \forall i \in M, t \in T. \quad (96)$$

$$F_i \leq T, \forall i \in M. \quad (97)$$

$$g_{it}, y_{it}, z_{ijt}, x_{jj'} \in \{0, 1\}, \forall i \in M, j, k \in J, t \in T. \quad (98)$$

La fonction objectif (83) minimise le coût d'électricité total qui comprend deux types de coût énergétique :

- Lorsque la machine est en marche :  $\sum_{i \in M} \sum_{j \in J} \sum_{t \in T} c_t \times b_i \times z_{ijt}$
- Lorsque la machine est en idle :  $\sum_{i \in M} \sum_{t \in T} c_t \times d_i \times y_{it}$ .

Les contraintes (84) assurent que chaque machine ne peut traiter qu'une tâche à la fois.

Les contraintes (85) définissent l'état de fonctionnement de la machine  $i$  après avoir été démarrée, soit elle est en marche soit elle est en idle.

Les contraintes (86) garantissent que chaque tâche sera entièrement traitée.

Les contraintes (87), (88) et (89) assurent que toutes les tâches seront traitées de manière non préemptive.

Les contraintes (90), (91) et (92) déterminent la relation de précédence entre deux tâches  $j$  et  $j'$ , c'est à dire qu'elles assurent que  $j$  précèdent  $j'$  ou inversement.

Les contraintes (93) assurent que toutes les tâches  $j \in J$  doivent être traitées tout d'abord sur la machine 1 et ensuite sur la machine 2.

Les contraintes (94), (95), (96) et (97) déterminent la date d'arrêt des machines.

Les contraintes (98) donnent la définition des variables.

[Aghelinejad et al. (2019)] ont démontré que le problème d’ordonnement sur une seule machine sous la tarification de *ToU* dont la consommation énergétique est liée aux états de fonctionnement des machines est NP-difficile. Comme un problème d’ordonnement sur une seule machine est un cas particulier du problème d’ordonnement de flow shop, alors notre problème est également NP-difficile.

Nous avons proposé la propriété (2) fondée sur la règle de Johnson pour déterminer la séquence des tâches lorsque la durée de chaque intervalle de prix est relativement longue. Nous proposons ensuite un nouveau modèle mathématique qui intègre la propriété développée pour améliorer l’approche de résolution.

#### 4.4.2 La transformation de la fonction objectif

Comme proposé par [Wang et al. (2018)], la fonction objectif (83) est exprimée en fonction des variables  $z_{ijt}$  et  $y_{it}$ . Nous pouvons trouver la relation entre  $z_{ijt}$  et  $y_{it}$  dans les équations (85).

Nous avons :

$$\begin{aligned} \sum_{j \in J} z_{ijt} + y_{it} &= g_{it} && , \forall i \in M, t \in T. \\ \Leftrightarrow y_{it} &= g_{it} - \sum_{j \in J} z_{ijt} && , \forall i \in M, t \in T. \end{aligned}$$

Alors, nous pouvons réécrire la fonction objectif comme ci-dessous :

$$\begin{aligned} TEC &= \sum_{i \in M} \sum_{j \in J} \sum_{t \in T} c_t \times b_i \times z_{ijt} + \sum_{i \in M} \sum_{t \in T} c_t \times d_i \times (g_{it} - \sum_{j \in J} z_{ijt}). \\ &= \sum_{i \in M} \sum_{j \in J} \sum_{t \in T} c_t \times (b_i - d_i) \times z_{ijt} + \sum_{i \in M} \sum_{t \in T} c_t \times d_i \times g_{it}. \end{aligned}$$

Nous proposons de nouvelles variables  $\alpha_{ijp}$  qui sont la charge de temps de traitement de la tâche  $J_j$  sur la machine  $i$  pendant l’intervalle de prix  $p$  (Fig.(21)). Les paramètres  $fc_p$  sont le prix de l’électricité de l’intervalle de prix  $p$ .

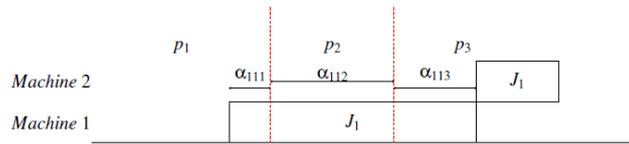


Figure 21 – La présentation de variable  $\alpha_{ijp}$  pour la tâche  $J_1$  sur la machine 1

Nous avons :

$$c_t = fc_p, \forall St_p \leq t \leq e_p$$

$$\alpha_{ijp} = \sum_{St_p \leq t \leq e_p} z_{ijt}, \forall i \in M, j \in J, p \in K.$$

Alors, la fonction objectif de [Wang et al. (2018)] peut être modifiée de la manière suivante :

$$TEC = \sum_{i \in M} \sum_{j \in J} \sum_{p \in K} f c_p \times (b_i - d_i) \times \alpha_{ijp} + \sum_{i \in M} \sum_{t \in T} c_t \times d_i \times g_{it}. \quad (99)$$

A partir de l'équation (99), nous pouvons calculer le coût d'électricité total d'une séquence en fonction de l'affectation de charge de traitement des tâches à chaque intervalle de prix  $p$  ( $\alpha_{ijp}$ ) et de la date d'arrêt de deux machines ( $g_{it}$ ). Lorsque deux solutions ont les mêmes dates d'arrêt de deux machines, elles ont le même sous-ensemble de valeur de  $g_{it}$ . Leurs deuxièmes termes de la fonction de  $TEC$  obtiennent la même valeur. Alors la différence se situe dans l'affectation de charge de traitement des tâches à chaque intervalle de prix ( $\alpha_{ijp}$ ).

#### 4.4.3 Nouveau modèle mathématique (M1)

Nous proposons un nouveau modèle mathématique (M1) qui prend en compte de nouvelles variables et de nouvelles formulations de fonction objectif.

La fonction objectif :

$$\min TEC = \sum_{i \in M} \sum_{j \in J} \sum_{p \in K} f c_p \times (b_i - d_i) \times \alpha_{ijp} + \sum_{i \in M} \sum_{t \in T} c_t \times d_i \times g_{it}. \quad (100)$$

Sous contraintes :

$$\beta_{ijp} \leq 1 - \frac{(S_{ij} - e_p)}{V}, \forall i \in M, j \in J, p \in K. \quad (101)$$

$$\beta_{ijp} \leq 1 + \frac{(C_{ij} + 1 - St_p)}{V}, \forall i \in M, j \in J, p \in K. \quad (102)$$

$$\sum_{p \in K} \alpha_{ijp} = p_{ij}, \forall i \in M, j \in J. \quad (103)$$

$$\alpha_{ijp} \leq (e_p - S_{ij}) + V \times (1 - \beta_{ijp}), \forall i \in M, j \in J, p \in K. \quad (104)$$

$$\alpha_{ijp} \leq (C_{ij} - St_p) + V \times (1 - \beta_{ijp}), \forall i \in M, j \in J, p \in K. \quad (105)$$

$$\alpha_{ijp} \leq (e_p - St_p) \times \beta_{ijp}, \forall i \in M, j \in J, p \in K. \quad (106)$$

$$C_{ij} = S_{ij} + p_{ij}, \forall i \in M, j \in J. \quad (107)$$

$$C_{1j} \leq S_{2j}, \forall j \in J. \quad (108)$$

$$F_i \geq C_{ij}, \forall i \in M, j \in J. \quad (109)$$

$$F_i \leq t + V \times g_{it}, \forall i \in M, t \in T. \quad (110)$$

$$F_i \geq t + 1 - V \times (1 - g_{it}), \forall i \in M, t \in T. \quad (111)$$

$$F_i \leq T, \forall i \in M. \quad (112)$$

$$x_{jj'} + x_{j'j} \leq 1, \forall j, j' \in J : j \neq j'. \quad (113)$$

$$C_{ij} \leq S_{ij'} + V \times (1 - x_{jj'}), \forall i \in M, \forall j, j' \in J : j \neq j'. \quad (114)$$

$$C_{ij'} \leq S_{ij} + V \times x_{jj'}, \forall i \in M, \forall j, j' \in J : j \neq j'. \quad (115)$$

$$C_{ij}, S_{ij}, \alpha_{ijp} \geq 0, \forall i \in M, j \in J, p \in K. \quad (116)$$

$$\beta_{ijp}, g_{ij}, x_{jj'} \in \{0, 1\}, \forall i \in M, j, j' \in J, p \in K. \quad (117)$$

La fonction objectif (100) minimise le coût d'électricité total.

Les contraintes (101) assurent que dans les intervalles de prix  $p$  qui précèdent la date de début de la tâche  $j$  sur la machine  $i$ , aucune partie de la tâche  $j$  n'a été traitée.

Les contraintes (102) assurent que dans les intervalles de prix  $p$  qui succèdent la date de fin de traitement de tâche  $j$  sur la machine  $i$ , aucune partie de la tâche  $j$  n'a été traitée.

Les contraintes (103), (104), (105) et (106) assignent les charges de traitement de la tâche  $j$  sur la machine  $i$  à chaque intervalle de prix  $p \in K$  selon la date de fin et la date de début de traitement de la tâche  $j$ .

Les contraintes (107) assurent que toutes les tâches sont traitées de manière non-préemptive.

Les contraintes (108) assurent que toutes les tâches  $j$  doivent être traitées tout d'abord sur la machine 1 et puis sur la machine 2.

Les contraintes (109), (110), (111) et (112) définissent la date d'arrêt des machines.

Les contraintes (113), (114) et (115) déterminent la séquence.

Les contraintes (116) et (117) donnent la définition des variables.

#### 4.4.4 L'amélioration de modèle mathématique par la règle de Johnson

Pour pouvoir intégrer la propriété (2) dans le modèle mathématique  $M1$ , nous introduisons de nouvelles variables et de nouvelles contraintes. Les nouvelles variables visent à déterminer si les dates de début sur la machine 2 de deux tâches  $j$  et  $j'$  sont dans le même intervalle de prix. Si leurs dates de début sur la machine 2 sont dans le même intervalle de prix, l'ordre entre eux suivra la règle de Johnson comme indiqué dans la propriété (2).

Dans les contraintes données ci-dessous, entre deux tâches  $j$  et  $j'$ , la relation ( $j < j'$ ) signifie que la tâche  $j$  précède la tâche  $j'$  dans la séquence de Johnson. Des nouvelles variables et de nouvelles contraintes sont données comme suit :

Variables:

- $\gamma_{jp} = 1$  si la tâche  $j$  a commencée le traitement sur la machine 2 pendant l'intervalle de prix  $p$ , et  $= 0$  sinon.
- $\omega_{jj'}^p = 1$  si la tâche  $j$  et la tâche  $j'$  ont commencée leurs traitement sur la machine 2 pendant le même intervalle de prix  $p$ , et  $= 0$  sinon.

Sous contraintes :

$$S_{2j} + V \times (1 - \gamma_{jp}) \geq St_p, \forall t \in T, p \in K, j \in J. \quad (118)$$

$$S_{2j} - V \times (1 - \gamma_{jp}) \leq e_p, \forall t \in T, p \in K, j \in J. \quad (119)$$

$$\sum_{p \in K} \gamma_{jp} = 1, \forall j \in J. \quad (120)$$

$$\omega_{jj'}^p \geq (\gamma_{jp} + \gamma_{j'p}) - 1, \forall p \in K, \forall j, j' \in J : j \neq j'. \quad (121)$$

$$\omega_{jj'}^p \leq \frac{\gamma_{jp} + \gamma_{j'p}}{2}, \forall p \in K, \forall j, j' \in J : j \neq j'. \quad (122)$$

$$\delta_{jj'} \geq \sum_{p \in K} \omega_{jj'}^p, \forall j, j' \in J : j < j'. \quad (123)$$

$$\delta_{jj'} \geq \frac{\sum_{p \in K} \gamma_{j'p} \times p - \sum_{p \in K} \gamma_{jp} \times p}{P}, \forall j \in J, j' \in J : j \neq j'. \quad (124)$$

$$\gamma_{jp}, \omega_{jj'}^p \in \{0, 1\}, \forall j, j' \in J, p \in K. \quad (125)$$

Les contraintes (118) assurent que, si la date de début de la tâche  $j$  sur la machine 2 est dans l'intervalle de prix  $p$ , alors sa valeur est supérieure à la date de début de l'intervalle de prix  $p$ .

Les contraintes (119) assurent que, si la date de début de la tâche  $j$  sur la machine 2 est dans l'intervalle de prix  $p$ , alors sa valeur est inférieure à la date de début de l'intervalle de prix  $p$ .

Les contraintes (120) assurent que la date de début sur la machine 2 d'une tâche  $j$  se trouve au plus dans un intervalle de prix.

Les contraintes (121) et (122) déterminent si la date de début sur la machine 2 de tâche  $j$  et de tâche  $j'$  sont dans un même intervalle de prix  $p$ .

Les contraintes (123) assurent que, si la date de début sur la machine 2 de deux tâches  $j$  et  $j'$  sont dans un même intervalle de prix, alors l'ordre entre deux tâches suit la règle de Johnson.

Les contraintes (124) déterminent la séquence des tâches selon l'ordre temporel des intervalles de prix.

Les contraintes (125) donnent la définition des variables.

Nous proposons un nouveau modèle mathématique ( $M2$ ) qui intègre la propriété (2) en ajoutant les contraintes (118), (119), (120), (121), (122), (123), (124) et (125) au modèle mathématique ( $M1$ ).

#### 4.5 La “Logic-Based Benders Decomposition”

Cette section présente l'approche de résolution par la “Logic-Based Benders Decomposition”. Nous présentons le schéma de décomposition de notre problème fondé sur la “Logic-Based Benders Decomposition” ainsi que des éléments principaux de l'approche : le problème maître (MP), le problème esclave (SP) et la coupe de Benders.

### 4.5.1 Introduction

La décomposition de Benders utilise une stratégie de résolution de problèmes qui peut traiter des modèles de programmation linéaires mixtes à grande échelle possédant à la fois des variables entières et continues [Benders (1962)]. La stratégie de résolution fondée sur le partitionnement du problème en deux vecteurs de variables :  $x$  et  $y$ . En générale, nous pouvons modéliser le problème sous la forme :

$$\begin{aligned} & \text{Min } f(x, y). \\ & \text{Sous contraintes : } (x, y) \in S, \\ & \quad x \in D_x, y \in D_y. \end{aligned}$$

Dont :

- $f(x, y)$  : la fonction objectif.
- $S$  : l'ensemble des solutions faisables (défini par l'ensemble des contraintes).
- $D_x, D_y$  : les domaines de valeur de  $x$  et  $y$  respectivement.

Premièrement, un problème maître (MP) est utilisé pour traiter un sous-ensemble de variables  $y$ . Le MP est une relaxation du modèle global. Ensuite, les valeurs des variables  $y$  obtenues sont ensuite utilisées pour définir un problème esclave (SP) qui ne contient que  $x$  comme variables. La solution du SP peut révéler que les valeurs de  $y$  sont irréalisables, alors la solution sera utilisée pour générer une coupe de Benders. La coupe de Benders sert à éliminer les valeurs irréalisables de  $y$  de l'ensemble de solutions. Ensuite, nous obtenons le prochain ensemble de valeurs d'essai de  $y$  en résolvant le MP contenant toutes les coupes de Benders générées jusqu'à présent. Nous résolvons de manière itérative le MP à l'optimalité et utilisons la solution de MP pour générer des problèmes esclaves (SP) jusqu'à ce que le MP et le SP convergent vers la solution optimale du modèle global.

Le problème  $F2|Pu_{ij} = Pu_{ij}', Pu_{idle}, ToU|TEC$  nécessite deux décisions différentes : la séquence optimale des tâches et la détermination des dates de début des tâches dans la séquence. Alors, une approche de décomposition peut être bien adaptée. Cependant, la décomposition classique de Benders nécessite que le MP soit un modèle de variables entières mixtes et que les SP soient des problèmes de programmation linéaire ou non linéaire. Nous utilisons la "Logic-Based Benders Decomposition" (LBBD) [Hooker and Ottosson (2003)] qui exclut cette nécessité de la modélisation de MP et des SP.

Dans cette étude, le modèle de MP en nombres entiers mixtes affecte des tâches à des intervalles de prix, puis en s'appuyant sur la propriété (2), nous pouvons obtenir une séquence globale des tâches. Ensuite, le SP utilise la programmation dynamique [Wang et al. (2018)] pour déterminer les dates de début optimales des tâches de la séquence donnée par le MP. Nous présentons des détails du problème MP et SP, ainsi la coupe de Benders dans les sections suivantes.

#### 4.5.2 Le problème maître d'affectation des tâches aux intervalles de prix

Comme nous l'avons présenté dans la section précédente, nous pouvons calculer le coût d'électricité total en fonction de la date d'arrêt de chaque machine et de la charge de traitement des tâches affectée à chaque intervalle de prix. Le MP détermine la charge de traitement des tâches à chaque intervalle de prix en affectant la date de début sur la machine 2 des tâches aux intervalles de prix. Selon l'affectation des tâches, MP peut estimer les charges de traitement dans chaque intervalle de prix et les dates d'arrêt approximatives de chaque machine. Avant de formuler le MP en tant que modèle de programmation en nombres entiers mixtes, nous représentons des résultats intéressants qui peuvent améliorer les performances du modèle.

Chaque intervalle de prix a une certaine durée et peut contenir au plus une certaine charge de traitement des tâches. Soit  $Cap_p^i$  le nombre maximal de tâches pouvant être traitées sur la machine  $i$  pendant l'intervalle de prix  $p$ . Sur chaque machine  $i$ , nous ordonnons les tâches selon la règle de temps de traitement le plus court (SPT). Pour chaque tâche selon l'ordre de SPT,  $Cap_p^i$  est incrémenté d'une unité lorsque la charge de traitement total sur la machine  $i$  n'est pas supérieure à la durée de l'intervalle de prix  $p$ . En considérant séparément la machine 1 et la machine 2, nous pouvons obtenir le nombre maximal de tâches affectées à chaque intervalle de prix sur chaque machine.

La règle de Johnson donne une solution optimale pour  $F2||C_{max}$  en temps polynomial. Soit  $\pi$  la séquence de  $F2||C_{max}$ , pour n'importe quelle séquence, leur dates d'achèvement sur la machine 1 et sur la machine 2 sont supérieures à celles de  $\pi$ . Pour formuler le MP, nous avons besoin de nouvelles variables et paramètres :

Les paramètres :

- $Cap_p^i$  : le nombre maximal de tâches pouvant être traitées sur la machine  $i$  pendant l'intervalle de prix  $p$  selon SPT.
- $F_i$  : la date d'arrêt de la machine  $i$  selon la solution de Johnson pour le problème  $F2||C_{max}$ .

Les variables :

- $\gamma_{jp}^i = 1$  si la tâche  $j$  a commencé sur la machine  $i$  pendant l'intervalle de prix  $p$ , et 0 sinon.
- $TEC_1^h$  le coût d'électricité total de la solution de MP à  $h^{eme}$  itération.

Le modèle mathématique de MP est présenté ci-dessous :

La fonction objectif :

$$\min TEC_1^h = \sum_{i \in M} \sum_{j \in J} \sum_{p \in K} f c_p \times (b_i - d_i) \times \alpha_{ijp} + \sum_{i \in M} \sum_{t \in T} c_t \times d_i \times g_{it}. \quad (126)$$

Sous contraintes :

$$\sum_{p \in K} \gamma_{jp}^i = 1, \forall i \in M, \forall j \in J. \quad (127)$$

$$\sum_{j \in J} \sum_{p \in K} \gamma_{jp}^i \leq \text{Cap}_p^i, \forall i \in M. \quad (128)$$

$$\beta_{ijp} \leq 1 - \frac{\sum_{p' \in K} \gamma_{jp'}^i \times p' - p}{K}, \forall i \in M, \forall j \in J, \forall p \in K. \quad (129)$$

$$\beta_{ijp} \leq 1 + \frac{\sum_{p' \in K} \gamma_{jp'}^i \times e_{p'} + p_{ij} - St_p}{T}, \forall i \in M, j \in J, p \in K. \quad (130)$$

$$\beta_{1jp} \times p \leq \sum_{p' \in K} \gamma_{jp'}^2 \times p', \forall j \in J, p \in K. \quad (131)$$

$$\sum_{p \in K} \alpha_{ijp} = p_{ij}, \forall i \in M, j \in J. \quad (132)$$

$$\sum_{j \in J} \alpha_{ijp} \leq e_p - St_p, \forall i \in M, p \in K. \quad (133)$$

$$\alpha_{ijp} \leq (e_p - St_p) \times \beta_{ijp}, \forall i \in M, j \in J, p \in K. \quad (134)$$

$$\gamma_{jp}^i \leq \alpha_{ijp}, \forall i \in M, j \in J, p \in K. \quad (135)$$

$$F_i \geq \beta_{ijp} \times St_p + \sum_{j' \in J} \alpha_{ij'p}, \forall i \in M, j \in J, p \in K. \quad (136)$$

$$F_i \leq t + T \times g_{it}, \forall i \in M, t \in T. \quad (137)$$

$$F_i \geq t + 1 - T \times (1 - g_{it}), \forall i \in M, t \in T. \quad (138)$$

$$F_i \leq T, \forall i \in M. \quad (139)$$

$$F_1 \leq F_2. \quad (140)$$

$$F_i \geq F_i, \forall i \in M. \quad (141)$$

$$\text{MIP cuts.} \quad (142)$$

La fonction objectif (126) minimise le coût d'électricité total.

Les contraintes (127) assurent que la date de début de la tâche  $j$  sur la machine  $i$  se trouve au plus dans un intervalle de prix  $p$ .

Les contraintes (128) assurent que le nombre de tâche  $j$  dont la date de début sur la machine  $i$ , dans l'intervalle de prix  $p$  respecte la capacité de cet intervalle.

Les contraintes (129) assurent que pour n'importe quelle tâche  $j$ , si sa date de début sur la machine  $i$  se trouve dans l'intervalle de prix  $p'$  alors cette tâche ne

peut pas être traitée pendant n'importe quel intervalle  $p$  qui se situe temporellement avant  $p'$ ,  $\beta_{ijp} = 0$ .

Les contraintes (130) déterminent pour n'importe quelle tâche  $j$ , des intervalles de prix dans lequel elle est traitée en fonction de sa date de début et de son temps de traitement.

Les contraintes (131) assurent que pour n'importe quelle tâche  $j$ , sa date de début sur la machine 2 se situe temporellement après sa date de début sur la machine 1.

Les contraintes (132) assurent que toutes les tâches sont traitées de manière non préemptive.

Les contraintes (133) assurent que la charge totale de traitement des tâches dans chaque intervalle de prix respecte la durée de cet intervalle.

Les contraintes (134) assurent que pour toutes les tâches, elles ne peuvent être traitées que dans les intervalles de prix où elles sont affectées.

Les contraintes (135) assurent que pour n'importe quelle tâche  $j$ , si sa date de début est assignée à l'intervalle de prix  $p$ , alors dans cet intervalle de prix, au moins une partie de cette tâche doit être traitée.

Les contraintes (136) assurent que la date d'arrêt de la machine  $i$  doit être supérieure à toutes les valeurs de date de début de l'intervalle de prix utilisé additionné de la durée total de traitement de tâches  $y$  étant affectées.

Les contraintes (137), (138) déterminent l'état de fonctionnement de la machine selon la date d'arrêt de la machine.

Les contraintes (139) assurent que toutes les tâches doivent être traitées pendant l'horizon de travail.

Les contraintes (140) assurent que la machine 1 sera arrêtée avant la machine 2.

Les contraintes (141) assurent que la date d'arrêt de chaque machine doit être égale ou supérieure à celle de la solution de  $F2||C_{max}$ .

Les contraintes (142) sont les coupes de Benders.

### 4.5.3 Le problème esclave de date de début optimale et la coupe de Benders

La solution de MP donne un ensemble de tâches dont les dates de début sur la machine 2 sont affectées à chaque intervalle de prix. Soit  $W^h$  l'ensemble des solutions à l'itération  $h$  du MP,  $\gamma_{jp}^2 = 1, \forall j \in W^h$ . Grâce à la propriété (2), en connaissant l'affectation des tâches, nous pouvons obtenir une séquence globale.

Le SP détermine la date de début optimale des tâches dans la séquence donnée par MP. Le travail [Wang et al. (2018)] propose une programmation dynamique qui, lorsque la séquence des tâches est prédéterminée, peut déterminer la date de début optimale de chaque tâche sur chaque machine en  $O(nT^4)$  étapes. Nous utilisons donc la DP proposée par [Wang et al. (2018)] pour résoudre les SP afin d'obtenir des solutions avec un coût total d'électricité  $TEC_2^h$ . La solution obtenue de SP à chaque itération  $h$  permet de générer des coupes de Benders, qui seront ensuite

ajoutées au MP en prochaine itération. A une itération donnée, une coupe de Benders est dite valable si elle permet d'exclure l'affectation irréalizable des tâches obtenue par MP sans éliminer la solution optimale de l'espace de solution. Nous proposons une coupe de Benders comme suit :

$$TEC \geq TEC_2^h - \sum_{\gamma_{jp}^2 \in W^h} (1 - \gamma_{jp}^2) \times V \quad (143)$$

Dont :  $W^h$  est l'ensemble de  $\gamma_{jp}^2 = 1$  : l'affectation de date de début sur la machine 2 de tâche  $j$  à l'intervalle de prix  $p$ .

La procédure de notre approche de solution fondée sur "Logic-Based Benders Decomposition" est illustrée dans la Fig.(22).



Figure 22 – La "Logic-based Benders Decomposition" pour le problème étudié  $F2|Pu_{ij} = Pu_{ij'}, Pu_{idle}, ToU|TEC$ .

## 4.6 Les tests numériques

Dans cette section, nous menons des tests numériques pour évaluer la performance de nos approches de résolution. Nous comparons la performance de nos modèles MILP proposés  $M1$ ,  $M2$ , la “Logic-Based Benders Decomposition” ( $LBB$ ) avec le modèle MILP ( $MC$ ) proposé par [Wang et al. (2018)].

Les modèles mathématiques  $M1$ ,  $M2$ ,  $MC$  et  $LBB$  sont implémentés sur g++ (Ubuntu 7.4.0-lubuntu 18.04.1) 7.4.0 couplé avec Cplex Studio IDE 12.9.0. Nous menons des tests numériques sur un ordinateur “IntelCore i7-6820HQ @ 2,7 GHz 2,7 GHz” avec 8 GB de mémoire et le système d’exploitation Ubuntu 18.04.2 LTS.

### 4.6.1 La génération des données

Nous évaluons la performance des approches de résolution proposées par un ensemble d’instances de tests numériques. Les instances sont générées de manière aléatoire dont les paramètres reposent sur l’étude de [Wang et al. (2018)]. Tout d’abord, nous générons aléatoirement le temps de traitement des tâches  $p_{ij}$  à partir de la distribution uniforme sur  $\{1, 2, \dots, 10\}$ . Ensuite, pour compléter l’ensemble d’instances, nous générons les autres paramètres suivants :

- L’horizon de travail est calculé par :  $T = \lceil 0.8 \times \sum_{i=1}^2 \sum_{j \in J} p_{ij} \rceil$ .
- Pour représenter la fluctuation de prix d’électricité sous  $ToU$ , nous divisons l’horizon de travail en 6 intervalles de prix : “off-peak”, “peak”, “semi-peak”, “peak”, “off-peak” et “semi-peak”. La durée de chaque intervalle de prix est approximativement égale à  $T/6$ . Pour représenter les marchés de l’électricité stables et instables, le niveau  $\max\{c_t\}/\min\{c_t\} = 3$  et le niveau  $\max\{c_t\}/\min\{c_t\} = 6$  sont employés respectivement.
- Pour les taux de consommation d’énergie, nous considérons trois scénarios de taux de consommation d’énergie des machines :
  - $R1$  :  $b_1 = 2, d_1 = 1, b_2 = 2, d_2 = 1$ , deux machines sont identiques;
  - $R2$  :  $b_1 = 2, d_1 = 1, b_2 = 6, d_2 = 2$ , la machine 2 a une puissance énergétique plus élevée;
  - $R3$  :  $b_1 = 6, d_1 = 2, b_2 = 2, d_2 = 1$ , la machine 1 a une puissance énergétique plus élevée;

Nous générons 30 instances pour chaque scénario  $R1, R2$  et  $R3$ .

Nous testons d’abord sur des instances de petites tailles,  $n = 6, 8$  tâches, puis sur des instances de grandes tailles  $n = 20, 30$  tâches. Le temps de calcul pour chaque instance est limité à 1800 secondes.

#### 4.6.2 Les résultats numériques

Afin de tester les performances de nos approches de résolution, nous comparons notre MILP ( $M1$  et  $M2$ ), LBBDD avec le modèle mathématique de [Wang et al. (2018)] ( $MC$ ). Les résultats sont donnés dans les tableaux (23), (24), (25) et (26). Pour les instances de petites tailles ( $n = 6, 8$  tâches), toutes les approches fournissent le même  $TEC$  dans la limite de temps de calcul. Donc, pour ces instances, nous ne nous intéressons qu'à la comparaison du temps de calcul. En revanche, pour les instances de grandes tailles ( $n = 20, 30$  tâches), toutes les approches ne peuvent pas fournir la solution optimale dans la limite de temps de calcul. Par conséquent, nous présentons uniquement la comparaison entre toutes les approches de résolutions avec la "meilleure solution trouvée" (BKS). Le  $Gap_i$  représente l'écart entre la solution de l'approche de résolution  $i$  avec la "meilleure solution trouvée":

$$Gap_i = \frac{TEC_i - TEC^{BKS}}{TEC^{BKS}}.$$

Les tableaux (23) - (25) résument les résultats des tests numériques. Dans chaque tableau, "Max", "Moyenne" et "Min" représentent respectivement la valeur maximale, la valeur moyenne et la valeur minimale de l'ensemble des solutions. Le tableau (23) représente les résultats de l'ensemble des instances pour chaque configuration  $n = 6, 8, 20, 30$  tâches. Cela donne un bilan des performances des approches de résolution pour tous les scénarios. Ensuite, le tableau (24) et le tableau (25) donnent plus de détails sur les tests numériques pour chaque scénario. Ces deux tableaux fournissent le comportement d'approches de résolution lié à la stabilité ou l'instabilité du marché de l'électricité ou aux différentes intensités énergétiques des machines. Dans chaque tableau, "Max", "Moyenne", "Min" et " $\sigma$ " représentent respectivement la valeur maximale, la valeur moyenne, la valeur minimale et l'écart type de l'ensemble de la solution.

Tout d'abord, nous nous concentrons sur les instances de petite taille,  $n = 6, 8$  tâches, pour évaluer les performances de nos MILP  $M1$ ,  $M2$  et LBBDD. A partir de tableau (23a) et tableau (24), nous obtenons les observations suivantes :

- Le tableau (23a) montre que le temps de calcul moyen de  $M1$ ,  $M2$  et LBBDD sont approximativement égaux à  $MC$ .  $MC$  semble être légèrement plus rapide que  $M1$  et  $M2$  pour  $n = 8$  tâches mais la différence n'est pas significative.
- Dans le tableau (24),  $MC$  fonctionne relativement mieux dans le scénario  $R1$  où des machines sont identiques. De l'autre côté,  $M1$  et  $M2$  fonctionnent mieux sur le scénario  $R3$  où la puissance énergétique de la machine 2 est plus élevée.

Ensuite, pour les instances de grandes tailles, nous fournissons dans le tableau (26) l'écart à l'optimalité de LBBDD donné par CPLEX. A partir des tableaux (23b), (25) et (26) nous avons les observations suivantes :

- LBBDD surpasse toutes les autres approches de résolution en termes de  $TEC$ . LBBDD fournit la meilleure solution pour toutes les instances. De plus, dans le tableau (26), nous pouvons voir que l'écart d'optimalité maximal donné

par *CPLEX* pour *LBB*D est inférieur à 6.2% pour  $n = 20$  tâches et inférieur à 3.3% pour  $n = 30$  tâches dans la limite de temps de calcul. Son écart à l'optimalité moyen est inférieur à 3% .

- Les performances de *MC* se dégradent rapidement lorsque  $n$  augmente. Dans certains cas, *MC* obtient un écart avec la “meilleure solution trouvée” jusqu’à 15.8% pour  $n = 20$  tâches et 60.76% pour  $n = 30$  tâches. Pour  $n = 20$  tâches, les performances de *M1* et *M2* sont approximativement égales. Cependant, la performance de *M1* se détériore plus rapidement lorsque  $n = 30$  tâches. *M1* obtient un écart maximum avec la “meilleure solution trouvée” jusqu’à 18.02%, tandis que *M2* n’obtient qu’un écart de 12.79%.

Tableau 23 – Le bilan des tests numériques

(a) Le temps de calcul moyen (secondes) pour les instances de petite taille

Nombre de tâches		<i>LBB</i> D	<i>M1</i>	<i>M2</i>	<i>MC</i>
6	Max	188.0	<b>33.0</b>	37.0	52.0
	Moyenne	14.3	<b>3.8</b>	4.6	7.1
	Min	0.0	<b>1.0</b>	1.0	1.0
8	Max	1800.0	1800.0	1800.0	<b>1800.0</b>
	Moyenne	167.2	342.2	362.1	<b>163.8</b>
	Min	0.0	13.0	15.0	<b>4.0</b>

(b) L'écart moyen (%) avec la “meilleure solution trouvée” pour les instances de grande taille

Nombre de tâches		<i>LBB</i> D	<i>M1</i>	<i>M2</i>	<i>MC</i>
20	Max	<b>0.00%</b>	3.66%	4.68%	15.80%
	Moyenne	<b>0.00%</b>	0.26%	0.55%	4.26%
	Min	<b>0.00%</b>	0.00%	0.00%	0.00%
30	Max	<b>0.00%</b>	18.02%	12.79%	60.76%
	Moyenne	<b>0.00%</b>	0.87%	1.07%	23.53%
	Min	<b>0.00%</b>	0.00%	0.00%	0.22%

## 4.7 Conclusion

Dans ce chapitre, nous avons étudié le problème d’ordonnancement de flow shop à deux machines sous *ToU* avec la minimisation de coût d’électricité total dont la consommation énergétique est liée aux états de fonctionnement des machines ( $F2|Pu_{ij} = Pu_{ij}', Pu_{idle}, ToU|TEC$ ). Premièrement, pour améliorer l’approche de résolution, nous avons proposé une propriété fondée sur la règle de Johnson. La propriété détermine la séquence optimale pour chaque affectation de tâches aux intervalles de prix donnée. Deuxièmement, nous avons développé un nouveau modèle linéaire en nombres entiers mixtes qui intègre la propriété développée.

Tableau 24 – La comparaison en terme de temps de calcul entre les approches de résolution pour les instances de 6, 8 tâches.

		Nombre de tâches															
		6 tâches						8 tâches									
Niveau de prix	Scénario	LBBD	MI	M2	MC	Niveau de prix	Scénario	LBBD	MI	M2	MC	Niveau de prix	Scénario	LBBD	MI	M2	MC
3	1	Max 13.5 Moyenne 1.0 Min 1.0 σ 36.5	33.0 6.3 1.0 7.6	37.0 7.0 1.0 8.9	9.0 3.3 1.0 2.0	3	1	Max 1800.0 Moyenne 128.9 Min 0.0 σ 348.1	1800.0 568.4 32.0 686.2	1800.0 600.0 22.0 690.1	1800.0 25.0 4.0 39.6	3	1	Max 1800.0 Moyenne 93.9 Min 1.0 σ 328.7	1800.0 308.5 41.0 319.2	1800.0 317.8 43.0 318.8	1800.0 46.0 7.0 70.5
	2	Max 185.0 Moyenne 14.2 Min 1.0 σ 38.3	9.0 3.3 1.0 1.8	10.0 3.6 1.0 2.1	14.0 5.3 1.0 3.6		2	Max 1800.0 Moyenne 93.9 Min 1.0 σ 328.7	1800.0 308.5 41.0 319.2	1800.0 317.8 43.0 318.8	1800.0 46.0 7.0 70.5						
	3	Max 56.0 Moyenne 7.0 Min 0.0 σ 11.8	13.0 4.8 1.0 3.0	17.0 5.6 2.0 4.1	27.0 8.2 1.0 6.8		3	Max 1800.0 Moyenne 101.8 Min 1.0 σ 174.5	1800.0 407.3 58.0 442.2	1800.0 443.9 81.0 424.6	1800.0 217.8 6.0 361.0						
6	1	Max 188.0 Moyenne 25.3 Min 1.0 σ 46.4	10.0 3.8 1.0 2.7	12.0 5.2 1.0 3.4	50.0 7.9 1.0 9.8	6	1	Max 1800.0 Moyenne 248.4 Min 1.0 σ 405.2	1800.0 405.7 40.0 432.2	1800.0 425.4 44.0 411.3	1800.0 197.1 5.0 295.1	6	1	Max 1800.0 Moyenne 305.4 Min 2.0 σ 574.3	1800.0 193.3 21.0 172.6	1800.0 221.7 31.0 199.6	1800.0 153.0 6.0 229.4
	2	Max 144.0 Moyenne 19.5 Min 1.0 σ 36.5	5.0 2.4 1.0 1.0	6.0 3.0 1.0 1.2	19.0 6.8 1.0 4.4		2	Max 1800.0 Moyenne 1058.0 Min 5.0 σ 338.0	1800.0 1058.0 170.1 13.0	1800.0 1072.0 221.7 199.6	1800.0 940.0 197.1 295.1						
	3	Max 36.0 Moyenne 6.3 Min 1.0 σ 8.3	7.0 2.5 1.0 1.5	9.0 3.3 1.0 2.0	52.0 10.9 1.0 9.8		3	Max 1800.0 Moyenne 124.4 Min 5.0 σ 338.0	1800.0 124.4 170.1 13.0	1800.0 164.0 15.0 183.9	1800.0 1800.0 333.9 9.0 383.8						

Tableau 25 – La comparaison en terme de l'écart avec la “meilleure solution trouvée” entre les approches de résolution pour les instances de 20, 30 tâches.

Nombre de tâches		30 tâches									
		Niveau de prix	Scénario	Max	Moyenne	Min	$\sigma$	LBBD	M1	M2	MC
20 tâches	3	1	Max	4.94%	0.87%	1.59%	0.00%	0.00%	14.41%	12.79%	38.89%
			Moyenne	2.11%	0.11%	0.28%	0.00%	1.29%	0.00%	1.50%	8.23%
			Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.49%
	2	2	Max	10.84%	1.97%	2.83%	0.00%	0.00%	5.63%	6.09%	47.18%
			Moyenne	3.96%	0.19%	0.61%	0.00%	1.12%	0.00%	1.45%	18.95%
			Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	3.58%
	3	3	Max	3.00%	0.41%	0.73%	0.00%	1.41%	1.66%	8.53%	
			Moyenne	5.67%	1.86%	1.78%	0.00%	3.49%	6.56%	45.59%	
			Min	2.67%	0.32%	0.41%	0.00%	0.31%	1.13%	14.69%	
30 tâches	6	1	Max	12.42%	1.20%	3.19%	0.00%	18.02%	4.45%	41.78%	
			Moyenne	5.07%	0.16%	0.75%	0.00%	1.01%	0.68%	25.35%	
			Min	0.18%	0.00%	0.00%	0.00%	0.00%	0.00%	2.36%	
	2	2	Max	14.58%	3.66%	4.12%	0.00%	5.06%	5.37%	60.76%	
			Moyenne	6.72%	0.53%	0.69%	0.00%	0.67%	1.14%	39.11%	
			Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	6.48%	
	3	3	Max	3.99%	0.84%	1.03%	0.00%	1.15%	1.68%	16.14%	
			Moyenne	15.80%	1.49%	4.68%	0.00%	15.63%	2.87%	57.15%	
			Min	0.64%	0.00%	0.55%	0.00%	0.81%	0.53%	34.88%	
3	3	Max	3.71%	0.40%	1.03%	0.00%	2.84%	0.66%	16.61%		
		Moyenne	5.02%	0.24%	0.55%	0.00%	0.00%	0.00%	5.13%		
		Min	0.64%	0.00%	0.00%	0.00%	0.00%	0.00%	5.13%		

Tableau 26 – L'écart d'optimalité donné par *CPLEX* de *LBB*D pour les instances de 20, 30 tâches

Niveau de prix	Scénario	20 tâches			30 tâches		
		L'écart d'optimalité			L'écart d'optimalité		
		Max	Moyenne	Min	Max	Moyenne	Min
3	1	6.20%	2.80%	0.10%	2.70%	0.70%	0.10%
	2	4.40%	2.40%	0.30%	3.30%	1.50%	0.10%
	3	3.40%	1.30%	0.10%	2.10%	0.70%	0.10%
6	1	4.00%	1.80%	0.00%	2.20%	0.90%	0.10%
	2	4.10%	2.20%	0.20%	2.20%	0.90%	0.10%
	3	3.20%	1.30%	0.10%	2.10%	0.40%	0.10%

Troisièmement, nous avons proposé une méthode exacte fondée sur “Logic-Based Benders Decomposition” pour résoudre le problème. Enfin, nous avons mené des tests numériques pour comparer les performances de nos approches de résolution proposées avec le modèle mathématique de [Wang et al. (2018)].

Nos nouveaux modèles linéaires en nombres entiers mixtes *M1* et *M2* donnent la meilleure performance que le modèle mathématique de [Wang et al. (2018)] pour les instances de grande taille. Les nouveaux modèles *M1* et *M2* fonctionnent mieux pour le marché de l'électricité instable et lorsque la puissance énergétique de la machine 2 est plus élevée. De plus, nous avons proposé une méthode exacte fondée sur “Logic-Based Benders Decomposition” pour résoudre le problème. Dans la limite de temps de 1800 secondes, *LBB*D fournit la meilleure solution par rapport aux approches de résolution dans la littérature pour presque toutes les instances données. Pour les instances de grandes tailles, *LBB*D donne des solutions avec un écart moyen à la borne inférieure sous 2,8%.

Nous avons également constaté qu'en modifiant les variables et en reformulant la fonction objectif, nous avons considérablement amélioré les performances des modèles linéaires en nombres entiers mixtes. Pour les instances de grande taille, nous avons réduit l'écart maximal avec la “meilleure solution trouvée” (BKS) de 60,76% de *MC* à 18,02% de *M1*. De plus, en intégrant la propriété (2) dans *M1*, nous avons encore réduit l'écart avec “BKS” de 18,02% à 12,79%. Les managers peuvent utiliser nos approches proposées *LBB*D ou le modèle mathématique *M2* pour améliorer la qualité de la solution.

Pour les travaux à venir, il pourrait être intéressant de concevoir et d'analyser une borne inférieure plus performante afin d'évaluer ou d'améliorer les approches de résolution. Pour mieux exploiter la propriété développée, il est intéressant de construire une heuristique ou une méta-heuristique intégrant la propriété pour réduire le temps de calcul. On peut aussi considérer le cas de plusieurs états de machine, plusieurs vitesses de fonctionnement qui consomment différentes quantités d'énergie pour étendre notre modèle mathématique. Un problème d'optimisation multi-objectif permettrait de trouver le compromis entre la productivité et le coût de l'électricité.

## 5 Conclusion et perspectives

Au terme de cette étude, nous avons vu que l'augmentation prévue de la consommation énergétique et le réchauffement climatique incitent à ajuster notre comportement énergétique. Le secteur manufacturier joue un rôle important car il représente l'une des parts les plus importantes de la consommation totale d'énergie et de source d'émissions de  $CO_2$ . Avec l'émergence du Smart Grid et la maîtrise de la demande en énergie pendant ces dernières années, l'amélioration de l'efficacité énergétique et la réduction de coût énergétique devient une tendance incontournable pour des entreprises de fabrication de nos jours.

L'objectif de cette thèse est la modélisation et la résolution des problèmes d'ordonnancement de flow shop à deux machines prenant en compte les aspects énergétiques dans le but de minimiser les coûts de consommation d'énergie. L'atelier de flow shop est mis sous la tarification d'électricité de type "Time of Use". Le prix d'électricité dans ce tarif varie d'une période à une autre, cela apportera des gains financiers significatifs si le plan de production est bien adapté.

Le chapitre 2 est consacré à l'état de l'art qui aborde les différentes techniques d'optimisation énergétique dans le problème d'ordonnancement de systèmes de production. Ces techniques sont fondées sur deux axes principaux : l'efficacité énergétique et la réponse à la demande. L'efficacité énergétique vise à minimiser la consommation énergétique totale du système en se reposant sur ses caractéristiques énergétiques. La réponse à la demande vise à minimiser le coût énergétique du système en adaptant le plan de production à la fluctuation de prix du marché. Ce chapitre nous informe que les études d'ordonnancement de flow shop sous la tarification d'électricité de type "Time of Use" de la réponse à la demande sont plutôt clairsemées dans la littérature. Notre thèse permet d'enrichir la littérature et de fournir des outils d'aide à la décision qui s'adaptent mieux le plan de production du marché.

Le chapitre 3 porte sur l'optimisation conjointe du makespan et du coût de l'électricité dans le problème d'ordonnancement de flow shop à deux machines sous la tarification d'électricité "Time of Use". La consommation énergétique dépend des caractéristiques des tâches, en d'autres termes, sur une machine, la puissance énergétique consommée pour traiter une tâche est différente d'une tâche à autre. Dans cette étude, nous prenons la valeur optimale de makespan du problème ( $F2|perm|C_{max}$ ) pour une contrainte à respecter. Toutes les solutions faisables doivent avoir un makespan minimum. L'approche de résolution se compose de deux phases. La première phase est la recherche des solutions optimales équivalentes. Ensuite, la deuxième phase consiste à déterminer la date de début optimale des tâches dans chaque solution trouvée lors de première phase. Enfin, nous prenons le plan de production dont le coût d'électricité est minimum pour notre solution.

Premièrement, nous étudions la théorie de solutions optimales équivalentes qui permet de générer un ensemble de plusieurs solutions optimales de ( $F2|perm|C_{max}$ ). Nous développons des propositions pour étendre l'ensemble des solutions opti-

males de la littérature. Cette étude sur l’extension de l’ensemble des solutions équivalente permet non seulement de résoudre notre problématique mais également d’enrichir l’approche d’ordonnancement robuste. Ça permet au manager de basculer d’une solution à l’autre dans le cas de perturbations car toutes les solutions ont la même valeur de makespan. Deuxièmement, nous avons prouvé que sous le makespan optimal et la séquence prédéterminée, le problème d’ordonnancement écoénergétique de flow shop à deux machines peut être résolu optimalement dans  $O(T^3)$  étapes. Finalement, nous avons proposé 7 heuristiques dont 5 heuristiques ont amélioré les résultats de la littérature en termes de qualité de solutions. Dans la pratique, un manager de production peut appliquer nos 5 heuristiques pour réduire le coût d’électricité sans risques d’augmentation de makespan.

Nous avons appliqué notre approche pour résoudre le problème sous la tarification *ToU*. Cependant, l’approche proposée peut être utilisée pour d’autres tarifications CPP, RTP. Deuxièmement, les fonctions de coût d’électricité dans nos études sont linéaires. Dans la littérature, des fonctions non-linéaire se trouvent dans les études de [Haouassi et al. (2016)], [Zhang et al. (2017)]. Ces fonctions sont favorables pour optimiser le pic dans la consommation d’électricité. Nous pouvons étendre notre étude en linéarisant les fonctions objectif non-linéaires. Mais le temps d’exécution devient considérable, cela demande des études supplémentaires pour pouvoir obtenir une approche plus efficace. De l’autre côté, il est connu que l’ensemble des solutions optimales de  $F2|perm|C_{max}$  n’est pas pleinement exploité et d’autres propositions pour étendre l’ensemble de solutions doivent être étudiées. Les études approfondies seront très favorables pour l’approche d’ordonnancement robuste.

Le chapitre 4 traite le problème la minimisation du coût d’électricité de flow shop à deux machines sous *ToU* dont la consommation énergétique est liée aux états de fonctionnement des machines : en marche et idle. Nous étudions des méthodes exactes pour résoudre ce problème : la programmation mathématique et la décomposition de Benders. Premièrement, nous étudions des structures spécifiques de prix d’électricité de *ToU* qui ne comprennent que deux intervalles de prix. Pour deux intervalles de prix, si le prix de l’électricité diminue, nous pouvons obtenir la solution optimale en appliquant la règle de Johnson et la programmation dynamique. Si le prix de l’électricité augmente, nous fournissons une condition d’optimalité pour la règle de Johnson. Deuxièmement, pour les structures de prix générales de *ToU*, en adaptant la règle de Johnson, nous développons une propriété qui permet de déterminer la séquence optimale à partir de l’ensemble des tâches dans chaque intervalle. L’ensemble des tâches dans un intervalle continent des tâches dont la date de début sur la machine 2 se trouve dans cet intervalle. Troisièmement, pour intégrer la propriété développée, nous proposons un nouveau modèle mathématique linéaire en nombres entiers mixtes. Ce dernier repose sur le modèle mathématique de [Wang et al. (2018)]. Nous remplaçons des variables binaires qui déterminent l’état de fonctionnement des machines par des variables entières. Ensuite, nous concevons une méthode exacte utilisant la décomposition de Benders pour résoudre le problème. Le modèle mathématique de problème maître

est relaxé dans le modèle que nous développons. Le problème esclave se base sur la propriété développée et la programmation dynamique. Enfin, nous menons des tests numériques pour évaluer la performance de nos approches de résolution. Les résultats numériques affirment qu'une amélioration significative est apportée par nos approches de résolution par rapport aux résultats existants dans la littérature en termes de coût d'électricité total.

Plusieurs pistes de recherche seraient pertinentes pour poursuivre cette étude: nous pouvons nous concentrer sur l'étude d'une borne inférieure plus performante afin d'évaluer ou d'améliorer les approches de résolution. Nous pouvons également construire une heuristique ou une méta-heuristique qui intègre notre propriété développée pour réduire le temps de calcul. Pour pouvoir développer notre étude, nous pourrions intégrer les techniques de réduction de la consommation énergétique de l'approche d'efficacité énergétique dans notre étude tel que : la stratégie "Turn off/Turn on" machine, le choix de plusieurs vitesses de fonctionnement qui consomment différentes quantités d'énergie. Nous envisageons également l'atelier de flow shop à plusieurs machines pour étendre notre modèle mathématique. En plus, nous pouvons examiner un problème d'optimisation multi-objectif pour étudier le compromis entre la productivité et le coût de l'électricité. L'étude de multi objectif est intéressante pour offrir aux managers plus de possibilités selon le front de Pareto. Enfin, comme une des plus importantes motivations de recherche, nous voulons considérer l'application de nos approches de résolutions dans le cas pratique avec des instances de la vie réelle ou industrielle. L'ordonnement avec l'aspect énergétique est un sujet de plus en plus d'actualité et proche de l'industrie, une application industrielle de notre étude serait très intéressante.

## 6 Bibliographie

### References

- Aghelinejad, M., Ouazene, Y., and Yalaoui, A. (2018a). Production scheduling optimisation with machine state and time-dependent energy costs. *International Journal of Production Research*, 56(16):5558–5575.
- Aghelinejad, M., Ouazene, Y., and Yalaoui, A. (2019). Complexity analysis of energy-efficient single machine scheduling problems. *Operations Research Perspectives*, 6:100105.
- Aghelinejad, M. M., Ouazene, Y., and Yalaoui, A. (2018b). Energy efficient scheduling problems under time-of-use tariffs with different energy consumption of the jobs. *IFAC-PapersOnLine*, 51(11):1053–1058.
- Albers, S. and Fujiwara, H. (2007). Energy-efficient algorithms for flow time minimization. *ACM Transactions on Algorithms (TALG)*, 3(4):49–es.
- Antoniadis, A. and Huang, C.-C. (2013). Non-preemptive speed scaling. *Journal of Scheduling*, 16(4):385–394.
- Antoniadis, A., Huang, C.-C., and Ott, S. (2014). A fully polynomial-time approximation scheme for speed scaling with sleep state. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1102–1113. SIAM.
- Bampis, E., Kononov, A., Letsios, D., Lucarelli, G., and Nemparis, I. (2015). From preemptive to non-preemptive speed-scaling scheduling. *Discrete Applied Mathematics*, 181:11–20.
- Bego, A., Li, L., and Sun, Z. (2014). Identification of reservation capacity in critical peak pricing electricity demand response program for sustainable manufacturing systems. *International Journal of Energy Research*, 38(6):728–736.
- Bellman, R. (1966). Dynamic programming. *Science*, 153(3731):34–37.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252.
- Billaut, J. and Lopez, P. (1998). Enumeration of all optimal sequences in the two-machine flowshop. In *Computational Engineering in Systems Application (CESA'98), Symposium on Industrial and Manufacturing Systems, IEEE-SMC/IMACS*, page 378.
- Braithwait, S., Hansen, D., and O'Sheasy, M. (2007). Retail electricity pricing and rate design in evolving markets.
- Briand, C., La, H. T., and Erschler, J. (2006). A new sufficient condition of optimality for the two-machine flowshop problem. *European Journal of Operational Research*, 169(3):712–722.
- Bruzzone, A. A., Anghinolfi, D., Paolucci, M., and Tonelli, F. (2012). Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shops. *CIRP annals*, 61(1):459–462.

- Castro, P. M., Harjunkski, I., and Grossmann, I. E. (2009). New continuous-time scheduling formulation for continuous plants under variable electricity cost. *Industrial & engineering chemistry research*, 48(14):6701–6714.
- Castro, P. M., Harjunkski, I., and Grossmann, I. E. (2011). Optimal scheduling of continuous plants with energy constraints. *Computers & Chemical Engineering*, 35(2):372–387.
- Che, A., Wu, X., Peng, J., and Yan, P. (2017). Energy-efficient bi-objective single-machine scheduling with power-down mechanism. *Computers & Operations Research*, 85:172–183.
- Che, A., Zeng, Y., and Lyu, K. (2016). An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs. *Journal of Cleaner Production*, 129:565–577.
- Chen, B. and Zhang, X. (2019). Scheduling with time-of-use costs. *European Journal of Operational Research*, 274(3):900–908.
- Chen, L., Megow, N., Rischke, R., Stougie, L., and Verschae, J. (2018). Optimal algorithms for scheduling under time-of-use tariffs. *arXiv preprint arXiv:1811.12657*.
- Chen, L., Wang, J., and Xu, X. (2019). An energy-efficient single machine scheduling problem with machine reliability constraints. *Computers & Industrial Engineering*, 137:106072.
- Cheng, J., Steiner, G., and Stephenson, P. (2002). Fast algorithms to minimize the makespan or maximum lateness in the two-machine flow shop with release times. *Journal of Scheduling*, 5(1):71–92.
- Chretienne, P., Coffman Jr, E. G., Lenstra, J. K., and Liu, Z. (1997). Scheduling theory and its applications. *Journal of the Operational Research Society*, 48(7):764–765.
- Cohen, J., Cordeiro, D., and Raphael, P. L. F. (2014). Energy-aware multi-organization scheduling problem. In *European Conference on Parallel Processing*, pages 186–197. Springer.
- ComED (2019). *Real-time hourly prices*. <https://hourlypricing.comed.com/live-prices/>.
- COP21 (2016). *Paris agreement*. [https://ec.europa.eu/clima/policies/international/negotiations/paris\\_en#tab-0-0](https://ec.europa.eu/clima/policies/international/negotiations/paris_en#tab-0-0).
- Dabur, P., Singh, G., and Yadav, N. K. (2012). Electricity demand side management: Various concept and prospects. *International Journal of Recent Technology and Engineering (IJRTE)*, 1(1):1–6.
- Dai, M., Tang, D., Giret, A., Salido, M. A., and Li, W. D. (2013). Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 29(5):418–429.
- Dantzig, G. B. (1998). *Linear programming and extensions*, volume 48. Princeton university press.

- Dantzig, G. B. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations research*, 8(1):101–111.
- Ding, J.-Y., Song, S., Zhang, R., Chiong, R., and Wu, C. (2015). Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. *IEEE Transactions on Automation Science and Engineering*, 13(2):1138–1154.
- Dorigo, M. (1992). Optimization, learning and natural algorithms. *PhD Thesis, Politecnico di Milano*.
- Drake, R., Yildirim, M. B., Twomey, J. M., Whitman, L. E., Ahmad, J. S., and Lodhia, P. (2006). Data collection framework on energy consumption in manufacturing.
- Dréo, J., Pétrowski, A., Siarry, P., and Taillard, E. (2003). *Métaheuristiques pour l'optimisation difficile*.
- Dupty, M. H., Agrawal, P., and Rao, S. (2016). Scheduling under power and energy constraints. *arXiv preprint arXiv:1609.07354*.
- Fang, K., Uhan, N., Zhao, F., and Sutherland, J. W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, 30(4):234–240.
- Fang, K., Uhan, N. A., Zhao, F., and Sutherland, J. W. (2013). Flow shop scheduling with peak power consumption constraints. *Annals of Operations Research*, 206(1):115–145.
- Fang, K., Uhan, N. A., Zhao, F., and Sutherland, J. W. (2016). Scheduling on a single machine under time-of-use electricity tariffs. *Annals of Operations Research*, 238(1-2):199–227.
- Fang, K.-T. and Lin, B. M. (2013). Parallel-machine scheduling to minimize tardiness penalty and power cost. *Computers & Industrial Engineering*, 64(1):224–234.
- Gellings, C. W. (1996). Then and now: The perspective of the man who coined the term ‘dsm’. *Energy Policy*, 24(4):285–288.
- Glover, F. (1989). Tabu search—part i. *ORSA Journal on computing*, 1(3):190–206.
- Glover, F. (1990). Tabu search—part ii. *ORSA Journal on computing*, 2(1):4–32.
- Goldman, C. (2010). Coordination of energy efficiency and demand response.
- Gong, X., De Pessemer, T., Joseph, W., and Martens, L. (2016). A generic method for energy-efficient and energy-cost-effective production at the unit process level. *Journal of Cleaner Production*, 113:508–522.
- Gong, X., De Pessemer, T., Martens, L., and Joseph, W. (2019). Energy-and labor-aware flexible job shop scheduling under dynamic electricity pricing: A many-objective optimization investigation. *Journal of cleaner production*, 209:1078–1094.
- Gong, X., Van der Wee, M., De Pessemer, T., Verbrugge, S., Colle, D., Martens, L., and Joseph, W. (2017). Energy-and labor-aware production scheduling for sustainable manufacturing: A case study on plastic bottle manufacturing. *Procedia CIRP*, 61:387–392.

- Graham, R. L., Lawler, E. L., Lenstra, J. K., and Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics*, volume 5, pages 287–326. Elsevier.
- Gupta, J. N. and Stafford Jr, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169(3):699–711.
- Gutowski, T., Murphy, C., Allen, D., Bauer, D., Bras, B., Piwonka, T., Sheng, P., Sutherland, J., Thurston, D., and Wolff, E. (2005). Environmentally benign manufacturing: observations from japan, europe and the united states. *Journal of Cleaner Production*, 13(1):1–17.
- Hadda, H., Dridi, N., and Hajji, M. K. (2018). On the optimality conditions of the two-machine flow shop problem. *European Journal of Operational Research*, 266(2):426–435.
- Hadera, H., Harjunkski, I., Sand, G., Grossmann, I. E., and Engell, S. (2015). Optimization of steel production scheduling with complex time-sensitive electricity cost. *Computers & Chemical Engineering*, 76:117–136.
- Haouassi, M., Desdouts, C., Giroudeau, R., and Le Pape, C. (2016). Production scheduling with a piecewise-linear energy cost function. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–8. IEEE.
- He, Y., Li, Y., Wu, T., and Sutherland, J. W. (2015). An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops. *Journal of Cleaner Production*, 87:245–254.
- He, Y., Liu, B., Zhang, X., Gao, H., and Liu, X. (2012). A modeling method of task-oriented energy consumption for machining manufacturing system. *Journal of Cleaner Production*, 23(1):167–174.
- Hooker, J. N. and Ottosson, G. (2003). Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60.
- IEA (2019). *CO2 emissions from fuel combustion 2018 overview*. <https://www.iea.org/statistics/co2emissions/>.
- Ji, M., Wang, J.-Y., and Lee, W.-C. (2013). Minimizing resource consumption on uniform parallel machines with a bound on makespan. *Computers & operations research*, 40(12):2970–2974.
- Jia, Z.-h., Wang, Y., Wu, C., Yang, Y., Zhang, X.-y., and Chen, H.-p. (2019). Multi-objective energy-aware batch scheduling using ant colony optimization algorithm. *Computers & Industrial Engineering*, 131:41–56.
- Jiang, E.-d. and Wang, L. (2019). An improved multi-objective evolutionary algorithm based on decomposition for energy-efficient permutation flow shop scheduling problem with sequence-dependent setup time. *International Journal of Production Research*, 57(6):1756–1771.
- Jiang, Z., Le, Z., et al. (2014). Study on multi-objective flexible job-shop scheduling problem considering energy consumption. *Journal of Industrial Engineering and Management (JIEM)*, 7(3):589–604.

- Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval research logistics quarterly*, 1(1):61–68.
- Kamburowski, J. (1997). The nature of simplicity of johnson’s algorithm. *Omega*, 25(5):581–584.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.
- Lei, D. and Guo, X. (2015). An effective neighborhood search for scheduling in dual-resource constrained interval job shop with environmental objective. *International Journal of Production Economics*, 159:296–303.
- Li, C., Tang, Y., Cui, L., and Li, P. (2015). A quantitative approach to analyze carbon emissions of cnc-based machining systems. *Journal of Intelligent Manufacturing*, 26(5):911–922.
- Liang, J., Wang, Y., Zhang, Z.-H., and Sun, Y. (2019). Energy efficient production planning and scheduling problem with processing technology selection. *Computers & Industrial Engineering*, 132:260–270.
- Liu, C., Yang, J., Lian, J., Li, W., Evans, S., and Yin, Y. (2014a). Sustainable performance oriented operational decision-making of single machine systems with deterministic product arrival time. *Journal of cleaner production*, 85:318–330.
- Liu, C.-H. (2014). Approximate trade-off between minimisation of total weighted tardiness and minimisation of carbon dioxide (co<sub>2</sub>) emissions in bi-criteria batch scheduling problem. *International Journal of Computer Integrated Manufacturing*, 27(8):759–771.
- Liu, C.-H. and Huang, D.-H. (2014). Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *International Journal of Production Research*, 52(2):337–352.
- Liu, G.-S., Yang, H.-D., and Cheng, M.-B. (2017). A three-stage decomposition approach for energy-aware scheduling with processing-time-dependent product quality. *International Journal of Production Research*, 55(11):3073–3091.
- Liu, Y., Dong, H., Lohse, N., and Petrovic, S. (2016). A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance. *International Journal of Production Economics*, 179:259–272.
- Liu, Y., Dong, H., Lohse, N., Petrovic, S., and Gindy, N. (2014b). An investigation into minimising total energy consumption and total weighted tardiness in job shops. *Journal of Cleaner Production*, 65:87–96.
- Lu, C., Gao, L., Li, X., Pan, Q., and Wang, Q. (2017). Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *Journal of Cleaner Production*, 144:228–238.
- Mansouri, S. A., Aktas, E., and Besikci, U. (2016). Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption. *European Journal of Operational Research*, 248(3):772–788.

- Mashaie, M. and Lennartson, B. (2012). Energy reduction in a pallet-constrained flow shop through on-off control of idle machines. *IEEE Transactions on Automation Science and Engineering*, 10(1):45–56.
- May, G., Stahl, B., Taisch, M., and Prabhu, V. (2015). Multi-objective genetic algorithm for energy-efficient job shop scheduling. *International Journal of Production Research*, 53(23):7071–7089.
- Meyabadi, A. F. and Deihimi, M. H. (2017). A review of demand-side management: Reconsidering theoretical framework. *Renewable and Sustainable Energy Reviews*, 80:367–379.
- Modos, I., Scha, P., and Hanzalek, Z. (2017). Algorithms for robust production scheduling with energy consumption limits. *Computers & Industrial Engineering*, 112:391–408.
- Mokhtari, H. and Hasani, A. (2017). An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. *Computers & Chemical Engineering*, 104:339–352.
- Moon, J.-Y., Shin, K., and Park, J. (2013). Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. *The International Journal of Advanced Manufacturing Technology*, 68(1-4):523–535.
- Mouzon, G. and Yildirim, M. B. (2008). A framework to minimise total energy consumption and total tardiness on a single machine. *International Journal of Sustainable Engineering*, 1(2):105–116.
- Mouzon, G., Yildirim, M. B., and Twomey, J. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of production research*, 45(18-19):4247–4271.
- Newsham, G. R. and Bowker, B. G. (2010). The effect of utility time-varying pricing and load control strategies on residential summer peak electricity use: a review. *Energy policy*, 38(7):3289–3296.
- Outlook, B. E. (2019). 2019 edition. *London, United Kingdom 2019*.
- Palensky, P. and Dietrich, D. (2011). Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE transactions on industrial informatics*, 7(3):381–388.
- Pilerood, A. E., Heydari, M., and Mazdeh, M. M. (2018). A two-stage greedy heuristic for a flowshop scheduling problem under time-of-use electricity tariffs. *The South African Journal of Industrial Engineering*, 29(1):143–154.
- Rager, M., Gahm, C., and Denz, F. (2015). Energy-oriented scheduling based on evolutionary algorithms. *Computers & Operations Research*, 54:218–231.
- Ramos, A. G. and Leal, J. (2017). Ilp model for energy-efficient production scheduling of flake ice units in food retail stores. *Journal of Cleaner Production*, 156:953–961.
- Ritchie, H. and Roser, M. (2020a). *co<sub>2</sub> and greenhouse gas emissions. Our World in Data.* <https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions>.

- Ritchie, H. and Roser, M. (2020b). Energy. *Our World in Data*. <https://ourworldindata.org/energy>.
- Rozycki, R. and Weglarz, J. (2012). Power-aware scheduling of preemptable jobs on identical parallel processors to meet deadlines. *European Journal of Operational Research*, 218(1):68–75.
- RTE (2019). *Réseau de transport d'électricité*. <https://www.rte-france.com/fr/eco2mix/eco2mix-co2>.
- Sevaux, M. (2004). *Métaheuristiques: Stratégies pour l'optimisation de la production de biens et de services*. PhD thesis.
- Sharma, A., Zhao, F., and Sutherland, J. W. (2015). Econological scheduling of a manufacturing enterprise operating under a time-of-use electricity tariff. *Journal of Cleaner Production*, 108:256–270.
- Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., and Ortega-Mier, M. (2014). Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *Journal of Cleaner Production*, 67:197–207.
- Subai, C., Baptiste, P., and Niel, E. (2006). Scheduling issues for environmentally responsible manufacturing: The case of hoist scheduling in an electroplating line. *International Journal of Production Economics*, 99(1-2):74–87.
- Summerbell, D. L., Khripko, D., Barlow, C., and Hesselbach, J. (2017). Cost and carbon reductions from industrial demand-side management: Study of potential savings at a cement plant. *Applied Energy*, 197:100–113.
- Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2):278–285.
- Tang, D., Dai, M., Salido, M. A., and Giret, A. (2016). Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. *Computers in Industry*, 81:82–95.
- Wang, S., Zhu, Z., Fang, K., Chu, F., and Chu, C. (2018). Scheduling on a two-machine permutation flow shop under time-of-use electricity tariffs. *International Journal of Production Research*, 56(9):3173–3187.
- Wang, Y. and Li, L. (2016). Critical peak electricity pricing for sustainable manufacturing: Modeling and case studies. *Applied energy*, 175:40–53.
- Wu, X., Shen, X., and Li, C. (2019). The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously. *Computers & Industrial Engineering*, 135:1004–1024.
- Yalaoui, A., Chehade, H., Yalaoui, F., and Amodeo, L. (2012). Optimisation en logistique: méthodes et techniques.
- Yildirim, M. B. and Mouzon, G. (2011). Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm. *IEEE transactions on engineering management*, 59(4):585–597.

- Zhang, B., Pan, Q.-k., Gao, L., Li, X.-y., Meng, L.-l., and Peng, K.-k. (2019). A multi-objective evolutionary algorithm based on decomposition for hybrid flowshop green scheduling problem. *Computers & Industrial Engineering*, 136:325–344.
- Zhang, H., Zhao, F., and Sutherland, J. W. (2017). Scheduling of a single flow shop for minimal energy cost under real-time electricity pricing. *Journal of Manufacturing Science and Engineering*, 139(1):014502.
- Zhang, R. and Chiong, R. (2016). Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production*, 112:3361–3375.
- Zheng, X., Zhou, S., Xu, R., and Chen, H. (2019). Energy-efficient scheduling for multi-objective two-stage flow shop using a hybrid ant colony optimisation algorithm. *International Journal of Production Research*, pages 1–18.

# Minh Hung HO

## Doctorat : Optimisation et Sûreté des Systèmes

### Année 2020

#### L'optimisation énergétique de l'ordonnement des systèmes de production

Avec l'émergence du *Smart Grid* et des outils de communication entre clients et compagnies d'électricité, la réponse à la demande attire de plus en plus attentions avec des tarification d'électricité tels que « *Time-of-Use* » (ToU), « *Critical-Peak-Pricing* ». Dans ce contexte, nous menons une étude sur l'optimisation de coût d'électricité (TEC) dans le problème d'ordonnement de *flow shop* à deux machines sous la tarification d'électricité ToU. Nous traitons deux sous problèmes. Le premier minimise le TEC de la solution optimale de (F2||Cmax). Nous construisons un plan de production de coût d'électricité minimal avec le *makespan* optimal. L'approche de résolution repose sur la théorie des solutions équivalentes de (F2||Cmax) qui sera appliquée pour la première fois dans l'optimisation de coût d'électricité. Le deuxième minimise le TEC de *flow shop* à deux machines avec la prise en compte de plusieurs états de fonctionnement de machines. Nous développons une proposition fondée sur la règle de Johnson pour déterminer la séquence optimale pour les structures générales de ToU. Nous concevons une nouvelle programmation linéaire et une méthode exacte fondée sur la décomposition de Benders pour résoudre le problème. Les tests numériques montrent que l'approche proposée améliore d'une manière significative la qualité des résultats existants dans la littérature. La thèse permet de construire un outil d'aide de décision au manager de réduire le coût d'électricité de production, d'ouvrir de nouvelles perspectives pour l'optimisation énergétique dans l'industrie.

**Mots clés :** ordonnancement (gestion) – services de l'électricité, tarifs – optimisation combinatoire – coût de production – prise de décision, modèles mathématiques.

#### Energy Optimization of the Scheduling Problem in Production Systems

With the emergence of Smart Grid and communication technology between customer and supplier, there has been a growing interest in Demand Response research with different electricity pricing such as "Time-of-Use" (ToU), "Critical-Peak-Pricing". As a response to these circumstances, we study a two-machine flow shop scheduling problem with minimization of total electricity cost (TEC) under ToU. We deal with two sub-problems. The first one tackles the joint optimization of makespan and TEC in a two-machine flow shop scheduling problem under ToU. We optimize TEC without increasing the makespan. The resolution approach is based on the theory of equivalent solutions of (F2 || Cmax) which is applied for the first time in the optimization of electricity costs. The second problem minimizes the TEC of a two machines flow shop scheduling problem with the consideration of several machine operating states under ToU. We develop a property based on Johnson's rule to determine the optimal sequence for general ToU pricing structures. We design a new linear programming and an exact method based on the Benders decomposition to solve the problem. The numerical tests show that our proposed approach significantly improves the quality of existing results in the literature. The thesis builds a decision support tool for managers to reduce the cost of electricity in production and opens up new perspectives for energy optimization in industry.

**Keywords:** production scheduling – electric utilities, rates – combinatorial optimization – costs, industrial – decision making, mathematical models.

Thèse réalisée en partenariat entre :

