



HAL
open science

Investigation pour la résolution des problèmes flow shop de permutation sous contrainte de ressource

Imane Laribi, Fayçal Belkaid, Zaki Sari, Farouk Yalaoui

► To cite this version:

Imane Laribi, Fayçal Belkaid, Zaki Sari, Farouk Yalaoui. Investigation pour la résolution des problèmes flow shop de permutation sous contrainte de ressource. Xème Conférence Internationale : Conception et Production Intégrées, Dec 2015, Tanger, Maroc. hal-01260704

HAL Id: hal-01260704

<https://hal.science/hal-01260704v1>

Submitted on 22 Jan 2016

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.

Investigation pour la résolution des problèmes flow shop de permutation sous contrainte de ressource

Imane Laribi, Fayçal Belkaid
Laboratoire MELT, Université Abou
Bakr Belkaid
Tlemcen, Algérie

Zaki Sari
Department of Industrial Engineering,
Izmir University of Economics, Izmir,
35330 Turkey

Farouk Yalaoui
Laboratoire LOSI, Université de
Technologie de Troyes
Troyes, France

Abstract—Nous traitons, dans ce papier, un problème d'ordonnancement dans un environnement de production flow shop de permutation sous contrainte de ressource non-renouvelable (matière première). Dans ce type de problème, les ressources nécessaires à l'exécution d'une tâche peuvent être consommées comme elles peuvent être indisponibles pour des raisons diverses, par conséquent, le manque de ressources perturbe et affecte l'efficacité de l'ordonnancement. Un état de l'art a été réalisé, ce dernier a pour objectif d'expliquer les choix théoriques qui constituent la base de notre problématique et qui ont guidé nos recherches. Par la suite, nous proposons un modèle linéaire en nombres entiers pour minimiser la durée totale d'ordonnancement. Ce problème est NP- difficile au sens fort.

Keywords—Optimisation; flowshop; ressource non renouvelable; makespan; modèle mathématique.

I. INTRODUCTION

Un ordonnancement efficace peut être un outil essentiel pour les entreprises afin d'atteindre des performances élevées. En effet, l'ordonnancement constitue un aspect important dans la gestion de production, il consiste à définir la planification de l'utilisation des ressources disponibles (consommables et/ ou renouvelables) de façon à déterminer à quel moment traiter quelle tâche avec quelle ressource tout en réglant un ensemble de conflits et en optimisant un ou plusieurs critères.

L'ordonnancement sous contrainte de ressource ont fait l'objet de recherche depuis plus de 50 ans dont l'une des premières publications qui ont utilisé ce terme était la thèse de doctorat de [1] qu'elle porte sur l'ordonnancement de projet.

Selon la littérature, Les ressources peuvent être classifiées en :

- Ressource renouvelable, une fois qu'elle est utilisé par un job, elle peut être utilisé à nouveau par un autre job.

- Ressource non-renouvelable, une fois utilisé par certain job, elle ne peut pas être disponible pour un autre.

Dans ce travail nous nous intéressent uniquement aux ressources non-renouvelables. Les problèmes

Xème Conférence Internationale : Conception et Production Intégrées, CPI 2015, 2-4 Décembre 2015, Tanger - Maroc.

Xth International Conference on Integrated Design and Production, CPI 2015, December 2-4, 2015, Tangier - Morocco.

d'ordonnancement avec ressources consommables (non renouvelables) sont devenus fréquents dans les entreprises manufacturières. Toutefois, une partie importante des études des problèmes d'ordonnancement est placée dans un contexte où les ressources sont toujours disponibles, or ces ressources peuvent être consommées et peuvent être indisponibles pour des raisons diverses, de ce fait cette hypothèse peut ne pas être satisfaite dans de nombreuses situations pratiques et par conséquent, le manque de ressources perturbe et affecte l'efficacité de l'ordonnancement.

II. ÉTAT DE L'ART

L'ordonnancement sous contrainte de ressources non renouvelables est un domaine intéressant avec une énorme importance pratique, dans ce contexte les tâches doivent être traitées sur la machine tout en respectant la disponibilité de certaines ressources non renouvelables qui seront consommées par les différentes tâches mais réapprovisionnées au cours du temps à partir de ressources externe, ce qui implique que non seulement la machine doit être libre quand la tâche commence son traitement mais aussi les ressources nécessaires doivent être disponible en quantités suffisantes. Dans l'analyse suivante, les travaux sont évalués par rapport à trois thèmes principaux :

- Environnement machine.
- Fonction objectif
- Approche de résolution.

La recherche dans le domaine d'ordonnancement sous contrainte de ressources non renouvelables est assez limitée. Dans [2], les auteurs présentent des algorithmes polynomiaux pour les problèmes d'ordonnancement avec des contraintes de précédence. Dans [3], le papier étudie la complexité d'ordonnancement avec des contraintes de ressources, un schéma de classification a été proposé dont le but est de minimiser le makespan. Quelque résultat pour l'ordonnancement préemptif dans un environnement de machines parallèles avec des contraintes de ressource renouvelable et non renouvelable (contraintes financières) a

été présenté par [4]. Dans ce papier, l'auteur suppose que le taux de consommation des ressources financières soit constant pendant le traitement des jobs, l'auteur propose une procédure exacte à deux phases utilisant la programmation linéaire pour minimiser la durée totale d'ordonnancement (notée C_{max}).

[5] présentent certaines variations pour le problème d'ordonnancement préemptif sur machines parallèles indépendantes, ils considèrent la contraintes de ressources consommables dont l'approvisionnement varie avec le temps, les auteurs présentent une méthode a deux phases, dont la première phase consiste à résoudre un programme linéaire, la deuxième phase consiste à construire l'ordre de passage.

Dans [6], Les auteurs prouvent que si la quantité de ressources qui devient disponible à chaque période de temps est constante dans un environnement avec une seule machine et une seule ressource non renouvelable, le problème devient équivalent à un flow shop à deux machines sans contrainte de ressource, l'algorithme de Johnson [7] a été appliqué dans le but de minimiser la durée totale de l'ordonnancement.

Dans [8], Les auteurs présentent un modèle pour l'ordonnancement préemptive dans un environnement open shop avec l'addition d'une seule ressource non renouvelable.

Dans [9], les auteurs considèrent le problème d'ordonnancement open shop avec des contraintes de ressource renouvelable et consommable dans le but de minimiser le makespan, ils présentent une approche de coloration de bord pour modéliser le problème. Les auteurs présentent un algorithme polynomial pour le cas où la courbe de la disponibilité des ressources est sous forme d'escalier et cela pour étendre les résultats présentés dans [8].

Dans [10], les auteurs proposent deux heuristiques et une procédure par séparation évaluation afin de minimiser le retard maximum dans un environnement de machines parallèles avec contrainte de ressources.

Dans ce papier [11] l'auteur généralise le résultat de [6] au problème d'ordonnancement à une machine avec plusieurs ressources financières, ou chaque ressource est disponible en quantité différentes, l'auteur montre que le problème peut être réduit à un flow shop a deux machines si les ressources arrivent de manière uniforme dans le temps, il a également montré que la règle LPT est optimal si les ressources financières sont consommées de façon uniforme par toutes les tâches, des algorithmes polynomiaux ont été proposé pour minimisé la durée total d'ordonnancement pour les deux cas précédents.

Des algorithmes approximatifs pour plusieurs cas particulier de l'ordonnancement a une seul machine avec l'addition de contrainte de ressources non renouvelables dont l'objectif est de minimiser C_{max} et L_{max} , ont été proposés dans [12].

Les auteurs [13] considèrent le problème d'ordonnancement sur machines parallèles identiques où les temps de traitement des jobs sont contrôlables par l'allocation d'une ressource commune non renouvelable, L'objectif est d'affecter les jobs aux machines, de séquencer les jobs sur chaque machine et d'allouer les ressources de sorte que le makespan ou le flow time est minimisé. L'optimisation se fait pour les deux cas préemptifs et non préemptifs. Les auteurs montrent que le problème de C_{max} avec non préemptif jobs est NP-difficile.

Cette thèse [14] étudie un problème d'ordonnancement pour une plateforme logistique sur une seule machine avec des ressources consommables et des dates de livraison fixe. Le problème est caractérisé par l'arrivée de plusieurs composants (ressources) suivant des courbes sous forme d'escaliers, une méthode exacte basée sur la procédure par séparation et évaluation a été proposée pour minimiser makespan.

Dans [15] les auteurs étendent le travail présenté par [12] en présentant des résultats de complexités pour plusieurs fonctions objectifs particulièrement la minimisation de la somme des retards.

Dans ce papier [16] le système étudié est composé de plusieurs machines parallèles identiques avec l'addition des ressources consommables, l'arrivée des composants est représentée par une courbe sous forme d'escalier. Pour résoudre ce problème les auteurs développent un modèle mathématique linéaire en nombres entiers, modélisé par des variables de position pour minimiser le makespan, ainsi qu'une métaheuristique à base d'algorithme génétique a été proposé et comparé avec des heuristiques à base de règles de priorité, le résultat obtenu montre l'efficacité de la métaheuristique proposée.

Une extension de contrainte de ressources non renouvelables au problème d'ordonnancement flow shop hybride est étudiées par [17], dans ce travail plusieurs heuristiques à base de règles de priorité ont été évaluées et comparées sur des instances différentes.

D'après l'état de l'art, nous avons remarqué que la contrainte de ressource non-renouvelable n'a pas été traitée dans l'environnement flow shop de permutation. De ce fait, cela nous a donné la motivation pour étudier le problème proposé, qui étend les travaux menés précédemment.

Une synthèse des travaux portant sur les problèmes d'ordonnancement avec contrainte de ressources non renouvelables est présentée dans le tableau I.

TABLE I. SYNTHÈSE DE TRAVAUX SUR LES PROBLÈMES D'ORDONNACEMENT AVEC CONTRAINTES DE RESSOURCES NON RENOUVELABLES

Les travaux	Environnement machine						Les mesures de performances		Méthode de résolution
	<i>I</i>	<i>PM</i>	<i>JS</i>	<i>OS</i>	<i>FS</i>	<i>FSH</i>	<i>Cmax</i>	<i>Autres</i>	
[2]							X		Algorithmes en temps polynomial
[3]		X					X		Schéma de classification
[4]		X					X	X	Méthode exacte
[5]		X					X		Programme linéaire
[6]	X						X		Algorithmes en temps polynomial
[8]				X			X		Etude de complexité du problème
[9]				X			X		Algorithme polynomial
[10]		X						X	Heuristiques Procédure par séparation évaluation
[11]		X					X		Algorithme polynomial
[12]	X						X	X	Algorithme approximative
[13]		X					X	X	Méthode de charge équivalente
[14]	X						X		Procédure par séparation évaluation

[15]	X					X	X	Etude de complexité de problème pour des différentes mesures de performance
[16]		X				X		Modèle mathématique Metaheuristique (AG)
[18]	X					X		Schéma d'approximation polynomial.
[17]					X	X		Heuristiques

1 : une seule machine
 PM : machines parallèles.
 JS: job shop.
 OS: open shop.
 FS: flow shop
 FSH: flow shop hybride.

III. DESCRIPTION DE PROBLEME

Nous considérons le problème d'ordonnancement avec la contrainte de ressource non-renouvelable afin de minimiser le makespan dans un environnement flow shop de permutation. La consommation de ressource s'effectue uniquement sur la dernière machine. Le problème peut être décrit comme suit :

- N jobs doivent être traités sur un ensemble de M machines.
- Toutes les jobs passent sur toute les machines dans le même ordre (m_1, m_2, \dots, m_M).
- Le traitement des jobs sur la dernière machine nécessitent un ou plusieurs composants (ressources non renouvelables).
- L'arrivée de chaque ressource est représentée par une courbe sous forme d'escalier.
- Les quantités totales des ressources disponibles à un instant défini sont limitées.
- La consommation totale d'une ressource par l'ensemble des jobs ne doit pas dépasser la quantité disponible de cette ressource.
- Un job peut commencer sur la dernière machine dès qu'il termine son traitement sur la machine précédente et seulement si toutes les ressources nécessaires sont disponibles à cet instant.
- Toutes les machines ne peuvent exécuter qu'une opération à la fois.
- Les N jobs sont à exécuter sans interruption (préemption non autorisée).
- Toutes les machines et tous les jobs sont disponibles à partir de l'instant zéro.
- Le temps de traitement sur les machines étant des entiers.

D'après la notation établie par [19], le problème d'ordonnancement de type flow-shop de permutation sous contrainte de ressources non renouvelables est symbolisé par $F/perm, NR : a_{it} / C_{max}$. Où $NR : a_{it}$ indique l'existence de

ressources non renouvelables, qui deviennent disponibles en une quantité de a_{it} dans un temps t .

IV. MODELE MATHEMATIQUE

Afin de déterminer une solution optimale pour résoudre le problème de minimisation du makespan d'un flow shop de permutation sous contrainte de ressource non-renouvelable, nous allons proposer un modèle mathématique linéaire en nombres entiers, cette modélisation est inspirée du modèle présenté par [14]. Pour éviter la grande quantité de variables de précédence et de contraintes indexées par le temps, nous avons choisi d'utiliser des variables de position.

A. Notations

TABLE II. PARAMÈTRES

Paramètre	Description
N	Nombre de jobs.
M	Nombre de machines.
R	Nombre de ressources.
P_{ij}	Temps opératoire du job i sur la machine j
Q_{il}	Quantité de ressource l que le job i consomme
Z_{it}	La quantité totale de ressource l arrivée à l'instant t
M	Nombre positif très grand

TABLE III. INDICES

Indice	Pour	Echelle
i	Jobs	$\{1 \dots N\}$
j	Machines	$\{1 \dots M\}$
l	Ressources	$\{1 \dots R\}$
k	Position	$\{1 \dots N\}$
t	Temps	$\{T_1, \dots, T_{max}\}$

TABLE IV. VARIABLES

Variable	Description
C_{kj}	Date de fin de traitement du job en position k sur la machine j
S_{kj}	Date de début de traitement du job en position k sur la machine j
W_{kl}	Quantité de ressource l que le job en position k consomme
X_{ik}	Variable binaire, égale à 1 si le job i est en position k , 0 sinon
Y_{kt}	Variable binaire, égale à 1 si $S_{kj} \geq t$, 0 sinon.

B. Modèle

En utilisant les paramètres et les variables décrits précédemment, le problème peut être modélisé de la façon suivante :

$$\text{Min } C_{max} \quad (1)$$

Sous les contraintes :

$$\sum_{i=1}^N X_{ik} = 1, \quad \forall k = 1 \dots N \quad (2)$$

$$\sum_{k=1}^N X_{ik} = 1, \quad \forall i = 1 \dots N \quad (3)$$

$$C_{kj} \geq C_{(k-1)j} + \sum_{i=1}^N X_{ik} * P_{ij}, \quad \forall k = 2 \dots N, \forall j = 1 \dots M \quad (4)$$

$$C_{kj} \geq C_{k(j-1)} + \sum_{i=1}^N X_{ik} * P_{ij}, \quad \forall k = 1 \dots N, \forall j = 2 \dots M \quad (5)$$

$$C_{1j} \geq C_{1(j-1)} + \sum_{i=1}^N X_{i1} * P_{ij}, \quad \forall j = 2 \dots M \quad (6)$$

$$C_{11} = \sum_{i=1}^N X_{i1} * P_{i1} \quad (7)$$

$$S_{kj} = C_{kj} - \sum_{i=1}^N X_{ik} * P_{ij}, \quad \forall k = 1 \dots N, \forall j = 1 \dots M \quad (8)$$

$$W_{kl} = \sum_{i=1}^n X_{ik} * Q_{il}, \quad \forall k = 2 \dots M, \forall l = 1 \dots R \quad (9)$$

$$\sum_{v=1}^k W_{vl} \leq \sum_{t=T_1}^{T_{max}} Z_{tl} * Y_{kt}, \quad \forall l=1 \dots R, k=1 \dots N \quad (10)$$

$$M(Y_{kt-1}) \leq S_{km} - t, \quad \forall t=T_1 \dots T_{max}, k=1 \dots N \quad (11)$$

$$C_{kj} \geq 0, \quad \forall k = 1 \dots N, \forall j = 1 \dots M \quad (12)$$

$$C_{max} \geq C_{NM} \quad (13)$$

Nous donnons dans ce paragraphe, la signification de chacune des contraintes de notre modèle :

- (1) représente la fonction objectif de notre problème qui consiste à minimiser la date de fin de l'ordonnancement, notée C_{max} .
- Contrainte (2) assure que chaque job doit occuper exactement une seule position
- Contrainte (3) garantit que chaque job doit être affecté exactement une seule fois sur chaque machine.
- Contrainte (4) assure qu'aucun job ne peut commencer sur une machine avant la fin de job en cours.

- Contrainte (5) contrôle que le job en position k doit finir son opération sur la machine $j-1$, avant de pouvoir démarrer son opération sur la machine suivante (Cette contrainte représente la relation de précedence entre deux opérations successives d'un même job).
- Contrainte (6) a pour objectif le calcul de temps de fin de traitement de job placer en position un sur la machine j , qui est supérieur ou égale au temps de fin de traitement de job placer en position un sur la machine $j-1$.
- Contrainte (7) représente la date de fin de traitement du job placer en position 1 sur la machine 1.
- Contrainte (8) calcule la date de début du job qui est en position k sur la machine j .
- Contrainte (9) permet de calculer le nombre de ressource l consommé par le job qui est en position k .
- Contrainte (10) vérifie que la quantité de ressource l consommée par le job en position k est inférieure ou égale au nombre qu'on dispose de cette ressource.
- Contrainte (11) met le lien entre la variable de décision Y_{kt} et l'instant de début de traitement du job en position k .
- Contrainte (12) assure que la date de fin du job sur n'importe qu'elle position ne peut pas être négative.
- Contrainte (13) impose que la valeur de C_{max} doit être supérieure ou égale à la date de fin de tous les jobs sur la dernière machine.

C. Résultats expérimentaux

Afin d'évaluer les limites du modèle mathématique proposé, nous présentons ici les résultats numériques obtenue pour des instances de différentes dimensions.

Nous générons ces instances afin que le total des ressources nécessaires pour le traitement de chaque job soit inférieur ou égal au nombre de ressource disponible dans le système.

Le temps de traitement et la consommation de ressource pour chaque job sont générés aléatoirement comme suit :

- Le temps de traitement de chaque job est généré uniformément dans l'intervalle $[1,100]$.
- La consommation de ressources de chaque job est générée uniformément dans l'intervalle $[1,10]$.
- Nous effectuons des tests lorsque le nombre de ressources R varie dans l'intervalle $[1 - 2]$.

Le modèle mathématique est résolu avec le solveur de programmation linéaire CPLEX sur un Core (TM) i3 CPU - 1,50 GHz - 4,00 Go de Ram.

La durée d'exécution d'un modèle linéaire en nombres entiers étant généralement trop importante pour la déployer entièrement, par la suite, nous avons limité le temps de simulation à 1200 secondes.

Le tableau V présente le temps de calcul CPUtime (en secondes) nécessaire pour résoudre le problème de type flow shop de permutation sous contrainte de ressources non renouvelables.

TABLE V. EVALUATION DES LIMITES DU MODELE MATHEMATIQUE POUR UNE ET DEUX RESSOURCES

	Instances		CPLEX	
	Nombre de jobs	Nombre de machines	CPU _{time} R=1	CPU _{time} R=2
	Petites instances	10	5	0,53
10			1,62	0,34
20			2,70	0,62
20		5	64,91	0,23
		10	2,15	0,69
		20	>1200	>1200

	Instances		CPLEX	
	Nombre de jobs	Nombre de machines	CPU _{time} R=1	CPU _{time} R=2
	Moyennes instances	40	5	14,06
10			164,3	682,9
20			>1200	>1200
60		5	63,98	>1200
		10	>1200	>1200
		20	>1200	>1200

	Instances		CPLEX	
	Nombre de jobs	Nombre de machines	CPU _{time} R=1	CPU _{time} R=2
	Grandes instances	80	5	55,43
10			>1200	>1200
20			>1200	>1200
100		5	>1200	>1200
		10	>1200	>1200
		20	>1200	>1200

Dans le tableau V, nous remarquons que pour les petites instances, CPLEX a pu résoudre des instances jusqu'à 20 jobs

dans un temps raisonnable moins de 92,68s et cela pour une et deux ressources. Pour les moyennes instances, nous constatons que le temps de calcul augmente avec le nombre de machines et le nombre de jobs, dans ce cas CPLEX n'a pas pu résoudre des instances supérieures à 40 jobs avec plus de 5 machines.

Pour les grandes instances, nous observons que CPLEX a pu résoudre des instances jusqu'à 80 jobs avec 5 machines uniquement et une seule ressource non renouvelable.

En outre, il n'a pas pu résoudre ces instances lorsqu'on a augmenté le nombre de ressource non renouvelable à deux. L'incapacité de CPLEX à résoudre certaines instances est due à la contrainte de ressource non-renouvelable qui a augmenté la complexité du problème et par conséquent une croissance exponentielle de temps de calcul.

En effet, prendre en considération la contrainte de ressource non renouvelable complique la résolution du modèle mathématique qui devient NP-difficile. Afin de prouver cette affirmation, nous relaxons cette contrainte dans le tableau VI, c'est-à-dire que nous supposons que les jobs ne consomment pas de ressources, ensuite nous analysons les résultats obtenus par cette relaxation.

TABLE VI. EVALUATION DES LIMITES DU MODELE MATHEMATIQUE LORSQUE R= 0

	Instances		CPLEX
	Nombre de machines	Nombre de jobs	CPU _{time}
5	5	10	0,95
		20	50,67
		40	18,56
		60	3,01
		80	4,96
		100	10,89
		200	129,03
		400	922,05
		500	>1200

	Instances		CPLEX
	Nombre de machines	Nombre de jobs	CPU _{time}
10	10	10	3,63
		20	20,75
		40	898,06
		60	>1200

	Instances		CPLEX
	Nombre de machines	Nombre de jobs	CPU _{time}
20	20	10	17,32
		20	>1200

À travers le tableau V, nous pouvons remarquer que CPLEX a pu résoudre des instances jusqu'à 400 jobs lorsque le nombre de machines est égale à 5 dans un temps moins de 992 s. En outre, il a pu résoudre jusqu'à 40 jobs pour 10 machines et 10 jobs pour 20 machines.

V. CONCLUSION

Dans ce papier, nous avons présenté un modèle linéaire en nombres entiers pour résoudre le problème d'ordonnement d'un flow shop de permutation sous contrainte de ressource non renouvelable.

Ce modèle a été résolu avec le solveur de programmation linéaire CPLEX. Le modèle révèle performant pour les petites instances, mais il rencontre des difficultés pour résoudre les moyennes et les grandes instances. Ceci nous a encouragés à proposer, dans les travaux à venir, des méthodes approchées afin de résoudre les problèmes de tailles importantes dans un temps raisonnable.

References

- [1]. T. J. R. Johnson, "An algorithm for the resource-constrained project scheduling problem." 1967.
- [2]. J. Carlier and A. H. G. R. Kan, "scheduling subject to nonrenewable-resource constraints", *Operations research letters*, vol.1, pp. 2–5, 1982.
- [3]. J. Blazewicz, "Scheduling subject to resource constraints: classification and complexity ", *Discrete Applied Mathematics*, vol. 5 , pp.11- 24, 1983.
- [4]. R. Slowinski, "Preemptive scheduling of independent jobs on parallel machines subject to financial constraints," *European Journal of Operational Research*, vol. 15, pp. 366–373, 1984.
- [5]. M. Cochand, D. Werra, and R. Slowinski, "Preemptive scheduling with staircase and piecewise linear resource availability", *Methods and Models of Operations Research*, vol. 33, no. 5, pp. 297–313, sep. 1989.
- [6]. A. Toker, S. Kondakci, and N. Erkip, "scheduling under a non-renewable resource constraint," *Operational research society*., vol. 42, no. 9, pp. 811–814, 1991.
- [7]. S. M. Johnson, "Optimal two- and three-stage production schedules with set-up times included," *Naval Research Logistics Quarterly*, vol. 1, no. 1, pp. pp.61–68, 1954.
- [8]. D. Werra, J. Blazewicz, W. Kubiak "A preemptive open shop scheduling problem with one resource", *Operations Research Letters*, vol. 10, pp. 9–15, February, 1991.
- [9]. D. Werra and J. Blazewicz, "Some preemptive open shop scheduling problems with a renewable or a nonrenewable resource," *Discrete Applied Mathematics*., Vol. 35, no. 3, pp. 205–219, Mar. 1992.
- [10]. J. Blazewicz, W. Kubiak, and S. Martello, "Algorithms for minimizing maximum lateness with unit length tasks and resource constraints", *Discrete Applied Mathematics*, vol. 42, pp. 123–138, 1993.
- [11]. J. Xie, "Polynomial algorithms for single machine scheduling problems with financial constraints", *Operations Research Letters*., vol. 21, no. 1, pp. 39–42, Aug. 1997.
- [12]. A. Grigoriev, M. Holthuijsen, and J. van de Klundert, "Basic scheduling problems with raw material constraints," *Naval Research Logistics*, vol. 52, no. 6, pp. 527–535, Sep. 2005.
- [13]. D. Shabtay and M. Kaspi, "Parallel machine scheduling with a convex resource consumption function," *European Journal of Operational Research*., vol. 173, no. 1, pp. 92–107, Aug. 2006.
- [14]. S. Carrera, "Planification et Ordonnement de Plateformes Logistiques par," thèse, Laboratoire Lorrain de Recherche en Informatique, 2010.
- [15]. E. R. Gafarov, A. a. Lazarev, and F. Werner, "Single machine scheduling problems with financial resource constraints: Some complexity results and properties," *Mathematical Social Sciences*, vol. 62, no. 1, pp. 7–13, Jul. 2011.
- [16]. F. Belkaid, Z. Sari, and M. Souier, "A Genetic Algorithm for the Parallel Machine Scheduling Problem with Consumable Resources," *International Journal of Applied Metaheuristic Computing*, vol. 4, no. 2, pp. 17–30, 2013.
- [17]. I. Laribi, F. Belkaid, F. Yalaoui, and Z. Sari, "Investigation for solving hybrid flow shop problem with consumable resource," in *International Conference On Metaheuristics And Nature Inspired Computing, META 2014*.
- [18]. Györgyi, P. & Kis, T., " Approximation schemes for single machine scheduling with non-renewable resource constraints". *Journal of Scheduling*, 17(2), pp.135–144. 2014.
- [19]. Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G., « Optimization and Approximation in Deterministic Sequencing and Scheduling : A Survey », *Annals of Discrete Mathematics*, vol. 5, p. 287-326, 1979.