

Création d'un horaire d'examens

Projet d'informatique fondamentale - INFO-F-302

Titouan CHRISTOPHE et Florentin HENNECKER

ULB BA3 info 2014-2015

15 mai 2015

Table des matières

1	Énoncé	1
1.1	Définitions	1
1.2	Objectif	1
2	Questions	2
2.1	Définir la notion de correction pour une fonction μ	2
2.2	Formaliser ces contraintes en langage mathématique	2
2.3	Construire une formule Φ en FNC	2
2.4	Implémenter et tester sur les exemples, et proposer éventuellement d'autres exemples . . .	2
2.4.1	Format d'entrée	3
2.4.2	Script run.sh	3
2.5	Ajouter une contrainte de durée pour chaque examen	3
2.6	Les examens ne peuvent pas avoir lieu tout le temps, par exemple la nuit	3
2.7	Maximum k changements de salle pour tous les étudiants	4
2.8	Une unité de temps tampon	4
2.9	Proposer d'autres contraintes naturelles et donner leur modélisation.	4
2.9.1	n unités de temps entre tout examen pour un étudiant	4
3	Conclusion	4

1 Énoncé

1.1 Définitions

Voici :

- $X = \{x_1, \dots, x_{n_X}\}$, l'ensemble des examens à organiser ;
- $S = \{s_1, \dots, s_{n_S}\}$, l'ensemble des salles disponibles ;
- $c : S \mapsto \mathbb{N}$, l'application qui à toute salle associe sa capacité
- $E = \{e_1, \dots, e_{n_E}\}$, l'ensemble des étudiants ;
- $a : E \mapsto 2^X$, l'application qui à tout étudiant associe les examens qu'il doit passer ;
- $P = \{p_1, \dots, p_{n_P}\}$, l'ensemble des professeurs ;
- $b : P \mapsto 2^X$, l'application qui à tout professeur associe les examens qu'il doit organiser ;
- $T = \{t_1, \dots, t_{n_T}\}$, l'ensemble des tranches horaires ;
- $\tau_{x,t}$ est vrai ssi l'examen x commence à la période t avec $x \in X, t \in T$
- $\sigma_{x,s}$ est vrai ssi l'examen x se déroule dans la salle s avec $x \in X, s \in S$
- $\omega_{s,x}$ est vrai ssi la salle s peut accueillir l'examen x (donc si la capacité de la salle est assez élevée), avec $s \in S, x \in X$

1.2 Objectif

On cherche à définir une application $\mu : X \mapsto S \times t_1, \dots, t_{n_T} \forall x \in X$, qui associe à chaque examen la salle et l'horaire auquel il a lieu.

Si $\mu(x) = (s, t)$, alors l'examen x a lieu dans la salle s à l'heure t .

2 Questions

2.1 Définir la notion de correction pour une fonction μ

La fonction μ doit respecter les contraintes suivantes :

1. Chaque examen doit avoir lieu une et une seule fois
2. Chaque examen doit avoir lieu dans une et une seule pièce
3. Un étudiant ne peut avoir qu'un seul examen par tranche horaire ;
4. Un professeur ne peut avoir qu'un seul examen par tranche horaire ;
5. Une salle ne peut accueillir qu'un seul examen à la fois ;
6. Le nombre d'étudiants présentant un examen ne peut excéder la capacité de la salle qui l'accueille ;

2.2 Formaliser ces contraintes en langage mathématique

1. $\forall x_1, x_2 \in X, \mu(x_1) \mapsto (s_1, t_1), \mu(x_2) \mapsto (s_2, t_2) : t_1 \neq t_2 \rightarrow x_1 \neq x_2$
2. $\forall x_1, x_2 \in X, \mu(x_1) \mapsto (s_1, t_1), \mu(x_2) \mapsto (s_2, t_2) : s_1 \neq s_2 \rightarrow x_1 \neq x_2$
3. $\forall x_1, x_2 \in a(e_i), 1 \leq i \leq n_E, \mu(x_1) \mapsto (s_1, t_1), \mu(x_2) \mapsto (s_2, t_2) : x_1 \neq x_2 \rightarrow t_1 \neq t_2$
4. $\forall x_1, x_2 \in b(p_i), 1 \leq i \leq n_P, \mu(x_1) \mapsto (s_1, t_1), \mu(x_2) \mapsto (s_2, t_2) : x_1 \neq x_2 \rightarrow t_1 \neq t_2$
5. $\forall x_1, x_2 \in X, \mu(x_1) \mapsto (s_1, t_1), \mu(x_2) \mapsto (s_2, t_2) : (x_1 \neq x_2) \wedge (t_1 = t_2) \rightarrow s_1 \neq s_2$
6. $\forall x \in X, 1 \leq i \leq n_E, \mu(x) \mapsto (s, t) \rightarrow \#\{e_i \mid x \in a(e_i)\} \leq c(s)$ (où $\#\eta$ désigne le cardinal de l'ensemble η)

2.3 Construire une formule Φ en FNC

On commence par exprimer chacune des contraintes énoncées en FNC :

1. $C_1 = \bigwedge_{x \in X} \bigwedge_{t_i, t_j \in T, t_i \neq t_j} \neg \tau_{x, t_i} \vee \neg \tau_{x, t_j}$
2. $C_2 = \bigwedge_{x \in X} \bigwedge_{s_i, s_j \in S, s_i \neq s_j} \neg \sigma_{x, s_i} \vee \neg \sigma_{x, s_j}$
3. $C_3 = \bigwedge_{t \in T} \bigwedge_{e \in E} \bigwedge_{x_i, x_j \in a(e), x_i \neq x_j} \neg \tau_{x_i, t} \vee \neg \tau_{x_j, t}$
4. $C_4 = \bigwedge_{t \in T} \bigwedge_{p \in P} \bigwedge_{x_i, x_j \in b(p), x_i \neq x_j} \neg \tau_{x_i, t} \vee \neg \tau_{x_j, t}$
5. $C_5 = \bigwedge_{t \in T} \bigwedge_{s \in S} \bigwedge_{x_i, x_j \in X, x_i \neq x_j} \neg \tau_{x_i, t} \vee \neg \tau_{x_j, t} \vee \neg \sigma_{x_i, t} \vee \neg \sigma_{x_j, t}$
6. $C_6 = \bigwedge_{x \in X} \bigwedge_{s \in S} \neg \sigma_{x, s} \vee \omega_{s, x}$

La FNC complète est donc la suivante :

$$\bigwedge_{i \in \{1, 2, 3, 4, 5, 6\}} C_i$$

2.4 Implémenter et tester sur les exemples, et proposer éventuellement d'autres exemples

Les contraintes sont exprimées par des variables Minisat, rangées dans une matrice $\mu[x][s][t]$. La variable est vraie si l'examen x commence dans la salle s au moment t .

2.4.1 Format d'entrée

Nous utilisons le format d'entrée donné dans les consignes avec les adjonctions suivantes, dans l'ordre :

1. On rajoute les durées des examens, dans l'ordre, séparées par un ' ; ' ;
2. On rajoute les périodes interdites, au format `debut,fin` ;

Par ailleurs, le programme accepte les options suivantes :

- `-n N` : lire N problèmes consécutivement depuis l'entrée standard
- `-k K` : Autoriser au plus K changements de salle
- `-s` : Minimum 1h entre 2 exams pas dans la même salle pour un étudiant
- `-f` : Lire les périodes qui ne devraient pas avoir d'exams dans le fichier d'entrée
- `-t` : Lire la durée de chaque exam dans le fichier d'entrée

Le programme lit le problème sur l'entrée standard (stdin), et écrit la solution sur la sortie standard (stdout). Il affiche en outre l'horaire sous forme de tableau sur la sortie d'erreur (stderr).

2.4.2 Script run.sh

Nous avons en outre rédigé un script `run.sh` permettant d'exécuter nos scénarios de test, qui contiennent des commentaires. Il s'utilise sans paramètres pour exécuter l'exemple par défaut, ou alors avec la syntaxe `run.sh fichier [options]`. Les options sont celles décrites ci-dessus.

2.5 Ajouter une contrainte de durée pour chaque examen

On ajoute au problème l'application $d : X \rightarrow \mathbb{N}$, qui à chaque examen associe sa durée. Pour modéliser la nouvelle contrainte dans le programme existant, on s'intéresse seulement à la période à laquelle commence l'examen x . On garantit ensuite qu'aucun autre examen ne commence dans la même salle durant la durée $d(x)$, qu'aucun examen x' présenté par un étudiant présentant aussi l'examen x ne commence dans les $d(x)$ périodes suivant le début de x , et similairement pour les paires d'examens surveillés par un même professeur.

$$\begin{aligned}
C'_3 &= \bigwedge_{t \in T} \bigwedge_{0 \leq \delta \leq d(x)-1} \bigwedge_{e \in E} \bigwedge_{x_i, x_j \in a(e), x_i \neq x_j} \neg \tau_{x_i, t} \vee \neg \tau_{x_j, t+\delta} \\
C'_4 &= \bigwedge_{t \in T} \bigwedge_{0 \leq \delta \leq d(x)-1} \bigwedge_{p \in P} \bigwedge_{x_i, x_j \in b(p), x_i \neq x_j} \neg \tau_{x_i, t} \vee \neg \tau_{x_j, t+\delta} \\
C'_5 &= \bigwedge_{t \in T} \bigwedge_{0 \leq \delta \leq d(x)-1} \bigwedge_{s \in S} \bigwedge_{x_i, x_j \in X, x_i \neq x_j} \neg \tau_{x_i, t} \vee \neg \tau_{x_j, t+\delta} \vee \neg \sigma_{x_i, t} \vee \neg \sigma_{x_j, t}
\end{aligned}$$

Notons que ce sont des généralisations de C_3, C_4 et C_5 . En effet, si on considère que tous les examens ont une durée d'une période ($0 \leq \delta \leq 0 \rightarrow \delta = 0$), les formules homologues sont équivalentes.

2.6 Les examens ne peuvent pas avoir lieu tout le temps, par exemple la nuit

On introduit les périodes interdites $I = \{(k_1^1, k_1^2), \dots, (k_i^1, k_i^2)\}$. On ajoute une simple contrainte unaire : aucun examen ne peut commencer entre ces deux bornes. Cette contrainte peut être exprimée en FNC de cette manière :

$$\bigwedge_{x \in X} \bigwedge_{k_i^1, k_i^2 \in I} \bigwedge_{k_j | k_i^1 \leq k_j \leq k_i^2} \neg \tau_{x, k_j}$$

Cette condition peut se lire de la manière suivante : Pour tout examen x , pour tout intervalle de temps interdit, pour toute heure faisant partie de cet intervalle, l'examen x ne peut commencer à cette heure.

Cependant, cette modélisation ne prend pas en compte des examens de durée supérieure à 1 unité de temps. Il faut donc rajouter ce paramètre dans la condition :

$$\bigwedge_{x \in X} \bigwedge_{k_i^1, k_i^2 \in I} \bigwedge_{k_j | k_i^1 - d(x) + 1 \leq k_j \leq k_i^2} \neg \tau_{x, k_j}$$

2.7 Maximum k changements de salle pour tous les étudiants

On introduit la variable $\kappa_{k,X,S,T}$ qui est vraie ssi la suite d'examens x_i de l'ensemble X qui se déroulent respectivement dans les salles $s_i \in S$ aux heures $t_i \in T$ provoque plus de k changements.

$$\bigwedge_{e \in E} \bigwedge_{x \in a(e)} \bigwedge_{s \in \mathcal{S}_{S,|x|}} \bigwedge_{t \in \mathcal{S}_{T,|x|}} \neg \kappa_{k,x,s,t} \vee \left(\bigvee_{x_i \in x, s_i \in s, t_i \in t, i \in |x|} \neg \sigma_{x_i, s_i} \vee \neg \tau_{x_i, t_i} \right)$$

Notation : $\mathcal{S}_{S,|x|}$ est l'ensemble des sous-ensembles ordonnés de taille $|x|$ de l'ensemble S

Intuitivement, on dit que pour tous les ordres d'examens d'un certain étudiant en temps et en salle, si cet ordre provoque plus de k changements, on l'invalidé. Voici le développement qui a été fait pour atteindre la clause ci-dessus :

$$\begin{aligned} \kappa_{k,x,s,t} &\rightarrow \neg(\sigma_{x_1, s_1} \wedge \tau_{x_1, t_1} \wedge \dots \wedge \sigma_{x_i, s_i} \wedge \tau_{x_i, t_i}) \\ \neg \kappa_{k,x,s,t} &\vee \neg(\sigma_{x_1, s_1} \wedge \tau_{x_1, t_1} \wedge \dots \wedge \sigma_{x_i, s_i} \wedge \tau_{x_i, t_i}) \\ \neg \kappa_{k,x,s,t} &\vee \left(\bigvee_{x_i \in x, s_i \in s, t_i \in t, i \in |x|} \neg \sigma_{x_i, s_i} \vee \neg \tau_{x_i, t_i} \right) \end{aligned}$$

Nous avons implémenté une fonction récursive qui génère toutes les combinaisons d'horaires différents qui provoqueraient une infraction de la contrainte. Chacune de ces combinaisons est ensuite invalidée en insérant une clause de littéraux négatifs (un littéral par examen qui appartient à la combinaison en infraction).

2.8 Une unité de temps tampon

Il suffit de rajouter cette contrainte (qui prend également en compte la durée d'un examen) :

$$\bigwedge_{e \in E} \bigwedge_{x_i, x_j \in a(e), x_i \neq x_j} \bigwedge_{s_i, s_j \in S, s_i \neq s_j} \bigwedge_{t \in T} \neg \sigma_{x_i, s_i} \vee \neg \sigma_{x_j, s_j} \vee \neg \tau_{x_1, t} \vee \neg \tau_{x_2, t+d(x_1)}$$

Elle spécifie simplement que deux examens passés par un même étudiant dans deux salles différentes ne peuvent pas commencer à la période qui suit directement la dernière période du premier examen.

2.9 Proposer d'autres contraintes naturelles et donner leur modélisation.

2.9.1 n unités de temps entre tout examen pour un étudiant

Nous avons trouvé utile d'insérer des périodes de temps entre les examens pour laisser du temps pour réviser aux étudiants. On peut modéliser cette contrainte en FNC (notons qu'il faut prendre en compte la durée des examens ici aussi) :

$$\bigwedge_{e \in E} \bigwedge_{x_i, x_j \in a(e), x_i \neq x_j} \bigwedge_{t \in T} \bigwedge_{0 \leq \delta \leq n} \neg \tau_{x_1, t} \vee \neg \tau_{x_2, t+d(x_1)+\delta}$$

3 Conclusion

Une multitude de bâtons fut placée dans nos roues ; toutefois, nous parvînmes à modéliser et implémenter avec brio la pléthore de contraintes requise. Notre programme – qui, soit dit en passant, produit des visuels qui impressionneraient le plus insensible des critiques artistiques – fonctionne sans accroc ; tout du moins avec les tests tumultueux que nous avons réussi à lui lancer. Succès s'ensuit.