Problème d'équilibre des charges de travail dans l'affectation de patients aux infirmières. *

Pierre Schaus, Pascal Van Hentenryck, Jean-Charles Régin

Dynadec, One Richmond Square, Providence, RI 02906, USA
Brown University, Box 1910, Providence, RI 02912, USA
Université de Nice-Sophia Antipolis, France
pschaus@dynadec.com pvh@cs.brown.edu regin@polytech.unice.fr

Résumé

Cet article traite de l'affectation journalière d'enfants à des infirmières dans un hôpital. L'objectif est d'équilibrer la charge de travail des infirmières tout en satisfaisant diverses contraintes. Des travaux précédents ont proposé un modèle MIP pour ce problème, qui malheureusement rencontre des difficultés pour résoudre de grandes instances. De plus ce modèle MIP ne fait qu'approximer la fonction objectif. En effet la minimisation de la variance n'est pas une expression linéaire. Des modèles de programmation par contraintes (PC) sont présentés de complexités croissante permettant finalement de résoudre des instances de grande tailles avec des centaines de patients et d'infirmières en quelques secondes avec le système d'optimisation Comet. Les modèles de PC utilisent la contrainte globale spread pour minimiser la variance ainsi qu'une technique de décomposition du problème.

1 Introduction

Cet article traite le problème d'affectation journalière de bébés aux infirmières dans un hôpital décrit dans [5]. Dans ce problème, certains enfants ne demandent pas trop d'attention, par contre d'autres demandent une attention considérable. La quantité de travail requise par un enfant est appelée son acuité. Une infirmière sera chargée de s'occuper d'un groupe d'enfants et la somme des acuités des enfants dont elle a la charge est sa charge de travail. Il est essentiel que les charges de travail soient correctement équilibrées entre les infirmières. Cela permet d'assurer à la fois des soins de santé de meilleure qualité mais également de concevoir des horaires équitables entre les infirmières. La solution finale devra également satisfaire diverses contraintes additionnelles imposées par la législation en vigueur :

- Une infirmière ne peut travailler que dans une zone de l'hôpital alors que les enfants sont localisés dans p zones différentes.
- Une infirmière ne peut pas prendre en charge plus de $children^{\max}$ enfants.
- L'acuité totale d'une infirmière ne peut pas depasser $acuity^{\max}$.

L'objectif d'équilibre et les contraintes additionnelles rendent le problème difficile à résoudre. Etant donné que les infirmières ne peuvent travailler que dans une zone, le nombre d'infirmières attribuées à chaque zone a déjà une grande influence sur la qualité d'équilibre des charges de travail dans la solution finale.

Ce problème a initialement été résolu dans [5] avec un modèle MIP. Malheureusement les résultats en terme de qualité d'équilibre et temps de calcul ne sont pas satisfaisants. Dans cet article, nous présentons une série de modèles de programmation par contraintes de complexités croissantes. Notre dernier modèle permet de d'obtenir rapidement des solutions de grande qualité tout en passant très bien à l'échelle lorsque la taille de l'instance augmente.

L'organisation de l'article est la suivante. La Section 2 présente les instances proposées dans [5] et la Section 3 décrit le modèle MIP et ses limitations. La Section 4 rappelle la contrainte *spread* utilisée pour équilibrer les charges de travail ainsi que la caractérisation de son filtrage tel qu'implémenté dans Comet. La Section 5 présente le premier modèle de programmation par contrainte (PC). Ce modèle permet la résolution d'instances avec deux zones. La Section 6 présente une approche en deux temps où les infirmières sont d'abord

*traduction d'un article accepté à CPAIOR09

affectées aux zones avant de leur assigner des enfants. Finalement, la Section 7 montre que la deuxième étape peut être décomposée par zone sans perdre de garantie sur l'optimalité. Ce dernier modèle permet la résolution d'instances de grandes tailles avec des dizaines de zones et des centaines de patients.

2 Instances du problème

Un modèle statistique introduit dans [5] permettant de générer des instances similaires aux instances réelles rencontrées par les auteurs. Ce modèle statistique a également été utilisé dans leur article pour tester la robustesse de leur approche par rapport aux nombre d'infirmières, d'enfants, et de zones. Ce modèle ne comprend qu'un seul paramètre : le nombre de zones. L'acuité maximum par infirmière est fixée à $acuity^{\max} = 105$ et le nombre maximum d'enfants par infirmière est fixé à $children^{\max} = 3$. Le générateur d'instances fixe le nombre d'infirmières, le nombre d'enfants et leur acuité ainsi que la zone à laquelle il appartient. Voici précisément les différentes étapes suivies par le modèle statistique pour générer une instance :

- Le nombre de patients par zone suit une distribution de Poisson avec une moyenne 3.8 qui est décalée de +10.
- L'acuité Y d'un patient est obtenue en générant d'abord un nombre $X \sim Binomial(n = 8, p = 0.23)$ et ensuite en générant l'acuité $Y \sim Unif(10 \cdot (X+1), 10 \cdot (X+1) + 9)$.
- Le nombre total d'infirmière est obtenu en solutionnant une procédure *First Fit Decreasing* (FFD) dans chaque zone. Plus précisément, le nombre total est le nombre d'infirmières trouvées dans chaque zone par la procédure FFD. La procédure FFD commence par trier les patients par ordre décroissant d'acuité. Ensuite le patient avec la plus grande acuité est assigné à la première infirmière. Les patients suivants sont assignés successivement à la première infirmière pouvant l'accepter sans violer la contrainte d'acuité maximale et la contrainte du nombre maximum de patients par infirmière.

3 Le modèle MIP

Par manque de place, nous ne reproduisons pas ici le modèle MIP introduit dans [5] dans son entièreté. Nous donnons les principales variables du modèle ce qui est suffisent pour le cerner. Nous expliquons ensuite les limites de ce modèle et nous expliquons pourquoi une approche de PC peut pallier à ces limites. Le modèle MIP contient quatre familles de variables :

1. $X_{ij} = 1$ si l'enfant i est affecté à l'infirmière j et 0 sinon ;
2. $Z_{jk} = 1$ si l'infirmière j est assigné à la zone k et 0 sinon ;
3. $Y_{k,\max}$ est l'acuité maximum parmi toutes les infirmières de la zone k ;
4. $Y_{k,\min}$ est l'acuité minimum parmi toutes infirmières de la zone k.

Toutes ces variables sont liées par des contraintes linéaires pour satisfaire les contraintes du problème. La fonction objectif implémente ce que nous appelons le critère *range-sum* qui consiste en la minimisation de la somme des écarts entre l'acuité maximum et minimum dans les p zones, i.e.,

$$\sum_{k=1}^{p} (Y_{k,\max} - Y_{k,\min}).$$

Le modèle MIP comporte plusieurs limitations : La fonction objectif peut produire des solutions très inégales en termes de charge de travail. En effet elle tend à égaliser les charges de travail intra zone mais peut produire de grandes différences entre les temps de travail inter zone. Ce fait est illustré sur la Figure 1. Les charges de travail sont montrées en haut à droite de chacune des visualisations Comet. La solution de gauche est obtenue en minimisant le critère range-sum alors que la solution de droite est obtenue en minimisant la variance (nommée norme L_2 dans la section suivante). L'objectif range-sum est optimal dans la solution de gauche puisque les temps de travail à l'intérieur de chaque zone sont identiques. Malheureusement les infirmières de la première zone travaillent deux fois plus que les infirmières de la seconde zone. La solution de droite obtenue par minimisation de la variance est significativement meilleure. *Cette exemple illustre clairement que l'objectif que "toutes les infirmières devraient recevoir une même charge de travail" [5] n'est pas garanti avec le critère range-sum.*

Il n'apparait pas directement comment remédier à ce problème dans le modèle MIP. En effet la variance est un critère non linéaire et ne peut être modéliser facilement dans une approche MIP. De plus, une approche de PC peut exploiter directement l'aspect combinatoire de la structure du bin-packing et des contraintes additionnelles tandis que le MIP possède une relaxation linéaire généralement assez mauvaise pour les problèmes de bin-packing. Finalement, le modèle MIP n'évite pas certaines symétries du problème : pour une solution donnée, les infirmières sont complètement interchangeables. Dans la suite nous rappelons les contraintes d'équilibre existantes en PC avant d'introduire les modèles Comet de PC pour résoudre le problème.

FIGURE 1 – Comparaison entre deux solutions sur une instance à 6 infirmières, 14 enfants et 2 zones. La solution de gauche est obtenue en minimisant le critère `range-sum`. La solution de droite est obtenue en minimisant la variance des charges de travail.

4 Les contraintes d'équilibre en PC

Les contraintes d'équilibre apparaissent dans de nombreuses applications réelles, la plupart du temps pour exprimer une répartition équitable d'objets ou du travail. Par exemple, Simonis [15] suggère d'utiliser une contrainte globale pour équilibrer la distribution des équipes dans les problèmes de rostering. Pesant propose dans [7] d'utiliser une contrainte d'équilibre pour une allocation équitable des horaires individuels.

Deux contraintes globales et leurs propagateurs ont été proposés en programmation par contrainte pour optimiser l'équilibre : `spread` [6, 11], qui contraint la variance et la moyenne d'un ensemble de variables, et `deviation` [12, 13], qui contraint la moyenne et l'écart absolu moyen à la moyenne d'un ensemble de variables. On dit aussi que `spread` et `deviation` contraignent respectivement les normes L_2 et L_1 d'un ensemble de variables $X_1..X_n$ par rapport à leur moyenne ($s = \sum_{i\in[1..n]} X_i$), i.e.,

- $L_1 : \sum_{i\in[1..n]} |X_i - s/n|$;
- $L_2 : \sum_{i\in[1..n]} (X_i - s/n)^2$.

Ces deux critères ne sont pas équivalents : Minimiser L_1 ou L_2 ne conduit pas aux mêmes solutions et il n'est pas toujours évident de choisir l'un plutôt que l'autre. Ce choix entre L_1 et L_2 est récurrent et ne date pas d'aujourd'hui (voir par exemple [3]). Pour cette application, nous utilisons le critère L_2 et sa contrainte `spread` car L_2 est plus sensible aux points extrêmes que nous voulons absolument éviter dans cette application.

Nous utilisons les définitions et notations suivantes pour décrire la sémantique de la contrainte `spread` et de ses propagateurs.

Definition 1 *Soit X une variable à domaine fini (discrète). Le domaine de X est un ensemble de valeurs entières ordonnées pouvant être assignées à X dénoté $Dom(X)$. La valeur minimum (resp. maximum) du domaine est dénotée par $X^{\min} = \min(Dom(X))$ (resp. $X^{\max} = \max(Dom(X))$. Un intervalle entier aux bornes entières a et b est dénoté $[a..b] \subseteq \mathbb{Z}$, alors que nous dénotons $[a, b] \subseteq \mathbb{Q}$ l'intervalle rationnel. Une affectation des variables $\boldsymbol{X} = [X_1, X_2, ..., X_n]$ est dénoté par le tuple $\mathbf{x}$ et la ième entrée de ce tuple par $\mathbf{x}[i]$. L'intervalle domaine étendu rationnel de X_i est $I_D^{\mathbb{Q}}(X_i) = [X_i^{\min}, X_i^{\max}]$ et son intervalle domaine étendu entier est $I_D^{\mathbb{Z}}(X_i) = [X_i^{\min} \,..\, X_i^{\max}]$.*

Nous définissons maintenant la contrainte `spread` avec moyenne fixe.

Definition 2 *Etant donné les variables à domaine fini $\boldsymbol{X} = (X_1, X_2, ..., X_n)$, une valeur entière s et une*

variable à domaine fini Δ, $\texttt{spread}(\boldsymbol{X}, s, \Delta)$ *est satisfaite si et seulement si*

$$\sum_{i\in[1..n]} X_i = s \quad \text{et} \quad \Delta \geq n \cdot \sum_{i\in[1..n]} |X_i - s/n|^2.$$

Observons également

$$n \cdot \sum_{i\in[1..n]} |X_i - s/n|^2 = n \cdot \sum_{i\in[1..n]} X_i^2 - s^2. \qquad (1)$$

Comme s est entier, cette quantité est entière. C'est la raison pour laquelle il est plus facile de travailler avec $n \cdot \sum_{i\in[1..n]} X_i^2 - s^2$ plutôt que $\sum_{i\in[1..n]} |X_i - s/n|^2$.

Exemple 1 $\mathbf{x} = (4, 6, 2, 5)$ *est une solution de* $\texttt{spread}([X_1, X_2, X_3, X_4], s = 17, \Delta = 40)$ *mais* $\mathbf{x} = (3, 6, 2, 6) \notin \texttt{spread}([X_1, X_2, X_3, X_4], s = 17, \Delta = 40)$ *car* $4 \cdot (3^2 + 6^2 + 2^2 + 6^2) - 17^2 = 51 > 40$.

L'algorithme de filtrage pour `spread` réalise la $\mathbb{Z}$-consistance aux bornes :

Definition 3 (Consistance aux bornes) *Une contrainte* $C(X_1, ..., X_n)$ $(n > 1)$ *est* $\mathbb{Q}$*-consistante aux bornes (resp.* $\mathbb{Z}$*-consistante aux bornes) par rapport aux domaines* $Dom(X_i)$ *si pour tout* $i \in \{1, ..., n\}$ *et chaque valeur* $v_i \in \{X_i^{\min}, X_i^{\max}\}$*, il existe les valeurs* $v_j \in I_D^{\mathbb{Q}}(X_j)$ *(resp.* $v_j \in I_D^{\mathbb{Z}}(X_j)$*) pour tout* $j \in \{1, ..., n\} - i$ *telles que* $(v_1, ..., v_n) \in C$.

Les propagateurs décrits dans [6, 11] réalisent la $\mathbb{Q}$-consistance aux bornes, ce qui signifie qu'ils considères que les variables peuvent être assignées à des nombres rationnels. Les propagateurs implémentés dans Comet réalisent la plus forte $\mathbb{Z}$-consistance aux bornes en adaptant les algorithmes de [6, 11]. En particulier pour réaliser la $\mathbb{Z}$-consistance aux bornes, les propagateurs de `spread` calculent $\underline{\Delta}^{\mathbb{Z}}$ pour filtrer $\Delta^{\min}$, $\overline{X}_i^{\mathbb{Z}}$ et $\underline{X}_i^{\mathbb{Z}}$ pour filtrer $X_i^{\max}$ and $X_i^{\min}$:

$$\underline{\Delta}^{\mathbb{Z}} = \min_{\mathbf{x}}\{n \cdot \sum_{i\in[1..n]} (\mathbf{x}[i] - s/n)^2 \text{ s.t. } \sum_{i\in[1..n]} \mathbf{x}[i] = s \quad (2)$$
$$\text{et } \forall i \in [1..n] : \mathbf{x}[i] \in I_D^{\mathbb{Z}}(X_i)\}$$

$$\overline{X}_i^{\mathbb{Z}} = \max_{\mathbf{x}}\{\mathbf{x}[i] \text{ s.t. } n \cdot \sum_{j\in[1..n]} (x[j] - s/n)^2 \leq \Delta^{\max} \quad (3)$$
$$\text{et } \sum_{j\in[1..n]} \mathbf{x}[j] = s \text{ et } \forall j : x[j] \in I_D^{\mathbb{Z}}(X_j)\}.$$

Le filtrage de Δ est implémenté dans le système Comet $O(n \cdot log(n))$ et celui de $\mathbf{X}$ en $O(n^2)$ [2, 10].

5 Un modèle simple de PC.

Nous présentons maintenant un modèle de PC qui adresse les problèmes du modèle MIP que nous avons relevés.

Le modèle de PC. Soit m le nombre d'infirmières, n le nombre de patients, et a_i l'acuité du patient i. L'ensemble des patients dans la zone k est dénoté $\mathcal{P}_k$ et $[\mathcal{P}_1, ..., \mathcal{P}_p]$ forme une partition de $\{1, ..., n\}$. Pour chaque patient i, nous introduisons une variable de décision $N_i \in [1..n]$ représentant l'infirmière qui s'occupe de lui. La charge de travail de l'infirmière j est représentée par la variable $W_j \in [0..acuity^{\max}]$. L'objectif et les contraintes sont modélisées comme suit.

- L'objectif, i.e., la minimisation de la norme L_2, est exprimé par la contrainte `spread` liant les variables de charges de travail $[W_1, ..., W_m]$, l'acuité totale, et la variable représentant la variance de l'acuité : `spread(`$[W_1, ..., W_m]$`,totalAcuity,spreadAcuity)`. Notons que `spreadAcuity` est la variable qu'il faut minimiser.
- Pour exprimer le fait que les infirmières ne peuvent avoir une acuité totale supérieure à $acuity^{\max}$, nous lions les variables N_i, W_j, et les acuités des patients avec une contrainte globale de multiknapsack [14] : `multiknapsack(`$[N_1, ..., N_n], [a_1, ..., a_n], [W_1, ..., W_m]$`)`.
- Pour modéliser le fait qu'une infirmière ne peut s'occuper de plus de $children^{\max}$ enfants, nous utilisons une contrainte globale de cardinalité [8] : `cardinality(`$1, [N_1, ..., N_n], children^{\max}$`)`.
- La contrainte qu'une infirmière ne peut travailler que dans une seule zone est modélisée avec une contrainte disant que toutes les paires de tableaux de variables ont des valeurs disjointes : `pairwiseDisjoint(`$[Z_1, ..., Z_p]$`)`, où Z_k est le tableau de variables contenant les variables N_i associées à la zone k.

Le programme Comet Le modèle Comet est donné dans le Listing 1. Les lignes 1–3 déclarent les variables de décision. La ligne 4 déclare les tableaux pour les zones qui sont remplis aux lignes 5–7. La fonction objectif est spécifiée aux lignes 8–9 et 11. Les lignes 12–14 sont les contraintes du problème. La contrainte `pairwiseDisjoint` introduit des variables ensemble représentant l'ensemble des infirmières travaillant dans une zone particulière $NS_k = \bigcup_{i\in\mathcal{P}_k} N_i$. L'ensemble NS_k est maintenu avec une contrainte globale `unionOf`. Ensuite, l'intersection vide de toutes les paires d'ensembles se fait avec une contrainte globale d'ensembles disjoints. Comet utilise une remormula-

tion de cette contrainte avec une contrainte de cardinalité comme expliqué dans [9, 1].

La recherche est implémentée dans le bloc `using` aux lignes 16–24. La recherche casse dynamiquement les symétries de valeurs induites par l'interchangeabilité des infirmières. Le patient ayant la plus grande acuité est sélectionné d'abord à la ligne 17. Ensuite la recherche tente d'assigner ce patient en commençant d'abord par les infirmières ayant la plus petite charge de travail (lignes 19–22). Le cassage des symétries est implémenté en considérant les infirmières déjà assignées et au plus une infirmière additionnelle n'ayant aucun patient (une technique similaire est utilisée pour résoudre le steel mill slab problem dans [4]). La valeur `mn` est l'indice maximum d'une infirmière déjà assignée à un patient. L'instruction `tryall` considère tous les indices d'infirmières jusqu'à `mn+1` (l'infirmière `mn+1` n'ayant pas de patient).

TABLE 1 – Résultats sur des instances à deux zones et minimisation du critère L_2 avec `spread`.

m	n	#fails	time(s)	avg workload	sd. workload
11	28	511095	170.2	86.09	2.64
11	29	1126480	302.0	80.27	1.76
10	26	104931	24.7	76.50	2.29
12	30	259147	136.5	83.42	1.93
10	28	2990450	1138.5	91.80	6.84
10	26	779969	206.9	88.40	2.29
12	29	555243	198.2	80.08	2.72
10	27	931858	343.9	90.60	5.33
10	25	1616689	434.5	82.70	7.32
8	22	4160	1.2	87.50	3.12

Résultats expérimentaux Pour la première expérience, nous avons généré 10 instances avec 2 zones, telles que les instances réelles étudiées dans [5]. Ces instances ont de l'ordre de 10–15 infirmières, 20–30 enfants, et ne peuvent être résolue avec le modèle MIP. Toutes les instances ont pu être résolues de manière optimale par notre modèle COMET en moins de 30 minutes (contrainte de temps spécifiée par l'hôpital dans [5]). La Table 1 montre les résultats expérimentaux. Tous les résultats utilisent COMET 1.1 [2] sur un Intel 2.4 GHz Core Duo avec 4GB sous MacOS 10.5.6.

6 Un modèle de PC en deux temps

Le modèle de PC basique peut résoudre des instances à deux zones mais a de grandes difficultés pour trois zones ou plus. Nous montrons comment simplifier la résolution par une approche en deux temps qui calcule d'abord le nombre d'infirmières assignées dans chaque zone et dans ensuite assigne les patients aux infirmières. Cette approche simplifie résolution en

1. supprimant le degré de flexibilité du nombre d'infirmières dans chaque zone.
2. supprimant la nécessité d'une contrainte d'ensembles disjoints puisque les ensembles d'infirmières pouvant être assignés à chaque patient sont précalculés.

Une Relaxation La première étape, c'est à dire déterminer le nombre d'infirmières dans chaque zone, est très importante. En effet si ces choix ne sont pas les bons, la solution finale peut être fortement sous optimale. Il est clair que le nombre d'infirmières assigné à chaque zone aura un grand impact sur la qualité de l'équilibre des charges de travail. Apres avoir visionné quelques solutions optimales, nous avons constaté que les temps de travail intra zone étaient trés équilibrés (quasiment les mêmes). Cela nous a inspiré la résolution d'une relaxation du problème pour découvrir une bonne répartition des infirmières entre les zones. La relaxation permet que l'acuité d'un enfant soit répartie sur plusieurs infirmières de la zone de l'enfant (relaxation continue de l'acuité). Etant donné que l'acuité peut être divisée, le problème relâché aura une solution optimale où toutes les infirmières d'une zone ont exactement la même charge de travail $\frac{A_k}{x_k}$, i.e., l'acuité totale $A_k = \sum_{i \in \mathcal{P}_k} a_i$ de la zone k divisé par le nombre d'infirmières x_k de la zone k. Cela est illustré sur la Figure 2 pour une relaxation d'un problème à deux zones. Le Lemme 1 justifie pourquoi la solution optimale de la relaxation prend une telle configuration. Intuitivement, tant que deux charges de travail peuvent être rendues plus similaires, le critère L_2 peut être diminué.

Lemme 1 *Etant donné m variables $[W_1, \ldots, W_m]$ avec une somme $s = \sum_{i=1}^{m} W_i$, le critère L_2 peut être amélioré si deux d'entre elles peuvent être rapprochées.*

Preuve 1 *Soit W_i et W_j les deux variables pouvant être rapprochées et considérons sans perte de généralité que $W_i > W_j$. Les variables après modification sont respectivement W'_i et W'_j. Si W_i et W_j sont rapprochées, cela signifie que $W'_i - W'_j < W_i - W_j$. Comme la somme est fixée nous avons $W'_i + W'_j = W_i + W_j$. Donc $W_i - W'_i = W'_j - W_j$ et il existe δ avec $\frac{(W_i - W_j)}{2} \geq \delta > 0$ tel que $W_i - W'_i = \delta = W'_j - W_j$. Nous avons $W'_i = W_i - \delta$ et $W'_j = W_j + \delta$. La somme des écarts quadratiques de départ avec la formule (1) est $\Delta = m \cdot \sum_{i=1}^{m} (W_i)^2 - s^2$. Avec W'_i et W'_j elle devient $\Delta' = m \cdot (\sum_{k \neq i,j} (W_k)^2 + (W_i - \delta)^2 + (W_j + \delta)^2) - s^2 = \Delta - 2\delta \cdot (W_i - W_j - \delta)$. Finalement comme $(W_i - W_j - \delta > 0)$, nous avons $\Delta' < \Delta$.*

Listing 1 – Modèle COMET pour l'affectation Patients-Infirmières

```
1  var<CP>{int} N[patients](cp,nurses);
2  var<CP>{int} W[nurses](cp,1..MaxAcuity);
3  var<CP>{int} spreadAcuity(cp,0..System.getMAXINT());
4  var<CP>{int}[] Z[zones];
5  int k = 1;
6  forall(i in zones,j in 1..nbPatientsInZone[i])
7      Z[i][j] = N[k++];
8  minimize<cp>
9     spreadAcuity
10 subject to {
11    cp.post(spread(W,sum(p in patients) acuity[p],spreadAcuity));
12    cp.post(multiknapsack(N,acuity,W));
13    cp.post(cardinality(minNbPatients,N,maxNbPatients));
14    cp.post(pairwiseDisjoint(Z));
15 }
16 using {
17   forall(p in patients: !N[p].bound()) by (−acuity[p],N[p].getSize()) {
18     int mn = max(0,maxBound(N));
19     tryall<cp>(n in nurses: n <= mn + 1) by (W[n].getMin())
20         cp.label(N[p],n);
21     onFailure
22         cp.diff(N[p],n);
23   }
24 }
```

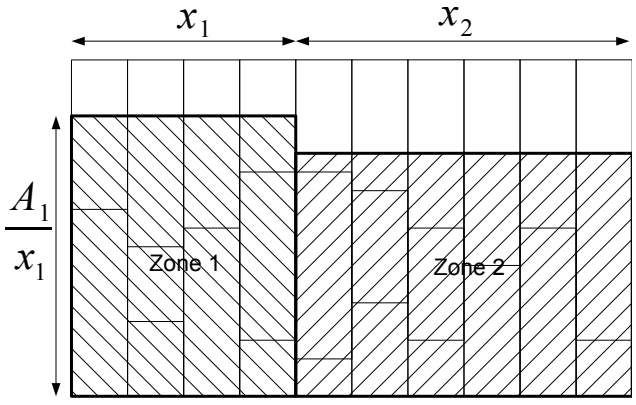

FIGURE 2 – Illustration d'une solution de la relaxation résolue pour trouver le nombre d'infirmières de chaque zone.

Le Lemme 1 prouve que la solution optimale du problème relâché comporte la même charge de travail pour toutes les infirmières d'une même zone. Par conséquent, la formulation mathématique du problème relâché est la suivante :

$$\min \quad \sum_{k=1}^{p} x_k \cdot \left(\frac{A_k}{x_k} - \sum_{j=1}^{p} \frac{A_j}{m} \right)^2 \tag{4}$$

$$s.t. \quad \sum_{k=1}^{p} x_k = m \tag{5}$$

$$x_k \in \mathbb{Z}_0^+ \tag{6}$$

La charge de travail des infirmières de la zone k est $\frac{A_k}{x_k}$ et la charge de travail moyenne est $\sum_{j=1}^{p} \frac{A_j}{m}$. Donc la contribution au critère L_2 pour les x_k infirmières de la zone k est $x_k \cdot \left(\frac{A_k}{x_k} - \sum_{j=1}^{p} \frac{A_j}{m} \right)^2$.

Résolution de la relaxation Dans notre modèle de PC, nous approximons la relaxation en $O(p \cdot log(p))$. D'abord nous solutionnons la relaxation continue du problème, i.e., nous laissons tomber la contrainte d'intégrité (6). La solution de ce problème d'optimisation continue est $x_k = m \cdot \frac{A_k}{\sum_{j=1}^{p} A_j}$, ce qui corres-

pond à attribuer une même charge de travail moyenne $\sum_{j=1}^{p} \frac{A_j}{m}$ pour toutes les infirmières. Ensuite cette solution continue $x_k = m \cdot \frac{A_k}{\sum_{j=1}^{p} A_j}$ peut être transformée de manière vorace en une solution entière en suivant les étapes suivantes :

- En développant la formule de l'objectif (4), il apparait qu'il est équivalent de minimiser $\sum_{k=1}^{p} \frac{(A_k)^2}{x_k}$.
- La transformation en une solution entière commence par arrondir vers le haut le nombre d'infirmières dans chaque zone $x_k = \lceil m \cdot \frac{A_k}{\sum_{j=1}^{p} A_j} \rceil$. La conséquence étant que la contrainte (5) peut être violée et la fonction objectif peut avoir diminué.
- Ensuite, les $x_k > 1$ sont considérés pour être diminués d'une unité jusqu'à rétablir la contrainte (5). L'indice k du x_k suivant qui est diminué est $\text{argmin}_k\{\frac{A_k^2}{x_k - 1} - \frac{A_k^2}{x_k}\}$, i.e., la variable qui augmente le moins son terme correspondant dans la fonction objectif $\sum_{k=1}^{p} \frac{A_k^2}{x_k}$.

Nos résultats expérimentaux montre que l'approximation est optimale sur toutes les instances résolues par le premier modèle de PC.

Borne inférieure sur la variance Le précalcul du nombre d'infirmières assignées à chaque zone peut aussi servir à calculer une borne inférieure sur le critère L_2. A l'intérieur d'une zone, la charge de travail moyenne est $\mu_k = A_k/x_k$. Puisque l'acuité des patients est entière, nous pouvons obtenir une meilleure borne inférieure sur l'objectif (4) en considérant que la charge de travail d'une infirmière de la zone k est soit $\lfloor \mu_k \rfloor$ soit $\lceil \mu_k \rceil$. Cela est illustré sur la Figure 3. Puisque la charge totale de travail pour la zone k doit rester A_k, la répartition des charges de travail entre $\lfloor \mu_k \rfloor$ et $\lceil \mu_k \rceil$ est donnée respectivement par $\alpha_k = A_k + x_k \cdot (1 - \lceil \mu_k \rceil)$ and $\beta_k = x_k - \alpha_k$. La borne inférieure sur la variable de variance $\underline{\Delta}^{\mathbb{Z}}$ calculée à l'aide de la formule (1) est donc

$$m \cdot \sum_{k=1}^{p} (\alpha_k \cdot \lceil \mu_k \rceil^2 + \beta_k \cdot \lfloor \mu_k \rfloor^2) - (\sum_{k=1}^{p} A_k)^2. \quad (7)$$

Le modèle COMET Le modèle COMET en deux temps est donné au Listing 2 et il considère que les x_k ont déjà été calculés. Le modèle ne crée pas les N variables à la ligne 2 : ces variables seront créées au même moment que le tableau de la zone puisque leur domaine est restreint à un sous ensemble d'infirmières. Les lignes 6–12 créent les tableaux de variable des zones, la ligne 10 construisant le tableau de variable de la zone i. Notons que le domaine de

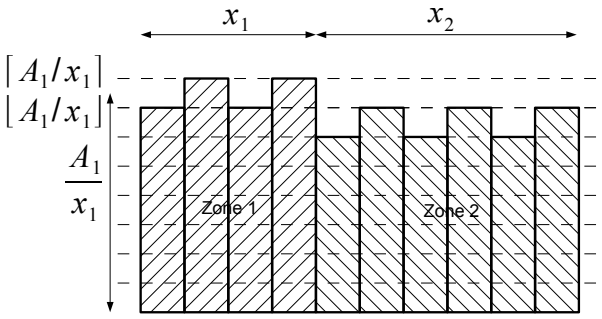

FIGURE 3 – Illustration de la borne inférieure sur L_2 en utilisant le précalcul du nombre d'infirmières dans chaque zone.

ces variables sont définis aux lignes 9 et 11 en utilisant le nombre d'infirmières assignées à chaque zone. Les lignes 13–15 assignent les variables de zone aux variables infirmière (le contraire du premier modèle car les variables de zone ont maintenant des domaines restreints). Les contraintes sont similaires mais il n'y a plus besoin de la contrainte `pairwiseDisjoint`. La recherche est implémentée aux lignes 23–34. Elle est un peu plus compliquée puisque les patients sont affectés une zone à la fois. Le cassage de symétries dynamiques est le même mais adapté à cette affectation par zone.

La Table 2 donne les résultats obtenus pour les même instances à deux zones que pour la Table 1 utilisant le précalcul du nombre d'infirmières allouées à chaque zone. La dernière collonne est la borne inférieure obtenue avec l'équation (7). Une première observation est que les temps de calculs sont fortement réduits. Ils n'excedent pas 10 secondes avec le nouveau modèle alors qu'ils pouvaient atteindre 1000 secondes pour les instances les plus difficiles avec le premier modèle. Le nombre d'infirmières trouvé dans la première étape est correct puisque les écarts types sont les mêmes que les valeurs optimales trouvées avec le premier modèle dans la Table 1. Il est également intéressant d'observer que la borne inférieure est raisonablement proche des valeurs optimales ce qui valide également l'approche.

Comme l'instance à deux zones peut maintenant être résolue aisément, nous avons essayé de résoudre des instances à trois zones. Les résultats sont présentés à la Table 3. Seules 6 instances sur 10 peuvent être résolues à l'optimum en moins de 30 minutes avec cette approche en deux temps.

Listing 2 – Modèle en deux temps pour l'assignation Patients-Infirmières.

```
1  Solver<CP> cp();
2  var<CP>{int} N[patients];
3  var<CP>{int} W[nurses](cp,1..MaxAcuity);
4  var<CP>{int} spreadAcuity(cp,0..System.getMAXINT());
5  var<CP>{int}[] Z[zones];
6  range nursesOfZone[zones];
7  int j=1;
8  forall(i in zones) {
9      nursesOfZone[i] = j..j+x[i]−1;
10     Z[i] = new var<CP>{int}[1..nbPatientsInZone[i]](cp,nursesOfZone[i]);
11     j += x[i];
12 }
13 int k = 1;
14 forall(i in zones,j in 1..x[i])
15         N[k++] = Z[i][j];
16 minimize<cp>
17     spreadAcuity
18 subject to {
19     cp.post(spread(W,sum(p in patients) acuity[p],spreadAcuity));
20     cp.post(multiknapsack(N,acuity,W));
21     cp.post(cardinality(minNbPatients,N,maxNbPatients));
22 }
23 using {
24   forall(i in zones){
25     forall(p in Z[i].rng(): !Z[i][p].bound()) by(−acuityByZone[i][p],Z[i][p].getSize()){
26         int shift = i==1? 0 : nursesOfZone[i−1].getUp();
27         int mn = max(0,maxBound(Z[i]))+shift;
28         tryall<cp>(n in nursesOfZone[i]: n <= mn + 1) by (W[n].getMin())
29             cp.label(Z[i][p],n);
30         onFailure
31             cp.diff(Z[i][p],n);
32     }
33   }
34 }
```

TABLE 2 – Résultats sur des instances à deux zones avec précalcul du nombre d'infirmières dans chaque zone.

m	n	#fails	time(s)	avg workload	sd. workload	lb. sd.
11	28	25385	4.5	86.09	2.64	2.23
11	29	4916	1.4	80.27	1.76	0.62
10	26	458	0.1	76.50	2.29	2.29
12	30	17558	6.7	83.42	1.93	1.19
10	28	29865	4.8	91.80	6.84	6.81
10	26	3705	1.0	88.40	2.29	1.43
12	29	6115	1.2	80.08	2.72	0.64
10	27	1109	0.4	90.60	5.33	5.22
10	25	3299	0.6	82.70	7.32	6.71
8	22	127	0.0	87.50	3.12	3.04

TABLE 3 – Résultats sur des instances à trois zones avec précalcul du nombre d'infirmières dans chaque zone.

sol	m	n	#fails	time(s)	avg wl.	sd. wl.	lb. sd.
1	15	42	19488	5.3	84.20	3.04	2.93
1	18	43	3619310	919.2	79.78	5.84	5.49
0	17	43	9023072	1800.0	81.41	4.75	3.45
1	17	42	483032	106.9	83.82	5.65	5.59
0	18	43	7124370	1800.0	81.00	7.11	4.94
1	14	38	590971	145.2	85.36	3.08	2.16
0	19	48	3786580	1800.0	87.42	3.18	2.30
1	16	44	3888210	839.8	84.88	6.70	6.39
0	19	49	5697272	1800.0	86.00	2.70	1.95
1	17	41	61250	17.3	82.18	3.40	3.07

7 Un modèle de PC en deux temps avec décomposition

L'approche précédente peut résoudre facilement les problèmes à deux zones mais présente des difficultés pour les instances à trois zones ou plus. Il paraît naturel de décomposer le problème en zone et d'équilibrer les charges de travail des infirmières dans chaque zone indépendamment plutôt que que d'équilibrer les charges de toutes les infirmières globalement. De manière intéressante, cette décomposition préserve l'optimalité, i.e., elle obtient la même solution pour le critère L_2 que l'approche en deux temps de la Section 6 pour un précalcul donné du nombre d'infirmières attribué à chaque zone. Autrement dit, étant donne un précalcul du nombre d'infirmières dans chaque zone, il est équivalent de minimiser L_2 entre toutes les infirmières en une fois ou de minimiser L_2 séparément dans chaque zone. Nous prouvons ce résultat formellement.

Lemme 2 *Minimiser* $n \cdot \sum_{i=1}^{x_k}(y_i - A_k/x_k)^2$ *tel que* $\sum_{i=1}^{x_k} y_i = A_k$ *est équivalent la minimisation de* $n \cdot \sum_{i=1}^{x_k}(y_i - (A_k/x_k + c))^2$ *tel que* $\sum_{i=1}^{x_k} y_i = A_k$.

TABLE 4 – Résultats sur des instances à trois zones avec précalcul du nombre d'infirmières dans chaque zone et décomposition par zone.

m	n	#fails	time(s)	avg workload	sd. workload	lb. sd.
15	42	203	0.1	84.20	3.04	2.93
18	43	608	0.1	79.78	5.84	5.49
17	43	8134	1.1	81.41	4.46	3.45
17	42	345	0.1	83.82	5.65	5.59
18	43	24994	3.2	81.00	5.77	4.94
14	38	151	0.0	85.36	3.08	2.16
19	48	3695	0.8	87.42	3.07	2.30
16	44	384	0.1	84.88	6.70	6.39
19	49	2056	0.4	86.00	2.49	1.95
17	41	776	0.2	82.18	3.40	3.07

Preuve 2 *Le premier objectif peut être reformulé au depart de la formule (1) comme* $x_k \cdot \sum_{i=1}^{x_k} y_i^2 - A_k^2$. *Le second peut être reformulé après quelques manipulations algébriques comme* $c^2 \cdot x_k^2 + x_k \cdot \sum_{i=1}^{x_k} y_i^2 - A_k^2$. *Comme ils ne diffèrent que par une constante, leur minimisation produit le même ensemble de solutions optimales.*

Corollaire 1 *Il est équivalent de minimiser* L_2 *entre toutes les infirmières en une fois ou de minimiser* L_2 *séparément dans chaque zone.*

Preuve 3 *Cela est une conséquence directe du Lemme 2. Si la minimisation de* L_2 *est faite globalement pour toutes les infirmières, le critère des moindres carrés* L_2 *est calculé par rapport à la charge moyenne de toutes les infirmière c'est à dire par rapport à* $\sum_{k=1}^{p} A_k/m$. *Cela correspond à un choix pour* c *dans le Lemme 2 égal à la différence entre la charge de travail moyenne dans la zone* k *et la charge globale moyenne :* $c = \sum_{k=1}^{p} A_k/m - A_k/x_k$.

Nous avons résolu à nouveau les instances à trois zones avec la méthode de la décomposition. Les résultats sont présentés sur la Table 4. Nous pouvons observer que comme attendu, les valeurs de la fonction objectif sont les mêmes pour toutes les instances qui ont été résolues de manière optimale dans la Table 3. Pour les autres, l'algorithme produit des solutions strictement meilleures. Le temps est également significativement moindre. La Figure 4 montre une visualisation COMET d'une solution à 15 zones avec 81 infirmières et 209 patients. Cette instances a pu être résolue en seulement 7 secondes et 10.938 échecs.

8 Conclusion

Cet article traite de l'affectation journalière des patients nouveaux nés aux infirmières dans un hôpital.

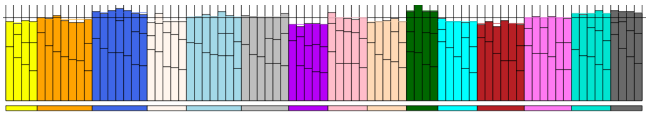

FIGURE 4 – Solution d'une instance à 15 zones.

L'objectif du problème est d'équilibrer la charge de travail des infirmières tout en satisfaisant certaines contraintes. Les travaux antécédents ont suggéré un modèle MIP pour résoudre ce problème qui avait deux limitations. Il ne permettait pas de résoudre des instances de grandes tailles et la fonction objectif ne modélise pas correctement le fait que l'on souhaite optimiser l'équilibre des charges de travail. Cet article présente une modèle de PC qui réalise correctement l'objectif d'équilibre et qui peut résoudre facilement des instances à deux zones. Pour permettre de résoudre des instances de grande taille, nous avons utilisé une décomposition du problème en deux temps : la première étape assigne les infirmière aux zones suivie de l'affectation des infirmières aux patients. La première étape est obtenue en solutionnant une relaxation du problème facile à résoudre. La seconde étape est résolue à l'aide d'une simplification du modèle de PC direct. Cette approche en deux temps améliore significativement les résultats sur les instances à deux zones et permet de résoudre des instances à trois zones. Nous montrons ensuite que les problèmes de chaque zone peuvent être résolus indépendamment sans perte de qualité. Le modèle de PC résultant résoud les problèmes à trois zones quasi instantanément et est très robuste quand le nombre de zones augmente. Par exemple, nous pouvons résoudre une problème à 15 zones, 81 infirmières et 209 patients en 7 seconds.

Il y a un certain nombre de problèmes intéressants à approfondir. Il serait intéressant d'étudier la qualité de l'approximation effectuée dans la première étape. Nos résultats expérimentaux indiquent que l'approximation est optimale sur toutes les instances testées néanmoins il est serait souhaitable d'avoir des garantie sur la qualité. Aussi nous pourrions envisager de résoudre la première étape de manière exacte. Nous devons envisager cet aspect algorithmique également. De plus, il serait intéressant d'étudier des problèmes où les infirmières ont diverses qualifications limittant leurs possibles affecations aux zones.

Références

[1] Christian Bessiere, Emmanuel Hebrard, Brahim Hnich, and Toby Walsh. Disjoint, partition and intersection constraints for set and multiset variables. In *Principles and Practice of Constraint Programming (CP 2004)*, pages 138–152, 2004.

[2] DYNADEC. Comet 1.1 release. *www.dynadec.com*, 2009.

[3] Stephen Gorard. Revisiting a 90-year-old debate : The advantages of the mean deviation. *British Journal of Educational Studies*, pages 417–439, 2005.

[4] P. Van Hentenryck and L. Michel. The steel mill slab design problem revisited. *CP'AI'OR-08, Paris, France*, 5015 :377–381, May 2008.

[5] C Mullinax and M Lawley. Assigning patients to nurses in neonatal intensive care. *Journal of the Operational Research Society*, 53 :25–35, 2002.

[6] G. Pesant and J.C. Régin. Spread : A balancing constraint based on statistics. *Lecture Notes in Computer Science*, 3709 :460–474, 2005.

[7] Gilles Pesant. Constraint-based rostering. *The 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008)*, 2008.

[8] J-C. Régin. Generalized arc consistency for global cardinality constraint. *AAAI-96*, pages 209–215, 1996.

[9] J.C. Régin. Habilitation à diriger des recherches (hdr) : Modelization and global constraints in constraint programming. *Université Nice*, 2004.

[10] P. Schaus. Balancing and bin-packing constraints in constraint programming. *PhD thesis, Université catholique de Louvain, INGI,*, 2009.

[11] P. Schaus, Y. Deville, P. Dupont, and J.C. Régin. Simplification and extension of spread. *3th Workshop on Constraint Propagation And Implementation*, 2006.

[12] P. Schaus, Y. Deville, P. Dupont, and J.C. Régin. The deviation constraint. *Proceedings of CP-AI-OR*, 4510 :269–284, 2007.

[13] Pierre Schaus, Yves Deville, and Pierre Dupont. Bound-consistent deviation constraint. *13th International Conference on Principles and Practice of Constraint Programming (CP 2007)*, 4741, September 2007.

[14] Paul Shaw. A constraint for bin packing. In *Principles and Practice of Constraint Programming (CP 2004)*, pages 648–662, 2004.

[15] Helmut Simonis. Models for global constraint applications. *Constraints*, 12 :63–92, March 2007.