

P.I. : PLANNING INTELLIGENT

Dossier des Spécifications

Auteurs : Corentin COUDRAY

Christophe JULIEN

Khanh An Noël TRAN

Groupe : M2_9

Date : 20/10/2013



38 rue de Molière
94200 IVRY sur SEINE
Tel. : 01 56 20 62 00
Fax. : 01 56 20 62 52
[http ://www.esme.fr](http://www.esme.fr)

Sommaire

1	Contexte	2
1.1	Introduction	2
1.2	Problématique	2
2	Existants	3
3	Objectifs	4
3.1	Fonctionnalités	4
3.2	Limites	4
	Références	5

1 Contexte

1.1 Introduction

Ce projet a pour but la réalisation d'une application de production d'un emploi du temps, et ainsi de faciliter le travail de M. Maeso. En effet, chaque année, il doit créer manuellement l'intégralité des emplois du temps de l'école, en tenant compte de toutes les contraintes de disponibilités des professeurs, des salles, des élèves, etc.

Mais ce problème peut être généralisé à toutes les planifications horaires d'employés, comme dans les écoles, les hôpitaux et toutes autres entreprises ayant un roulement de personnel à organiser.

La planification sous contraintes est la discipline des mathématiques appliquées consistant à ordonner diverses séquences ou événements dans un espace-temps limité. Ces planifications impliquent un grand nombre de contraintes, qu'elles soient liées aux disponibilités de personnes, d'emplacements ou à tous autres problèmes divers et variés.

La réalisation d'une planification est à la fois longue et périlleuse. Il est très difficile de concilier l'ensemble des contraintes qui accompagnent cet ordonnancement et de pouvoir organiser cela de manière optimale. La mise en place d'un calendrier est souvent faite à la main ou de manière approximative avec d'autres applications.

L'objectif de ce projet est donc de réaliser une application, permettant de générer une planification tenant compte de toutes les contraintes que l'on aura indiquées au préalable. Dans un premier temps, nous nous contenterons de générer un emploi du temps propre à l'école, quitte à élargir notre champ d'action si nous en avons le temps.

1.2 Problématique

Les algorithmes nécessaires à l'élaboration de ce projet relèvent des problèmes dit "NP-complets".

Les problèmes NP-complets sont des problèmes qui sont au minimum aussi difficiles que les problèmes NP. Ces derniers sont des problèmes de décision à temps polynomial. Ils vont être réalisables en listant l'ensemble des solutions puis en les testant. Le souci est que cela peut prendre énormément de temps en fonction de ce que l'on a en entrée du problème.

Les problèmes NP-complets peuvent avoir des solutions mais elles ne vont pas être trouvées de manière optimale. Le temps pour résoudre ces problèmes est de l'ordre exponentiel par rapport à ce que l'on a en entrée. On devra toujours se contenter d'une solution qui sera au mieux "satisfaisante". [1]

2 Existants

Aujourd'hui, il existe déjà des systèmes de planification informatique, mais ils correspondent toujours à des cas bien spécifiques : organisation du personnel médical dans un hôpital, organisation du personnel dans une base militaire, etc... Mais il semble impossible à l'heure actuelle d'imaginer un logiciel générique permettant de lister des contraintes et d'élaborer le planning correspondant et adapté quel que soit le milieu professionnel.

Ainsi, il nous faudra donc partir de rien. Nous pourrons tout de même nous inspirer des méthodes de résolution de ces plannings pour essayer de les adapter à notre problème.

Nous avons également trouvé des éléments de recherche :

- **Algorithme des colonies de fourmis** : cet algorithme est basé sur le comportement réel des fourmis, qui parviennent, par le biais de rétroactions, à trouver le chemin le plus court entre une source de nourriture et leur nid. [2]
- **Recherche tabou** : cet algorithme consiste à partir d'une position donnée, et à explorer ensuite le voisinage pour choisir la position qui minimise la fonction objectif. Le mot 'tabou' vient du fait qu'une fois une position écartée, on ne peut plus y revenir. Cet algorithme procède donc par élimination jusqu'à trouver une solution convenable. [3]

Il existe probablement d'autres algorithmes de ce genre, dont nous aurons besoin pour aboutir à une solution convenable. Nous devons employer des méthodes de résolution pas à pas, procédant par élimination, car il ne semble pas possible de créer un algorithme optimal qui génère un emploi du temps idéal, prenant en compte toutes les contraintes, et en les organisant de manière aussi intelligente que le ferait un être humain.

3 Objectifs

3.1 Fonctionnalités

L'objectif de notre projet est de réaliser un planning en fonction des contraintes que l'on aura indiquées. Nous allons donc réaliser l'emploi du temps de notre école de la manière la plus optimale possible.

Dans notre cas, l'utilisateur entrera la liste des matières qui sont enseignées dans l'école avec les différentes caractéristiques qui les accompagnent comme le nombre d'heures ou encore l'occupation des salles. Le logiciel indiquera l'ensemble des professeurs avec les différentes matières qu'ils enseignent et leurs disponibilités. Puis, il notera la liste des promotions avec les cours qui leurs sont associés. Et enfin, la liste des salles qui vont être adaptées pour une liste de cours. En effet, certains professeurs peuvent dispenser plusieurs cours, et certains cours peuvent être dispensés par plusieurs professeurs. De même, certains cours comme les TP ne peuvent être dispensés que dans certaines salles spécifiques. Il faudra donc dresser des listes d'associations entre les différents éléments de notre emploi du temps.

Une fois toutes ces contraintes posées, l'emploi du temps va être généré en fonction de celles-ci.

3.2 Limites

Le projet étant très complexe, ("np-complet"), nous ne pourrons pas faire un algorithme totalement optimisé. En effet, notre projet s'inscrit dans le cadre de l'heuristique. L'heuristique, est utilisée en Mathématiques Discrètes afin de trouver une solution réalisable, mais pas nécessairement optimale.

De plus, au regard du nombre de contraintes associées au problème, nous devons avancer pas à pas et commencer par écarter un bon nombre d'entre elles pour nous faciliter la tâche. Ainsi, au fur et à mesure que le projet avancera, on le complexifiera en incluant toujours plus de contraintes. Mais, le temps imparti étant limité, nous ne savons pas s'il nous sera possible d'inclure tous les facteurs liés à la planification dans notre logiciel.

Références

- [1] Michael R. Garey and David S. Johnson. *Computers and Intractability ; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [2] M. Dorigo, V. Maniezzo, and A. Coloni. Ant system : optimization by a colony of cooperating agents. *Trans. Sys. Man Cyber. Part B*, 26(1) :29–41, February 1996.
- [3] Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.