

Remplissage de camions citerne, routage et gestion d'inventaire d'hydrocarbures avec OptalCP

Diego Olivier Fernandez Pons, Petr Vilim
OptalCP

Résumé

Nous résolvons un problème de remplissage de camions citerne, transport et gestion d'inventaire d'hydrocarbures. Le problème combine ordonnancement, affectation de tâches à des shifts, emplois du temps de personnel, gestion de niveau d'inventaires et routage. Il illustre les variantes complexes de flexible-jobshop que l'on rencontre dans la pratique. Le problème étant trop complexe pour être résolu directement, nous procédons à des simplifications qui sont discutées et évaluées dans ce travail.

Mots-clés

Ordonnancement, routage, inventaire, programmation par contraintes, optalcp

Abstract

We solve a problem from the oil industry that combines scheduling, transportation with shifts, inventory management and routing. This problem is an example of the complex variants of the flexible jobshop problem that appear in industrial applications. The original problem cannot be solved in all its complexity with existing scheduling engines hence we had to evaluate various simplifications of the problem. The paper details the modeling process.

Keywords

Scheduling, routing, inventory management, constraint programming, optalcp

1 Présentation du problème

1.1 Production et transport d'hydrocarbures

L'extraction, raffinement et distribution d'hydrocarbures sont usuellement séparés en opérations *upstream*, tout ce qui est antérieur à la raffinerie, donc l'extraction de pétrole et son transport par oléoduc, et en opérations *downstream*, qui relèvent de la distribution de produits finis par oléoduc, bateau, train ou camion citerne.

Le transport downstream ressemble à un arbre où les niveaux successifs transportent des volumes de plus en plus faibles à des points de distribution de plus en plus nombreux : un oléoduc pourra par exemple desservir 2 ports depuis la raffinerie, des bateaux assureront l'acheminement vers une dizaine d'autres ports, où le pétrole sera transporté par camion vers une centaine de points de distribution (des villes) d'où il sera acheminé aux points de vente finaux (stations essence).

Tous les problèmes de transport d'hydrocarbures comportent une dimension de gestion d'inventaire : les points en aval consomment des hydrocarbures en permanence (et de façon variable en fonction de critères divers comme la saisonnalité), pendant que les points en amont produisent des hydrocarbures. Et la capacité de stockage est finie en tout point du réseau.

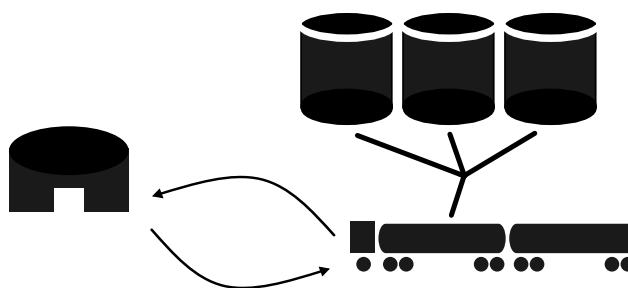


FIGURE 1 – Camion citerne à double cuve dans une ligne de remplissage ayant accès à 3 réservoirs, puis transport et décharge chez un client.

1.2 Distribution par camions citerne

Les centres de distribution planifient sur une fenêtre glissante de 14 jours de la façon suivante :

- Les camions citerne font la queue dans des lignes de distribution afin de remplir leurs cuves, ensuite ils se dirigent vers un client à livrer, déchargent après possiblement une attente, puis reviennent au centre de distribution
- Les lignes de remplissage ont des horaires de fonctionnement et peuvent distribuer plusieurs types d'hydrocarbures, avec un temps supplémentaire si changement d'hydrocarbure
- Certains clients ont des fenêtres de livraison et des installations de décharge particulières (nombre de tuyaux, volume maximal)
- Les conducteurs ont des contraintes horaires (shift de 7h + heures supplémentaires, 12h de repos minimum, 1 shift par jour maximum, 2 x 2 jours de repos consécutifs)

Dans le problème de distribution secondaire (du port vers les villes) qui nous concerne les camions ont deux (2) cuves (figure 1)

2 La solution existante

De nombreuses contraintes présentes dans la solution existante ne sont pas motivées par des limitations techniques mais par le besoin de simplifier le problème pour rendre sa résolution plus rapide. Nous appelons *approximation* une contrainte qui modifie ainsi le problème.

2.1 Approximations utilisées

Les approximations suivantes étaient déjà présentes dans la solution que le client désire remplacer

2.1.1 Simplification de la gestion d'inventaire

La motivation de ces simplifications est de réduire la taille des problèmes à résoudre en le coupant en journées.

- **Séparation du problème en journées**

L'horizon de 14 jours est divisé en 13 problèmes de 24h et les livraisons sont réparties uniformément. Les livraisons qui n'ont pu être réalisées au jour j sont reportées au jour $j + 1$.

2.1.2 Simplification du remplissage des camions

La motivation de ces simplifications est de rendre le problème d'ordonnancement plus simple.

- **Les réservoirs ont un volume illimité**

Les lignes de remplissage sont connectées à des réservoirs qui contiennent chacun un type d'hydrocarbure (figure 1). Si la capacité de ces réservoirs est considérée illimitée, le problème se réduit à des lignes de remplissage qui ont une liste de types d'hydrocarbures qu'elles peuvent utiliser pour remplir les cuves des camions citerne.

- **Les temps de remplissage des cuves est constant**

Bien que très légèrement différent pour chaque type d'hydrocarbure en raison de la différence de viscosité, on peut le considérer constant.

- **Le temps de changement de camion dans une ligne de remplissage est constant**

Il peut être inclus dans la durée de remplissage.

- **Le temps de changement d'hydrocarbure dans une ligne de remplissage est constant**

On fixe ce temps à 10 minutes indépendamment du type d'hydrocarbure.

- **Le temps de changement par un camion d'une ligne à une autre est constant**

On néglige la distance entre les lignes de remplissage.

2.1.3 Simplification du transport

Ces simplifications évitent les variantes complexes de problèmes de transport (selection de clients, multi-stop, gestion des quais, emplois du temps).

- **Clients transportant leurs hydrocarbures**

Les centres de distribution n'ayant pas la capacité de livrer à tous les clients, un accord est en place avec certains clients pour qu'ils cherchent leur hydrocarbures au centre de distribution avec leur propres camions (dont la taille et capacité sont hétérogènes).

- **Les deux cuves d'un camion citerne sont destinées au même client**

Cette limitation permet de faire du routage en étoile au lieu de routage "multi-stop".

- **Les clients n'ont qu'un tuyau de décharge**

Si un client a une capacité supérieure, il est dupliqué et les livraisons sont réparties entre les deux clients virtuels. Cela permet d'avoir un problème disjonctif à la décharge plutôt que cumulatif.

- **Pas de temps de préparation entre décharges**

Ce temps est négligé à la décharge.

2.1.4 Simplification des contraintes légales de travail

Un centre de distribution a un nombre fixe de camions et un nombre modifiable de conducteurs. En toute généralité il faudrait attribuer à la fois des livraisons et un camion à chaque shift de conducteur puis ajouter les contraintes légales de travail et enfin des temps morts pour la maintenance des camions. Mais comme le problème est divisé en journées, les camions et les conducteurs ont été fusionnés en un "shift par camion par jour".

- **Un shift par camion par jour**

Résultat de la fusion du conducteur et du camion.

- **Les shifts sont limités à 12h**

Un shift fait 7h + heures supplémentaires à distribuer avec précaution sur l'ensemble des conducteurs. La limite de 12h est une première approche à réviser selon les résultats.

- **Pas de temps de repos**

Les conducteurs ont le droit à plusieurs temps de repos (break) pendant leur shift. Ces temps de repos ont été ignorés car la solution actuelle laissait des "trous" suffisants pour les ajouter en post-traitement.

- **Pas de durée minimale entre les shifts**

Les conducteurs ont droit à 12h de repos entre les shifts. Cette contrainte est ignorée dans le problème journalisé.

- **Pas de journée repos**

Les conducteurs de camion ont le droit à des journées de repos alternant avec les journées de travail. Ces contraintes sont ignorées dans le problème journalisé.

2.2 Solution par fenêtre glissante

La solution complète opère de la façon suivante

1. L'horizon de 14 jours est découpé en 13 problèmes journaliers par pré-allocation uniforme des livraisons de chaque client aux journées.
2. Les problèmes journaliers sont résolus en séquence en reportant au jour $j + 1$ les livraisons qui n'ont pas pu être satisfaites pendant le jour j .
3. Les 13 journées sont assemblées et les shifts de 12h qui amalgament camion et conducteur sont attribués à des conducteurs en respectant les contraintes légales. Tout ce qui provoque des violations lors de l'assemblage est envoyé en 14e journée. Enfin les clients devant chercher leurs hydrocarbures au centre de distribution sont enfin ajoutés dans tous les lignes de chargement.

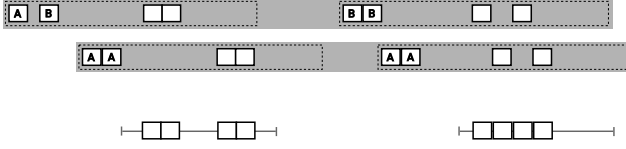


FIGURE 2 – Deux camions en gris (Truck) faisant chacun 2 voyages en pointillés (Trip). Chaque voyage comporte le remplissage des 2 cuves du camion citerne avec les hydrocarbures de type A ou B (Load), et deux décharges chez le client (Unload). En dessous les mêmes décharges à l'intérieur des fenêtres temporelles

2.3 Formalisation

Nous formalisons le problème journalier tel que résolu par la solution existante au moyen du langage d'ordonnement de OptalCP [1] (similaire à celui de IBM CP Optimizer [2]).

Cette formalisation est issue de discussions et d'observations, nous ne pouvons garantir que l'outil en place adhère complètement à cette spécification.

2.3.1 Modèle journalier de référence

Dans le problème il y a deux chargements et deux déchargements par voyage (figure 2). Afin de ne pas complexifier davantage la formulation, et parce que c'est la première simplification faite par la suite, notre modèle de référence ne fait apparaître qu'un seul chargement et déchargement.

Constantes

- Les livraisons pour le client c sont indexées par $k \in [1 \dots N^c]$
- Le nombre de livraisons du produit p pour chaque client c est N_p^c avec $\sum_p N_p^c = N^c$
- M_p^c est le nombre de livraisons pour le client c de l'hydrocarbure p qui n'ont pas pu être livrées la veille est sont obligatoires
- Nous appelons LP les paires (ligne, produit) compatibles, L_p les lignes compatibles avec le produit p et P_l les produits disponibles à la ligne l
- T_c est le temps pour conduire du centre de distribution au client c
- La matrice $setup$ est le temps de changement d'hydrocarbure dans les lignes, avec 10 minutes si les hydrocarbures sont de type différent, 0 sinon

Variables

- $Load^{c,k}$ la charge optionnelle de la k -ième livraison pour le client c
- $Load_l^{c,k}$ la charge optionnelle sur la ligne l de la k -ième livraison pour le client c
- $Load_p^{c,k}$ la charge optionnelle en produit p de la k -ième livraison pour le client c
- $Unload^{c,k}$ la décharge optionnelle de la k -ième livraison pour le client c
- $Trip^{c,k}$ le k -ième voyage optionnel pour le client c
- $Trip_t^{c,k}$ le k -ième voyage optionnel pour le client c opéré sur le camion t
- $Truck_t$ la tâche optionnelle qui représente l'ensemble des voyages du camion t

Équations

$$\max \sum_{c,k} \text{presence}(\text{Trip}^{c,k})$$

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_l^{c,k} \mid \forall l\}) \quad (1)$$

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_p^{c,k} \mid \forall p\}) \quad (2)$$

$$\forall c k \quad \text{presence}(\text{Load}^{c,k}) = \sum_{l,p \in LP} \text{presence}(\text{Load}_l^{c,k}) \times \text{presence}(\text{Load}_p^{c,k}) \quad (3)$$

$$\forall c p \quad M_p^c \leq \sum_k \text{presence}(\text{Load}_p^{c,k}) \leq N_p^c \quad (4)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_l^{c,k} \mid \forall c k\}, \text{setup}) \quad (5)$$

$$\forall c k l \quad \text{Load}_l^{c,k} \in \text{Operating hours}_l \quad (6)$$

$$\forall c k \quad \text{start}(\text{Trip}^{c,k}) = \text{start}(\text{Load}^{c,k}) \quad (7)$$

$$\forall c k \quad \text{end}(\text{Load}^{c,k}) + T_c \leq \text{start}(\text{Unload}^{c,k}) \quad (8)$$

$$\forall c k \quad \text{end}(\text{Unload}^{c,k}) + T_c = \text{end}(\text{Trip}^{c,k}) \quad (9)$$

$$\forall c k \quad \text{presence}(\text{Trip}^{c,k}) = \text{presence}(\text{Load}^{c,k}) \quad (10)$$

$$\forall c k \quad \text{presence}(\text{Trip}^{c,k}) = \text{presence}(\text{Unload}^{c,k}) \quad (11)$$

$$\forall c \quad \text{noOverlap}(\{\text{Unload}^{c,k} \mid \forall k\}) \quad (12)$$

$$\forall c k \quad \text{Unload}^{c,k} \in \bigcup \text{Time Window}_c \quad (13)$$

$$\forall c k \quad \text{alternative}(\text{Trip}^{c,k}, \{\text{Trip}_t^{c,k} \mid \forall t\}) \quad (14)$$

$$\forall t \quad \text{noOverlap}(\{\text{Trip}_t^{c,k} \mid \forall c k\}) \quad (15)$$

$$\forall t \quad \text{Truck}_t = \text{span}(\{\text{Trip}_t^{c,k} \mid \forall c k\}) \quad (16)$$

$$\forall t \quad \text{size}(\text{Truck}_t) \leq 12 \text{ hours} \quad (17)$$

Le problème peut être vu comme la fusion de plusieurs problèmes classiques

Flexible jobshop (1-6)

(1) et (2) exige que pour chaque tâche Load soient choisis une ligne et un produit qui sont obligés d'être compatibles par (3). Le nombre de livraisons d'un produit à un client ne doit pas être supérieur au nombre qu'il a commandé ni inférieur au nombre de livraisons abandonnées la veille (4). (5) impose que chargements qui ont lieu sur une ligne ne se chevauchent pas, et (6) qu'ils aient lieu aux heures d'ouverture de la ligne.

Span complexe (7-11)

(10) et (11) imposent que si une livraison a lieu, elle comporte un chargement et un déchargement (et inversement). (7) aligne le chargement et le début de la livraison, (8) qu'il y ait assez de temps entre le chargement et le déchargement pour se déplacer chez le client, (9) que le camion retourne au centre à peine son déchargement fini.

Livraison avec fenêtres temporelles (12-13)

(12) impose aux décharges de ne pas se chevaucher, (13) de s'effectuer pendant les fenêtres d'ouverture.

Bin packing temporel (14-17)

(14) exige que pour chaque tâche Trip soit choisi un camion, (15) que les voyages d'un camion ne se chevauchent pas, (16) que le shift d'un camion couvre exactement ses voyages et (17) que le shift d'un camion dure au maximum 12 heures.

2.3.2 Fonction objectif

Livraisons de la veille

L'objectif pousse le moteur à faire le maximum de livraisons possibles durant la journée tandis que la borne inférieure de la contrainte (4) garantit que les livraisons ignorées de la journée précédente seront nécessairement livrées dans la journée courante. Comme les livraisons d'un même produit sont interchangeable, cette contrainte s'exprime sur la somme et non sur les livraisons individuelles.

Dans les jeux de données fournis on considère $M_p^c = 0$, autrement dit la borne inférieure de la contrainte (4) est ignorée pour que les 13 journées puissent être testées indépendamment.

Équilibrage des heures supplémentaires

La durée de 12h pour les shifts est la durée maximale légale, en réalité les shifts font 7h + heures supplémentaires qu'il faudrait distribuer de façon équilibrée au moyen d'un objectif **leximin**. Comme le modèle journalier ne représente pas les chauffeurs, et que les objectifs de maximisation du nombre de livraisons et limitation des heures supplémentaires sont directement en conflit, cet objectif est ignoré.

2.3.3 Modèle d'assemblage des journées

Les solutions des 13 modèles journaliers forment un ensemble I d'intervalles indexés par les journées j et les camions t . Afin d'y associer un conducteur d , on introduit les variables booléennes $x_{jt}^d \in \{0, 1\}$ et le modèle linéaire

$$\max \sum_d \sum_j \sum_t x_{jt}^d$$

$$\forall d \quad \forall j \quad \sum_t x_{jt}^d \leq 1 \quad (18)$$

$$\forall d \quad \sum_{jt} x_{jt}^d \leq 10 \quad (19)$$

$$\forall d \quad \forall j_1 \quad j_2 \quad t_1 \quad t_2 \quad \text{distance}(I_{j_1 t_1}, I_{j_2 t_2}) \leq 12 \text{ hours} \\ \Rightarrow x_{j_1 t_1}^d + x_{j_2 t_2}^d \leq 1 \quad (20)$$

La contrainte (18) impose à tout conducteur au plus un shift par jour, tandis que (19) indique que le conducteur a 4 jours de repos. La contrainte (20) empêche qu'un conducteur travaille des shifts consécutifs séparés de moins de 12 heures. Les shifts qui n'auront pas pu être attribués sont reversés dans la 14e journée.

2.4 Limitations du modèle journalier

La limitation la plus sévère du modèle journalier est que le concept de conducteur ne peut pas être représenté directement, si bien que les contraintes légales liées aux conducteurs ne sont pas considérées. Quand l'algorithme d'assemblage identifie une impossibilité due à une contrainte liée aux conducteurs, il est trop tard pour remettre en cause la solution du modèle journalier.

Le choix de cette décomposition malgré ces limitations est motivé par la nécessité de réduire autant que possible la complexité du problème afin qu'il soit soluble en un temps raisonnable.

2.5 Plan de travail

Nos objectifs dans ce travail sont

1. Réduire le temps de résolution du problème journalier au moyen d'approximations. Ces approximations modifient le problème étudié (autrement dit éliminent des solutions auparavant admissibles).
2. Réduire le temps de résolution par une étude de différents modèles. Ces modèles améliorent la vitesse de résolution sans altérer la nature du problème étudié.
3. Essayer de transformer le modèle journalier ainsi amélioré en un modèle multi-jours afin d'y ajouter les considérations liées aux conducteurs.

3 Nouvelles approximations pour le problème journalier

Nous allons considérer les approximations suivantes

- La réduction de 2 cuves à une cuve double
- L'élimination des temps d'attente avant la décharge
- La spécialisation des lignes de remplissage

3.1 Reduction de deux cuves à une

Appelons une *configuration* le couple d'hydrocarbures qui est chargé dans les cuves d'un camion citerne, par exemple AA, AB, BA, BB, etc.

- Si l'on impose que les deux cuves transportent le même produit alors le nombre de configurations requises de chaque type possible est connu à l'avance. Par exemple la seule façon de livrer 6 ordres de produit A et 4 ordres de produit B est $6A + 4B = 3AA + 2BB$.
- Au contraire si les contenu des 2 cuves est indépendant, le nombre de configurations ne peut être déterminé à l'avance. Par exemple $6A + 4B$ peut être aussi obtenu par $2AA + AB + BA + BB$ ou encore $4AB + AA$.

Remplacer les doubles cuves par une cuve unique de double volume, permet de remplacer la suite d'intervalles $\text{Load}^{c,k}$ avec $k \in [1 \dots \sum_p N_p^c]$ par autant de suites d'intervalles que de produits $\text{Load}^{c,p,k}$ avec $k \in [1 \dots N_p^c]$ (figure 3).

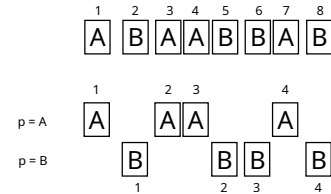


FIGURE 3 – Avoir une seule cuve permet de créer une séquence par type de produit

3.1.1 Modèles sans double alternative

La réduction de deux cuves à une permet l'utilisation de modèles sans la double alternative. Avec les variables $\text{Load}^{c,k}$ $k \in [1 \dots \sum_p N_p^c]$ il faut choisir la ligne et le produit (double alternative) puis garantir à la fois leur compatibilité et le nombre total de produits livrés. Dans le modèle

de référence les équations correspondantes sont :

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_l^{c,k} \mid \forall l\}) \quad (1)$$

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_p^{c,k} \mid \forall p\}) \quad (2)$$

$$\forall c k \quad \text{presence}(\text{Load}^{c,k}) == \sum_{l,p \in LP} \text{presence}(\text{Load}_l^{c,k}) \times \text{presence}(\text{Load}_p^{c,k}) \quad (3)$$

$$\forall c p \quad \sum_k \text{presence}(\text{Load}_p^{c,k}) \leq N_p^c \quad (4)$$

Avec les variables $\text{Load}^{c,p,k}$ $k \in [1 \dots N_p^c]$ il faut juste choisir la ligne. Le choix du produit disparaît, la compatibilité est pré-calculée via L_p et le nombre de livraisons du produit p est par construction $\leq N_p^c$

$$\forall c p k \quad \text{alternative}(\text{Load}^{c,p,k}, \{\text{Load}_l^{c,p,k} \mid l \in L_p\})$$

La seconde approche n'est pas nécessairement plus rapide car on perd la possibilité de remplacer de la contrainte `noOverlap` par des précédences après rupture de symétrie. L'étude de ces modèles devra déterminer les gains.

3.2 Élimination du temps d'attente

3.2.1 Durée minimale d'un voyage

Un voyage est défini à partir de ses composants. Les équations correspondantes dans le modèle de référence sont :

$$\forall c k \quad \text{start}(\text{Trip}^{c,k}) = \text{start}(\text{Load}^{c,k}) \quad (7)$$

$$\forall c k \quad \text{end}(\text{Load}^{c,k}) + T_c \leq \text{start}(\text{Unload}^{c,k}) \quad (8)$$

$$\forall c k \quad \text{end}(\text{Unload}^{c,k}) + T_c = \text{end}(\text{Trip}^{c,k}) \quad (9)$$

L'équation (8) est une inégalité car le camion peut arriver au point de livraison et devoir attendre pour la décharge. Inversement l'équation (9) est une égalité car il est inutile d'attendre après la dernière livraison du camion avant de revenir au centre de distribution.

Ces équations impliquent que les voyages ont une durée minimale

$$\text{size}(\text{Trip}^{c,k}) \geq 2T_c + \text{size}(\text{Load}^c) + \text{size}(\text{Unload}^c)$$

3.2.2 Durée maximale d'un voyage

Le produit étant déchargé n'étant pas pertinent pour le problème de décharge, il est inutile qu'un camion arrivé avant chez le client décharge après un camion arrivé après (s'il y avait des temps de changement de produit à la décharge ce ne serait plus le cas). Cette propriété se traduit par :

$$\begin{aligned} \forall c k \quad \text{start}(\text{Trip}^{c,k}) &\leq \text{start}(\text{Trip}^{c,k+1}) \Rightarrow \\ \text{start}(\text{Unload}^{c,k}) &\leq \text{start}(\text{Unload}^{c,k+1}) \end{aligned}$$

Parce que les décharges sont disjointes, on peut remplacer `start` par `end` dans le second membre de l'implication

$$\begin{aligned} \forall c k \quad \text{start}(\text{Trip}^{c,k}) &\leq \text{start}(\text{Trip}^{c,k+1}) \Rightarrow \\ \text{end}(\text{Unload}^{c,k}) &\leq \text{start}(\text{Unload}^{c,k+1}) \end{aligned}$$

Ces équations signifient que l'on peut imposer la conservation de l'ordre des départs et des décharges. En absence de fenêtres temporelles, l'attente d'un camion ne peut être que causée par le camion précédent (qui lui même a pu attendre en raison de celui d'avant), donc l'attente du k -ième camion est bornée par $(k-1) \times \text{size}(\text{Unload}^c)$ avec $(k \geq 1)$.

3.2.3 Fixer la durée des voyages

Quand on impose un ordre unique aux charges et décharges, en absence de fenêtres temporelles, les attentes à la décharge sont dues aux départs rapprochés des camions se dirigeant vers un même client (figure 4). Dès lors, imposer que les livraisons se fassent sans attente est équivalent à imposer une distance minimale de `size(Unload)` entre les départs pour un même client. Cela revient à constater qu'une contrainte `noOverlap`($[x_k \dots x_k + C]$) est équivalente à une contrainte `allMinDistance`(x_k, C).

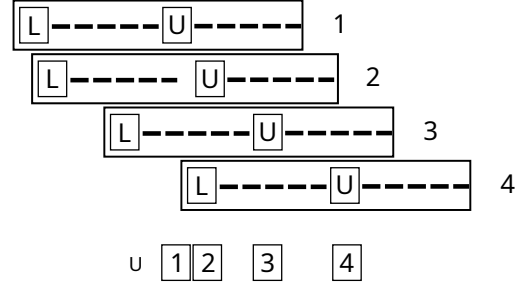


FIGURE 4 – Quatre camions qui livrent un même client, les pointillés désignent le temps de parcours, L les charge-ments et U les déchargements. On constate que U_2 doit attendre que U_1 soit fini.

3.2.4 Voyages de durée fixe et fenêtres temporelles

La présence de fenêtre temporelles peut avoir pour conséquence que l'approximation utilisant des voyages sans attente rende certaines livraisons impossibles. Les tests devront déterminer si c'est le cas.

3.2.5 Simplification de la structure des voyages

L'absence d'attente fixe la structure des voyages et permet de la simplifier : les seules contraintes agissant sur les décharges sont la contrainte `noOverlap` empêchant les décharges de se chevaucher, et les fenêtres temporelles.

$$\forall c \quad \text{noOverlap}(\{\text{Unload}^{c,k} \mid \forall k\}) \quad (12)$$

$$\forall c k \quad \text{Unload}^{c,k} \in \bigcup \text{Time Window}_c \quad (13)$$

Comme la distance entre `Load` et `Unload` est fixe en l'absence d'attente, les variables de déchargement `Unload` peuvent être éliminées et les contraintes posées sur les variables `Load`

$$\forall c \quad \text{noOverlap}(\{\text{Load}^{c,k} \mid \forall k\}) \quad (21)$$

$$\forall c k \quad \text{Load}^{c,k} \in \bigcup (\text{Time Window}_c - T_c) \quad (22)$$

3.2.6 Bin packing à éléments de taille fixe

Les équations (14 - 17) définissent un problème de bin-packing temporel où les éléments $\text{Trip}^{c,k}$ de taille variable doivent être mis dans les boîtes Truck_t .

$$\forall c k \quad \text{alternative}(\text{Trip}^{c,k}, \{\text{Trip}_t^{c,k} \mid \forall t\}) \quad (14)$$

$$\forall t \quad \text{noOverlap}(\{\text{Trip}_t^{c,k} \mid \forall c k\}) \quad (15)$$

$$\forall t \quad \text{Truck}_t = \text{span}(\{\text{Trip}_t^{c,k} \mid \forall c k\}) \quad (16)$$

$$\forall t \quad \text{size}(\text{Truck}_t) \leq 12 \text{ hours} \quad (17)$$

La taille variable des éléments empêche l'ajout d'une contrainte redondante de packing ([4] [3]). Avec l'approximation $\text{size}(\text{Trip}^{c,k}) = L^c$ les contraintes de packing peuvent être utilisées.

Les contraintes de packing ignorant la dimension temporelle du problème, leur efficacité dépendra de la propension des contraintes temporelles (précédences, etc) à créer des trous (figure 5)



FIGURE 5 – La contrainte de packing n'identifie pas les trous - les espaces dans lesquels aucun autre item ne peut aller, par exemple une précédence entre b et c crée un trou entre a et b .

3.3 Spécialisation des lignes de chargement

La figure (6) montre les produits disponibles sur chaque ligne de chargement. On constate que seules 7 lignes sont multi-produit (0 à 5 et 20).

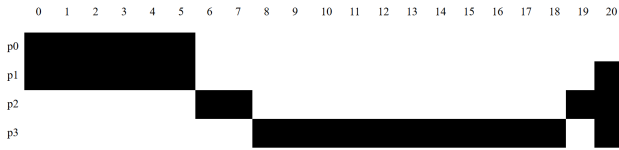


FIGURE 6 – Table de compatibilité entre lignes de remplissage et types d'hydrocarbures

Les avantages des lignes spécialisées sont l'absence de temps de changement de produit et la simplicité de gestion pratique (figure 7)

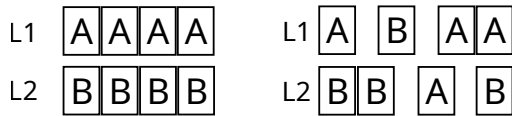


FIGURE 7 – Toutes choses égales par ailleurs, l'opérateur préfère une solution avec moins de changements (setups).

Plusieurs options de complexité croissante sont possibles

1. Spécialiser statiquement les lignes (pre-traitement) en proportion des données.
2. Modifier le modèle pour sélectionner pour chaque ligne un unique type de produit.
3. Modifier le modèle pour minimiser le nombre de changement de produit sur l'ensemble des lignes.

Nous nous limitons ici à la première approche, les deux autres options feront l'objet de travaux ultérieurs.

3.4 Modèle après approximations

Le modèle après approximations, dans la variante double index client c et produit p , s'écrit

Constantes

- Les livraisons du produit p pour le client c sont indexées par $k \in [1 \dots N_p^c]$
- M_p^c est le nombre de livraisons pour le client c de l'hydrocarbure p qui n'ont pas pu être livrées la veille est sont obligatoires
- L_p est l'ensemble des lignes compatibles avec le produit p , et les lignes ne peuvent utiliser qu'un seul type de produit ($p \neq q \Rightarrow L_p \cap L_q = \emptyset$)
- T_c est le temps requis pour se déplacer du centre de distribution au client c

Variables

- $\text{Load}^{c,p,k}$ la charge optionnelle de la k -ième livraison du produit p pour le client c
- $\text{Load}_l^{c,p,k}$ la charge optionnelle sur la ligne l de la k -ième livraison du produit p pour le client c
- $\text{Trip}^{c,p,k}$ la k -ième livraison optionnelle du produit p pour le client c
- $\text{Trip}_t^{c,p,k}$ la k -ième livraison optionnelle du produit p pour le client c opéré sur le camion t
- Truck_t la tâche optionnelle qui représente l'ensemble des voyages du camion t

Équations

$$\max \sum_{c,p,k} \text{presence}(\text{Trip}^{c,p,k})$$

$$\forall c p k \quad \text{alternative}(\text{Load}^{c,p,k}, \{\text{Load}_l^{c,p,k} \mid l \in L_p\}) \quad (23)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_l^{c,p,k} \mid \forall c p k\}) \quad (24)$$

$$\forall c p k l \quad \text{Load}_l^{c,p,k} \in \text{Operating hours}_l \quad (25)$$

$$\forall c p \quad M_p^c \leq \sum_k \text{presence}(\text{Load}^{c,p,k}) \quad (26)$$

$$\forall c k \quad \text{start}(\text{Trip}^{c,p,k}) = \text{start}(\text{Load}^{c,p,k}) \quad (27)$$

$$\forall c k \quad \text{presence}(\text{Trip}^{c,p,k}) = \text{presence}(\text{Load}^{c,p,k}) \quad (28)$$

$$\forall c \quad \text{noOverlap}(\{\text{Load}^{c,p,k} \mid \forall p k\}) \quad (29)$$

$$\forall c p k \quad \text{Load}^{c,p,k} \in \bigcup (\text{Time Window}_c - T_c) \quad (30)$$

$$\forall c k \quad \text{size}(\text{Trip}^{c,p,k}) = 2 \times T_c + 2 \times \text{size}(\text{Load}^{c,p,k}) \quad (31)$$

$$\forall c p k \quad \text{alternative}(\text{Trip}^{c,p,k}, \{\text{Trip}_t^{c,p,k} \mid \forall t\}) \quad (32)$$

$$\forall t \quad \text{noOverlap}(\{\text{Trip}_t^{c,p,k} \mid \forall c p k\}) \quad (33)$$

$$\forall t \quad \text{Truck}_t = \text{span}(\{\text{Trip}_t^{c,p,k} \mid \forall c p k\}) \quad (34)$$

$$\forall t \quad \text{size}(\text{Truck}_t) \leq 12 \text{ hours} \quad (35)$$

On reconnaît dans les équations (23 - 25) un flexible job-shop, (27 - 29) définissent une distance minimale entre les débuts de livraison pour un client c tandis que (30) leur impose des fenêtres temporelles. Les équations (31 - 35) sont un bin-packing temporel. La contrainte (26) impose un minimum de livraisons.

4 Modélisation des sous-problèmes

Une fois les approximations fixées (section 3.4), nous considérons les différents modèles possibles pour les sous problèmes identifiés.

4.1 Flexible jobshop avec modes

Considérons à nouveau les équations du sous-problème de sélection des lignes et des produits

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_l^{c,k} \mid \forall l\}) \quad (1)$$

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_p^{c,k} \mid \forall p\}) \quad (2)$$

$$\forall c k \quad \text{presence}(\text{Load}^{c,k}) == \sum_{l,p \in LP} \text{presence}(\text{Load}_l^{c,k}) \times \text{presence}(\text{Load}_p^{c,k}) \quad (3)$$

$$\forall c p \quad \sum_k \text{presence}(\text{Load}_p^{c,k}) \leq N_p^c \quad (4)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_l^{c,k} \mid \forall c k\}) \quad (5)$$

Ce problème peut être vu comme un flexible-jobshop avec

- un job par client
- une ressource principale, la ligne sur laquelle est exécutée la tâche de remplissage (1) avec une contrainte d'absence de chevauchement (5)
- une ressource secondaire, le type de produit (2)
- une contrainte de compatibilité entre les ressources primaires et secondaires (3)
- une contrainte de nombre total de ressources secondaires utilisées (4)

Les contraintes secondaires de cette forme sont parfois appelées **modes** et sont très courantes dans les problèmes réels de manufacture (outils, travailleurs, etc).

Plusieurs modèles sont possibles pour représenter un flexible jobshop avec modes

1. double alternative
2. alternatives hiérarchiques - ligne en premier
3. alternatives hiérarchiques - produit en premier
4. séparation par mode
5. alternative simple bidimensionnelle

4.1.1 Double alternative

Ce sont les équations (1- 5). L'équation de compatibilité (3) se simplifie en vertu de la spécialisation des lignes :

$$\forall c k p \quad \text{presence}(\text{Load}_p^{c,k}) == \sum_{l \in L_p} \text{presence}(\text{Load}_l^{c,k}) \quad (36)$$

indiquant que si le produit p est choisi pour la charge (c, k) , elle doit avoir lieu dans une ligne compatible avec p .

Si on remplace (36) dans le décompte des produits (4), on constate ne plus avoir besoin des variables choisissant les produits. On retrouve le modèle d'alternative hiérarchique avec ligne en premier.

4.1.2 Alternatives hiérarchiques - ligne en premier

La double alternative peut être transformée en une alternative hiérarchique dans laquelle on choisit la ligne et conditionnellement à cette ligne, le type de produit.

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_l^{c,k} \mid l \in \text{lines}\}) \quad (37)$$

$$\forall c k l \quad \text{alternative}(\text{Load}_l^{c,k}, \{\text{Load}_{l,p}^{c,k} \mid p \in P_l\}) \quad (38)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_l^{c,k} \mid \forall c \forall k\}) \quad (39)$$

$$\forall c p \quad \sum_k \sum_{l \in L_p} \text{presence}(\text{Load}_{l,p}^{c,k}) \leq N_p^c \quad (40)$$

La contrainte (2) devient conditionnelle à la ligne l , la contrainte de compatibilité disparaît dans le pré-calcul de P_l . Sous hypothèse supplémentaire de spécialisation des lignes on a $P_l = \{p\}$ donc la seconde alternative disparaît entièrement

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_l^{c,k} \mid l \in \text{lines}\}) \quad (41)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_l^{c,k} \mid \forall c \forall k\}) \quad (42)$$

$$\forall c p \quad \sum_k \sum_{l \in L_p} \text{presence}(\text{Load}_l^{c,k}) \leq N_p^c \quad (43)$$

4.1.3 Alternatives hiérarchiques - produit en premier

Si on choisit le produit en premier on obtient

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_p^{c,k} \mid \forall p\}) \quad (44)$$

$$\forall c k p \quad \text{alternative}(\text{Load}_p^{c,k}, \{\text{Load}_{l,p}^{c,k} \mid l \in L_p\}) \quad (45)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_{l,p}^{c,k} \mid \forall c k p\}) \quad (46)$$

$$\forall c p \quad \sum_k \text{presence}(\text{Load}_p^{c,k}) \leq N_p^c \quad (47)$$

Dans ce cas on a $k \in [1 \dots \sum_p N_p^c]$ et il y a $(\sum_p N_p^c) \times |P|$ variables de type $\text{Load}_p^{c,k}$. On peut réduire le nombre de variables en séparant les indexes $k \in [1 \dots \sum_p N_p^c]$ par type de produit, donnant lieu à la séparation par modes

4.1.4 Séparation par modes

Si on sépare l'indice $k \in [1 \dots \sum_p N_p^c]$ en $\forall p k_p \in [1 \dots N_p^c]$, et si on fixe le type de produit p pour chacun de ces nouveaux indices, on obtient le modèle

$$\forall c p k \quad \text{alternative}(\text{Load}^{c,p,k}, \{\text{Load}_l^{c,p,k} \mid l \in L_p\}) \quad (48)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_l^{c,p,k} \mid \forall c k p\}) \quad (49)$$

4.1.5 Alternative bidimensionnelle

On peut enfin faire une alternative simple sur un indice bi-dimensionnel représentant les couples ligne $\times$ produit admissibles.

$$\forall c k \quad \text{alternative}(\text{Load}^{c,k}, \{\text{Load}_{l,p}^{c,k} \mid (l, p) \in LP\}) \quad (50)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_{l,p}^{c,k} \mid \forall c k p \in P_l\}) \quad (51)$$

$$\forall c p \quad \sum_k \sum_l \text{presence}(\text{Load}_{l,p}^{c,k}) \leq N_p^c \quad (52)$$

Si chaque ligne a un seul produit, le produit p devient une fonction de l et les équations deviennent identiques aux équations que l'alternative hiérarchique qui commence par choisir la ligne, après simplification en vertu de l'approximation de ligne spécialisée (41 - 43)

4.2 Rupture de symétrie et noOverlap

Les ruptures de symétrie sont des **dominations** : ce sont des approximations, donc elles éliminent des solutions mais gardent un représentant de chaque classe de solutions si bien qu'au moins une solution optimale est conservée.

Quand les intervalles impliqués dans une contrainte noOverlap sont identiques, on peut introduire un ordre entre eux et éliminer la contrainte de non chevauchement. Deux approches sont possibles

Précédences

$$\forall c \forall i < j \quad \text{end}(\text{Load}^{c,i}) \leq \text{start}(\text{Load}^{c,j})$$

chaînes

$$\forall c \ k \quad \text{end}(\text{Load}^{c,k}) \leq \text{start}(\text{Load}^{c,k+1})$$

$$\forall c \ k \quad \text{presence}(\text{Load}^{c,k+1}) \Rightarrow \text{presence}(\text{Load}^{c,k})$$

4.2.1 Impact de la rupture de symétrie sur les équations de flexible jobshop

Le modèle de séparation par modes (section 4.1.4) utilise les variables $\text{Load}^{c,p,k}$. On peut poser une rupture de symétrie par couple (c, p) mais il restera une contrainte noOverlap pour l'ensemble des produits

En revanche, en absence de rupture de symétrie, le modèle *alternative hiérarchique - ligne en premier* (section 4.1.2) et le modèle *séparation de modes* (section 4.1.4) se confondent car on peut artificiellement ordonner les variables $\text{Load}^{c,k}$ en décidant que les N_p^c premières seront attribuées le produit p , etc.

Ces modèles ne sont différents que si on pose des précédences sur $\text{Load}^{c,k}$ de la forme $\text{end}(\text{Load}^{c,k}) \leq \text{start}(\text{Load}^{c,k+1})$. De manière plus générale une séquence $\text{Load}^{c,0} \text{Load}^{c,1} \dots \text{Load}^{c,k}$ n'implique pas nécessairement des précédences entre ces tâches. Et on peut exploiter cette liberté ou bien pour imposer une chaîne de précédences, ou bien pour fixer les produits de certaines tâches.

4.3 Fenêtres temporelles

4.3.1 Fenêtres et chaînes

Quand les décharges forment une chaîne et doivent avoir lieu à l'intérieur de fenêtres temporelles, on peut raffiner les temps de début de certaines décharges dans la chaîne. Par exemple dans la figure 8 la décharge 5 ne peut avoir lieu dans la première fenêtre temporelle.



FIGURE 8 – Deux fenêtres temporelles et 6 décharges

4.3.2 Disjonction sur les fenêtres

Considérer une disjonction sur les fenêtres de livraison, c'est à dire un intervalle par livraison par fenêtre semble raisonnable dans la mesure où (i) seule la moitié des clients a des fenêtres de livraison et (ii) dans le jeu de données existant il n'y a que deux fenêtres (9h-12h et 14h-16h).

5 Extension à plusieurs journées

Nous étendons maintenant le modèle journalier à 7 jours, en séparant conducteurs et camions. Les contraintes portant sur les conducteurs sont

- nb_conducteurs conducteurs
- nb_camions camions
- 5 journées de travail (shifts) de 8h par conducteur
- 2 jours de repos consécutifs par conducteur
- 12 heures entre shifts
- au plus un shift par jour
- les shifts commencent et terminent à l'heure ronde

5.1 Représentation des shifts

Étendre le modèle journalier à plusieurs jours exige de revisiter le sous problème de transport (14 - 17) qui garantit que les voyages ont lieu à l'intérieur des shifts des camion/conducteur choisis. En particulier la contrainte span qui garantissait que les voyages aient lieu à l'intérieur du shift journalier du camion doit être étendue à plusieurs shifts.

5.1.1 Alternative sur les conducteurs $\times$ shifts

Considérons les variables

- Shift_d^s représentant le s -ième shift $s \in [1 \dots 5]$ du conducteur d
- $\text{Trip}_{d,s}^{c,k}$ représentant le fait que le voyage (c, k) est effectué par le conducteur d dans son shift s

Les équations suivantes remplacent les équations (14 - 17)

$$\forall c \ k \quad \text{alternative}(\text{Trip}^{c,k}, \{\text{Trip}_{d,s}^{c,k} \mid \forall d \forall s\}) \quad (53)$$

$$\forall d \forall s \quad \text{Shift}_d^s = \text{span}(\{\text{Trip}_{d,s}^{c,k} \mid \forall c \forall k\}) \quad (54)$$

$$\forall d \forall s \quad \text{noOverlap}(\{\text{Trip}_{d,s}^{c,k} \mid \forall c \forall k\}) \quad (55)$$

Cette représentation crée nb_voyages $\times$ nb_shifts $\times$ nb_conducteurs intervalles pour représenter les voyages

5.1.2 noOverlap avec anti-shifts

Une seconde approche évite l'alternative en introduisant une contrainte noOverlap entre les tâches représentant le complémentaire des shifts (les anti-shift) et les voyages (figure 9)

$$\forall c \ k \quad \text{alternative}(\text{Trip}^{c,k}, \{\text{Trip}_d^{c,k} \mid \forall d\}) \quad (56)$$

$$\forall d \forall s \quad \text{end}(\text{Anti_shift}_s^d) = \text{start}(\text{Shift}_s^d) \quad (57)$$

$$\forall d \forall s \quad \text{end}(\text{Shift}_s^d) = \text{start}(\text{Anti_shift}_{s+1}^d) \quad (58)$$

$$\forall d \quad \text{noOverlap}(\{\text{Anti_shift}_s^d \mid \forall s\} \cup \{\text{Trip}_d^{c,k} \mid \forall c \forall k\}) \quad (59)$$

Cette représentation crée nb_voyages $\times$ nb_conducteurs intervalles pour représenter les voyages



FIGURE 9 – tâches entre les anti-shifts en noir.

5.2 Modèle multi-jours

Nous pouvons à présent écrire un modèle pour une semaine

Variables

- $\text{Load}^{c,p,k}$ la charge optionnelle de la k -ième livraison du produit p pour le client c
- $\text{Load}_l^{c,p,k}$ la charge optionnelle sur la ligne l de la k -ième livraison du produit p pour le client c
- $\text{Trip}^{c,p,k}$ la k -ième livraison optionnelle du produit p pour le client c
- $\text{Trip}_t^{c,p,k}$ la k -ième livraison optionnelle du produit p pour le client c opéré sur le camion t
- $\text{Shift}^{d,s}$ le s -ième shift du conducteur d
- $\text{Anti_shift}^{d,s}$ l'espace entre les shifts $s - 1$ et s pour le conducteur d
- $\text{Shift_day}^{d,s}$ intervalle indiquant la journée dans laquelle se trouve le shift s du conducteur d
- Off^d sequence de 2 jours de repos pour le conducteur d

Équations

$$\max \sum_{c,p,k} \text{presence}(\text{Trip}^{c,p,k})$$

$$\forall c \forall p \forall k \quad \text{alternative}(\text{Load}^{c,p,k}, \{\text{Load}_l^{c,p,k} \mid \forall l \in L_p\}) \quad (60)$$

$$\forall l \quad \text{noOverlap}(\{\text{Load}_l^{c,p,k} \mid \forall c \forall p \in P_l \forall k\}) \quad (61)$$

$$\forall c \forall p \forall k \forall l \quad \text{Load}_l^{c,p,k} \in \bigcup \text{Operating hours}_l \quad (62)$$

$$\forall c \forall k \quad \text{size}(\text{Trip}^{c,k}) = 2 \times (T_c + \text{size}(\text{Load}^c)) \quad (63)$$

$$\forall c \forall p \forall k \quad \text{start}(\text{Trip}^{c,p,k}) = \text{start}(\text{Load}^{c,p,k}) \quad (64)$$

$$\forall c \quad \text{noOverlap}(\{\text{Load}^{c,p,k} \mid \forall p \forall k\}) \quad (65)$$

$$\forall c \forall p \forall k \quad \text{Load}^{c,p,k} \in \bigcup \text{Time Window}_c - T_c \quad (66)$$

$$\forall c \forall k \quad \text{alternative}(\text{Trip}^{c,k}, \{\text{Trip}_d^{c,k} \mid \forall d\}) \quad (67)$$

$$\forall d \forall s \quad \text{end}(\text{Anti_shift}_s^d) = \text{start}(\text{Shift}_s^d) \quad (68)$$

$$\forall d \forall s \quad \text{end}(\text{Shift}_s^d) = \text{start}(\text{Anti_shift}_{s+1}^d) \quad (69)$$

$$\forall d \forall s \quad \text{size}(\text{Anti_shift}_s^d) \geq 12 \text{ hours} \quad (70)$$

$$\forall d \forall s \quad \text{size}(\text{Shift}_s^d) \leq 7 \text{ hours} \quad (71)$$

$$\forall d \quad \text{noOverlap}(\{\text{Anti_shift}_s^d \mid \forall s\} \cup \{\text{Trip}_d^{c,k} \mid \forall c \forall k\}) \quad (72)$$

$$\forall d \forall s \quad \text{start}(\text{Shift_day}_s^d) = \text{start}(\text{Shift}_s^d) \text{ div } 24 \text{ hours} \quad (73)$$

$$\forall d \quad \text{noOverlap}(\{\text{Shift_day}_s^d \mid \forall s\} \cup \{\text{Off}^d\}) \quad (74)$$

$$\forall d \forall s \quad \text{size}(\text{Shift}_s^d) = 1 \quad (75)$$

$$\forall d \quad \text{size}(\text{Off}^d) = 2 \quad (76)$$

$$\sum_d \sum_s \text{pulse}(\text{Shift}_s^d) \leq \text{nb_trucks} \quad (77)$$

Les équations (60 - 66) définissent le flexible jobshop comme auparavant. L'équation (67) impose que pour tout voyage soit choisi un conducteur - et non plus un camion comme auparavant. Les équations (68 - 69) construisent la séquence de shifts et anti-shifts, tandis que les équations (70) et (71) imposent des contraintes sur la durée des shifts

et la durée des espaces entre les shifts (les anti-shifts). La contrainte (72) impose que les voyages et les anti-shifts ne se chevauchent pas, autrement dit que les voyages se placent quelque part dans les shifts / entre les anti-shifts. L'équation (73) synchronise les shifts et leur représentation modulo 24h tandis que l'équation (74) impose un non chevauchement entre les journées travaillées et les deux jours de repos consécutifs. Les contraintes (75) et (76) fixent la durée en jours des journées de travail et de repos. Enfin la contrainte (77) impose qu'à tout instant il n'y ait pas plus de shifts de conducteurs que de camions disponibles.

5.3 Taille du modèle multi-jours

En théorie c'est au modèle de déterminer combien de conducteurs sont nécessaires pour les opérations mais un nombre variable de conducteurs s'accommode mal avec la représentation en shifts et anti-shifts. A la place nous faisons un calcul de dimensionnement de la forme suivante

- dans le modèle journalier il y a $H = \text{nb_camions} \times 12h$ heures de travail
- dans le modèle multi-jour il y a $M = \text{nb_conducteurs} \times \text{nb_jours_de_travail} \times 7h$ heures de travail
- on pose $7 \times H = M$ pour un modèle sur 7 jours

Numériquement on obtient $c = \frac{7 \times H}{M/c} = \frac{7 \times 61 \times 12}{5 \times 7} = 147$ conducteurs pour 2000 livraisons. L'alternative qui permet d'allouer une livraison à un conducteur (67) crée donc 300 000 intervalles, ce qui est à la limite de ce que peuvent supporter les moteurs d'ordonnancement comme OptalCP ou CPO. Donc le problème multi-jour est encore trop grand.

6 Expérimentations

6.1 Données de test

Les données contiennent les ordres de livraison pour 13 journées indépendantes (entre 43 et 373 livraisons par jour) tout comme les clients qui chargent leurs hydrocarbures sur leurs propre flotte. La journée 10 ne compte que 43 livraisons car c'est un jour férié où seul 1 shift de 8h de travail est possible mais nous la résolvons avec les mêmes contraintes que les autres pour l'instant.

La séparation en 13 journées est due à l'heuristique de fenêtres glissantes de la solution en place. Les données brutes pour un horizon de 14 jours sont les ordres de livraison des 13 journées concaténés, et tout autre partition de ces données est un jeu de test valide.

Ces données et les modèles testés sont disponibles en ligne à l'adresse <https://optalcp.com/docs/benchmarks/> sous le nom tank-truck

6.2 Modèles testés

Nous avons considéré

1. Approximations

- réduction de 2 cuves à une
- élimination du temps d'attente
- spécialisation des lignes

2. Ruptures de symétrie pour les chargements

- (a) noOverlap (pas de rupture de symétrie)
- (b) précédences
- (c) chaînes (traitement spécial par OptalCP)
- (d) chaînes suivant la définition

3. Représentations du flexible jobshop

- (a) double alternative
- (b) alternatives hiérarchiques - ligne en premier
- (c) alternatives hiérarchiques - produit en premier
- (d) alternative bidimensionnelle
- (e) séparation par modes

4. Représentations des fenêtres temporelles

- (a) calendriers fournis par le moteur (forbiddenExtent)
- (b) bornes en présence de chaînes
- (c) disjonction sur la fenêtre

5. Représentations du sous problème de bin-packing

- (a) alternatives
- (b) alternatives + contrainte pack redondante

Par ailleurs, en fonction des approximations utilisées certaines de ces représentations fusionnent (notamment les modèles pour le flexible jobshop avec la spécialisation des lignes ou l'absence de rupture de symétrie).

Nous ne présenterons ici que les résultats pour deux modèles (1abc2a3b4a5a) et (1abc2c3e4a5a) autrement dit toutes les approximations et d'un côté l'alternative hiérarchique ligne en premier avec rupture de symétrie par chaîne, de l'autre le modèle par séparation de modes sans rupture de symétrie. Nous les nommons **3b** et **3e** respectivement.

6.3 Moteurs testés

Nous n'avons pas l'autorisation de présenter les résultats obtenus par la solution en place chez le client. A défaut nous comparons **OptalCP 2024.11.1.20** contre **ILOG CP Optimizer 22.1.1**

Les test on eu lieu en 300 secondes sur un laptop 4 cores i7-1185G7@3.00GHz avec 16GB de RAM.

6.4 Résultats

Quelques solutions des instances journalières pour OptalCP et CPO en 5 minutes.

Les bornes inférieures (autrement dit d'autres solutions au problème de maximisation) de la colonne LB ont été trouvées en laissant OptalCP tourner pendant 10-30 minutes pour les instances 5,6,15,13,11 et une semaine pour les instances 4 et 12. Les bornes supérieures sont égales au nombre d'ordres de livraison, aucun moteur n'ayant pu démontrer qu'il n'est pas possible de livrer tous les ordres. Le modèle sans rupture de symétrie obtient de meilleurs résultats.¹ Tous les autres variantes de modèles ont des résultats significativement inférieurs, et les 3 approximations réduisent grandement le problème par rapport au modèle de référence (a 2 cuves).

1. Une version antérieure de cette étude montrait des résultats plus proches mais les améliorations dans OptalCP ont bénéficié davantage **3e**.

jour	livraisons	LB	UB	OptalCP		CPO
				3b	3e	3e
10	43	43	43	43	43	43
9	294	294	294	294	294	294
16	319	319	319	317	319	317
8	322	322	322	322	322	320
14	331	331	331	331	331	326
7	341	341	341	341	340	336
5	344	341	341	337	339	329
6	344	344	344	338	344	305
15	349	349	349	339	348	334
13	358	358	358	346	353	339
11	366	366	366	349	352	317
4	371	369	371	339	353	344
12	373	367	373	342	357	339

TABLE 1 – Résultats de OptalCP et CPO sur les 13 journées considérées indépendamment. En gras le modèle+moteur ayant obtenu la meilleure solution dans ce test pour une instance donnée.

7 Conclusion

Nous avons résolu un problème de transport d'hydrocarbures par camion citerne avec le moteur d'ordonnancement OptalCP. Le problème est un exemple de problème réel complexe où sont mélangés des aspects de flexible-jobshop, de tournée de véhicules avec fenêtres, d'emplois du temps de personnel et de gestion d'inventaire. Après analyse de chaque sous-problème nous avons proposé diverses approximations et modélisations que nous avons ensuite testées. Un jeu d'instances public a été crée afin que d'autres équipes puissent également tester leurs approximations et modèles. Il nous semble important de souligner une fois de plus qu'il s'agit d'un problème dont les contours ne sont pas fixés et que toutes les approximations et modèles qui en pratique aboutissent à une amélioration sont acceptables.

Références

- [1] Diego Olivier Fernandez Pons and Petr Vilím. OptalCP un nouveau moteur d'ordonnancement. *Journées Francophones de Programmation par Contraintes 2024 (JFPC 2024)*.
- [2] Philippe Laborie, Jérôme Rogerie, Paul Shaw, and Petr Vilím. IBM ILOG CP optimizer for scheduling : 20+ years of scheduling with constraints at IBM/ILOG. *Constraints*, 23 :210–250, 2018.
- [3] Pierre Schaus, Yves Deville, et al. A global constraint for bin-packing with precedences : Application to the assembly line balancing problem. In *AAAI*, pages 369–374, 2008.
- [4] Paul Shaw. A constraint for bin packing. In *International conference on principles and practice of constraint programming*, pages 648–662. Springer, 2004.