REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Mohamed Boudiaf de M'sila
Faculté des Mathématiques et de l'Informatique
Département des Mathématiques

Mémoire de Master

Domaine : Mathématiques et Informatique
Filière : Mathématiques
Option : Analyse Mathématique et Numérique

Thème

Etude d'un problème d'ordonnancement avec des exemples

Présentée par :
LEBOUKH Amira

Devant le jury composé de :

Dr SELT Omar	**MCA**	Université Mohamed Boudiaf de M'sila	**Encadreur**
Dr GAGUI Bachir	**MCA**	Université Mohamed Boudiaf de M'sila	**Président**
Dr DILMI Mustapha	**MCB**	Université Mohamed Boudiaf de M'sila	**Examinateur**

Année universitaire 2019/2020


بسم الله الرحمن الرحيم

Remerciement

Ce mémoire marque une étape très importante dans ma vie couronnant mes années d'études universitaires, je voudrais remercier ici les personnes qui m'ont accompagné au cours de mon parcours, et qui ont participé de prés ou de loin à la réalisation de ce projet

Je remercie avant tout Allah, tout puissant pour la volonté, la santé et la patience qu'il j'ai donnée durant toutes ces langues années d'études afin que je peux arriver à ce stade

En second lieu, je tiens à remercier le professeur SELT Omar mon encadreur pour son encadrement, disponibilité et patience, Aussi je les remercie pour son encouragement, conseils et orientations et pour son soutien tout au long de cette période

Un grand Mercie à mes parents qui m'ont données la chance de pour suivre mes études et qui m'ont appris à surpasser les moments difficiles

Enfin, je ne oublie pas de remercier mes chères amies et mes collègues et à toutes les personnes qui auront contribué de prés ou de loin à l'élaboration de ce mémoire ainsi qu'un la réussite de cette formidable année universitaire

Dédicace

Mon voyage universitaire s'est terminé

Après la fatigue et les epreuves, et aujourd'hui je

Termine mes recherches de fin d'études

Avec toute la vigueur et l'activité

Je dédie ce mémoire

A ma très chère ma

Mère pour tout le courage et l'amour et le patience qui ma donnée durant tout cette période et qui ma supporter et mon père

Pour son effort et son soutien et

Encouragement

A mes frères et ma sœur

A mes amies qui m'ont soutenu et aidé et tous ceux qui ont participé à ce travail

Introduction générale :

Les mathématiques sont une science très large qui contient de nombreuses difficultés et concepts et qui développe encore à ce jour, et parmi les concepts que les scientifiques étudient encore L'ordonnancement et ses problèmes.

Les problèmes d'ordonnancement formulés en problèmes d'optimisation sont souvent classées NP –Difficiles, en particulier ceux liés aux systèmes de production la résolution de tels problèmes nécessitent des méthodes dédiées, tandis que les méthodes exactes ne peuvent pas résoudre ce types de problème vu le temps de calcul énorme les méthodes approchées offrent la possibilité de trouver une solution réalisable en un temps raisonnable.

Parmi les méthodes approchées les plus utilisées pour la résolution des problèmes NP-Difficiles on trouve les méta heuristiques. Durant des années plusieurs méta heuristiques ont été construits et appliquées pour la résolution de tels problèmes

Mon travail est composé de 3 chapitres :

Le premier chapitre est consacré la présentation des notions générales liées à mon travail, j'ai apprendre ce que cela signifie L'ordonnancement et leur fonctions techniques de l'industrie et on va expliquer les données d'un problème d'ordonnancement et leur classification.

Le deuxième chapitre a pour but d'introduire les méthodes de résolutions des problèmes L'ordonnancement, commençant par les méthodes approchés allant aux heuristiques et méta heuristiques, avec la présentation des principes et des algorithmes de certaines techniques.

Le troisième chapitre est une présentation générale sur quelques types des problèmes d'ordonnancement.

Je finirais ce document par une conclusion générale qui résumé les différents phases de travail et ouvre la porte devant certaines perspectives

Table des matières

Chapitre 1

Génnéralité sur l'ordonnancement

1.1 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 termes de coûts de production et de délais de livraison. Les problèmes d'ordonnancement sont présents dans tous les secteurs d'activités de l'économie, depuis l'industrie manufacturière jusqu'à l'informatique

L'ordonnancement s'interésse au calcule de dates d'exécution optimales de taches. pour cela,il est trés souvent nécessaire d'affecter en méme temps les ressources nécessaire à l'exécution de ces taches.

Un probléme d'ordonancement peut étre considéré comme sous probléme de planification dans lequel il s'agit de décider de l'exécution opérationnelle des taches planifiées.

La théorie de l'ordonnancement traite des modèles mathématiques mais analyse également des situations réelles fortes complexes.L'ordonnancement est lié à plusieurs secteurs de recherches et d'activités très variés. En informatique, le choix des tâches à envoyer aux processeurs se modélise comme un problème d'ordonnancement. En production,l'ordonnancement consiste à déterminer les séquences d'opérations à réaliser sur les différentes machines de l'atelier. En gestion de projet, ordonnancer, c'est déterminer les

dates d'exécution des activités constituant le projet. Chacun de ces contextes nécessite la détermination des caractéristiques propres des tâches et des ressources, le type des décisions à prendre, les modalités d'exécution des tâches par les ressources, qui déterminentdes contraintes sur les décisions et aussi les différents critères à optimiser.

Dans ce chapitre,on s'intéresse ici uniquement aux problèmes d'ordonnancement dans les entreprises manufacturières. Il constitue un rappel de quelques notions de base relatives aux problèmes d'ordonnancement. Il rappelle aussi quelques concepts sur la théorie de la complexité des problèmes d'ordonnancement.

1.2 définition :

un problème d'ordonancement consiste à organiser dans le temps la réalisation de taches,compte tenu de contraintes temporélle (délais,contraintes d'enchainement) et de contrainte portant sur la disponabilité des ressources requses.

L'Ordonancement appartient à trois fonctions techniques de l'industrie.

Etude

production

maintenance

1.2.1 L'Ordonancement dans l'etude :

un Ordonancement constitue une solution au probléme d 'Ordonancement il est définir par le planning d'exécution des tàches (ordre et calendrier) et d'allocation des ressources et vise àsatisfaire un ou plusieurs objectifs

1.2.2 L'Ordonancement dans la production :

En production (manufacturiére,de biens , de service), on peut le présenter comme un probléme ou'il faut réaliser le déclenchement et le controle de l'avancement d'un ensemble de commandes à traverse les différents centres composant le système

1.2.3 L'Ordonancement dans la maintenance :

Systeme de communication rélatif à une intervention corrective ,entre le moment d'apparition et la remise a niveau de l'equipement défaillant pour pouvoir fonctionner correctement le service d'Ordonancement a besoins d'informations provenant du bureau des méthodes ces principale

1.3 Mission de l'ordonancement :

prévoir la chronologie du déroulement des tàches ,optimiser l'utilisation des moyens nécessaires. et les rendre disponibles, Lancer les travaux au mement choisi, controler l'avancement el la fin des taches , et prendre en comptr les écartes entre prévisions et réalisations.

1.4 Les trois étapes de l'ordonancement :

1.4.1 la planification :

qui vise à déterminer les différents operations à réaliser , les dates correspondantes , et les moyens materiels et humains à Yeffecter

1.4.2 L'exécution :

qui consiste à la mise en oeure des différentes opérations définis dans la phase de planification

1.4.3 Le controle :

qui consiste à effectuer une comparaison entre planification et excution ,soit au niveau des couts , soit au niveau des dates de réalisation

1.5 Analyse de temps en maintennance :

1.5.1 préparation :

c'est le délais de sous traitance, budget,état.

1.5.2 Ordonnancement :

planning charge repartition,de travail le temps

1.5.3 intervention :

le temps passé est un delais à réspecter.

1.6 Les données d'un problème d'ordonnancement :

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

Ainsi, étant donnés un ensemble de tâches et un ensemble de ressources, il s'agit de programmer les tâches et affecter les ressources de façon à optimiser un ou plusieurs objectifs (un objectif correspondant à un critère de performance), en respectant un ensemble de contraintes.

1.6.1 Les tâches :

Une tâche est une entité élémentaire de travail localisée dans le temps par une date de début et une date de fin d'exécution et qui consomme des ressources avec des quantités déterminées. Un coût (ou poids) est attribué à une tâche pour estimer sa priorité, son degré d'urgence, ou son coût d'immobilisation dans le système.

On distingue deux types de tâches :

- les tâches morcelables (préemptives) qui peuvent être exécutées en plusieurs fois,facilitant ainsi la résolution de certains problèmes,

- les tâches non morcelables (indivisibles) qui doivent être exécutées en une seule foiset ne sont interrompues qu'une fois terminées.

On note en général N= $\{J_1, J_2, \ldots, J_n\}$ l'ensemble des tâches, chaque tâche est caractérisée par :

- La durée opératoire de la tâche i sur la machine j : (p_{ij}).
- La date de disponibilité de la tâche i : (r_i).
- La date de début d'exécution de la tâche i :(s_i).
- Ladate de fin d'exécution de la tâche i :(C_i).
- Ladate d'achèvement souhaitée de la tâche i :(d_i).
- Le facteur de priorité ou poids de la tâche i : (w_i).
- Le retard algébrique de la tâche i : $(L_i = C_i - d_i)$.
- Le retard vrai de la tâche i : $(T_i = \max (C_i - d_i, 0))$.
- L'indicateur de retard de la tâche i : ($U_i = 1$, si $T_i > 0$, $U_i = 0$, si non.

1.6.2 Les ressources :

Une ressource est un moyen technique ou humain, destiné à être utilisé pour la réalisation d'une tâche et disponible en quantité limitée . On note en général M= $\{M_1, M_2, \ldots, M_m\}$ l'ensemble des ressources. Plusieurs types de ressources sont à distinguer. :

Les ressources renouvelables :

Une ressource est dite renouvelable, si après avoir été utilisée par une tâche ou allouée à une tâche, elle redevient disponible pour les autres tâches en mêmequantité. La quantité disponible est renouvelée d'une tâche à une autre, elle peutêtre constante, comme elle peut varier d'une tâche à une autre. Exemples deressources renouvelables : les machines, les hommes, l'équipement, lesprocesseurs, les fichiers...

On distingue deux types de ressources renouvelables :

- les ressources disjonctives (ou non partageables) : qui ne peuventexécuter qu'une tâche à la fois (machine, robot) ;

- les ressources cumulatives (ou partageables) : qui peuvent être utilisées par plusieurs tâches simultanément (équipe d'ouvriers).

Les ressources non renouvelables :

Une ressource est non renouvelable (ou consommable), si après avoir étéutilisée par une tâche, elle n'est plus disponible pour les autres tâches. La consommation globale en cours du temps est limitée, on dit que la ressource est épuisée en l'utilisant. Exemples de ressources consommables : le capital (oubudget), le carburant, l'énergie dans une batterie, la matière première,...

Les ressources non renouvelables sont généralement produites dans un système indépendant de la machine sur laquelle les tâches sollicitant cetteressource sont exécutées. Toutefois, certains problèmes peuvent exister oùcertaines tâches produisent des ressources qui peuvent être consommées plus tard par d'autres tâches.

Les ressources non renouvelables peuvent également être stockées dans un ou plusieurs dépôts (entrepôts, magasins ou warehouses) ayant une capacité destockage. Les ressources peuvent éventuellement avoir un stock initial dans un ou plusieurs dépôts.

Durant la consommation des ressources, la condition générale qui doit toujours être satisfaite est que la quantité totale de la ressource demandée par lestâches ne doit pas dépasser la quantité totale disponible.

1.6.3 Les contraintes :

Suivant la disponibilité des ressources et suivant l'évolution temporelle, deux types de contraintes peuvent être distingués : contraintes de ressources et contraintes temporelles.

- les contraintes de ressources : plusieurs types de contraintes peuvent être induits par la nature des ressources. A titre d'exemple, la capacité limitée d'une ressource implique un certain nombre, à ne pas dépasser, de tâches à exécuter sur cette ressource.

Les contraintes relatives aux ressources peuvent être disjonctives, induisant une contrainte de réalisation des tâches sur des intervalles temporels disjoints pour une même ressource, ou cumulatives impliquant la limitation du nombre de tâches àréaliser en parallèle.

- les contraintes temporelles : elles représentent des restrictions sur les valeurs que peuvent prendre certaines variables temporelles d'ordonnancement. Ces contraintes peuvent être :

- des contraintes de dates butoirs, certaines tâches doivent être achevées avantune date préalablement fixée, des contraintes de précédence, une tâche i doit précéder la tâche j,

- des contraintes de dates au plus tôt, liées à l'indisponibilité de certains facteurs nécessaires pour commencer l'exécution des tâches.

1.6.4 les critéres (fonction objectif) :

Un critère correspond à des exigences qualitatives et quantitatives à satisfaire permettantd'évaluer la qualité de l'ordonnancement établi.Les critères dépendant d'une application donnée sont très nombreux ; plusieurs critèrespeuvent être retenus pour une même application. Le choix de la solution la plus satisfaisantedépend du ou des critères préalablement définis, pouvant être classés suivant deux types,réguliers et irréguliers.

Les différents critères ne sont pas indépendants ; certains même sont équivalents. Deuxcritères sont équivalents si une solution optimale pour l'un est aussi optimale pour l'autre et inversement.

Les critères réguliers constituent des fonctions décroissantes des dates d'achèvementdes opérations. Quelques exemples sont cités ci-dessous :

- la minimisation des dates d'achèvement des actions,

- la minimisation du maximum des dates d'achèvement des actions,

- la minimisation de la moyenne des dates d'achèvement des actions,

- la minimisation des retards sur les dates d'achèvement des actions,

- la minimisation du maximum des retards sur les dates d'achèvement des actions.

Les critères irréguliers sont des critères non réguliers, c'est-à-dire qui ne sont pas

desfonctions monotones des dates de fin d'exécution des opérations, tels que :

- la minimisation des encours,

- la minimisation du coût de stockage des matières premières,

- l'équilibrage des charges des machines,

- l'optimisation des changements d'outils.

La satisfaction de tous les critères à la fois est souvent délicate, car elle conduit souvent à dessituations contradictoires et à la recherche de solutions à des problèmescomplexes d'optimisation.

Les critères lesplus utilisés font intervenir la durée totale, les retards et le coût des stocks desencours.

Minimiser la durée totale : La durée totale de l'ordonnancement notée C_{max} est égale à la date d'achèvement de la tâche la plus tardive : c_{max}=max (c_i). Minimiser la durée totale, revient à minimiser C_{max} c'est-à dire min (max (c_i)).

Minimiser les retards : On peut rencontrer plusieurs problèmes dans lesquels il faut respecter les délais d_i. Les critères retenus sont les retards vrais T_{max}= max(T_i) et les retards algébriques L_{max} = max(L_i). Parfois on retient le nombre de tâches en retard $\sum U_i$

Minimiser les encours : Les encours sont déterminés par le temps de présence des tâches dans le système : $F_i = C_i$ - r_i. Le critère essentiel est de minimiser la somme de ces temps $\sum_{i=1}^{n} F_i$. Lorsqu'on tient compte des coûts de présence des tâches dans le système ou de leur importance, on attache un poids wi au temps de présence, le critère devient $\sum_{i=1}^{n} wiFi$. Puisque les dates de disponibilité r_i sont fixées (constantes), on retient seulement le critère $\sum_{i=1}^{n} Ci$ (somme de la date de fin d'exécution) et le critère $\sum_{i=1}^{n} wiCi$ (somme pondérée des dates de fin d'exécution).

1.7 Classification des problèmes d'ordonnancement :

un des paramétres primodiaux pour la génération d'un ordonnancement optimale dans les problémes d'ordonnancement job shop, est l'exploitation maximale du temps, car un simple décalage des opérations retardées peut éviter les trous de temps (temps morts) afin de préparer un or

donnancement (compact). en effet, plus un ordonnancement est compact, meilleure est sa qualité

cette notion de compacité qui est un objectif majeur de tout ordonnancement est la base de distinction de plusieurs classes d'ordonnancement : admissible, semi-actif , actif, sans délai

1.7.1 ordonnancement admissible :

si toutes les contraintes du probleme sont bien respectées ,l'ordonnancement est dite :(admissible)

dans certains cas , des décalage à gauche sur certaines opérations sont nécessaires. selon que l'ordre des opérations reste inchangé ou non, nous distinguons deux cas :

1 décalage gauche locale :l'anvancement du début d'une opération ne remet pas en cause l'ordre des autre opérations

2 décalage gauche globale :l'anvancement du début d'une opération engendre une modification au niveau de l'ordre relatif aux deuxopérations au minimum

1.7.2 ordonnancement semi-actif :

si aucun décalage locale n'est possible , l'ordonnancement est dit semi- actif donc aucune opération ne peut étre exécutée en plus tot sans modifier l'ordre relatif au moins de deux opérations

1.7.3 ordonnancement actif :

si aucun décalage à gauche que se soit local ou global n'est pas possible , l'ordonnancement est dit actif en concéquence il est impossible d'avancer une opération sans reporter le début d'une autre opération

1.7.4 ordonnancement sans délais :

un ordonnancement est dit sans délai ou retard , si et selement si aucune opération n'est mise en attente alors qu'une machine est disponible pour l'exécuter

A noter que la transformation en un ordonnacement sans délai peut mener à une solution plus mauvaise du point de vue makespan

il existe une relation d'inclution entre les différentes classes d'ordonnancement précédentes

1.8 Notion de complexité de problèmes :

La théorie de la complexité a pour but d'apporter des informations sur la difficulté théorique d'un problème à résoudre. Elle permet de classer "du point de vue mathématique"les problèmes selon leur difficulté

La complexité analyse le temps nécessaire pour obtenir une solution (on peut également s'intéresser à la mémoire nécessaire, mais nous ne nous intéresserons pas ici à cet aspect de la complexité). On peut considérer la durée moyenne ou la durée dans le pire des cas. Nous traitons essentiellement ce dernier cas.

A première vue, comment définir un algorithme efficace Pour un problème donné ? chercher un algorithme efficace, veut dire trouver un algorithme où le temps nécessaire à son exécution ne soit pas trop important.Un problème est dit facile si on peut le résoudre facilement, c'est-à-dire s'il ne fait pas trop de temps pour arriver à la solution.Donc, s'il existe un algorithme efficace pour un problème donné, alors ce dernier est dit facile.Un problème pour lequel on ne connaît pas d'algorithme efficace, alors ce dernier est dit diffi-

cile Pour résoudre un problème d'ordonnancement, il ne suffit pas de prouver l'existence d'une solution, il faut également la construire, il est clair que construire la solution est plus difficile que de prouver son existence ce qui nous conduit donc à classer les problèmes comme étant difficiles ou faciles.

Les problèmes indécidables sont ceux pour lesquels aucun algorithme, quel qu'il soit, n'a été trouvé pour les résoudre.A l'opposé, les problèmes décidables sont ceux pour lesquels il existe au moins un algorithme pour les résoudre

1.8.1 La classe NP :

La classe NP (Non déterministe Polynomial) est celle des problèmes d'existence dont une proposition de solution est Oui et qui est vérifiable polynomialement. Parmi les problèmes décidables, les plus simples à résoudre sont regroupés dans la classe NP.

1.8.2 La classe P :

Un problème est dit polynomial ,s'il existe un algorithme de complexité polynomiale-permettant de répondre à la question posée dans ce problème, quelle que soit la donnée de celui-ci La classe P est l'ensemble de tous les problèmes de reconnaissance polynomiaux.Pour le reste de la classe NP, on n'est pas sûr qu'il n'existe pas un algorithme polynomialpour résoudre chacun de ses problèmes. Ainsi, on sait que P est inclue dans NP mais on n'a paspu prouver que P n'est pas NP.

1.8.3 La classe NP-Complet :

La classe NP-Complet regroupe les problèmes les plus difficiles de la classe NP. Elle-contient les problèmes de la classe NP tels que n'importe quel problème de la classe NP leur estpolynomialement réductible. Entre eux, les problèmes de la classe NP-Complet sont aussi difficiles.

1.8.4 La classe NP-Difficile :

La classe NP-Difficile regroupe les problèmes (pas forcément dans la classe NP) tels que n'importe quel problème de la classe NP leur est polynomialement réductible.

Pour les problèmes d'ordonnancement à une machine, certains peuvent êtrerésolus par un algorithme polynomial, certains autres sont démontrésNP-difficiles. Certains ne sont ni démontrés NP-difficiles, ni polynomialementrésolvables. Ils restent donc ouverts. Et pour les problèmes d'ateliers, la plupart de ces problèmes ont été démontrés NP-difficiles

1.9 Problème d'optimisation :

Un problème d'optimisation se définit comme la recherche, parmi un ensemble de solution possibles s (appelé aussi espace de décision ou espace de recherche), la (ou des solution(s)) x^*qui rend(ent) minimale (ou maximale) une fonction mesurant la qualité de cette solution.Cette fonction est appelée fonction objectif ou fonction coût. Si l'on pose $f : s \rightarrow \mathbb{R}$ la fonction objectif à minimiser (respectivement à maximiser) à valeurs dans $\mathbb{R}$, le problème revient alors à trouver l'optimum $x^* \in s$ tel que f(x^*) soit minimal (respectivement maximal).

Lorsque l'on veut résoudre un problème d'optimisation, on recherche la meilleure solution possible à ce problème, c'est-à-dire l'optimum global. Cependant, il peut exister des solutionsintermédiaires, qui sont également des optimums, mais uniquement pour un sous-espace restreintde l'espace de recherche : on parle alors d'optimums locaux. La seule hypothèse faite sur s est qu'il s'agit d'un espace topologique, i.e.sur lequel est définie une notion de voisinage. Cette hypothèse est nécessaire pour définir lanotion de solutions locales du problème d'optimisation. On peut alors définir un optimum local(relativement au voisinage V) comme la solution x^* de s telle que $f\ (\ x^*)\ \leq\ f\ (x)\,;\forall\ x \in\ \mathrm{v}\ (\ \ x^*)$

1.10 conclusion :

L'ordonnancement est généralement décrit comme une fonction particulière de décision au sein d'un système de gestion du travail concernant la production de bien, d'ouvrages ou de services.

La majorité des problèmes d'ordonnancement sont NP-difficile, ça veut dire que, dans la pratique, la complexité croît exponentiellement avec le nombre de tâches et de ressources. Il n'est, donc, pas envisageable de résoudre de tels problèmes avec les méthodes exactes. C'est pour cela qu'il faut développer des nouveaux méthodes qui donnent des solutions certes sous-optimales, maisobtenues rapidement. dont l'objectif est de fournir des solutions aussi proches que possible de la solution exacte en un temps raisonnable.

Chapitre 2

méthode de résolution des problémes d'ordonnancement

2.1 introduction :

Nous avons vu précédemment qu'il existe des problèmes d'ordonnancementde complexités différentes. Les problèmes appartenant à la classe P ont des algorithmes polynomiaux permettant de les résoudre.Pour les problèmes appartenant à la classe NP, l'existence d'algorithmes polynomiaux semble peu réaliste. Ainsi, différentes méthodes de résolution sont largement utilisées pour appréhender les problèmes NP-difficiles.

Dans ce chapitre, nous exposons en générale les méthodes de résolution des problèmes d'optimisation qui sont classées en deux grandes catégories : les méthodes exactes et les méthodes approchées. On explique brièvement les méthodes les plus connues dans chaque catégorie

2.2 Les méthodes d'optimisation :

Comme on l'a rappelé dans le paragraphe sur la complexité, la plupart des problèmes d'ordonnancement sont NP-difficiles. On ne peut donc pas espérer trouver un ordonnan-

cement optimal en un temps raisonnable pour des problèmes de taille industrielle. Les méthodes utilisées sont donc des méthodes approchées.

Pour des problèmes de petite taille, nous pouvons obtenir une solution exacte.Une méthode exacte peut servir pour résoudre des sous-problèmes d'un problème de grande taille de manière optimale.

2.2.1 Les méthodes exactes :

Les méthodes exactes utilisent surtout deux approches de résolution très connues : la programmation dynamique et les procédures par séparation et évaluation. Ces méthodes sont souvent utilisées pour résoudre les problèmes combinatoires de manière exacte, en ordonnancement tout particulièrement. Ce sont des méthodes d'énumération implicite.

L'énumération explicite construit toutes les solutions réalisables et retient une parmi les meilleures.L'énumération implicite consiste à explorer l'ensemble de toutes les solutions réalisables en éliminant des sous-ensembles de solutions moins intéressants sans avoir à les construire. Nous pouvons citer trois approches particulièrement célèbres : La programmation dynamique introduite par Bellman dans les années 504 , la méthode par séparation et évaluation (Branch and Bound en anglais : notée B&B) et l'algorithme de retour arrière (Backtracking).

2.2.2 les méthodes approchés :

Une méthode approchée est une méthode d'optimisation qui a pour but de trouver une solution réalisable de la fonction objectif en un temps raisonnable, mais sans garantie d'optimalité. L'avantage principal de ces méthodes est qu'elles peuvent s'appliquer à n'importe quelle classe de problèmes, faciles ou très difficiles, d'un autre côté les algorithmes d'optimisation tels que les algorithmes de recuit simulé, les algorithmes tabous et les algorithmes génétiques ont démontré leurs robustesses et efficacités face à plusieurs problèmes d'optimisation combinatoires.

Les méthodes approchées englobent deux classes : les heuristiques et les métaeuristiques. La particularité qui différencie les méthodes métaheuristiques des méthodes heuristiques c'est que les métaheuristiques sont applicables sur de nombreux problèmes. Tandis que, les heuristiques sont spécifiques à un problème donné.

Les heuristiques :

Les méthodes heuristiques sont des méthodes spécifiques à un problème particulier. Elles nécessitent des connaissances du domaine du problème traité. En fait, se sont des règles empiriques qui se basent sur l'expérience et les résultats acquis afin d'améliorer les recherches futures. Plusieurs définitions des heuristiques ont été proposées par plusieurs chercheurs dans la littérature, parmi les quelles :

Définition : Une heuristique (règle heuristique, méthode heuristique) est une règle d'estimation, une stratégie, une astuce, une simplification, ou tout autre type de dispositif qui limite considérablement la recherche de solutions dans des espaces problématiques importants. Lesheuristiques ne garantissent pas des solutions optimales. En fait, elles ne garantissent pas une solution du tout. Tout ce qui peut être dit d'une heuristique utile, c'est qu'elle propose des solutions qui sont assez bonnes la plupart du temps.

Définition : Une méthode heuristique (ou simplement une heuristique) est une méthode qui aide à découvrir la solution d'un problème en faisant des conjectures plausibles mais faillible de ce qui est la meilleure chose à faire.

Définition : Les heuristiques sont des ensembles de règles empiriques ou des stratégies qui fonctionnent, en effet, comme des règles d'estimation.

Les métaheuristiques :

Les métaheuristiques d'optimisation sont des algorithmes générauxd'optimisation applicables à une grande variété de problèmes. Elles sont apparues à partir des années 80,

dans le but de résoudre au mieux des problèmes d'optimisation. Les métaheuristiques s'efforcent de résoudre tout type de problème d'optimisation. Elles sont caractérisées par leur caractère stochastique.Ainsi que par leur origine discrète.

Elles sont inspirées par desanalogies avec la physique (recuit simulé, recuit micro-canonique), avec labiologie (algorithmes évolutionnaires) ou encore l'éthologie (colonies defourmis, essaims particulaires). Cependant, elles ont l'inconvénient d'avoir plusieurs paramètres à régler. Il est à souligner que les métaheuristiques se prêtent à toutes sortes d'extensions, notamment en optimisation mono-objectif et multi-objectif.

Les métaheuristiques utilisent des recherches stratégiques afin d'explorer plus efficacement l'espace de recherche, et souvent se focalisent sur les régions prometteuses. Ces méthodes commencent par un ensemble de solutions initiales ou une population initiale, et après, elles examinent étape par étape une séquence de solutions pour atteindre, oude s'approcher de la solution optimale du problème. Les métaheuristiques ont plusieursavantages par rapport aux algorithmes traditionnels. Les deux avantages les plus importants sont la simplicité et la flexibilité. Les métaheuristiques sont souvent simples à implémenter, pourtant, elles sont capables de résoudre des problèmes complexes avec la capacité de s'adapter à plusieurs problèmes d'optimisation du monde réel, à partir du domaine de la recherche opérationnelle, d'ingénierie vers l'intelligence artificielle.

De nos jours les gestionnaires et les décideurs sontconfrontés quotidiennement à des problèmes de complexité grandissante,qui surgissent dans des secteurs très divers. Le problème à résoudre peutsouvent s'exprimer sous la forme générale d'un problème d'optimisation, dans lequeL

on définit une ou plusieurs fonctions objectif que l'on chercheà minimiser ou à maximiser par rapport à tous les paramètres concernés

Le mot métaheuristique est dérivé de la composition de deux mots grecs :

- heuristique qui vient du verbe heuriskein et qui signifie 'trouver'.

- méta qui est un suffixe signifiant 'au -delà', 'dans un niveau supérieur'.

Les propriétés fondamentales des métaheuristiques : - Les métaheuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale.

- Le but visé par les métaheuristiques est d'explorer l'espace de recherche efficacement afin de déterminer dessolutions (presque) optimales.

- Les techniques qui constituent des algorithmes de type métaheuristique vont de la simple procédure derecherche locale à des processus d'apprentissage complexes.

- Les métaheuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité

- Les métaheuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régionsde l'espace de recherche.

- Les concepts de base des métaheuristiques peuvent être décrit de manière abstraite, sans faire appel à unproblème spécifique.

- Les métaheuristiques peuvent faire appel à des heuristiques qui tiennent compte de la spécificité du problèmetraité, mais ces heuristiques sont contrôlées par une stratégie de niveau supérieur.

- Les métaheuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum, pourmieux guider la suite du processus de recherche

Classification des métaheuristiques : On peut regrouper les métaheuristiques en deux grandes classes : les métaheuristiques à solution unique (c.à.d. évoluant avec une seule solution) et celles à solutions multiples ou population de solutions. Les méthodes d'optimisation à population de solutions améliorent, au fûr et à mesure des itérations, une population de solutions. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité.

Les métaheuristiques à solution unique : Dans cette section, nous présentons les métaheuristiques à base de solution unique, aussi appelées méthodes de trajectoire.

Contrairement aux métaheuristiques à base de population, lesmétaheuristiques à solution unique commencent avec une seule solution initiale et s'en éloignentprogressivement, en construisant une trajectoire dans l'espace de recherche. Les méthodes detrajectoire englobent essentiellement la méthode de descente, le recuit simulé, la recherche tabou, la recherche à voisinage variable, la méthode GRASP, la recherche locale itérée, la recherche locale guidée et leurs variantes.

Les méthodes de descente (DM : Descent method) : Ces méthodes s'articulent toute sautour d'un principe simple. Partir d'une solution existante, chercher une solution dans le voisinage et accepter cette solution si elle améliore la solution courante.

L'algorithme 1 présente le squelette d'une méthode de descente simple (Simple descent). Apartir d'une solution initiale $\varkappa$, on choisit une solution $\varkappa\prime$ dans le voisinage $N(\varkappa)$ de $\varkappa$. Si cette solution est meilleure que $\varkappa$,($f\ (\varkappa') < \ f(\varkappa)$) alors on accepte cette solution comme nouvelle solution $\varkappa$ et on recommence le processus jusqu'à ce qu'il n'y ait plus aucune solution améliorante dans le voisinage de $\varkappa$

algorithme :1 simple descente

1 :initialise :find an initial solution $\varkappa$

2 :repeat

3 :neighbourhood search :find a solution $\varkappa\prime$

4 : if $f(\varkappa') \prec \ f(\varkappa)$ then

5 : $\varkappa\prime \longleftarrow \varkappa$

6 : end if

7 : until $f\ (y) \succeq \ f(\varkappa), \forall \quad y \ \epsilon\, N(\varkappa)$

Une version plus agressive de la méthode de descente est la méthode deplus grande descente (Deepest descent). Au lieu de choisir une solution $\varkappa\prime$dans le voisinage de $\varkappa$,on choisit toujours la meilleure solution $\varkappa\prime$du voisinage de $\varkappa$.

l'algorithme 2 donne une description de cette méthode.

algorithme 2 :deepest descente

1 :initialise : find an initial solution $\varkappa$

2 :repeat

3 : neighbourhood search : find a solution $\varkappa\prime \in N(x)/f(x') \leq f(x''), \forall x'' \in N(x)$

4 : if $f\ (x') \prec\ f\ (x)$ then

5 :$x' \longleftarrow x$

6 :end if

7 :until $f\ (x') \geq\ \ f\ (x), \forall x' \in N(x)$

Ces deux méthodes sont évidemment sujettes a de nombreuses critiques. Ellesse basent toutes les deux sur une amélioration progressive de la solution et doncresteront bloquées dans un minimum local des qu'elles en rencontreront un. Il existe de manière évidente une absence de diversification. L'équilibre souhaité entre intensification et diversification n'existe donc plus et l'utilisateur de ces deux méthodes doit en être conscient.

Un moyen très simple de diversifier la recherche peut consister à re-exécuterun des algorithmes en prenant un autre point de départ. Comme l'exécution de ces méthodes est souvent très rapide, on peut alors inclure cette répétition au sein d'une boucle générale. On obtient alors un algorithme de type "Multi-start descent" décrit par l'algorithme 3.

algorithme3 :multi- start descente :

1 :initialise : find an initial solution $x, k \longleftarrow 1, f(B) \longleftarrow +\infty$

2 :repeat

3 :starting point :choose an initial solution $x_0$at random

4 : $x \longleftarrow$ result of simple Descent or Deepest Descent

5 : if $\ \ f\ (x) \prec\ f\ (B)$ then

6 : $\ B \longleftarrow x$

7 : end if

8 :$k \longleftarrow k + 1$

9 : unitl stopping criterion statisfied

De manière évidente, la diversification esttotalement absente des algorithmes 1 et 2. En ne conservant que l'aspect intensification,la convergence est souvent trop rapide et

on se trouve très rapidement bloqué dans un optimum local. Les résultats communément admis indiquent que ces techniques conduisent engénéral à des solutions en moyenne à 20% de l'optimum. Dans le cas de l'algorithme 3, la diversification est simplement insérée par le choix aléatoire d'une solution de départ

La recherche tabou (TS : Tabu Search) : La recherche tabou est une méta-heuristique originalement développée parGlover, 1986 et indépendamment par Hansen, 1986 . Elle estbasée sur des idées simples, mais elle est néanmoins très efficace. Cette méthodecombine une procédure de recherche locale avec un certain nombre de règles et demécanismes permettant à celle-ci de surmonter l'obstacle des optima locaux, touten évitant de cycler. Elle a été appliquée avec succès pour résoudre de nombreuxproblèmes difficiles d'optimisation combinatoire : problèmes de routage devéhicule, problèmes d'ordonnancement, problèmes de coloration de graphes, etc.

Dans une première phase, la méthode de recherche tabou peut être vuecomme une généralisation des méthodes d'amélioration locales. En effet, en partantd'une solution quelconque $\mathscr{x}$ appartenant à l'ensemble de solutions S, on se déplacevers une solution $\mathscr{x}\prime$ située dans le voisinage $N(x)$. Donc l'algorithme exploreitérativement l'espace de solutions S.

algorithme 4 :TS

1 :initialise : find an initial solution x

2 :repeat

3 : neighbourhood search : find a solution $\mathscr{x}\prime \in N^*(x)$

4 :update memory : tabu list ,frequency-based memory, aspiration level,...

5 :move $x \longleftarrow x'$

6 : unitl stopping criterion statisfied

Afin de choisir le meilleur voisin $\mathscr{x}\prime$ dans $N(\mathscr{x})$, l'algorithme évalue lafonction objectif f en chaque point $\mathscr{x}\prime$, et retient le voisin qui améliore la valeur dela fonction objectif, ou au pire celui qui la dégrade le moins

Les métaheuristiques à population de solutions : Contrairement aux algorithmes partant d'une solution singulière, les métaheuristiques à population de solutions améliorent, au fur et à mesure des itérations, une population de solutions.On distingue dans cette catégorie, les algorithmes évolutionnaires, qui sont une famille d'algorithmes issus de la théorie de l'évolution par la sélection naturelle, énoncée par Charles Darwin et les algorithmes d'intelligence en essaim qui, de la même manièreque les algorithmes évolutionnaires, proviennent d'analogies avec des phénomènes biologiquesnaturels.

Les algorithmes génétiques (GA :Genetic Algorithm) : Proposé dans les années 1975 par Holland, les algorithmes génétiques doivent leur popularité à Goldberg. Avant la parution de son livre qui est une des références les plus citées dans le domaine de l'informatique, on a puvoir un certain nombre d'autres présentations, citons Goldberg , Holland, Schwefel Le sujet connaît une très grande popularité. Il existe aujourd'huiplusieurs milliers de références sur le sujet et le nombre de conférences dédiéesau domaine (que ce soit sur les techniques elles-mêmes ou sur les applications)ne fait qu'augmenter.

De manière générale, les algorithmes génétiques utilisent un même principe.Une population d'individus (correspondants à des solutions) évoluent en même temps comme dans l'évolution naturelle en biologie. Pour chacun des individus, on mesure sa faculté d'adaptation au milieu extérieur par le fitness.

Les algorithmes génétiques s'appuient alors sur trois fonctionnalités :

-La Sélection : Pour déterminer quels individus sont plus enclins à se reproduire, une sélection est opérée. Il existe plusieurs techniques de sélection, les principales utilisées sont la sélection par tirage à la roulette (roulette-wheel selection), la sélection par tournoi (tournament selection), la sélection par rang (ranking selection), etc

- Le Croisement : L'opérateur de croisement combine les caractéristiques d'un ensemble d'individus parents (généralement deux) préalablement sélectionnés, et génère de nouveaux individus enfants. Là encore, il existe de nombreux opérateurs de croisement, par exemple le croisement en un point, le croisement en n-points ($n \geq 2$) et le croisement

uniforme 5

-La Mutation et le remplacement : Les descendants sont mutés, c'est-à-dire que l'on modifie aléatoirement une partie de leur génotype, selon l'opérateur de mutation.Le remplacement (ou sélection des survivants), comme son nom l'indique, remplace certains des parents par certains des descendants. Le plus simple est de prendre les meilleurs individus de la population, en fonction de leurs performances respectives, afin de former une nouvelle population (typiquement de la même taille qu'au début de l'itération).

La représentation des solutions (le codage) est un point critique de la réussite d'un algorithme génétique. Il faut bien sûr qu'il s'adapte le mieux possible au problème et à l'évaluation d'une solution. Le codage phénotypique ou codage direct correspond en général à une représentation de la solution très proche de la réalité. L'évaluation d'une solution représentée ainsi est en général immédiate.

L'algorithme 3 propose une version qualifiée en Anglais par population replacement (ou algorithme générationnel)

algorithme 5 :GA

1 :initial : generate an initial population p of solution with size $|p| - n$

2 : repeat

3 : $p' \longleftarrow \theta$

4 : repeat θ

5 :selection : choose 2 solution x and x' from p with propability proportional to their fitness

6 : crossover :combine parent solution x and x' to from child solutions y and $y'$with high propability

7 :mutate y and y' with smal propability

8 :add y and y' to p'

9 :until $|p'| = n$

10 : $p \longleftarrow p'$

11 : until stopping criterion

Algorithme de colonies de fourmis (ACO : Ant Colony Optimization) : Les algorithmes de colonies de fourmis ont été proposés par Colorni,Dorigo et Maniezzo en 1992 et appliquées la première fois au problème du voyageur de commerce. Ce sont des algorithmes itératifs à population où tous les individus partagent un savoir commun qui leur permet d'orienter leurs futurs choix et d'indiqueraux autres individus des choix à suivre ou à éviter. Le principe de cette métaheuristique repose sur le comportement particulier des fourmis, elles utilisent pour communiquer une substance chimique volatile particulière appelée phéromone grâce à une glande située dans leur abdomen. Enquittant leur nid pour explorer leur environnement à la recherche dela nourriture (Food), les fourmis arrivent à élaborer des chemins qui s'avèrentfréquemment être les plus courts pour aller du nid vers une source de nourriture. Chaque fourmi dépose alors une quantité de phéromonessur ces pistes qui deviendront un moyen de communication avec leurscongénères, les fourmis choisissent ainsi avec une probabilité élevée les chemins contenant les plus fortes concentrations de phéromones à l'aide des récepteurs situés dans leurs antennes.

des figures illustre et confirme ce constat, une expérience a été faite par Gauss et Deneubourg en 1989 appelée expérience du pont à double branche, les fourmis qui retournent au nid rapidement, après avoir visité la source de nourriture, sont celles qui ont choisi la branche courteet les fourmis empruntant cette branche faisant plus d'aller retour, et par conséquent la quantité de phéromones déposée sur la plus courte branche est relativement supérieure que celle présente sur la plus longue branche. Puisque les fourmis sont attirées plus vers les pistes de plus grande concentration en phéromones, alors la branche courte sera la plus empruntée par la majorité des fourmis.

algirithme 6 :ACO

1 :unitialise :create an initial population of ants

2 :repeat

3 : for each and do

4 : construct a solution based on the construction procedure , biased by the pheromone

trails

5 : update the pheromone trails based on the quality of the solutions found

6 : end for

7 : until stopping criterion satisfied

2.3 conclusion :

Dans ce chapitre j'ai essayé de présenter les principes et les algorithmes de différentes méthodes de résolution de problèmes d'optimisation commençant par les méthodes exactes aux méthodes approchées. j'ai constaté que les méthodes exactes permettent d'aboutir à la solution optimale, mais elles sont trop gourmandes en termes de temps de calcul et d'espace mémoire requis. Cependant, les méthodes approchées demandent des coûts de recherche raisonnables. Mais, elles ne garantissent pas l'optimalité de la solution. j' ai pu constater que les méthodes approchées peuvent être partagées en deux classes : des méthodes heuristiques et des méthodes métaheuristiques. Une méthode heuristique est applicable sur un problème donné. Tandis qu'une méthode métaheuristique est plus générique et elle peut être appliquée sur plusieurs problèmes d'optimisation. En outre, j'ai constaté que les méthodes métaheuristiques peuvent être partagées en deux sous classes : des méthodes à base d'une solution unique et des méthodes à base de population de solutions. Les méthodes de la première sous classe (i.e. les méthodes à base d'une solution unique) se basent sur la recherche locale pour trouver la solution du problème à traiter. Elles sont souvent piégées par l'optimum local d'un voisinage donné. Par contre, les méthodes de la deuxième classe (i.e. les méthodes à base de population de solutions) se basent sur une recherche globale ce qui leur permet d'échapper au problème de la convergence vers l'optimum global et augmente leur possibilité de fournir des solutions de bonnes qualités , l 'existence de plusieurs methodes a pour but la résolution des problémes d'odonnancement .

Chapitre 3

quelques types des problèmes d'ordonnancement avec des exemples

3.1 introduction :

Les problèmes d'ordonnancement sont généralement classés en deux principaux modèles dépendamment du nombre d'opérations que requièrent les jobs : des modèles à une opération (machine unique et machines parallèles) et des modèles à plusieurs opérations (flow-shop, open shop et job shop) dans ce chapitre on va expliquer le modéle à une opération .

3.2 Modèles à une opération :

3.2.1 Modèle à machine unique :

Dans un modèle à machine unique, l'ensemble des tâches à réaliser est exécuté par une seule 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.

3.2.2 Modèle à machines parallèle :

Ce modèle est utilisé surtout dans les secteurs industriels tels que : l'industrie alimentaire, les industries plastiques, les fonderies et en particulier l'industrie textile. Le processus de déroulement de ce système de production est le suivant : à chaque fois qu'une machine i se libère, ont lui affecte un job j . Dans le cas d'un processus d'assemblage industriel,

definition :

Le principe des problèmes d'ordonnancement à machines parallèles est Chaque job est constitué d'une seule opération et chaque opération peut être réalisée par n'importe laquelle des machines, disposées en parallèles, mais n'en nécessite qu'une seule. On considère dans toute cette section le critère de minimisation de la durée totale. Dans le cadre de l'informatique on peut ainsi modéliser les « m » processeurs d'une machine parallèle. Ces problèmes sont presque toujours NP-difficiles.

Makespan (cmax) :

Le makespan représente le temps de fin d'exécution du dernier job dans une séquence. Il est l'un des critères les plus utilisés pour évaluer le coût d'un ordonnancement. En minimisant ce critère, on peut améliorer le rendement et réduire le temps moyen d'inactivité des machines. La minimisation du makespan s'accompagne généralement de contraintes qui peuvent être temporelles ou liées aux ressources. Les contraintes temporelles se divisent en deux catégories : des contraintes de temps alloué (impératif de gestion : délai de livraison, disponibilité, achèvement) et des contraintes d'antériorité (cohérence technologique : gammes de fabrication, inégalité de potentiels : précédence). Les contraintes

liées aux ressources peuvent être des contraintes disjonctives (une tâche i doit s'exécuter avant ou après une tâche}) ou des contraintes cumulatives (respect des capacités des ressources).

On peut aussi considérer d'autres critères de performance, tels que le temps moyen d'achèvement des jobs, le temps total de traitement, le temps de retard total, le temps d'attente des jobs, le taux d'occupation de machines, le nombre de jobs en retard, le temps de séjour d'un job dans le système avant sa réalisation, etc.

Les différents modèles :

Pour permettre la classification des problèmes d'ordonnancement, ceux-ci ont été séparés en deux grands modèles suivant les tâches à exécuter

1- Les modèles dits statiques : le nombre de tâches à exécuter et leurs dates de disponibilité sont connus à l'avance et ne peuvent être modifiés par la suite,

2- Les modèles dynamiques : les tâches arrivent de façon aléatoire. Leurs nombres, ainsi que leurs dates de disponibilité ne sont pas connus à l'avance.

Les modèles mathématique des problèmes à machine parallèle :

Après les études des modèles mathématiques, les données du problème et le modèle qui s'expriment de la manière suivante :

n : le nombre de tâches.

m : le nombre de machines.

j : l'indice de la tâche, où $j = 1, ..., n$

k : l'indice de la machine, où $k = 1, ...m$.

r : la position de la tâche dans une machine, où $r = 1, ...nk$

r_{jk} : date de début.

d_{jk} : date de fin la tâche j.

s_{jk} : temps de préparation de la tâche j effectuée immédiatement après la tâche i sur une machine.

p_{jk} : temps opératoire de la tâche j.

c_{jk} : date de fin d'exécution de la tâche j.

T_{jk} : retard réel de la tâche j.

$c\max$: makespan.

n_k : nombre des tâches affectées à la machine k.

Minimiser $(c\max, \sum T\ )$ (1)

sous contraintes :

$\sum_{j=1}^{n} X_{jkr}$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (2)

$\sum_{k=1}^{m} \sum_{r=1}^{nk} X_{jkr}$ $j = 1, 2, ..., n$ (3)

$p_{[kr]} = \sum_{j=1}^{n} X_{jkr} p_{jk} r_{ij}$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (4)

$s_{[kr]} = \sum_{i=1}^{n} \sum_{j=1}^{n} X_{jkr} y_{ij} s_{ij}$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (5)

$r_{[kr]} = \sum_{j=1}^{n} X_{jkr} r_{jk}$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (6)

$d_{[kr]} = \sum_{j=1}^{n} X_{jkr} d_{jk}$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (7)

$c_{[kr]} = \max(c_{[k,r-1]} + s_{[kr]}, r_{[kr]}) + p_{[kr]}$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (8)

$T_{[kr]} = \max(c_{[kr]} - d_{[kr]}, 0)$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (9)

$c_{\max} = \max_{k=1}^{m} \max_{r=1}^{n_k} c_{[kr]}$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (10)

$\sum T_{jk} = \sum_{k=1}^{m} \sum_{r=1}^{n_k} T_{[kr]}$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k$ (11)

$X_{jkr} = 0 or 1$ $k = 1, 2, ..., m \quad r = 1, 2, ..., n_k \quad j = 1, 2, ...n$ (12)

$Y_{ij} = 0 or 1$ $i = 1, 2, ...n \quad j = 1, 2, ...n$ (13)

La fonction (1) exprime directement les objectifs de notre problème, i.e., la minimisation du makespan et la minimisation de la somme des retards.

Les contraintes (2) et (3) sont des contraintes de singularité des tâches sur la machine k et à la position r. Elles garantissent qu'il y a seulement une tâche sur la machine k et à la position r, et que chaque tâche est déplacée seulement une fois sur ces machines.

La contrainte (4) calcule la durée d'opération de la tâche qui est en position r sur la machine k. La contrainte (5) définit le temps de préparation de la tâche qui est en position r sur la machine k.

Les contraintes (6) - (9) concernent respectivement la date de début au plus tôt, la date de fin au plus tard, la date de fin d'exécution et le retard réel de la tâche qui est en

position r sur la machine k.

Les contraintes (10) et (11) représentent le calcul du makespan et de la somme des retards. La variable binaire X_{jkr} est égale à 1, si la tâche j est ordonnée en position r sur la machine k et 0 sinon. La variable binaire Y_{ij} contrôle la position relative de deux tâches i et j. Si la tâche i précède immédiatement la tâche j, alors Y_{ij} est égale à 1, sinon elle est égale à 0. Par contre, si j'est la premier tâche (j=1), alors Y_{ij} est égale à 1, car aucune tâche ne précède j .

Représentation des problèmes d'ordonnancement à machines parallèles :

Dans les problèmes d'ordonnancement à machines parallèles, on peut afficher les ordonnancements pour nous problèmes avec le diagramme de Gantt, car il est permis de visualiser les séquences des opérations des tâches, en représentant chaque tâche par une ligne sur laquelle sont visibles, les périodes d'exécution des opération et les périodes ou la tâche est en attente des ressources.

Insertion d'une tâche venant aléatoirement :

Une approche basée sur le contexte aléatoire, c'est la méthode d'insertion des tâches dans un ordonnancement prévisionnel. Cette méthode consiste à sélectionner et choisir le bon endroit (emplacement) où ces nouvelles tâches doivent être inclues. Dans la littérature on trouve que l'utilisation de la méthode d'insertion de tâches n'été pas limitée au problème d'ordonnancement d'une seule ressource, mais aussi pour plusieurs ressources (comme les ressources humaines dans les problèmes de d'ordonnancement de la maintenance).

exemple :

si l'une des étapes d'assemblage nécessite beaucoup de temps, il serait très intéressant alors d'avoir plusieurs machines parallèles qui effectuent la même tâche. D'un autre côté, les machines parallèles sont classées suivant leur rapidité. Si toutes les machines

de l'ensemble ont la même vitesse de traitement et effectuent les mêmes tâches, elles sont identiques. Si les machines ont des vitesses de traitement différentes mais linéaires alors elles sont dites uniformes. Dans le cas où les vitesses des machines sont indépendantes les unes des autres, on parle alors de modèle de machines parallèles non reliées ou indépendantes.

3.3 conclusion :

le but de ce chapitre est le connaitre quelques types des problèmes d'ordonnancement les plus important dans ce domaine .

Conclusion générale:

L'ordonnancement repose essentiellement sur les résolutions de problèmes combinatoires, ces problèmes consistent à rechercher, dans un ensemble de solution possibles, une solution particulièrement intéressante au regard d'un ou de plusieurs critères, ces problèmes sont pour plupart NP-complets.

ce document s'intéresse à l'étude des problèmes d'ordonnancement de grande complexité.

Tout d'abord, nous nous intéressées à l'étude de l'ordonnancement en globalité ,avec ses différentes composantes (tâches, ressources, contraintes,critères), en rapport les différents méthodes d'optimisation pour trouver de bonnes solutions aux problèmes correspondants en utilisant pour leur résolutions, des méthodes exactes et des méthodes approchées (la recherche de tabou TS, algorithme de descente DS, les algorithmes génétiques GA, algorithme de colonies de fourmis ACO)

Dans la dernière partie on a également proposé des types des problèmes d'ordonnancement comme le modèle à une opération.

REFERENCES BIBILIOGRAPHIQUES:

1) J. Carlier et P. Chrétienne, « Problèmes d'ordonnancement, Modélisation, Complexité, Algorithmes ». Edition Masson, Paris, 1988.
2) C. Esswein . « Un apport de flexibilité séquentielle pour l'ordonnancement robuste ». Thèse de doctorat, Université François Rabelais, Tours, France, Décembre 2003.
3) P. Esquirol et P. Lopez, L'ordonnancement Economique, 1999.
4) K. Ghedira, optimisation combinatoire par méta heuristiques .origines, concepts et éléments de base. algorithmes canoniques et en duis .
5) D.E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning .Addison Wesley, 1989.
6) D. Goldberg. Genetic algorithms with sharing for multimodal function optimization .In Proceedings of the Second International Conference on Genetic Algorithms, pages 41--49, 1987.
7) D. E. Goldberg & K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In Foundations of Genetic Algorithms, p p. 69--93. Morgan Kaufmann, 1991
8) M. Nasser, Algorithmes de construction de graphes dans les problèmes d'ordonnancement de projet.
9) M. SAKAROVICH, Optimisation combinatoire, HERMANN, Paris, France, 1984.

10) K. Zakaria ,mémoire de fin d'etude présenté pour l'obtention du diplôme de MASTER, sujet : insertion d'une opération dans un problème d'ordonnancement a machines parallèles , université MOHAMED BOUDIAF_M'SILA

Liste des algorithmes:

Algorithme 1:simple descente

Algrithme2: Deepest descente

Algorithme 3:multi-Start descente

Algrithme4:TS

Algrithme5:GA

ملخص :

من خلال هذا البحث سنتعرف إلى أهم مفاهيم الجدولة مع ذكر طرق حل مشاكلها مثل الطرق التقريبية التي نلجأ إليها عندما نعجز عن حلها بالطرق المضبوطة مع تقديم بعض أنواع الجدولة مثل نوع ترتيبات أحادية المسار

الكلمات المفتاحية :

الجدولة، فوقيات الاستدلال،آلة موازية، آلة أحادية .

Résumé :

Grace à ce mémoire, on va savoir les notions plus importantes de l'ordonnancement, avec les méthodes de résolutions de leur problèmes telle que les méthodes approchées aux quelles nous avons recours lorsque nous ne pouvons pas éliminer les résoudre par les méthodes exactes, ainsi qu'en présentant certains types des problèmes d'ordonnancement comme le modèle à une opération.

Les mots clés :

Ordonnancement, machine unique, machine parallèle,

métaheuristique .

Abstract :

Through this research, we will get acquainted with the most important scheduling concepts with mentioning the methods of solving their problems, such as the approximate methods that we resort to when are unable to solve them by the controlled methods, while introducing some types of scheduling such as the type of single track arrangements.

Key word:

Scheduling, met heuristic, parallel machines, unique machines.