

Soutenance de Projet

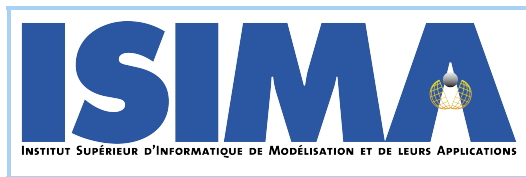
Problème de Gestion de Personnel en Soins à Domicile

LE PAPE Sandy

13 mars 2019

Introduction

Acteurs du projet



Introduction

Contexte et enjeux

Le problème : on dispose

- d'un personnel soignant, composé de plusieurs infirmières,
- d'un ensemble de patients à soigner à leur domicile respectif,
- d'un ensemble de contraintes de travail / horaires.

Objectif : trouver un modèle affectant automatiquement les infirmières aux patients, en satisfaisant les contraintes des 2 partis.

Limitations et objectif

Problème : implémenter un problème de type *Workforce Scheduling* appliqué à ce contexte.

Limitations de la méthode actuelle :

- aucune méthode parfaite, le problème étant NP-Complet ;
- beaucoup d'heuristiques implémentées à ce jour.

Etapes de travail

- travailler le TSP en y introduisant des contraintes de temps,
- l'adapter pour un problème de type VRP.

Sommaire

- 1 Modélisations principales
 - Programmation Linéaire
 - Programmation Par Contraintes
- 2 Le TSP : le Problème du Voyageur de Commerce
 - Présentation du problème et modélisations possibles
 - Ajouts de temps caractéristiques et des prédécesseurs
 - Mise en place de stratégies
- 3 Le VRP : les Problèmes de Tournées de Véhicules
 - Présentation du problème et modélisation linéaire
 - Construction de la solution Programme Par Contraintes
- 4 Conclusion

Modélisations principales

Programmation Linéaire



Principe

Optimiser un problème de nature linéaire sur un polyèdre convexe.

Composantes du problème :

- un ensemble de variables,
- un ensemble de contraintes linéaires,
- une fonction objectif linéaire.

Outils

Outils + logo	GUSEK 	CPLEX OPTIMIZER 
Liberté d'utilisation	libre de droits	privé
Langage	basé sur AMPL	basé sur AMPL + OPL
Solveur	GPLK	solveur IBM

Programmation Par Contraintes

Principe

Optimiser un problème de nature combinatoire sur un espace de recherche.

Composantes :

- un ensemble de variables et leur domaine de valeurs respectif,
- un ensemble de contraintes,
- une fonction objectif,
- un ensemble de stratégies de recherche.

Outils et structure du code



- Librairie libre de droits,
- Française,
- Ecrite en Java,
- Dispose de son propre solveur,
- Dédiée à la PPC.

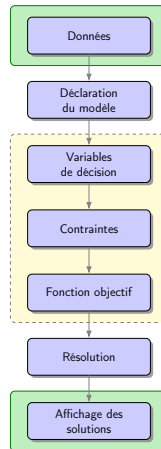


Figure – Pipeline Choco

Le TSP : le Problème du Voyageur de Commerce

Présentation du problème et modélisations possibles

Modélisation par un graphe

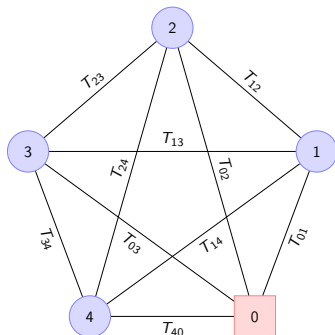


Figure – Modélisation du TSP

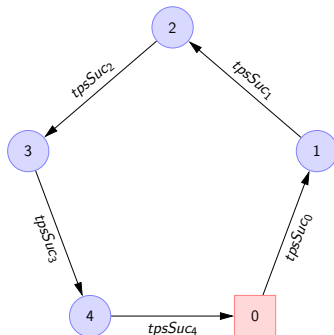


Figure – Tournée possible

Programme Linéaire final

$$\min d = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} x_{ij} \cdot T_{ij} + x_{x_N 1} \cdot T_{ij}$$

$$\text{s.c.} \quad \sum_{i=0}^{n-1} x_{ij} = 1 \quad \forall i \in [0, n-1] \quad (1)$$

$$\sum_{i=0}^{n-1} x_{ij} = 1 \quad \forall j \in [0, n-1] \quad (2)$$

$$x_{ij} = 0 \quad \forall i \in [0, n-1] \quad (3)$$

$$x_{v_1 v_2} + x_{v_2 v_3} + \dots + x_{v_t v_1} \leq t - 1 \quad \forall \{v_1, \dots, v_t\} \subset V \quad (4)$$

Principe

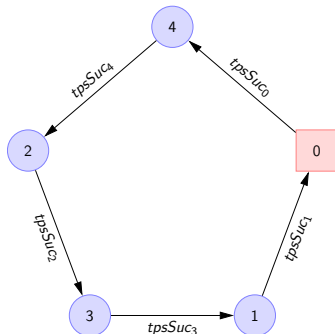


Figure – Tournée possible

	0	1	2	3	4
liste r	1	0	4	2	3

Figure – Vecteur du modèle

Problème : Choco ne permet pas l'implémentation d'expressions du type $r_i r_{i+1}$.

Modélisation adaptée à Choco

→ Introduction d'un vecteur successeur s .

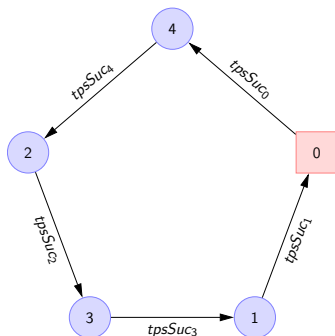


Figure – Tournée possible

	0	1	2	3	4
liste r	1	0	4	2	3

	0	1	2	3	4
liste s	4	0	3	1	2

Figure – Vecteurs du modèle

Problème Par Contraintes pour Choco

$$\min d = \sum_{i=0}^{n-1} tpsSuc_i$$

s.c.

$$\text{Contraintes domaines de déf} \quad + \quad (5)$$

$$r_i \neq r_j \quad \forall i, j \in [0, n-1] \quad (6)$$

$$s_i \neq s_j \quad \forall i, j \in [0, n-1] \quad (7)$$

$$r_{i+1} = s_{r_i} \quad \forall i \in [0, n-2] \quad (8)$$

$$tpsSuc_i = T_{is_i} \quad \forall i \in [0, n-1] \quad (9)$$

$$s_i \neq i \quad \forall i \in [0, n-1] \quad (10)$$

Ajouts de temps caractéristiques et des prédécesseurs

Introduction de notions de durées / dates

But n° 1 : indiquer pour chaque patient ses disponibilités.

→ Associer à chaque sommet une *time windows* $[TW_{inf}, TW_{sup}]$.

But n° 2 : indiquer pour chaque patient la durée de soins requise.

→ Associer à chaque sommet une durée de soins .

⇒ Associer à chaque infirmière un ensemble de dates pour la gestion des nouvelles notions introduites.

Contraintes de temps

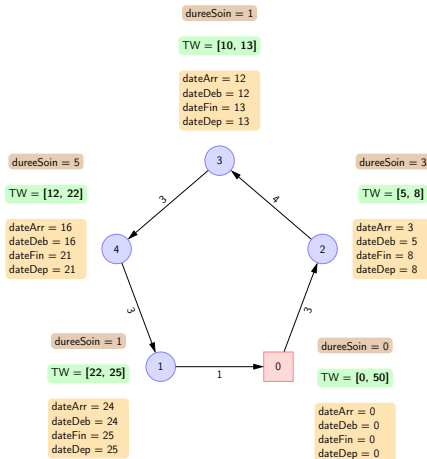
$$dateArr_{s_i} = dateFin_i + tpsSuc_i \quad \forall i \in [|0, n-1|] - r_{n-1} \quad (11)$$

$$dateArr_{s_i} = 0 \quad \text{si } i = r_{n-1} \quad (12)$$

$$dateDeb_{s_i} = \max(dateArr_{s_i}, TWInf_{s_i}) \quad \forall i \in [|0, n-1|] \quad (13)$$

$$dateFin_i = dateDeb_i + dureeSoin_i \quad \forall i \in [|0, n-1|] \quad (14)$$

Résultats obtenus



$$\Rightarrow d_{total} = 26.$$

Figure – Modélisation du TSP avec durées + dates

Prédécesseurs

Finalité : aider le solveur dans ses choix de branchements en lui donnant des informations supplémentaires.

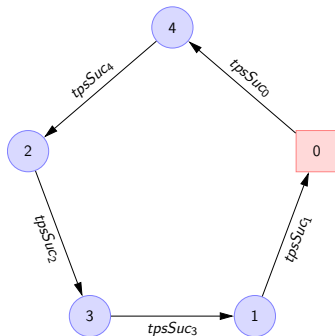


Figure – Tournée possible

	0	1	2	3	4
liste p	1	3	4	2	0

	0	1	2	3	4
liste r	1	0	4	2	3

	0	1	2	3	4
liste s	4	0	3	1	2

Figure – Vecteurs du modèle

Résultats obtenus

Cas	AVEC/SANS PREC	NODES	BACKTRACKS	FAILS
1	sans	32	59	27
	avec	8	13	5
2	sans	21	39	18
	avec	12	19	7
3	sans	7570	15135	7565
	avec	60	111	51

Conclusion : économie de nombreux noeuds, surtout pour les graphes ≥ 10 sommets.

Mise en place de stratégies

Stratégies en PPC

Principe : guider le solveur dans ses choix de branchements en lui indiquant ceux à affectuer en priorité.

Composantes :

- un sélecteur de variable,
- un sélecteur de valeur,
- la variable de branchement.

Stratégies proposées

Sélecteurs de variables Choco :

- FirstFail,
- MaxRegret,
- DomWOverReg.

S1 : brancher sur tpsSuc ;

S2 : brancher sur r ;

S3 : brancher sur des discriminateurs.

Résultats obtenus

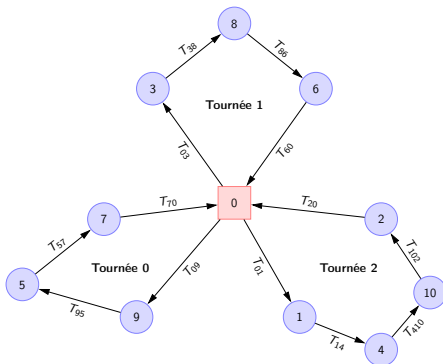
Cas	STRAT TESTÉE	NODES	BACKTRACKS	FAILS
1	aucune	8	13	15
	avec tpsSuc	5	9	4
	avec r	7	11	14
2	aucune	12	19	7
	avec tpsSuc	4	7	3
	avec r	5	7	2
3	aucune	60	111	51
	avec tpsSuc	18	35	17
	avec r	122	217	95

Conclusion : meilleurs résultats avec branchement sur tpsSuc.

Le VRP : les Problèmes de Tournées de Véhicules

Présentation du problème et modélisation linéaire

Modélisation par un graphe



Différence majeure avec le TSP :
 plusieurs infirmières
 \Rightarrow 1^{ers} pas vers les problèmes de type *Workforce Scheduling*.

Figure – Modélisation du VRP

Programme Linéaire final

$$\min d = \sum_{i=0}^{np-1} \sum_{j=0}^{np-1} \sum_{k=0}^{nv-1} x_{ij}^k \cdot T_{ij}$$

$$\text{s.c.} \quad \sum_{j=1}^{np-1} x_{1j}^k \leq 1 \quad \forall k \in [0, nv - 1] \quad (15)$$

$$\sum_{j=0}^{np-1} \sum_{k=0}^{nv-1} x_{ij}^k = 1 \quad \forall i \in [1, np - 1] \quad (16)$$

$$\sum_{i=0}^{np-1} \sum_{k=0}^{nv-1} x_{ij}^k = 1 \quad \forall j \in [1, np - 1] \quad (17)$$

$$+ 2 \text{ autres} \quad (18)$$

Construction de la solution Programme Par Contraintes

Principe

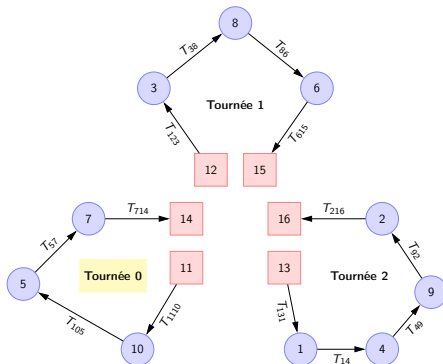


Figure – Modélisation du VRP

Astuces :

- hôpital = dépôt d'entrée + dépôt de sortie ;
- les dupliquer autant de fois qu'il y a d'infirmières.

Construction des vecteurs

→ Introduction des vecteurs a , s , pos et de l'ensemble b^v .

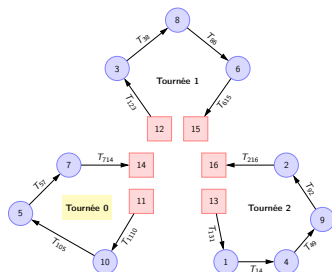


Figure – Modélisation

liste a

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	2	2	1	2	0	1	0	1	2	0	0	1	2	0	1	2

liste s

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
					7		14			5	10			0		

liste pos

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
					2		3			1	0			4		

Figure – Vecteurs du modèle

⇒ Construction de partitions de clients en fonction des tournées.

Partition tournée 0 : $b^0 = \{11, 10, 5, 7, 14\}$

Programme Par Contraintes final

$$\min d = \sum_{i=0}^{|V|-1} T_{i,s_i}$$

s.c.

$$\text{Contraintes domaines de déf} \quad + \quad (19)$$

$$s_i \neq s_j \quad \forall i, j \in \{Np \cup H^d \cup H^f\}^2 \quad (20)$$

$$a_i = a_{s_i} \quad \forall i \in Np \cup H^d \quad (21)$$

$$s_i \neq i \quad \forall i \in Np \cup H^d \quad (22)$$

$$pos_{s_i} = pos_i + 1 \quad \forall i \in Np \cup H^d \quad (23)$$

$$b^v = \{u \in N \mid a_u = v\} \quad \forall v \in Nv \quad (24)$$

Conclusion

Problème de Gestion de Personnel en Soins à Domicile

✓ Bilan des apports :

- Maîtrise de modèles importants de l'I.A.,
- Compréhension de stratégies algorithmiques liées à la PPC,
- Rapidité à corriger les erreurs de compilation.

✓ Perspectives :

- Implémenter d'autres contraintes de plus en plus réalistes,
- Faire des phases de tests plus solides avec CPLEX,
- Appliquer l'algorithme à petite échelle dans le monde réel.

Merci de votre attention
