

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université M'hamed Bougara de Boumerdès



Faculté des Sciences
Département des Mathématiques

Mémoire présenté,
Par

M^r. Ferdi Ahmed Oussama & M^r. Slimi Mouloud

Pour l'obtention du diplôme de Master en Recherche Opérationnelle
Option : Recherche Opérationnelle, Optimisation et Management
Stratégique

*Optimisation et Planification de l'Ordonnancement des
Organes véhicule Cas : SNVI-VIR*

Soutenu publiquement le 10/11/2020, devant le jury composé de :

Présidente	M ^{me}	N.Larbi	M.A.A	U.M.B.B.
Promotrice	M ^{me}	B.Ferdjallah	M.A.A	U.M.B.B.
Examinatrice	M ^{me}	W.Drici	M.C.B	U.M.B.B.

Année Universitaire 2019 – 2020

Table des matières

Table des figures	4
Liste des tableaux	6
1 Introduction générale	10
2 Présentation de l'entreprise	12
2.1 Présentation de la Société National des Véhicules Industriels « SNVI » .	12
2.2 Historique de la SNVI	13
2.2.1 Ancienne Organisation	14
2.2.2 Nouvelle Réorganisation du Holding	14
2.3 Mission, Objectif et Organisation de L'entreprise	15
2.3.1 Mission de la « SNVI »	15
2.3.2 Objectifs de la « SNVI »	15
2.3.3 Organisation de La « SNVI »	16
2.4 Clients	19
2.5 Drections Usinage	19
2.5.1 Batiment Mécanique	19
2.5.2 Service Méthode	20
2.5.3 Service Production	20
2.5.4 Service Maintenance	20
2.5.5 Service Ordonnancement	20
2.5.6 Généralité sur le pont	22
2.5.7 Les différents opérations pour le montage de pont	23
2.6 Conclusion	25
3 Généralités sur l'ordonnancement	26
3.1 Introduction	26
3.2 Qu'est-ce que l'ordonnancement?	26
3.2.1 Définition :	26
3.3 Les données d'un problème d'ordonnancement	27
3.3.1 Les tâches	27
3.3.2 Les ressources	28

3.3.3	Les contraintes :	28
3.3.4	Les critères :	29
3.3.5	L'objectif de l'ordonnancement :	29
3.4	Diagramme de Gantt	30
3.5	Les types d'ordonnancement :	31
3.5.1	Ordonnancement simple :	31
3.5.2	L'ordonnancement de travaux dans un atelier non simple :	31
3.6	Caractérisation d'atelier :	31
3.6.1	Problème à machine unique :	31
3.6.2	Problème à machine parallèles	32
3.6.3	Problème à machine Spécialisée	33
3.7	Classification des problèmes d'ordonnancement	34
3.7.1	Le champ α	34
3.7.2	Champ β	35
3.7.3	Champ γ :	35
3.8	Conclusion	36
4	Modelisation du problème	37
4.1	Introduction	37
4.1.1	Position du problème	37
4.1.2	Les hypothèses	38
4.1.3	Objectif à atteindre	38
4.2	Principe de la modélisation	38
4.2.1	Position du Flow Shop de permutation	39
4.2.2	Etude de l'état de l'art du problème $F_m/permu/C_{max}$	40
4.2.3	Complexité	43
4.2.4	Calcul de la longueur d'un ordonnancement	45
4.2.5	Modèle mathématique	45
4.3	Conclusion	48
5	Approche de résolution	49
5.1	Introduction	49
5.1.1	Les Algorithmes Génétiques	50
5.1.2	Terminologie des algorithmes génétiques	50
5.1.3	Avantages des algorithmes génétiques	51
5.1.4	Principes de base des algorithmes génétiques	51
5.1.5	Fonctionnement des algorithmes génétiques	52
5.2	Conclusion	58

6 Implémentaion	59
6.1 Introduction	59
6.1.1 présentation du logiciel Python	59
6.1.2 Bibliothèques utilisées	60
6.2 Les données	60
6.3 Implémentation et résultats	61
6.4 Conclusion	65
7 Conclusion générale	66
Bibliographie	67

Table des figures

2.1	SNVI	12
2.2	organigramme de la S.N.V.I	17
2.3	organigramme de la filiale VIR	19
2.4	organigramme de la S.N.V.I.	21
2.5	corp de pont	22
2.6	le pont	24
3.1	Diagrammes de Gantt d'un ordonnancement.	30
3.2	Problème à machine unique	32
3.3	Problème à machine parallèles	32
3.4	Flow shop	33
3.5	job shop	34
4.1	Etapes de la modélisation	39
4.2	Diagramme de gantt associé à l'instance de problème $f_4/permu/c_{max}$	42
4.3	Diagramme de gantt associé à l'instance de problème f_4/c_{max}	43
4.4	Diagramme de Gantt associé à l'instance du problème $F2/permu/C_{max}$	44
4.5	Contrainte de conservation du temps	46
4.6	Fonction objectif	47
5.1	opérateur de croisement à un point	55
5.2	opérateur de croisement à deux points	56
5.3	échange de deux gènes quelconques	56
5.4	échange de deux gènes consécutifs	56
5.5	déplacement d'un gène	57
5.6	inversement d'une séquence de gènes	57
5.7	principe de fonctionnement d'un algorithme génétique	58
6.1	Logo python	60
6.2	bibliothèque et fonction utilisé	61
6.3	création des données dans un fichier excel	61
6.4	appelée le fichier excel	61
6.5	entrer les probabilités	62

6.6	les populations	62
6.7	resultat des individus	62
6.8	population optimale	63
6.9	c_{max} optimal	63
6.10	le temps d'exécution de l'algorithme	63

Liste des tableaux

2.1	Décomposition de la filiale VIR.	18
2.2	Les camions	18
2.3	Cars et Bus	18
2.4	Situation de la production de pont 01/01/2019 au 30/06/2020	23
2.5	ligne de montage d'un pont.	23
4.1	Exemple de $f_4/permu/c_{max}$	41
4.2	Exemple de $f_2/permu/c_{max}$	44
5.1	Exemple de $f_4/permu/c_{max}$	51
5.2	Sélection par rang pour un problème de maximisation.	53
6.1	résultas des testes effectués sur 4 taches et 33 machines.	64

Remerciements

Nous voudrions tout d'abord remercier ALLAH le Tout-Puissant et Miséricordieux, qui nous a donné le courage, la force et la patience pour la réalisation de ce mémoire.

*Nous tenons à exprimer notre profonde gratitude à madame **B.Ferdjallah** notre promotrice de mémoire, pour la confiance qu'elle nous a faite en acceptant de diriger nos recherches, et pour ses précieux conseils et orientations, ainsi que pour l'intérêt particulier qu'elle a accordé à ce travail. Nous la remercions pour sa grande contribution à l'aboutissement de ce travail.*

*nous remercions également à l'ensemble des enseignants du Département des Mathématiques et spécialement spécialité Recherche Opérationnelle sur tout Madame **W.Drici**.*

*Notre reconnaissance s'adresse particulièrement à madame **S.Rayasse** Cher de service programmation et coordination PR / direction Gestion Industrielle et Informatique SNVI filiale VIR-ROUIBA, de nous avoir accueilli et de nous avoir proposé ce thème.*

Un grand merci à nos très chers parents, qui nous ont aidé à suivre nos études dans les meilleures conditions et qui nous ont toujours soutenus et encouragés sans limite.

Nous remercions nos frères et soeurs pour leur encouragement, ainsi que toute la famille. Sans oublier de remercier aussi tous nos collègues, nos amies et tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces de mouloud

je dédie ce travail

À mes chers parents et mes amies

Dédicaces de oussama

je dédie ce travail

À tous qu'on aime.

1

Introduction générale

La recherche opérationnelle (R.O) est une discipline qui vise à résoudre par une démarche scientifique des problèmes de décision complexes issus du monde réel. Sa vocation donne et construit des modèles pour les problèmes généraux d'aide à la décision. En particulier les problèmes d'optimisation proposent des méthodes de résolution des problématiques définies sur ces modèles.

Le problème d'ordonnancement est classé parmi les problèmes fortement combinatoires, et il est toujours renouvelable, car jusqu'à maintenant, il n'existe aucune méthode de résolution générale et de faible complexité algorithmique.

Les problèmes d'ordonnancement sont très nombreux, ils se retrouvent systématiquement dans toute production de produits divers dont la réalisation est décomposée en de multiples tâches élémentaires. En fonction des objectifs. l'optimisation vise alors à établir l'ordre d'exécution des tâches afin de produire le maximum de produits de bonne qualité et à bas prix, dans des délais de plus en plus raccourcis.

L'organisation et la gestion de la production conditionnent le succès des projets du monde de l'entreprise et de la recherche. Dans ce processus, la fonction ordonnancement vise à organiser l'utilisation des ressources technologiques ou humaines pour répondre à une demande ou satisfaire un plan de production préparé par la fonction planification. Ainsi des programmes ambitieux privés ou publics ont recours à la fonction ordonnancement pour appréhender la complexité, améliorer les délais ou même s'adapter à des événements imprévus. Les problèmes d'ordonnancement apparaissent dans de nombreux domaines : l'industrie (atelier, gestion de production), la construction (suivi de projets), mais aussi l'informatique (gestion de processus) et l'administration (emploi de temps).

Dans ce mémoire, on s'intéresse aux problèmes d'ordonnement des ateliers, plus précisément au problème du FLOW SHOP de permutation, noté $F_m/permu/c_{max}$.

Ce problème étant NP difficile au sens fort, des solutions optimales en temps raisonnable ne sont pas garanties. Par conséquent, on fait appel à des méthodes dites approchées afin de trouver des solutions assez proches de l'optimum mais en un temps raisonnable.

Ce travail s'inscrit dans le cadre d'utilisation et d'hybridation de métaheuristiques pour la résolution de problèmes Flow Shop avec minimisation de makespan. Notre contribution consiste à l'adaptation, la programmation et l'hybridation de l'algorithme génétique (AG).

Le présent mémoire est organisé comme suit :

le premier chapitre est divisé en deux parties. Dans la première partie, nous allons donner une brève présentation de l'entreprise SNVI, et dans la deuxième partie, nous présentons la procédure de travail au sein du centre mécanique.

Dans **le deuxième chapitre**, nous abordons des définitions générales concernant la théorie d'ordonnement.

Dans **le troisième chapitre**, nous posons notre problématique et nous spécifions son objectif.

Dans **le quatrième chapitre**, nous modélisons le problème de l'atelier de préparation de la société, donc nous construisons un modèle mathématique pour notre problème et nous calculons sa complexité.

Finalement, un bilan et quelques perspectives de recherche de ce travail sont présentés en conclusion générale.

2

Présentation de l'entreprise

2.1 Présentation de la Société National des Véhicules Industriels « SNVI »



FIGURE 2.1: SNVI

L'Entreprise nationale des véhicules industriels (SNVI) est issue d'une restructuration, intervenue au début des années 1980 de l'ex-Société nationale de construction mécanique (SONACOME).

La vocation de l'Entreprise Nationale des Véhicules Industriels (SNVI) est la conception, la fabrication, la commercialisation et le service après-vente d'une importante

gamme de produits, à savoir : des camions et camions-tracteurs, autocars, des autobus et des équipements de carrosserie industrielle.

La Société nationale des véhicules industriels (SNVI) a réussi dès sa création à s'imposer en tant que leader régional dans sa spécialité et à bâtir un label grâce à ses produits de haute qualité et un service après-vente performant.

A l'instar des autres entreprises nationales, la SNVI a été affectée par la décennie noire, période durant laquelle elle avait enregistré une récession de l'activité et une chute des ventes sans précédent. Les unités situées dans les régions à forte activité terroriste ont été le théâtre d'actes de sabotage et de vandalisme.

Soucieux de la sécurité de leur personnel, les responsables de la société étaient contraints de fermer provisoirement les unités enclavées. Cette situation d'insécurité porta un coup fatal à la santé financière de la SNVI. En ce sens que le niveau de la production annuelle de l'entreprise de l'exercice 1994 a chuté de 4000 véhicule comparativement à celui de 1981 qui était de 6.200 véhicules.

A partir de 2010 la SNVI a bénéficié d'un ambitieux plan d'investissement destiné à renforcer sa compétitivité et à consolider ses parts de marché.

Dans le cadre du défi que se sont lancés les pouvoirs publics pour répondre à la forte demande locale et réduire le coût des importations de voitures, sans cesse croissantes, la SNVI est appelée à contribuer au lancement d'une construction automobile en Algérie.

A ce titre l'Etat l'a chargé depuis quelques années de mener des pourparlers «serrés» avec des leaders mondiaux de l'industrie automobile, à l'instar de l'allemand Volkswagen et du français Renault, tous deux intéressés par l'implantation d'usines en Algérie.

La filiale Véhicule industrielle de Rouiba regroupe cinq centres de production :

- Forge : obtention des bruts par déformation plastique à chaud.
- Mécanique : produit des ponts, des essieux, des directions et des pièces de liaisons.
- Tolerie et Emboutissage : produit des longerons pour cadres chassis, des cabines et des pièces de liaisons.
- Montage camions : assemble les camions.
- Montage autocars et autobus : produit les caisses, les trielles et assemble les cars bus et produit également des pièces en polystyrène et siège.

Le but de notre mémoire est de proposer un planning qui optimise la durée totale des tâches pour trouver la durée d'exécution minimum de fabrication d'un pont dans le centre mécanique pour réaliser le programme, dans ce centre sont usinées à partir des barres laminées, des bruts de forges, bruts de fonderie et des organes (ponts, essieux, boîtes de vitesses, boîtes de directions...). Tous ces organes sont assemblés après usinage dans ce centre.

2.2 Historique de la SNVI

Le cycle d'évolution de l'industrie mécanique en Algérie est marqué par quatre grandes phases essentielles :

2.2.1 Ancienne Organisation

1957 à 1967 :

La création de la société française BERLIET le 02 juin 1957 sur le territoire algérien par la construction en juin 1957, d'une usine de montage de véhicules « poids lourds à 30 km » à l'Est d'Alger.

1967 – 1981 :

Après 1962, l'industrie mécanique en Algérie était au stade embryonnaire, se limitant aux besoins de l'époque. En 1967, fut créée la SONACOME regroupant en son sein 11 entreprises, qui deviennent autonomes à partir de 1980.

1981-1995 :

Suite au bilan dressé par les autorités publiques du pays sur l'état de fonctionnement de l'économie nationale, une circulaire présidentielle N13 du 20 novembre 1980, fixe le cadre réglementaire organisant les opérations de restructuration des entreprises publiques.

La S.N.V.I est née à l'issue de la restructuration de la SONACOME, et le décret de sa création lui consacra, un statut d'Entreprise Socialiste à caractère économique, régie par les principes directeurs de la gestion socialiste des entreprises (G.S.E) en vigueur au plan institutionnel.

1995-2000 :

La S.N.V.I, Entreprise Publique Economique.
Dès mai 1995, la S.N.V.I change de statut juridique pour devenir une EPE régie par le droit commun : la S.N.V.I est alors érigée en Société Par Actions, au capital social 2,2 milliards de dinars.

2.2.2 Nouvelle Réorganisation du Holding

Le 03 juin 2000 :

le Holding Public Mécanique devient HOMELEC (Holding Mécanique et Electronique), holding constitué de regroupement de deux holdings mécanique, électronique.

La S.N.V.I Entreprise Nationale de Véhicules Industriels est chargée dans le cadre du plan national de développement économique et social de la recherche, du développement de la production de l'exportation, de la distribution et de la maintenance des véhicules industriels.

En 2002 :

La SNVI a pris le contrôle de l'entreprise publique «FONDAL» Société Par Action (SPA), depuis Août 2002, et ce conformément aux résolutions arrêtées par l'Assemblée Générale des Holdings Publics en liquidation.

À Partir de 2011 :

Signature le 22 Mars 2011 d'un protocole d'accord de partenariat entre le ministère de la défense nationale, la société Nationale des véhicules Industriels (SNVI), le Fond d'investissements Emirati Aabar et le groupe Allemand Daimler AG (Mercedes - Benz).

Cet accord, qui entre dans le cadre du renforcement des relations de coopération Algéro-Emirat-Allemande dans le domaine de l'industrie militaire, porte sur la modernisation et l'extension de la plate-forme de production de véhicules industriels située à Rouiba.

L'objectif de ce partenariat se focalise notamment sur la modernisation et l'extension des centres de production de Camions et Autobus de la division Véhicules Industriels de Rouiba.

2.3 Mission, Objectif et Organisation de L'entreprise

2.3.1 Mission de la « SNVI »

La S.N.V.I, Entreprise Nationale de véhicules industriels est chargée dans le cadre du plan national de développement économique et social de la recherche, du développement, de la production, de l'exportation, de la distribution et de la maintenance des véhicules industriels. Par véhicules industriels, il faut entendre :

- Les camions.
- Les autobus.
- Les autocars.
- Les camions spéciaux.
- Les remorques et semi-remorques.

Et d'une manière générale, tous les véhicules destinés au transport routier des personnes, des biens et des matières d'une charge utile supérieure à 1,5 tonnes.

D'autre part, la SNVI est chargée d'assurer et de promouvoir les activités d'après-vente des véhicules industriels et d'assister également les gros utilisateurs de ses produits dans la mise en place de leurs propres moyens de maintenance.

On peut retenir trois caractéristiques à la SNVI :

- La S.N.V.I est une entreprise très **Spécialisée et Verticalement Intégrée**.
- La S.N.V.I dispose d'une grande **Variété de Produits**.
- La S.N.V.I est présente sur **L'ensemble du Territoire National** d'où une intégration économique appréciable.

2.3.2 Objectifs de la « SNVI »

1-Satisfaire Les Besoins Nationaux en Véhicules Industriels.

- Maximiser la production.
- Acquérir rapidement une gamme de technologie (taux d'intégration élevé, grand nombre de produits).
- Rationaliser l'emploi.
- Utiliser les techniques performantes et adaptées.

2-Comblent l'écart Entre la Production et la Demande.

- Commercialiser les véhicules industriels fabriqués localement.
- Assurer la disponibilité de la pièce de rechange de la gamme SNVI.
- Assurer le service après-vente.

3-Minimiser le Prix des Produits et de Service en Assurant sa Rentabilité.

- Minimiser le prix de revient.
- Financer partiellement le développement de l'entreprise par la commercialisation.

4-Formation Massive des Hommes.

5-Contribuer au Progrès économique et Social.

- Assurer l'implantation industrielle et commerciale sur l'ensemble du territoire.
- Assurer la disponibilité des produits sur l'ensemble du territoire au prix uniforme.

2.3.3 Organisation de La « SNVI »

La S.N.V.I adopte le modèle organisationnel uniforme structuré en cohérence parfaite avec les objectifs du contrat de performance de la manière suivante :

- Une assemblée des travailleurs soutenue par des commissions permanentes :
- Un conseil de surveillance ;
- Un Directeur Général Unique de l'Entreprise épaulé par des Cadres Dirigeants ; des Directeurs d'unités (commerciales, production et Prestation de Services).

Pour accomplir ses différentes missions, la S.N.V.I a développée toute une variété d'activités qu'on peut regrouper en deux grandes catégories homogènes : les activités opérationnelles et les activités fonctionnelles, qui sont des activités de la production et sont sous la responsabilité de la Direction Industrielle.

Les unités de production sont :

- Véhicules Industriel de Rouïba (**V I R**).
- Carrosserie Industrielle Rouïba (**C I R**).
- Fonderie de Rouïba (**F O R**).

ORGANIGRAMME ACTUEL DE LA S.N.V.I

organigramme de la S N V I l'organigramme ci-dessus présente l'ensemble de l'organisation de la société :

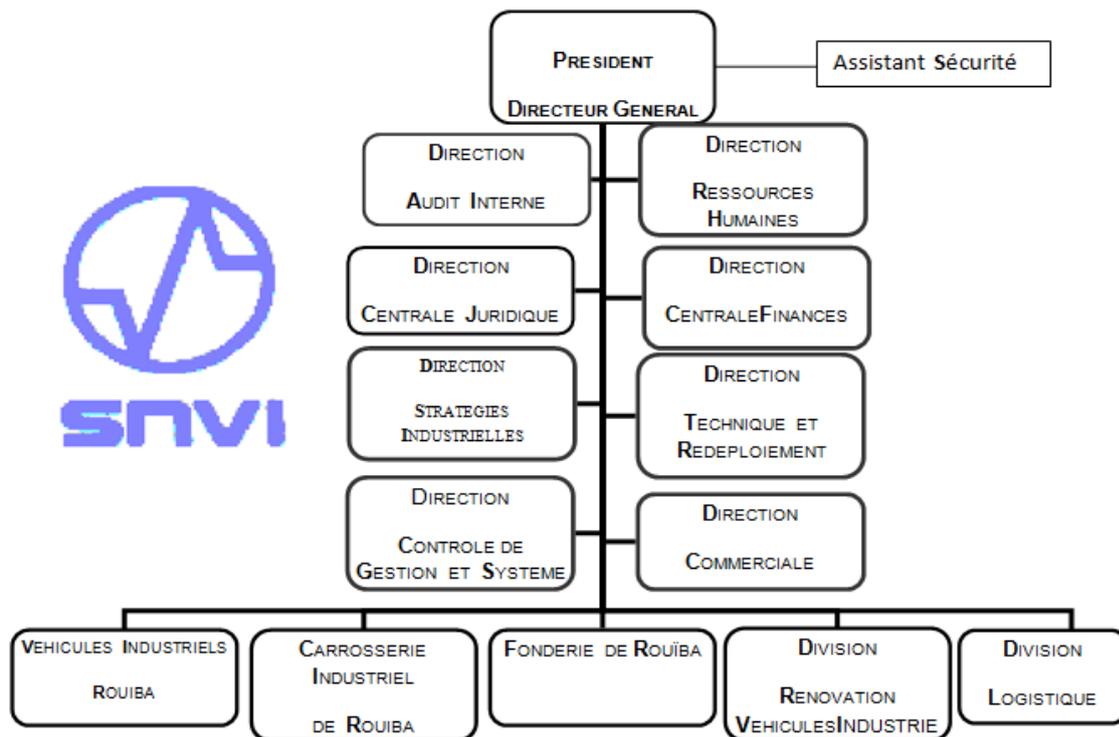


FIGURE 2.2: organigramme de la S.N.V.I

1. La Filiale Fonderie de Rouïba « FOR »

Aura pour principales missions, la fabrication de pièces de fonderie destinées à l'industrie de la mécanique en général et de l'automobile en particulier.

Le plan de charge de la fonderie au-delà de ses engagements en interne aura à répondre à la satisfaction aux besoins en bruts des différents projets et particulièrement du projet moteurs d'Oued Hamimine (Constantine).

2. La Filiale Carrosseries Industrielles de Rouïba « CIR »

Aura pour mission, l'étude, la conception, la fabrication et l'assemblage d'équipement (citernes, plateaux, bennes etc... et équipements spéciaux) pour les véhicules de la gamme SNVI et pour les futures partenaires de l'entreprise.

Une partie de ses ateliers sera consacrée à la fabrication de pièces de rechange nécessaire à l'après-vente et à une activité de rénovation intégrée.

Les filiales, devenues autonomes, disposent, par le statut de SPA, d'une assemblée générale et d'un conseil d'administration à travers lequel l'état unique actionnaire et propriétaire, reste parfaitement en contact avec les dirigeants des filiales et veille à ce que ses intérêts soient prioritaires étant donné que sa principale mission se trouve être de fructifier le portefeuille d'actions de l'état.

3. La Filiale Véhicule Industrielle de Rouïba « VIR »

Aura pour mission, l'étude, la conception, la fabrication et l'assemblage de véhicules industriels motorisés destinés au transport routier. Une importante partie de ses activités sera consacrée à la fabrication de pièces de rechange des véhicules de sa gamme et destinées à la satisfaction des besoins du réseau après-vente et aux activités de rénovation. Nous avons effectué notre stage pratique au sein de la Filiale VIR. A ce titre nous allons présenter cette filiale comme suit :

Elle est composée de 5 centres de production qui sont représentés dans le tableau ci-dessous :

Mécanique (ME)	Forge (FO)	Tolerie Emboutissage (TE)	Montage Camions (MC)	Montage Auto-bus (MA)
Usinage des pièces mécaniques, traitement thermique, essieux, ponts, boîtes à vitesses, direction.	Brut des pièces forgées.	Longerons, cabines camions, réservoir, toles embouties.	Pièces en polyester, montage des camions.	Les siège, pièces en tube, montage des cars, minicars, bus et minibus.

TABLE 2.1: Décomposition de la filiale VIR.

Produit de La Filiale VIR :

Camions	Porteur	Tracteurs	Chantiers	Militaire
Gamme	K 66, K 120, B 260, B 400 (6*4)	TB 400 (6*4), TB 400 (4*2)	C 260 (4*2), C 260 (6*4)	M 120 (4*4), M 230 (6*6)

TABLE 2.2: Les camions

Cars et Bus	Cars	Bus	Minicars
Gamme	SAFIR	100V8 NC, 100 L6	18L4, 25L4

TABLE 2.3: Cars et Bus

Organigramme de La filiale VIR

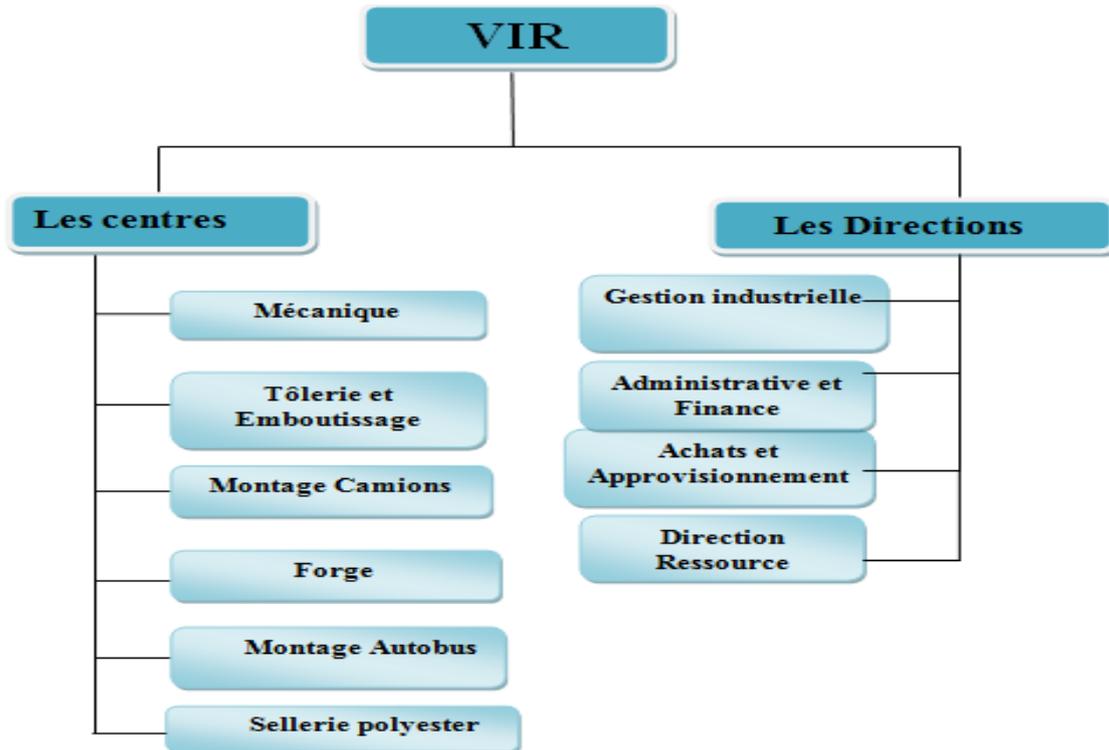


FIGURE 2.3: organigramme de la filiale VIR

2.4 Clients

- Export : pays arabes et afrique.
- Publics : entreprises publics.
- privé : entreprises privé et particuliers.
- Les institus nationales : militaire et les APC.

2.5 Drections Usinage

2.5.1 Batiment Mécanique

Surface : 43.000 méttre carré.

Effectife : 856 personnes dont 26 cadres.

Dans ce Batiment sont usinés des :

- Ponts.
- Essieux.
- Boites de vitesses.
- Cardes de derections.
- Pièces diverses (ferrures pour cadre chassis, pédaliers pourcabines,leviers de vitesses,tiges d'accélération,ect...).

2.5.2 Service Méthode

Ce service est l'intermédiaire entre les bureaux d'étude et la fabrication ses tâches principales sont les suivantes :

- Réalisations des gammes d'usinage.
- Définition des moyens humains et matériels à mettre en place.
- Dessins des plans.
- Gestion des magasins outillage.

ce service contient 6 secteurs :

1. AMTAX.
2. Outils coupant.
3. Essieux et ponts.
4. Boîtiers ferrure.
5. Etudes.
6. DG ENG TTG.

Secteur Ponts et Essieux

Contient 3 modes :

- a. 3050 cuivres de ponts et corps d'essieux.
- b. 3060 les pièces de pont et d'essieux.
- c. 3070 montages ponts et essieux.

2.5.3 Service Production

Ce service est divisé en deux parties :

A-Usinage : tournage, taillage, traitement.

B-Montage : boîte à vitesse, pont, essieux.

2.5.4 Service Maintenance

Il est chargé de :

- garantir le bon état de toutes les machines de centre.
- suivi et entretien des matériels existants.
- prévoir toutes les pièces de rechange susceptibles d'être nécessaires à la casse.

2.5.5 Service Ordonnancement

Le but principal de l'Ordonnancement est de définir un planning de travail pour les ateliers de telle sorte que les dates planifiées de mise à disposition des produits soient respectées.

Il assure la coordination et les orientations entre les différents services qui existent dans chaque Centre.

Il a un rôle important, car il s'occupe de la planification des activités à court terme des ateliers. Il coordonne les moyens nécessaires à la réalisation du plan de production et définit l'ordre de passage des différentes séries à fabriquer sur les différentes machines. Les services Ordonnements Centres sont aussi responsables de la gestion des pièces achetées ou fabriquées nécessaires à la réalisation des objectifs fixés dans les normes et délais.

Le service est composé de trois secteurs :

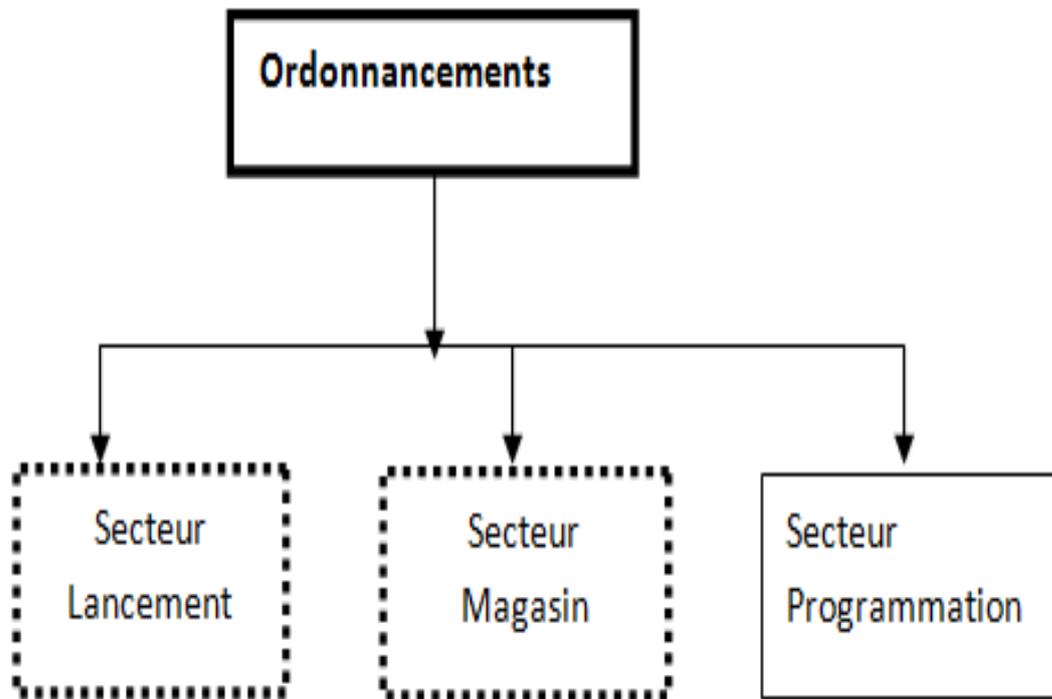


FIGURE 2.4: organigramme de la S.N.V.I.

1- Secteur Lancement :

Le Secteur Lancement est Chargé de faire les expressions des besoins (gestion du stock) ainsi que des prévisions à long et court terme. Il assure les tâches suivantes :

- L'expression des besoins à partir d'un programme déterminé
- La gestion des pièces fabriquées et achetées.

2- Secteur Magasin :

Il stock des pièces à partir d'un bordereau de réception. Les stocks de la direction sont placés dans un magasin afin de les ranger entre leur réception et leur mise à disposition.

Le secteur magasin est composé de plusieurs Magasins organisés comme suit :

Saisie du tableau Magasins.

Toutes les Entrées et sorties des flux sont transites par les fichiers.

3- Secteur programmation et Montage :

Programmation des véhicules sur la chaîne du montage , Alimentation de tous les postes.

A partir d'un programme d'engagement annuel ou mensuel, le secteur effectue des programmes journaliers d'engagement en fonction de l'état de stock (stock organes, stock pièces maîtresses) et des données des Méthodes.

2.5.6 Généralité sur le pont

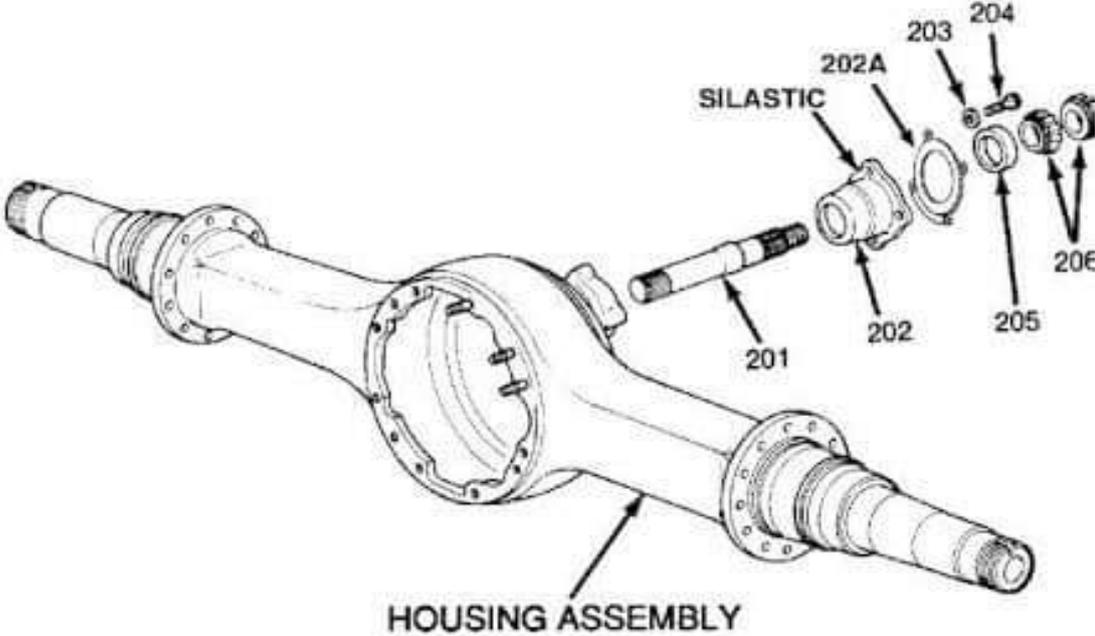


FIGURE 2.5: corp de pont

véhicule	organes	reference	jan	fev	mar	av	mai	jui	total
K66-25L4-18-L4	E2A	1095896	20	60	50	50	30	50	260
K120	E4A21	1096651	4	20	20	30	20	20	114
B260-C260-TB400-B400	E6A211	5600104912	0	0	0	0	0	34	34

TABLE 2.4: Situation de la production de pont 01/01/2019 au 30/06/2020

Section	Taches	Temps(mn)
127	mise en main	
127	lavage	12.00
127	montage moyeu	6.60
127	montage fusée	12.00
127	montage tamboure	6.60
127	peinture	12.00

TABLE 2.5: ligne de montage d'un pont.

2.5.7 Les différents opérations pour le montage de pont

- M_i : les machines, $i \in \{1, \dots, 33\}$
- a_j : les opérations pour la fabrication de corp de pont(A), $j \in \{1, \dots, 8\}$
- b_k : les opérations pour la fabrication de tambour(B), $k \in \{1, \dots, 11\}$
- c_l : les opérations pour la fabrication de moyau (C), $l \in \{1, \dots, 11\}$
- d_m : les opérations pour la fabrication de la fusée(D), $m \in \{1, \dots, 16\}$

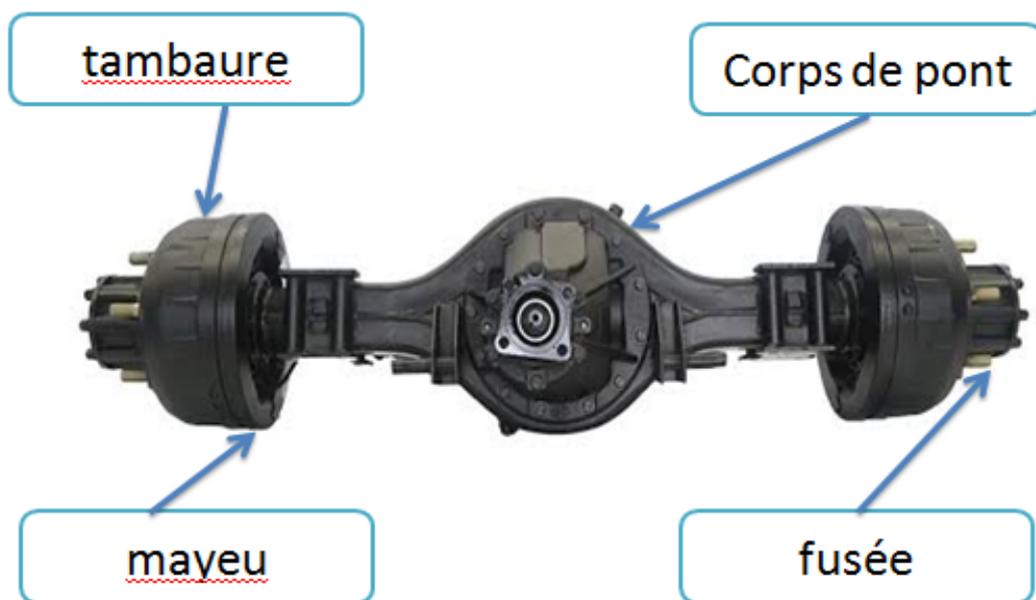


FIGURE 2.6: le pont

2.6 Conclusion

L'entreprise SNVI est une organisation et une unité économique qui réunit plusieurs services et moyens pour produire de biens ou services destinés à la vente sur le marché afin de réaliser ses objectifs.

Dans un système de production, le problème de l'ordonnancement consiste à organiser dans le temps l'exécution d'opérations interdépendantes à l'aide de ressources disponibles en quantités limitées pour réaliser un plan de production.

La fonction "ordonnancement" étant l'objet principal de notre étude, elle est abordée plus en détail dans le deuxième chapitre.

3

Généralités sur l'ordonnancement

3.1 Introduction

L'ordonnancement est une branche de la recherche opérationnelle et de la gestion de la production qui vise à améliorer l'efficacité d'une entreprise en terme de coût de production et de délais de livraison[20].

Les problèmes d'ordonnancement sont présents dans tous les secteurs d'activité de l'économie[4].

Dans cette partie nous présenterons les éléments essentiels qui introduisent et préparent la formulation du problème étudié. Nous commençons par introduire les problèmes pour lancement ainsi que les notations utilisées dans la suite de ce mémoire. Le plan de charge permet de vérifier si la charge occasionnée dans l'atelier par les commandes n'est pas supérieur à la capacité des ressources de l'atelier.

Pour le cas contraire des réajustements des charges (lissage) ou de capacité (soustraction, heure ou équipe supplémentaire) peuvent être fait.

Les données sont transmises à la fonction ordonnancement que nous détaillons dans cette partie.

3.2 Qu'est-ce que l'ordonnancement ?

3.2.1 Définition :

L'ordonnancement est la programmation dans le temps de l'exécution d'une série de tâches (ou activités, opérations) sur un ensemble de ressources physiques[7], en recherchant à optimiser certains critères, financière ou technologique, et en respectant les contraintes de fabrication et d'organisation[26][10].

Les ordres de fabrication (OF) suggéré par le calcul de besoin représentant chacun une requête pour fabriquer une quantité déterminée de pièces pour une date donnée.

Ils constituent les données d'entrée de l'ordonnancement en permettent de définir au moyen des gammes de fabrication, l'ensemble des tâches que la fonction ordonnancement doit planifier.

une tâche est localisée pour le temps une date de début et une date de fin. Elle utilise

une ou plusieurs ressources. Elle est dite primitive ici elle peut être interrompu ou non primitive si elle ne peut pas être interrompue.

On sortie de la fonction d'ordonnancement on obtient un planning ou ordonnancement qui restitue l'affectation des tâches fournies en entrée à des dates précises pour des durées déterminées sur les différentes ressources. Ce planning cherche à satisfaire des objectifs en respectant le plus possible de contraintes que nous allons préciser.

La fonction d'ordonnancement est une fonction à court terme, même si elle est parfois utilisée à moyen terme [10] son horizon et sa période sont donc relativement courts. L'horizon d'un ordonnancement est en général d'une à trois semaines la période vrai de 2 ou 3 jours à une semaine.

3.3 Les données d'un problème d'ordonnancement

Les problèmes d'ordonnancement apparaissent dans tous les domaines : informatique, industrie, construction, administration, etc[6].

Les différentes données d'un problème d'ordonnancement sont les tâches, les ressources, les contraintes et les critères.

Ainsi étant donné un ensemble de tâches et un ensemble de ressources, il s'agit de programmer les pages et affecter les ressources de façon à optimiser un ou plusieurs objectifs (un objectif correspondant à un critère de performance), on respecte ton un ensemble de contraintes[2].

3.3.1 Les tâches

Une passe est une entité élémentaire organisée dans le temps, par une date de début ou de fin, et dont la réalisation nécessite une durée préalable définie. Elle est constitué d'un ensemble d'opérations qui requiert pour son exécution certaines ressources et qu'il est nécessaire de programmer de façon à optimiser un certain objectif[24]. On distingue deux types de tâches :

Les tâches morcelables (préemptives) qui peuvent être exécuté en plusieurs fois facilitant ainsi la résolution de certains problèmes.

Les tâches non morcelables (indivisibles) qui doivent être exécutés en une seule fois et ne sont interrompus qu'une fois terminé.

Les notations suivantes sont introduites pour décrire une opération. Chaque variable représentera une information particulière se rapportant à la réalisation de l'opération :

- r_i : pour représenter la date de disponibilité de la tâche i ;
- t_i : pour représenter la date de début de la tache i ;
- c_i : pour représenter la date de fin d'exécution de la tâche i ;
- d_i : pour représenter la date d'échéance de la tâche i ;
- p_i : pour représenter la durée opératoire de la tâche i ;
- $F_i = c_i - r_i$ pour représenter la durée de séjour de l'opération i sur la machine avant qu'elle redevienne disponible ;
- $L_i = c_i - d_i$ exprime le retard algébrique entre la fin d'exécution et la tache i par rapport à sa date d'échéance ;

- $T_i = \max(L_i, 0)$ qui exprime le retard absolu de la tâche i .
- $E_i = \max(-L_i, 0)$ qui exprime l'avancement de la tâche i .

3.3.2 Les ressources

Une ressource est un moyen technique ou humain utilisé pour réaliser une tâche. On trouve plusieurs types des ressources[29] :

Les ressources renouvelables : qui après avoir été à louer à une tâche redevienne disponible (machine personnel. ... etc.).

Les ressources consommables : qui l'ours qu'après sa libération elle n'est pas disponible en même quantité (argent matière première...). Dans le cas des ressources renouvelables on distingue principalement les ressources disjonctives qui ne peuvent exécuter qu'une passe à la fois et les ressources cumulatives qui peuvent être utilisées par plusieurs pages simultanément mais en nombre limité.

3.3.3 Les contraintes :

Une contrainte est une restriction sur les valeurs qui peuvent prendre une ou plusieurs variable de décision sur le temps (variable ordonnancement) ou bien sur les ressources (variable affectation)[28].

Selon [5] les contraintes aux quelles sont soumises les diverses tâches qui concourent à la réalisation du projet sont de divers types. On distingue :

Les contraintes potentielles : Les contraintes de localisation temporelle : qui permettent de localiser dans le temps chacune des tâches concerné (tâche j doit être achevé à telle date, ou son exécution ne doit commencer avant telle date)[21].

Les contraintes d'intériorité : qui implique une tâche donnée ne peuvent commencer à être exécuté que si on finit d'exécuter un sous-ensemble de tâches dite prédécesseurs de cette tâche.

Les contraintes disjonctives :

une contrainte disjonctive impose la non réalisation simultané de deux tâches a ou b . On trouve de telles contraintes dans le cas d'utilisation d'une ressource présente en un seul exemplaire (une grue une équipe...). Ou pour formuler des interdiction de réalisation simultané pour des raisons de sécurité ou des problèmes de place. Arbitrer une contrainte disjonctive consiste à décider si A sera fait avant B ou l'inverse [12]

Les contraintes cumulatives :

ce type de contraintes limite les possibilités d'ordonnancement. Elles sont relatives à l'évolution au cours du temps de la disponibilité des moyens de réalisation des différentes tâches.

3.3.4 Les critères :

pour compléter la construction d'un problème d'ordonnancement l'objectif peut-être d'optimiser un critère (la durée totale de l'ordonnancement par exemple). Les critères d'optimisation consistent à minimiser ou bien maximiser une fonction objectif trouve :

- C_{max} : date de fin de traitement de l'ensemble des tâches (Makespan).
- $\sum C_i$: la somme des date de fin de traitement.
- $\sum W_i C_i$: la somme pondérée des date de fin de traitement.
- D_{max} : le grand retard.
- L_{max} : la grande tardiveté.
- $\sum U_i$: le nombre de taches en retard Dans ce document le critère étudié est le makespan.

3.3.5 L'objectif de l'ordonnancement :

Le traitement de l'ordonnancement dans la littérature c'est tout d'abord orienté vers une optimisation Monocritère. L'environnement manufacturier évoluant rapidement de la concurrence devenant de plus en plus acharné, les objectifs des entreprises se sont diversifiés et le processus d'ordonnancement est devenu de plus en plus multicritère.

Les critères que doit satisfaire un ordonnancement sans variés. D'une manière générale on distingue plusieurs classes d'objectif concernant un ordonnancement[10] :

- **Les objectifs liés au temps** : on trouve par exemple la minimisation du temps total d'exécution. Du temps moyen d'achèvement, des durée totale des réglage ou des retards par rapport aux dates de livraison.
- **Les objectifs liés aux ressources** : maximiser la charge d'une ressource ou minimiser le nombre de ressources nécessaires pour réaliser un ensemble de tâches sont des objectifs de ce type.
- **Les objectif lié au cout** : ses objectifs sont généralement de minimiser les coûts de lancement de production de stockage de transport. ...
- **Les objectifs liés à une énergie ou un débit** : Les objectifs à satisfaire à niveau d'ordonnancement sont issus des objectifs globaux de l'entreprise par décomposition. Cette décomposition conduit à une structure d'objectifs qui permet de gérer la contradiction et les compromis [11] et [13].

3.4 Diagramme de Gantt

Un ordonnancement est représenté à l'aide de diagramme de Gantt, ceux-ci développés en 1910 par l'ingénieur en mécanique et consultant en management Henry Laurence Gantt à destination de l'ordonnancement de projet puis adapté à l'ordonnancement de la production en 1917.

Ce sont des diagrammes à barrer où les tâches sont représentées par des segments de longueur proportionnelle à leur durée. En ordonnancement de projet chaque activité est représentée sur une ligne qui lui est propre alors qu'en ordonnancement de production une ligne correspond à une ressource[15].

La figure suivante représente un exemple d'un diagramme de Gantt avec trois machines et 5 tâches[19].

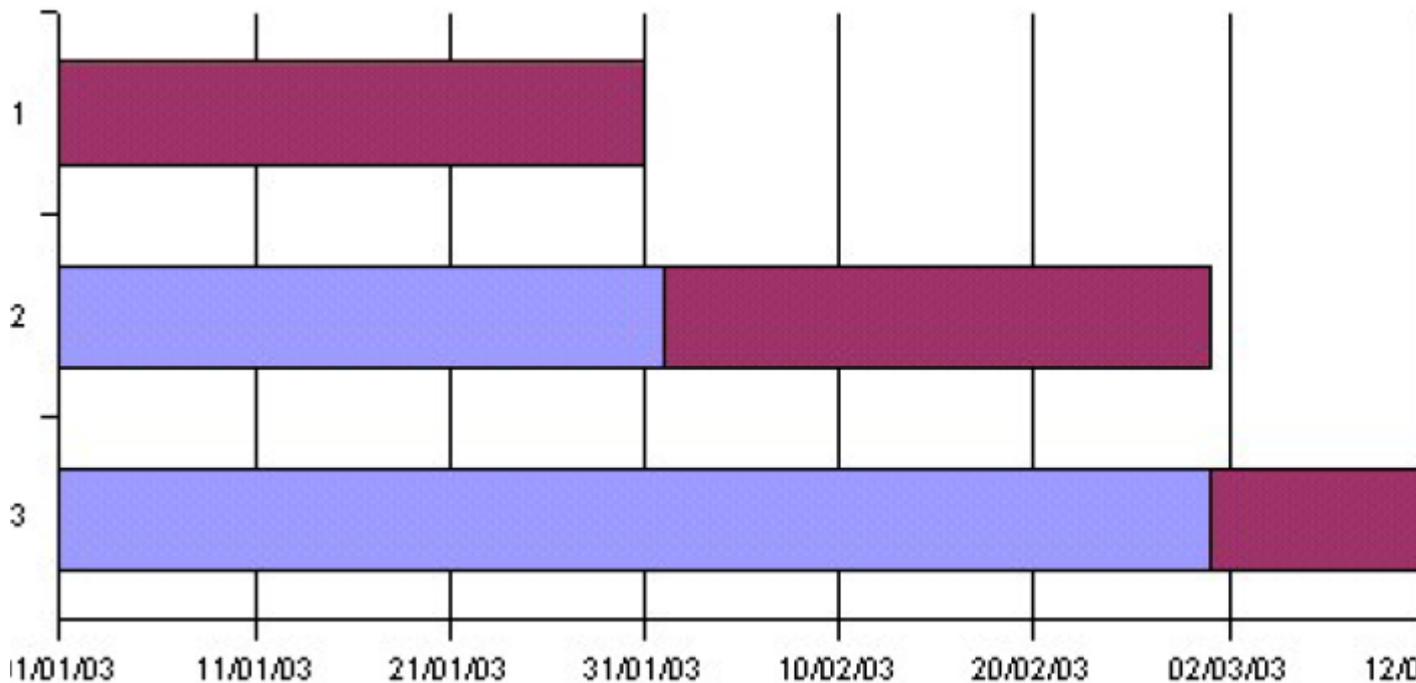


FIGURE 3.1: Diagrammes de Gantt d'un ordonnancement.

3.5 Les types d'ordonnancement :

On distingue deux types d'ordonnancement[22] :

3.5.1 Ordonnancement simple :

pour réaliser un travail complexe on doit avoir mené à bien un certain nombre de tâches dont les périodes d'exécution ne sont pas indépendante on peut effectuer les tâches en parallèle mais certaines entre elles doivent être achevé avant que les autres commencent. Le problème est alors de savoir quand on peut et quand on doit commencer les tâches pour achever l'ensemble de travail [8].

3.5.2 L'ordonnancement de travaux dans un atelier non simple :

Ce problème peut être présenté de la manière suivante : soit un atelier comportant m machines et dans la quelle n les travaux doivent être effectués. La réalisation de chaque travail nécessite le passage sur un sous-ensemble de machine dans un ordre prescrit et avec une durée d'immobilisation de la machine (pour la machine M_j la durée d'immobilisation due au traitement de travail T_i durée pendant la quelle la machine est disponible pour le traitement d'autre travaux peut dépendre de ce travail T_i). la séquence de passage de chaque travail sur l'ensemble des machines sur laquelle ce travail doit être traité et supposée fixe. Le problème consiste à déterminer un ordre de traitement des travaux qui minimise la durée totale de traitements[18].

3.6 Caractérisation d'atelier :

Dans la littérature classique les problèmes d'ordonnancement d'ateliers sont classés par famille de problème, selon la structure d'atelier on distingue [3] :

3.6.1 Problème à machine unique :

Dans ce cas, l'ensemble des tâches à réaliser est fait par une seule machine. Les tâches alors sont composées d'une seule opération qui nécessite la même machine. L'une des situations intéressantes où on peut rencontrer ce genre de configurations est le cas où on est devant un système de production comprenant une machine goulot qui influence l'ensemble du processus. L'étude peut alors être restreinte à l'étude de cette machine[17].

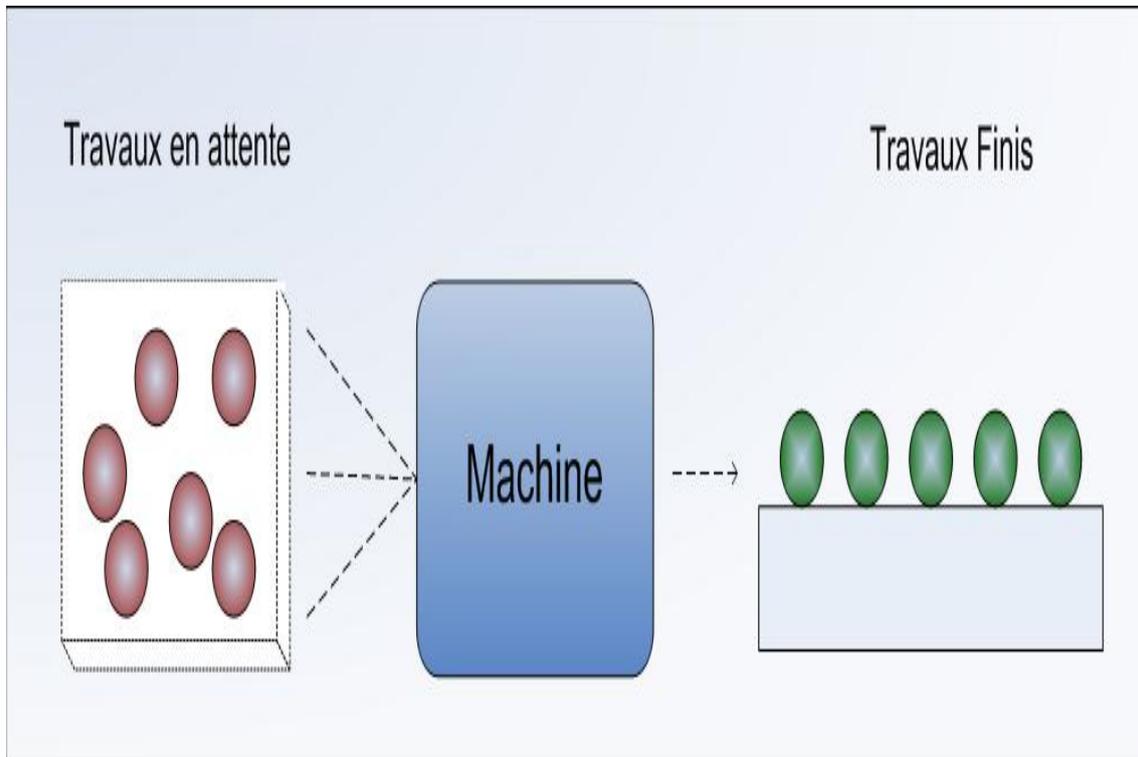


FIGURE 3.2: Problème à machine unique

3.6.2 Problème à machine parallèles

Dans ce cas, on dispose d'un ensemble de machines identiques pour réaliser les travaux. Les travaux se composent d'une seule opération et un travail exige une seule machine. L'ordonnancement s'effectue en deux phases : la première phase consiste à affecter les travaux aux machines et la deuxième phase consiste à établir la séquence de réalisation sur chaque machine

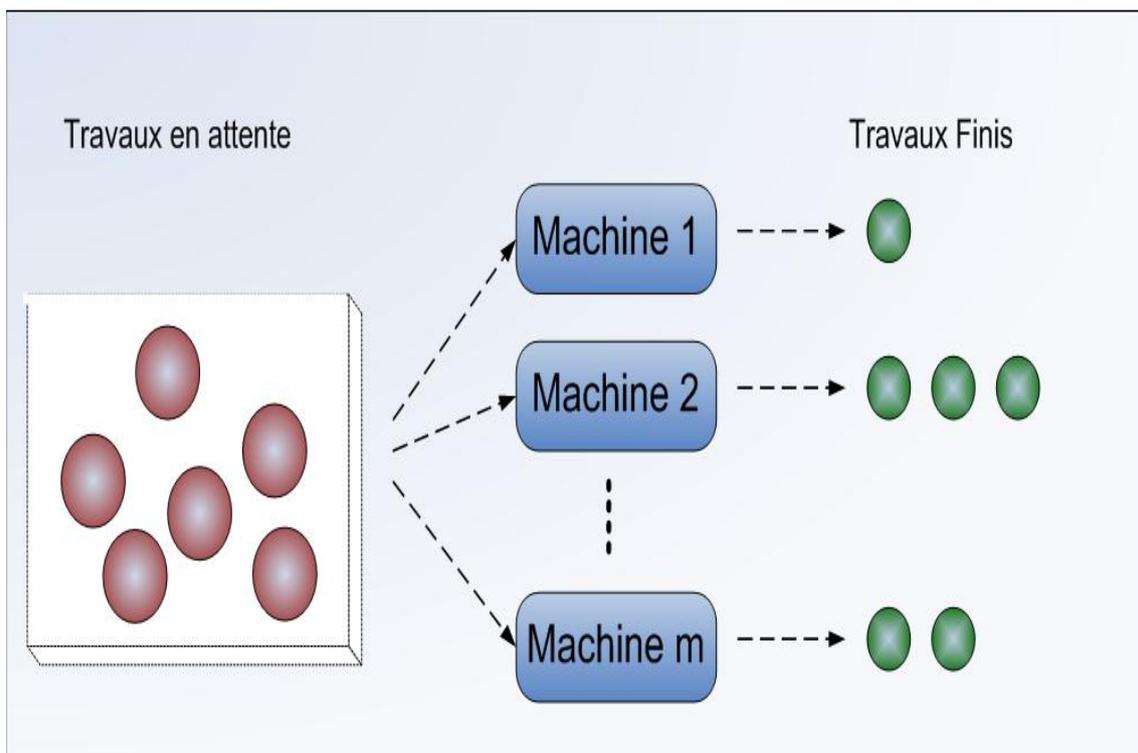


FIGURE 3.3: Problème à machine parallèles

Nous distinguons 3 types de machines parallèles :

- **Le problème à Machine identique** : pour ces machines les vitesses de traitement des tâches sont toute égales.
- **Le problème à machine uniformes** : chaque machine a sa propre vitesse qui ne dépend pas de la tâche exécuté.
- **Le problème à machine générales** : contrairement aux deux cas précédents la vitesse d'exécution dépend de la machine et de la tâche exécutée.

3.6.3 Problème à machine Spécialisée

Dans ce cas nous distinguons trois systèmes de traitement :

Le problème d'atelier à cheminement unique (flow shop) : Dans ce système les tâches doivent être traitées par toute les autres machines dans le même ordre [16].

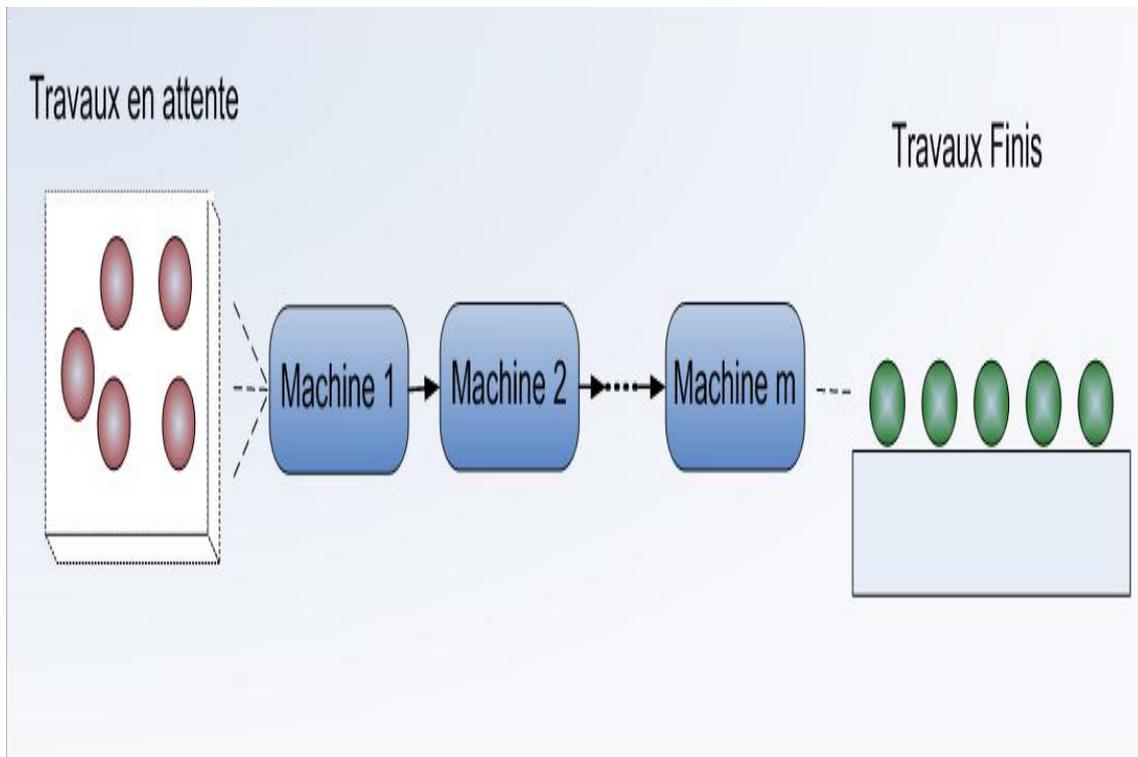


FIGURE 3.4: Flow shop

Le problème l'atelier à cheminement multiple (job shop) : L'ordre du traitement est arbitraire mais il doit être spécifié à l'avance[1].

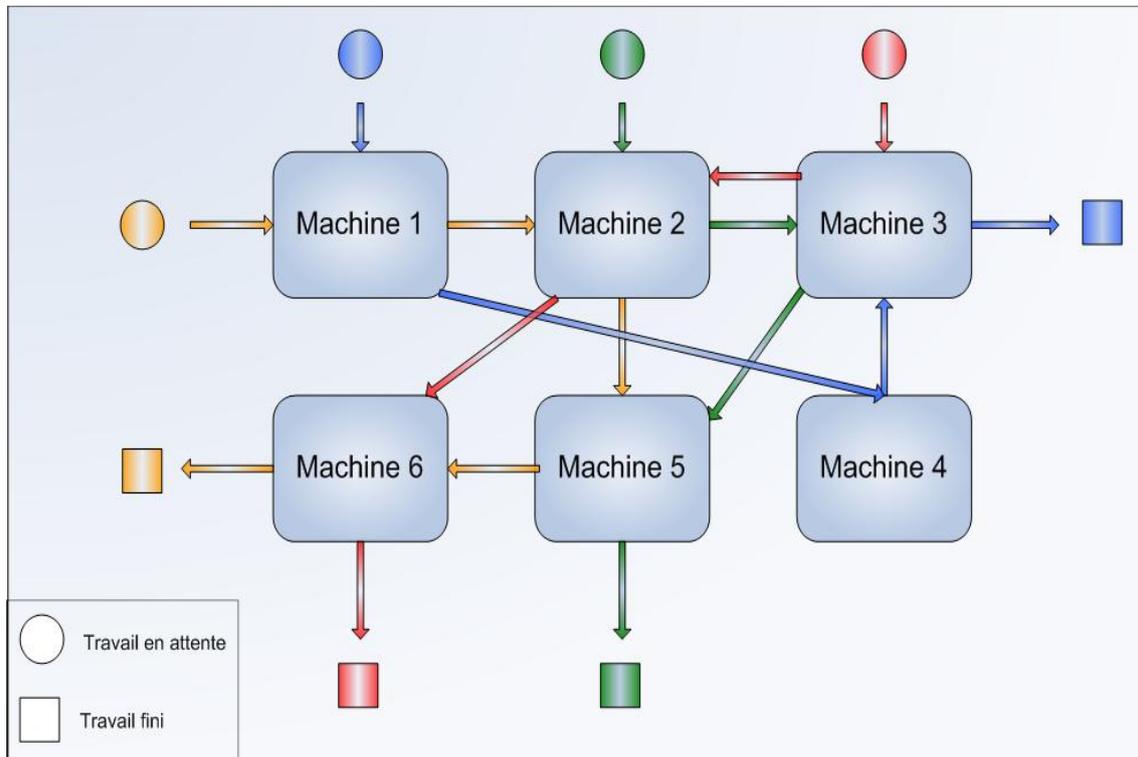


FIGURE 3.5: job shop

Le problème d'atelier à cheminement quelconque (Open shop) : Dans un système open shop les tâches doivent être traitées par toutes les machines mais il n'existe aucun ordre de traitement pour les tâches.

3.7 Classification des problèmes d'ordonnancement

Pour des raisons de compréhension et que les choses soient claires. Il est important d'adopter un descripteur qui soit basé sur l'existant et homogénéisant les notions actuelles.

La notion $\alpha/\beta/\gamma$ introduite par Graham et Al [14] est couramment utilisée pour distinguer les différents problèmes d'ordonnancement entre eux et les classifier.

3.7.1 Le champ α

Renseigne les informations relatives aux ressources à savoir le nombre de machine, type de machine et type d'atelier. Il est composé de deux sous champs et désigné par $\alpha_1 \alpha_2$ telle que :

- $\alpha_1 \in \{IPQRFJ\}$.
- I : Machine unique.
- P : machines parallèles identiques.
- Q : Machine parallèle uniforme.
- R : machines parallèles générales.
- O : open show.
- F : flow show.

- J : job shop.
- $\alpha_2 \in \{\theta, m\}$.
- θ : le nombre de machines est arbitraire.
- m : le nombre de machine est égal à m .

3.7.2 Champ β

Permet d'obtenir des informations sur les contraintes prises en compte aussi les taches qui doivent être ordonnancée. Il est composé de 5 sous champs $\beta_1\beta_2\beta_3\beta_4\beta_5$.

- $\beta_1 \in \{\theta, pmtn\}$.
- θ : la préemption n'est pas autorisée.
- $Pmtn$: ma préemption est autorisée.
- $\beta_2 \in \{\theta, prec, tree, chain\}$.
- θ : pas de contrainte précédence.
- $Prec$: les contraintes précédence sont quelconques.
- $Tree$: les contraintes précédence sont données sous forme d'arbre.
- $Chain$: contrainte précédence sont donnée sous forme de chaine.
- $\beta_3 \in \{\theta, r_i\}$.
- θ : toute les dates de disponibilité sont nulles.
- R_i : il existe des dates de disponibilité pour les taches.
- $\beta_4 \in \{\theta, p_i = p, pa \leq p_i \leq pb\}$.
- θ : les temps de traitement des taches sont arbitraires.
- $P_i = p$: les temps de traitement des taches sont égaux à p .
- $pa \leq p_i \leq pb$: les temps de traitement de chaque taches est entre pa et pb .
- $\beta_5 \in \{\theta, D_i, d_i\}$.
- θ : il n'y a pas des dates échues pour les taches.
- D_i : il existe des dates échues pour les taches.
- d_i : il existe une date limite pour chaque taches.

3.7.3 Champ γ :

Le troisieme champ γ décrit le critère d'optimalité, on cherche à minimiser γ avec : $\gamma \in \{C_{max}, \sum C_i, \sum W_i C_i, D_{max}, L_{max}, F_{max}, \sum U_i, \sum W_i U_i\}$, en général se sont les critères les plus connus.

3.8 Conclusion

Dans ce chapitre, nous avons tout d'abord présenté les caractéristiques générales d'un problème d'ordonnancement, précisant notamment les différents éléments qui déterminent les tâches, les ressources, les contraintes et les critères d'optimisation, ainsi que la classification de ces problèmes. Finalement, nous avons présentés les modèles d'ateliers importants.

4

Modelisation du problème

4.1 Introduction

Le gestionnaire d'atelier souhaite optimiser les performances de son ordonnancement par rapport à des critères, dont le nombre et la Complexité augmentent. En utilisant les premiers logiciels d'ordonnancement, les gestionnaires d'ateliers essayaient d'optimiser un ou deux critères qu'ils jugeaient importants comme par exemple « minimiser les temps de Cycle », « minimiser le retard maximal » ou « minimiser le nombre de tâches en retard ».

A l'heure actuelle, les entreprises souhaitent que l'ordonnancement satisfasse des objectifs diversifiés, nombreux et parfois complexes et contradictoires. De plus en plus. On demande ainsi à l'ordonnancement de fournir un compromis entre tous ces objectifs plutôt qu'une optimisation réelle.

Ce chapitre est consacré à la présentation de la problématique et au principal objectif assigné à notre étude.

4.1.1 Position du problème

La résolution des problèmes rencontrés pour la gestion de leur poste de production. Constitue une des principales préoccupations des industriels.

En effet, comment réussir à produire le plus possible, le rapidement possible avec le moins de cout possible, le moins d'arrêts possible, le moins de personnel possible tout en respectant toutes les contraintes liées au produit (précédence, achèvement, périssabilité, ...)? Notre travail rentre dans ce cadre et consiste donc à développer des outils d'aide à l'ordonnancement permettant la gestion du poste de production et en particulier celui du conditionnement au niveau des industries véhicules.

La situation actuelle au niveau de l'atelier montage pont dans l'entreprise SNVI est caractérisée par des insuffisances dans les procédures d'alimentation des lignes.

Les principaux points caractérisant ces insuffisances sont :

- Un mauvais suivi et absence d'analyse des écarts de stocks lignes.
- Certains éléments ont un caractère aléatoire, non prévisible avec certitude (pannes, temps de réglage, retards de livraison, maladie, ...).

- Mauvaise coordination entre l'ordonnancement montage et les autres services.
- Réorganisation de travail manuel, flux d'informations très importantes d'où non maîtrise de l'information au moment opportun.

Une exploitation des moyens de production qui répond plus à un souci d'activation qu'à la réalisation des objectifs au niveau des ateliers.

4.1.2 Les hypothèses

- Les tâches indépendantes les unes des autres.
- Les sections des machines avec leurs ouvriers sont disponibles en permanence.
- Les matières premières toujours disponibles.
- Les pannes ne sont pas prises en compte dans ce modèle.
- La description des pièces est connue à l'avance.

Les contraintes Les contraintes suivantes doivent être respectées lors du traitement des tâches sur l'ensemble des machines :

- Chaque tâche doit visiter chaque machine de l'atelier où les machines sont placées en série.
- Une tâche ne peut pas être traitée sur deux machines au même temps.
- L'ordre de passage des tâches sur les machines est le même pour toutes les tâches.
- L'ordre traitement des tâches est le même pour toutes les machines.
- Pour chaque tâche, une fois que le traitement d'une opération sur une machine est achevé, elle rejoint la file d'attente pour terminer les autres opérations de son traitement sur les autres machines.

4.1.3 Objectif à atteindre

L'objectif principal fixé par l'atelier est de minimiser la durée d'exécution des commandes, autrement dit c'est de réaliser un ordonnancement des tâches de production. Tenant compte de certaines contraintes, on souhaite traiter l'ensemble des tâches sur l'ensemble des machines tout en minimisant la date de fin de traitement de toutes les tâches dites « Makespan ».

4.2 Principe de la modélisation

Les modèles mathématiques occupent une place privilégiée et distinctive dans l'arsenal de la Recherche Opérationnelle, de façon générale, un modèle est une représentation d'un fragment ou d'un aspect de la réalité.

Un modèle mathématique doit incorporer suffisamment de détails pour refléter fidèlement la situation réelle.

La modélisation est la première étape lors de la résolution d'un problème, cela révèle l'importance et la sensibilité de celle-ci en ce qui concerne la recherche et le développement d'une solution, tout le travail repose sur le modèle élaboré lors du traitement du problème et les résultats obtenus en dépendent entièrement, ce qui rend évident le fait de faire preuve d'une grande prudence pendant et construction du modèle.

Un problème d'optimisation se présente mathématiquement sous la forme fonctionnelle d'un objectif a plusieurs variables en le maximisant ou en minimisant tout en respectant certaines contraintes et variables qui le régissent.

L'état du système à optimiser est connu lorsque l'on connaît la variation des variables d'état.

Les variables que l'on doit ajuster pour optimiser le système sont appelées variables de décision.

La fonction a optimisé s'appelle indifféremment critère de performance du système ou fonction objectif.

La politique optimale est celle qui consiste a régler les variables de décision pour obtenir l'optimum recherché.

Après avoir défini le problème qui fait l'objet de notre étude ainsi que toute ses caractéristique, nous allons établir un modèle mathématique, qui donne les des résultats qui se rapproche d'avantage de la réalité et qui atteignent le mieux l'objectif ciblé.

Pour ce faire et pour plus d'éclaircissement, nous allons suivre les étapes suivantes :

- Identifications des indices, variables et données des problèmes.
- Etablissement des contraintes du modèle.
- Formulation mathématique des objectifs de l'entreprise.
- Modèle mathématique.

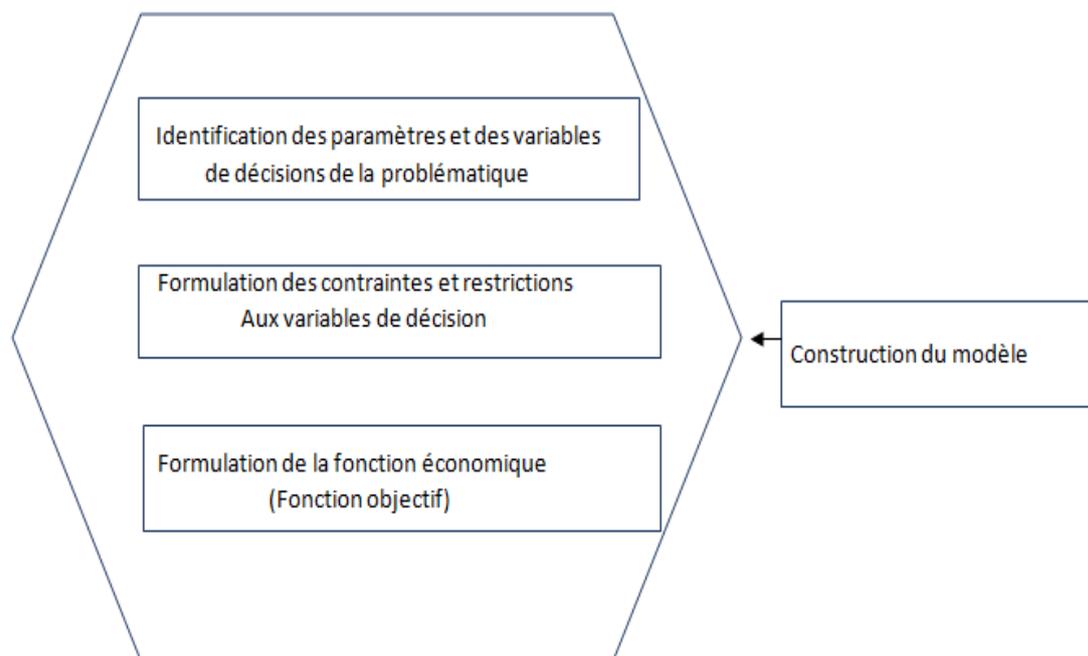


FIGURE 4.1: Etapes de la modélisation

4.2.1 Position du Flow Shop de permutation

Le problème traité est un problème d'ordonnancement à m machines spécialisées qui a comme objectif, minimiser la longueur de l'ordonnancement. Ce problème consiste à ordonner dans le temps n tâches sur m machines en respectant certaines propriétés qu'on va définir ci-dessous.

Notation

Suivant la notation de Graham citée dans le deuxième chapitre, le problème étudié est noté comme suit : $F_m/Permu/C_{max}$

4.2.2 Etude de l'état de l'art du problème $F_m/permu/C_{max}$

Plusieurs chercheurs se sont intéressés à ce problème dans le passé, et ont proposé des méthodes de résolutions (heuristiques, métaheuristiques).

Notons que c'est en 1980 que l'application des algorithmes génétiques sur les problèmes d'ordonnancement a commencé et les deux premières applications pour le Flow Shop de permutation ont été faites par Wagner.

En 1954, S.M. Johnson a établi un algorithme polynomial en $O(n \log(n))$ dit "algorithme de Johnson" pour la résolution du problème $F_m/Permu/C_{max}$ avec $m = 2$, où les tâches sont ordonnancées de sorte que, une tâche T_i soit placée avant une autre tâche T_j si et seulement si :

$$\min(p_{i1}, p_{j2}) \leq \min(p_{i2}, p_{j2})$$

Notons que dans le cas où,

$$\min_{1 \leq i \leq n} p_{i1} \geq \max_{1 \leq i \leq n} p_{i2} \text{ ou bien } \min_{1 \leq i \leq n} p_{i3} \geq \max_{1 \leq i \leq n} p_{i2}$$

le problème $F_3/permu/C_{max}$ peut être résolu en utilisant l'algorithme de Johnson. [Johnson.S.M, 1954]

En 1995, Cartier et Rebai ont proposé deux méthodes exactes de type "Branch and bound"

En 1995, Chen et al ont développé un simple algorithme génétique, utilisant les heuristiques CDS^1 et RA^2 pour la génération de la population initiale et PMX^3 comme opérateur de croisement. Concernant la mutation, aucun opérateur de mutation n'a été appliqué.

Cet algorithme a été comparé avec deux heuristiques existantes, et s'est avéré être meilleur.

En 1995, Reeves a présenté un algorithme génétique utilisant l'heuristique NEH^4 Pour générer une partie de la population initiale et le reste de la population a été généré aléatoirement. Un croisement à un point a été utilisé ainsi qu'une mutation en déplaçant la position d'un seul croisement gène. Cet algorithme à été comparé à un qu'un algorithme de recuit simulé et un autre de simple recherche de voisinage sur les instances connues de Taillard⁵. Les tests ont montré que cet algorithme a été très efficace comparé au recuit simulé et à l'algorithme de recherche.

En 1998, Reeves et Yamada ont amélioré l'algorithme présenté par "Reeves" en 1995, suggérant un nouvel opérateur de croisement noté MSXF (multi-step fusion). Ce nouvel algorithme a été testé sur les instances connues de Taillard avec 20 machines, et plusieurs bornes supérieures connues de ces instances ont été amélioré avec.

En 2001, Ponnambalam et al ont fait une comparaison entre les heuristique B constructives et itératives pour le problème du Flow Shop, et ont présenté à la septime section de leurs papier un algorithme génétique où la population initiale est générée aléatoirement, GPMX (generalized position crossover) est utilisé comme opérateur de croisement et un échange de deux gènes est utilisé comme opérateur de mutation. Cet algorithme génétique a été comparé avec plusieurs autres heuristiques comme NEH, CDS...etc et s'est avéré être meilleur que les heuristiques constructives mais seulement

sur 11 des 21 instances.

En 2001, Lorena et Furtado ont proposé un algorithme génétique constructif (CGA). Les résultats de cet algorithme ont été comparés avec des résultats fournis par des algorithmes efficaces connus dans la littérature sur la suite des tests de Taillard, et il s'est avéré que CGA fournit des résultats compétitifs pour le problème Flow Shop de permutation.

- **CDS** : une heuristique mise en oeuvre par Campbell, Dudek et Smith, d'où son nom.
- **RA** : (Rapid access) une heuristique mise en oeuvre par Dannenbring.
- **PMX** : opérateur de croisement mise en oeuvre par Baudet et al.
- **NEH** : Heuristique conçu par Nawaz, Ensore et Ham pour la minimisation du makespan dans un flow shop de permutation.
- **Instances de Taillard** : C'est des instances révélées être très dures.

Exemples du Flow shop de permutation

Exemple avec 4 machines :

On se donne 6 tâches à exécuter sur 4 machines, où la matrice des temps de traitement est donnée ci-dessous :

P_{ij}	T_1	T_2	T_3	T_4	T_5	T_6
M_1	3	4	5	4	2	1
M_2	4	2	5	2	2	3
M_3	1	3	2	4	2	2
M_4	1	2	3	2	2	1

TABLE 4.1: Exemple de $f_4/permu/c_{max}$.

La séquence $T_1 T_3 T_5 T_6 T_2 T_4$ est une solution réalisable pour les problèmes. $F_4/permu/C_{max}$ avec $C_{max} = 29$ Diagramme de Gantt associé :

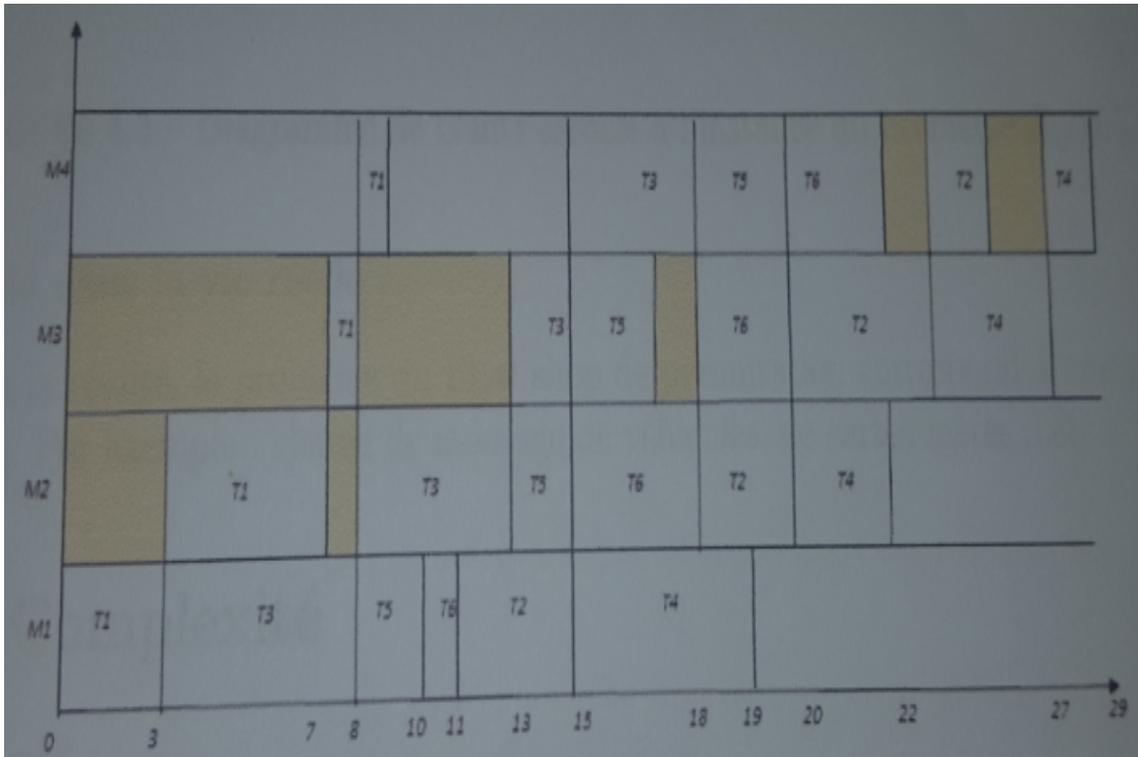


FIGURE 4.2: Diagramme de gantt associé à l'instance de problème $f_4/permu/c_{max}$

On voit bien que la permutation des tâches est la même sur toutes les machines, et que pour chaque tâche, l'ordre de passage sur les machines est le même (toutes les tâches passent par la machine M_1 puis M_2 puis M_3 puis M_4).

La figure 4-3 présente un diagramme de Gantt d'une solution réalisable au problème $F_4//C_{max}$.

On voit bien sur cette figure la différence entre un Flow shop de permutation et un Flow shop simple.

Ici la permutation des tâches sur les 4 machines n'est pas la même, il y a que l'ordre de Passage sur les machines qui est le même pour toutes les tâches (c'est à dire toutes les tâches passent par M_1 puis M_2 puis M_3 et enfin M_4).

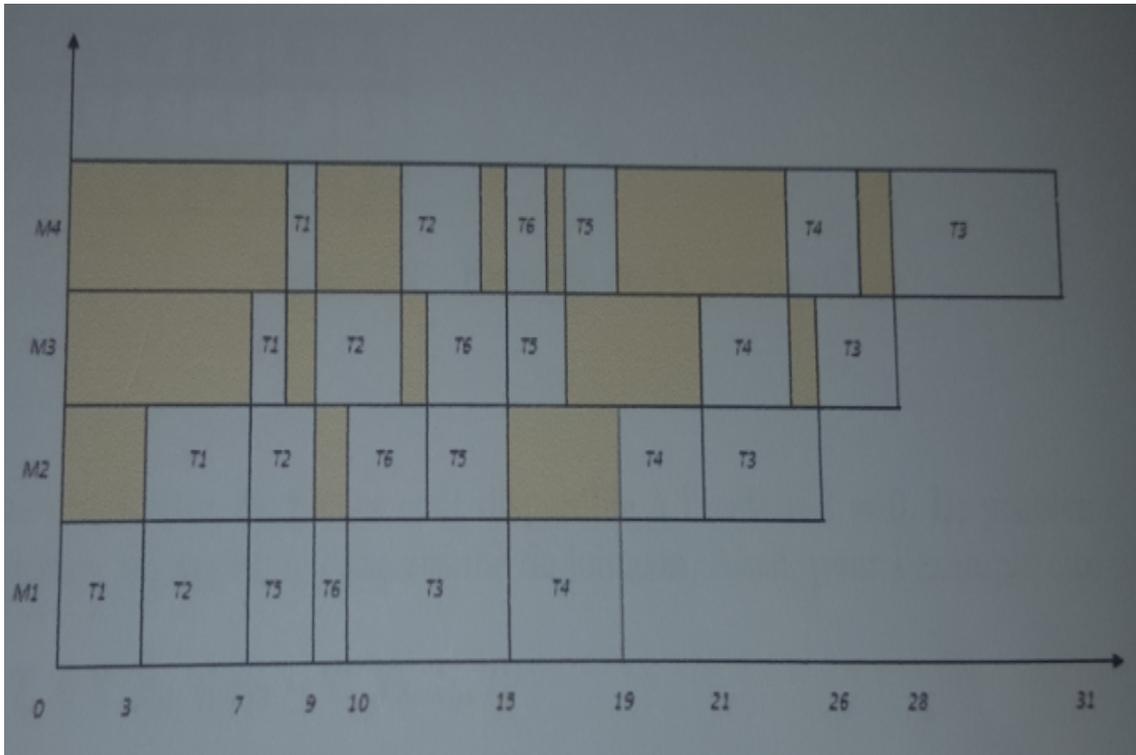


FIGURE 4.3: Diagramme de gantt associé à l'instance de problème $f_4//c_{max}$

Dans la réalité, le problème du Flow shop de permutation correspond une chaîne de montage. Par exemple : chaîne de montage de véhicules, de cartes mères ...etc.

4.2.3 Complexité

Pour $m = 2$: le problème est polynomial, il est résolu en $O(n \log(n))$ par l'algorithme de Johnson.

Algorithme de Johnson :

- Début
- Construire un ensemble X contenant les tâches qui ont un temps de traitement sur la première machine inférieur ou égale à celui sur la deuxième machine.
- Construire un ensemble Y contenant les tâches de T qui n'appartiennent pas X.
- Ordonner sur les deux machines, les tâches de X suivant l'ordre croissant des temps de traitement sur la première machine.
- Ordonner sur les deux machines, les tâches de Y suivant l'ordre décroissant des temps de traitement sur la deuxième machine.
- Fin.

Exemple de $F2/permu/C_{max}$

On se donne 6 tâches à exécuter sur 2 machines dans un système de type flow shop de Permutation, où la matrice des temps de traitement est donnée ci-dessous :

T _{ij}	T1	T2	T3	T4	T5	T6
M1	3	4	5	4	2	1
M2	4	2	5	2	2	3

TABLE 4.2: Exemple de $f_2/permu/c_{max}$.

Notons que toutes les tâches sont disponibles à l'instant t_0 . Le problème étant deux machines, on applique l'algorithme de Johnson. Ainsi, pour l'exemple cité plus haut :

— $X = T_i \in T/p_{i1} \leq p_{i2} = T_1, T_3, T_5, T_6$

— $Y = T_2, T_4$

Séquence de tâches : $T_6 T_5 T_1 T_3 T_2 T_4$

Diagramme de Gantt associé :

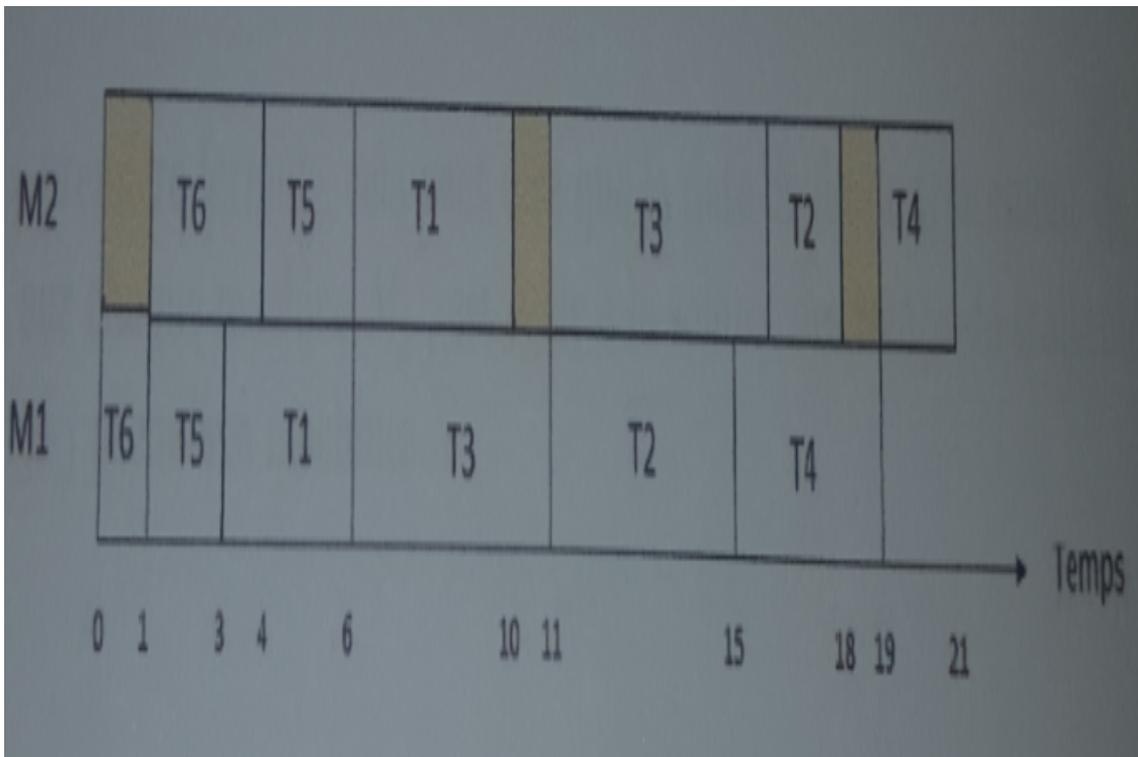


FIGURE 4.4: Diagramme de Gantt associé à l'instance du problème $F2/permu/C_{max}$

Ainsi $C_{max} = 21$ est une solution optimale pour cette instance.

Pour $m = 3$:

le problème est polynomial seulement dans le cas où :

$Min_{1 \leq i \leq n} p_{i1} \geq Max_{1 \leq i \leq n} p_{i2}$ ou bien $Min_{1 \leq i \leq n} p_{i3} \geq Max_{1 \leq i \leq n} p_{i2}$

C'est-à-dire si le minimum de temps de traitements sur une machine M_j ($j = 1$ ou $j = 2$ ou $j = 3$) est supérieur au maximum des temps de traitements sur une machine M_k ($k \neq j$ et $k = 1$ ou $k = 2$ ou $k = 3$),

alors la machine M_k peut être négligée. Ainsi on considère uniquement deux machines (on néglige la machine M_k) et on y applique l'algorithme de Johnson pour obtenir la permutation des tâches, puis on ordonnance les tâches suivant la permutation obtenue sur les 3 machines.

Pour : $m \geq 3$: le problème est NP difficile au fort.

4.2.4 Calcul de la longueur d'un ordonnancement

Une solution au problème $F_m/permu/C_{max}$ étant une permutation de tâches, une fois cette séquence obtenue les formules suivantes sont utilisées pour le calcul de la longueur de l'ordonnancement :

$$\begin{aligned} - C_{1j} &= \sum_{k=1}^j p_{1k} & \forall j = 1 \dots m \\ - C_{i1} &= \sum_{k=1}^i p_{k1} & \forall i = 1 \dots n \\ - C_{ij} &= \max(C_{(i-1)j}, C_{i(j-1)}) + P_{ij} & \forall j = 1 \dots m; i = 1 \dots n \\ - C_{max} &= C_{mn} \end{aligned}$$

Pour la première formule, cela veut dire que la date de fin de traitement de la tâche en position 1 sur chaque machine M_j , est égale à la somme des temps de traitement de cette tâche sur les J premières machines.

La deuxième formule veut dire que la date de fin de traitement de chaque tâche T_i sur la première machine est égale à la somme des temps de traitement des i premières tâches sur M_1 .

La dernière formule veut dire que la date de fin de traitement d'une tâche T_i sur une machine M_j est égale à la somme du temps de traitement de T_i sur M_j , et le maximum entre la date de fin de traitement de la tâche T_{i-1} sur M_j et celui de la tâche T_i sur M_{j-1} .

4.2.5 Modèle mathématique

Dans cette section, nous commençons par introduire les notations utilisées tout au long de notre étude. Ensuite, nous définissons les contraintes liées à notre problème et l'objectif à atteindre. Nous terminons, enfin, par la formulation du problème d'optimisation.

Notations

La notation utilisée dans ce mémoire est résumée dans ce qui suit :

Indice :

i : indice de tâche.

j : indice de machine.

Paramètre :

n : le nombre de tâches .

m : le nombre de machines.

Variables de décisions

$$x_{ik} = \begin{cases} 1 & \text{si la tâche } T_i \text{ est traité à la position } k \\ 0 & \text{sinon} \end{cases}$$

I_{jk} : temps d'inactivité de la machine M_j après le traitement de la tâche en position k (temps d'attente de la tâche suivante).

w_{jk} : temps d'attente de la tâche i en position k la machine M_j (temps d'attente de disponibilité de la machine M_{j+1}).

Avec $\forall j = 1..m, i = 1..n$ et $k = 1..n$

Contraintes

Les différentes contraintes du problème et leurs formulations mathématiques peuvent être résumées comme suit :

- $\sum_{i=1}^n x_{ik} = 1 \quad k = 1..n$ une seule tâche par position
- $\sum_{k=1}^n x_{ik} = 1 \quad i = 1..n$ une seule position pour chaque tâche
- $I_{jk} + \sum_{i=1}^n x_{ik+1} * p_{ij} + w_{jk+1} = w_{jk} + \sum_{i=1}^n x_{ik} * p_{ij+1} + I_{j+1k} \quad \forall k = 1..n-1, j = 1..m-1$
contrainte de conservation du temps
- $w_{j1} = 0 \quad \forall j = 1..m-1$
- $I_{1k} = 0 \quad \forall k = 1..n-1$

Conditions initiales : pas d'attente pour la tâche en position une et pas d'inactivité pour la première machine.

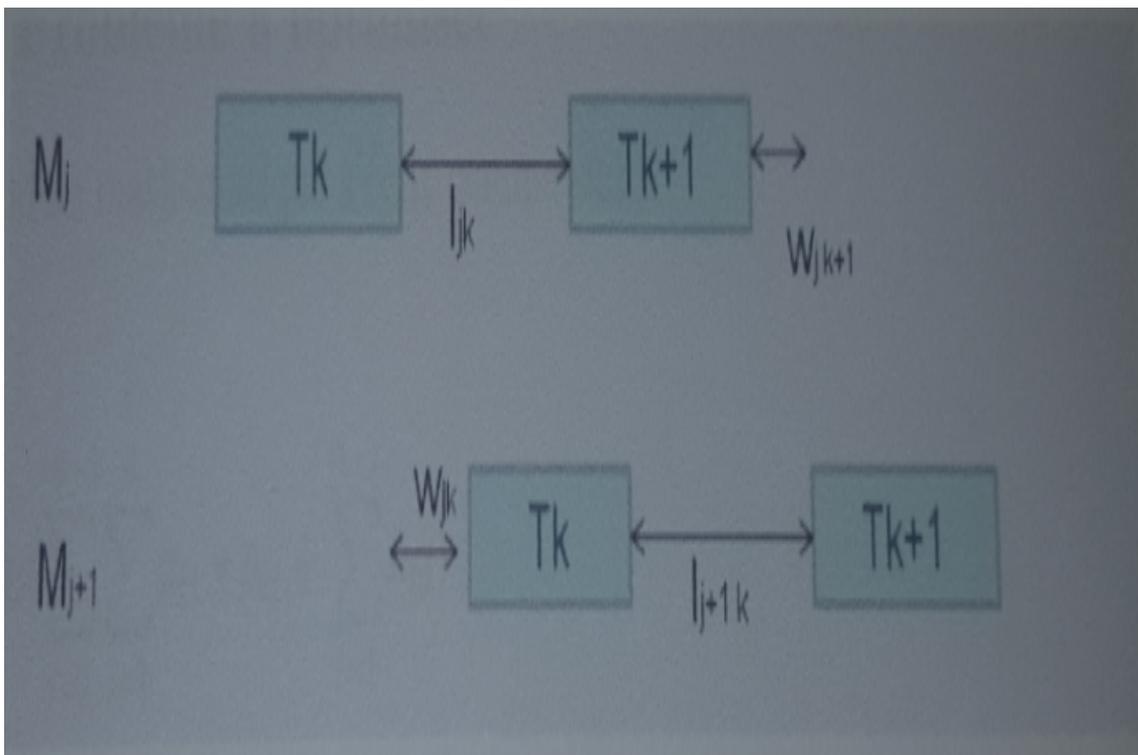


FIGURE 4.5: Contrainte de conservation du temps

La fonction objectif

On cherche à minimiser le temps d'attente global sur la dernière machine qui est composée des temps de traitement de la tâche en première position sur les $(m - 1)$ machines, et les temps d'inactivité entre les différentes tâches de la dernière machine, donc la fonction objectif s'écrit :

$$\text{Min} \left(\sum_{j=1}^{m-1} \sum_{i=1}^n x_{ij} * p_{ij} + \sum_{i=1}^{n-1} I_{mi} \right)$$

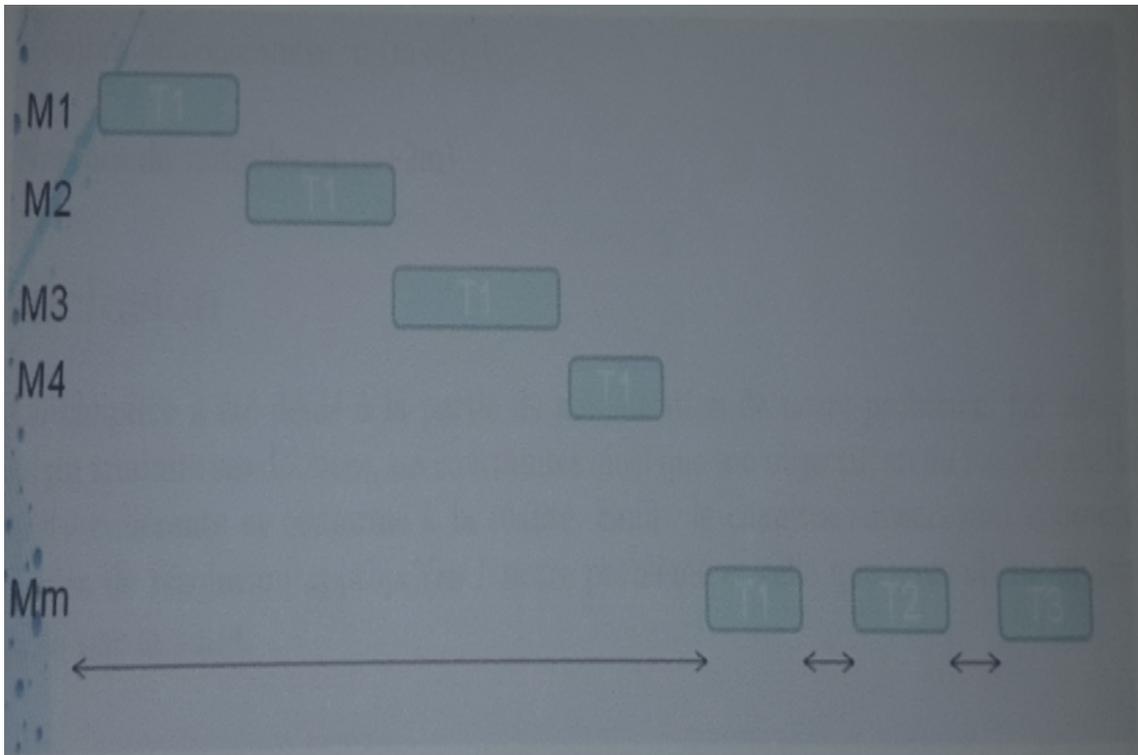


FIGURE 4.6: Fonction objectif

Problème à optimiser

Le modèle mathématique s'écrit comme suit :

$$\left\{ \begin{array}{l} \min \left(\sum_{j=1}^{m-1} \sum_{i=1}^n x_{ij} * p_{ij} + \sum_{i=1}^{n-1} I_{mi} \right) \\ \sum_{i=1}^n x_{ik} = 1 \quad k = 1..n \\ \sum_{k=1}^n x_{ik} = 1 \quad i = 1..n \\ I_{jk} + \sum_{i=1}^n x_{ik+1} * p_{ij} + w_{jk+1} = w_{jk} + \sum_{i=1}^n x_{ik} * p_{ij+1} + I_{j+1k} \quad \forall k = 1..n-1, j = 1..m-1 \\ w_{j1} = 0 \quad \forall j = 1..m-1 \\ I_{1k} = 0 \quad \forall k = 1..n-1 \end{array} \right.$$

Nombre de contraintes : $n(m+2) - 1$

Nombre de variable : $n(n+2m)$

4.3 Conclusion

À travers ce chapitre, nous avons exploité la problématique, les différentes contraintes aux quelles l'entreprise est soumise, ainsi que son objectif principal qui consiste à minimiser les délais de fabrications de centre mécanique pour réaliser le programme.

Afin de répondre à cette problématique, nous abordons dans la suite sa modélisation et sa méthode de résolution.

Il s'agit dans cette étape, de traduire les différentes contraintes rencontrées et l'objectif à atteindre dans un modèle mathématique.

A la partie modélisation de notre problème. En effet, nous avons pu traduire ses données, ses contraintes ainsi que son objectif, en un modèle mathématique explicite cohérente et conforme à la réalité. Enfin, le chapitre suivant sera consacré aux méthodes de résolution appropriées à notre problème et celle que nous allons choisir sera adapté par la suite.

5

Approche de résolution

5.1 Introduction

L'optimisation combinatoire occupe une place très importante en Recherche Opérationnelle. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solutions algorithmiques efficaces valables pour toutes les données. de nombreuses méthodes de résolution ont été développées et peuvent être classées globalement en deux catégories :

— **les méthodes exactes Dantzig :**

Permettent de chercher une solution optimale d'un problème donné, dans un intervalle de temps bien déterminé. Leur principe essentiel est d'énumérer de manière intelligente l'ensemble des solutions de l'espace de recherche. Parmi ces méthodes on retrouve les algorithmes relatifs aux graphes, Simplexe , branch and bound programmation dynamique, relaxation lagrangienne...

— **méthodes approchées Glover :**

Face au caractère intraitable de certains problèmes, et vu la nécessité de fournir de bonnes solutions du problème pratique en des temps raisonnables, se sont développées des méthodes approchées appelées heuristiques. Ces algorithmes sont généralement spécifiques à chaque problème et leur performance est évaluée empiriquement par comparaison avec d'autres approches. D'autres heuristiques sont conçues comme une méthode générale qui peut être adaptée à divers problèmes d'optimisation. Elles sont désignées sous le terme " métaheuristique ", et permettent d'améliorer la valeur la fonction objectif.

Vu qu'on est en présence d'un problème non linéaire assez complexe nous proposons d'utiliser, une méta-heuristique comme méthode de résolution. Parmi les méta-heuristiques existantes, nous privilégions un algorithme génétique pour la résolution de notre problème.

5.1.1 Les Algorithmes Génétiques

Parmi tous les types d'algorithmes existants, certains ont la particularité de s'inspirer de l'évolution des espèces dans leur cadre naturel. Ce sont les algorithmes génétiques, initiés dans les années 1970 par John Holland[25].

Avec les algorithmes évolutionnaires, nous passons à une autre catégorie de méta-heuristiques, celles des méthodes dites évolutionnaires, qui manipulent un ensemble de plusieurs solutions simultanément[25].

La métaheuristique la plus connue dans cette branche est celle reposant sur un algorithme génétique, inspiré du concept de sélection naturelle élaboré par Darwin[3]. Ici, le vocabulaire employé est directement calqué sur celui de la théorie de l'évolution et de la génétique.

Le but de ces algorithmes génétiques est d'optimiser une fonction prédéfinie, appelée fonction objective, ou fitness; ils travaillent sur un ensemble de solutions candidates, appelé "population" d'individus ou chromosomes (on utilisera indifféremment individu ou chromosome). Ces derniers sont constitués d'un ensemble d'éléments, appelés "gènes", qui peuvent prendre plusieurs valeurs, appelées "allèles", les problèmes d'ordonnancement peuvent être pris comme exemple.

Les différentes notions décrites dans ce chapitre ont été pris du livre de Goldberg "Algorithmes génétiques, Exploration, optimisation et apprentissage automatique".

5.1.2 Terminologie des algorithmes génétiques

Étant donné que les algorithmes génétiques ont leurs racines à la fois dans la biologie et l'informatique, le tableau 5.1 illustre les termes utilisés et leurs équivalents naturels[25].

Termes AG Signification biologique

Genès Caractère d'un individu Une unité d'information génétique transmise par un individu à ses descendantes Chromosome Une chaîne représentant un individu Une structure contenant les gènes.

Locus Position dans la chaîne Emplacement d'un gène dans son chromosome Allèle Valeur du caractère de l'individu Une des différentes formes que peut prendre un gène. ils occupent le même locus[9].

Termes	AG	Signification biologique
Gène	caractère d'un individu	une unité d'information génétique transmise par un individu à ses descendants
Chromosome	une chaîne représentant un individu	une structure contenant les gènes
Locus	position dans la chaîne	emplacement d'un gène dans son chromosome
Allèle	valeur du caractère de l'individu	une des différentes formes que peut prendre un gène.ils occupent le meme locus

TABLE 5.1: Exemple de $f_4/permu/c_{max}$

5.1.3 Avantages des algorithmes génétiques

Comparé aux autres algorithmes d'optimisation, les algorithmes génétiques présentent plusieurs points forts, citons :

Le fait de travailler sur plusieurs solutions au même temps (population de solution), plusieurs points de l'espace de recherche sont par cour Utilisation de l'évaluation de la fonction objectif sans prendre en compte sa nature. En effet, nous n'avons besoin d'aucune propriété sur la fonction à optimiser (continuité, dérivabilité ...).

Simplicité de leur mécanisme.

5.1.4 Principes de base des algorithmes génétiques

Indépendamment de la problématique traitée, les algorithmes génétiques sont basés sur les principes suivants[23] :

- Choix du codage des individus (solutions).
- Génération d'une population de solutions initiales de taille fixe N qui soit non homogène, dite génération initiale.
- Définition d'une fonction d'évaluation dite fitness permettant d'évaluer un individu et le comparer aux autres.
- Choix des individus sujets des opérateurs génétiques par un mécanisme de sélection.
- Génération de nouveaux individus l'aide des opérateurs génétiques suivants :
Opérateur de croisement : permet de produire de nouveaux individus (enfants) partir de deux individus (parents) selon une probabilité p_c dite probabilité de croisement.
Opérateur de mutation : consiste à modifier les valeurs des gènes selon une probabilité p_m dite probabilité de mutation. Ceci afin d'éviter d'avoir des populations uniformes incapables d'évoluer.
- -Génération d'une nouvelle population de même taille que la génération initiale(N).

5.1.5 Fonctionnement des algorithmes génétiques

Dans ce qui suit, les différentes étapes d'un algorithme génétique seront expliquées en détail et dans l'ordre de leur déroulement[27] :

Codage des individus

Un codage est choisi pour représenter les individus. Trois types de codage sont utilisés généralement :

- Codage binaire : chaque gène sera représenté par un entier $X = \{0; 1\}$.
- Codage alphanumérique : chaque gène sera représenté par une combinaison de lettres et chiffres, par exemple dans le problème du PVCI un gène peut être codé par V_i où V_i signifie la ville numéro i .
- Codage réel : Chaque gène sera représenté par un réel.

Fonction d'évaluation

Afin d'évaluer la qualité des individus, une fonction dite fitness est utilisée. Elle nous permet de comparer les individus entre eux. Généralement on cherche à maximiser cette fonction.

Population initiale

Un nombre fixe d'individus est généré au début de l'algorithme, cette génération peut être aléatoire comme elle peut être obtenue suite à une application d'une heuristique au problème.

Opérateurs de sélection

La sélection permet de choisir les individus susceptibles de subir un croisement dans une population et par la suite ceux qui vont être reproduits dans la nouvelle population. Cet opérateur est très important car il permet aux individus d'une population de survivre, se reproduire ou mourir.

Plusieurs méthodes de sélection existent dans la littérature, citons :

Sélection par rang :

Consiste à attribuer à chaque individu un rang selon son évaluation. Pour un problème de maximisation les individus seront classés selon l'ordre croissant des valeurs de la fonction fitness. Ainsi, le rang numéro 1 sera attribué à l'individu qui a la plus petite valeur de la fonction fitness et ainsi de suite (voir table 5.2). Pour un problème de minimisation, les individus seront classés selon l'ordre décroissant des valeurs de la fonction fitness.

Après avoir attribué des rangs aux individus de la population, des probabilités de sélection seront calculées pour chaque individu comme suit :

Probabilité de sélection :

$$individu_i = \frac{\text{rang } individu_i}{\sum_{j \in \text{population}} \text{rang } individu_j}$$

Ainsi, les meilleurs individus auront une grande probabilité de croisement comparés aux autres, donc ils auront plus de chances d'être choisis comme parents. (voir exemple 2.1) :

Individus	Fitness	Rangs	Probabilités de sélection
I1	25	3	50%
I2	40	2	33.33%
I3	10	1	16.67%
Total	75	6	100%

TABLE 5.2: Sélection par rang pour un problème de maximisation.

Sélection par roulette :

Imaginons une roulette de casino sur laquelle sont placés tout les individus de la population où le secteur accordé à un individu est proportionnel à la valeur de sa fitness. Ainsi, les individus ayant une grande valeur de la fonction fitness auront plus de chances d'être sélectionnés. (voir figure 5.1)

Afin de sélectionner les individus on procède comme suit :

- On calcule la probabilité de sélection de chaque individu, avec :

$$P(individu_i) = \frac{\text{fitness } individu_i}{\sum_{j \in \text{population}} \text{fitness } individu_j}$$

On calcule la somme :

$$s_1 = \sum_{i=1}^n p(individu_i)$$

On tire aléatoirement un nombre $r \in [0, S_1]$

On calcule S_2 la somme des probabilité de sélection et on s'arrête lorsque cette somme dépasse le nombre r .

Le dernier individu ajouté la somme S_2 est sélectionné.

On réitère les étapes précédente jusqu'à la sélection du nombre d'individus souhaité.

Individu	1	2	3	4	5
valeur fitness	15	40	16	20	13

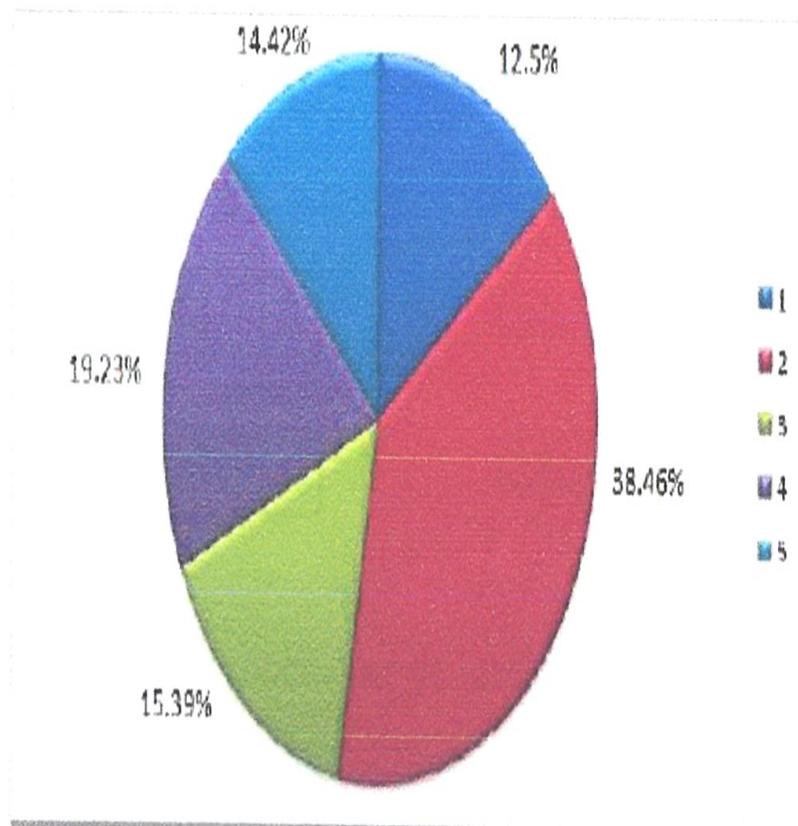


FIGURE 5.1 – Exemple de représentation des proportions des individus.

Sélection aléatoire :

Cette sélection se fait aléatoirement et sans intervention de la valeur de la fonction d'évaluation. Ainsi, chaque individu aura une probabilité uniforme ($1/N$) d'être sélectionné.

Sélection par tournois :

Le principe de cette méthode est de choisir une sous population de taille $M (M \leq N)$ où M est fixé par l'utilisateur. L'individu de meilleure qualité par rapport cette sous population sera considéré comme vainqueur et sélectionnée pour être parent. Le choix du paramètre M est très important, par exemple quand $M = 1$, cette sélection correspondra à une sélection aléatoire.

Opérateurs de croisement :

d'enrichir la diversité de la population, on utilise un opérateur de croisement Généralement cet opérateur est appliqué à deux parents et génère deux enfants par la suite.

croisement s'effectue selon une probabilité p_c fixée au début par l'utilisateur. Par exemple, si $p_c = 0.6$

alors sur une population de 100 individus, exactement 60 individus seront sélectionnés pour subir un croisement.

Les plus anciens opérateurs de croisement utilisés sont l'opérateur de croisement à un point et celui à deux points.

croisement à un point :

Consiste à diviser chaque parent en deux parties à la même position choisie au hasard. Le premier enfant est constitué de la première partie du parent 1 et le reste des gènes seront complétés avec les valeurs manquantes selon leur ordre dans le deuxième parent.

Le deuxième enfant sera constitué de la première partie du parent 2 et le reste des gènes sera complété par les valeurs manquantes selon leur ordre dans le premier parent. (Voir exemple 2.2)



FIGURE 5.1: opérateur de croisement à un point

Croisement à deux points :

Consiste à choisir deux points de coupures afin de diviser les deux parents en trois parties. Le premier enfant héritera la partie comprise entre les deux points de coupure du parent 1, et le reste des gènes de cet enfants seront complété par les valeurs manquantes selon leur ordre dans le parent 2. Le deuxième enfant héritera la partie comprise entre les deux points de coupure du parent 2, et le reste des gènes de ces enfants seront complétés par les valeurs manquantes selon leur ordre dans le parent1. (voir exemple 2.3)



FIGURE 5.2: opérateur de croisement à deux points

Opérateurs de mutation :

Afin que l'exploration de l'espace de recherche soit efficace, un opérateur de mutation est utilisé. Cet opérateur apporte une certaine diversité à la population et permet d'atteindre la plupart des sous-espaces de solution réalisables, d'où sa très grande importance. En effet, cerf prouve qu'un algorithme génétique peut converger vers une bonne solution sans croisement, rien qu'en utilisant un opérateur de mutation. L'opérateur de mutation s'effectua avec une probabilité de mutation noté P_m , qui est généralement très petite. Par exemple si $p_m = 0.03$, dans ce cas, sur une population de 100 individus uniquement 3 individus vont subir une mutation. Dans la littérature on trouve les différents opérateurs de mutation suivants :

Echange de deux gènes quelconques :

Comme l'indique son nom, dans cette opérateur de mutation deux gènes seront échangé. Pour cela, on tire aléatoirement deux positions et on inverse les se trouvant ces deux positions. (Voir exemple 2.4)



FIGURE 5.3: échange de deux gènes quelconques

Echange de deux gènes consécutifs : Deux gènes consécutifs seront inversés. Pour cela, on tire une position d'un gène aléatoirement et on l'inverse avec le gène qui le suit. (Voir exemple 2.5).



FIGURE 5.4: échange de deux gènes consécutifs

Déplacement d'un gène : Dans cette méthode un gène est déplacé de sa position initiale vers une autre position. Pour cela, deux positions seront tirées aléatoirement pour ensuite déplacer le gène se trouvant la position 1 vers la position 2 en faisant un décalage d'une position des gènes se trouvant entre les deux tirées. (Voir exemple 2.6)



FIGURE 5.5: déplacement d'un gène

Inversement d'une séquence de gènes : Dans cette méthode, deux positions non consécutives sont générées aléatoirement pour ensuite inverser la séquence de gènes se trouvant entre ces deux positions. (voir exemple 2.7)



FIGURE 5.6: inversement d'une séquence de gènes

Opérateur de remplacement

Cet opérateur est basé sur la fitness de l'individu. Son rôle consiste à déterminer quel individu parmi la population courante va survivre dans la génération qui suit. On trouve dans la littérature essentiellement deux méthodes de remplacement différentes :

Remplacement stationnaire :

Dans ce cas les enfants remplaceront les parents dans la génération qui suit tout en gardant la taille de la population constante (N).

Remplacement élitiste :

Dans ce cas, les individus de la population courante seront ordonnés suivant leurs fitness et les N meilleurs individus survivront pour la génération suivante.

Test d'arrêt :

Un critère doit être fixé pour arrêter l'algorithme, ce critère indique que la solution courante est assez proche de l'optimum.

Plusieurs critères d'arrêt de l'algorithme génétique sont possibles :

On peut choisir d'arrêter l'algorithme après un certain nombre d'itérations suffisant pour que l'espace de recherche soit convenablement exploré. Mais ce critère peut être coûteux en temps de calcul si le nombre d'individus à traiter dans chaque population est grand.

On peut aussi choisir d'arrêter l'algorithme lorsque la population n'évolue plus suffisamment.

— L'algorithme peut être arrêté aussi quand la valeur de la fonction fitness d'un individu dépasse un seuil fixé au départ.

On a donné dans ce chapitre un aperçu général sur les algorithmes génétiques et leur

fonctionnement tout en expliquant chaque étape de leur déroulement. On conclut par l'organigramme ci-dessous résumant les différentes étapes des AG.

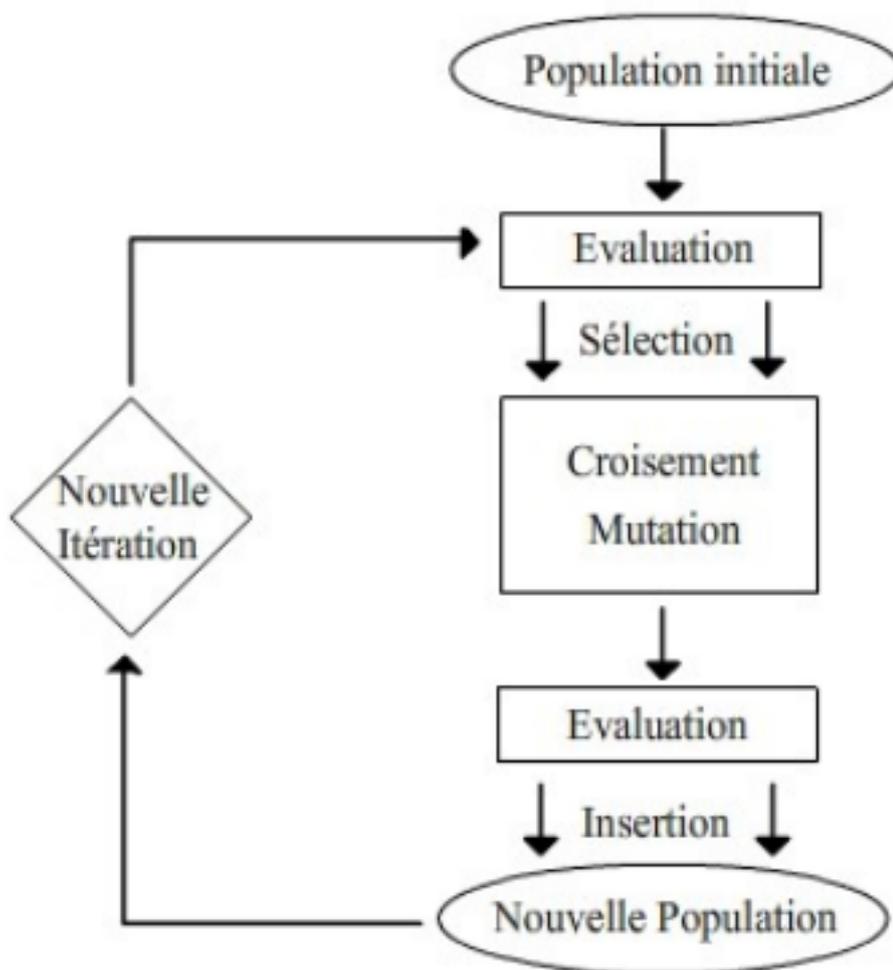


FIGURE 5.7: principe de fonctionnement d'un algorithme génétique

5.2 Conclusion

Dans ce chapitre, nous avons introduit quelques méthodes d'optimisation combinatoire, exactes et approchées, qui sont couramment utilisées pour la résolution des problèmes d'ordonnancement d'atelier.

Nous avons vu dans cette partie le principe générale de l'algorithme génétique qui peut être une bonne solution pour résoudre des problèmes d'optimisation combinatoire.

Le chapitre suivant illustre l'implémentation de cette méthode, et donne une description du logiciel développé, ainsi que les résultats obtenus.

6

Implémentaion

6.1 Introduction

l'implémentaion d'un modèle et la programmation de son système de résolution représentent une étape nécessaire en recherche opérationnelle.

dans les problèmes de décisions et d'optimisation, l'implémentation d'une solution fait appel à des logiciels puissants, bien structurés et qui donnent souvent des solutions satisfaisantes et efficaces.

Nous commençons tout d'abord par la présentation des ressources et du langage que nous avons utilisé. Puis les étapes de la réalisation du modèle et on termine par les tests effectués.

6.1.1 présentation du logiciel Python

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Créé par Guido van Rossum et sorti en 1991, Python a une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces importants. Il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire. Il prend en charge de multiples paradigmes de programmation, y compris orientés objet, impératifs, fonctionnels et procéduraux, et dispose d'une bibliothèque standard vaste et complète. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation.

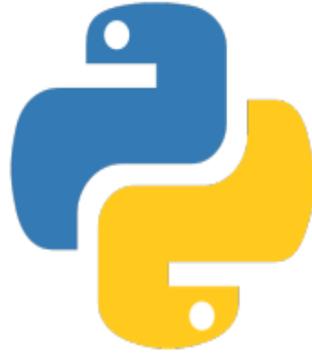


FIGURE 6.1: Logo python

6.1.2 Bibliothèques utilisées

Numpy :

Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Il implémente des calculs sur des tableaux multidimensionnels et matrices.

Pandas :

Pandas est une bibliothèque d'analyse de données de haut niveau efficace. Il offre des structures de données rapides, flexibles et faciles à manipuler, à savoir "DataFrame" et "Séries".

6.2 Les données

pour fabriquer un pont on'a 33 machines spécialisé et 4 taches principale (corps de pont, et ensemble) :

matrice des temps d'exécution de les taches principale sur les 33 machines :

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
A	0	0	0	0	0	0	0	0	0	0	0
B	12.7	2.85	4.5	6.55	3	13.9	5	3	4.45	8	1
C	0	1.6	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0

	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22
A	0	18.56	6.2	0	0	14.8	0	14.1	15.5	0	10.5
B	0.65	0	0.65	0	1.65	10.3	6.4	0	0	0	0
C	0	0	0	4.45	3.65	0	0	0	0	6.75	1.5
D	0	0	0	0	0	0	0	0	0	6.75	6.9

	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33
A	19.5	10.5	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	4.45	3.65	0	0	0	0	6.75	1.5
D	0	0	19.5	6.55	15.95	1.4	3.55	1.9	2.55	2.55	3.7

6.3 Implémentation et résultats

nous avons implémenté l'algorithme génétique par l'utilisation du logiciel python.

```
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 11 00:52:30 2020

@author: clinique
"""
import numpy as np
import pandas as pd
import random
import math
import time
import random
import DateTime
```

FIGURE 6.2: bibliothèque et fonction utilisé

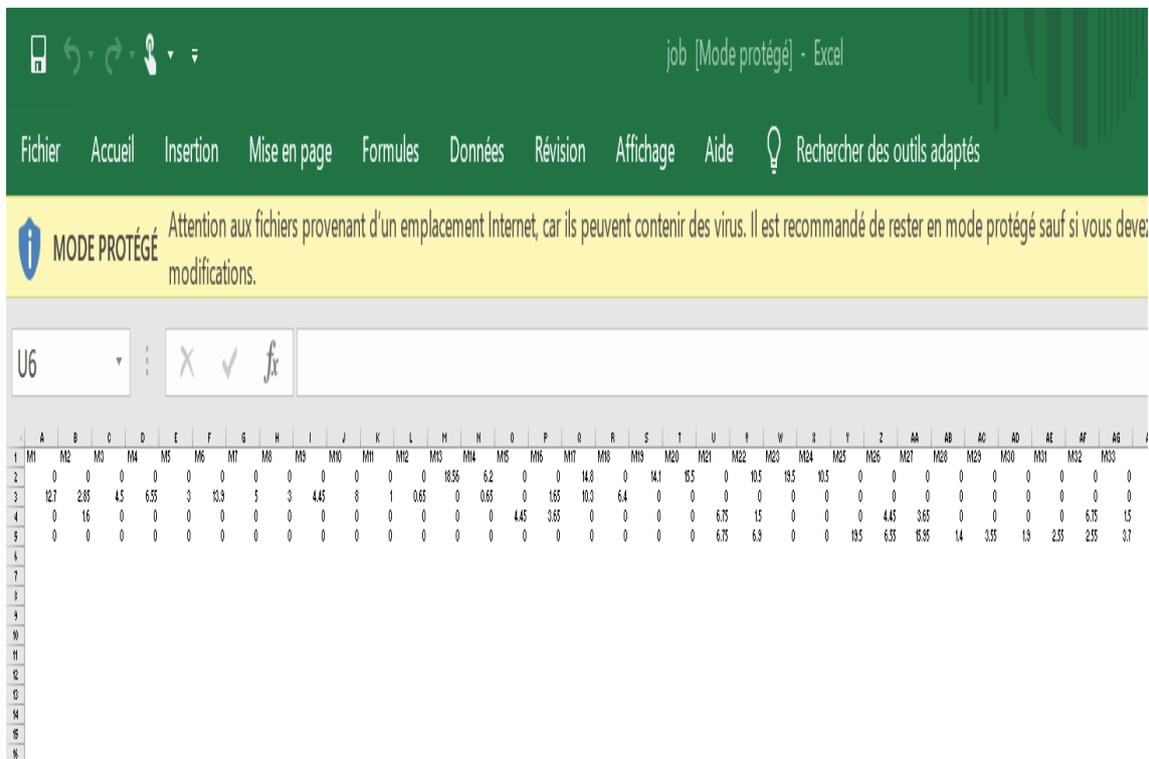


FIGURE 6.3: création des données dans un fichier excel

```
# read the 5 datasets
car2 = pd.read_excel(r'C:\Users\pc\OneDrive\Bureau\New folder\job.xlsx')
car2 = np.array(car2)
ca=car2.T

#chose the car number
p_ij=ca
```

FIGURE 6.4: appelée le fichier excel

```

nbm=len(p_ij)
nbj=len(p_ij[0])

print('Number of Machines:{0},Number of Jobs:{1}'.format(nbm,nbj))

#set the parameters
Npop = 24      # Number of population
Pc = 0.9      # Probability of crossover
Pm = 0.7      # Probability of mutation

```

FIGURE 6.5: entrer les probabilités

```

Population:
[[3, 2, 0, 1], [0, 2, 1, 3], [2, 0, 3, 1], [3, 0, 1, 2], [1, 2, 0,
3], [2, 1, 3, 0], [1, 0, 3, 2], [0, 3, 2, 1], [2, 3, 0, 1], [1, 0,
3, 2], [1, 3, 0, 2], [0, 2, 3, 1], [2, 3, 0, 1], [0, 2, 3, 1], [0,
1, 2, 3], [2, 3, 0, 1], [0, 2, 3, 1], [3, 2, 0, 1], [2, 0, 1, 3],
[2, 3, 1, 0], [0, 3, 2, 1], [3, 2, 0, 1], [2, 0, 3, 1], [0, 2, 1,
3]]

```

FIGURE 6.6: les populations

sachant que :

- 0 représente : A.
- 1 représente : B.
- 2 représente : C.
- 3 représente : D.

```

In [29]: runfile('C:/Users/pc/.spyder-py3/temp.py', wdir='C:/Users/
pc/.spyder-py3')
Number of Machines:33,Number of Jobs:4
makespan value 167.31000000000003
makespan value 168.91000000000003
makespan value 126.01
makespan value 232.91000000000003
makespan value 175.26
makespan value 237.46000000000004
makespan value 171.86000000000004
makespan value 111.25999999999999
makespan value 237.46000000000004
makespan value 191.60999999999999
makespan value 167.31000000000003
makespan value 111.25999999999999
makespan value 167.31000000000003
makespan value 167.31000000000003
makespan value 111.25999999999999
makespan value 167.31000000000003
makespan value 111.25999999999999
makespan value 168.91000000000003
makespan value 175.26
makespan value 171.86000000000004
makespan value 111.25999999999999
makespan value 168.91000000000003
makespan value 167.31000000000003

```

FIGURE 6.7: resultat des individus

```
Solution:  
[3, 2, 0, 1]
```

FIGURE 6.8: population optimale

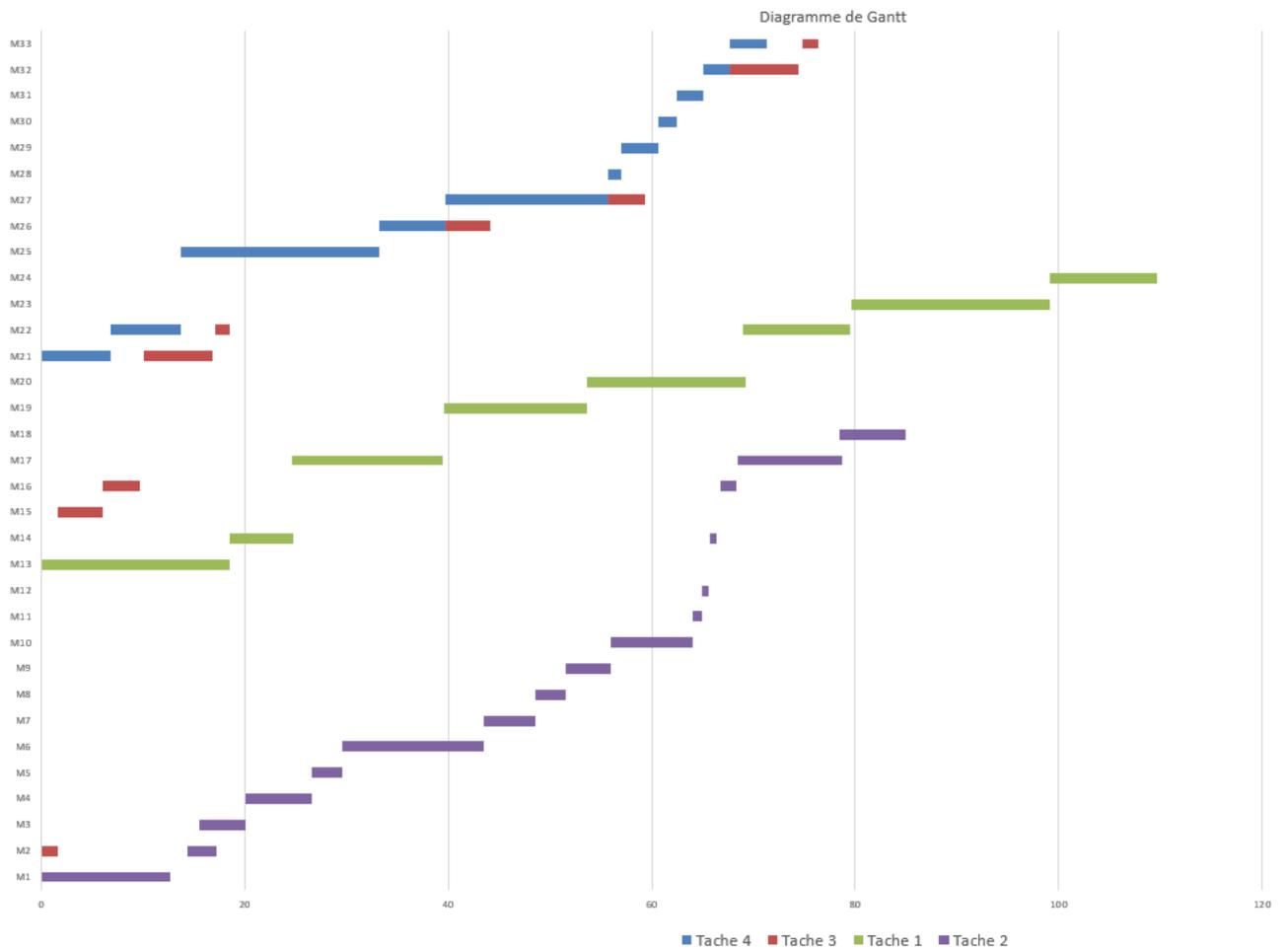
```
Best Objective Value:  
111.25999999999999
```

FIGURE 6.9: c_{max} optimal

```
CPU Time (s)  
0.97
```

FIGURE 6.10: le temps d'exécution de l'algorithme

représentation de la solution optimal par le Diagramme de Gantt :



les Individus(l'ordre des taches)	temps d'exécution (AG)
DCAB	111.25
ACBD	167.31
CADB	168.91
DABC	126.01
BCAD	232.91
CBDA	175.26
BADC	237.46
ADCB	171.86
CDAB	111.25
BADC	237.46
BDAC	191.60
ACDB	167.31
CDAB	111.25
ACDB	167.31
ABCD	167.31
CDAB	111.25
ACDB	167.91
BACD	111.25
CDBA	168.91
ADCB	175.25
DCAB	171.86
CADB	111.25
ACBD	168.91
ACDB	167.31

TABLE 6.1: résultats des testes effectués sur 4 taches et 33 machines.

discussion des résultats :

- Les résultats obtenus sont donnés sous forme de tableau. Ce dernier est composé de 2 colonnes et 24 lignes.
- Les lignes correspondent aux 24 permutations.
- Le premier définit les différents permutation de tâches.
- Le deuxième donne c_{max} (la date de fin de traitement de l'ensemble des tâches).
- Tous les résultats de la proposition sont des solutions réalisables lorsqu'il.
- Les problèmes rencontrés de l'atelier est la non ponctualisé de conception des ponts. Ou il fabriquent les pièces de façon aléatoire.
- Par la métaheuristique proposée, nous obtenons des solutions réalisables meilleurs que celles obtenues par la méthode d'atelier (S.N.V.I).
- Enfin notre application a répondu aux besoins de la société.

6.4 Conclusion

Dans ce chapitre nous avons présenté et expliqué l'implémentation du programme que nous avons proposé pour la résolution du problème de flow shop. Ce programme est basé sur métaheuristicues qui nous donnent des résultats proches de l'optimum dans des temps de traitement très faibles. La simulation a été faite, en utilisant le logiciel python, nous avons essayé de tester l'efficacité de notre algorithme (AG). A partir des différents résultats obtenus nous avons remarqué que notre algorithme est très efficace et donne des meilleures solutions.

7

Conclusion générale

Dans un problème d'atelier, une pièce doit être usinée ou assemblée sur différentes machines. Chaque machine est une ressource disjonctive, c'est-à-dire qu'elle ne peut exécuter qu'une tâche. Les tâches sont liées exclusivement par des contraintes d'enchaînement. Plus précisément, les tâches sont regroupées en n entités appelées travaux ou lots. Chaque lot est constitué de m tâches à exécuter sur m machines distinctes. Il existe trois types de problèmes d'atelier, selon la nature des contraintes liant les tâches d'un même lot. Lorsque l'ordre de passage de chaque lot est fixé et commun à tous les lots, on parle d'atelier à cheminement unique (flow-shop).

Les travaux présentés dans ce mémoire traitent de l'évolution de la fonction ordonnancement au sein de l'entreprise, et plus précisément du processus d'amélioration de l'ordonnancement.

Nous avons, au cours de ce mémoire, rappelé les notions de base en théorie de l'ordonnancement.

Nous avons défini la problématique de l'atelier et par la suite nous avons étudié l'ensemble des contraintes ainsi que l'objectif de l'entreprise, par la suite, on a parlé du problème $F_m/permu/c_{max}$ et des travaux qui ont eu lieu au sujet de ce problème, puis nous avons fourni le modèle mathématique du Wagner pour le problème étudié.

On a donc établi par la suite un algorithme génétique pour une résolution approchée qui a déjà prouvé son efficacité dans différents domaines tels que celui de l'optimisation.

Nous avons programmé la méthode proposée en programme python, l'application de ces méthodes a confirmé le bien fondé de l'implémentation.

Enfin notre application a répondu aux besoins de la société.

Bibliographie

- [1] Razika BEN AHMED and Zakiya KHERROUBI. *Optimisation d'un problème d'ordonnancement de type job shop avec contrainte de transport*. PhD thesis.
- [2] Imene Benkalai. *Ordonnancement d'ateliers en présence d'opérateurs*. PhD thesis, Université du Québec à Chicoutimi, 2018.
- [3] IMENE Benkalai, PIERRE Baptiste, and DJAMAL Rebaine. Ordonnancement d'ateliers de type flow shop avec contrainte d'opérateurs. *11ème Congrès International de Génie Industriel-CIGI 2015*, 2015.
- [4] Jacek Blazewicz, Klaus H Ecker, Erwin Pesch, Günter Schmidt, and Jan Weglarz. *Scheduling computer and manufacturing processes*. springer science & Business media, 2013.
- [5] Brecht Cardoen, Erik Demeulemeester, and Jeroen Beliën. Operating room planning and scheduling : A literature review. *European journal of operational research*, 201(3) :921–932, 2010.
- [6] Jacques Carlier, Philippe Chrétienne, Jacques Erschler, Claire Hanen, Pierre Lopez, Alix Munier, Eric Pinson, Marie-Claude Portmann, Christian Prins, Christian Proust, et al. Les problèmes d'ordonnancement. *RAIRO-Operations Research*, 27(1) :77–150, 1993.
- [7] Susana Carrera. *Planification et ordonnancement des plateformes logistiques*. PhD thesis, 2010.
- [8] CC Chang. H. jerome keisler. model theory. actes du congrès international des mathématiciens 1970, gauthier-villars, paris 1971, vol. 1, pp. 141–150. *The Journal of Symbolic Logic*, 38(4) :648–648, 1973.
- [9] Moataz Elsied, Amrane Oukaour, Hamid Gualous, and Radwan Hassan. Gestion de l'énergie et optimisation du système multisources basée sur l'algorithme génétique. 2014.
- [10] Patrick Esquirol, Pierre Lopez, and Pierre Lopez. *L'ordonnancement*. Economica, 1999.
- [11] F Farhoodi. A knowledge-based approach to dynamic job-shop scheduling. *International Journal of Computer Integrated Manufacturing*, 3(2) :84–95, 1990.
- [12] Gerry F Funk, Lucy Hynds Karnell, Robert A Robinson, Weining K Zhen, Douglas K Trask, and Henry T Hoffman. Presentation, treatment, and outcome of oral cavity cancer : a national cancer data base report. *Head & neck*, 24(2) :165–180, 2002.
- [13] B Grabot. Objective satisfaction assessment using neural nets for balancing multiple objectives. *International Journal of Production Research*, 36(9) :2377–2395, 1998.
- [14] Ronald L. Graham. Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics*, 17(2) :416–429, 1969.

- [15] Abdelkader HADRI. *L'ORDONNANCEMENT PAR INSERTION EN TEMPS REEL DE LA PRODUCTION DANS UN ATELIER FLEXIBLE*. PhD thesis, Université de Batna 2, 2012.
- [16] Slim Hammadi. *Une méthode d'ordonnancement minimisant les temps d'attente et de transit dans les systèmes de production flexibles de type job-shop*. PhD thesis, Lille 1, 1991.
- [17] Faicel Hnaien and Farouk Yalaoui. Optimisation biobjectif d'ordonnancement et de maintenance d'un atelier flow-shop. In *2010 8e Conférence Internationale de MODélisation et SIMulation-MOSIM10*, 2010.
- [18] Habiba HOUARI. *Planification et Ordonnancement en temps réel d'un Job shop en utilisant l'Intelligence Artificielle*. PhD thesis, 2012.
- [19] Mounya KACIMI EL HASSANI. *Etude comparative des algorithmes génétiques pour un problème d'ordonnancement d'atelier M-Machines identiques en parallèles*. PhD thesis, UNIVERSITE MOHAMED BOUDIAF-M'SILA FACULTE DES MATHEMATIQUES ET DE L ... , 2019.
- [20] Young Hoon Lee and Michael Pinedo. Scheduling jobs on parallel machines with sequence-dependent setup times. *European Journal of Operational Research*, 100(3) :464–474, 1997.
- [21] Arnaud Malapert. *Techniques d'ordonnancement d'atelier et de fournées basées sur la programmation par contraintes*. PhD thesis, 2011.
- [22] Pascale Marangé, Alexis Aubry, and Jean-François Pétin. Ordonnancement d'ateliers à partir de patrons de modélisation basés sur des automates communicants. 2016.
- [23] Edern Menou. *Conception d'alliages par optimisation combinatoire multiobjectifs : thermodynamique prédictive, fouille de données, algorithmes génétiques et analyse décisionnelle*. PhD thesis, Nantes, 2016.
- [24] Falou Ndoye. *Ordonnancement temps réel préemptif multiprocesseur avec prise en compte du coût du système d'exploitation*. PhD thesis, Université Paris Sud-Paris XI, 2014.
- [25] Zineb Otsmani, M Khiat, and A Chaker. Minimisation du coût de la maintenance préventive périodique d'un système série-parallèle par l'algorithme génétique. *Acta Electrotehnica*, 52(2) :74–80, 2011.
- [26] Bernard Roy and Daniel Carré. *Les problèmes d'ordonnancement : applications et méthodes*, volume 2. Dunod, 1964.
- [27] Rachid Saou. *Modélisation et optimisation de machines lentes à aimants permanents : Machines à double saillance et à inversion de flux*. PhD thesis, Ecole Nationale Polytechnique, 2008.
- [28] Pierre-François et Mouni 'e Gr é gory Saule, Erik et Dutot. Planification avec contraintes de stockage. In *Symposium international IEEE 2008 sur le traitement parallèle et distribué*, page 1 à 8.
- [29] Qiao Zhang. *Contribution à l'ordonnancement d'ateliers avec ressources de transports*. PhD thesis, 2012.