

## AG41 : Challenge 2006

### Description du problème :

La tâche de réalisation de l'emploi du temps pour l'Université de Technologie de Belfort-Montbéliard est une tâche fastidieuse. Chaque semestre, une course contre la montre est entamée au service des moyens de l'enseignement pour établir l'emploi du temps des enseignements à partir des vœux des étudiants et des déclarations de charge des enseignants. Souvent l'emploi du temps est amené à s'adapter à la lumière des changements s'opérant sur les inscriptions des étudiants. La procédure de réalisation de l'emploi du temps occupe alors toute la durée de l'inter semestre soit environ 2 mois.

Il s'agit dans ce travail de proposer des outils automatiques d'aide à la réalisation de l'emploi du temps dans notre établissement.

On suppose que les horaires des séances de cours, TD et TP sont préfixés de manière à éviter les conflits sur les enseignants intervenants et sur les locaux. Il s'agit alors de concevoir un algorithme d'optimisation pour l'affectation automatique des étudiants aux séances (groupes) de TD et de TP de manière à réduire (annuler) le nombre de chevauchements de séances pour les étudiants.

### Formulation :

Soit  $I = \{(e, UV) / \text{l'étudiant } e \text{ est inscrit à l'unité de valeur } UV\}$  l'ensemble des inscriptions des étudiants aux UV.

Soit  $E = \{(UV, \text{type}, h) / \text{l'enseignement de l'unité de valeur } UV \text{ de type Cours, TD ou TP à lieu à la période } h\}$  l'emploi du temps pré-fixé. L'horaire  $h$  indique le jour, l'heure de début et de fin de la séance et enfin le type de semaine  $\{A, B, A/B\}$

La résolution du problème d'affectation des étudiants aux séances d'enseignement consiste à déterminer pour chaque enseignement  $(e, UV) \in I$  : les horaires  $h_C$ ,  $h_{TD}$  et  $h_{TP}$  associés à l'unité de valeur tel que  $(UV, 'Cours', h_C), (UV, 'TD', h_{TD}), (UV, 'TP', h_{TP}) \in E$ .

$S = \{(e, UV, h_C, h_{TD}, h_{TP})\}$  la solution du problème.

La solution doit satisfaire les conditions suivantes :

- Le nombre d'étudiants affectés au même horaire de TD ne dépasse pas  $Max_{TD}$ .
- Le nombre d'étudiants affectés au même horaire de TP ne dépasse pas  $Max_{TP}$ .

et réduire le nombre de conflits représentés par la formule suivante :

$$\sum_{(e, UV1) \in I} \sum_{\substack{(e, UV2) \in I \\ UV2 \neq UV1}} CONFLIT((e, UV1, h_{C1}, h_{TD1}, h_{TP1}), (e, UV2, h_{C2}, h_{TD2}, h_{TP2}))$$

CONFLIT représente le nombre de conflits entre les cours, TD et TP de l'UV1 et cours, TD et TP de l'UV2 et ceci pour l'étudiant  $e$ .

**Description des entrées/sorties :**

Les données d'entrée du problème sont représentées par deux fichiers texte correspondant aux deux ensembles I et E.

**Format du fichier des inscriptions**

<b>étudiant</b>	<b>Unité de Valeur</b>
e1	UV1
e1	UV2
...	
e2	UV2
e2	UV5
...	

**Format du fichier emploi du temps**

<b>UV</b>	<b>Type</b>	<b>Jour</b>	<b>Heure Deb</b>	<b>Heure Fin</b>	<b>Semaine</b>
UV1	Cours	MAR	1015	1215	A/B
UV1	TD	MER	1400	1600	A/B
UV1	TD	LUN	1015	1215	A/B
UV1	TP	VEN	0800	1100	A
UV1	TP	VEN	0800	1100	B
UV1	TP	JEU	1300	1600	A
UV1	TP	JEU	1300	1600	B
UV2	Cours	MAR	1615	1815	A/B
...					

Les données de sortie se résument dans le fichier solution représentant l'affectation des étudiants aux séances d'enseignement.

**Format du fichier solution**

<b>étudiant</b>	<b>UV</b>	<b>hC</b>	<b>hTD</b>	<b>hTP</b>
e1	UV1	MAR1015	LUN1015	VEN0800A
e1	UV2	MAR1615	JEU1015	MER1400B
...				
e2	UV2	MAR1615	JEU1015	MAR1400A
...				

**Règles du challenge**

- L'objectif du mini projet est de mettre les étudiants en compétition afin de les motiver et de les pousser à fournir les meilleurs résultats.
- Les étudiants doivent s'organiser en équipes de 3 ou 4 personnes afin d'effectuer le travail.
- Les étudiants peuvent utiliser les heures de TP afin d'avancer sur le sujet et de discuter avec le chargé de TP au sujet des obstacles qu'ils rencontrent.

- Avant toute implémentation chaque équipe doit fournir une proposition technique succincte et précise sur les structures de données à utiliser pour résoudre le problème ainsi que les bases de la méthode d'optimisation qu'il compte employer.
- Les étudiants peuvent s'échanger les résultats qu'ils obtiennent sur les scénarios intermédiaires fournis par les enseignants.
- Pour l'évaluation du meilleur algorithme, chaque équipe doit fournir un fichier exécutable Windows ou Unix admettant la syntaxe d'appel suivante :

*optim.exe fichier\_inscription fichier\_emploiutemps fichier\_solution temps*

Le paramètre *temps* indique le temps de recherche en secondes accordé au scénario.

Le fichier *optim.exe* doit être suffisant pour lancer l'application sur n'importe quel système Windows ou Unix.

- L'évaluation se fera sur la base de trois scénarios finaux qui ne seront connus qu'une semaine avant l'évaluation.
- Un prix sera décerné à la meilleure équipe.