

Cet article a été téléchargé par : [141.209.144.159] Le : 13 janvier 2015, à : 09:04 Éditeur : Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS est situé dans le Maryland, aux États-Unis



Science du management

Détails de la publication, y compris les instructions pour les auteurs et les informations d'abonnement :
<http://pubsonline.informs.org>

Une heuristique de recherche taboue pour le dimensionnement capacitif des lots Problème avec le report de configuration

Mohan Gopalakrishnan, Ke Ding, Jean-Marie Bourjolly, Srimathy Mohan,

Pour citer cet article :

Mohan Gopalakrishnan, Ke Ding, Jean-Marie Bourjolly, Srimathy Mohan, (2001) A Tabu-Search Heuristic for the Capacitated Lot-Sizing Problem with Set-up Carryover. Sciences de gestion 47(6):851-863. <http://dx.doi.org/10.1287/mnsc.47.6.851.9813>

Termes et conditions d'utilisation complets : <http://pubsonline.informs.org/page/terms-and-conditions>

Cet article ne peut être utilisé qu'à des fins de recherche, d'enseignement et/ou d'étude privée. Un usage commercial ou le téléchargement systématique (par des robots ou d'autres processus automatiques) est interdit sans l'accord explicite de l'Éditeur, sauf mention contraire. Pour plus d'informations, contactez permissions@informs.org.

L'éditeur ne garantit pas l'exactitude, l'exhaustivité, la qualité marchande, l'adéquation à un usage particulier ou l'absence de contrefaçon de l'article. Les descriptions ou références à des produits ou publications, ou l'inclusion d'une publicité dans cet article, ne constituent ni n'impliquent une garantie, une approbation ou un soutien des réclamations faites à propos de ce produit, publication ou service.

© 2001 INFORME

Veuillez faire défiler vers le bas pour l'article - il se trouve sur les pages suivantes



INFORMS est la plus grande société professionnelle au monde pour les professionnels dans les domaines de la recherche opérationnelle, de la science de la gestion et de l'analyse.

Pour plus d'informations sur INFORMS, ses publications, ses membres ou ses réunions, visitez <http://www.informs.org>

Une heuristique de recherche taboue pour Problème de capacité de dimensionnement des lots avec Report de configuration

Mohan Gopalakrishnan • Ke Ding • Jean-Marie Bourjolly • Srimathy Mohan School of Management,
Arizona State University West, MC 2451, PO Box 37100, Phoenix, Arizona 85069-7100 Prestige Telcomm.

Ltée, 575 Boul. Morgon, Baie d'Urfé, PQ, Canada H9X 3T6 Département
des sciences de la décision et SIG, Université Concordia, 1455 boul. de Maisonneuve.

West Montreal, PQ, Canada H3G 1M8 School

of Management, Arizona State University West, MC 2451, PO Box 37100, Phoenix, Arizona 85069-7100 mohan@asu.edu • kding@prestige-
tel.com • brjolly@vax2.concordia.ca • srimathy.mohan@asuwest-online.asu.edu

Cet article présente une heuristique de recherche taboue pour le problème capacité de dimensionnement des lots (CLSP) avec report de configuration. Ces problèmes de planification de la production permettent de produire plusieurs articles dans une période de temps et de reporter les configurations d'articles d'une période à l'autre. Deux décisions interdépendantes, le séquençage et la taille des lots, sont présentes dans ce problème. Notre heuristique de recherche taboue se compose de cinq types de mouvement de base - trois pour les décisions de séquençage et deux pour les décisions de dimensionnement des lots. Nous permettons que des solutions irréalisables soient générées moyennant une pénalité au cours de la recherche. Nous utilisons plusieurs stratégies de recherche, telles que la liste taboue dynamique, la mémoire adaptative et les pénalités auto-ajustables, pour renforcer notre heuristique. Nous proposons également une procédure de borne inférieure pour estimer la qualité de notre solution heuristique. Nous avons également modifié notre heuristique pour produire de bonnes solutions pour le CLSP sans report de configuration. L'étude informatique, menée sur un ensemble de 540 problèmes de test, indique qu'en moyenne nos solutions heuristiques sont à moins de 12% d'une limite d'optimalité. De plus, pour l'ensemble des problèmes de test, nos résultats indiquent une réduction de 8 % du coût total grâce au report de configuration.

(Dimensionnement du lot ; recherche taboue ; report de configuration)

1. Introduction Dans cet

article, nous présentons une heuristique de recherche taboue pour un problème de dimensionnement de lot capacité (CLSP) à un seul niveau, multi-items, avec report de configuration. Le problème de lot-sizing que nous considérons implique la production de P articles sur une seule machine pour un horizon de planification de T périodes de temps. À chaque période, les multiples articles se disputent une capacité de production limitée. La production de l'élément i au cours de la période t , X_{it} , entraîne un coût de production unitaire c_{it} et consomme b_i unités de capacité. Inventaire de fin de période I_t est reporté sur la période suivante à un coût unitaire d'accès. Les changements de production d'un article à un autre entraînent un temps de préparation, s_i et un coût de préparation, sc_i , qui dépendent du produit.

période à la suivante peut avoir lieu dans les circonstances suivantes :

- le même article est produit le dernier à la période t et le premier à la période $t+1$;
 - un temps d'inactivité suffisamment grand à la fin de la période t est utilisé pour régler la machine pour produire l'article programmé en premier dans la période $t+1$;
 - une période d'inactivité t est utilisée pour configurer la machine afin de produire l'article programmé en premier dans la période $t+1$, ou pour maintenir l'état de la machine afin qu'elle soit prête à produire l'article programmé en premier dans la période $t+1$.
- L'objectif du CLSP avec report d'installation est d'obtenir un calendrier de production réalisable qui minimise la somme des coûts de production, d'inventaire et d'installation.

tout en permettant de reporter les configurations d'une période à l'autre.

Dans le CLSP avec des temps de réglage, un réglage doit être effectué chaque fois qu'un article est produit (Trigeiro et al. 1989). Par conséquent, il est simple de suivre le nombre d'installations au cours d'une période, qui est égal au nombre de produits fabriqués au cours de cette période.

Cependant, dans notre problème, nous devons garder une trace des informations de report de configuration d'une période à l'autre en séquençant partiellement les articles produits à chaque période, en spécifiant les articles produits en premier et en dernier et en gardant une trace de l'état de la machine (c'est-à-dire savoir quel article la machine est prête à produire) à la fin d'une période. Ainsi, le CLSP avec report de configuration est composé de deux décisions interdépendantes, à savoir le dimensionnement des lots et le séquençage partiel.

L'un des principaux problèmes posés par les recherches du CLSP est la modélisation des temps de mise en place. Dans la littérature, on trouve deux méthodes distinctes pour intégrer les temps de mise en place. La première méthode utilise les coûts d'établissement dans la fonction objectif comme substitut des temps d'établissement. La deuxième méthode inclut à la fois les coûts d'établissement dans la fonction objectif et les temps d'établissement dans les contraintes de capacité. De plus, l'intégration du report de configuration est essentielle pour de nombreux environnements de fabrication dans lesquels les changements de production prennent beaucoup de temps (Trigeiro et al. 1989). En outre, lorsque la capacité de production est restreinte, le report de configuration peut faire la différence entre l'obtention ou non d'un calendrier réalisable.

Le CLSP est connu pour sa capacité de calcul. Florian et al. (1980) ont montré que le cas général du CLSP mono-item est NP-difficile.

Trigeiro et al. (1989) déclarent que lorsque les temps de configuration sont introduits dans le CLSP multi-items, même le problème de faisabilité devient NP-complet. Lorsque le report de configuration est également inclus, le problème qui en résulte devient encore plus difficile.

Pour le CLSP avec temps de mise en place, plusieurs chercheurs ont développé des procédures de résolution heuristique. Nous présentons ci-dessous quelques-unes des heuristiques les plus pertinentes pour notre recherche. Aras et Swanson (1982) ont développé un algorithme heuristique pour déterminer à la fois la taille des lots et la séquence des tâches pour le CLSP avec des temps de configuration. Leur formulation n'inclut pas les coûts d'installation et le modèle minimise les coûts d'inventaire sous réserve des contraintes de capacité et de demande. Leur

Le cadre prend également en compte les coûts d'inventaire encourus pendant la période de production et, par conséquent, ils doivent déterminer la séquence des produits à chaque période.

Ces informations de séquençage sont utilisées pour transporter les configurations entre les périodes. Cependant, le principal moteur de l'algorithme est le coût total de l'inventaire et non le nombre de configurations.

Trigeiro et al. (1989) ont développé une heuristique basée sur la relaxation lagrangienne pour le CLSP avec des temps d'établissement. Ils assouplissent les contraintes de capacité et le problème résultant se décompose en un ensemble de problèmes indépendants, sans capacité et à un seul élément. Leur algorithme utilise la programmation dynamique pour résoudre les problèmes à un seul élément, une heuristique de lissage de la production pour corriger les violations de capacité et une procédure d'optimisation des sous-gradients pour renforcer la limite inférieure.

L'écart de solution entre l'heuristique et la borne inférieure pour un grand nombre de problèmes de test est de 2,6 % en moyenne.

Lozano et al. (1991) ont résolu le problème dual lagrangien du CLSP avec des temps d'établissement en utilisant un algorithme ~~double~~ qui fournit des solutions monotones convergentes à l'optimum lagrangien. Ils assurent la faisabilité du plan de production grâce à une heuristique qui effectue à la fois le fractionnement et le déplacement des lots. Diaby et al. (1992) ont proposé une autre procédure heuristique basée sur la relaxation lagrangienne pour résoudre CLSP avec des temps d'établissement, un temps régulier limité et un temps supplémentaire limité. Un travail similaire d'Anderson et Cheah (1993) résout le CLSP avec une taille de lot minimale et des temps de configuration.

Salomon et al. (1993) rapportent une procédure basée sur la génération de colonnes pour résoudre le CLSP à un niveau avec des temps d'établissement. Ils utilisent le recuit simulé ou la recherche tabou pour générer les nouvelles colonnes à ajouter au problème maître. Ils ont testé leur procédure sur des problèmes de test dans Trigeiro et al. (1989), et ont montré que la qualité de leur solution est similaire à celle de Trigeiro et al., mais les temps de calcul sont beaucoup plus importants.

Il existe quelques formulations pour modéliser les reports de configuration dans un petit intervalle de temps. Ici, la durée de chaque période de temps est très petite et la période entière est utilisée pour un réglage ou pour produire au plus un article, ou la machine est inactive. Dans ce contexte, le report de la configuration de la modélisation est relativement simple.

Cattrysse et al. (1993) ont développé une double ascension

et une heuristique basée sur la génération de colonnes pour planifier les produits dans un petit intervalle de temps. L'application d'une telle approche de modélisation au grand intervalle de temps augmenterait considérablement le nombre de variables entières. Dans de tels scénarios, les heuristiques basées sur la programmation mathématique développées pour le petit intervalle de temps seraient très inefficaces.

Par conséquent, il est nécessaire de modéliser les reports de configuration dans un grand intervalle de temps et de développer des procédures de solution pour ceux-ci.

Gopalakrishnan et al. (1995) ont développé un cadre de programmation linéaire à nombre entier mixte (MILP) pour le CLSP avec un temps de configuration constant et un report de configuration dans un grand intervalle de temps. Ce modèle a été motivé par un problème d'ordonnancement dans une grande papeterie, où les articles (serviettes gaufrées personnalisées) étaient produits sur plusieurs machines.

Les changements entre les éléments impliquaient un temps de réglage constant important. Le problème de l'usine de papier a été résolu à l'aide d'une procédure de branchement et de liaison. Les détails de la demande sont contenus dans Gopalakrishnan et al. (1995). Gopalakrishnan (2000) a étendu le modèle MILP au scénario de temps de mise en place dépendant du produit (modèle CLSPSC). Ce cadre est applicable dans une plus grande variété de situations de fabrication par rapport au cadre à temps de réglage constant, améliorant ainsi la pertinence pour la gestion.

Dans cet article, nous présentons TABU-CLSPSC, une heuristique basée sur la recherche taboue pour le CLSP avec report de configuration. La recherche taboue est une métaheuristique qui utilise des structures de mémoire et des stratégies d'exploration basées sur des informations stockées en mémoire pour rechercher au-delà des optima locaux. La procédure se déplace à plusieurs reprises d'une solution à la meilleure parmi ses solutions voisines. Pour éviter le cycle, la procédure stocke les solutions récemment visitées dans une "liste taboue" continuellement mise à jour pour un nombre donné d'itérations et ne les visite pas tant qu'elles sont dans la liste. Plusieurs stratégies telles que la diversification et l'intensification sont utilisées pour faire de la recherche taboue une procédure efficace et robuste (Hertz et al. 1997, Glover et Laguna 1993, Glover et al. 1993, 1998). Des chercheurs ont développé avec succès des heuristiques de recherche taboue pour des problèmes combinatoirement difficiles dans plusieurs domaines tels que le routage de véhicules (Gendreau et al. 1994) et l'ordonnancement d'ateliers de flux (Dell'Amico et Trubian 1993).

TABU-CLSPSC se déplace d'une solution donnée à une solution voisine en utilisant l'un des cinq mouvements de base qui définissent le voisinage de la solution courante. Les déménagements impliquent l'échange d'articles au cours d'une période, le déplacement de configurations d'une période à une période antérieure ou ultérieure, et le déplacement de lots de production d'une période à une période antérieure ou ultérieure.

Notre algorithme autorise des solutions qui violent les contraintes de capacité mais pénalise de telles solutions. L'algorithme incorpore également un schéma de mémoire adaptative proposé par Rochat et Taillard (1995) qui a des caractéristiques probabilistes d'intensification et de diversification.

Nous avons également développé TABU-CLSP, une heuristique basée sur la recherche taboue pour le CLSP avec des temps de configuration. Nous avons modifié TABU-CLSPSC et avons développé un mouvement spécifiquement pour TABU-CLSP. Nous avons testé nos deux heuristiques sur un large ensemble de problèmes de test utilisés par Trigeiro et al. (1989). Nos résultats indiquent que nous avons une bonne heuristique pour le CLSP avec des temps de configuration et un report de configuration.

Le reste du papier est organisé comme suit. La section 2 décrit en détail les mécanismes des cinq mouvements de base. Nous décrivons les différentes composantes des heuristiques TABU-CLSPSC et TABU-CLSP au §3. La section 4 fournit les détails et les résultats de notre étude computationnelle, et la section 5 fournit un résumé et une conclusion.

2. Description des mouvements de base Cette section contient la description des cinq mouvements qui aident le processus de recherche à passer d'une solution actuelle à une solution voisine. Nous introduisons la notation suivante avant de décrire la représentation de la solution et les déplacements.

X_{it} – quantité de production de l'article i pendant la période t ;
 t – type d'article (1 2 P) produit en premier dans la période t ($t = 0$, si la période t est inactive) ; – type d'article produit en dernier pendant la période t ($T = 0$, si la période t est inactive) ; – état de la machine (type d'article que la machine est prête à produire) à la fin de la période t ;
 où $j = 1$ P, et $t = 1$ T

En utilisant la représentation de la solution définie ci-dessus, nous pouvons déterminer le nombre d'articles produits dans

période t en comptant le nombre de variables X_{it} non nulles. Le nombre d'articles produits au cours de la période t ainsi que les valeurs de déterminent l'ensemble $t-1$, t , $t+1$, et $j+1$ d'articles configurés au cours de la période t .

Par exemple, supposons que les articles 1 et 2 soient produits à la période t . Si $t = 3$, $t = 1$, $t = 3$, alors les éléments 2 et 3 sont configurés à la période t . Nous déplaçons le stock initial $lit = X_{it} + lit - 1 - Dit$, où Dit est la demande de l'article i à la période t , et $lit - 1$ est le stock de l'article i à la fin de période $t-1$.

Étant donné une solution X , nous pouvons passer à une solution voisine X en utilisant l'un des cinq mouvements suivants. Les trois premiers mouvements opèrent sur les variables de séquençage partiel et les deux derniers mouvements opèrent sur les variables de lotissement, X_{it} .

2.1. Swap Items dans une période (SWAP)

Le premier type de mouvement, SWAP, échange les objets produits en premier et en dernier dans une période donnée, ou échange l'un d'entre eux avec un autre objet produit dans cette période. Nous pouvons effectuer un mouvement SWAP pour toute période au cours de laquelle au moins deux articles sont produits. Un mouvement SWAP pourrait réduire le nombre de mises en place dans deux périodes adjacentes grâce à un report de mise en place. Après SWAP, les valeurs de $t-1$ et t sont à jour.

Supposons que tous les éléments P sont produits dans chacune des périodes T . Ensuite, dans chaque période, on peut effectuer $P - 2$ mouvements d'échange entre un élément produit entre les deux et l'élément produit en premier, $P - 2$ échanges similaires impliquant l'élément produit en dernier et un mouvement d'échange entre les éléments produits en premier et en dernier. Ainsi, la taille du sous-voisinage correspondant au mouvement d'échange est au plus $P - 2 + P - 2 + 1T = 2p - 3T = OPT$.

2.2. Déplacer la configuration au début d'une période vers une période antérieure (SETUP1)

Ce déplacement décale le premier réglage de la période t à la fin de la période $t-1$, lorsque l'état de la machine à la fin de la période $t-1$ est différent de l'article produit en premier dans la période t . Ce mouvement crée la mise en place de « fin de période » et peut affecter les capacités de deux périodes adjacentes. Si la période t n'est pas inactive, SETUP1 affecte simplement la valeur de S . S'il est inactif, il affecte la valeur de

t pour $t-1$. La motivation derrière ce mouvement est d'aider à effectuer une configuration sur une série d'une ou plusieurs périodes d'inactivité adjacentes. La taille du sous-quartier associé à ce déplacement est au plus $T - 1 = OT$, parce que nous pouvons effectuer au plus un mouvement dans chacune des $T - 1$ premières périodes.

2.3. Déplacer la configuration de fin de période vers ce qui suit Période (SETUP2)

Le troisième mouvement, SETUP2, est l'inverse du deuxième mouvement. Ce déplacement décale le réglage de fin de période de la période t au début de la période $t + 1$, lorsque le dernier article produit de la période t est différent de l'état de la machine à la fin de cette période. SETUP2 supprime un établissement de fin de période dans la période t en le déplaçant à la période suivante, réduisant ou évitant ainsi une éventuelle violation de capacité dans la période t . Si la période t n'est pas inactive, SETUP2 affecte la valeur de t ; si la période t est inactive, il affecte la valeur de $t-1$ pour

t . La taille du sous-voisinage associé à SETUP2 est également OT .

2.4. Déplacer le lot de production vers une période antérieure (LOT1)

Le quatrième mouvement, LOT1, déplace une certaine quantité de production d'un article (par exemple, i) de la période t vers une période antérieure s , où $1 \leq s \leq t - 1$. En fonction de la capacité disponible dans la période s , nous pouvons déplacer soit le lot entier de l'article i ou juste une fraction de celui-ci vers la période s . LOT1 pourrait réduire les violations de capacité au cours de la période t grâce à une réduction de la configuration ou à un changement de lot.

Avant qu'un lot ne soit déplacé, nous vérifions la capacité disponible sur période s . Si l'article i est déjà produit au cours de la période s , aucune nouvelle configuration n'est nécessaire et toute la capacité de réserve disponible au cours de la période s peut être utilisée pour produire davantage d'article i . S'il n'existe pas de capacité excédentaire suffisante au cours de la période s pour absorber une quantité suffisante de l'article i de la période t , nous pouvons autoriser le transfert de la totalité du lot vers la période s . Nous considérons le compromis entre la pénalité pour violation de la capacité au cours de la période s ainsi que le coût de stockage de l'article i résultant du changement de lot, et la réduction du coût d'installation et de la violation de la capacité au cours de la période t , tout en exécutant le LOT1.

Si l'article i n'est pas produit à l'origine dans la période s , en plus de considérer le compromis de coût décrit ci-dessus, nous devons considérer les deux éléments suivants

mis en situation :

- Deux articles ou plus sont déjà produits dans la période s , et le nouvel article i est produit entre-temps.

Par conséquent, une nouvelle configuration est nécessaire et la quantité de capacité de production supplémentaire au cours de la période s est égale à la quantité de capacité excédentaire moins le temps nécessaire pour effectuer une configuration pour l'élément i . Nous maintenons les valeurs des variables de séquençage partiel dans les périodes t et s en ne produisant pas l'item i en premier ou en dernier

dans la période s . • Moins de deux articles sont produits dans la période s . Ici, nous devons examiner les séquences partielles dans les périodes s et $s - 1$ pour déterminer le nombre d'établissements pour la période s . Notez que la valeur de certaines des variables de séquençage partiel, s , peut également changer. Supposons, par exemple, qu'un seul article (disons, j) est produit dans la période s à l'origine, et que la machine est déjà configurée pour produire l'article i à la fin de la période $s - 1$ (c'est-à-dire, $= i$). Après le transfert, si nous produisons l'élément i en premier dans la période s (pour profiter du report de configuration), s passe de j à i et la valeur de reste la même (j).

s

Supposons que tous les éléments P sont produits dans chacune des périodes T . Comme nous pouvons déplacer le lot pour n'importe quel élément de la période t vers au plus $t - 1$ périodes précédentes ($t = 2$ à T), la taille du sous-voisinage correspondant au déplacement du LOT1 est au plus $1 + 2 + \dots + T - 1P = OPT2$.

2.5. Déplacer le lot de production vers une période ultérieure (LOT2)

Le cinquième coup, LOT2, est l'inverse de LOT1. Ici, une certaine quantité de production pour un article i dans la période t est déplacée vers une période ultérieure s ($t + 1 \leq s \leq T$), à condition qu'il y ait un stock final positif pour l'article i dans la période t et un stock final nul pour le même élément dans la période s . En outre, la quantité de production déplacée (lot entier ou fraction) dépend de la capacité disponible au cours de la période s , déterminée comme dans le LOT1. Notez que pour LOT1 et LOT2, si nous décalons un lot entier pour l'article i hors de la période t et que l'article i devait être produit en premier ou en dernier dans la période t , alors nous devons changer les valeurs de certains des séquençages partiels. variables pour les périodes t et $t - 1$. Comme nous pouvons déplacer le lot pour un élément de la période t vers au plus $T - t$ périodes ultérieures ($t = 1$ à $T - 1$), la taille du sous-voisinage est au plus $(1+2+\dots+T-1)P = OPT2$.

3. Description de l'heuristique

L'algorithme global TABU-CLSPSC contient les trois composants suivants.

- Un générateur de solution initiale qui fournit douze différentes solutions initiales.
- Une recherche taboue de base (BTS) qui travaille sur chacune des solutions initiales en utilisant les cinq mouvements décrits précédemment pour un nombre donné d'itérations et développe des calendriers de production améliorés. Nous décomposons chacune de ces solutions améliorées en programmes à un seul élément et les chargeons dans un ensemble que nous appelons la mémoire adaptative.
- Une phase de diversification et d'intensification qui sélectionne et combine probabilistiquement des programmes mono-item trouvés dans la mémoire adaptative pour former de nouvelles solutions.

Lors des phases de diversification et d'intensification, nous formons de nouvelles solutions, améliorons ces solutions à l'aide du BTS, et chargeons les plannings unitaires associés sur la mémoire adaptative jusqu'à ce qu'un critère d'arrêt soit atteint. Ensuite, nous décrivons en détail chacun des composants.

3.1. Générateur de solution initiale

Pour chaque solution initiale, nous devons déterminer les niveaux de production des éléments individuels et les valeurs des variables de séquençage partiel à chaque période. Nous utilisons les six méthodes suivantes pour déterminer les niveaux de production.

- Utilisez l'approche lot

pour lot, où le niveau de production d'un article au cours d'une période donnée correspond à sa demande au cours de cette période.

- Utilisez la méthode

Wagner-Whitin (Wagner et Whitin 1958) de générer un programme à un seul élément.

- Produire la demande totale pour un article dans la première période avec une demande non nulle pour cet article.
- Divisez l'horizon de planification en deux moitiés et produisez toute la demande pour un article dans une moitié donnée de l'horizon de planification, dans la première période de cet horizon.
- Divisez l'horizon de

planification en quatre parties égales et produisez toute la demande d'un article dans un trimestre donné de l'horizon de planification, dans la première période de cet horizon.

- Produire la demande combinée de tous les deux périodes dans la première des deux périodes.

Ensuite, nous attribuons des valeurs initiales aux variables de séquençage partiel en utilisant l'une des deux méthodes suivantes. Les deux méthodes attribuent des valeurs aux variables une période à la fois. Pour chaque période, considérez tous les articles dont la quantité de production n'est pas nulle au cours de cette période et trie-les par ordre croissant des quantités de production. La première méthode produit le premier élément de la liste comme premier produit de la période et le deuxième élément de la liste comme dernier produit. La deuxième méthode produit le premier élément de la liste comme premier produit de la période et le dernier élément de la liste comme dernier produit. Le réglage de fin de période est toujours égal au dernier produit fabriqué au cours de cette période (c'est-à-dire t). Pour les périodes d'inactivité, t Ainsi, les six programmes de production, associés à deux ensembles différents de variables de séquençage partiel, donnent douze solutions initiales.

3.2. Procédure de recherche tabou de base

Dans cette section, nous décrivons les éléments importants de la recherche tabou de base, puis fournissons une description étape par étape de l'heuristique de recherche.

3.2.1. Fonction objectif. Pour une solution réalisable X donnée, la fonction objectif $F1X$ se compose des coûts de stock pour le maintien des stocks sur plusieurs périodes, des coûts de mise en place encourus au cours d'une période et du coût de production pour produire l'article i pendant la période t . Cela peut être exprimé comme suit :

$$F1X = \sum_t hitlit + \sum_t SciNite + \sum_t citXit$$

où Nit est la variable entière comptant le nombre de configurations pour l'élément i pendant la période t . Parce que nous autorisons les violations des contraintes de capacité pendant le processus de recherche, nous définissons une autre fonction objectif $F2X$ pour les solutions violant les contraintes de capacité comme suit :

$$F2X = F1X + p \sum_t biXit + \sum_t stiNit - Ct$$

où $x+ = \max(0, x)$, p est le taux de pénalité et Ct est la capacité disponible à la période t . A noter que lorsqu'une solution X est réalisable, $F1X = F2X$. Soient respectivement F^* et F_1 les plus petites valeurs de $F1X$ et $F2X$ obtenues lors de la recherche, X^* la meilleure solution connue (réalisable ou non) et X_1 la meilleure solution réalisable connue.

3.2.2. Taux de pénalité. Le taux de pénalité p est lié aux violations de capacité. Pour toute solution irréalisable, nous pénalisons le montant de la violation de capacité dans n'importe quelle période par ce taux. Ce taux de pénalité est initialement fixé à une valeur arbitraire de 50. Le taux varie au cours de la recherche, en gardant à l'esprit qu'un taux élevé peut être en mesure d'éliminer rapidement l'infaisabilité, mais il peut également empêcher la recherche d'autres bonnes régions. Inspiré des travaux de Gendreau et al. (1994), nous utilisons le concept de pénalités auto-ajustables pour produire un mélange de solutions réalisables et irréalisables au cours du processus de recherche. Dans notre recherche tabou de base, nous gardons une trace du nombre de solutions irréalisables, n , obtenues au cours des h itérations précédentes. Si n est inférieur à $h/2$, on réduit la valeur de p à $0.5 + n/hp$, car plus de la moitié des solutions étaient réalisables. Si n est supérieur à $h/2$, on augmente la valeur de p à $2n/hp$ et on ne change pas la valeur de p si $n = h/2$. Sur la base de l'expérimentation initiale, nous avons défini la valeur de h sur 10 dans notre recherche tabou de base. Nous avons aussi défini des valeurs minimales et maximales de 0,5 et 10 000 pour p .

3.2.3. Structure du quartier. Nous générons le voisinage d'une solution donnée X en invoquant chacun des cinq types de mouvements décrits dans la section précédente. Notez que lorsque de nombreux éléments, mais pas les mêmes, sont produits au cours de deux périodes adjacentes, le mouvement SWAP peut produire de nombreuses solutions voisines qui ne sont ni meilleures ni pires que la solution donnée X . En effet, de nombreux éléments peuvent être produits. premier ou dernier, et la valeur de la fonction objectif reste la même pour chacun de ces swaps. Si aucun des autres types de mouvements ne donne une solution d'amélioration, le processus de recherche choisira la solution suivante dans la liste des solutions produites par le mouvement SWAP.

Cela est dû à l'existence d'un grand nombre de mouvements SWAP non tabous, même avec l'utilisation d'une longue liste tabou. Par conséquent, le processus de recherche pourrait stagner. Notre première expérimentation l'a confirmé. Pour éviter ce problème, nous invoquons le mouvement d'échange une fois en trois itérations. Ainsi, nous générons le voisinage d'une solution donnée en utilisant les cinq mouvements un tiers du temps et en utilisant les quatre derniers mouvements les deux tiers du temps.

3.2.4. Attributs Tabu-Move et liste tabou. Nous représentons un coup tabou par un ensemble d'attributs définis

pour chaque type de déménagement. Pour le déménagement SWAP, nous enregistrons le nombre ordinal de la période t et l'article produit en premier ou en dernier (t ou t) dans cette période avant le déménagement. Tout mouvement SWAP impliquant les t ou t valeurs initiales de dans la période t est tabou pour L itérations après celle en cours. Pour SETUP1, les attributs de déplacement associés sont le nombre ordinal de la période précédente, $t-1$, et le type d'élément impliqué dans le déplacement. De même pour SETUP2, les attributs sont le nombre ordinal de la période suivante, $t+1$, et l'état de la machine à la fin de la période, t . De plus, comme ces deux mouvements sont l'inverse l'un de l'autre, à chaque fois que l'un d'eux est effectué, le mouvement inverse est tabou pendant L itérations après celui en cours. De même, lorsqu'un lot est décalé vers une période antérieure ou ultérieure (LOT1 et LOT2), les attributs de déplacement correspondants sont l'élément décalé, i , et le nombre ordinal de la période cible vers laquelle le lot est décalé (c'est-à-dire s). Encore une fois, ces deux mouvements sont inverses l'un de l'autre, et si l'un est exécuté, l'inverse est tabou pour L itérations après le l'actuel.

TABU-CLSPSC utilise une liste tabou de longueur variable. Basé sur les travaux de Dell'Amico et Trubian (1993), l'algorithme ajuste la longueur de la liste L de la manière suivante : Diminuez la longueur de un si la solution obtenue après le déplacement est meilleure que celle en cours ou bien augmentez-la de un . De plus, la longueur de la liste est maintenue dans une plage [L_{min} L_{max}]. Après quelques expériences initiales, nous avons fixé les valeurs de L_{min} et L_{max} à $0,25(P+T)$ et $1,75(P+T)$, respectivement.

3.2.5. Règle d'arrêt. Si le nombre d'itérations pendant lesquelles la recherche progresse sans aucune amélioration de la fonction objectif est supérieur à une valeur k_{max} prédéterminée, on arrête la recherche tabou de base. Sur la base de l'expérimentation initiale, nous avons fixé la valeur de k_{max} à $15300+10P$.

3.2.6. Description de la recherche tabou de base. Nous fournissons maintenant une description étape par étape du processus de recherche tabou de base qui tente d'améliorer une solution X donnée.

Étape 0 : Initialisation

Soit $F \leftarrow F$ $\leftarrow 12$. Définissez le compteur d'itérations $k \leftarrow 1$.

Considérons la solution donnée X . Posons $F \neq F2X$ et $X = X$. Si X est faisable, posons $F \neq F1X$ et X La liste taboue est vide.

Plage de longueur de la liste tabou : Définir $L_{min} \leftarrow 0,25P + T$, $L_{max} \leftarrow 1,75P + T$

Taux de pénalité : Set $p \leftarrow 50$ $p_{min} \leftarrow 01$ $p_{max} \leftarrow 10000$ $h \leftarrow 10$

Règle d'arrêt : $k_{max} \leftarrow 15300+10P$ Étape 1 :

Évaluation du coup et sélection du meilleur coup Appelez SETUP1, SETUP2, LOT1, LOT2, et si $k \bmod 3 = 0$, appelez SWAP. Générer tous les voisins X de X comme décrit au §3.2.3. Ne pas considérer une solution voisine X générée par un mouvement tabou sauf si $F2X < F$ Set $X = \text{argmin } F2X$.

2 .

Étape 2 : Liste taboue et mise à jour de la solution Si $F2X < F$ 2 , videz la liste taboue existante et ajoutez les attributs du coup donnant X (les attributs du coup courant) à la liste taboue. Si $F2X \geq F$ procédez comme suit pour mettre à jour la liste tabou comme expliqué au §3.2.4. • Si la longueur de la liste actuelle

est inférieure à L_{min} , ajoutez le attributs de déplacement actuels vers la

liste. • Sinon,

• Si $F2X \leq F2X$: Supprimer deux éléments depuis le début et ajouter les attributs de mouvement en cours à la liste taboue si la longueur de la liste $> L_{min}$; supprimer un élément du début et ajouter les attributs de mouvement actuels à la liste tabou si la longueur de la liste $= L_{min}$; • Si $F2X > F2X$: et que la longueur de la

liste actuelle est inférieure à L_{max} , ajouter les attributs de mouvement actuels à la liste tabou si la longueur de la liste $< L_{max}$; supprimez un élément du début et ajoutez les attributs de déplacement actuels à la liste tabou si la longueur de la liste $= L_{max}$.

Définir $X \leftarrow X$. Si $F2X \leq F$ 2 , ensemble $X \leftarrow X2F \leftarrow F2X$.

Si $F1X \leq F$ 1 ensemble $X \leftarrow \text{SUP } 1 \leftarrow F1X$

Étape 3 : Mise à jour du taux de pénalité Si $k \bmod h = 0$, mettre à jour le taux de pénalité p comme expliqué au §3.2.2.

Étape 4 : vérification de la

terminaison Si F n'a pas diminué pour les dernières itérations k_{max} , arrêtez. Sinon, passez à l'étape 1.

3.3. Diversification et Intensification

La troisième composante de TABU-CLSPSC est un schéma probabiliste de diversification et d'intensification

motivé par les travaux de Roachat et Taillard (1995) pour le problème de tournées de véhicules. Le but de cette procédure est de fournir de nouvelles solutions de départ pour la recherche tabou de base grâce à une sélection probabiliste et à la combinaison de parties de différentes solutions précédemment obtenues. Nous décomposons les solutions visitées précédemment qui sont disponibles pour être combinées en programmes d'éléments uniques et stockons ces programmes dans un ensemble D appelé la mémoire adaptative.

Dans un premier temps, la mémoire adaptative est constituée de schémas mono-items obtenus en décomposant les douze solutions initiales au CLSP avec report de réglage, produites par le générateur de solutions initiales et améliorées par la recherche tabou de base. Avec chaque programme à un seul élément, nous stockons également les informations de report de configuration associées et la valeur de la fonction objectif de la solution à laquelle elle appartient.

Spécifiquement, les informations de report de réglage indiquent la période pendant laquelle un réglage est reporté pour cet article particulier. Nous trions les horaires à élément unique dans la mémoire adaptative par ordre croissant en fonction des valeurs de la fonction objectif. Tous les plannings unitaire appartenant à une même solution sont ensuite triés par ordre décroissant en fonction du nombre de report de réglage pour chacun d'eux. Ainsi, si l'on considère deux plannings mono-article de la même solution, le planning avec le plus de reports de réglages sera stocké avant l'autre planning mono-article. Nous stockons une copie de ce pool d'horaires à élément unique dans l'ensemble D et construisons une nouvelle solution historique à partir de D (c'est-à-dire, des horaires .

avec une valeur de fonction objectif plus faible et un nombre plus élevé de report de réglage). La probabilité de sélectionner le i ème pire programme mono-item dans l'ensemble D est de $\frac{2}{DD + 1}$ (Roachat et Taillard 1995). Une fois que nous avons sélectionné un programme d'article unique, nous essayons de l'ajouter à la nouvelle solution historique, X. Lors de l'ajout de nouveaux programmes d'article unique à la solution historique, nous vérifions la violation de capacité à chaque période. Afin de ne pas être trop restrictif et d'éliminer les bons horaires pour un seul article, nous autorisons un minimum de violation de capacité dans la solution en place. Si le programme à élément unique sélectionné ne dépasse pas la capacité de chaque période de plus de % (= 5 dans notre calcul), nous l'ajoutons à X. Sinon, nous

Jetez-le. Après avoir ajouté le programme mono-article à X, nous mettons à jour la capacité restante à chaque période. De plus, nous supprimons de D tous les horaires des différentes solutions correspondant à l'élément qui vient d'être ajouté à X.

Ensuite, nous sélectionnons un autre programme à élément unique à partir de D et répétons le processus jusqu'à ce que tous les éléments soient planifiés ou que l'ensemble D soit vide. Dans ce dernier cas, si la nouvelle solution historique X ne

contient pas d'horaires pour tous les éléments, nous construisons une solution complète comme suit. Pour chaque article qui n'est pas couvert par X, nous ajoutons à X un programme d'article unique lot pour lot correspondant, c'est-à-dire que le niveau de production de cet article dans chaque période est égal à sa demande dans cette période. Cette nouvelle solution en place sert ensuite de solution de départ pour le processus de recherche tabou de base. Le BTS essaie d'obtenir une solution améliorée, et les programmes à élément unique associés à la solution améliorée sont rajoutés dans la mémoire adaptative.

Ce processus est répété jusqu'à ce qu'un critère d'arrêt soit satisfait.

Nous limitons la taille de la mémoire adaptative pour qu'elle ne contienne que des ordonnancements à un seul élément $8 - P$ afin qu'elle ne croît pas continuellement au fur et à mesure que les phases de diversification et d'intensification progressent. A chaque fois que l'on trie les nomenclatures unitaire en D selon le schéma décrit précédemment, on compte également le nombre de nomenclatures unitaire en D. Si le nombre de nomenclatures unitaire est supérieur à $8 - P$, on supprime la dernière $D - 8 - P$ programmes de cette liste triée. Comme expliqué dans Roachat et Taillard (1995), une fois que nous effectuons plusieurs fois le processus de diversification et d'intensification, la recherche tend à se concentrer dans les régions prometteuses de l'espace des solutions. Cela est dû au fait que nous choisissons les horaires de manière probabiliste avec un biais vers les "meilleurs" horaires et que les pires

horaires sont supprimés de la mémoire adaptative pour maintenir sa taille. Notez également que parce que nous autorisons des horaires mono-élément (provenant de solutions différentes) à ajouter à la mémoire adaptative, les « meilleurs » horaires sont plus souvent utilisés pour former la nouvelle solution historique et le processus passe lentement d'une phase de diversification à une phase d'intensification.

3.3.1. Algorithme global. Nous fournissons ci-dessous une description détaillée de l'algorithme global qui

utilise la structure de la mémoire adaptative pour diversifier et intensifier la recherche.

Phase d'initialisation

$F \leftarrow \emptyset$, $X \leftarrow \emptyset$, et $X \leftarrow \emptyset$. Soit F

1. Générer 12 solutions initiales comme décrit au §3.1.

2. Pour chaque solution initiale X , effectuer la recherche tabou de base pour obtenir une solution améliorée X Si et $F = F2X$. De mettre à jour F , 3. Décomposer $F2X < F$ définir $X = X$ en X_1 et X .

programmes à un seul élément et, pour chaque élément, enregistrer la valeur de la fonction objectif associée et le nombre de reports de configuration.

4. Insérez chaque programme à élément unique dans la mémoire adaptative D , qui est triée dans un ordre croissant par valeur de fonction objectif. Tous les plannings mono-élément appartenant à la même solution sont triés par ordre décroissant en fonction du nombre de reports de paramétrage pour chacun d'eux.

5. Si le nombre de programmes dans D est supérieur à $8 - P$, supprimer les derniers horaires $D - 8 - P$ de D .

Phase de diversification et d'intensification Étape 0 :

Initialisation Définissez

le nombre d'exécutions de recherche tabou de base non améliorées, $c \leftarrow 0$.

Étape 1 : Nouvel

ensemble de solutions $D \leftarrow D$ et $X \leftarrow$ vides.

Tant que D n'est pas vide •

Choisissez un programme à un seul élément $d \in D$ avec une probabilité de $2^{-i/DD + 1}$ pour sélectionner le i ème pire programme à un seul élément, et retirez d de D .

• Si l'ajout de d à la nouvelle solution historique X ne dépasse pas la capacité de chaque période de plus de % de la limite de capacité de cette période, ajoutez d à X .

Sinon, jetez d . • Si d est

ajouté à X , mettez à jour la capacité à chaque période et supprimez de D tous les programmes monopostes qui correspondent à l'item d .

Si certains articles ne sont pas couverts par les programmes dans X , construisez une solution complète en ajoutant à X les programmes lot pour lot pour les articles non couverts. Pour le

nouvelle solution titulaire X , effectuer la recherche tabou de base pour obtenir une nouvelle solution améliorée X .

Étape 2 : Mettre à jour

Mettez à jour la mémoire adaptative à l'aide des étapes 3 à 5 de la phase d'initialisation. Si $F1X < F$, l'ensemble X .

Si $F2X < F$ Pas, ensemble $X = XF = F2X$, et passez à 2 0 ; sinon fixez $c \leftarrow c + 1$.

Étape 3 : Vérification de la

terminaison Si $c < 3$, passez à l'étape 1. Sinon, arrêtez. Si $F < X$ est la meilleure solution réalisable ; sinon, aucune solution réalisable n'a été trouvée.

3.4. Heuristique TABU-CLSP

Nous avons modifié TABU-CLSPSC et ajouté

un mouvement supplémentaire sur mesure pour développer TABU-CLSP, une heuristique pour le CLSP avec des temps de configuration. Une comparaison entre les performances de cette heuristique et l'heuristique TTM de Trigeiro et al. (1989) aiderait à comprendre l'efficacité des techniques de recherche tabou pour le CLSP.

Nous utilisons les mouvements LOT1 et LOT2 de TABU CLPSC qui traitent de la taille des lots. Cependant, nous n'autorisons pas les reports de configuration. Chaque fois qu'un article est produit, il subit une mise en place. Nous avons développé un mouvement supplémentaire, LOTSWAP, qui permet d'échanger des lots contre deux objets différents entre deux périodes. Le mouvement LOTSWAP déplace une certaine quantité de production d'un article (disons i) de la période t vers une période ultérieure $st + 1 \leq s \leq T$ et déplace simultanément une certaine quantité de production d'un article (disons $jj = i$) de la période s à la période t . Essentiellement, ce mouvement est une combinaison des mouvements LOT1 et LOT2. La différence est que la quantité d'un élément i qui peut être déplacé vers une période cible est déterminée en tenant compte de la capacité disponible dans la période cible après le déplacement d'un autre élément j . Nous déplaçons le lot entier ou une partie de celui-ci, selon la capacité disponible. Le mouvement LOTSWAP tente de réduire les violations de capacité de fusion et de transfert de lots.

Les attributs de déplacement sont les nombres ordinaux des deux périodes (t et s) impliquées dans l'échange de lot. De plus, si le déplacement est exécuté entre les deux périodes, alors tout autre déplacement qui tente d'échanger des éléments entre ces périodes est considéré comme tabou pour les itérations suivantes, où L est la longueur de la liste taboue. Supposons que tous les éléments P sont produits dans chacune des périodes de temps T . Nous pouvons déplacer le lot ou une partie de celui-ci pour tout article i dans la période de temps t vers au plus $T - t$ périodes ultérieures. Comme nous pouvons déplacer n'importe lequel des autres éléments $P - 1$ vers la période t , la taille du sous-voisinage correspondant à ce déplacement est au plus $1+2+\dots+T-1 = OP2T2$.

L'heuristique TABU-CLSP est similaire au TABU

Heuristique CLSPSC décrite au §3.3 avec la seule différence étant la définition du voisinage dans une solution. Dans la phase d'initialisation, nous n'utilisons que les coups LOT1 et LOT2 pour la recherche tabou de base. Dans

la phase de diversification et d'intensification, nous utilisons les mouvements LOT1, LOT2 et LOTSWAP dans la recherche tabou de base. Cependant, comme le sous-voisinage du déplacement LOTSWAP est relativement important, nous n'invoquons ce déplacement que dans 2 itérations sur 10. Toutes les valeurs des paramètres sont les mêmes que dans TABU-CLSPSC.

Nous avons également essayé le mouvement LOTSWAP dans TABU-CLSPSC. Après quelques tests initiaux, nous avons décidé de ne pas utiliser ce mouvement pour TABU-CLSPSC.

En raison de la taille du sous-voisinage, le temps de calcul supplémentaire était excessif et l'amélioration marginale de la qualité de la solution était minime. En fait, dans certains cas, la solution produite avec LOTSWAP était pire.

4. Étude informatique Le but de l'étude informatique était triple. Le premier objectif était de quantifier l'efficacité du report de configuration en utilisant l'heuristique TABU-CLSPSC ; la seconde était de tester l'efficacité de l'heuristique ; et la troisième consistait à comparer les performances de l'heuristique TABU-CLSP avec celles de Trigeiro et al. (1989). Nous avons utilisé les 751 problèmes tests obtenus de Trigeiro et al. pour calibrer et tester nos heuristiques.

Pour quantifier l'efficacité du report de configuration, nous avons résolu toutes les instances du problème, en utilisant d'abord l'heuristique TABU CLSPSC, puis la procédure TTM. Nous définissons la mesure d'efficacité comme la différence en pourcentage entre la valeur de la solution TTM et la valeur de la solution TABU CLSPSC. Cette mesure nous donne une indication des économies de coûts potentielles en incorporant les reports de configuration pour le même ensemble de problèmes.

La qualité de la solution heuristique et les temps d'exécution déterminent l'efficacité de l'heuristique. Parce que les solutions optimales ne sont pas faciles à trouver, nous avons développé une procédure de limitation inférieure pour déterminer l'écart de solution. Cette procédure tire parti du fait suivant : nous pouvons effectuer au plus un réglage pour chacune des périodes de temps T et, par conséquent, réduire le nombre de réglages dans n'importe quelle période d'un en utilisant au plus le cadre CLSPSC

(Gopalakrishnan 2000) par opposition à Trigeiro et al. (1989) (CLSP).

Nous pouvons intégrer indirectement la réduction du nombre d'installations dans le modèle CLSP en augmentant sa capacité à chaque période. Étant donné qu'au plus une configuration peut être effectuée, nous utilisons le temps de configuration maximal pour tous les éléments comme valeur pour augmenter la capacité dans le modèle CLSP. Notez également que, parce que nous réduisons de un le nombre d'installations dans chaque période de temps, la fonction objectif du modèle CLSP doit être réduite en conséquence. Là encore, parce que nous avons besoin d'une limite inférieure, nous utilisons le coût d'installation maximal pour tous les éléments.

Par conséquent, si nous ajoutons le temps d'établissement maximal à la capacité de chaque période dans le modèle CLSP et réduisons sa fonction objectif de (T le coût d'établissement maximal), la solution optimale à ce modèle CLSP modifié est une borne inférieure pour le CLSPSC. Cependant, le CLSP lui-même est un problème difficile, et il n'est pas pratique d'attendre des solutions optimales pour des instances de taille raisonnable. Cependant, la procédure lagrangienne de Trigeiro et al. (1989) fournit une borne inférieure pour le CLSP, et donc pour le CLSP modifié. Cette borne inférieure pour le CLSP modifié sert de borne inférieure pour la CLSPSC. Maintenant, la mesure de performance est l'écart d'optimalité, qui est la différence en pourcentage entre la solution heuristique et la borne inférieure.

Nous avons également résolu toutes les instances problématiques en utilisant l'heuristique TABU-CLSP. Nous comparons les performances de TABU-CLSP avec celles de l'heuristique TTM en utilisant les valeurs de solution et les temps de calcul.

Trigeiro et al. générer leurs instances de problème en trois phases. La phase 1 a 70 problèmes et a été utilisée pour affiner les paramètres de l'algorithme TTM, tandis que la phase 2 génère 141 problèmes en analysant les différentes caractéristiques du problème, et la phase 3 a 540 problèmes qui ont été utilisés pour tester l'algorithme.

Le tableau 1 donne les tailles de problème et le nombre d'instances de problème pour chaque catégorie.

Les heuristiques TABU-CLSPSC et TABU-CLSP ont été codées en C++ et exécutées sur un PC avec un microprocesseur Pentium III, 550 MHz. La procédure TTM (fournie par les auteurs) est codée en FORTRAN. Il a été recompilé et exécuté sur la même machine. Nous avons utilisé les 70 problèmes de la phase 1 pour les tests initiaux et pour fixer les valeurs des différents

Tableau 1 Nombre de problèmes dans la conception expérimentale

	Nombre d'objets	Nombre de périodes		
		15	30	20
Phase 2	6	116	5	—
	12	5	5	—
	24	5	5	—
Phase 3	dix	—	—	180
	20	—	—	180
	30	—	—	180

paramètres de notre algorithme. Ensuite, nous avons testé les deux heuristiques sur les problèmes de phase 2 et de phase 3 et présentons les résultats ci-dessous.

4.1. Résultats de calcul

Efficacité du Set-upCarryover. TABU-CLSPSC a trouvé des solutions moins coûteuses par rapport à TTM pour toutes les instances de problème sauf une dans la phase 3. Le tableau 2 résume les résultats de la mesure de l'efficacité du report d'installation. Comme attendu, les résultats montrent qu'en moyenne, le coût total des solutions TABU-CLSPSC est inférieur au coût total des

les solutions trouvées par la procédure TTM. La réduction moyenne des coûts est d'environ 24 % pour les problèmes de la phase 2 et de 8 % pour les problèmes de la phase 3. La différence dans les économies de coûts est due à la taille des problèmes en termes de nombre d'éléments. La plupart des problèmes (116 sur 141) de la phase 2 contiennent 6 éléments et 15 périodes, tandis que les problèmes de la phase 3 comportent 10, 20 et 30 éléments et 20 périodes. La réduction moyenne du coût total diminue à mesure que le nombre d'éléments dans une instance de problème augmente. La raison en est qu'au plus un montage peut être reporté à chaque période, et donc pour les problèmes

Tableau 2 Efficacité du report de configuration

	Efficacité Mesure (%)	Nombre d'objets	Nombre de périodes		
			15	30	20
Phase 2	2439	6	2734	2319	—
		12	1129	948	—
		24	559	404	—
		dix	—	1457	—
Phase 3	749	20	—	590	—
		30	—	338	—

avec plus d'articles, la réduction des coûts due au report de configuration représente une plus petite proportion du coût total.

Efficacité de l'heuristique TABU-CLSPSC. TABU CLSPSC a trouvé une solution réalisable pour tous les cas de problèmes. Nous présentons les résultats de calcul liés à la mesure de l'efficacité de notre heuristique dans le tableau 3. La première ligne présente les statistiques moyennes pour l'heuristique TTM, la deuxième ligne pour TABU-CLSP et la troisième ligne pour TABU-CLSPSC. L'écart d'optimalité moyen pour TABU-CLSPSC est de 25,6 % pour les problèmes de phase 2 et de 12,4 % pour les problèmes de phase 3. Encore une fois, l'augmentation de l'écart d'optimalité pour les problèmes de la phase 2 est due au grand nombre de problèmes à 6 items. Les écarts moyens pour TABU CLSPSC sont respectivement de 27,8 %, 13,9 % et 6,0 % pour les problèmes à 6, 12 et 24 items. La sensibilité de l'heuristique au nombre d'items explique la grande variation des écarts d'optimalité pour la phase 2 et la phase 3. En général, nous pensons que les écarts d'optimalité un peu grands sont probablement dus à la qualité des bornes inférieures. Notre borne est en fait une borne inférieure sur une borne inférieure, et donc les deux effets se cumulent.

Pour évaluer l'utilité de la mémoire adaptative, nous avons suivi l'écart d'optimalité à la fin de la procédure d'initialisation. Pour les problèmes de phase 2, TABU-CLSPSC a trouvé une solution réalisable pour les 141 problèmes à la fin de la procédure d'initialisation, et l'écart d'optimalité moyen était de 27,6 %. Celle-ci est ramenée à 25,6 % après la procédure probabiliste de diversification et d'intensification. Pour les problèmes de la phase 3, l'écart d'optimalité moyen initial était de 14,6 % et 4 instances de problèmes (sur 540) avaient des solutions irréalisables. L'écart final était de 12,4 % et toutes les instances de problème avaient des solutions réalisables. Ces résultats indiquent

Tableau 3 Efficacité de l'heuristique

	Phase 2		Phase 3	
	Moyenne temps CPU (en secondes)	Moyenne d'optimalité (%)	Moyenne temps CPU (en secondes)	Moyenne d'optimalité (%)
TTM	396	026	229	059
TABU-CLSP	32	248	401	9721
TABU-CLSPSC	2556	2076	1238	8166

que notre définition de quartier et les procédures de recherche tabou de base sont assez robustes et que la mémoire adaptative aide à affiner et à orienter le processus de recherche vers des régions plus prometteuses.

Le temps de calcul moyen est de 20,8 secondes pour la phase 2 et de 81,7 secondes pour la phase 3. Le temps de calcul tend à augmenter avec l'augmentation de la taille du problème, notamment avec le nombre de périodes. Ceci est principalement dû à la recherche exhaustive du quartier.

Comparaison de TABU-CLSP et de l'heuristique TTM.

Le tableau 3 présente les écarts d'optimalité moyens et les temps d'exécution pour les deux heuristiques. L'écart d'optimalité moyen pour la phase 2 est de 3,2 % pour TABU-CLSP et de 4,0 % pour TTM. Les résultats de cette phase indiquent que TABU-CLSP semble être plus performant que TTM en moyenne pour les problèmes avec moins d'items.

L'écart d'optimalité moyen pour la phase 3 est de 4,0 % pour TABU-CLSP et de 2,3 % pour TTM. TTM produit des solutions légèrement meilleures à mesure que le nombre d'éléments augmente. Étant donné que la phase 3 contient un grand nombre de problèmes avec 20 et 30 éléments, TTM surpasse en moyenne TABU-CLSP. Cependant, les solutions TABU-CLSP sont égales ou meilleures que les solutions TTM pour 138 instances de la phase 3. Les résultats indiquent que TABU-CLSP se compare raisonnablement à TTM, mais les temps de calcul pour TABU-CLSP sont beaucoup plus longs sur moyenne.

5. Conclusion Dans cet

article, nous avons décrit deux heuristiques de recherche tabou. Le premier, TABU-CLSPSC, concerne le CLSP avec report de configuration, qui implique à la fois des décisions de lotissement et de séquençage partiel. La deuxième heuristique, TABU-CLSP, produit des solutions pour le CLSP avec seulement des temps d'établissement. Les résultats de tests informatiques approfondis indiquent que notre heuristique est efficace pour produire de bonnes solutions à ces problèmes. TABU-CLSPSC est la seule heuristique que nous connaissons, pour traiter le report de configuration pour le CLSP dans un grand intervalle de temps.

Les écarts d'optimalité comparables pour TTM et TABU-CLSP nous aident à déterminer que notre schéma de recherche tabou de base est robuste et assez efficace. Parce qu'un schéma très similaire est utilisé dans TABU-CLSPSC, nous

estimons que notre heuristique trouve de bonnes solutions pour le CLSP avec report de configuration. Une des raisons des écarts d'optimalité un peu grands pour le TABU-CLSPSC est probablement due à la qualité des bornes inférieures.

Même si nos temps de calcul sont plus élevés par rapport à TTM, il est assez difficile de développer une heuristique basée sur la programmation mathématique adaptée pour des modèles complexes tels que le CLSP avec report de configuration. Dans de telles situations, il est pratique d'utiliser des métaheuristiques comme la recherche tabou. Un autre avantage est que cette heuristique peut être facilement étendue pour gérer des situations telles que la planification sur plusieurs machines.

Remerciements Les auteurs

tiennent à remercier les relecteurs anonymes et le rédacteur en chef adjoint pour leur relecture réfléchie et leurs précieuses suggestions. Le travail du premier auteur a été financé en partie par la subvention OGP0138205 du CRSNG, Canada.

Références

- Anderson, E.J., BS Cheah. 1993. Dimensionnement capacitif des lots avec des tailles de lots et des temps de préparation minimums. *Internat. J. Production Econom.* 30–31 137–152.
- Aras, OA, LA Swanson. 1982. Algorithme de dimensionnement et de séquençage d'un lot pour les demandes dynamiques d'une seule installation. *J. Oper. Gestion.* 2 177–185.
- Cattrysse, D., M. Salomon, R. Kuik, LN Van Wassenhove. 1993. Une heuristique à double ascension et génération de colonnes pour le problème de dimensionnement et d'ordonnement de lots discrets avec des temps de configuration. *Sciences de gestion.* 39 477–486.
- Dell'Amico, M., M. Trubian. 1993. Application de la recherche tabou au problème d'ordonnement de l'atelier de travail. *Anne. Oper. Rés.* 41 231–252.
- Diaby, M., HC Bahl, MH Karwan, S. Zionts. 1992. Une approche de relaxation lagrangienne pour le dimensionnement capacitif à très grande échelle. *Sciences de gestion.* 38 1329–1340.
- Florian, M., JK Lenstra, AHG Rinnooy Kan. 1980. Planification déterministe de la production : algorithmes et complexité. *Sciences de gestion.* 26 669–679.
- Gendreau, M., A. Hertz, G. Laporte. 1994. Une heuristique de recherche tabou pour le problème de routage de véhicules. *Sciences de gestion.* 40 1276–1290.
- Glover, F., M. Laguna. 1993. Recherche taboue. C. Reeves, éd. *Techniques heuristiques modernes pour les problèmes combinatoires.* Blackwell Scientific Publications, Oxford, Royaume-Uni, 70–150.
- , G. Kochenberger, B. Alidaee. 1998. Recherche tabou de mémoire adaptative. *Sciences de gestion.* 44 336–345.
- , E. Taillard, D. de Werra. 1993. Un guide de l'utilisateur pour la recherche tabou. *Anne. Oper. Rés.* 41 3–28.
- Gopalakrishnan, M. 2000. Un cadre modifié pour modéliser le report de configuration dans le problème de lotissement capacitif. *Internat. J. Production Res.* 38 3421–3424.

- , DM Miller, CP Schmidt. 1995. Un cadre pour modéliser le report de configuration dans le problème de dimensionnement de lot capacité. *Internat. J. Production Res.* 33 1973–1988.
- Hertz, A., E. Taillard, D. de Werra. 1997. Recherche taboue. E. Aarts, JK Lenstra, éd. Recherche locale en optimisation combinatoire. John Wiley & Sons, New York, 121–136.
- Lozano, S., J. Larraneta, L. Onieva. 1991. Approche primale-duale du problème de dimensionnement des lots à capacité unique. *Euro. J. Oper. Rés.* 51 354–366.
- Rochat, Y., E. Taillard. 1995. Diversification probabiliste et intention sification dans la recherche locale d'itinéraires de véhicules J. *Heuristics* 1 147–167.
- Salomon, M., R. Kuik, LN Van Wassenhove. 1993. Méthodes de recherche statistique pour les problèmes de lotissement. *Ann. Oper. Rés.* 41 453–468.
- Trigeiro, WW, LJ Thomas, JO McClain. 1989. Dimensionnement capacitif des lots avec temps d'installation. *Sciences de gestion.* 35 353–366.
- Wagner, HM, TM Whitin. 1958. Version dynamique du modèle de taille de lot économique. *Sciences de gestion.* 5 89–96.

Accepté par Luk Van Wassenhove; reçu le 15 juin 1998. Cet article était avec les auteurs 22 mois pour 2 révisions.