



UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIÈNE

THÈSE

Présentée par :

MOUMENE Khaled

Pour l'obtention du grade de : **MAGISTER**

En : **MATHÉMATIQUES**

Option : **RECHERCHE OPÉRATIONNELLE**

**LE PROBLÈME DE L'EMPLOI DU TEMPS, APPROCHE
THÉORIQUE ET STRATÉGIES PRATIQUES**

Soutenu le 12 septembre 2001

Devant le jury composé de :

Mr KHELLADI Abdelkader	Professeur à l'USTHB	Président
Mr AÏ DER Méziane	Maître de conférences à l'USTHB	Directeur de Thèse
Mr ABBAS Moncef	Maître de conférences à l'USTHB	Examineur
Mr BERRACHEDI Abdelhafid	Maître de conférences à l'USTHB	Examineur
Mr AIT-HADDADENE Hacène	Docteur d'Etat	Examineur
Mr CHERGUI Mohamed Amine	Chargé de Recherches à l'USTHB	Examineur
Mr MAAMRA Mohamed Saï d	Chargé de Recherches à l'USTHB	Examineur

Remerciements

Mes remerciements s'adressent à mes parents ainsi qu'à mes frères qui ont su me soutenir, m'encourager et conforter mon parcours. Qu'ils trouvent ici l'expression de ma sincère gratitude et ma plus grande reconnaissance.

Je tiens aussi à remercier Monsieur Méziane AÏDER, Maître de conférences, pour son suivi, ses orientations pertinentes et ses précieux conseils.

Je remercie Monsieur le Professeur Abdelkader KHELLADI qui me fait l'honneur de présider le jury.

Mes remerciements vont aussi à Messieurs Moncef ABBAS, Abdelhafid BERRACHEDI, Maîtres de conférences, le Docteur Hacène AIT-HADDADENE ainsi qu'aux Chargés de Recherches, Mohamed Amine CHERGUI et Mohamed Saïd MAAMRA, pour l'honneur qu'ils me font en participant au jury.

Mes plus vifs remerciements s'adressent aussi à mes chers amis qui n'ont jamais cessé de m'encourager et de me soutenir afin de mener à bien ce travail.

Enfin, je remercie tous les enseignants du Département de Recherche Opérationnelle ainsi que ceux qui ont participé de près ou de loin à ma formation et à l'aboutissement de ce travail.

Sommaire

Introduction générale	3
-----------------------------	---

Chapitre 1 : Introduction au problème de l'emploi du temps

1. Introduction.....	7
2. Description générale du problème	7
2.1.Contraintes du problème	8
2.2.Objectif du problème	10
3. Complexité et taille du problème.....	11
4. Quelques problèmes similaires	13
4.1.Construction d'un calendrier d'examens	13
4.2.Emploi du temps au sein d'une entreprise.....	14
4.3.Autres problèmes similaires	14

Chapitre 2 : Quelques modélisations et approches de résolution

1. Introduction.....	16
2. Simulation et conception manuelle.....	16
3. Nombre chromatique d'un graphe	16
4. Programmation mathématique	18
4.1.Affectation des enseignants aux groupes d'enseignement	18
4.2.Affectation des groupes d'enseignement aux locaux	19
5. Indice chromatique d'un graphe	21
6. Problème de flot dans les réseaux.....	22
6.1.Affectation des unités d'enseignement aux locaux	22
6.2.Affectation des groupes d'enseignement aux enseignants	23
7. Les heuristiques.....	24
7.1.Un exemple d'adaptation des algorithmes génétiques.....	25
8. Autres approches.....	30

Chapitre 3 : Définition et modélisation du problème

1. Introduction.....	32
2. Définition du problème à résoudre.....	32
2.1.Description du problème.....	32
2.2.Contraintes du problème	32
3. Formulation mathématique du problème	33
3.1.Groupement des séances et des salles.....	33
3.2.Modélisation du problème	34
4. Caractéristiques du modèle proposé	36
4.1.Propriétés d'un emploi du temps donné par le modèle.....	36
4.2.Nombre de conflits caractérisant un emploi du temps.....	37
4.3.Existence d'un ou de plusieurs emplois du temps	39
4.4.Taille du modèle	41
5. Adaptation du modèle à d'autres problèmes similaires	42
5.1.Construction d'un calendrier d'examens	42
5.2.Affectation des enseignants aux groupes d'enseignement	43
5.3.Catégorie de problèmes auxquels le modèle peut être adapté.....	44

Chapitre 4 : Méthode de résolution du problème

1. Introduction.....	46
2. Génération d'un emploi du temps à partir d'un autre	46
2.1.Définition du graphe de transition.....	47
2.2.Génération à partir d'un type de chemin du graphe de transition.....	48
2.3.Génération à partir d'un circuit du graphe de transition.....	50
2.4.Propriétés d'un flot généré à partir du graphe de transition.....	52
3. Choix des emplois du temps à générer	54
3.1.Définition d'un jeu de poids sur les arcs du graphe de transition.....	54
3.2.Poids spécifique d'un chemin sur le graphe de transition.....	58
3.3.L'algorithme "PlusCourtSpécifique".....	61
3.4.Succession de circuits de poids spécifique négatif.....	68
3.5.Complexité de l'algorithme "PlusCourtSpécifique"	71
4. Adaptation de la recherche taboue pour la résolution du problème.....	75
4.1.Présentation générale de la recherche taboue	75
4.2.Adaptation au problème.....	76
5. Une borne inférieure du nombre de conflits sur un emploi du temps.....	81

Chapitre 5 : Implémentation et résultats

1. Introduction.....	87
2. Implémentation.....	87
3. Données utilisées.....	88
4. Résultats obtenus.....	88
5. Amélioration des résultats.....	96

Conclusion générale.....	101
--------------------------	-----

Bibliographie	103
----------------------------	-----

Annexe

1. Introduction.....	106
2. Présentation du programme MTimeTabler.....	106

Introduction générale

Depuis la fin des années cinquante, plusieurs contributions relatives au problème de l'emploi du temps sont apparues, cela est dû, essentiellement, à sa complexité, sa grande taille et surtout au nombre important de problèmes pratiques que ce dernier regroupe.

D'une manière générale, le problème d'emploi du temps consiste à définir un certain nombre d'affectations qui permettent d'assigner plusieurs ressources (humaines, matérielles, ...etc.) sur une période de référence, tout en respectant les contraintes imposées par les entités citées (disponibilité des ressources, période de référence finie, ...etc.).

Les problèmes d'emplois du temps s'avèrent être NP-Dur et la taille de leurs instances se caractérisent souvent par leur très grande taille. Un exemple très simple de problème d'emploi du temps peut être décrit comme suit : nous considérons un ensemble d'événements E , un ensemble de périodes élémentaires P et un ensemble de contraintes C entre les événements du type «les événements x et y ne peuvent avoir lieu en même temps»; le problème consiste à assigner les événements aux périodes élémentaires de sorte que l'ensemble des contraintes est respecté. Ce type de problèmes est NP-Complet (pour $|P| \geq 3$) [14].

Les contraintes considérées peuvent différer d'un problème à un autre suivant la spécificité ainsi que les caractéristiques attendues de l'emploi du temps recherché. Les contraintes sont souvent classées en deux catégories, la première regroupe les contraintes impératives (un emploi du temps qui ne satisfait pas ce genre de contraintes et irréalisable et/ou inacceptable), la seconde catégorie regroupe des contraintes dont la satisfaction a différents degrés d'importances mais dont le non respect n'empêche pas une application plus ou moins «acceptable» de l'emploi du temps trouvé. Mis à part la *non simultanété* de réalisation de certains événements, nous pouvons citer parmi ces contraintes : l'imposition d'un *ordre* de réalisation d'un certain nombre d'événements, l'*exclusion* de la réalisation de certains événements en certaines périodes et la *spécification* explicite qu'un événement donné doit avoir lieu en une période donnée. Il est à noter qu'un emploi du temps respectant toutes les contraintes peut ne pas exister.

Pour ce problème, plusieurs objectifs peuvent être envisagés, nous pouvons par exemple, minimiser la période de référence sur laquelle l'emploi du temps sera appliqué, minimiser le nombre ou l'importance des ressources utilisées, minimiser le nombre de «trous» dans un emploi du temps, ...etc. Cela dit, vu la complexité de ce problème ainsi que sa taille,

la majorité des approches existantes relaxent les contraintes souhaitées (et parfois quelques autres) et se fixent comme objectif la minimisation des conflits par rapport à ces dernières.

Plusieurs approches ont été proposées pour la construction d'un emploi du temps, ces dernières sont souvent spécifiques aux problèmes qu'elles traitent. Nous pouvons citer parmi elles : les simulations manuelles, les méthodes basées sur la théorie des graphes et la programmation mathématique, les systèmes experts et les heuristiques.

La plus grande partie de ces approches utilisent, pour établir un emploi du temps, des heuristiques (dérivées ou non d'heuristiques générales d'optimisation comme la «recherche tabou», le «recuit simulé» ou autre), combinent des heuristiques avec des méthodes exactes, décomposent le problème en plusieurs sous problèmes plus faciles à cerner et à résoudre et cela, sans garantir qu'une solution existe ou qu'elle sera trouvée et peuvent même parcourir un domaine de solutions plus réduit que celui du problème initial, mais ces procédures permettent, généralement, d'avoir de «bonnes» solutions en un temps raisonnable et sont, en pratique, les méthodes les plus efficaces pour donner une réponse aux problèmes réels existants.

Nous nous intéressons, dans ce qui suit, aux problèmes d'emplois du temps qui peuvent se présenter au niveau des établissements d'enseignement. La conception manuelle de tels emplois du temps est une tâche ardue qui peut mobiliser plusieurs personnes pendant plusieurs jours; et face à cette situation les planificateurs d'emplois du temps sont prêts à adopter des outils basés sur des algorithmes d'optimisation robustes permettant d'automatiser et de faciliter leur tâche.

Nous décrivons, dans les pages qui suivent, une approche de résolution du problème d'emploi du temps basée sur l'heuristique de la recherche tabou couplée au problème de flots dans les réseaux.

L'approche en question a été programmée avec le langage orienté objets C++ sous l'environnement «Windows » et a été testée à la Faculté des Sciences Mathématiques de l'USTHB. Il est à noter que cette approche peut facilement être adaptée à d'autres problèmes similaires (problème d'examination, programmation de conférences, etc.).

Ce mémoire est constitué de cinq chapitres. Dans le premier, nous décrivons de façon générale et sommaire, le problème d'emploi du temps qui peut se poser au niveau des établissements d'enseignement. Nous passons en revue, dans le second chapitre, quelques modélisations et approches de résolution qui existent dans la littérature.

Au troisième chapitre, le problème que nous nous sommes proposés de résoudre est présenté puis modélisé sous forme d'un réseau sur lequel tout flot maximum correspond à un emploi du temps satisfaisant la majorité des contraintes. Certaines caractéristiques et propriétés du modèle en question y sont aussi exposées.

Le quatrième chapitre comporte l'essentiel de la démarche de résolution suivie. Elle consiste en l'élaboration d'une procédure efficace permettant le choix de l'emploi du temps à générer à partir d'un autre et cela en essayant de réduire, autant que possible, le nombre de séances d'un même groupe d'enseignement qui se déroulent simultanément. Cette procédure est ensuite exploitée lors de l'adaptation de la recherche taboue au problème. A la fin de ce chapitre, nous proposons une technique simple, basée sur les propriétés du modèle proposé, permettant d'évaluer une borne inférieure du nombre de conflits dus au déroulement simultané des séances d'un même groupe d'enseignement.

Le cinquième chapitre est réservé à l'implémentation de l'approche de résolution proposée ainsi qu'aux résultats obtenus. Nous y exposons aussi un algorithme simple permettant d'améliorer ces résultats.

Chapitre 1

Introduction au problème d'emploi du temps

I.1. Introduction

La confection d'emplois du temps a toujours été un des soucis majeurs pour la plupart des établissements d'enseignement. Les contraintes qui doivent être prises en considération sont généralement très nombreuses et diverses et peuvent être assez différentes d'un problème à un autre à cause de la diversité des normes pédagogiques et des méthodes de gestion qui régissent le fonctionnement de ces établissements.

Vu la complexité théorique et la taille colossale qui caractérisent les problèmes réels de ce type, la majorité des approches de résolution existantes se fixent comme objectif la satisfaction du plus grand nombre de contraintes possible en favorisant celles qui sont jugées les plus importantes et dont la satisfaction est primordiale pour engendrer un emploi du temps opérationnel.

I.2. Description générale du problème

Nous nous intéressons, dans ce qui suit, aux problèmes d'emplois du temps qui peuvent se présenter au niveau des écoles, universités ou autres établissements similaires et qui consistent à effectuer des affectations cohérentes des séances de cours, des enseignants et des locaux sur une période de référence finie tout en respectant un certain nombre de contraintes caractérisant l'emploi du temps voulu.

Pour ce faire, nous définissons les notions et paramètres suivants :

- *Programme pédagogique* : Définition d'un certain nombre de matières ou de modules devant être enseignés ainsi que le volume horaire nécessaire à l'enseignement de chacun d'entre eux.
- $GE = \{ge_1, ge_2, \dots, ge_{|GE|}\}$: Ensemble de tous les groupes d'enseignement (étudiants, élèves, ...etc) devant suivre le programme pédagogique. Toutes les personnes qui composent un même groupe d'enseignement suivent le même programme pédagogique.
- $ESN = \{esn_1, esn_2, \dots, esn_{|ESN|}\}$: Ensemble de tous les enseignants désignés pour assurer toutes les séances de tous les groupes d'enseignement.
- $L = \{l_1, l_2, \dots, l_{|L|}\}$: Ensemble des locaux (ou salles) existants de différents aménagements et capacités où doivent s'effectuer l'ensemble de tous les cours des groupes d'enseignements.
- $P = \{p_1, p_2, \dots, p_{|P|}\}$: Ensemble de périodes élémentaires de longueurs égales qui composent la période de référence.

- S : Ensemble de toutes les séances dictées par le programme pédagogique sur la période de référence pour tous les groupes d'enseignement. Chaque séance s ($\in S$) dure une période élémentaire et est associée à un unique groupe d'enseignement ge ($\in GE$).

Le problème consiste à établir un emploi du temps pour ces groupes d'enseignement, c'est à dire, à définir un ensemble : $ET = \{(s, e, l, p)_i\}$, où $i=1, \dots, N$ et $N \in \mathbb{N}$ } $\subset S \times E \times L \times P$, où $(s, e, l, p)_i$ désigne le fait que la séance s se déroule avec l'enseignant e dans la salle l à la période p ; et cela en respectant certaines contraintes qui garantissent la cohérence et la faisabilité de l'emploi du temps recherché.

I.2.1. Contraintes du problème

Comme déjà souligné précédemment, il existe une multitude de problèmes d'emplois du temps dont les formulations sont souvent spécifiques aux problèmes réels qu'ils traitent, cela est dû à plusieurs facteurs parmi lesquels nous pouvons citer :

- La différence des systèmes qui requièrent un emploi du temps (université, école, ...etc).
- La différence dans les méthodes de gestion et des règles ou lois relatives au système considéré (Norme pédagogique, règlement intérieur, ...etc).

Toute fois, nous pouvons citer un certain nombre de contraintes qui sont le plus souvent utilisées pour formuler ce type de problèmes :

I.2.1.1. Contraintes impératives

1. Les locaux ont une capacité limitée et sont en nombre limité.
2. Période de référence limitée.
3. Nombre d'enseignants limité.
4. Ne pas affecter un groupe d'enseignement à un local incompatible avec l'enseignement qui doit s'y effectuer.
5. Ne pas affecter un enseignant à plus d'un groupe d'enseignement ou à plus d'un local à la même période.
6. Ne pas affecter un local à plus d'un groupe d'enseignement ou à plus d'un enseignant à la même période.
7. Ne pas affecter un groupe d'enseignement à plus d'un local ou à plus d'un enseignant à la même période.
8. Un enseignant affecté à l'enseignement d'un groupe d'enseignement doit assurer toutes les occurrences de cet enseignement pour ce groupe durant toute la période de référence.

9. Le nombre de séances assurées par un enseignant ne doit pas dépasser sa charge horaire.
10. Réserver des créneaux consécutifs pour les enseignements de durées supérieures à une période élémentaire.
11. Disponibilité des enseignants.
12. Disponibilité des locaux.
13. Disponibilité des groupes d'enseignement.
14. Assurer le volume horaire prévu pour chaque enseignement (ce volume est défini par le programme pédagogique).
15. Les enseignements assurés par un enseignant doivent être compatibles avec ses qualifications.

I.2.1.2.Contraintes souhaitables

16. Les séances qui utilisent plus d'un créneau (une période élémentaire) doivent avoir lieu dans le même local.
17. Ne pas affecter un groupe d'enseignement à un local de capacité nettement supérieure au nombre d'étudiants dans ce groupe.
18. Un enseignant peut définir un groupe d'enseignants avec lesquels il préfère travailler.
19. Un enseignant peut définir un ensemble de périodes à exclure de son planning.
20. Il est préférable qu'un enseignant soit le plus qualifié possible pour assurer un enseignement donné.
21. Distribuer le plus uniformément possible, sur la période de référence, les enseignements d'un même cours de fréquence supérieure à un.
22. Il est préférable d'éviter de mettre des enseignements dans certains créneaux de la période de référence.
23. Si un enseignement doit être enseigné plus d'une fois sur la période de référence, il est en général exclu que deux de ses unités aient lieu dans la même journée.
24. Il faut éviter de faire venir les groupes d'enseignement (respectivement enseignants) pour très peu de séances de cours.
25. Eviter aux groupes d'enseignement (respectivement aux enseignants) des pertes de temps par de trop longs espacements entre deux séances d'une même journée.
26. Respecter quelques contraintes de pré-affectations (de certains enseignants à certains groupes d'enseignement, de certains groupes d'enseignement à certains locaux, ...etc.).

27. Aucun groupe d'enseignement (respectivement enseignant) ne doit avoir plus de x cours sur y périodes consécutives.
28. Uniformité de la distribution des enseignements sur la période de référence (par rapport aux groupes d'enseignement, aux enseignants et aux locaux).
29. Il est préférable que la distance entre les locaux dans lesquels se déroulent des enseignements successifs soit réduite le plus possible.
30. Certains enseignements ne peuvent être programmés en certaines périodes.
31. Il est préférable, généralement, d'éviter la succession de certains enseignements.

Remarque

- Selon le problème considéré, il peut y avoir des contraintes de plus et/ou de moins.
- L'appartenance d'une contrainte à une catégorie (impérative ou souhaitable) peut changer d'un problème à un autre.
- Un emploi du temps satisfaisant toutes les contraintes risque de ne pas exister vu l'aspect conflictuel qui peut exister entre certaines contraintes (par exemple, les contraintes 21 et 25 sont conflictuelles).

I.2.2.Objectif du problème

Selon les caractéristiques attendues de l'emploi du temps recherché, la formulation d'une fonction objectif peut être différente d'un problème à un autre; aussi, nous pouvons facilement voir que certaines des contraintes citées ci-dessus peuvent être formulées sous forme d'une fonction objectif.

Parmi les objectifs de ce type de problème, nous pouvons citer :

1. Minimiser la période totale pour effectuer tous les enseignements.
2. Minimiser le nombre ou la quantité des ressources utilisées (locaux, enseignants, ...etc).
3. La meilleure répartition possible des séances d'enseignement sur les salles, les enseignants ou bien les groupes d'enseignement.
4. Minimiser le nombre de « trous » dans l'emploi du temps (par rapport aux groupes d'enseignement ou aux enseignants).

Cela dit, dans la majorité des cas, plusieurs contraintes sont relaxées et introduites comme une fonction objectif dont l'optimisation permet de se « rapprocher le plus possible de la satisfaction » des contraintes. Ceci permet de réduire, relativement, la taille du problème et

de réduire, avec cette nouvelle formulation, le risque de non existence (ou d'existence de peu) de solutions au problème.

Remarque

Les travaux sur les problèmes de l'emploi du temps de ce type tiennent compte soit globalement ou partiellement de tous les aspects du problème posé, ainsi, il existe des formulations qui :

1. Ne considèrent que l'affectation des enseignants aux groupes d'enseignement sur la période de référence.
2. Ne considèrent que l'affectation des groupes d'enseignements (ou des séances) aux salles sur la période de référence.
3. Considèrent l'affectation des groupes d'enseignement, des enseignants et des salles sur la période de références.

I.3. Complexité et taille du problème

Généralement, les modèles qui reflètent les problèmes d'emplois du temps concrets n'ont pas une structure assez spécifique pour être résolus complètement par des méthodes exactes et efficaces [14] (comme le simplexe, les algorithmes pour le calcul des flots, ...etc) et possèdent souvent une taille assez importante, ce qui rend l'application des méthodes exactes assez inefficaces.

Bien qu'il existe plusieurs types et modélisations du problème, il a été prouvé que les problèmes communs d'emplois du temps de ce type sont NP-Dur [13], donc le temps de calcul nécessaire pour trouver une solution optimale ou de prouver qu'une solution donnée est optimale, devient important quand la taille du problème devient grande, ce qui est très souvent le cas des problèmes concrets rencontrés.

A titre d'exemple, une université comme l'USTHB comportait, durant l'année universitaire 1998/1999 [3] :

- 180 sections (ou groupes d'enseignement)
- 300 locaux
- 36 périodes d'études élémentaires (qui s'étalent sur une semaine)

Si nous adoptons les hypothèses simplificatrices suivantes :

- Nous ne considérons pas l'affectation des enseignants (qui s'élevait, pour l'année universitaire en question, à plus de 1700 enseignants).
- A chaque section correspond, en moyenne, 20 séances qui doivent s'effectuer sur la période de référence (en fait, ce nombre est en réalité plus important).
- Nous ne considérons que trois groupes de contraintes essentiels :
 1. A une période donnée, chaque salle doit accueillir, au plus, une unique séance.
 2. A une période donnée, chaque séance doit être affectée à, au plus, une unique salle.
 3. Toutes les séances prévues doivent être affectées.

Alors, en adoptant un programme linéaire en nombres entiers à variables bivalentes (0 ou 1), exprimant l'affectation (ou non) d'une séance donnée à une salle et à une période donnée, pour modéliser les contraintes du problème, nous pouvons facilement vérifier que :

- Le nombre de variables sera de l'ordre de 39.10^6 (*nombre de séances x nombre de salles x nombre de périodes*).
- Le nombre de contraintes sera de l'ordre de $1,4.10^5$ ([*nombre de séances x nombre de périodes*] + [*nombre de salles x nombre de périodes*] + *nombre de séances*).

Notons ici, que même avec un ordre de grandeur aussi important, nous restons très loin d'une modélisation pouvant engendrer un emploi du temps opérationnel vérifiant toutes les contraintes du problème réellement posé.

En fait, si cette modélisation couvrait la plus grande majorité des contraintes du problème, nous pouvons atteindre : un nombre de variables de l'ordre de 10^{11} et un nombre de contraintes de l'ordre de 10^9 [3].

Dans les problèmes d'emploi du temps réels, quand le nombre de séances devient supérieur à 1000, le nombre de variables et de contraintes devient si important que plusieurs algorithmes deviennent impraticables ou impossible à implémenter. De plus, si d'autres contraintes exprimant des préférences, des indisponibilités ou autres devaient être prises aussi en considération, le temps de calcul peut devenir excessivement long et complètement déraisonnable [31] (qui peut être de l'ordre de plusieurs années, voir plusieurs siècles avec les moyens de calcul actuels).

Face à la complexité théorique et la taille importante des problèmes d'emploi du temps réels, les approches heuristiques pour la résolution de ces derniers restent les plus efficaces et les plus souvent utilisées car elles réalisent le meilleur compromis entre la qualité de l'emploi du temps obtenu et le temps de calcul nécessaire pour le retrouver.

I.4. Quelques problèmes similaires

Le problème de l'emploi du temps regroupe une grande catégorie de problèmes d'optimisation combinatoire qui peuvent se présenter dans la réalité, il serait donc difficile de les énumérer tous. Toute fois, ils apparaissent tous comme étant la planification d'un certain nombre d'événements (rencontres, affectations, associations, ...etc) sur une période de référence avec un ensemble de contraintes entre eux, notamment, la non simultanété de réalisation de certains événements.

A titre d'exemple, nous pouvons citer les problèmes suivants :

I.4.1. Construction d'un calendrier d'examens

Etant donné une période de référence P et des groupes d'étudiants devant subir un certain nombre d'examens, le problème consiste à mettre un calendrier d'examens sur la période considérée de sorte que chaque étudiant passe tous ses examens. Ce problème reste l'objet de plusieurs travaux [14][30][8][9][40][7] car très répondu et devient particulièrement complexe dans les établissements qui gèrent une importante ressource humaine.

La similarité entre ce problème et celui de l'emploi du temps apparaît en tenant compte des considérations suivantes :

- Un examen subi par un étudiant correspond à une séance d'enseignement.
- Un examinateur (ou surveillant) correspond à un enseignant.

Ainsi, les contraintes considérées pour le problème d'emploi du temps peuvent s'appliquer au problème de construction d'un calendrier d'examens.

Il est à noter que le problème peut se poser d'une manière différente (ce qui est souvent le cas) : au lieu de considérer une période fixée à l'avance, il s'agit de construire un calendrier d'examens sur un *minimum de temps*.

Cette étroite similarité entre les deux problèmes ainsi que le fait que ces derniers ont été largement appréhendés, induit qu'une approche de résolution de l'un d'entre eux peut, éventuellement, être adaptée et appliquée à l'autre.

I.4.2. Emploi du temps au sein d'une entreprise

Alors que la compétitivité au niveau mondial devient de plus en plus importante, les entreprises sont poussées, entre autre, à organiser leurs modes de travail afin d'effectuer, de la meilleure manière possible, les tâches qui leurs sont assignées en un temps record.

Nous nous plaçons dans le cas d'une entreprise ayant un unique planificateur qui dirige un ensemble d'employés afin de mener à bien un ensemble de tâches répétitives sur une période de référence donnée.

Chaque employé est plus ou moins compétent sur chacune des tâches existantes et chaque tâche requière un nombre donné d'employés compétents sur cette dernière. L'entreprise se fixe aussi comme objectif la formation des employés sur les tâches pour lesquelles ils ne sont pas encore compétents afin d'assurer sa sécurité à long terme (indisponibilité future des employés : retraites, démissions, maladies, ...etc).

Le problème consiste à déterminer un planning d'affectations des employés aux tâches (sécurité à court terme) et aux formations (sécurité à long terme), et cela en tenant compte d'un certain nombre de contraintes qui peuvent être du type :

- Un employé ne peut être affecté qu'à une seule tâche sur une période élémentaire donnée.
- A un moment donné, un employé est soit assigné à une tâche soit il est en formation.
- Les indisponibilités des employés doivent être respectées.
- Un employé donné doit être affecté à une tâche pendant une durée minimale et une durée maximale.
- Respect de certaines affectations imposées par le planificateur ...etc. [29]

I.4.3. Autres problèmes similaires

Nous pouvons citer, parmi tant d'autres, les problèmes suivants : la programmation des conférences, la planification des compétitions sportives, programmation des tâches devant être exécutées sur des processeurs et l'établissement d'un planning de surveillance [11].

Chapitre 2

Quelques modélisations et approches de résolution

II.1. Introduction

Plusieurs modélisations et approches de résolution existent dans la littérature. Certaines donnent lieu à des problèmes d'optimisation standards et connus pour leur difficulté (coloration des sommets ou des arêtes d'un graphe, programmation linéaire en nombres entiers, ...etc.) pour ensuite adopter une procédure de résolution appropriée, alors que d'autres sont basées sur des formulations plus simples destinées à être résolues par des méthodes approchées.

Par ailleurs, du fait qu'il existe plusieurs types de problèmes d'emploi du temps, il ne peut exister une formulation commune à ces derniers; cela dit, certains modèles simples et techniques de résolution existent et peuvent, éventuellement, être développés pour répondre aux besoins d'un problème particulier.

Dans ce qui suit, nous passons en revue quelques modélisations et approches de résolution du problème d'emploi du temps qui existent dans la littérature.

II.2. Simulation et conception manuelle

Cela est fait, généralement, par un ou plusieurs personnes (experts) qui ont acquis, sur une période relativement longue, une certaine expérience du comportement du système, en se basant sur quelques méthodes et heuristiques simples ainsi que sur les emplois du temps établis ultérieurement et, suivant la taille et la complexité du système traité, ces procédures sont relativement longues et avec l'évolution des méthodes d'éducation et de gestion, ces méthodes risquent de devenir très vite inefficaces.

Certains travaux se sont orientés vers l'implémentation des méthodes et des heuristiques utilisées (ou non) par ces experts sous forme de systèmes interactifs d'aide à la décision pour faciliter leurs tâches [5][21].

II.3. Nombre chromatique d'un graphe

Il s'agit de colorier les sommets d'un graphe avec un nombre minimum de couleurs (*nombre chromatique du graphe*) tel que deux sommets adjacents quelconques n'ont pas la même couleur.

Cette approche est souvent proposée pour le problème d'examen où les sommets représentent les examens et il y a une arête entre deux sommets si les deux examens qui leur correspondent ne peuvent se dérouler en même temps (un étudiant en commun, salle

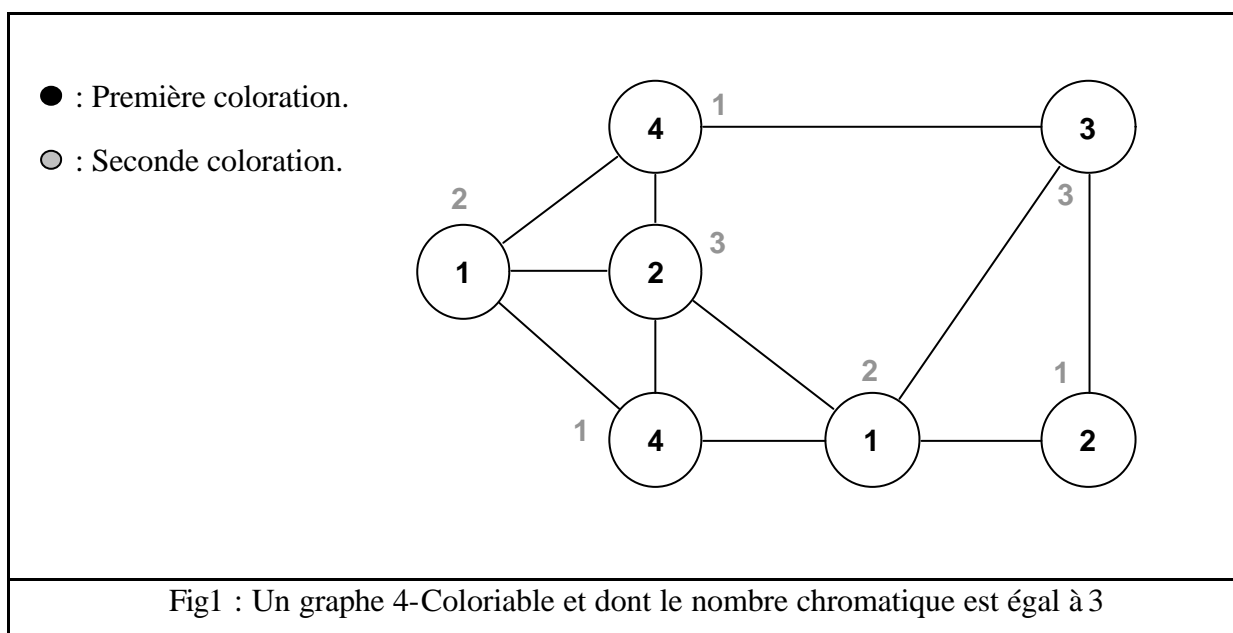
commune, ...etc.). Dans ce cas, le nombre minimum de couleurs est le nombre de périodes minimum pour le déroulement de tous les examens.

Le problème de coloration des sommets d'un graphe peut se poser autrement, en effet, il s'agit de vérifier qu'un graphe est *p-coloriable*, c'est à dire vérifier s'il est possible d'affecter aux sommets de ce dernier *p* couleurs sans que deux sommets adjacents n'aient la même couleur. Cette formulation peut être adoptée lorsque le nombre de périodes du problème d'emploi du temps est fixé au départ (ce qui est souvent le cas).

Plusieurs aspects du problème d'emploi du temps peuvent être modélisés sous cette forme, à titre d'exemple, nous pouvons citer les modèles suivants :

1. Les sommets correspondent aux unités d'enseignement, et deux enseignements sont mis en correspondance s'ils ne peuvent, pour une raison ou une autre, se dérouler en même temps.
2. Les sommets représentent l'ensemble des cours C (un cours est un couple (unité d'enseignement e, local l) ou e est affecté au local l) et deux cours sont mis en correspondance si les deux unités d'enseignement sont incompatibles ou se déroule dans le même local [11].

Le problème de recherche du nombre chromatique d'un graphe est NP-Complet ce qui rend la résolution des problèmes de grandes tailles de ce genre difficile et qui a conduit à l'élaboration de plusieurs heuristiques [14][9] pour trouver le nombre chromatique ainsi que la coloration correspondante.



L'une des méthodes les plus simples pour estimer le nombre chromatique d'un graphe consiste à ordonner les sommets de ce dernier par ordre décroissant de leur degré (nombre de sommets adjacents) et d'assigner à chaque sommet, dans cet ordre, la couleur la plus petite [14] (nous supposons ici que les couleurs sont identifiées par des nombres distincts).

II.4. Programmation mathématique

Plusieurs modèles de ce type ont été proposés [3][14][37][11][30] vu leur flexibilité considérable pour exprimer plusieurs types de contraintes, ces modèles sont souvent des programmes linéaires en nombres entiers à variables bivalentes (0 ou 1) qui expriment, généralement, des affectations (des groupes d'enseignement, des enseignants, des ressources matérielles, des périodes, ...etc.).

Les modèles de base les plus simples que nous pouvons citer à titre d'exemple sont les suivants :

II.4.1. Affectation des enseignants aux groupes d'enseignement

Soient GE et E les ensembles de groupes d'enseignement et d'enseignants respectivement, et soit r_{ij} le nombre de périodes pour lesquelles le groupe d'enseignement s_i ($\in GE$) doit être enseigné par l'enseignant e_j ($\in E$), et p le nombre de périodes de référence.

Il s'agit de résoudre le problème (P_1) suivant :

$$(P_1) \left\{ \begin{array}{ll} \sum_{k=1}^p x_{ijk} = r_{ij} & (i = 1, \dots, |GE|, j = 1, \dots, |E|) \quad (1) \\ \sum_{j=1}^{|E|} x_{ijk} \leq 1 & (i = 1, \dots, |GE|, k = 1, \dots, p) \quad (2) \\ \sum_{i=1}^{|GE|} x_{ijk} \leq 1 & (j = 1, \dots, |E|, k = 1, \dots, p) \quad (3) \\ x_{ijk} \in \{0,1\} & \end{array} \right.$$

avec :

$$x_{ijk} = \begin{cases} 1 & \text{Si } s_i \text{ rencontre } e_j \text{ à la } k^{\text{ème}} \text{ période} \\ 0 & \text{Sinon} \end{cases}$$

- (1) : Tous les enseignements doivent avoir lieu.
- (2) : Un groupe d'enseignement rencontre au plus un enseignant par période.
- (3) : Un enseignant rencontre au plus un groupe d'enseignement par période.

Proposition 1 [14]:

Il existe une solution au problème (P_1) si et seulement si :

$$\begin{cases} \sum_{i=1}^{|GE|} r_{ij} \leq p & (j = 1, \dots, |E|) \\ \sum_{j=1}^{|E|} r_{ij} \leq p & (i = 1, \dots, |GE|) \end{cases}$$

Ce résultat est une conséquence directe du théorème de König sur la coloration des arêtes dans un multigraphe biparti [14], il affirme qu'une condition nécessaire et suffisante pour l'existence d'un emploi du temps sur p périodes est qu'aucun enseignant (respectivement groupe d'enseignement) n'est requis pour plus de p périodes.

II.4.2. Affectation des groupes d'enseignement aux locaux

Nous considérons un ensemble de cours $K = \{K_1, K_2, \dots, K_q\}$ qui doivent s'effectuer sur p périodes élémentaires où chaque cours K_i ($i=1, \dots, q$) comporte k_i séances et chaque séance dure une période élémentaire, nous considérons aussi l'ensemble $GE = \{GE_1, GE_2, \dots, GE_r\}$ des groupes d'enseignement devant suivre les cours en question, et l_k le nombre de salles disponibles à la k ième période [14].

Il s'agit de résoudre le problème (P_2) suivant :

$$(P_2) \begin{cases} \text{Max} \sum_{i=1}^q \sum_{k=1}^p C_{ik} y_{ik} & & \\ \sum_{k=1}^p y_{ik} = k_i & (i = 1, \dots, q) & (4) \\ \sum_{i=1}^q y_{ik} \leq l_k & (k = 1, \dots, p) & (5) \\ \sum_{\{i / i \text{ est une séance de } K_i \text{ destinée à } GE_l\}} y_{ik} \leq 1 & (l = 1, \dots, r / k = 1, \dots, p) & (6) \end{cases}$$

où :

$$y_{ik} = \begin{cases} 1 & \text{Si une séance de cours } K_i \text{ est programmée à la période } k \\ 0 & \text{Sinon} \end{cases}$$

C_{ik} : Un coefficient qui mesure la préférence d'affecter une séance de cours de K_i à la $k^{\text{ème}}$ période.

- (4) : Toutes les séances doivent être programmées.
- (5) : Disponibilité suffisante des salles à chaque période pour le déroulement de toutes les séances.
- (6) : Chaque groupe d'enseignement a, au plus, une unique séance à chaque période.

La majorité des modèles formulés de cette manière sont résolus en utilisant la relaxation lagrangienne [14][37][11], où un certain nombre de contraintes sont introduites dans la fonction objectif; si pour cela nous considérons, par exemple, les contraintes du type (6), nous aurons à résoudre un problème de transport.

Bien entendu, les deux problèmes cités ci-dessus sont parmi les plus simples, et aux contraintes qu'ils comportent viennent souvent s'ajouter plusieurs autres pour refléter le problème réellement posé ce qui rend leur résolution encore plus difficile.

Il est à noter que les modèles ainsi construits n'ont pas, généralement, une structure assez spécifique pour être résolus complètement par des méthodes exactes et efficaces [11][14] (comme le simplexe, les algorithmes pour le calcul des flots, ...etc.) et possèdent souvent une taille assez importante, ce qui rend l'application des méthodes exactes inefficace en terme de temps de calcul.

D'une manière générale, le problème de l'emploi du temps ainsi modélisé est souvent résolu par l'une des méthodes suivantes :

- La décomposition en plusieurs sous problèmes [3][11], ce qui peut engendrer plusieurs types de problèmes assez spécifiques (comme le problème de Bin-packing, le problème de transport, problème d'affectation généralisée, ...etc.), avant de résoudre ces derniers par des méthodes appropriées.
- La relaxation Lagrangienne qui est souvent couplée à des méthodes exactes [37][11] (comme le simplexe, la méthode du sous gradient, Branch and bound, génération des colonnes, ...etc.) et/ou des heuristiques.
- Utilisation exclusive des heuristiques.

II.5. Indice chromatique d'un graphe

Il s'agit de colorier les arêtes d'un graphe avec le nombre minimum de couleurs (appelé *indice chromatique* du graphe) ou avec un nombre p donné de couleurs (si cela est possible) sans que deux arêtes adjacentes quelconques n'aient la même couleur.

Si nous reprenons l'exemple du problème cité en 4.1 (affectation des enseignants aux groupes d'enseignement), nous pouvons considérer un multigraphe biparti $G=(GE \cup E, R)$, où les sommets représentent les enseignants et les groupes d'enseignement et un sommet s_i (représentant le groupe d'enseignement i) est relié par r_{ij} arêtes au sommet e_j (représentant l'enseignant j) et chaque couleur attribuée à une arête correspond à une période élémentaire donnée.

Etablir un emploi du temps sur p périodes élémentaires revient à colorier les arêtes de G avec au plus p couleurs sans que deux arêtes adjacentes quelconques n'aient la même couleur [14].

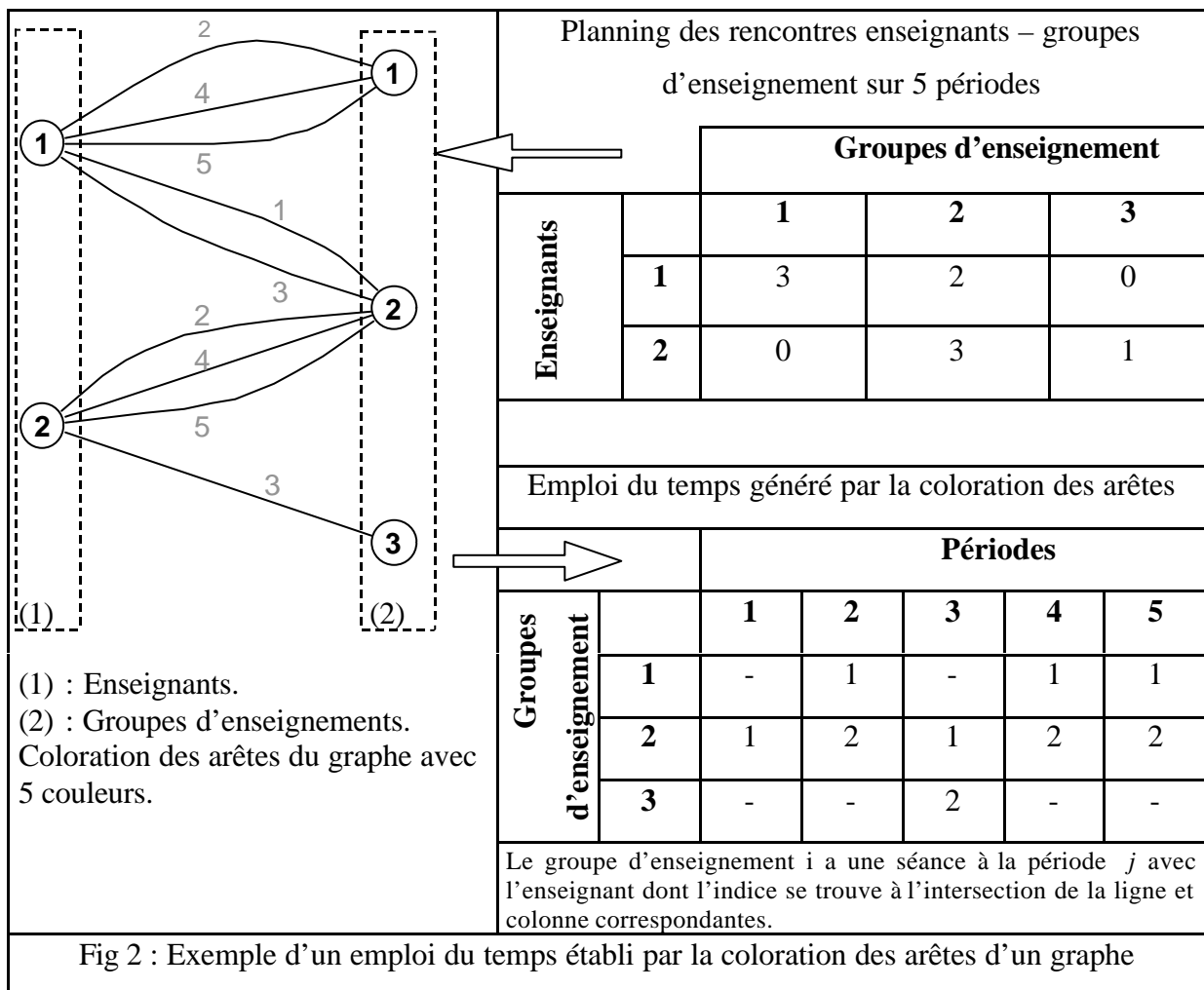


Fig 2 : Exemple d'un emploi du temps établi par la coloration des arêtes d'un graphe

II.6. Problème de flot dans les réseaux

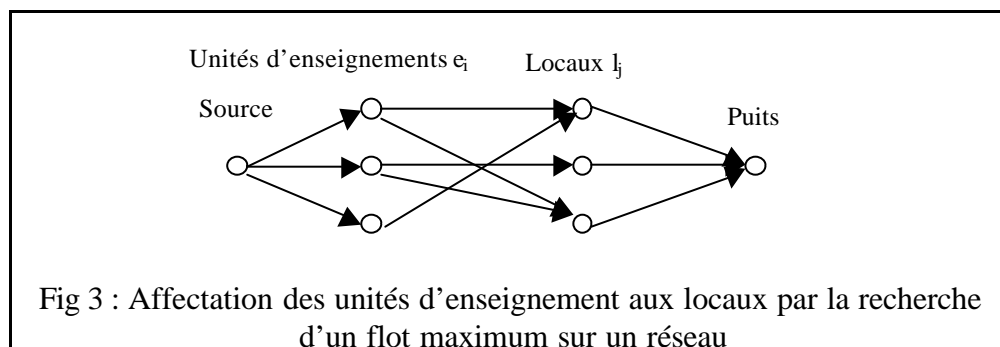
Les modèles du problème d'emploi du temps basés sur les réseaux peuvent être, comme observé dans plusieurs travaux [14], très intéressants puisqu'ils peuvent être résolus d'une manière efficace même si leur taille est importante, cela dit, généralement les problèmes d'emploi du temps réels ne peuvent être complètement formulés avec des réseaux.

Suivant les problèmes traités, plusieurs modèles basés sur le problème de flot ont été proposés. Parmi ces formulations, nous pouvons citer les exemples suivants :

II.6.1. Affectation des unités d'enseignement aux locaux

Nous considérons un ensemble U d'unités d'enseignement, chaque unité peut représenter un groupe d'enseignement et une séance de cours ; nous définissons aussi un ensemble L de locaux de différentes capacités et aménagements ou doivent se dérouler l'intégralité des unités d'enseignement en question.

Le problème peut être modélisé de la manière suivante :



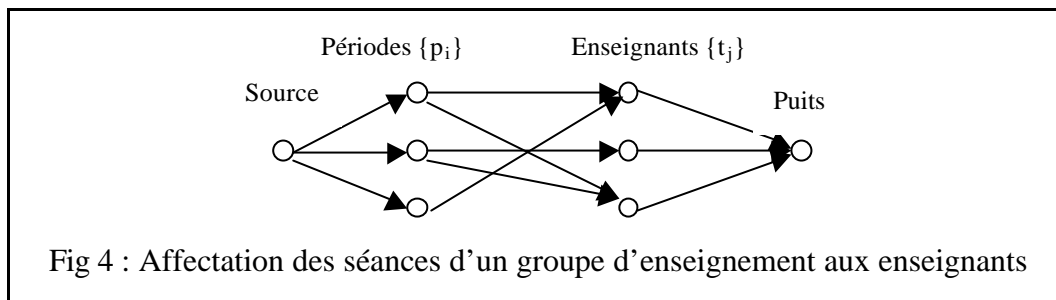
Les capacités des arcs reliant la source aux sommets représentant U sont égales à l'unité, la capacité d'un arc reliant un sommet l_j (représentant le local L_j) au puits est égale au nombre de périodes où le local L_j est disponible et les capacités sur le reste des arcs sont infinies.

Un arc reliant l'unité d'enseignement e_i et le local L_j existe si l'unité d'enseignement e_i est compatible avec le local L_j . Un flot maximum sur ce réseau affecte des unités d'enseignements aux locaux, et si la somme des flux entrants au puits est égale à la somme des unités d'enseignement, ces dernières seront toutes affectées [11].

II.6.2. Affectation des groupes d'enseignement aux enseignants

Pour cela, la méthode consiste à résoudre une séquence de problèmes de flot en essayant d'établir, pour chaque groupe d'enseignement, un calendrier de rencontres avec les enseignants sur la période de référence (qui est composée de plusieurs périodes élémentaires).

Pour ce faire, nous ne considérons qu'un groupe d'enseignement à la fois ainsi que tous les enseignants qui lui sont assignés, et nous construisons le réseau suivant :



Les arcs du type (t_j, puits) ont une capacité minimale et maximale égales à r_j (nombre de périodes où le groupe d'enseignement doit rencontrer l'enseignant t_j), le reste des arcs ont une capacité minimale et maximale égales à 0 et 1 respectivement. Un arc (p_i, t_j) existe si l'enseignant t_j est disponible à la période p_i.

Avec cette formulation, un flot sur ce réseau donne un emploi du temps pour un groupe d'enseignement.

Il est à noter que nous pouvons modéliser de cette manière un autre aspect du problème en considérant l'ensemble des locaux L au lieu de celui des enseignants E.

Cette procédure est poursuivie jusqu'à ce que tous les enseignements soient affectés ou jusqu'à ce qu'aucun enseignement ne puisse être affecté, à cette étape, certaines méthodes s'arrêtent et d'autres modifient les affectations précédentes pour continuer la résolution [14].

Le problème de recherche d'un flot maximum sur un réseau de transport est considéré comme « facile », en effet, il existe des algorithmes destinés à le résoudre en un temps polynomial [25] (en $O(N^3)$, où N est le nombre de sommets du graphe) et peuvent s'avérer assez efficaces pour des problèmes de grande taille.

II.7. Les heuristiques

Vu la complexité du problème ainsi que la grande taille qui, très souvent le caractérise, plusieurs travaux se sont orientés vers les heuristiques pour pouvoir le résoudre.

Il est à noter que parmi les approches de résolution utilisées, il en existe beaucoup qui sont spécifiques aux problèmes réels traités et suivant les cas, elles dérivent ou non d'heuristiques d'optimisation générales connues. Cela dit, les procédures qui dérivent d'heuristiques d'optimisation générales sont assez souvent utilisées et adaptées aux problèmes traités, nous pouvons trouver parmi elles : la recherche taboue [13][3][2][7] (très utilisée pour ce type de problème), le recuit simulé [30], FFD [11] (First Fit Decreasing), algorithme du Kangourou [3], algorithmes génétiques, réseaux de neurones [33] ...etc.

Nous donnons, ci-dessous, trois caractéristiques qui distinguent la majorité des procédures de résolution de ce type :

1. Il existe des approches de résolution qui sont appliquées à un modèle mathématique (expression des contraintes et éventuellement d'un objectif par un programme mathématique, un graphe, ...etc) préalablement établi comme étant une modélisation du problème. Ces approches n'exploitent que les caractéristiques mathématiques du modèle en question. Par ailleurs, d'autres approches exploitent directement les caractéristiques du problème posé sans passer, au préalable, par un modèle mathématique.
2. Le plus grand nombre de ces approches heuristiques affectent, un après un, les unités d'enseignement aux périodes après les avoir ordonnées par ordre « décroissant de leur difficulté d'affectation » (Respect d'un maximum de contraintes, priorité et importance d'une affectation,...etc). Pour cela, certaines méthodes se basent sur des algorithmes gloutons, c'est à dire que si une unité d'enseignement ne peut être affectée, elle est écartée (ne figurera pas dans l'emploi du temps) et la procédure continue en essayant d'affecter le reste des unités d'enseignement. D'autres méthodes procèdent autrement : si une affectation n'est pas possible, un retour aux affectations précédentes (un « Backtracking ») est prévu afin d'essayer de les modifier de sorte que l'affectation devienne possible.

3. A la fin de l'étape précédente, certaines procédures s'arrêtent alors que d'autres continuent en faisant une série de déplacements des unités d'enseignement en vue d'améliorer la qualité de l'emploi du temps. Dans ce cas, généralement beaucoup moins « d'efforts » sont faits pour retrouver un emploi du temps initial car, d'une part, cette solution n'est pas définitive et d'autre part, il est souvent préférable d'utiliser des algorithmes plus efficaces et qui nécessitent plus de temps de calcul lors de l'amélioration de la solution.

II.7.1.Un exemple d'adaptation des algorithmes génétiques

Nous donnons, ci-dessous, une description générale du principe des algorithmes génétiques ainsi qu'un exemple simple de la manière avec laquelle ils peuvent être adaptés au problème de l'emploi du temps.

II.7.1.1.Présentation générale des algorithmes génétiques

Les algorithmes génétiques peuvent être décrits comme étant des procédures qui imitent l'évolution génétique des espèces. L'idée générale consiste à considérer une *population* de « solutions » initiales (génération (1)) $\{x_1^{(1)}, x_2^{(1)}, \dots, x_N^{(1)}\}$ et de les faire évoluer dans le temps de sorte à avoir de meilleures « générations futures » (i.e de meilleures solutions).

Pour cela, les « solutions » d'une génération (n) doivent interagir pour créer des « enfants » (d'autres solutions) qui feront partie de la génération suivante ($n+1$), et cela à travers quatre opérations fondamentales : *sélection*, *croisement*, *mutation* et *substitution*. Ces opérations n'agissent pas directement sur l'espace des solutions, il faut tout d'abord les représenter en mots (ces représentations sont appelées aussi *chromosomes*) de longueurs finies composés d'éléments appartenant à un ensemble fini de symboles.

La sélection :

La sélection des « meilleurs » individus de la génération en cours se fait selon leur qualité (estimée à travers la valeur qu'ils donnent à la fonction objectif à optimiser) ; chaque individu est choisi au hasard avec une probabilité P_s qui augmente avec l'augmentation de la qualité de ce dernier.

Le croisement :

Cette opération est appliquée, avec une probabilité P_c , sur les paires d'individus sélectionnés à l'étape précédente. Il y a plusieurs manières d'effectuer cette opération, cela dépend de la spécificité du problème traité ainsi que de la manière avec laquelle est effectuée

le codage des solutions. Nous pouvons, par exemple, inter-changer les symboles se trouvant à une certaine position (choisie au hasard) des deux mots associés aux individus choisis, ceci engendrera deux autres individus (« enfants »).

La mutation :

Les individus obtenus à l'étape précédente sont soumis à une mutation avec une probabilité P_m . Un exemple simple illustrant cette opération consiste à remplacer, sur une position choisie au hasard, un symbole par un autre. Généralement, cette opération est appliquée sur une petite proportion de l'ensemble d'individus en question.

La substitution :

Après l'opération de mutation, les « plus mauvais individus » sont remplacés par les individus générés à l'étape précédente pour constituer la nouvelle génération. Un « mauvais individu » est sélectionné au hasard avec une probabilité P_v qui diminue avec l'augmentation de la qualité de ce dernier.

Ces opérations sont répétées jusqu'à ce que le nombre de générations parcourues soit jugé suffisant pour avoir une « assez bonne » solution.

Nous donnons ci-dessous une description schématique d'algorithme génétique typique, cependant, plusieurs variantes de ce dernier existent dans la littérature [34].

Nous considérons ici que la fonction objectif F du problème est à *maximiser* sur un ensemble X de solutions codées.

Algorithme :

(0) Choisir la population initiale d'individus $X^{(1)} \subseteq X$.

l : Nombre de générations à parcourir.

n : Indice de la génération $X^{(n)}$ en cours.

$\{x_1^{(n)}, x_2^{(n)}, \dots, x_N^{(n)}\}$: Solutions correspondantes à $X^{(n)}$.

F^* : Meilleure valeur actuelle de la fonction objectif (initialement, $F^* = \text{Max}\{F(x_i^{(1)}) / i=1, \dots, N\}$).

x^* : Meilleure solution rencontrée (initialement, x^* est la meilleure solution de $\{x_1^{(1)}, x_2^{(1)}, \dots, x_N^{(1)}\}$).

(1) **Pour** $n=1, \dots, l$ **faire**

(a) *Sélectionner* les meilleurs individus de $X^{(n)}$:

Soient $\{y_j, j=1, \dots, 2M / N \geq 2M\}$ $2M$ individus de $X^{(n)}$, la probabilité de choisir l'individu y_j , auquel correspond la solution $x_i^{(n)}$, est une fonction croissante de $F(x_i^{(n)})$.

(b) *Croisement*

Pour $k=1, \dots, M$, l'opération de croisement est appliquée sur les pairs (y_{2k}, y_{2k-1}) avec une probabilité p_c , ceci engendrera M pairs (z_{2k}, z_{2k-1}) (Chaque pair est identique à ses « parents » avec une probabilité $(1-p_c)$).

(c) *Mutation*

Pour $j=1, \dots, 2M$, l'opération de mutation est appliquée à z_j avec une probabilité p_m , ceci engendrera $2M$ individus « enfants » w_j (Chaque individu w_j est identique à z_j avec une probabilité $(1-p_m)$).

(d) *Substitution* des « mauvais » individus :

Choisir $2M$ individus de $X^{(n)}$, la probabilité de choisir un individu qui correspond à une solution $x_i^{(n)}$ étant une fonction décroissante de $F(x_i^{(n)})$. La génération $X^{(n+1)}$ est obtenue en remplaçant les $2M$ « mauvais » individus de $X^{(n)}$ par les individus « enfants » $\{w_j, j=1, \dots, 2M\}$.

Pour $p=1, \dots, N$ **faire**

si $F(x_p^{(n+1)}) > F^*$ **alors** $F^* = F(x_p^{(n+1)})$ et $x^* = x_p^{(n+1)}$.

Fin pour

Fin pour

Fin algorithme.

II.7.1.2. Adaptation des algorithmes génétiques au problème

Nous donnons ici une adaptation très simple des algorithmes génétiques au problème de l'emploi du temps. Pour cela reprenons le problème cité en 4.1 (Affectation des enseignants aux groupes d'enseignement).

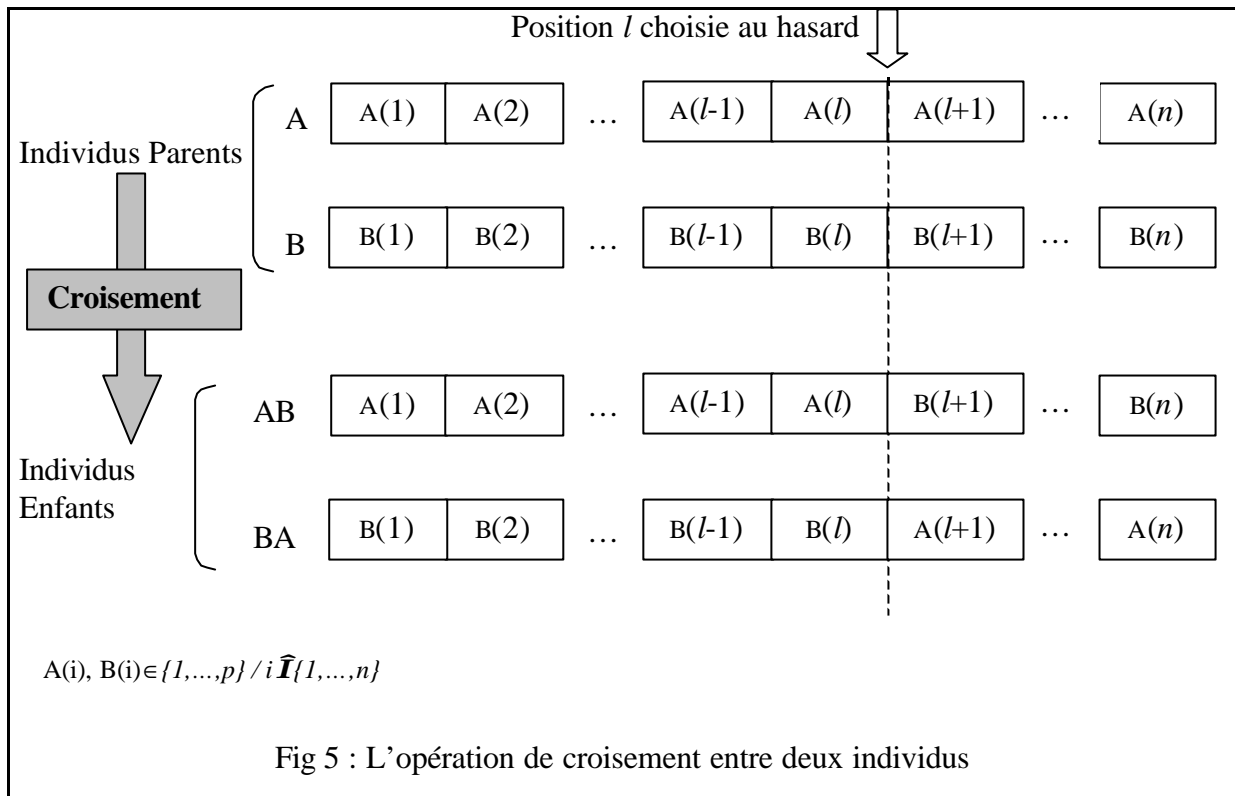
Comme déjà souligné, chaque solution doit être codée sous forme d'un mot (qui peut être vu comme étant un tableau de symboles appartenant à un ensemble fini). Pour cela, nous définissons un tableau pour lequel chaque cellule correspond à une unique rencontre définie par le triplet (groupe d'enseignement s_i , enseignant e_j , r) ($\in GE \times E \times \{1, \dots, r_{ij}\}$) et chaque rencontre correspond à une unique cellule.

Chaque solution (i.e : emploi du temps) sera représentée par le tableau défini ci-dessus et où chaque cellule prend ses valeurs dans $\{1, \dots, p\}$ (où p est le nombre de périodes élémentaires), définissant ainsi la période à laquelle doit se dérouler la rencontre qu'elle représente.

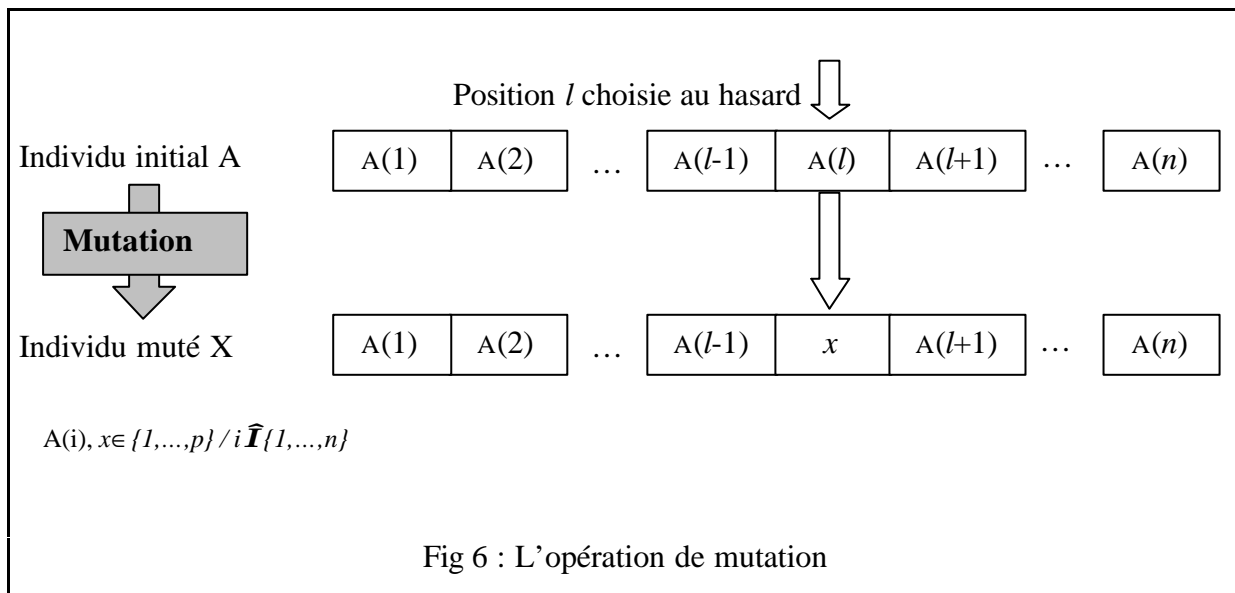
Avec ce codage, chaque solution parcourue vérifie le premier groupe de contraintes, à savoir, toutes les rencontres doivent avoir lieu. La fonction objectif peut être formulée de sorte qu'elle mesure le nombre de conflits dû au non respect du 2^{ème} et 3^{ème} groupe de contraintes, c'est à dire, un groupe d'enseignement (respectivement un enseignant) rencontre au plus un enseignant (respectivement un groupe d'enseignement) par période. Cette fonction sera donc à minimiser.

Chaque individu est sélectionné pour créer d'autres individus (respectivement être remplacé par d'autres individus) avec une probabilité qui augmente (respectivement diminue) avec l'augmentation de la qualité de l'individu en question.

L'opération de croisement se fait, avec une probabilité p_c , suivant le schéma ci-dessous :



L'opération de mutation se fait, avec une probabilité p_m , suivant le schéma ci-dessous :



II.8. Autres approches

D'autres approches de résolution du problème d'emploi du temps existent, parmi elles nous pouvons citer : la programmation logique [31], la programmation par contraintes [5] ainsi que les systèmes d'aide à la décision [21][8][9]. Ces derniers sont basés sur des heuristiques combinées ou non à des méthodes exactes et permettant à l'utilisateur de réajuster, tout en respectant certaines contraintes du problème, l'emploi du temps généré afin de refléter ses préférences. Ces systèmes permettent souvent aussi, la gestion d'une base de données (ce qui est très adéquat pour ce type de problèmes souvent caractérisés par leur grande taille) et sont, généralement, programmés sous un environnement interactif pour une meilleure aide dans la prise de décision.

Chapitre 3

Définition et modélisation du problème

III.1. Introduction

Après avoir abordé différentes formulations et approches de résolution de plusieurs types de problèmes d'emplois du temps, nous décrivons dans ce chapitre le problème que nous nous sommes proposés de résoudre puis nous donnons une modélisation mathématique de ce dernier sous forme d'un réseau sur lequel tout flot maximum correspond à un emploi du temps satisfaisant la majorité des contraintes du problème considéré.

Différentes propriétés du modèle ainsi formulé sont ensuite abordées, notamment celles qui permettront d'évaluer et d'améliorer la qualité d'un emploi du temps par rapport aux contraintes dont la satisfaction n'est pas garanti par le modèle proposé.

Nous montrons aussi comment ce modèle peut être adapté à plusieurs types de problèmes d'emploi du temps autres que celui considéré.

III.2. Définition du problème à résoudre

III.2.1. Description du problème

Nous nous intéresserons, dans ce qui suit, à la construction d'un emploi du temps en ne considérant que l'ensemble des séances S , des groupes d'enseignement GE , l'ensemble des locaux L ainsi que la période de référence P .

Le problème consistera donc à définir un ensemble $K = \{(s, l, p) \mid (s, l, p) \in S \times L \times P, i=1, \dots, n \text{ et } n \in N\}$ où un triplet $(s, l, p) \in K$ désigne le fait que la séance s se déroule à la période p dans la salle l , et cela en respectant un certain nombre de contraintes (qui seront définies par la suite) qui garantissent la cohérence et la faisabilité de l'emploi du temps qu'il représente.

L'objectif étant de satisfaire, de la « meilleure manière possible », l'ensemble des contraintes que nous aurons définies et d'affecter sans conflits le plus grand nombre de séances.

Aussi, nous considérerons par la suite, le problème de l'emploi du temps comme étant celui que nous aurons ainsi défini.

III.2.2. Contraintes du problème

Les contraintes considérées pour ce problème sont les suivantes :

- (1) Respect du programme pédagogique : Affectation de toutes les séances prévues.
- (2) Compatibilité entre types de séances et types de salles : Capacité et aménagement d'une salle compatible avec le nombre de personnes et la nature de la séance affectée à cette salle. Cela dit, d'une manière encore plus générale, nous considérons ici toutes les contraintes qui définissent la possibilité d'affecter une séance donnée à une salle donnée.

- (3) A une salle et à une période données est affectée, au plus, une unique séance.
- (4) Une séance $s (\in S)$ ne doit être affectée à une salle $l (\in L)$ à la période $p (\in P)$ que si s et l sont disponibles à cette période.
- (5) Satisfaction de certaines contraintes de pré-affectation (des séances aux salles et aux périodes).
- (6) Deux séances d'un même groupe d'enseignement ne peuvent se dérouler en même temps.

III.3. Formulation mathématique du problème

III.3.1. Groupement des séances et des salles

Les salles seront groupées par *types* suivant leur capacité et la nature de l'enseignement que nous pouvons y dispenser, de sorte que si un groupe d'enseignement peut avoir une leçon dans une salle d'un certain type, alors cela est possible dans n'importe quelle salle du même type.

L'ensemble des séances de chaque groupe d'enseignement doit être partitionné en ensembles de séances qui doivent se dérouler dans des types de salles différentes, de sorte que si une séance d'un ensemble donné peut se dérouler dans un certain type de salle alors cela est aussi possible pour le reste des séances du même ensemble (Exemple : si un groupe d'enseignement a trois (3) séances de mathématique et trois (3) séances de physique qui peuvent se dérouler dans les mêmes types de salles, ainsi qu'une (1) séance de travaux pratiques qui doit se dérouler dans une salle équipée d'ordinateurs, alors nous associons à ce groupe d'enseignement deux ensembles, le premier contiendra les six (6) premières séances et le second l'unique (1) séance de travaux pratiques).

Après l'opération de groupement nous obtenons une partition de l'ensemble des salles et une autre de l'ensemble des séances de chaque groupe d'enseignement.

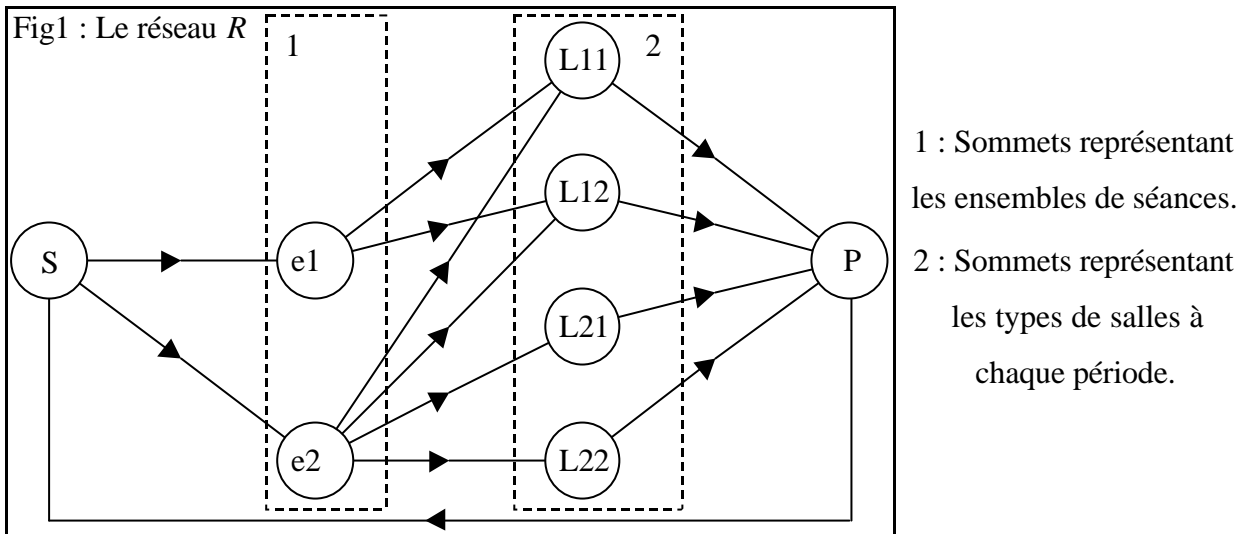
Afin de réduire le plus possible la taille du problème, cette opération doit être faite en essayant d'obtenir le moins de types de salles et d'ensembles de séances possibles, ceci implique une assez bonne connaissance de la nature des cours enseignés et des caractéristiques des salles disponibles.

Suite aux opérations de groupement, nous obtenons les ensembles suivants :

- $ES = \{es_1, es_2, \dots, es_{|ES|}\}$: Ensemble de tous les ensembles de séances associés aux groupes d'enseignement considérés.
- $TS = \{ts_1, ts_2, \dots, ts_{|TS|}\}$: Ensemble de tous les types de salles.

III.3.2. Modélisation du problème

Nous définissons un réseau $R=(X, U, d)$ de la manière suivante :



L'ensemble des sommets X est composé de :

- Sommets $\{e_i / i=1, \dots, |ES|\}$, où e_i représente le $i^{\text{ème}}$ ensemble de séances $es_i (\in ES)$.
- Sommets $\{L_{jk} / j=1, \dots, |TS| \text{ et } k=1, \dots, |P|\}$, où L_{jk} représente toutes les salles du type $ts_j (\in TS)$ à la $k^{\text{ème}}$ période.
- Une source s et un puits p .

L'ensemble des arcs U ainsi que l'application d définissant les capacités sur ces derniers sont définis de la manière suivante :

- Pour chaque sommet $e_i (i=1, \dots, |ES|)$ nous associons un arc (s, e_i) de capacité égale à $|es_i|$ (nombre de séances que représente le sommet e_i).
- Pour chaque sommet $L_{jk} (j=1, \dots, |TS| \text{ et } k=1, \dots, |P|)$ nous associons un arc (L_{jk}, p) de capacité égale au nombre de salles de type ts_j disponibles à la $k^{\text{ème}}$ période.
- Chaque sommet $e_i (i=1, \dots, |ES|)$ est relié à un sommet $L_{jk} (j=1, \dots, |TS| \text{ et } k=1, \dots, |P|)$ par un arc de capacité égale à l'unité si les trois conditions suivantes sont vérifiées :
 1. Les séances de es_i sont compatibles avec le type de salles ts_j .
 2. Le groupe d'enseignement correspondant à es_i et au moins une salle du type ts_j sont disponibles à la $k^{\text{ème}}$ période.
 3. Il n'y a pas d'autres raisons jugées assez suffisantes pour interdire le déroulement de toute séance de es_i dans un type de salle ts_j .
- Un arc de retour (p, s) de capacité infinie $(+\infty)$. Cet arc ne sera plus représenté sur R .

Dans ce qui suit, si un flot ϕ est calculé sur R , la quantité du flux sur l'arc de retour (p, s) sera appelée la valeur du flot ϕ de s à p .

Nous cherchons, par cette modélisation, à trouver un flot maximum de s à p de valeur égale à $|S|$. Cette condition est *nécessaire* pour l'existence d'un emploi du temps car elle assure l'affectation de chaque séance considérée à une période et à une salle données (satisfaction du premier groupe de contraintes du problème). Il aussi possible de vérifier la condition *nécessaire* suivante pour l'existence d'un emploi du temps : la somme des capacités des arcs (s, e_i) doit être *inférieure ou égale* à la somme des capacités des arcs (L_{jk}, p) .

Après le calcul du flot maximum en question, chaque arc du type (e_i, L_{jk}) ($i=1, \dots, /ES/, j=1, \dots, /TS/, k=1, \dots, /P/$) saturé par ce dernier, définit l'affectation d'une séance de es_i à une salle du type ts_j à la k ième période, ce qui définit complètement un emploi du temps pour tous les groupes d'enseignement.

Dans ce suit , nous ne ferons pas de distinction entre l'ensemble des séances es_i et le sommet e_i qu'il représente ($i \in \{1, \dots, /ES/\}$).

Nous donnons, ci-dessous, un exemple très simple d'une instance du problème considéré ainsi que le modèle qui lui correspond :

Nous nous intéressons à l'affectation des séances de deux (2) groupes d'enseignement à un ensemble de salles dont le regroupement fait ressortir deux (2) ensembles de séances de deux (2) séances chacun et deux (2) types de salles. A chaque période, deux (2) salles de chaque type sont disponibles.

Le problème consiste à établir un emploi du temps pour ces groupes d'enseignement sur une période de référence composée de deux (2) périodes élémentaires. Le reste des contraintes du problème peuvent se déduire du modèle représenté ci-dessous.

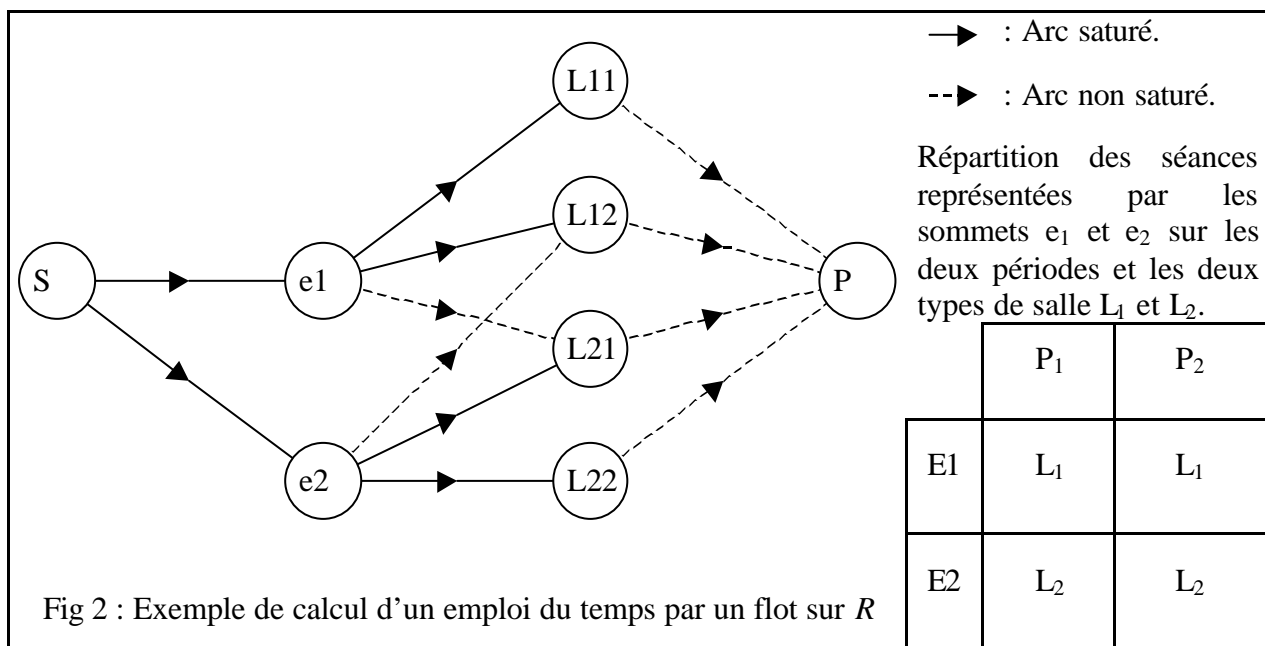


Fig 2 : Exemple de calcul d'un emploi du temps par un flot sur R

III.4. Caractéristiques du modèle proposé

III.4.1. Propriétés d'un emploi du temps donné par le modèle

L'emploi du temps résultant d'un flot maximum ϕ de s à p de valeur $|S|$ calculé sur le réseau R défini ci-dessus garantit la satisfaction des contraintes suivantes :

- (1) Affectation de toutes les séances : Ceci est assuré par la valeur du flot ϕ en question.
- (2) Compatibilité entre types de séances et types de salles : Ceci est garanti par l'existence (ou non existence) des arcs du type (e_i, L_{jk}) ($i=1, \dots, |ES|, j=1, \dots, |TS|, k=1, \dots, |P|$).
- (3) A une salle et à une période donnée est affectée, au plus, une unique séance : Ceci est vérifié grâce aux capacités sur les arcs du type (L_{jk}, p) ($j=1, \dots, |TS|, k=1, \dots, |P|$) qui limitent le nombre maximum de séances affectées aux salles du type ts_j au nombre de salles de ce type disponibles à la k ième période.
- (4) Une séance s ne doit être affectée à une salle l et à une période p_k que si s et l sont disponibles à cette période : Ceci est garanti par les propriétés citées en (2) et (3).
- (5) Satisfaction de certaines contraintes de pré-affectation (des séances aux salles et aux périodes) : si nous voulons affecter, au préalable, une séance s de e_{s_i} ($i \in \{1, \dots, |ES|\}$) à une salle du type ts_j ($j \in \{1, \dots, |TS|\}$) à une période p_k ($k \in \{1, \dots, |P|\}$), il suffit de réduire d'une unité la capacité sur les arcs (s, e_i) et (L_{jk}, p) et supprimer l'arc (e_i, L_{jk}) (nous épuisons les ressources liées à cette affectation). Cette affectation ne sera plus considérée dans le problème à résoudre mais figurera sur l'emploi du temps final.

L'unique groupe de contraintes qui *n'est pas forcément satisfait* par un tel emploi du temps est le suivant :

- (6) Deux séances d'un même groupe d'enseignement ne peuvent se dérouler en même temps.

Il faudra, par conséquent, chercher parmi tous les flots maximums de s à p de valeurs $|S|$ celui qui satisfait le « mieux possible » ce type de contraintes.

Notons ici une caractéristique assez importante d'une telle solution ; un emploi du temps donné par ce modèle consiste à dire qu' *une des séances d'un certain groupe* doit être affectée à *une des salles d'un certain type* à une période donnée, et non pas : *une certaine séance* doit être affectée à *une certaine salle* à une période donnée.

Si nous reprenons l'exemple de la figure 2, nous n'imposons pas, par exemple, que la première séance de es_1 soit affectée à la deuxième salle du type ts_1 à la deuxième période, bien que cela soit une possibilité, mais nous affirmons, tout simplement et d'une manière plus générale, qu'une des deux séances de es_1 doit être affectée à une des deux salles du type ts_1 à la première période et l'autre séance de es_1 doit être affectée à l'une des deux salles du type ts_1 à la deuxième période.

Il est clair, donc, qu'une solution donnée par ce modèle peut représenter en fait plusieurs emplois du temps, tous de « même qualité » et satisfaisant les mêmes types de contraintes. Par ailleurs, s'intéresser à l'affectation des séances et des salles prises individuellement sur la période considérée peut s'avérer, en pratique, non intéressant car la taille du problème risque d'augmenter d'une part et d'autre part, aller dans le « détails » de ces affectations ne présente pas beaucoup d'intérêts et ne donne aucun choix au planificateur puisqu'il n'aura, comme résultat, qu'un unique emploi du temps sans aucune « marge de manœuvre » lui permettant de l'ajuster à sa convenance.

Notons aussi que cette propriété reste très intéressante lorsque le 6^{ème} groupe de contraintes n'est pas satisfait par un emploi du temps donné, puisque nous donnons au planificateur le choix des séances qu'il veut affecter sans conflits parmi celles dont le déroulement simultané viole ce type de contraintes.

III.4.2. Nombre de conflits caractérisant un emploi du temps

Afin d'évaluer la qualité de l'emploi du temps calculé grâce au modèle proposé, nous définissons ici une quantité nbc que nous appellerons *nombre de conflits* qui permet de mesurer le degré de non satisfaction du groupe de contraintes non assuré par un tel emploi du temps (6^{ème} groupe de contraintes).

Nous avons regroupé les séances de chaque groupe d'enseignement ge_1 en un ou plusieurs groupes de séances. Nous définissons le nombre de conflits nbc_1 par rapport à ce groupe d'enseignement comme étant la somme des deux quantités suivantes :

nbc_1^1 : Le nombre de conflits résultant du fait que deux séances d'un même groupe de séances ne peuvent se dérouler en même temps.

nbc_1^2 : Le nombre de conflits résultant du fait que deux séances appartenant à des groupes de séances différents ne peuvent se dérouler en même temps.

Le calcul de ces deux quantités se fait de la manière suivante :

a) $nbcl^1$:

Soit es_i ($i \in \{1, \dots, /ES/\}$) un groupe de séances relatif à ge_l .

Si n ($n \geq 1$) est le nombre de séances de es_i qui se déroulent à la même période p_k ($k \in \{1, \dots, /P/\}$), alors le nombre de conflits relatifs à es_i (ou à e_i) à cette période est $(n-1)$ (ce qui signifie qu'il n'y a aucun conflit sur la période p_k lorsqu'une unique séance s'y déroule). Si n est nul ($n=0$), le nombre de conflits en question sera considéré comme étant nul.

Le nombre de conflits relatifs à es_i (ou e_i) est la somme des conflits par rapport à es_i sur toutes les périodes.

Le nombre $nbcl^1$ sera la somme des nombres de conflits relatifs à tous les groupes de séances de ge_l .

b) $nbcl^2$:

A une période p_k ($k \in \{1, \dots, /P/\}$), un conflit supplémentaire est comptabilisé à chaque fois qu'il y a au moins une séance s_{i1} ($\in es_{i1}$) et au moins une séance s_{i2} ($\in es_{i2}$) ($i1 \neq i2$) qui se déroulent à cette période pour chaque pair $\{es_{i1}, es_{i2}\}$ de groupes de séances relatifs à ge_l .

$nbcl^2$ sera la somme de ces conflits sur toutes les périodes.

Ainsi : $nbcl = nbcl^1 + nbcl^2$

Et le nombre de conflits $nbcl$ sur tout l'emploi du temps sera

$$nbcl = \sum_{l=1}^{|GE|} nbcl_l \left(= \sum_{l=1}^{|GE|} nbcl_l^1 + \sum_{l=1}^{|GE|} nbcl_l^2 \right)$$

Nous posons :

$$nbcl_1 = \sum_{l=1}^{|GE|} nbcl_l^1$$

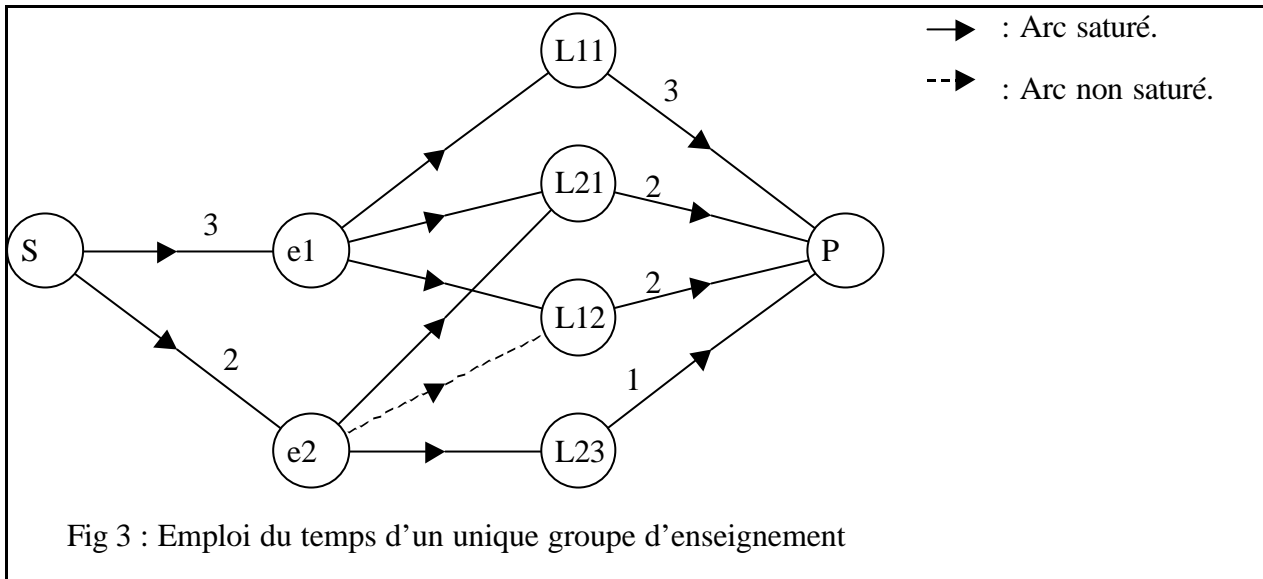
$$nbcl_2 = \sum_{l=1}^{|GE|} nbcl_l^2$$

Dans ce qui suit, nous désignerons les quantités $nbcl_1$ et $nbcl_2$ par :

$nbcl_1$: Le nombre de conflits relatifs aux mêmes groupes de séances ou encore le nombre de conflits relatifs aux sommets e_i .

$nbcl_2$: Le nombre de conflits relatifs à différents groupes de séances.

A titre d'exemple, considérons l'emploi du temps schématisé sur la figure suivante :



Le groupement des séances d'un unique groupe d'enseignement ge_1 fait ressortir deux groupes de séances représentés par e_1 et e_2 . Or, deux séances de e_1 se déroulent en même temps à la première période et la troisième séance se déroule à la deuxième période, ainsi, il existe *un unique conflit par rapport aux séances de e_1* .

De même, les deux séances de e_2 se déroulent dans deux périodes différentes, ainsi, *il n'y a aucun conflit par rapport aux séances de e_2* .

Il s'ensuit que $nb_{c_1}^1 = 1 + 0 = 1$.

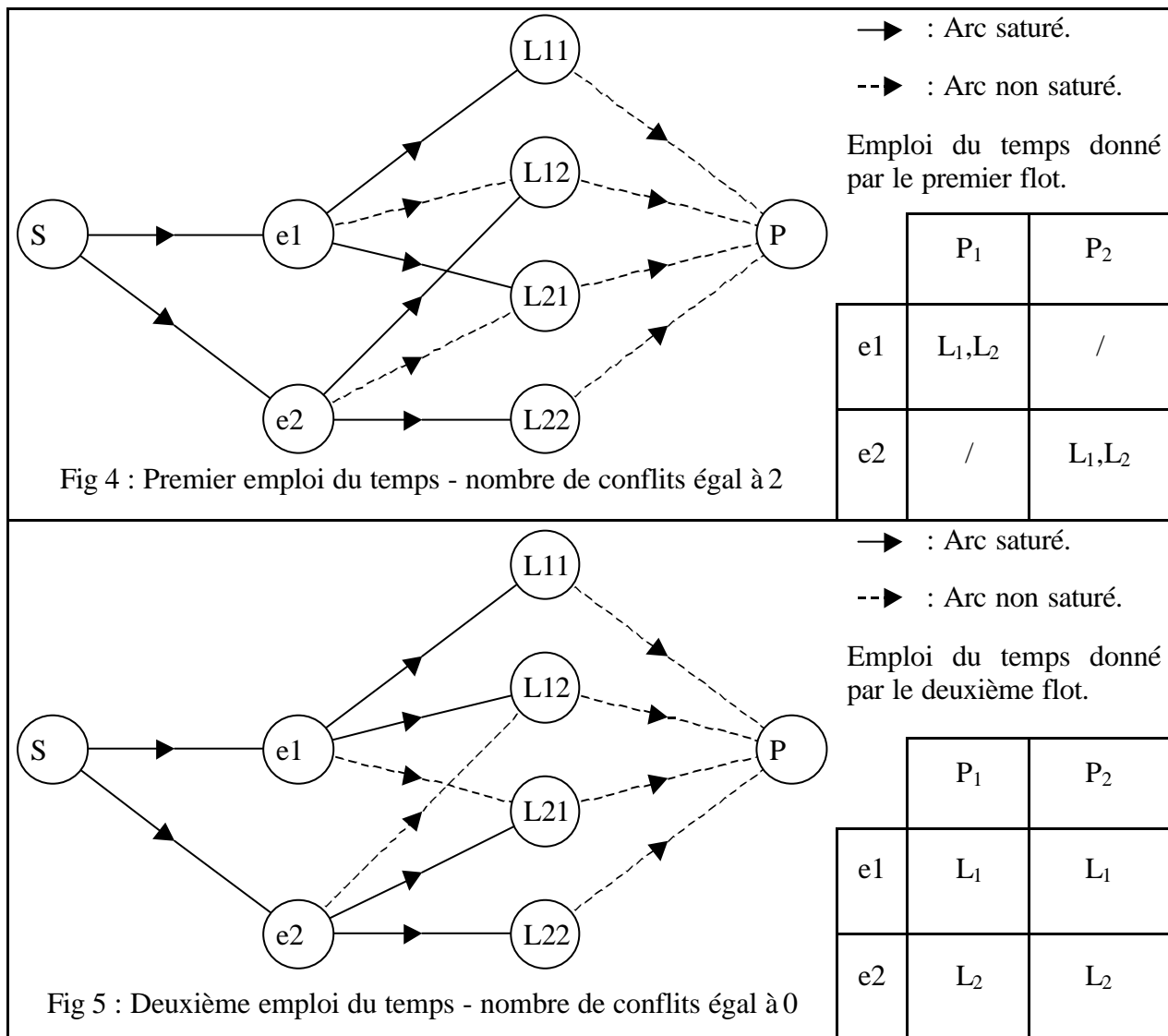
Par ailleurs, puisque e_1 et e_2 concernent le même groupe d'enseignement, il existe un conflit supplémentaire dû au fait qu'il y a une séance de e_2 et deux séances de e_1 qui se déroulent simultanément à la première période, ainsi $nb_{c_1}^2 = 1$.

Le nombre de conflits par rapport à ce groupe d'enseignement est donc $nb_{c_1} = 1+1 = 2$.

III.4.3. Existence d'un ou de plusieurs emplois du temps

Après modélisation du problème, il se peut qu'il existe plusieurs flots maximums de s à p de valeur $|S|$, ce qui signifie l'existence de plusieurs emplois du temps satisfaisant forcément les groupes de contraintes (1), (2), (3), (4) et (5) cités ci-dessus. L'idéal serait donc de choisir parmi ces derniers celui qui satisfait le « mieux possible » le groupe de contraintes (6), c'est à dire l'emploi du temps sur lequel le nombre de conflits est le plus réduit.

A titre d'exemple, reprenons le problème modélisé par le réseau de la figure 2. Nous pouvons aisément constater qu'il est possible de calculer sur ce dernier deux flots maximums (représentés sur les figures 4 et 5) de s à p de valeurs égales au nombre de séances à affecter, nous pouvons donc en déduire deux emplois du temps qui vérifient les cinq (5) premiers groupes de contraintes mais dont les nombres de conflits sont différents. Le meilleur étant celui avec un nombre de conflits nul (celui de la figure 5).



Cela dit, un emploi du temps peut évidemment ne pas exister, ceci peut être déduit du fait que la valeur du flot maximum n'est pas égale au nombre de séances à affecter ($< /S/$), c'est à dire que le groupe de contraintes (1) n'est pas satisfait (les groupes de contraintes (2), (3), (4) et (5) sont garantis par la structure même du réseau). Cela veut dire que même avec un nombre de conflits non nul (le déroulement simultané des séances d'un même groupe d'enseignement existe), nous ne pouvons affecter toutes les séances. Cela sera dû à une ou aux deux raisons suivantes :

1. Insuffisance dans le nombre, la capacité ou l'aménagement des locaux.
2. Période de référence trop courte.

III.4.4. Taille du modèle

Nous donnons ici une évaluation de la taille du réseau R après l'opération de groupement des séances et des salles considérées.

Soient $nb(e_i)$, $nb(L_{jk})$ et $nb(arcs)$ le nombre de sommets e_i ($i=1, \dots, /ES/$), le nombre de sommets L_{jk} ($j=1, \dots, /TS/$, $k=1, \dots, /P/$) et le nombre d'arcs du réseau R respectivement.

$Nb(e_i)$: représente le nombre d'ensembles de séances qui peuvent se dérouler dans un même type de salle et chacun de ces ensembles est relatif à un unique groupe d'enseignement. Ce nombre variera, donc, entre le nombre de groupes d'enseignement $/GE/$ (borne inférieure : lorsque toutes les séances de chaque groupe d'enseignement peuvent être regroupées dans un unique ensemble) et le nombre total de séances à affecter $/S/$ (borne supérieure : lorsque chaque ensemble regroupe une unique séance).

$Nb(L_{jk})$: Il est égal à $/P/$ fois le nombre de types de salles. Cette quantité est donc fonction du nombre de types de salles obtenu après l'opération de groupement, ce dernier varie entre 1 (borne inférieure : lorsque toutes les salles peuvent être regroupées en un unique ensemble ou type, ce qui signifie que toute séance considérée peut se dérouler dans n'importe quelle salle) et $|L|$ (borne supérieure : lorsque chaque type de salles regroupe une unique salle).

$Nb(arcs)$: Ce nombre dépend du degré de compatibilité entre la nature des séances et les types de salles existantes ainsi que des nombres $nb(e_i)$ et $nb(L_{jk})$. Il varie entre $[nb(e_i)+nb(L_{jk})]$ (borne inférieure : lorsque chaque séance ne peut être affectée à aucune salle, et dans ce cas il n'existe évidemment pas de solutions) et $[nb(e_i)+nb(L_{jk})]+[nb(e_i)\times nb(L_{jk})]$ (borne supérieure : lorsque n'importe quelle séance peut être affectée à n'importe quelle salle. En réalité, si une bonne modélisation est faite, ce cas de figure ne peut se présenter que s'il y a un unique type de salles ($/TS/=1$, $nb(L_{jk})=/P/$) puisque toutes les séances sont compatibles avec tous les types de salles, ces derniers peuvent être, donc, regroupés en un seul type).

Après modélisation, le réseau R obtenu aura :

Un nombre de sommets égal à $nb(e_i)+nb(L_{jk})+2$.

Au plus, un nombre d'arcs égal à $[nb(e_i)+nb(L_{jk})]+[nb(e_i)\times nb(L_{jk})]$.

Puisque :

$$\begin{cases} nb(e_i) \leq |S| & \text{Nombre total de séances} \\ nb(L_{jk}) \leq |P||L| & \text{Nombre de périodes} \times \text{Nombre de salles} \end{cases}$$

Et les bornes supérieures de ces quantités peuvent être atteintes, alors le réseau R aura *au plus* :

$$\begin{cases} N = |S| + |P||L| + 2 & \text{Sommets} \\ M = |S||P||L| + |S| + |P||L| & \text{Arcs} \end{cases}$$

Bien entendu, ces bornes ne peuvent être atteintes que pour des cas très rares en réalité ou bien être le résultat d'un mauvais groupement des séances et des salles.

III.5. Adaptation du modèle à d'autres problèmes similaires

Le modèle adopté permettant la construction d'un emploi du temps pour le problème considéré, peut très bien être adapté à plusieurs autres problèmes similaires. Nous donnons ici quelques uns d'entre eux ainsi que la manière avec laquelle le modèle proposé peut être adapté pour les résoudre.

III.5.1. Construction d'un calendrier d'examens

Soit GE l'ensemble des groupes d'enseignement devant subir un ensemble d'examens EX qui doivent se dérouler dans un ensemble L de salles sur une période de référence P .

Comme déjà souligné, il y a une certaine similarité entre ce problème et le problème d'emploi du temps considéré. En effet, il suffit de considérer l'ensemble des examens EX au lieu de l'ensemble des séances S , ainsi, l'opération de groupement se fera suivant la nature des examens et les types de salles existantes, et le nombre de conflits concernera les examens d'un même groupe d'enseignement qui ne peuvent se dérouler en même temps.

Tout flot maximum de s à p de valeur $|EX|$ sur le réseau ainsi construit affecte tous les examens et engendrera un planning respectant les groupes de contraintes suivants :

- (1) Affectation de tous les examens.
- (2) Compatibilité entre types d'examens et types de salles.
- (3) A une salle et à une période données est affecté un unique examen.
- (4) Disponibilité des salles.
- (5) Satisfaction de certaines contraintes de pré-affectation (des examens aux salles et aux périodes).

III.5.2. Affectation des enseignants aux groupes d'enseignement

Reprenons le problème cité au (Chapitre 2) qui consiste à établir un calendrier de rencontres entre les groupes d'enseignement GE et les enseignants E sur $|P|$ périodes élémentaires. Chaque groupe d'enseignement ge_i ($\in GE$, $i=1, \dots, |GE|$) doit rencontrer l'enseignant E_j ($\in E$, $j=1, \dots, |E|$) durant r_{ij} périodes élémentaires.

Notons ici que ce problème est d'une toute autre nature puisque nous ne considérons plus un ensemble de salles mais un ensemble d'enseignants qui ne peuvent être regroupé suivant un critère donné (par exemple : enseignants pouvant prendre en charge le même ensemble de cours) vu qu'il existe une contrainte forte supplémentaire, et très souvent exigée, qui restreint l'enseignement de toutes les occurrences d'un cours à un même enseignant pour un groupe d'enseignement donné.

Afin d'adapter ce problème au modèle proposé, nous adoptons les considérations suivantes :

1. Chaque ensemble de séances es_l ($\in ES$) implique un unique groupe d'enseignement ge_i ($\in GE$) et un unique enseignant E_j ($\in E$) et regroupe les r_{ij} séances les concernant. Si r_{ij} est nul, l'ensemble de séances es_l n'existe pas et le sommet e_l ne sera pas représenté.
2. Un sommet L_{jk} ($j \in \{1, \dots, |E|\}$, $k \in \{1, \dots, |P|\}$) représentera un unique enseignant E_j à la k ième période. Si l'enseignant E_j n'est pas disponible à la k ième période, le sommet L_{jk} ne sera pas représenté.
3. Il y a un arc de la source s à tout sommet e_i ($i \in \{1, \dots, |ES|\}$) et de tout sommet L_{jk} ($j \in \{1, \dots, |E|\}$, $k \in \{1, \dots, |P|\}$) au puits p .
4. Il y a des arcs d'un sommet e_i ($i \in \{1, \dots, |ES|\}$) vers tous les sommets L_{jk} ($k \in \{1, \dots, |P|\}$) représentant l'enseignant E_j qui doit assurer les séances de es_i .
5. Un arc (s, e_i) ($i \in \{1, \dots, |ES|\}$) aura une capacité égale à $|es_i|$ et les autres arcs auront des capacités égales à l'unité.

Ainsi, tout flot de s à p de valeur $\sum_{i=1}^{|ES|} |es_i| \left(= \sum_{i=1}^{|GE|} \sum_{j=1}^{|E|} r_{ij} \right)$ engendre un emploi du temps

vérifiant les groupes de contraintes suivants :

- (1) Toutes les rencontres sont effectuées.
- (2) Chaque séance est affectée à l'enseignant prévu pour l'assurer.
- (3) A un enseignant et à une période donnée est affectée une unique séance.
- (4) Disponibilité des enseignants.
- (5) Satisfaction de certaines contraintes de pré-affectation (des séances aux enseignants et aux périodes).

Comme déjà souligné, ce modèle ne permet pas forcément, par un tel flot, d'assurer que deux séances d'un même groupe d'enseignement ne peuvent se dérouler en même temps.

Il faudra donc, parmi tous les flots maximums de s à p ayant une valeur $\sum_{i=1}^{|ES|} |es_i|$, retenir celui qui minimise le nombre de conflits.

III.5.3. Catégorie de problèmes auxquels le modèle peut être adapté

Il est clair que nous ne pourrions citer ici tous les problèmes pouvant être modélisés de cette manière, cela dit, nous pouvons aisément constater la catégorie de problèmes auxquels nous pouvons adapter ce type de modèle.

Généralement, ces problèmes consistent à affecter un certain nombre d'événements EV à certaines ressources RC sur une période de référence P en respectant certaines contraintes de *compatibilité* (entre les événements et les ressources), de *disponibilité* (des ressources), de *pré-affectation* (des événements aux ressources et aux périodes), ainsi que d'autres contraintes du type « les événements x et y ne peuvent se dérouler en même temps » et « A une ressource r à une période donnée, nous pouvons affecter au plus, un nombre n d'événements ».

L'ensemble des événements et des ressources doivent pouvoir être regroupés de sorte que :

- Chaque groupe d'événements ne doit contenir que des événements qui ne peuvent se dérouler en même temps.
- Si un événement peut être affecté à une ressource d'un certain groupe, alors cet événement peut être affecté à n'importe quelle ressource du même groupe.
- Si un événement d'un certain groupe peut être affecté à une certaine ressource, alors tout événement du même groupe peut être affecté à cette même ressource.

Chapitre 4

Méthode de résolution du problème

IV.1. Introduction

Après avoir décrit et modélisé le problème considéré, nous présentons ici une méthode heuristique pour sa résolution basée sur un procédé itératif permettant plusieurs passages d'un emploi du temps à un autre, en retenant à la fin le meilleur d'entre eux. L'apport essentiel de l'heuristique proposée réside dans la manière avec laquelle se fait le passage d'une solution à une autre et qui permet de faire des déplacements efficaces dans l'espace des solutions considéré.

Aussi, dans ce qui suit, nous adopterons les considérations suivantes :

- Puisqu'un sommet e_i ($i \in \{1, \dots, /ES/\}$) (respectivement L_{jk} , $j \in \{1, \dots, /TS/\}$ et $k \in \{1, \dots, /P/\}$) représente l'ensemble des séances e_{s_i} (respectivement les salles du type ts_j à la k ième période), nous ne ferons pas de distinction entre ces derniers.
- Un sommet du type e_i (respectivement L_{jk}) désigne un sommet e_i ($i \in \{1, \dots, /ES/\}$) (respectivement L_{jk} ($j \in \{1, \dots, /TS/\}$, $k \in \{1, \dots, /P/\}$)).
- Un arc du type (s, e_i) (respectivement (L_{jk}, p)) désigne un arc (s, e_i) ($i \in \{1, \dots, /ES/\}$) (respectivement (L_{jk}, p) ($j \in \{1, \dots, /TS/\}$, $k \in \{1, \dots, /P/\}$)).
- $\{e_i\}$ (respectivement $\{L_{jk}\}$) désigne l'ensemble $\{e_i / i \in \{1, \dots, /ES/\}\}$ (respectivement $\{L_{jk} / j \in \{1, \dots, /TS/\}, k \in \{1, \dots, /P/\}\}$).
- Affecter une séance à L_{jk} ($j \in \{1, \dots, /TS/\}$, $k \in \{1, \dots, /P/\}$) désigne le fait d'affecter cette séance à une salle du type ts_j à la k ième période.
- Le « déplacement d'une séance de L_{jk} » ($j \in \{1, \dots, /TS/\}$, $k \in \{1, \dots, /P/\}$) désigne le déplacement d'une séance déjà affectée à une salle du type ts_j à la k ième période à une autre salle $ts_{j'}$ à la k' ième période ($j' \in \{1, \dots, /TS/\}$, $k' \in \{1, \dots, /P/\}$) avec $(j, k) \neq (j', k')$.

IV.2. Génération d'un emploi du temps à partir d'un autre

Ayant trouvé un emploi du temps initial sous forme d'un flot maximum de s à p sur le réseau R de valeur égale au nombre de séances à effectuer (qui est une condition nécessaire pour l'existence d'une solution au problème), la procédure générale consiste à générer un autre flot de même valeur en essayant de réduire le plus possible le nombre de conflits sur les séances relatives aux sommets du type e_i .

La transition d'un tel flot à un autre se fera après avoir défini un autre réseau R^* (que nous appellerons *graphe de transition*), un jeu de poids sur les arcs de ce dernier ainsi qu'un algorithme permettant de retrouver le nouveau flot. Par l'algorithme en question, nous essayerons de déplacer une séance qui se déroule dans un type de salle à une période donnée et faire des modifications sur certaines autres affectations afin de réduire le nombre de conflits sur ces dernières.

IV.2.1. Définition du graphe de transition

Comme déjà souligné, ce graphe permettra la transition d'un flot maximum φ sur R à un autre φ' (donc, d'un emploi du temps à un autre). La structure de ce graphe dépendra de celle du réseau R ainsi que du flot φ .

Nous définissons le *graphe de transition* $R^*(\varphi)$ comme suit :

1. L'ensemble des sommets de $R^*(\varphi)$ est constitué de tous les sommets de R sauf le sommet s .
2. Les arcs du type (s, e_i) ($i \in \{1, \dots, /ES/\}$) ainsi que l'arc de retour (p, s) n'existent pas.
3. Si un arc du type (e_i, L_{jk}) ($j \in \{1, \dots, /TS/\}$, $k \in \{1, \dots, /P/\}$) est saturé par φ dans R il sera remplacé par l'arc (L_{jk}, e_i) , sinon, il sera représenté comme il est.
4. Tout arc du type (L_{jk}, p) non saturé par le flot φ dans R sera représenté dans $R^*(\varphi)$, sinon, il sera supprimé.

Remarque

1. Dans ce qui suit, lorsque nous décrivons un chemin de $R^*(\varphi)$, le passage d'un arc du type (L_{jk}, e_i) *supprime l'affectation* d'une séance de e_i à L_{jk} et lorsque nous décrivons un arc du type (e_i, L_{jk}) , nous affectons une séance de e_i à L_{jk} .
2. Vu la structure spécifique des réseaux R et $R^*(\varphi)$, un chemin de L_{jk} à p dans $R^*(\varphi)$ n'est constitué que de séquences d'arcs du type (L_{jk}, e_i) et (e_i, L_{jk}) sauf éventuellement le dernier (dont l'extrémité finale peut être le sommet p), car le graphe engendré par les sommets $\{e_i\} \cup \{L_{jk}\}$ est biparti.

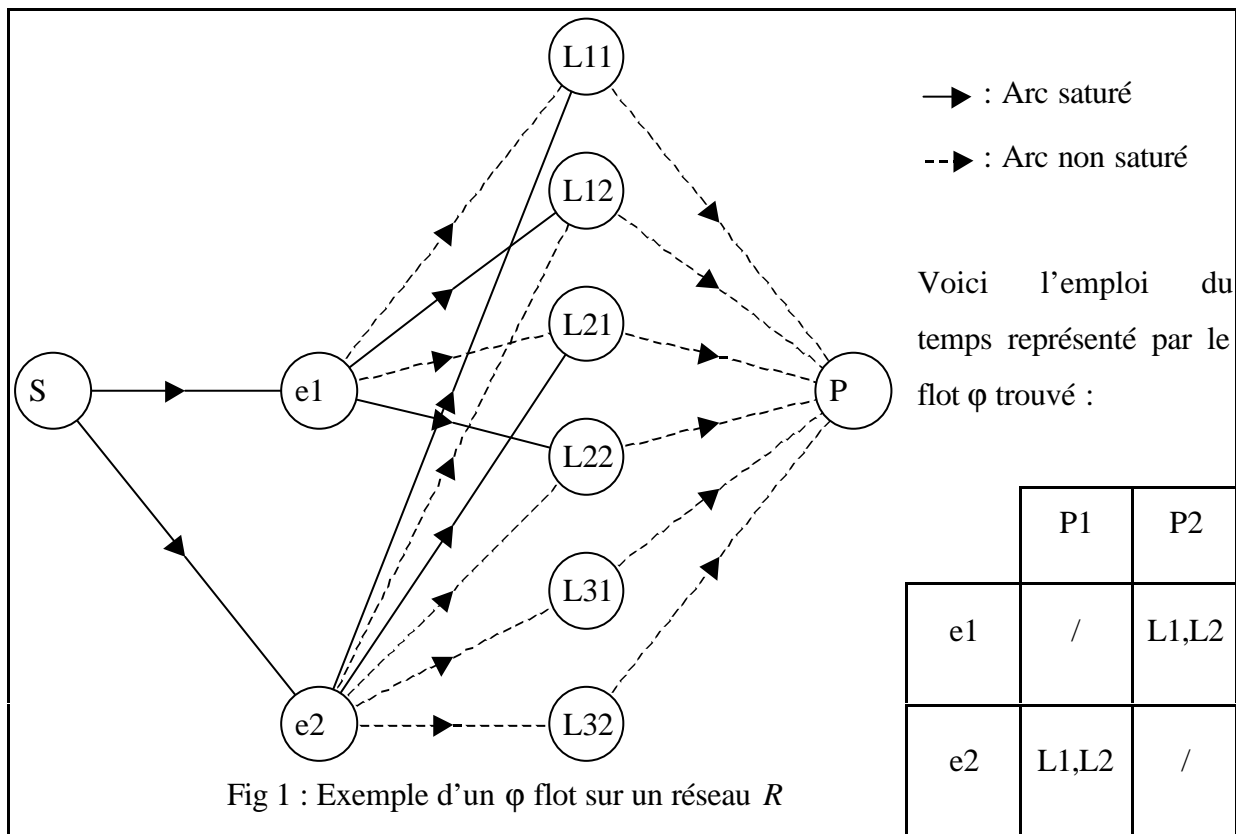
IV.2.2. Génération à partir d'un type de chemin du graphe de transition

Si nous voulons déplacer une séance qui se déroule dans un type de salle à une période représentés par un sommet L_{jk} , la valeur du flot φ peut être diminuée d'une unité de sorte que le flux sur l'arc (L_{jk}, p) soit également réduit d'une unité et puis l'augmenter à travers un autre chemin de R sans passer par l'arc (L_{jk}, p) .

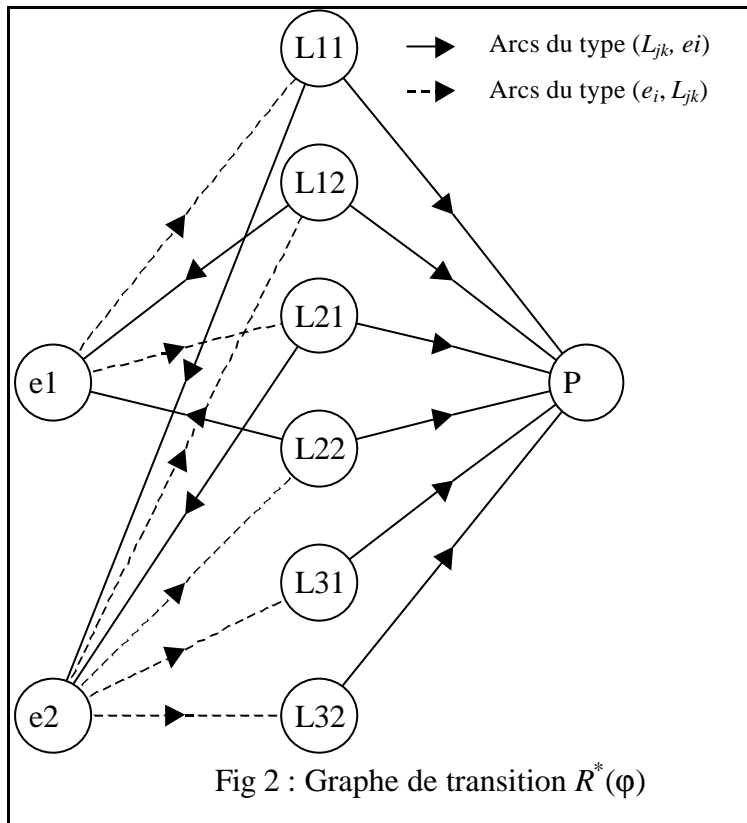
Ceci revient à chercher un chemin élémentaire de L_{jk} à p dans $R^*(\varphi)$ sans passer par l'arc (L_{jk}, p) (c'est à dire, après l'avoir supprimé de $R^*(\varphi)$).

De cette manière, nous engendrons un nouveau flot φ' de même valeur que φ , ce qui génère donc, un nouvel emploi du temps satisfaisant les mêmes contraintes satisfaites par l'emploi du temps relatif à φ .

Considérons à présent, l'exemple suivant, où nous avons deux (2) ensembles de séances de deux (2) groupes d'enseignement, trois (3) types de salles et deux (2) périodes :



Nous constatons bien, ici, qu'il y a un conflit sur les séances de e_1 et un autre sur celles de e_2 (les deux (2) séances de e_1 se déroulent à la deuxième période et les deux (2) séances de e_2 se déroulent à la première période). Le graphe transition $R^*(\varphi)$ associé à la figure ci-dessus est le suivant (Fig 2) :



Si nous choisissons de déplacer une séance de L_{21} , il suffit de chercher un chemin de L_{21} à p dans $R^*(\varphi)$ sans passer par l'arc (L_{21}, p) et de réduire la valeur du flot sur l'arc (L_{21}, p) d'une unité dans R .

Or, sur le réseau $R^*(\varphi)$, nous pouvons considérer plusieurs chemins de L_{21} à p .

Examinons, grâce au tableau ci-dessous, quelques uns d'entre eux :

Chemins	Emplois du temps générés			Nombre de conflits par rapport aux sommets e_i	Nombre de conflits réduits par rapport à l'emploi du temps initial (de la fig(1))
		P1	P2		
$\{L_{21}, e_2, L_{31}, p\}$				2	0
	e1	/	L_1, L_2		
	e2	L_1, L_3	/		
$\{L_{21}, e_2, L_{22}, p\}$				1	1
	e1	/	L_1, L_2		
	e2	L_1	L_2		
$\{L_{21}, e_2, L_{22}, e_1, L_{11}, p\}$				0	2
	e1	L_1	L_1		
	e2	L_1	L_2		

Tableau 1 : Caractéristiques de 3 emplois du temps générés à partir de 3 chemins différents

Il est clair, d'après le tableau précédent, que la qualité de l'emploi du temps (en terme du nombre de conflits) généré à partir d'un sommet L_{jk} dépend étroitement du chemin choisi de ce sommet jusqu'à p ; sur l'exemple ci-dessus, le meilleur chemin est $\{L_{21}, e_2, L_{22}, e_1, L_{11}, p\}$ qui génère un emploi du temps sans conflits et réduit, donc, le nombre de ces derniers de 2 unités par rapport à l'emploi du temps initial.

L'idéal est donc de choisir un chemin de L_{jk} à p qui réduit le plus possible le nombre de conflits sur les séances, par conséquent, le fait de pouvoir associer à chaque chemin de ce type un poids reflétant le nombre de conflits réduits ne peut qu'être bénéfique pour la recherche du meilleur d'entre eux.

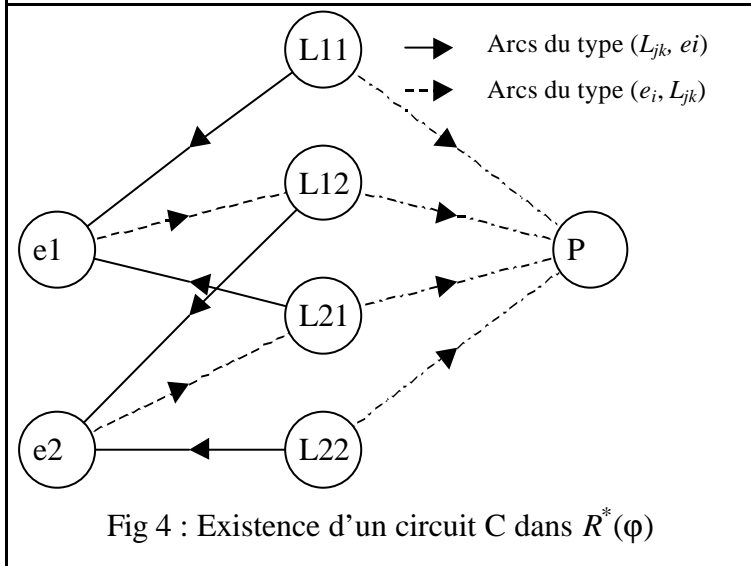
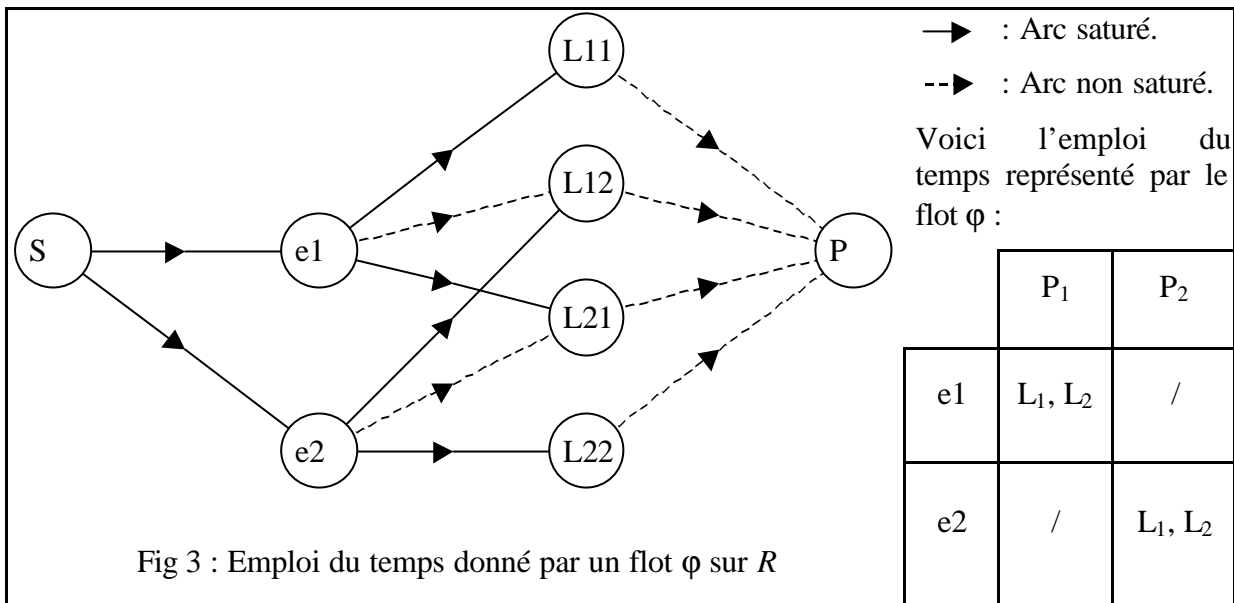
IV.2.3. Génération à partir d'un circuit du graphe de transition

Nous nous intéressons, à présent, à la modification d'un emploi du temps (représenté par un flot φ sur R) par un circuit de $R^*(\varphi)$ pour générer un autre flot φ' . Contrairement à la modification apportée par un chemin élémentaire de L_{jk} à p , la modification du flot par un circuit élémentaire de $R^*(\varphi)$ ne se fait que sur les arcs de R qui correspondent aux arcs du circuit en question, ceci garantit la cohérence du nouveau flot φ' retrouvé sur R .

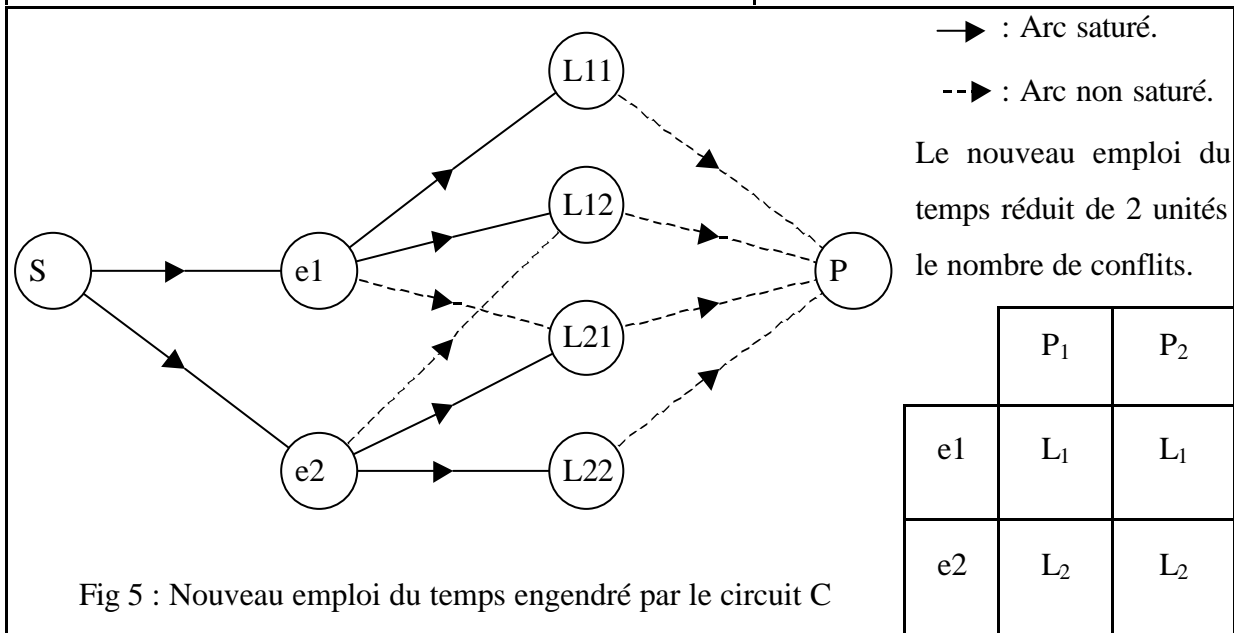
Par ce circuit, le nouveau flot φ' ne modifie en rien le nombre de séances affectées à chaque type de salle et à chaque période, de plus, la valeur de φ' est la même que celle de φ , ce qui veut dire que le nouvel emploi du temps engendré satisfait les mêmes contraintes satisfaites par l'emploi du temps précédent.

Nous donnons, ci-dessous, un exemple illustrant la manière avec laquelle nous pouvons générer un nouvel emploi du temps à partir d'un circuit du graphe de transition $R^*(\varphi)$:

Après calcul d'un flot maximum φ sur le réseau R représenté ci-dessous, nous retrouvons un emploi du temps qui contient deux conflits, le premier à la première période par rapport aux séances de e_1 , et le second à la deuxième période par rapport aux séances de e_2 .



L'existence d'un circuit $C = \{L_{12}, e_2, L_{21}, e_1, L_{12}\}$ permet d'engendrer un nouvel emploi du temps repr sent  par un nouveau flot (fig 5). Il est   noter que ce nouvel emploi du temps ne poss de aucun conflit par rapport aux s ances de e_1 et e_2 .



IV.2.4. Propriétés d'un flot généré à partir du graphe de transition

Dans ce qui suit, nous montrons qu'un flot φ' généré par un chemin de L_{jk} à p ou un circuit du graphe de transition $R^*(\varphi)$, possède les propriétés suivantes :

1. φ' est bien un flot sur R .
2. φ' est à composantes entières : cette caractéristique est très importante, sans elle, l'emploi du temps relatif à φ' n'a aucun sens.
3. φ' est maximum et est de valeur égale au nombre de séances à affecter : ceci garantit que l'emploi du temps généré satisfait les mêmes contraintes satisfaites par l'emploi du temps relatif au flot φ .

A. Rappel

Définition 1 : Flot sur un graphe [25]

Soit $G=(X, U)$ un graphe connexe dont les arcs sont numérotés de 1 à M . Un flot φ dans G est un vecteur à M composantes $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_M)^t \in \mathfrak{R}^M$ tel qu'en tout sommet $i \in X$ de G , la première loi de Kirchhoff est vérifiée, c'est à dire :

$$(1) \quad \sum_{u \in w^+(i)} j_u = \sum_{u \in w^-(i)} j_u, \text{ où}$$

$w^+(i) = \{e \in U / i \text{ est l'extrémité initiale de } e\}$ et $w^-(i) = \{e \in U / i \text{ est l'extrémité terminale de } e\}$.

Pour un arc u de U , la composante φ_u est appelée quantité du flot ou flux sur l'arc u .

La relation (1) exprime simplement que la somme des flux entrants en un sommet i est égale à la somme des flux sortants (loi de conservation aux nœuds).

Définition 2 : Problème du flot maximum sur un réseau [25]

Dans un graphe orienté $G' = (X, U')$ nous distinguerons un arc spécial $u_r=(p, s)$ dit de retour et nous posons $U' = U \cup \{u_r\}$.

Nous définissons, ensuite, une application $C : U' \rightarrow \mathfrak{R}^+$

$$u \rightarrow C_u = C(u) : \text{Capacité de l'arc } u.$$

Le problème du flot maximum de s à p sur le réseau $R'(X, U', C)$ consiste à chercher un vecteur $\varphi^{\text{Max}} \in \mathfrak{R}^m$ ($m = |U'|$) tel que :

1. φ^{Max} est un flot sur le réseau (X, U') .
2. $C_u \geq \varphi_u^{\text{Max}} \geq 0, \forall u \in U'$.
3. $\varphi_{u_r}^{\text{Max}}$ soit maximum sous les contraintes (1) et (2).

Théorème 1 [25]

Si les capacités des arcs sont des nombres entiers, le flot maximum de s à p dans G a toutes ses composantes entières.

B. Démonstration

Le fait que tout flot maximum de s à p sur le réseau de transition soit à composantes entières est une conséquence directe du Théorème 1 puisque la capacité des arcs sur le graphe de transition sont entières.

Montrons, à présent, qu'en se déplaçant d'un flot φ à un autre φ' à partir d'un chemin reliant un sommet $L_{j_0k_0}$ à p ou à partir d'un circuit du graphe de transition :

1. φ' est bien un *flot* sur R .
2. La valeur de φ' est égale à celle de φ .

Remarque

Les applications successives de la procédure permettant de générer de nouveaux flots ne sont faites qu'après le calcul d'un flot maximum initial φ^0 , or ce dernier ne sera accepté que si sa valeur est égale au nombre de séances à affecter (condition nécessaire de l'existence d'une solution au problème considéré), la démonstration du second point (2) ci-dessus garantit donc que l'ensemble des flots générés ont une valeur maximale égale au nombre de séances à affecter.

(i) Si \mathbf{j}' est généré par un chemin de $L_{j_0k_0}$ à p dans $R^*(\mathbf{j})$:

Soit $C = \{L_{j_0k_0}, e_{i_0}, \dots, e_{i_1}, L_{j_1k_1}, p\}$ le chemin choisi pour générer φ' . Vérifions si la première loi de Kirchhoff est satisfaite en chaque sommet de C (puisque le flot ne sera modifié que sur les arcs adjacents aux sommets de ce dernier).

Pour le sommet $L_{j_0k_0}$: le flux sur $(L_{j_0k_0}, p)$ sera diminué d'une unité et le flux sur $(e_{i_0}, L_{j_0k_0})$ sera aussi diminué d'une unité, donc il y a bien conservation des flux en ce sommet.

Pour le sommet p : le flux sur l'arc $(L_{j_0k_0}, p)$ a été diminué d'une unité et celui sur l'arc $(L_{j_1k_1}, p)$ a été augmenté d'une unité, ce qui conserve inchangée la somme des flux entrants à ce sommet.

Pour le sommet $L_{j_1k_1}$: les flux sur les arcs $(e_{i_1}, L_{j_1k_1})$ et $(L_{j_1k_1}, p)$ seront augmentés d'une unité, il y a donc conservation des flux sur le sommet $L_{j_1k_1}$.

Pour le reste des sommet : Sur un sommet e_i (respectivement L_{jk}), par le chemin C , nous diminuons le flux d'une unité sur un arc de $w^+(e_i)$ (respectivement $w^-(L_{jk})$) et nous l'augmentons sur un autre de $w^+(e_i)$ (respectivement $w^-(L_{jk})$).

Ainsi, φ' généré à partir de φ par le chemin C est bien un flot sur le réseau R .

(ii) Si \mathbf{j}' est généré à partir d'un circuit C de $R^*(\mathbf{j})$:

Dans ce cas aussi, le flot n'est changé que sur les arcs correspondant au circuit C de $R^*(\varphi)$ en question.

Pour chaque sommet e_i (respectivement L_{jk}) de C , nous diminuons le flux d'une unité sur un arc de $w^+(e_i)$ (respectivement $w^-(L_{jk})$) et nous l'augmentons sur un autre de $w^+(e_i)$ (respectivement $w^-(L_{jk})$).

Sur le reste des sommets les flux ne changent pas.

Ainsi, φ' généré à partir de φ par un tel circuit C est bien un flot sur le réseau R .

Donc :

1. φ' est bien un *flot* sur R .
2. φ' est de *même valeur* que φ (la somme des flux entrants au sommet p reste inchangée).
3. φ' est à *composantes entières*.

IV.3. Choix des emplois du temps à générer

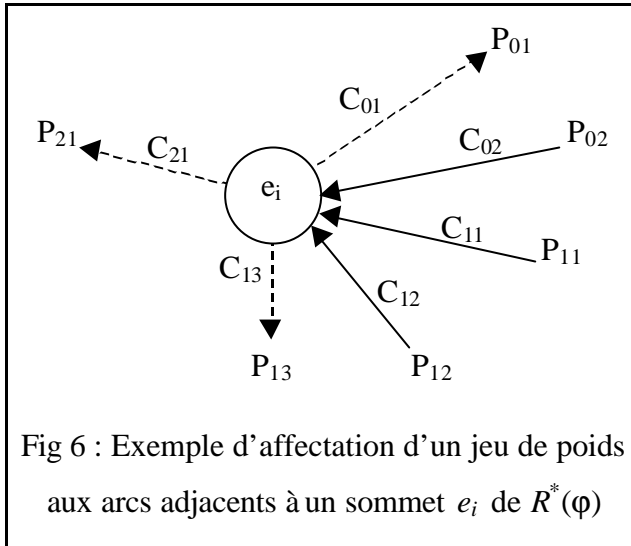
IV.3.1. Définition d'un jeu de poids sur les arcs du graphe de transition

Nous avons montré, dans les étapes précédentes, que nous pouvons modifier un emploi du temps donné par un flot φ sur R soit par un chemin élémentaire de L_{jk} à p ou par un circuit élémentaire du graphe de transition $R^*(\varphi)$. L'idéal est donc de choisir, pour générer de nouveaux emplois du temps, des chemins ou des circuits de ce type qui réduisent le plus possible le nombre de conflits sur les séances, par conséquent, le fait de pouvoir leur associer un poids reflétant le nombre de conflits réduits ne peut qu'être bénéfique pour la recherche du meilleur d'entre eux.

Or, comme déjà souligné, lorsque nous décrivons un tel chemin ou un tel circuit, mis à part éventuellement le dernier arc, toutes les séquences d'arcs parcourues sont du type $\{(L_{j_1k_1}, e_i), (e_i, L_{j_2k_2})\}$, il serait donc intéressant d'associer des poids à ces deux arcs de sorte que *leur somme soit égale au nombre de conflits ajoutés lorsque nous déplaçons une séance de e_i de $L_{j_1k_1}$ à $L_{j_2k_2}$* et d'associer aux arcs du type (L_{jk}, p) un poids nul, ainsi, le poids d'un chemin

élémentaire de L_{jk} à p ou d'un circuit élémentaire de $R^*(\varphi)$ sera égal au nombre de conflits ajoutés, et il suffira de chercher parmi eux celui ayant le poids minimum.

Cela dit, nous montrons par l'exemple qui suit, que la définition d'un tel jeu de poids n'est pas possible :



P_{kj} : Le $j^{\text{ième}}$ arc provenant d'un type de salle correspondant à une période P_k .

C_{kj} : Poids correspondant à l'arc relatif à P_{kj} .

Examinons ce que le jeu de poids $\{C_{kj}\}$ doit vérifier pour satisfaire l'hypothèse faite ci-dessus :

1. Si nous déplaçons une séance de P_{02} à P_{01} (en parcourant le chemin $\{P_{02}, e_i, P_{01}\}$) nous ne faisons que déplacer l'unique séance de e_i se déroulant à la période P_0 d'un type de salle à un autre, le nombre de conflits sur les séances du sommet e_i ne change donc pas, nous devons avoir :

$$C_{01} + C_{02} = 0 \quad (1).$$

2. Si nous déplaçons une séance de P_{11} à P_{01} , nous ne ferons que déplacer un conflit (avant, il y avait deux séances qui se déroulaient en même temps à la période P_1 , après il y aura deux séances à la période P_0 et une à la période P_1), donc, nous devons avoir :

$$C_{11} + C_{01} = 0 \quad (2).$$

3. Si nous déplaçons une séance de P_{02} à P_{13} (il n'y avait pas de conflits sur la période P_0 et à la période P_1 nous rajouterons une troisième séance, et donc un conflit) nous augmenterons, donc, le nombre de conflits d'une unité ; nous devons avoir :

$$C_{02} + C_{13} = 1 \quad (3).$$

4. Si nous déplaçons une séance de P_{11} à P_{13} , nous ne ferons que déplacer un conflit, alors :

$$C_{11} + C_{13} = 0 \quad (4).$$

Donc, une partie des poids de $\{C_{ij}\}$ doit vérifier, le système suivant :

$$\begin{cases} C_{01} + C_{02} = 0 & (1) \\ C_{01} + C_{11} = 0 & (2) \\ C_{02} + C_{13} = 1 & (3) \\ C_{11} + C_{13} = 0 & (4) \end{cases}$$

De (1), (2) et (3) nous trouvons : $C_{01} = -C_{02} = -C_{11} = C_{13}-1 \Rightarrow C_{11}+C_{13} = 1$, ce qui contredit l'équation (4). Par conséquent, un tel jeu de poids ne peut exister.

Cela dit, nous définissons, ci-dessous un jeu de poids sur les arcs de $R^*(\varphi)$ de sorte que la somme des poids des arcs du type (L_{j1k1}, e_i) et (e_i, L_{j2k2}) soit effectivement égale au nombre de conflits ajoutés sur les séances de e_i sauf dans *un seul et unique cas de figure* (qui est forcément le minimum de cas défavorables), puis nous donnerons un algorithme simple qui, au lieu d'associer à un chemin de L_{jk} à p ou un circuit élémentaire son poids classique (la somme des poids des arcs le composant) il lui associera le nombre de conflits réellement ajoutés en le parcourant.

IV.3.1.1. Définition du jeu de poids

Les poids des arcs de $R^*(\varphi)$, sauf ceux du type (L_{jk}, p) qui seront tous nuls, sont définis comme suit : pour chaque sommet e_i et une période P_k , le poids des arcs reliant e_i aux sommets représentant les types de salles correspondants à la période P_k sont les suivants :

Cas n°1

S'il y a un unique arc entrant d'une période P_k alors son poids sera égal à **(-1)** et les autres (arcs sortants) auront pour poids **(2)**.

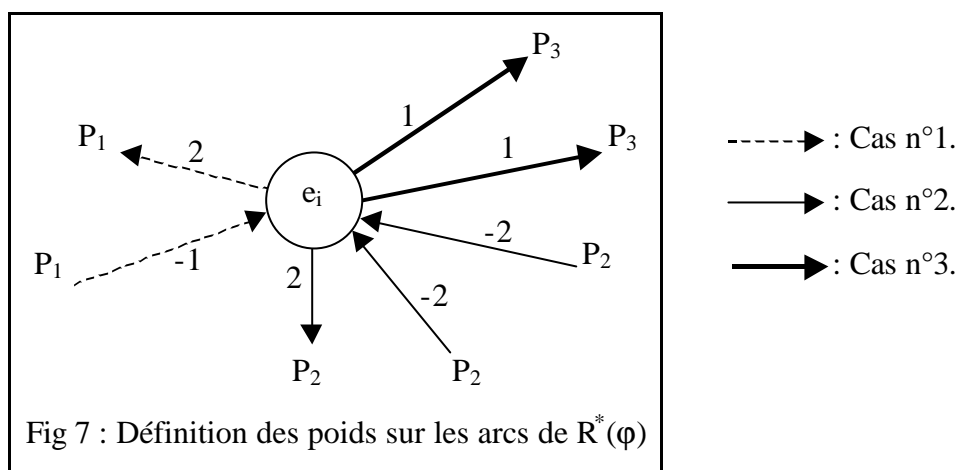
Cas n°2

S'il y a au moins deux arcs entrants d'une période P_k alors les arcs entrants auront pour poids **(-2)** et ceux sortants pour poids **(2)**.

Cas n°3

S'il n'y a que des arcs sortants, alors leur poids sera égal à **(1)**.

Voici une figure illustrant un exemple des trois cas cités ci-dessus :



Nous donnons, ci-dessous, un tableau qui montre que :

1. Les trois cas cités définissent bien des poids sur tous les arcs de $R^*(\varphi)$ et pour n'importe quel cas de figure.
2. Que la somme des poids d'un arc entrant à e_i et celui d'un arc sortant de e_i répond à l'hypothèse faite, sauf dans un seul et unique cas.

Type n°	Cas auxquels les arcs appartiennent		La somme des poids	Nature du résultat
	Arcs entrants	Arcs sortants		
1(*)	1	1	1	Erroné
2(**)	1	1	1	Correct
3	1	2	1	Correct
4	1	3	0	Correct
5	2	1	0	Correct
6	2	2	0	Correct
7	2	3	-1	Correct

Tableau 2 : Caractéristiques de tous les chemins du type $\{P_{k1}, e_i, P_{k2}\}$ possibles passant par le sommet e_i .

(*) : Les arcs entrant et sortant concernent la même période.

(**) : Les arcs entrant et sortant concernent des périodes différentes.

Si les arcs (entrant et sortant) sont du type 1, le résultat de la somme des poids est erroné (n'est pas significatif) car nous déplaçons l'unique séance d'une période d'un type de salle à un autre et à la même période et cela ne créera aucun conflit supplémentaire alors que la somme est égale à 1.

La somme des poids des arcs des autres types (2, 3, 4, 5, 6 et 7) correspond, effectivement, au nombre de conflits ajoutés. Par conséquent, le poids d'un chemin (élémentaire) de L_{jk} à p ou d'un circuit élémentaire ne contenant pas une séquence d'arcs du type 1 donne effectivement le nombre de conflits total ajoutés sur l'ensemble des séances des sommets du type e_i parcourus, ce qui ne serait le cas si le chemin contenait au moins une séquence d'arcs du type 1.

IV.3.2. Poids spécifique d'un chemin sur le graphe de transition

Voici un algorithme simple qui permet d'associer à chaque chemin de L_{jk} à p ou circuit élémentaire C sur $R^*(\varphi)$ le nombre de conflits ajoutés. Cet algorithme ne fait, en fait, que sommer les poids des arcs de C sauf ceux du type 1 (dont la somme est supposée être nulle pour refléter réellement le nombre de conflits ajoutés).

Algorithme « LongSpécifique »

Nous parcourons les arcs dans l'ordre par lequel ils sont représentés sur C .

(0) $\text{NbConf} := 0$;

(1) *Tant que* (le nombre d'arcs restants est ≥ 2) *faire*

Début

Si (les deux prochains arcs ne sont pas du type 1) *faire*

- $\text{NbConf} := \text{NbConf} +$ la somme des poids des deux prochains arcs.
- Avancer de deux arcs.

Fin

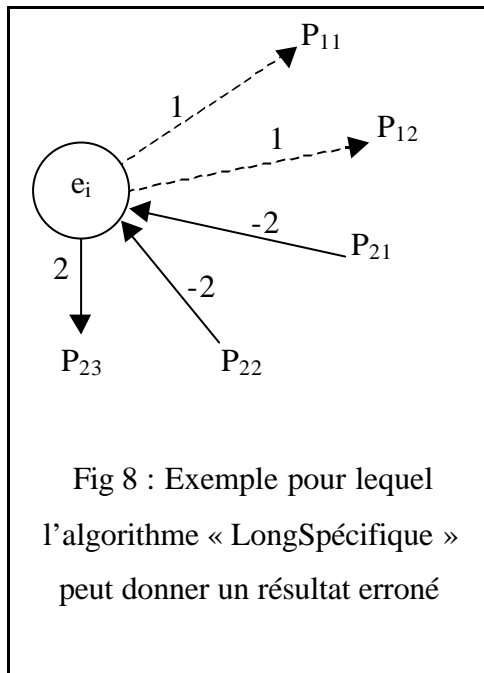
Fin de l'algorithme.

Dans ce qui suit, nous utiliserons l'appellation « *poids spécifique* » pour désigner le nombre « NbConf » donné à la fin de l'algorithme et qui représente le nombre de conflits ajoutés (si ce nombre est négatif, sa valeur absolue sera le nombre de conflits réduits) sur les séances des sommets du type e_i parcourus dans C .

Remarque

Une condition *suffisante* pour que le poids spécifique d'un chemin de L_{jk} à p ou d'un circuit C de $R^*(\varphi)$ soit le nombre de conflits ajoutés, est que C soit *élémentaire*.

En effet, le tableau 2 ainsi que l'algorithme « LongSpécifique » décrits ci-dessus montrent bien cela (il est à noter que les résultats mis sur le tableau 2 sont obtenus à partir de chemins qui parcourent un sommet e_i une seule fois). Voici un cas de figure où le parcours d'un sommet e_i se fait deux fois et où l'algorithme « LongSpécifique » donne un résultat erroné (« NbConf » n'est pas égal au nombre de conflits ajoutés):

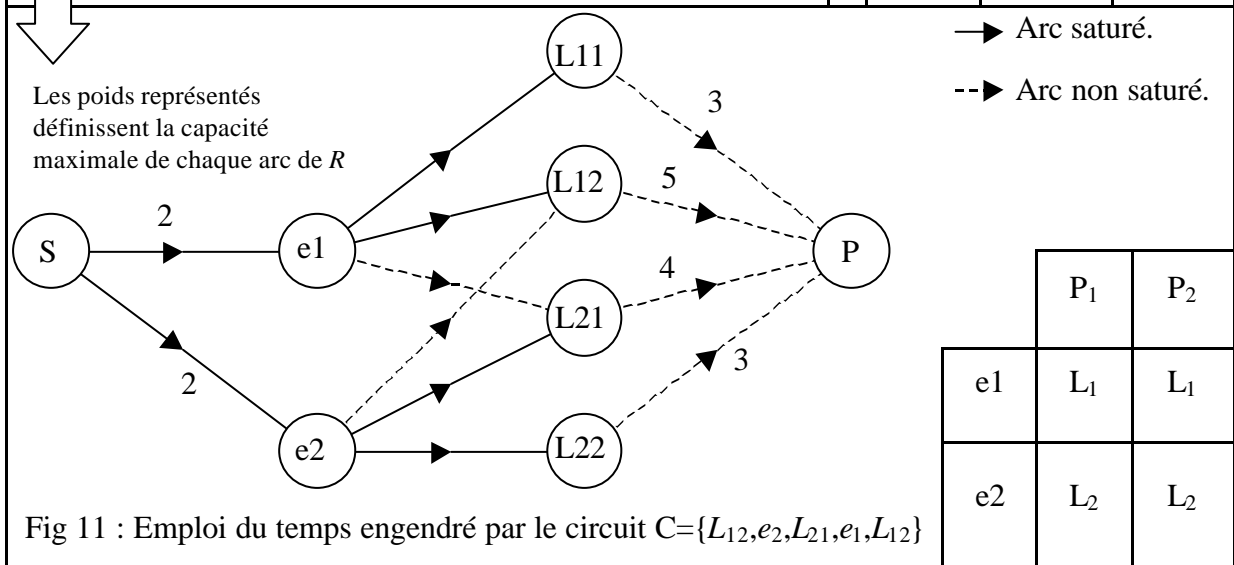
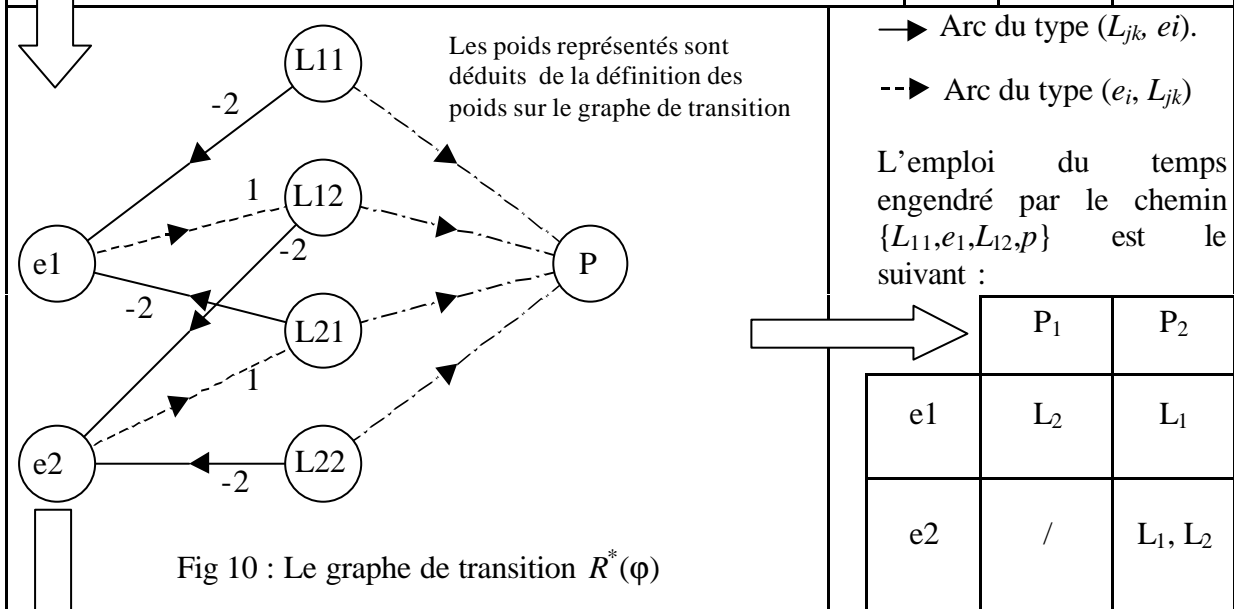
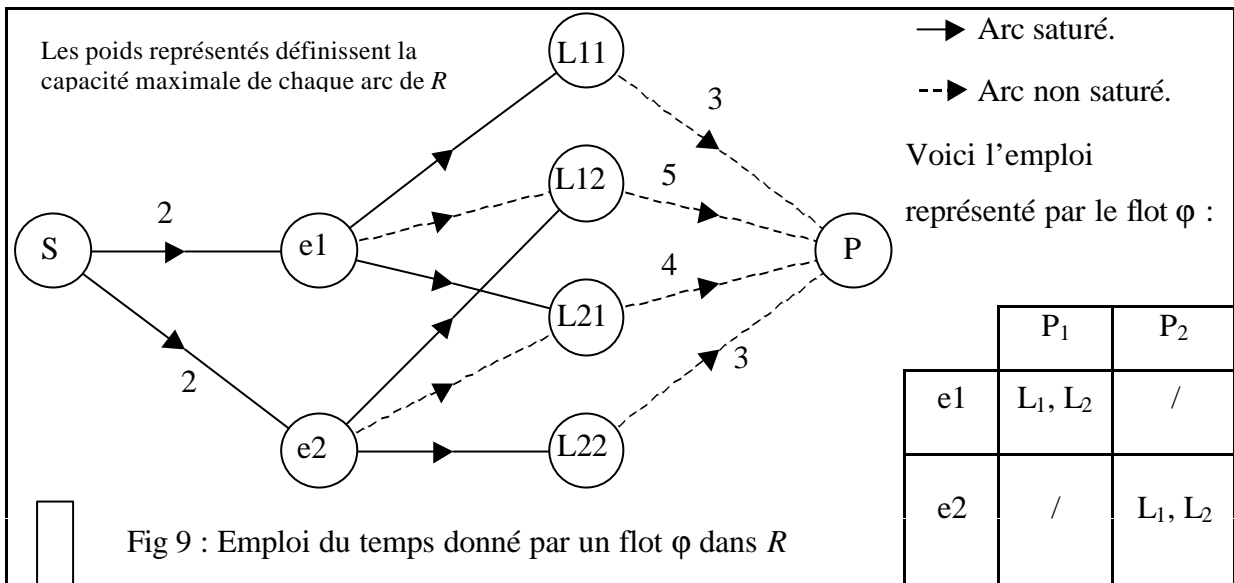


Si C contenait à la fois les séquences $\{(P_{21}, e_i), (e_i, P_{11})\}$ et $\{(P_{22}, e_i), (e_i, P_{12})\}$, l'algorithme « LongSpécifique » comptabilisera (-2) le nombre de conflits ajoutés sur ces deux séquences alors qu'il est clair que le parcours de la première diminue le nombre de conflits d'une unité mais *lorsqu'ensuite* nous parcourons la seconde, le nombre de conflits augmente d'une unité, et en définitive, le conflit est tout simplement déplacé de la seconde période à la première, ce qui ne change en rien le nombre de conflits sur les séances du sommet e_i .

Par conséquent, le poids de tout chemin ou circuit *élémentaire* C donné par l'algorithme « LongSpécifique » est effectivement le nombre de conflits ajoutés sur les sommets e_i parcourus. Aussi, dans ce qui suit, nous ne considérerons que :

1. Des chemins élémentaires des sommets du type L_{jk} à p sur $R^*(\varphi)$.
2. Des circuits élémentaires sur $R^*(\varphi)$.

Nous donnons, ci-dessous, un exemple illustrant la définition des poids sur un graphe de transition ainsi que deux emplois du temps générés ; le premier grâce à un chemin élémentaire d'un sommet L_{jk} à p et le second par un circuit élémentaire dont les poids spécifiques sont, effectivement, le nombre de conflits ajoutés sur ces derniers.



IV.3.3.L'algorithme « PlusCourtSpécifique »

A présent, si nous optons pour déplacer une séance de L_{jk} , nous savons définir un ou plusieurs chemins de ce sommet jusqu'à p ou bien un ou plusieurs circuits qui permettent de générer de nouveaux emplois du temps dans lesquels le déplacement en question peut être effectué.

Nous savons aussi, grâce à la notion de *poids spécifique*, associer à chaque chemin ou circuit de ce type le nombre de conflits ajoutés (ou réduits) par le nouvel emploi du temps qu'il engendre.

Ainsi, après avoir choisi un sommet initial L_{jk} , nous pourrions nous poser la question suivante : quelle est la meilleure manière (en terme du nombre de conflits réduits) avec laquelle nous pouvons changer le reste des affectations pour générer le nouvel emploi du temps ?.

La méthode que nous proposons ici ne répond pas complètement à cette question mais par cette dernière, nous tentons de modifier l'emploi du temps initial de la « meilleure manière possible » à travers :

1. Soit un circuit élémentaire de poids spécifique négatif.
2. Soit un chemin élémentaire de L_{jk} à p de poids spécifique minimum.

De cette façon, si un circuit élémentaire de poids spécifique négatif existe, alors cela permet de réduire le nombre de conflits, sinon, si au moins un chemin élémentaire de L_{jk} à p de poids spécifique négatif existe, alors c'est le meilleur de ces derniers qui sera sélectionné (celui dont le poids spécifique est le plus réduit) sinon, ce sera celui qui augmentera le moins possible le nombre de conflits qui sera sélectionné pour générer le nouveau emploi du temps.

Il est à noter que si la notion de poids spécifique était équivalente à celle du poids classique d'un chemin élémentaire (c'est à dire égal à la somme des poids des arcs le composant), l'utilisation des algorithmes de recherche d'un chemin de longueur minimum entre un sommet et tous les autres et qui peuvent détecter un circuit de poids négatif (comme celui de Bellman ou de Floyd [25]) peuvent être appliqués, mais vu la particularité de la notion de poids spécifique (la somme des poids des arcs du type 1 n'est pas significative en terme du nombre de conflits ajoutés), ce type d'algorithmes ne peut être utilisé.

Pour cela, nous proposons un algorithme (qui est, en fait, une modification de celui de Bellman) pour retrouver soit *un circuit élémentaire de poids spécifique négatif ou bien un chemin élémentaire de L_{jk} à p de poids spécifique minimum.*

Commençons, tout d'abord, par examiner de près l'algorithme de Bellman qui recherche le plus court chemin d'un sommet 1 à tous les autres (numérotés de 2 à N), ou bien détecte un circuit de poids négatif.

Algorithme de Bellman

(a) Poser $k := 1$; $\pi^0(1) := 0$ et $\pi^0(i) := +\alpha$ pour tous les autres sommets i ($2 \leq i \leq N$).

(b) A l'itération k , pour tous les sommets i , poser :

$$\pi^k(i) := \text{Min}[\pi^{k-1}(i), \text{Min}_{j \in \Gamma^{-1}(i)} (\pi^{k-1}(j) + l_{ji})]$$

(c) Si $\pi^k(i) = \pi^{k-1}(i)$ pour tout i , Fin (*).

Si $(N-1) \geq k$, aller en (b) avec $k := k+1$;

Si $k = N$, il y a un circuit de poids négatif. Fin.

Fin de l'algorithme.

l_{ij} : est le poids de l'arc (i,j) .

$\Gamma^{-1}(i)$: Désigne l'ensemble des prédécesseurs du sommet i .

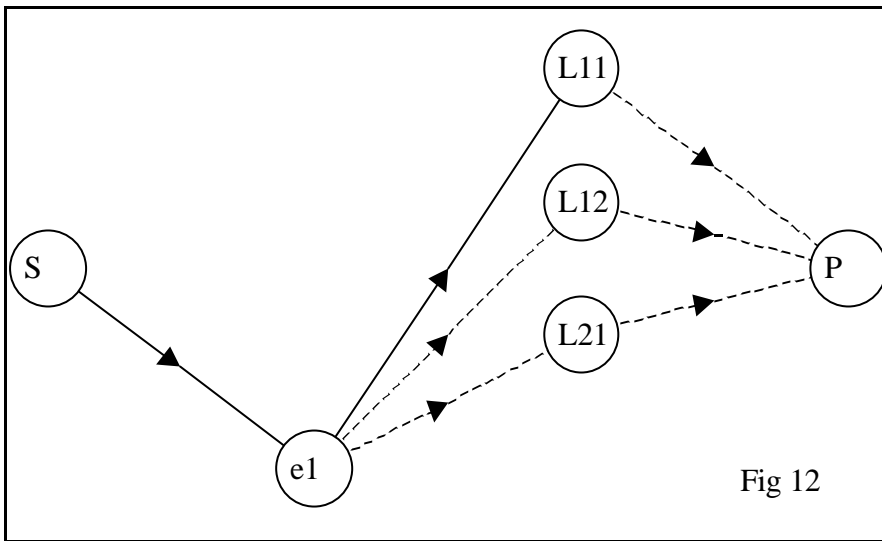
π : Une application (de marquage) qui associe à chaque sommet i la longueur du plus court chemin entre le sommet 1 et i si l'algorithme s'arrête à l'étape (*).

L'algorithme de Bellman est un algorithme de marquage qui attribut, à chaque itération k , à tout sommet i la marque $\pi^k(i)$ qui dépend des marques attribuées aux sommets à l'itération précédente.

Lors de chaque opération de marquage d'un sommet i , nous pouvons connaître le sommet j à partir duquel il a été marqué (sommet pour lequel le minimum est atteint à l'étape (b)), et c'est d'ailleurs, grâce à cela que nous pourrons retrouver le plus court chemin ou bien un circuit à la fin de l'algorithme et affirmer qu'un arc (i,j) en fait partie.

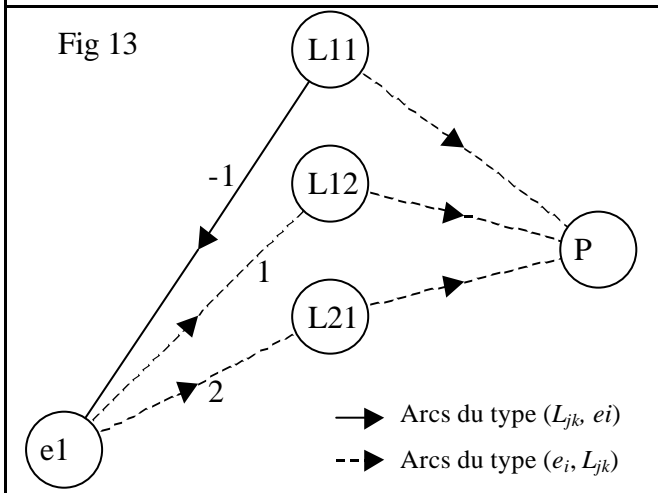
Or, si nous voulons retrouver le chemin de poids spécifique minimum entre un sommet 1 et tous les autres ou bien un circuit de poids spécifique négatif, l'application de l'algorithme tel qu'il est ignore la spécificité du marquage d'un sommet se trouvant à la fin de deux arcs du type 1 sur le réseau $R^*(\varphi)$ (l'algorithme comptabilisera un conflit de plus alors qu'il n'y en a aucun) et donnera, donc un résultat erroné sur un chemin ou un circuit contenant au moins deux arcs du type 1.

Voici un exemple :



→ : Arc saturé.
 --> : Arc non saturé.

Le graphe de transition $R^*(\varphi)$ qui correspond à l'emploi du temps représenté par la fig 12 est schématisé sur la Fig13.



Si nous décidons de déplacer l'unique séance de e_1 à partir de L_{11} , nous commençons par enlever l'arc (L_{11}, p) . Le marquage résultant de l'application de l'algorithme de Bellman est donné sur le tableau suivant :

Sommets	L_{11}	L_{21}	L_{12}	e_1	P
Marques	0	<u>1</u>	0	-1	0

Tableau 3 : Application de l'algorithme de Bellman (Plus courts chemins à partir de L_{11})

Ce marquage montre que si le chemin sélectionné est $\{L_{11}, e_1, L_{21}, p\}$ nous rajoutons un conflit sur le prochain emploi du temps, ce qui n'est pas le cas car, par ce chemin, nous ne faisons que déplacer l'unique séance de e_1 de L_{11} à L_{21} ; ce résultat est dû, comme déjà souligné, au fait que le chemin emprunté contient deux arcs du type 1 (ici : (L_{11}, e_1) et (e_1, L_{21})).

La modification de l'algorithme doit intervenir, donc, sur ce point et doit marquer les sommets de sorte que la valeur de leur marque soit significative et reflète bien le nombre de conflits ajoutés.

Pour ce faire, si un sommet du type L_{jk} doit être marqué à partir d'un sommet du type e_i , et si l'arc (e_i, L_{jk}) ainsi que l'arc à partir duquel le sommet e_i a été marqué sont du type 1, alors la marque attribuée à L_{jk} sera la marque attribuée par l'algorithme de Bellman à laquelle nous retranchons une unité, et cela afin de « corriger » l'erreur induite par les arcs du type 1 dans le calcul du nombre de conflits ajoutés.

Ainsi, l'algorithme sera le suivant :

Algorithme PlusCourtSpécifique

/* Recherche des chemins de poids spécifiques minimums à partir du sommet 1 ou détection d'un circuit de poids spécifique négatif dans $R^*(\varphi)$ (les sommets sont numérotés de 1 à N) */.

(0) Poser $k := 1$; $\pi^0(1) := 0$ et $\pi^0(i) := +\alpha$ pour tous les autres sommets i ($2 \leq i \leq N$).

(1) A l'itération k , pour tous les sommets i , poser :

$$\pi^k(i) := \text{Min}[\pi^{k-1}(i), \underset{\substack{j \in \Gamma^{-1}(i) \\ j \text{ vérifie (D1)}}}{\text{Min}} (\pi^{k-1}(j) + l_{ji}), \underset{\substack{j \in \Gamma^{-1}(i) \\ j \text{ vérifie (D2)}}}{\text{Min}} (\pi^{k-1}(j) + l_{ji} - 1)]$$

(2) Si $\pi^k(i) = \pi^{k-1}(i)$ pour tout i , Fin (S1).

Si $\pi^k(1) < 0$, alors il y a un circuit de poids spécifique négatif, Fin (S2).

Si $(N-1) \geq k$, aller en (1) avec $k := k+1$ (S3).

Si $k = N$, il y a un circuit de poids spécifique négatif, Fin (S4).

Fin de l'algorithme.

Où :

l_{ij} : est le poids de l'arc (i,j) .

$\Gamma^{-1}(i)$: Désigne l'ensemble des prédécesseurs du sommet i .

π : Une application (de marquage) qui associe à chaque sommet i le poids spécifique du chemin de 1 à i de poids spécifique minimum si l'algorithme s'arrête à l'étape (S1).

j vérifie (D1) : L'arc (j, i) ainsi que l'arc à partir duquel le sommet j a été marqué *ne sont pas* du type 1 ou i n'est pas du type L_{jk} .

j vérifie (D2) : L'arc (j, i) ainsi que l'arc à partir duquel le sommet j a été marqué *sont* du type 1 et i est du type L_{jk} .

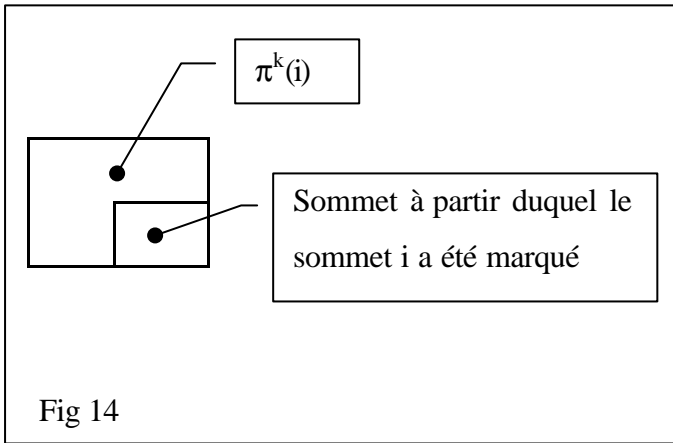
Remarque

1. La modification de l'algorithme de Bellman s'est faite, essentiellement, à l'étape (1) et cela afin de tenir compte de la définition de la notion de poids spécifique en marquant les sommets se trouvant à la fin de deux arcs du type 1 d'une manière différente.
2. Si l'algorithme s'arrête à l'étape (S2) ou (S4), alors il existe un circuit de poids spécifique négatif.
3. Si, sur le réseau $R^*(\varphi)$, il n'y a pas d'arcs du type 1, l'algorithme ci-dessus est équivalent à celui de Bellman.
4. Si l'algorithme ne détecte pas de circuit de poids spécifique négatif, nous pouvons, grâce aux marques attribuées aux sommets du type L_{jk} , ordonner l'ensemble des chemins de ce dernier jusqu'à p , par ordre croissant de leur poids spécifique.

Reprenons, à présent, le dernier exemple cité (Fig 13), et appliquons l'algorithme « PlusCourtSpécifique » à ce dernier. Voici un tableau résumant l'ensemble des itérations de l'algorithme :

Itérations k	L_{11}	L_{21}	L_{12}	e_1	P
0	0 /	$+\infty$ /	$+\infty$ /	$+\infty$ /	$+\infty$ /
1	0 /	$+\infty$ /	$+\infty$ /	-1 L_{11}	$+\infty$ /
2	0 /	0 e_1	0 e_1	-1 L_{11}	$+\infty$ /
3	0 /	0 e_1	0 e_1	-1 L_{11}	0 L_{21}
4	0 /	0 e_1	0 e_1	-1 L_{11}	0 L_{21}

Tableau 4 : Application de l'algorithme « PlusCourtSpécifique » à la Fig 13.

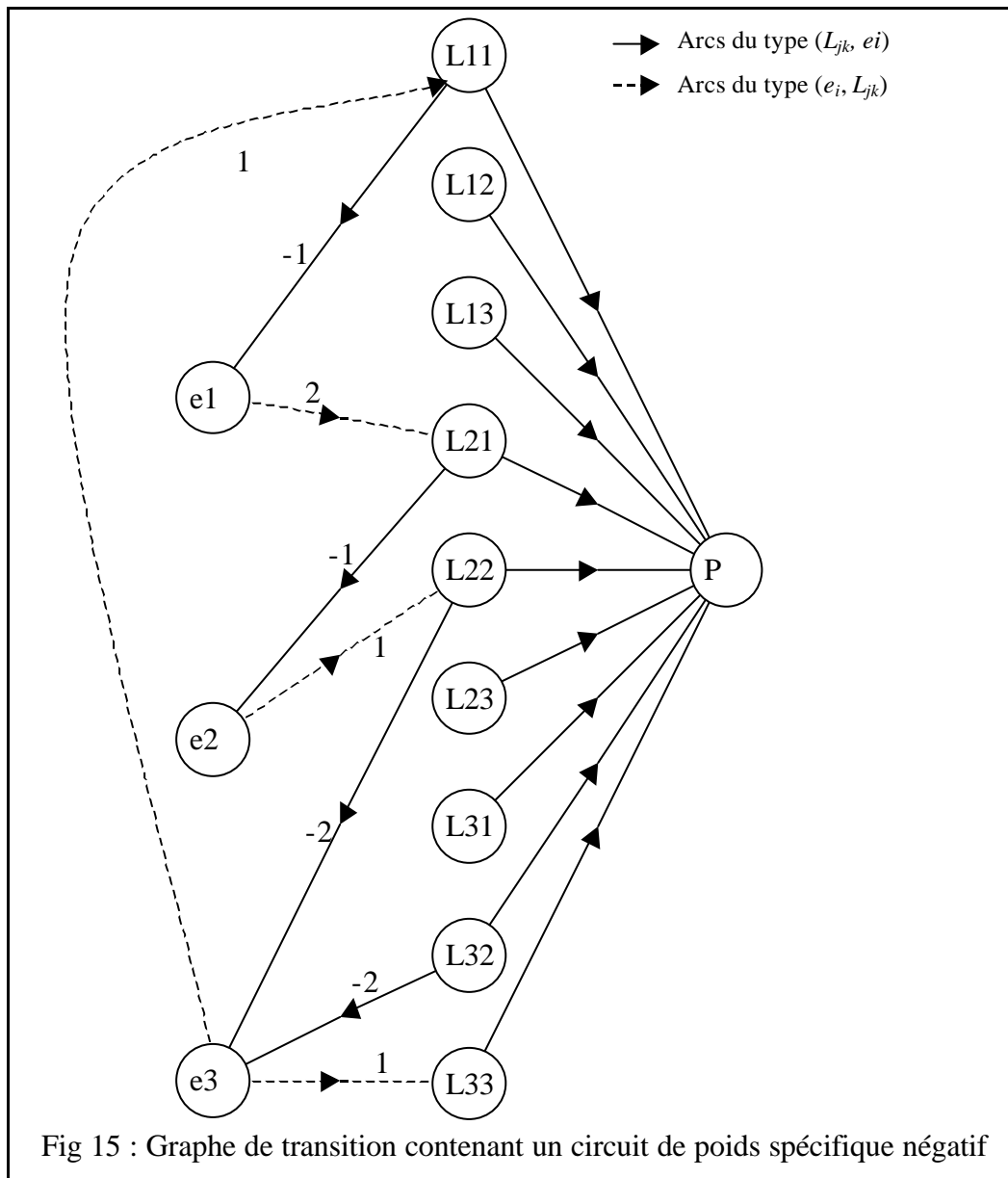


A présent, les marques attribuées aux sommets reflètent bien le nombre de conflits ajoutés.

A la troisième itération ($k = 2$), le sommet L_{21} a été marqué à partir du sommet e_1 qui, à son tour, a été marqué par le sommet L_{11} , or (L_{11}, e_1) et (e_1, L_{21}) sont bien du type 1 et donc :

$\pi^2(L_{21}) = [(-1)+2]-1 = 0$. Ce qui reflète bien le fait que si nous déplaçons une séance relative au sommet e_1 de L_{11} à L_{21} , le nombre de conflits sur les séances de e_1 ne change pas.

Considérons, à présent, un exemple où $R^*(\varphi)$ contient un circuit de poids spécifique négatif et qui comporte au moins deux arcs du type 1, et appliquons à ce dernier l'algorithme « PlusCourtSpécifique ».



Nous remarquons, dans ce cas, l'existence d'un circuit $C = \{L_{11}, e_1, L_{21}, e_2, L_{22}, e_3, L_{11}\}$; ce dernier a un poids nul et un poids spécifique égal à (-1).

Si nous voulons déplacer une séance de L_{11} alors nous enlevons l'arc (L_{11}, p) du graphe et nous appliquons l'algorithme dont les itérations sont les suivantes :

Itérations k	L11	L21	L22	L32	L33	e1	e2	e3	P
0	0 /	$+\infty$ /	$+\infty$ /	$+\infty$ /	$+\infty$ /	$+\infty$ /	$+\infty$ /	$+\infty$ /	$+\infty$ /
1	0 /	$+\infty$ /	$+\infty$ /	$+\infty$ /	$+\infty$ /	-1 L ₁₁	$+\infty$ /	$+\infty$ /	$+\infty$ /
2	0 /	0 e ₁	$+\infty$ /	$+\infty$ /	$+\infty$ /	-1 L ₁₁	$+\infty$ /	$+\infty$ /	$+\infty$ L ₂₁
3	0 /	0 e ₁	$+\infty$ /	$+\infty$ /	$+\infty$ /	-1 L ₁₁	-1 L ₂₁	$+\infty$ /	0 L ₂₁
4	0 /	0 e ₁	0 e ₂	$+\infty$ /	$+\infty$ /	-1 L ₁₁	-1 L ₂₁	$+\infty$ /	0 L ₂₁
5	0 /	0 e ₁	0 e ₂	$+\infty$ /	$+\infty$ /	-1 L ₁₁	-1 L ₂₁	-2 L ₂₂	0 L ₂₁
6	-1 e ₃	0 e ₁	0 e ₂	$+\infty$ /	-1 e ₃	-1 L ₁₁	-1 L ₂₁	-2 L ₂₂	0 L ₂₁

Tableau 5 : Application de l'algorithme « PlusCourtSpécifique » à la Fig 15.

Nous constatons qu'il y a bien un circuit de poids spécifique négatif, et pour l'identifier, il faut se placer à la dernière case de la colonne qui correspond au sommet L_{11} , il est marqué à partir du sommet e_3 qui est, à son tour, marqué par L_{22} et ainsi de suite jusqu'à ce que nous tombons sur un sommet déjà parcouru, et dans ce cas le circuit trouvé est C.

Remarque

1. Après avoir choisi un sommet L_{jk} , et même s'il n'existe pas de circuit de poids spécifique négatif, il se peut qu'un chemin de ce dernier jusqu'au sommet p n'existe pas dans $R^*(\varphi)$ (dans ce cas, le sommet p sera marqué par $+\infty$ à la fin de l'algorithme). Cela signifie qu'aucune séance de L_{jk} ne peut être déplacée ou réaffectée ailleurs.
2. Rappelons que la procédure décrite pour générer un emploi du temps à partir d'un autre se base sur le nombre de conflits par rapport aux sommets $\{e_i\}$ et non pas par rapport à tout l'emploi du temps qui prend aussi en compte les conflits générés par les séances d'un même groupe d'enseignement qui sont représentées par plus d'un sommet e_i . Ainsi, plus le nombre de sommets du type e_i représentant chaque groupe d'enseignement est réduit, plus la procédure sera efficace.

IV.3.4. Succession de circuits de poids spécifique négatif

L'algorithme « PlusCourtSpécifique » a été conçu pour être appliqué à plusieurs reprises et générer ainsi une succession d'emplois du temps en espérant une amélioration rapide et efficace du nombre de conflits sur ces derniers.

Or, nous ne pouvons considérer une telle démarche sans penser à l'éventualité de « bouclage » qu'il peut y avoir et qui risque de stopper toute amélioration possible du nombre de conflits.

Comme déjà souligné, l'application de l'algorithme « PlusCourtSpécifique » se fait à partir d'un sommet L_{jk} choisi au préalable. Si un chemin de poids spécifique minimum est trouvé et que ce dernier peut induire un « bouclage », ce chemin peut être évité en considérant un autre chemin de L_{jk} à p (en choisissant le meilleur qui n'induit pas de « bouclage », ce choix sera facile puisque le marquage des sommets est déjà effectué) ou bien choisir un autre sommet du type L_{jk} et appliquer l'algorithme encore une fois.

Par contre, si un circuit de poids spécifique négatif est détecté, l'application de l'algorithme « PlusCourtSpécifique » risque de toujours détecter ce même circuit quelque soit le sommet du type L_{jk} choisi au départ.

Examinons, donc, l'éventualité suivante : nous disposons d'un emploi du temps sous forme d'un flot maximum φ_1 sur le réseau R pour lequel il existe un circuit C_1 de poids spécifique négatif dans $R^*(\varphi_1)$, nous modifions l'emploi du temps en cours grâce au circuit C_1 pour obtenir un nouveau flot φ_2 de même valeur que φ_1 (c'est à dire un nouveau emploi du temps). Est-ce qu'au circuit C_1 de $R^*(\varphi_1)$ correspond un circuit C_2 de $R^*(\varphi_2)$ de poids spécifique négatif ? Ce qui pourrait, éventuellement, engendrer une suite infinie de circuits de poids spécifique négatif.

Tout d'abord, si nous modifions φ_1 par C_1 pour obtenir φ_2 alors la structure du graphe $R^*(\varphi_2)$ ne sera différente de celle $R^*(\varphi_1)$ que sur le circuit C_1 et $R^*(\varphi_2)$ contiendra un circuit C_2 qui n'est autre que C_1 mais dont les arcs sont inversés (donc C_1 n'existe pas dans $R^*(\varphi_2)$), de plus, le poids des arcs sur le graphe de transition ainsi que les algorithmes cités ci-dessus ont été définis de sorte que le poids spécifique de n'importe quel circuit C soit *exactement* égal au nombre de conflits ajoutés en modifiant l'emploi du temps en cours par le circuit C (affectations et désaffectations) ; donc, si C_1 ajoute $p_s(C_1)$ conflits et engendre φ_2 , il est clair que parcourir C_2 engendrera de nouveau φ_1 (il annulera les modifications apportées par C_1 en inversant, encore une fois, les arcs de C_2), et par conséquent, $p_s(C_2) = -p_s(C_1)$, d'où, si $p_s(C_1) < 0$ alors $p_s(C_2) > 0$. Voici un exemple illustrant cela :

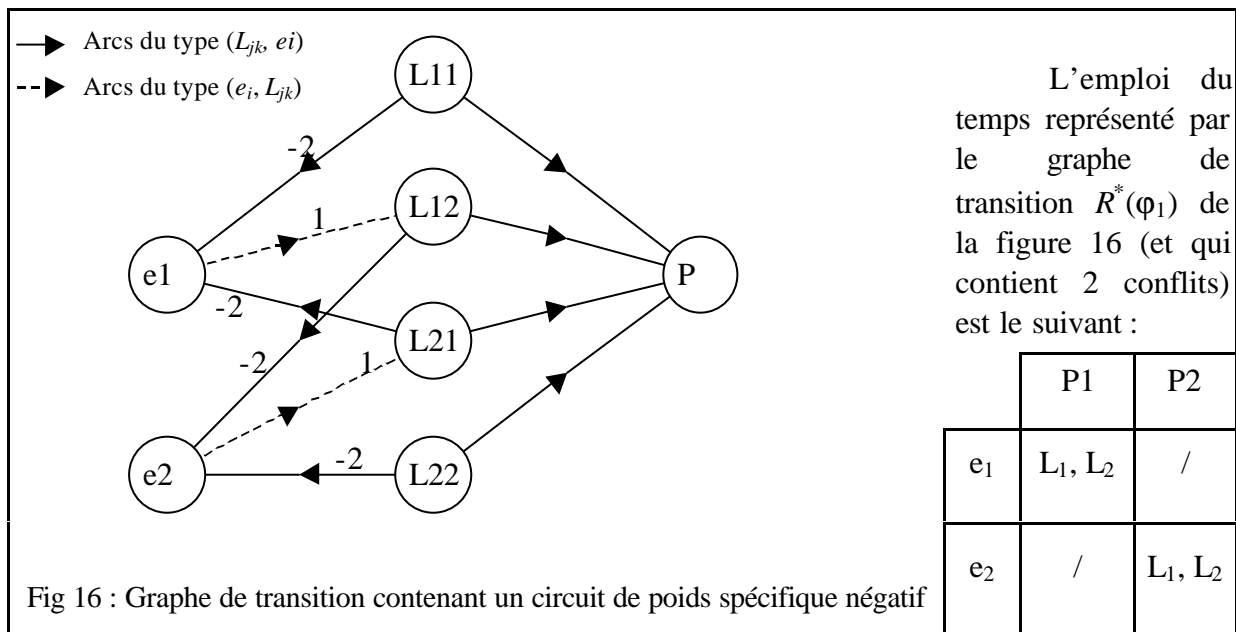
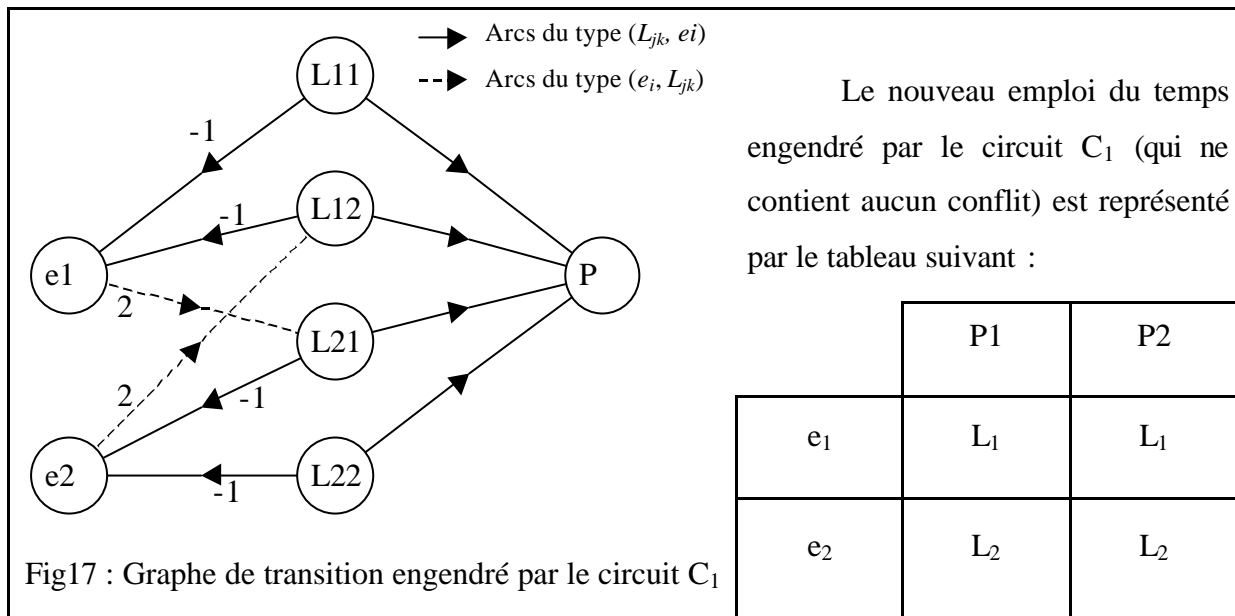


Fig 16 : Graphe de transition contenant un circuit de poids spécifique négatif

Nous considérons le circuit $C_1 = \{L_{12}, e_2, L_{21}, e_1, L_{12}\}$ dont le poids spécifique est égal à $p_s(C_1) = (-2)$ (réduit le nombre de conflits de deux unités) et soit φ_2 le flot engendré par φ_1 (flot dont le graphe de transition est représenté ci-dessus sur la figure 16) modifié par le circuit C_1 . Alors, le graphe de transition $R^*(\varphi_2)$ sera le suivant :



Nous voyons bien, ici, que le circuit $C_2 = \{L_{12}, e_1, L_{21}, e_2, L_{12}\}$ n'est autre que le circuit C_1 dont les arcs ont été inversés (et c'est la seule différence qui existe entre la structure des graphes de la figure 16 et 17), de plus, $p_s(C_2) = 2 = - p_s(C_1)$, et ceci était tout à fait prévisible car grâce à C_1 nous avons réduit le nombre de conflits de 2 unités et C_2 ne fait qu'annuler les modifications apportées par C_1 .

Il est à noter aussi que puisque la définition des poids sur les arcs du graphe de transition ainsi que la notion de poids spécifique d'un chemin sont *directement liées* au nombre de conflits ajoutés sur l'ensemble de toutes les séances relatives aux sommets du type e_i , alors une succession infinie de circuits de poids spécifique négatif sur les graphes de transition générés ne peut exister car cela entraînerait une diminution infinie du nombre de conflits sur toutes les séances, ce qui est impossible car ce nombre possède une borne inférieure nulle.

A titre d'exemple, l'emploi du temps correspondant à la figure 17 n'a aucun conflit sur les séances relatives aux sommets du type e_i et tout chemin d'un sommet L_{jk} à p ou circuit de $R^*(\varphi_2)$ a, forcément, un poids spécifique positif ou nul.

Un autre exemple

Reprenons l'exemple de la figure 2 considéré au début du chapitre pour appliquer l'algorithme « PlusCourtSpécifique ». Sur cet exemple, nous avons distingué trois chemins dont le meilleur réduit le nombre de conflits de 2 unités et génère un emploi du temps sans conflits par rapport aux séances des sommets $\{e_i\}$.

Nous pouvons facilement constater que les poids spécifiques des chemins définis sur le tableau 1 correspondent, effectivement, aux nombres de conflits ajoutés.

Si nous voulons déplacer une séance de L_{21} et trouver le chemin de poids spécifique minimum de L_{21} à p , il faut supprimer l'arc (L_{21}, p) et appliquer l'algorithme, dont la dernière itération est la suivante :

k	L_{11}	L_{12}	L_{21}	L_{22}	L_{31}	L_{32}	e_1	e_2	P
4	-2	-1	-2	-1	0	-1	-3	-2	-1
	e_2	e_2	e_1	e_2	e_2	e_2	L_{22}	L_{21}	L_{32}

Puisque le sommet L_{21} a été marqué par une valeur négative, alors il existe un circuit de poids spécifique négatif qui est $C=\{L_{21}, e_2, L_{22}, e_1, L_{21}\}$ et qui réduit, par conséquent, le nombre de conflits de 2 unités pour avoir l'emploi du temps suivant :

	P1	P2
e_1	L_2	L_1
e_2	L_1	L_2

Le circuit C génère un emploi du temps qui ne possède aucun conflit sur les séances des sommets $\{e_i\}$. Il est à noter que ce dernier est différent du meilleur emploi du temps obtenu sur le tableau 1.

IV.3.5.Complexité de l'algorithme « PlusCourtSpécifique »

Dans ce qui suit, nous montrons que le passage d'un flot φ à un autre φ' (et donc, d'un emploi du temps à un autre) par l'application de l'algorithme « PlusCourtSpécifique » sur le graphe de transition $R^*(\varphi)$ se fait en un temps polynomial.

A. Rappel

Définition 1 : Codage raisonnable des données [35]

Un codage des données est dit *raisonnable* si les nombres ne sont pas codés sous forme unaire, c'est à dire, si l'encombrement mémoire nécessaire pour stocker le nombre entier positif N est égal à $\lceil \text{Log}(N+1) \rceil$ (où $\lceil x \rceil$ désigne la partie entière supérieure de l'entier positif x).

Définition 2 : Algorithme polynomial [35]

Un algorithme est dit *polynomial* si le nombre d'opérations élémentaires nécessaires pour résoudre un exemple de taille N est borné par un polynôme en N.

Définition 3 : Problème de reconnaissance [35]

Un problème de *reconnaissance* est un problème dont les résultats ne peuvent être que vrai ou faux.

B. Démonstration

Soient N et M le nombre de sommets et d'arcs respectivement du graphe de transition $R^*(\varphi)$, où φ est le flot calculé sur le réseau initial R.

Il est prouvé que l'algorithme de Bellman, pour la recherche d'un plus court chemin d'un sommet donné à tous les autres, est polynomial (il est en $O(N.M)$ et $M \leq N^2$), or la différence entre l'algorithme « PlusCourtSpécifique » et celui de Bellman réside dans la procédure de marquage à l'étape (1) (car nous nous basons sur les poids spécifiques et pas « classiques » des chemins) où, au lieu de considérer l'ensemble des prédécesseurs d'un sommet i pour le marquer avec la même règle à une étape k (à savoir, le minimum des quantités $\pi^{k-1}(i)$ et $\text{Min}_{j \in \Gamma^{-1}(i)} (\pi^{k-1}(j) + l_{ji})$), cet ensemble est partitionné en deux ($\{j \in \Gamma_i^{-1} \text{ et } j \text{ vérifie } (D_1)\}$ et $\{j \in \Gamma_i^{-1} \text{ et } j \text{ vérifie } (D_2)\}$) et selon chacun de ces ensembles il existe une règle de marquage différente :

1. Si $j \in \Gamma_i^{-1}$ et j vérifie (D_1) : $\text{Min}[\pi^{k-1}(i), \text{Min}_{\substack{j \in \Gamma^{-1}(i) \\ j \text{ vérifie } (D1)}} (\pi^{k-1}(j) + l_{ji})]$.
2. Si $j \in \Gamma_i^{-1}$ et j vérifie (D_2) : $\text{Min}[\pi^{k-1}(i), \text{Min}_{\substack{j \in \Gamma^{-1}(i) \\ j \text{ vérifie } (D2)}} (\pi^{k-1}(j) + l_{ji} - 1)]$.

Il est à noter que si un sommet e_i doit être marqué, nous sommes forcément dans le cas $(D1)$. Par ailleurs, pour marquer un sommet L_{jk} nous pouvons vérifier en un temps polynomial si l'arc (e_i, L_{jk}) et l'arc $(L_{j'k}, e_i)$, à partir duquel le sommet e_i a été marqué, sont du type 1 (ce qui déterminera si ce sommet vérifie (D_1) ou bien (D_2)).

En effet, il suffira de vérifier les deux conditions suivantes (qui définissent, en fait, les arcs correspondants au cas $n^o 1$) :

1. $k = k'$
2. $(L_{j'k}, e_i)$ est l'unique arc entrant relatif à la période k' ($=k$) vers e_i .

Montrons, donc, que le problème de reconnaissance P suivant est polynomial :

« Il y a un unique arc entrant vers un sommet e_i donné d'une période k donnée ».

(i).Codage des données du problème :

Le codage des données de ce problème se fera de la manière suivante :

$d_i, j_1, k_1, E_1, j_2, k_2, E_2, \dots, j_{d_i}, k_{d_i}, E_{d_i}$ pour désigner le fait que :

1. Le sommet e_i est degré d_i .
2. Le sommet e_i est adjacent, dans le graphe de transition, aux sommets $L_{j_1 k_1}, L_{j_2 k_2}, \dots, L_{j_{d_i} k_{d_i}}$.
3. Le sens de l'arc reliant e_i à $L_{j_s k_s}$ ($s = 1, \dots, d_i$) est donné par le digit E_s , qui vaut 0 s'il est entrant à e_i et 1 sinon.

La taille du problème sera donc répartie sur les quantités suivantes :

1. Codage de d_i : $\lceil \text{Log}(d_i + 1) \rceil$.
2. Codage des sommets adjacents à e_i : $\sum_{s=1..d_i} (\lceil \text{Log}(j_s + 1) \rceil + \lceil \text{Log}(k_s + 1) \rceil)$.
3. Codage de la nature des arcs (entrant ou sortant) : d_i .
4. Codage des séparateurs : $3d_i$.

La taille totale C du codage sera donc :

$$C = 4d_i + \lceil \text{Log}(d_i + 1) \rceil + \sum_{s=1..d_i} (\lceil \text{Log}(j_s + 1) \rceil + \lceil \text{Log}(k_s + 1) \rceil)$$

$$\text{Comme } \begin{cases} d_i \leq M \leq N^2 \\ j_s \leq N \leq N^2 \\ k_s \leq N \leq N^2 \end{cases}$$

$$\text{Alors : } C \leq 4N^2 + \lceil \text{Log}(N^2 + 1) \rceil + 2d_i \lceil \text{Log}(N^2 + 1) \rceil$$

$$\text{Donc : } \exists N_0 \in \mathbb{N} / \forall N \geq N_0 : C \leq 4N^2 + N + 2N^3$$

$$\Rightarrow \exists N_1 \in \mathbb{N} / \forall N \geq N_1 : C \leq 3N^3$$

Posons $T(N) = 3N^3$, borne supérieure de la taille du problème P.

(ii).Nombre d'opérations :

Calculons, à présent, le nombre d'opérations nécessaires pour résoudre le problème de reconnaissance P.

Pour cela, nous considérons l'algorithme suivant qui, après le calcul du nombre d'arcs entrants vers le sommet e_i de la période k, répond au problème P :

Algorithme

Nb := 0 ; (0)

Pour l := 1 à d_i **faire** (1)

Début

Si (k_l = k et E_l = 0) **alors** Nb := Nb + 1 ;

Fin ; (2) (3)

Si (Nb = 1) **alors** réponse := Vrai ; (5)

(4) **sinon** réponse := Faux ;

Fin de l'algorithme.

Pour résoudre le problème P, l'algorithme effectuera au plus :

(0) Une (1) affectation.

(1) d_i affectations.

(2) 2d_i comparaisons.

(3) d_i affectations.

(4) Une (1) comparaison.

(5) Une (1) affectation.

Soit en tout : O = 4d_i+3 opérations.

Or, O = 4d_i+3 ≤ 4N²+3

Donc ∃N₀ ∈ N / ∀N ≥ N₀ : O ≤ 5N²

Nous posons Op(N) = 5N², borne supérieure du nombre d'opérations. Par conséquent :

$\left\{ \begin{array}{l} T(N) = 3N^3 \text{ Est une borne supérieure de la taille du problème P} \\ Op(N) = 5N^2 \text{ Est une borne supérieure du nombre d'opérations pour résoudre P} \end{array} \right.$

$$\text{Or, } Op(N) = 5 \left(\frac{1}{3} T(N) \right)^{\frac{2}{3}} = \left[5 \left(\frac{1}{3} \right)^{\frac{2}{3}} \right] (T(N))^{\frac{2}{3}}$$

Donc, nous pouvons effectivement vérifier en temps polynomial si un sommet L_{jk} vérifie (D₁) ou bien (D₂) avant de le marquer.

Comme l'algorithme de Bellman est polynomial, en substituant l'étape (b) de ce dernier par une procédure polynomiale, l'algorithme résultant sera aussi polynomial. Il s'en suit que l'algorithme « PlusCourtSpécifique » est polynomial.

IV.4. Adaptation de la recherche taboue pour la résolution du problème

La procédure décrite dans les étapes précédentes, à savoir, la définition à partir d'un flot φ sur R du graphe de transition $R^*(\varphi)$ ainsi que des poids sur les arcs de ce dernier, puis la construction et l'application de l'algorithme « PlusCourtSpécifique » pour générer un autre flot φ' , vise à associer à chaque emploi du temps (représenté par φ) un autre emploi du temps (représenté par φ') en essayant de réduire le plus possible le nombre de conflits sur ce dernier.

Ainsi, en adoptant une métaheuristique qui se base sur une stratégie de recherche locale, comme la recherche taboue, pour la résolution de ce problème, nous aurons complètement défini la notion de voisinage en donnant une procédure efficace permettant le déplacement d'un emploi du temps à un autre.

IV.4.1. Présentation générale de la recherche taboue

La recherche taboue est une heuristique destinée à la recherche d'un minimum global d'une fonction f sur un ensemble de solutions réalisables X . La procédure commence par une solution initiale S^0 et cherche à l'améliorer en se déplaçant d'une solution S à une autre S' choisie dans le voisinage $V(S)$ de S en appliquant à cette dernière une certaine transformation TR .

Une liste taboue T est définie pour garder la trace des $|T|$ dernières modifications faites par l'algorithme afin de réduire le risque de bouclage et de revenir à des solutions récemment explorées. Cette façon de faire peut, malheureusement, aussi nous interdire d'atteindre des solutions « intéressantes » et non encore parcourues ; pour cela, nous pouvons ignorer le statut taboue d'une modification quand cela semble raisonnable en introduisant un critère d'aspiration A qui est une règle permettant de savoir s'il est possible ou non de faire un déplacement interdit par la liste T .

La procédure peut être arrêtée si la meilleure solution rencontrée n'est pas améliorée au bout d'un certain nombre d'itérations successives ou lorsque l'évaluation de cette dernière est jugée assez satisfaisante (suffisamment proche d'une valeur f_{Min} fixée au départ).

Algorithme

Initialisation

- Générer une solution initiale S de X .
- $S^* := S$; (Meilleure solution disponible).
- $NbIter := 0$; (Nombre d'itérations).
- $BestIter := 0$; (Itération à laquelle la meilleure solution S^* a été trouvée).
- $Nimax$: Nombre maximum d'itérations successives à faire sans améliorer la meilleure solution rencontrée avant d'arrêter la procédure.

Tant que $(f(S) > f_{Min})$ **et** $((NbIter - BestIter) < Nimax)$ **faire**

Début

- $NbIter := NbIter + 1$;
- Choisir la meilleure solution dans le voisinage $V(S)$ de S qui n'est pas taboue ou qui satisfait le critère d'aspiration A , et l'affecter à S .
- Mettre à jour la liste taboue et la fonction d'aspiration A .

Si $(f(S) < f(S^*))$ **alors**

Début

- $S^* := S$;
- $BestIter := NbIter$;

Fin

Fin

Fin de l'algorithme.

IV.4.2. Adaptation au problème

A. Listes taboues

La liste taboue a pour objectif d'éviter de revenir sur des solutions récemment explorées et rester dans un même voisinage. Pour ce faire, nous prévoyons pour le problème considéré, deux listes taboues T_1 et T_2 .

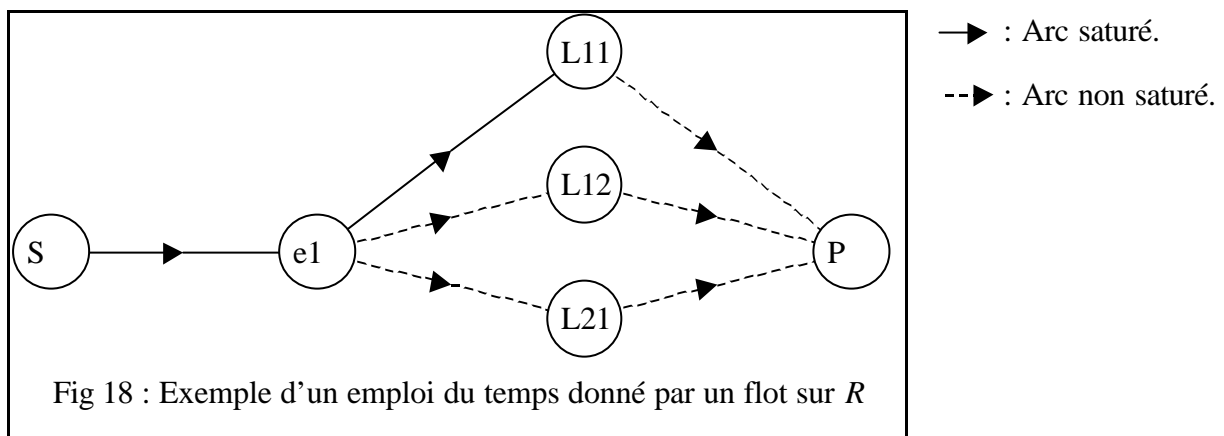
(i). La liste T_1

Si nous optons pour un sommet $L_{j_1 k_1}$ de $R^*(\varphi)$ pour chercher un chemin C de ce dernier jusqu'au sommet p , alors nous sauvegardons dans la liste T_1 le triplet $(L_{j_2 k_2}, L_{j_1 k_1}, -p_s(C))$ où :

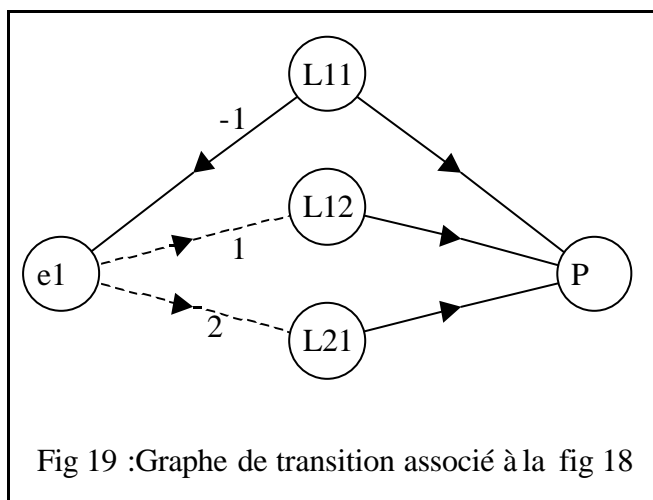
$L_{j_2k_2}$: est le prédécesseur de p dans C .

$P_s(C)$: est le poids spécifique du chemin C .

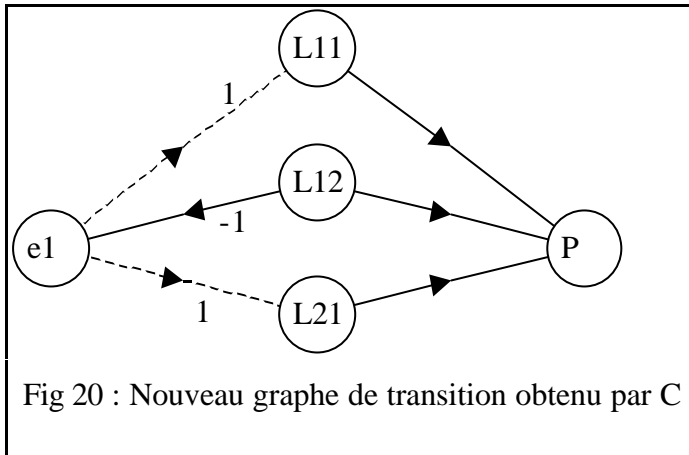
En fait, par le chemin C , nous rajoutons $p_s(C)$ conflits sur l'ensemble des séances des sommets $\{e_i\}$ et cela en « réaffectant » une séance de $L_{j_1k_1}$ et en rajoutant une séance à $L_{j_2k_2}$. Or, par la sauvegarde du triplet $(L_{j_2k_2}, L_{j_1k_1}, -p_s(C))$, nous évitons sur un certain nombre d'itérations de diminuer de $p_s(C)$ conflits sur l'ensemble des séances de $\{e_i\}$ par un chemin C' dont l'extrémité initiale est $L_{j_2k_2}$ (« réaffectation » d'une séance de $L_{j_2k_2}$) et dont le sommet p a comme prédécesseur le sommet $L_{j_1k_1}$ (nous rajoutons une séance à $L_{j_1k_1}$). Voici un exemple illustrant cela :



Le graphe de transition associé au graphe ci-dessus sera le suivant :



Si nous optons pour déplacer la séance de L_{11} par le chemin $C=\{L_{11},e_1,L_{12},p\}$ (après avoir enlever l'arc (L_{11}, p)) qui est de poids spécifique nul, alors le nouveau graphe de transition sera le suivant (Fig 20) :



Dans ce cas, si nous voulons déplacer une séance de L_{12} nous pouvons opter pour le chemin $\{L_{12}, e_1, L_{11}, p\}$ de poids spécifique nul, et cela engendrerait de nouveau l'emploi du temps précédent, et le fait de sauvegarder le triplet $(L_{12}, L_{11}, 0)$ empêchera le choix d'un tel chemin.

(ii). La liste T_2

Nous prévoyons aussi une seconde liste taboue T_2 , dans laquelle nous mettrons les sommets du type L_{jk} récemment pris comme référence pour la recherche d'un chemin dans le graphe $R^*(\varphi)$.

B. Critère d'aspiration

Un triplet $(L_{j2k2}, L_{j1k1}, -p_s(C))$ de la liste T_1 perd son statut tabou si la solution qu'il engendre est meilleure que la meilleure solution retenue depuis le début du processus.

C. La fonction objectif

la fonction objectif, à minimiser, comptabilise le nombre de conflits total de l'emploi du temps généré

D. Procédure générale de résolution

Nous disposons d'un emploi du temps initial φ^0 sous forme d'un flot maximum de s à p sur le réseau R de valeur $\sum_{i=1}^{|ES|} |es_i|$. Nous définissons les quantités suivantes :

- φ^* : Représente le meilleur emploi du temps rencontré au cours de la procédure.
- φ : Représente l'emploi du temps en cours.
- Nimax : Nombre maximum d'itérations à faire avant d'arrêter le processus si la meilleure solution rencontrée n'est pas améliorée.
- Na : Nombre d'itérations faites sans que la meilleure solution ne soit améliorée.
- T_1, T_2 : Les deux listes taboues.
- $f(\varphi)$: Le nombre de conflits sur l'emploi du temps correspondant à φ .

(0) $\varphi^* := \varphi^0$; $\varphi := \varphi^0$; $N_a := 0$; $T_1 := \emptyset$; $T_2 := \emptyset$;

(1) **Tant que** ($(N_a < N_{\max})$ **et** $(f(\varphi^*) > 0)$) **faire**

Début

- $N_a := N_a + 1$;
- Choisir, au hasard, un sommet du type L_{jk} non tabou par rapport à la liste T_2 .
- Mettre à jour la liste tabou T_2 .
- Appliquer l'algorithme « PlusCourtSpécifique » à l'emploi du temps en cours :

Si le sommet P est marqué

Début

1. *Si* un circuit est détecté, φ représentera le nouveau emploi du temps engendré par ce circuit.

2. *Sinon*

Début

- φ représentera le nouveau emploi du temps donné par le meilleur chemin (*) non tabou ou qui satisfait le critère d'aspiration.
- Mettre à jour la liste tabou T_1 .

Fin ;

Fin ;

Sinon

- Choisir, au hasard, un sommet e_i marqué tel que l'arc (e_i, L_{jk}) existe. φ représentera le nouveau emploi du temps engendré par le circuit formé par le chemin de L_{jk} à e_i (grâce auquel e_i a été marqué) et l'arc (e_i, L_{jk}) .
- *Si* $(f(\varphi) < f(\varphi^*))$ **alors**

Début

$\varphi^* := \varphi$;

$N_a := 0$;

Fin ;

Fin ;

Fin de l'algorithme.

(*) : Lorsqu'un sommet L_{j1k1} est sélectionné pour la recherche d'un chemin C de ce dernier jusqu'au sommet p , alors dans C, p admet comme prédécesseur un sommet L_{j2k2} dont la marque (à la fin de l'application de l'algorithme « PlusCourtSpécifique ») est égale au poids spécifique de C (car les arcs adjacents à p dans $R^*(\varphi)$ ont un poids nul). Or, si le chemin C est tabou (par rapport à liste T_1), nous entendons par *meilleur chemin non tabou* le chemin C_n de L_{j1k1} à p non tabou dans lequel le prédécesseur L_{jnkn} de p porte la marque la plus petite.

Voici un exemple :

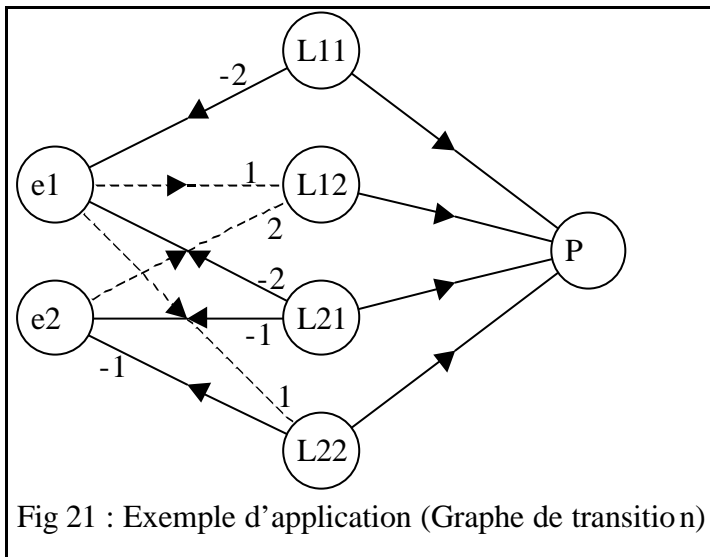


Fig 21 : Exemple d'application (Graphe de transition)

Si nous voulons déplacer une séance qui correspond au sommet L_{11} , alors nous supprimons l'arc (L_{11}, p) , puis nous appliquons l'algorithme « PlusCourtSpécifique » dont les itérations sont résumées dans le tableau ci-dessous :

Itérations k	L11	L12	L21	L22	e1	e2	P
0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	$+\infty$	$+\infty$	$+\infty$	-2	$+\infty$	$+\infty$
2	0	-1	$+\infty$	-1	-2	$+\infty$	$+\infty$
3	0	-1	$+\infty$	-1	-2	-2	-1
4	0	-1	$+\infty$	-1	-2	-2	-1

Tableau 6 : Itérations de l'algorithme « PlusCourtSpécifique » sur l'exemple de la fig 21

L'algorithme s'arrête à la 4^{ième} itération (la 3^{ième} et la 4^{ième} itération donnent les mêmes marquent aux sommets), et de la dernière ligne nous pouvons déduire qu'il y a deux chemins (mis à part L_{11} , il n'y a que deux sommets du type L_{jk} qui ont été marqués par une valeur finie, à savoir, L_{12} et L_{22}) de poids spécifique égal à (-1) et qui sont :

- Par rapport à L_{12} : $\{L_{11}, e_1, L_{12}\}$.
- Par rapport à L_{22} : $\{L_{11}, e_1, L_{22}\}$.

Si un des chemins ci-dessus est tabou par rapport à T_1 , c'est l'autre qui sera sélectionné et si les deux chemins sont tabous, c'est un autre sommet que L_{11} qui sera sélectionné à la prochaine itération, car L_{11} devient tabou par rapport à la liste T_2 , ce qui nous assure que les deux chemins ci-dessus ne seront pas engendrés à la prochaine itération.

IV.5. Une borne inférieure du nombre de conflits sur un emploi du temps

Nous décrivons, dans ce qui suit, une procédure permettant de donner une borne inférieure du nombre de conflits minimum que peut avoir un emploi du temps.

Le procédure en question se compose de deux étapes principales : par la première, nous calculons une borne inférieure $NBC1$ du nombre minimum de conflits *par rapport aux séances appartenant aux mêmes groupes de séances* (une borne inférieure pour nbc_1). La seconde étape nous permettra de calculer une borne inférieure $NBC2$ du nombre de conflits minimum par rapport aux *séances appartenant à des groupes de séances différents* mais relatifs à de mêmes groupes d'enseignement (une borne inférieure pour nbc_2).

La borne inférieure NBC du nombre de conflits total ne sera que la somme des deux quantités retrouvées à la fin de la première et de la seconde étape ($NBC=NBC1+NBC2$).

A. Calcul de $NBC1$

Désignons, dans ce qui suit, par (c..) un circuit ou un chemin de L_{jk} à P sur un graphe de transition.

La méthodologie que nous proposons ici est un procédé itératif permettant le passage d'un emploi du temps et_1 à un autre et_2 si le graphe de transition relatif à et_1 contient un (c..) de poids spécifique négatif, ceci nous garantit une amélioration du nombre de conflits nbc_1 tout au long de la procédure. Dès que cela n'est plus possible, la procédure s'arrête.

Cela peut être résumé par l'algorithme suivant :

Algorithme

Tant que (le graphe de transition de l'emploi du temps en cours contient un (c..) de poids spécifique négatif) faire

 Considérer le (c..) en question pour générer le nouveau emploi du temps.

Fin de tant que

Fin de l'algorithme

Notons ici que le nombre d'itérations de cet algorithme est au plus égal au nombre de conflits sur les mêmes groupes de séances de l'emploi du temps initial qui est aussi borné par le nombre total de séances considérées.

En pratique, pour appliquer l'algorithme ci-dessus, nous pourrions ajouter au graphe de transition un sommet p_0 qui sera relié à tous les sommets $\{L_{jk}\}$ par des arcs $\{(p_0, L_{jk})\}$ de poids nuls puis appliquer l'algorithme « PlusCourtSpécifique » à partir du sommet p_0 . le procédé s'arrête dès que p est marqué par une valeur positive ou nulle (indiquant ainsi la non existence d'un (c..) de poids spécifique négatif).

Une fois que nous aurons retrouvé un tel emploi du temps, nous pourrions évaluer une borne inférieure $NBC1$ du nombre de conflits relatifs aux mêmes groupes de séances.

Pour ce faire, nous définissons pour chaque sommet e_i du graphe de transition trois ensembles de périodes P_i^1 , P_i^2 et P_i^3 qui correspondent aux trois cas utilisés pour affecter les poids aux arcs du graphe de transition, c'est à dire :

P_i^1 : Contiendra les périodes pour lesquelles il y a un unique arc entrant à e_i (nous avons affecté une unique séance de e_i à chacune de ces périodes).

P_i^2 : Contiendra les périodes pour lesquelles il y a au moins deux arcs entrants à e_i (nous avons affecté plus d'une séance de e_i à chacune de ces périodes).

P_i^3 : Contiendra les périodes pour lesquelles il n'y a que des arcs sortants de e_i (nous n'avons affecté aucune séance de e_i à chacune de ces périodes).

Nous considérons, à présent, un emploi du temps et* dont le graphe de transition R^* ne contient pas de (c..) de poids spécifique négatif.

Examinons pour chaque sommet e_i , la possibilité d'améliorer le nombre de conflits par rapport à ce dernier :

Cas a : $P_i^2 = \bar{A}$ et $P_i^3 = \bar{A}$

Donc e_i n'est adjacent qu'à des sommets qui correspondent à des périodes de P_i^1 .

Cela veut dire :

- A chaque période où *il est possible d'affecter* une séance de e_i est affectée *une unique* séance.

Il est clair dans ce cas que toutes les séances de e_i sont affectées sans conflits, il n'est donc pas possible de diminuer le nombre de conflits sur ce genre de sommets.

Cas b : $P_i^2 = \bar{A}$ et $P_i^3 \neq \bar{A}$

Donc e_i n'est adjacent qu'à des sommets qui correspondent à des périodes de P_i^1 ou P_i^3 .

Cela veut dire :

- A chaque période où *il est possible d'affecter* une séance de e_i est affectée, *au plus*, *une unique* séance de ce dernier.

Il est clair dans ce cas que toutes les séances de e_i sont affectées sans conflits, il n'est donc pas possible de diminuer le nombre de conflits sur ce genre de sommets.

Cas c : $P_i^2 \neq \bar{A}$ et $P_i^3 = \bar{A}$

Donc e_i n'est adjacent qu'à des sommets qui correspondent à des périodes de P_i^1 ou P_i^2 .

Cela veut dire :

- A chaque période où *il est possible d'affecter* une séance de e_i est affectée, *au moins*, *une* séance de ce dernier.

Il est clair que si pour un sommet e_i il y a un nombre de séances supérieur ou égal au nombre de périodes sur lesquelles elles peuvent être affectées, la meilleure manière (un nombre de conflits minimum) d'affecter ces séances serait d'assigner au moins une séance à chaque période.

Il n'est donc pas possible d'améliorer le nombre de conflits sur ce genre de sommets.

Cas d : $P_i^2 \neq \bar{A}$ et $P_i^3 \neq \bar{A}$

Notons ici que le déplacement d'une des séances relatives à P_i^2 à P_i^3 n'est pas possible car cela induirait un chemin de poids spécifique négatif.

Cela dit, une amélioration du nombre de conflits sur ce type de sommet est éventuellement possible.

D'après les quatre cas cités ci-dessus, nous ne pourrions améliorer le nombre de conflits que sur des sommets étant dans le cas d.

Nous désignons par $nbsei$ le nombre de séances relatives à e_i qui se déroulent aux périodes P_i^2 et par $nbci^*$ le nombre de conflits par rapport aux mêmes groupes de séances sur l'emploi du temps t^* .

Dans le meilleur cas de figure, le nombre de conflits sur ce type de sommets e_i (sommets e_i vérifiant le cas d) ne peut être réduit que d'au plus :

$$D_i = \text{Min} (nbsei - |P_i^2|, |P_i^3|).$$

Car le nombre de séances qui créent un conflit sur les périodes P_i^2 est $nbsei - |P_i^2|$ et le nombre de périodes par rapport à e_i où nous n'avons affecté encore aucune autre séance de ce dernier est $|P_i^3|$.

Dans ce cas, il est clair que le nombre $NBC1$ suivant est bien la borne inférieure recherchée :

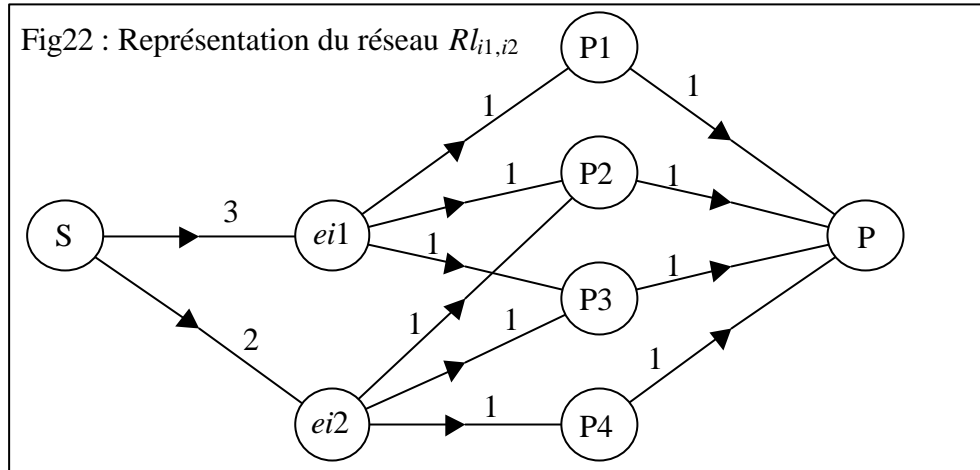
$$NBC1 = \text{Max}(0, nbci^* - \sum_{i / e_i \text{ vérifie le cas d}} D_i)$$

Bien entendu, si aucun sommet e_i n'est dans le cas d, nous aurons un emploi du temps avec un nombre de conflits par rapport aux mêmes groupes de séances *minimum*.

B. Calcul de $NBC2$

Pour cela, nous calculerons une borne inférieure $NBC2_1$ du nombre de conflits relatifs aux différents groupes de séances pour chaque groupe d'enseignement ge_l . La somme de ces quantités sur tous les groupes d'enseignement sera la borne inférieure $NBC2$ recherchée sur tout l'emploi du temps.

Considérons, donc, un groupe d'enseignement ge_l et tous ses groupes de séances. Pour chaque couple $\{esl_{i1}, esl_{i2}\}$ de groupes de séances relatifs à ge_l , nous construisons un réseau $Rl_{i1,i2}$ de la manière suivante :



L'ensemble des sommets comporte :

- Une source S et un puits P.
- Deux sommets e_{i1} , e_{i2} qui correspondent aux groupes de séances esl_{i1} , esl_{i2} respectivement.
- Un ensemble de périodes $\{P_k\}$ auxquelles il est possible d'affecter au moins une séance du groupe d'enseignement ge_l .

L'ensemble des arcs ainsi que les poids sur ces derniers sont définis de la manière suivante :

- Les arcs (S, e_{in}) de poids N_{in} où, N_{in} est le nombre *minimum* de périodes auxquelles les séances de e_{in} peuvent être affectées ($n \in \{1,2\}$). Pour évaluer le nombre N_{in} il suffit d'ordonner les périodes par ordre décroissant du nombre de types de salles disponibles auxquelles il est possible d'affecter les séances de esl_{in} . Il faudra ensuite considérer les premières périodes pour lesquelles la somme du nombre de types de salles disponibles est supérieure ou égale au nombre de séances représentées par esl_{in} .
- Il y a un arc (e_{in}, P_k) de poids égal à l'unité si au moins une séance de esl_{in} peut être affectée à la période P_k ($n \in \{1,2\}$).
- Il y a un arc (P_k, P) de poids égal à l'unité pour tout sommet P_k .

Soit $CF_{l_{i1,i2}}$ la valeur d'un flot maximum de S à P sur $Rl_{i1,i2}$. Le nombre de conflits par rapport aux deux groupes de séances esl_{i1} , esl_{i2} est au moins égal à :

$$NBC2_{l_{i1,i2}} = (N_{i1} + N_{i2}) - CF_{l_{i1,i2}}.$$

$$\text{Ainsi : } NBC2_l = \sum_{\{i1,i2 / esl_{i1} \text{ et } esl_{i2} \text{ sont des groupes de séances de } ge_l\}} NBC2_{l_{i1,i2}}$$

Et enfin, la borne inférieure $NBC2$ du nombre de conflits relatifs aux différents groupes de séances sur tout l'emploi du temps sera égale à :

$$NBC2 = \sum_{\{l / ge_l \text{ est un groupe d'enseignement}\}} NBC2_l .$$

Chapitre 5

Implémentation et résultats

V.1. Introduction

Nous présentons, dans ce qui suit, une application de la méthode de résolution proposée pour la construction d'un emploi du temps au niveau de la Faculté des Sciences Mathématiques à l'USTHB. Nous proposerons, par la suite, un algorithme simple permettant de l'améliorer.

Les données utilisées sont celles de l'année universitaire 2000/2001.

Pour cela, nous avons considéré l'ensemble des modules des sections de Recherche Opérationnelle (RO) et ceux des DES dont le nombre s'élève à 57 modules et pour lesquels il faudra affecter un total de 201 séances.

Les emplois du temps à l'USTHB comportent 36 créneaux horaires (36 périodes élémentaires) d'une heure et demi chacun qui s'étalent sur 6 jours de la semaine (6 créneaux par jour). Nous avons constaté, à travers les emplois du temps établis, que le dernier créneau de chaque jour n'est jamais utilisé et donc, le déroulement des séances en ces périodes n'est pas souhaitable. C'est pourquoi, Nous n'avons considéré que les 30 périodes élémentaires restantes.

La Faculté des Sciences Mathématiques n'utilise, pour affecter ces séances, que des salles de capacité dites simples ($40 \times 1 = 40$) ou doubles ($40 \times 2 = 80$). Etant donné que l'ensemble des salles de ce type ne sont pas exclusivement utilisées que par la Faculté des Sciences Mathématiques, il est difficile de fixer un certain ensemble de salles comme étant le « cota » de la faculté. Pour cela, nous n'avons considéré que les salles disponibles dans les créneaux figurants sur les emplois du temps affectés à ces sections durant l'année universitaire 2000/2001. Bien que cela réduit énormément l'ensemble des salles disponibles, mais cela nous garantit de faire un choix assez réaliste du nombre de salles disponibles de chaque type et à chaque période.

V.2. Implémentation

Pour mettre en œuvre et tester l'approche de résolution proposée, nous avons élaboré un programme nommé « MTimeTabler » qui a été écrit avec le langage orienté objet C++ sous « Windows 95 » avec Borland C++Builder 5.0 (dernière version apparue en 2000). Cet outil présente une interface de développement RAD (Rapid Application Development) et a largement fait ses preuves en matière de développement de logiciels car il intègre, à la fois, toute la puissance du langage C++, la convivialité du système d'exploitation « Windows » et offre une bibliothèque d'objets riches, puissants et facilement adaptables aux besoins de l'application.

« MTimeTabler » a été programmé et testé sur une machine dotée d'un processeur 133Mhz et offrant 80Mo de RAM.

V.3. Données utilisées

Nous présentons ici l'ensemble des données utilisées pour le problème considéré afin de construire un emploi du temps avec le programme « MTimeTabler ». Ces données se composent :

- De l'ensemble des groupes d'enseignement, modules (ou cours) étudiés, nombre de créneaux nécessaires pour chaque module et le nombre d'étudiants dans chaque groupe d'enseignement.
- Du nombre de salles de chaque type (simple et double) disponibles sur chacun des 30 créneaux considérés.

V.3.1. Modules et groupes d'enseignement

Ces informations figurent dans le tableau 1 ci-dessous qui est structuré de la manière suivante :

- *Année* : L'année d'étude.
- *Spécialité* : La spécialité ou branche des groupes d'enseignement.
- *Groupes* : Le code du groupe.
 - G0 : Identifie toute la section (G1+G2).
 - G1 : Identifie le premier groupe.
 - G2 : Identifie le deuxième groupe.
- *Modules* : Codes des modules (ou cours) étudiés.
- *Nbc* : Le nombre de séances (ou créneaux) nécessaires à l'étude d'un module donné correspondant à un groupe donné.
- *Nb_Etudiants* : Le nombre d'étudiants qui composent chaque groupe.

Nous avons procédé au groupement de ces séances suivant le tableau 2 ci dessous.

V.3.2. Nombre de salles disponibles de chaque type

Nous décrivons sur les deux tableaux 3 et 4 l'ensemble des salles disponibles de chaque type et à chacune des 30 périodes considérées.

Chaque case sur les deux tableaux contient le nombre de salles disponibles au jour et au créneau qui lui correspondent.

V.4. Résultats obtenus

L'application de « MTimeTabler » sur ces données nous a permis, en deux (2) secondes, de construire un emploi sans conflits. Ce dernier est représenté sur le tableau 5 ci-dessous.

Les valeurs choisies pour les paramètres utilisés pour la construction de cet emploi du temps sont les suivantes : $N_{\max} = 5000$, taille de la liste $T_1 = 5$, taille de la liste $T_2 = 30$.

Tableau1 : Modules et groupes d'enseignement					
Année	Specialite	Groupe	Module	Nbc	Nb_Etudiants
2	RO	G0	Tec611	2	86
2	RO	G1	Tec611	2	43
2	RO	G2	Tec611	2	43
2	RO	G0	TM012	1	86
2	RO	G1	TM012	1	43
2	RO	G2	TM012	1	43
2	RO	G0	Tec610	2	86
2	RO	G1	Tec610	2	43
2	RO	G2	Tec610	2	43
2	RO	G0	Sem310	2	86
2	RO	G1	Sem310	2	43
2	RO	G2	Sem310	2	43
2	RO	G0	TM06	1	86
2	RO	G1	TM06	1	43
2	RO	G2	TM06	1	43
2	RO	G0	Tec599	1	86
2	RO	G1	Tec599	1	43
2	RO	G2	Tec599	1	43
2	RO	G0	Arabe	1	86
3	RO	G0	Tec630	2	50
3	RO	G1	Tec630	2	25
3	RO	G2	Tec630	2	25
3	RO	G0	Teg40	1	50
3	RO	G1	Teg40	1	25
3	RO	G2	Teg40	1	25
3	RO	G0	Tec633	2	50
3	RO	G1	Tec633	2	25
3	RO	G2	Tec633	2	25
3	RO	G0	Tec632	2	50
3	RO	G1	Tec632	2	25
3	RO	G2	Tec632	2	25
3	RO	G0	Tec631	2	50
3	RO	G1	Tec631	2	25
3	RO	G2	Tec631	2	25
3	RO	G0	Anglais/Arabe	1	50
4	RO	G0	PM	1	51
4	RO	G1	PM	1	25
4	RO	G2	PM	1	26
4	RO	G0	OC	2	51
4	RO	G1	OC	2	25
4	RO	G2	OC	2	26
4	RO	G0	PAAP	2	51
4	RO	G1	PAAP	2	25
4	RO	G2	PAAP	2	26
4	RO	G0	FBD	1	51
4	RO	G1	FBD	1	25
4	RO	G2	FBD	1	26
4	RO	G0	MODELI	1	51
4	RO	G0	Anglais	1	51
4	RO	G0	Arabe	1	51
5	RO	G0	NeoTech	4	59
2	DESLN	G0	Sem330	4	12
2	DESLN	G0	Sem302	4	12
2	DESLN	G0	Sem301	4	12

Tableau1 : Modules et groupes d'enseignement					
Année	Specialite	Groupe	Module	Nbc	Nb_Etudiants
2	DESLF	G0	Sem330	4	37
2	DESLF	G0	Sem302	4	37
2	DESLF	G0	Sem301	4	37
3	DESLN	G0	Sem305	4	16
3	DESLN	G0	Sem340	2	16
3	DESLN	G0	Sem307	4	16
3	DESLN	G0	Sem304	2	16
3	DESLN	G0	Sem331	2	16
3	DESLF	G0	Sem305	4	41
3	DESLF	G0	Sem340	2	41
3	DESLF	G0	Sem307	4	41
3	DESLF	G0	Sem304	2	41
3	DESLF	G0	Sem331	2	41
4	DESANA	G0	Sem324	4	13
4	DESANA	G0	Sem306	4	13
4	DESANA	G0	ANA1	4	13
4	DESANA	G0	ANA2	4	13
4	DESANA	G0	Anglais	1	13
4	DESRO	G0	Sem306	4	22
4	DESRO	G0	Sem324	4	22
4	DESRO	G0	RO1	4	22
4	DESRO	G0	RO2	4	22
4	DESRO	G0	Anglais	1	22
4	DESPS	G0	Sem306	4	11
4	DESPS	G0	Sem324	4	11
4	DESPS	G0	PS1	4	11
4	DESPS	G0	PS2	4	11
4	DESPS	G0	Anglais	1	11
4	DESALGEBRE	G0	Sem324	4	13
4	DESALGEBRE	G0	Sem306	4	13
4	DESALGEBRE	G0	Algèbre1	4	13
4	DESALGEBRE	G0	Algèbre2	4	13
4	DESALGEBRE	G0	Anglais	1	13

Tableau 2: Groupement des séances			
Indices	Groupes	Nombre de séances	Signification
1	RO2	10	Ensemble des cours de 2 ^{ème} année RO
2	RO3	10	Ensemble des cours de 3 ^{ème} année RO
3	RO4	9	Ensemble des cours de 4 ^{ème} année RO
4	RO5	4	Ensemble des cours de 5 ^{ème} année RO
5	RO2G1	9	Ensemble des travaux dirigés du groupe G1 de 2 ^{ème} année RO
6	RO2G2	9	Ensemble des travaux dirigés du groupe G2 de 2 ^{ème} année RO
7	RO3G1	9	Ensemble des travaux dirigés du groupe G1 de 3 ^{ème} année RO
8	RO3G2	9	Ensemble des travaux dirigés du groupe G2 de 3 ^{ème} année RO
9	RO4G1	6	Ensemble des travaux dirigés du groupe G1 de 4 ^{ème} année RO
10	RO4G2	6	Ensemble des travaux dirigés du groupe G2 de 4 ^{ème} année RO
11	DESLN2	12	Ensemble des cours et travaux dirigés de 2 ^{ème} année DES LN
12	DESLF2	12	Ensemble des cours et travaux dirigés de 2 ^{ème} année DES LF
13	DESLN3	14	Ensemble des cours et travaux dirigés de 3 ^{ème} année DES LN
14	DESLF3	14	Ensemble des cours et travaux dirigés de 3 ^{ème} année DES LF
15	DESANA4	17	Ensemble des cours et travaux dirigés de 4 ^{ème} année DES ANA
16	DESRO4	17	Ensemble des cours et travaux dirigés de 4 ^{ème} année DES RO
17	DESPS4	17	Ensemble des cours et travaux dirigés de 4 ^{ème} année DES PS
18	DESALGEBRE4	17	Ensemble des cours et travaux dirigés de 4 ^{ème} année DES ALGEBRE

Tableaux 3 et 4 : Nombre de salles disponibles de chaque type

Tableau 3 : Nombre de salles doubles disponibles à chaque période

	8h - 9h30	9h40 - 11h10	11h20 - 12h50	13h - 14h30	14h40 - 16h10	16h20 - 17h50
Samedi	0	2	2	2	1	
Dimanche	0	0	2	2	3	
Lundi	2	1	1	0	0	
Mardi	2	1	0	1	2	
Mercredi	2	3	0	1	1	
Jeudi	0	0	1	1	0	

Tableau 4 : Nombre de salles simples disponibles à chaque période

	8h - 9h30	9h40 - 11h10	11h20 - 12h50	13h - 14h30	14h40 - 16h10	16h20 - 17h50
Samedi	6	8	8	8	6	
Dimanche	4	4	5	4	1	
Lundi	4	5	8	10	8	
Mardi	9	8	10	6	2	
Mercredi	5	6	8	8	2	
Jeudi	5	6	2	2	0	

Tableau 5 : Emploi du temps obtenu

		Types de salles	
		Salles doubles	Salles simples
Périodes	Période 1		[RO2G2] [RO3G1] [RO3G2] [DESALGEBRE4] [RO4G2] [DESPS4]
	Période 2	[RO3] [RO5]	[RO2G2] [DESALGEBRE4] [RO4G1] [RO4G2] [DESLF2] [DESLN3] [DESLF3] [DESRO4]
	Période 3	[RO2] [RO3]	[RO4G1] [RO4G2] [DESLN2] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4]
	Période 4	[RO2] [RO3]	[RO4G1] [RO4G2] [DESLN2] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4]
	Période 5	[RO2]	[RO3G1] [DESALGEBRE4] [RO4G1] [RO4G2] [DESLN2] [DESLF2]
	Période 6		[RO2G2] [RO3G1] [DESALGEBRE4] [DESPS4]
	Période 7		[RO2G2] [RO3G1] [DESALGEBRE4] [DESPS4]
	Période 8	[RO2] [RO3]	[RO4G1] [RO4G2] [DESLN2] [DESLF2] [DESPS4]
	Période 9	[RO2] [RO3]	[DESLN2] [DESLF2] [DESLN3] [DESLF3]
	Période 10	[RO2] [RO3] [RO4]	[DESLN2]
	Période 11	[RO2] [RO3]	[DESLN2] [DESLF2] [DESLN3] [DESLF3]
	Période 12	[RO2]	[DESALGEBRE4] [RO3G2] [DESLN2] [DESLF2] [DESLN3]
	Période 13	[RO2]	[DESALGEBRE4] [RO3G2] [DESLN2] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4]
	Période 14		[DESALGEBRE4] [RO3G1] [RO3G2] [DESLN2] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4]
	Période 15		[DESALGEBRE4] [RO3G1] [RO3G2] [DESLN2] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4]
	Période 16	[RO3] [RO4]	[RO2G1] [DESALGEBRE4] [RO2G2] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4]
	Période 17	[RO3]	[RO2G1] [DESALGEBRE4] [RO2G2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4]
	Période 18		[RO2G1] [DESALGEBRE4] [RO2G2] [RO3G1] [RO3G2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4]
	Période 19	[RO3]	[RO2G1] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4]
	Période 20	[RO4] [RO5]	[DESLN3] [DESLF3]
	Période 21	[RO4] [RO5]	[RO2G1] [DESLF3] [DESANA4] [DESRO4] [DESPS4]
	Période 22	[RO2] [RO4] [RO5]	[DESALGEBRE4] [RO3G1] [RO3G2] [DESANA4] [DESRO4] [DESPS4]
	Période 23		[DESALGEBRE4] [RO2G1] [RO2G2] [RO3G1] [RO3G2] [DESANA4] [DESRO4] [DESPS4]
	Période 24	[RO4]	[DESALGEBRE4] [RO2G1] [RO2G2] [RO3G2] [DESLN2] [DESANA4] [DESRO4] [DESPS4]
	Période 25	[RO4]	[DESANA4] [DESRO4]
	Période 26		[DESALGEBRE4] [RO2G1] [DESANA4] [DESRO4] [DESPS4]
	Période 27		[DESALGEBRE4] [RO2G1] [RO4G1] [DESANA4] [DESRO4] [DESPS4]
	Période 28	[RO4]	[DESANA4] [DESRO4]
	Période 29	[RO4]	[DESANA4] [DESRO4]
	Période 30		

Autres tests

Afin de mesurer d'avantage l'approche de résolution proposée, nous avons voulu tester le programme sur deux autres cas où certaines propriétés des emplois du temps à construire sont prévisibles.

Pour ce faire, nous avons modifié le nombre de salles doubles disponibles des deux manières suivantes :

Cas 1 : A la 10^{ième} période, 4 salles seront disponibles (au lieu de 3) et à la 22^{ième}, 2 salles seulement seront disponibles (au lieu de 3).

Cas 2 : A la 10^{ième} période, 5 salles seront disponibles (au lieu de 3), à la 22^{ième} période, 2 salles seulement seront disponibles (au lieu de 3) et à la 29^{ième} période, aucune salle ne sera disponible (au lieu de 1).

Notons ici que dans les deux cas, nous avons toujours 33 salles doubles disponibles sur les 30 périodes pour 33 séances de RO2, RO3, RO4 et RO5 qui ne peuvent se dérouler que dans ces types de salles.

Dans le premier cas, il est clair que l'emploi du temps recherché doit affecter 4 séances à la 10^{ième} période (une séance à chaque salle double) ; comme il ne peut y avoir de conflits, il y aura forcément une séance de chacun des groupes RO2, RO3, RO4 et RO5 affectée à cette période. L'emploi du temps donné par le programme et illustré sur le tableau 6 donne bien ce résultat.

Dans le second cas, il est clair que si un emploi du temps sans conflits existe, il doit affecter aux 5 salles doubles de la 10^{ième} période, 5 séances des groupes RO2, RO3, RO4 et RO5 sans conflits, ce qui est impossible. Le programme signale bien, dans ce cas, qu'il n'est pas possible d'affecter sans conflits ces séances.

Tableau 6 : Emploi du temps obtenu dans le cas 1

Types de salles	
Salles doubles	Salles simples
Période 1	[RO2G2] [RO3G1] [RO3G2] [DESALGEBRE4] [DESLN2] [DESPS4]
Période 2	[RO3] [RO4]
Période 3	[RO2] [RO3]
Période 4	[RO2] [RO3]
Période 5	[RO2]
Période 6	[RO2G1] [DESALGEBRE4] [RO4G2] [DESLN2] [DESLF2] [DESLN3]
Période 7	[RO2G2] [RO3G1] [DESALGEBRE4] [DESPS4]
Période 8	[RO2] [RO3]
Période 9	[RO2] [RO3]
Période 10	[RO2] [RO3] [RO4] [RO5]
Période 11	[RO2] [RO3]
Période 12	[RO2]
Période 13	[RO2]
Période 14	[DESALGEBRE4] [RO3G1] [RO3G2] [DESLN2] [DESLF2] [DESLN3] [DESALGEBRE4] [DESRO4] [DESPS4]
Période 15	[DESALGEBRE4] [RO3G1] [RO3G2] [RO4G2] [DESLF2] [DESLN3] [DESALGEBRE4] [DESRO4]
Période 16	[RO3] [RO4]
Période 17	[RO3]
Période 18	[RO2G1] [DESALGEBRE4] [RO2G2] [RO3G1] [RO3G2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4]
Période 19	[RO3]
Période 20	[RO4] [RO5]
Période 21	[RO4] [RO5]
Période 22	[RO4] [RO5]
Période 23	[DESALGEBRE4] [RO2G1] [RO2G2] [RO3G1] [RO3G2] [DESANA4] [DESRO4] [DESPS4]
Période 24	[RO4]
Période 25	[RO4]
Période 26	[DESALGEBRE4] [RO2G1] [DESANA4] [DESRO4] [DESPS4]
Période 27	[RO2G1] [RO4G1] [RO4G2] [DESANA4] [DESRO4] [DESPS4]
Période 28	[RO4]
Période 29	[RO2]
Période 30	[DESANA4] [DESRO4]

V.5. Amélioration des résultats

Si nous considérons l'emploi du temps initial illustré sur le tableau 5, nous pouvons facilement constater qu'il y a une mauvaise *répartition* des séances sur la période de référence considérée. En fait, en relançant le programme à plusieurs reprises, nous pouvons facilement obtenir une meilleure répartition des séances.

Nous décrivons, dans ce qui suit, une procédure simple qui, à partir d'un emploi du temps donné, permet de faire une série de permutations entre les affectations pour mieux répartir l'ensemble des séances sans pour autant violer les contraintes satisfaites par l'emploi du temps obtenu initialement.

Pour ce faire, nous définirons par rapport à chaque groupe de séances es_i , une quantité que nous appellerons « Indice de répartition » qui mesurera la répartition des séances de es_i sur toute la période de référence.

Soient :

$|es_i|$: Le nombre de séances du groupe es_i .

$|P|$: Le nombre de périodes élémentaires sur la période de référence.

$\{s_j^i / j \in \{1, \dots, (|es_i|-1)\}\}$: L'ensemble des nombres de périodes élémentaires séparant deux périodes successives auxquelles sont affectées des séances de es_i .

Il est clair qu'une répartition « idéale » des séances de es_i sur toute la période de référence rend le nombre moyen de périodes élémentaires (ou de créneaux) entre deux affectations

successives égal à $R_i = \frac{|P| - |es_i|}{|es_i| - 1}$ (avec $|es_i| \geq 1$).

Ainsi, nous définirons l'indice de répartition IR_i par rapport au groupe de séances es_i de la manière suivante :

$$IR_i = \sum_{l=1}^{|es_i|-1} |s_l^i - R_i|$$

Il est évident que plus l'indice IR_i est réduit (s'approche de 0) plus la répartition des séances de es_i sur la période de référence sera meilleure (si $|es_i|=1$, nous considérerons que $IR_i=0$).

Nous définissons, par la suite, l'*indice de répartition* IR de toutes les séances considérées (de tous les groupes de séances) caractérisant un emploi du temps de la manière suivante :

$$IR = \sum_{i=1}^{|ES|} IR_i \text{ ou } |ES| \text{ est le nombre de groupes de séances.}$$

La quantité IR doit être, donc, minimisée sur un emploi du temps donné afin d'avoir une meilleure répartition des séances.

Pour cela, nous proposons un algorithme itératif qui, à partir d'un emploi du temps de départ, génère plusieurs emplois du temps satisfaisant l'ensemble des contraintes satisfaites par l'emploi initial.

Le passage d'un emploi du temps à un autre se fera grâce à une série de déplacements des séances de sorte que cela n'engendre aucun conflit supplémentaire et que le nombre de salles occupées de chaque type et à chaque période reste inchangé.

Ceci revient à générer des emplois du temps en ne considérant que des *circuits* sur le graphe de transition. À titre d'exemple, nous donnons ci-dessous quelques circuits ainsi que les changements des affectations qu'ils impliquent :

Circuits	Affectations initiales	Nouvelles affectations
$\{L_{12}, e_2, L_{21}, e_1, L_{12}\}$	$e_1 \rightarrow L_{21}$ $e_2 \rightarrow L_{12}$	$e_1 \rightarrow L_{12}$ $e_2 \rightarrow L_{21}$
$\{L_{11}, e_1, L_{12}, e_2, L_{22}, e_3, L_{11}\}$	$e_1 \rightarrow L_{11}$ $e_2 \rightarrow L_{12}$ $e_3 \rightarrow L_{22}$	$e_1 \rightarrow L_{12}$ $e_2 \rightarrow L_{22}$ $e_3 \rightarrow L_{11}$

Pour avoir plus de chances de sélectionner rapidement des circuits qui n'engendrent pas de conflits supplémentaires, nous avons exploité encore une fois l'algorithme « PlusCourtSpécifique » de la manière suivante :

Algorithme : Sélection de circuit sans conflit

Max_iter : Nombre maximum d'itérations à faire sans pouvoir sélectionner un circuit avant d'arrêter la procédure.

Nb_iter : Nombre d'itérations en cours.

(0) $Nb_iter=0$;

(1) **Si** ($Nb_iter \geq Max_iter$) il n'y pas de circuit sélectionné. **Fin**.

Sinon

Choisir un sommet L_{jk} au hasard et appliquer l'algorithme « PlusCourtSpécifique » à partir de ce dernier. Nous pouvons, à la fin de cette étape, décrire un chemin de L_{jk} à tous les sommets du type e_i marqués.

(2) Choisir au hasard un sommet e_i marqué tel que l'arc (e_i, L_{jk}) existe et le circuit C formé par le chemin de L_{jk} à e_i et l'arc (e_i, L_{jk}) n'engendre aucun conflit supplémentaire.

(3) **Si** un tel circuit C est trouvé **alors** C sera le circuit recherché. **Fin**.

Sinon

$Nb_iter=Nb_iter+1$;

Aller en (1) ;

Fin algorithme.

Nous considérons aussi l'algorithme « Sélection de circuit quelconque » qui est similaire à « Sélection de circuit sans conflit » mais qui ne vérifie pas si le circuit sélectionné engendre ou non des conflits supplémentaires. L'apport principal de cet algorithme réside dans le fait qu'il peut engendrer, après l'avoir appliqué à plusieurs reprises, des déplacements sans conflits supplémentaires que l'algorithme « Sélection de circuit sans conflit » ne pourrait faire.

Ainsi, pour améliorer la répartition des séances, nous proposons l'algorithme suivant :

Algorithme : Améliorer la répartition

Soient :

et^0 : L'emploi du temps initial.

IR^0 : L'indice de répartition de l'emploi du temps et^0 .

et^* : Meilleur emploi du temps obtenu (qui donne la meilleure répartition des séances).

IR^* : Meilleur indice de répartition obtenu (celui de et^*).

et : Emploi du temps de l'itération en cours.

IR : L'indice de répartition de l'emploi du temps et en cours.

et_temp : Un emploi du temps temporaire.

$NbConf$: Le nombre de conflits associé à l'emploi du temps et en cours.

$NbConfTemp$: Le nombre de conflits associé à l'emploi du temps et_temp .

N_iter : Nombre d'itérations faites sans améliorer l'indice de répartition IR^* .

N_max : Nombre maximum d'itérations à faire sans améliorer l'indice de répartition avant d'arrêter la procédure.

N_iter1 : Nombre d'itérations faites pour générer des circuits avec l'algorithme « Sélection de circuit quelconque ».

N_max1 : Nombre maximum d'itérations à faire pour générer des circuits avec l'algorithme « Sélection de circuit quelconque ».

(0) $et = et^* = et^0$;
 $IR = IR^* = IR^0$;
 $N_iter = 0$;

(1) **Tant que** ($N_iter < N_max$) **faire**

- **Avec une probabilité p** : Sélectionner un circuit du graphe de transition correspondant à l'emploi du temps et en cours par l'algorithme « Sélection de circuit sans conflit ». et devient le nouveau emploi du temps engendré par ce dernier.
- **Avec une probabilité $(1-p)$** :
 - Poser $et_temp = et$, $N_iter1 = 0$;
 - **faire**
 - $N_iter1 = N_iter1 + 1$;
 - Sélectionner un circuit du graphe de transition correspondant à l'emploi du temps et_temp par l'algorithme « Sélection de circuit quelconque ». et_temp devient le nouveau emploi du temps engendré par ce dernier.

Tant que ($NbConfTemp > NbConf$ **et** $N_iter1 < N_max1$)

- **Si** ($NbConfTemp \leq NbConf$) $et = et_temp$;
- $N_iter = N_iter + 1$;
- **Si** ($IR < IR^*$) **alors**
 - $IR^* = IR$;
 - $et^* = et$;
 - $N_iter = 0$;

Fin si

Fin tant que

Fin algorithme

L'application de l'algorithme « Améliorer la répartition » sur l'emploi du temps donné sur le tableau 5 (dont l'indice de répartition est 279) a permis l'obtention de l'emploi du temps illustré sur le tableau 7 (dont l'indice de répartition est 150). Il est clair que la répartition des séances sur ce dernier est nettement meilleure. Les valeurs retenues pour les paramètres de l'algorithme sont les suivantes : $N_max = 200$, $p = 0.7$, $N_max1 = 10$.

Notons ici que l'algorithme en question peut être appliqué autant de fois que nécessaire jusqu'à l'obtention d'une répartition assez satisfaisante.

Tableau 7 : Emploi du temps obtenu après application de l'algorithme <Améliorer la répartition>

		Types de salles	
		Salles doubles	Salles simples
Période 1			[RO2G1] [RO4G1] [RO4G2] [DESLN2] [DESLF2] [DESANA4]
Période 2	[RO4] [RO5]		[RO2G1] [RO3G1] [RO3G2] [DESLN2] [DESLN3] [DESLF3] [DESANA4] [DESRO4]
Période 3	[RO3] [RO4]		[RO2G2] [DESLN2] [DESLF2] [DESLN3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 4	[RO3] [RO4]		[DESLN2] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 5	[RO3]		[RO2G1] [DESLN3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 6			[RO2G2] [RO4G1] [DESLN2] [DESANA4]
Période 7			[RO3G1] [RO3G2] [RO4G2] [DESLN2]
Période 8	[RO2] [RO3]		[DESLF2] [DESLN3] [DESANA4] [DESPS4] [DESALGEBRE4]
Période 9	[RO3] [RO4]		[RO2G1] [RO2G2] [DESLN2] [DESALGEBRE4]
Période 10	[RO2] [RO4] [RO5]		[RO3G2]
Période 11	[RO2] [RO3]		[DESLN2] [DESLF3] [DESANA4] [DESRO4]
Période 12	[RO2]		[RO4G1] [DESLN2] [DESLN3] [DESRO4] [DESALGEBRE4]
Période 13	[RO2]		[RO3G1] [RO3G2] [DESLN2] [DESLF2] [DESLF3] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 14			[RO2G1] [RO2G2] [RO3G1] [RO4G2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 15			[RO2G2] [RO3G2] [DESLN2] [DESLF2] [DESLN3] [DESANA4] [DESRO4] [DESPS4]
Période 16	[RO3] [RO4]		[RO2G1] [DESLN2] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 17	[RO2]		[RO3G1] [RO3G2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 18			[RO2G1] [RO2G2] [RO4G1] [DESLF2] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 19	[RO2]		[RO3G1] [RO3G2] [RO4G2] [DESANA4] [DESRO4] [DESPS4]
Période 20	[RO2] [RO4]		[DESLF2] [DESLF3]
Période 21	[RO3] [RO4]		[RO2G1] [RO2G2] [DESLN3] [DESPS4] [DESALGEBRE4]
Période 22	[RO2] [RO4] [RO5]		[RO3G1] [DESLF2] [DESLF3] [DESANA4] [DESPS4] [DESALGEBRE4]
Période 23			[RO3G2] [RO4G1] [DESLN3] [DESLF3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 24	[RO3]		[RO2G2] [RO4G2] [DESLF2] [DESLF3] [DESANA4] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 25	[RO2]		[RO3G1] [DESLN3]
Période 26			[DESLF2] [DESLF3] [DESRO4] [DESPS4] [DESALGEBRE4]
Période 27			[RO2G1] [RO2G2] [RO3G1] [RO3G2] [DESPS4] [DESALGEBRE4]
Période 28	[RO3]		[DESLF3] [DESRO4]
Période 29	[RO5]		[RO4G1] [RO4G2]
Période 30			

Conclusion générale et perspectives

Ce mémoire décrit une approche de résolution du problème d'emploi du temps qui consiste à affecter les séances d'un ensemble de groupes d'enseignement à un ensemble de salles sur une période de référence donnée, tout en respectant un groupe de contraintes qui les relie.

Vu la complexité théorique et la taille importante qui, très souvent, caractérisent le problème, nous nous sommes orientés vers une adaptation de l'heuristique de la recherche tabou couplée au problème de flots dans les réseaux pour le résoudre.

L'essentiel de notre travail a porté, principalement, sur la définition d'une procédure performante permettant de générer un emploi du temps à partir d'un autre en essayant de réduire, le plus possible, le déroulement simultané des séances d'un même groupe d'enseignement.

La conception d'un logiciel interactif conçu avec l'outil « C++Builder 5 » qui implémente l'approche de résolution proposée, nous a permis d'obtenir de « bons » résultats sur des données de la Faculté des Sciences Mathématiques de l'USTHB où l'on a pu affecter 201 séances de 57 modules en 2 secondes avec des contraintes très restrictives sur la disponibilité des salles.

Cependant, on ne prétend pas donner ici une solution répondant parfaitement à toutes les contraintes réelles du problème, mais l'emploi du temps obtenu constitue une solution de base garantissant la satisfaction des contraintes majeures du problème et sur laquelle, quelques modifications manuelles très simples permettront d'engendrer un emploi du temps concrètement opérationnel.

Par ailleurs, il est clair que lors de l'exploration d'un domaine de solutions durant une période donnée, plus (respectivement moins) il y a « d'efforts » pour passer d'une solution à une autre, plus (respectivement moins) efficace sera chaque déplacement mais moins (respectivement plus) de solutions seront parcourues. La démarche de résolution proposée nous a permis d'apprécier les résultats positifs d'un bon compromis entre ces deux aspects.

Comme perspectives, nous proposons d'envisager l'adaptation de la procédure de résolution proposée à des problèmes d'emploi du temps ayant plus de contraintes ou formulés sous forme de problèmes multicritères, dans la mesure où nous avons donné un algorithme efficace pour le parcourt de l'ensemble des solutions qui peut être aussi exploité dans ces cas.

Par ailleurs, la notion de *poids spécifique* peut, éventuellement, être enrichie pour refléter une amélioration par rapport à davantage de contraintes, ce qui pourrait induire des déplacements encore plus intéressants dans l'ensemble des solutions.

Bibliographie

- [1] D.Abramson, Constructing school timetables using simulated annealing : Sequential and parallel algorithms, *Management Science* 37 (1991) 98-113
- [2] R.Alvarez, G.Martin, J.M.Tamarit, Constructing good solutions for the spanish scool timetabling problem, *Journal of Operational Research Society*. 47 (1996) 1203-1215
- [3] F.Amier, I.Belkadi, Résolution du problème d'emploi du temps par une méthode stochastique. Cas de l'USTHB (1999), Thèse d'Ingénieur en Recherche Opérationnelle (USTHB)
- [4] C.Berge, *Graphes*, Gauthier-Villars, Paris (1983)
- [5] J.P.Boufflet, *Emploi du temps dans un environnement fortement contraint, exemple de l'UTC* (1992), Thèse de Doctorat (UTC)
- [6] D.Brélaz, New methods to color the vertices of a graph, *Communications of the ACM* 22 (1979) 251-256
- [7] M.Cangalovic, V.Kovacevic-Vujcic, L.Ivanovic, M.Drazic, Modeling and solving a real-life assignment problem at universities, *European Journal of Operational Research* 110 (1998) 223-233
- [8] M.W.Carter, G.Laporte, J.W.Chinneck, A general examination scheduling system, *The Institute of Management Sciences, Interface* 24, (1994) 109-120
- [9] M.W.Carter, G.Laporte, S.Y.Lee, Examination timetabling : Algorithmic strategies and applications, *Journal of Operational Research Society*. 47 (1996) 373-383
- [10] N.Chahal, D.de Werra, An interactive system for constructing timetables on PC, *European Journal of Operational Research* 40 (1989) 32-37
- [11] M.E.A.Chergui, *Modèle structuré d'un algorithme d'affectation des enseignements aux locaux* (1991), Thèse de Magistère en Recherche Opérationnelle (USTHB)
- [12] A.Coloni, M.Dorigo, V.Maniezzo, Genetic algorithms : A new approach to timetable problem, *Research Report, Dip. Di Electronica Politecnico di Milano* (1990)
- [13] D.Costa, A tabu search for computing an operational timetable, *European Journal of Operational Research* 76 (1994) 98-110
- [14] D.de Werra, An introduction to timetabling, *European Journal of Operational Research* 19 (1985) 151-162
- [15] D.de Werra, Graphs, hypergraphs and timetabling, *Methods of Operations Research* 49 (1985) 201-213
- [16] D.de Werra, Some combinatorial models for course scheduling, *ORWP 95/08*, Swis Federal Institute of Technology, Lausanne (1995)
- [17] D.de Werra, N.V.R.Mahadev, U.N.Peled, Edge chromatic scheduling with simultaneity constraints, *SIAM Journal on Discrete Mathematics* 6 (1993) 631-641
- [18] V.Dhar, N.Ranganathan, Integer programming vs. Expert system : an experimental comparaison, *Communications of the ACM* 33/3 (1990) 323-336
- [19] R.W.Eglese, G.K.Rand, Conference seminar timetabling, *Journal of Operational Research Society* 38 (1987) 591-598
- [20] S.Even, A.Atai, A.Shamir, On the complexity of timetable and multicommodity flow problems, *SIAM Journal on Computing* 5 (1976) 691-703
- [21] J.A.Ferland, C.Fleurent, A decision support system for course scheduling, *The Institute of Management Sciences, Interface* 24, (1994) 105-115

- [22] J.A.Ferland, S.Roy, Timetabling for university as assignment of activities to ressources, *Computers and Operations Research* 12/2 (1985) 207-218
- [23] F.Glover, Tabu Search - Part I, *ORSA Journal on Computing* 1 (1989) 190-206
- [24] F.Glover, Tabu Search - Part II, *ORSA Journal on Computing* 2 (1990) 4-32
- [25] M.Gondran, M.Minoux, *Graphes et algorithmes*, Edition Eyrolles (2ème édition) 1985
- [26] A.Hertz, Finding feasible course schedule using Tabu Search, *Discrete Applied Mathematics* 35 (1992) 255-270
- [27] A.Hertz, Tabu Search for large scale timetabling problems, *European Journal of Operational Research* 54 (1991) 39-47
- [28] A.Hertz, V.Robert, Constructing a course schedule by solving a series of assignment type problems, *ORWP 94/10*, EPFL, November (1994)
- [29] A.Hertz, V.Robert, V.Berthod, Planification des emplois du temps de la formation au sein d'une grande entreprise, *RAIRO Operations Research* 34 (2000) 61-83
- [30] D.Johnson, Timetabling university examinations, *Journal of Operational Research Society*, Vol 41. No 1.(1990) 39-47.
- [31] L.Kang, G.M.White, A logic approach to the resolution of constraints in timetabling *European Journal of Operational Research* 61 (1992) 306-317
- [32] S.Kirkpatrick, C.D.Gelatt, M.P.Vecchi, Optimisation by simulated annealing, *Science* 220 (1983) 671-680
- [33] H.E.Mausser, M.J.Magazine, J.B.Moore, Application of an annealed neural network to a timetabling problem, *INFORMS Journal on Computing*. Vol 8, No 2 (Spring 1996) 103-117
- [34] M.Pirlot, General local search heuristics in combinatorial optimisation : a tutorial, *Belgian Journal of Operations Research* 32 (1992) 7-67
- [35] Sakarovitch.M, *Techniques Mathématiques de la Recherche Opérationnelle, Optimisation combinatoire*, Université Scientifique et Médicale, Institut National Polytechnique de Grenoble - ENSIMAG, Mai 1983
- [36] A.Tripathy, A lagrangian relaxation approach to course timetabling, *Journal of Operational Research Society* 31 (1980)
- [37] A.Tripathy, School timetabling - A case in large binary integer linear programming, *Management Science*, vol.30, No 12, December (1984)
- [38] G.M.White, S.K.S.Wong, Interactive timeabling in universities, *Computers and Education* 12/4 (1988) 521-529
- [39] J.M.Wilson, The scheduling of magistrates to courts, *Journal of Operational Research Society* 32 (1981) 121-124
- [40] M.Wright, School timetabling using heuristic search, *Journal of Operational Research* 47 (1996) 347-357

Annexe

1. Introduction

Nous abordons, dans ce qui suit, une présentation globale du programme « MTimeTabler » élaboré pour implémenter et mettre en œuvre l'approche de résolution proposée.

2. Présentation du programme MTimeTabler

Caractéristiques et fonctionnalités principales

Nous citons, ci-dessous, les principales fonctionnalités de « MTimeTabler » :

- Le programme offre une interface *SDI* (Single Document Interface). Il ne peut, par conséquent, traiter qu'un problème à la fois.
- Ce programme est muni d'une aide sous forme d'une page web détaillant toutes les fonctionnalités de « MTimeTabler ».

Données en entrer :

- Un assistant est prévu pour orienter l'utilisateur, étape par étape, lorsqu'il saisit toutes les données du problème.
- Toutes les données et paramètres de calculs sont sauvegardés dans un fichier qui possède une structure spécifique à « MTimeTabler ».
- L'assistant en question peut être appelé à n'importe quel moment pour changer certaines données ou paramètres afin de les sauvegarder (dans le même fichier ou dans un autre), les ajuster ou constater l'impact de ces modifications sur les résultats.
- Une allocation dynamique de la mémoire est prévue pour stocker les données du problème avant le début des calculs.

Calcul de l'emploi du temps :

- L'utilisateur est informé en permanence de l'état d'avancement des calculs.
- L'ensemble de tous les calculs sont faits par un processus lancé en arrière plan. Ce dernier permet de garder le contrôle sur l'application et surtout d'arrêter les calculs à n'importe quel moment.
- Si l'utilisateur met fin aux calculs, le meilleur emploi du temps rencontré sera affiché.

Résultats :

- L'emploi du temps résultant des calculs effectués sera affiché sur une grille dont les colonnes et les lignes correspondent, respectivement, aux types de salles et aux périodes et à l'intersection desquelles se trouvent les séances à affecter.
- L'emploi du temps affiché peut être exporté au format Microsoft Excel 97 ou 2000 pour être mis en forme, imprimé ou subir toute autre manipulation permise par ce puissant outil.

Environnement de MTimeTabler

Le logiciel offre une interface et un ensemble de commandes dont les caractéristiques sont détaillées et illustrées ci-dessous :

- 1 :** Ouvre la boîte de dialogue <Programme> (illustrée en 14) pour introduire, en cinq étapes, les données du programme (données du problème) : Groupes d'enseignement, groupes de séances, périodes, salles, compatibilité d'affectations, ...etc, pour lequel il faut construire un emploi du temps.
- 2 :** Pour charger, à partir d'un fichier, un programme déjà enregistré avec MTimeTabler. Le fichier en question possède un format spécifique et une extension (*.mtt).
- 3 :** Pour enregistrer les données du programme en cours dans un fichier (d'extension (*.mtt) et dont le format est spécifique à MTimeTabler), qui comprend : Groupes d'enseignement, groupes de séances, périodes, salles, compatibilité d'affectations, ...etc.
- 4 :** Pour enregistrer dans un nouveau fichier (au format MTimeTabler), les données relatives au programme en cours ouvert.
- 5 :** Ferme MTimeTabler après avoir demandé confirmation de sauvegarde du fichier ouvert en cours.
- 6 :** Calcule l'emploi du temps du programme en cours (s'il existe) et l'affiche sur la grille de la fenêtre principale ainsi que : le nombre de séances affectées, le nombre de conflits, le temps de calcul, ...etc.
- 7 :** Pour éditer les données du programme en cours à travers la boîte de dialogue <Programme>
- 8 :** Améliore la répartition des séances sur la période considérée et affiche le nouvel emploi du temps ainsi calculé sur la grille.

- 9** : Pour enregistrer au format Microsoft Excel 97/2000 l'emploi du temps affiché sur la grille de la fenêtre principale.
- 10** : Affiche sur un navigateur web une page d'aide.
- 11** : Affiche la boîte « A propos ».
- 12** : Grille réservée à l'affichage de l'emploi du temps calculé.
- 13** : Espace réservé à l'affichage de quelques paramètres de calculs et des propriétés de l'emploi du temps affiché.
- 14** : La boîte de dialogue <Programme> permettant d'introduire les données du problème ainsi que certains paramètres de calcul.

The screenshot shows the MTimeT calculator interface with the following elements and callouts:

- 1:** Points to the 'Fichier' menu.
- 2:** Points to the 'Nouvel...' option in the 'Fichier' menu.
- 3:** Points to the 'Enregistrer' option in the 'Fichier' menu.
- 4:** Points to the 'Enregistrer sous...' option in the 'Fichier' menu.
- 5:** Points to the 'Quitter' option in the 'Fichier' menu.
- 6:** Points to the 'Action' menu.
- 7:** Points to the 'Calculer un emploi du temps' option in the 'Action' menu.
- 8:** Points to the 'Ctrl+Alt+C' shortcut in the 'Action' menu.
- 9:** Points to the 'Editer le programme' option in the 'Action' menu.
- 10:** Points to the 'F1' key in the 'Aide' menu.
- 11:** Points to the 'A propos ...' option in the 'Aide' menu.
- 12:** Points to the 'Types de salles' window.
- 13:** Points to the summary statistics window.

Types de salles window:

Péripodes	1	2
1	[5,1][7,1][9,1][10,1][18,1]	[5,1][7,1][9,1][10,1][18,1]
2	[2,1][4,1]	[5,1][6,1][9,1][10,1][12,1][16,1][17,1][18,1]
3	[2,1][4,1]	[5,1][6,1][10,1][11,1][12,1][16,1][17,1][18,1]
4	[1,1][3,1]	[7,1][8,1][11,1][12,1][13,1][15,1][17,1][18,1]
5	[2,1]	[5,1][6,1][9,1][11,1][17,1][18,1]
6		[5,1][7,1][9,1][17,1]
7		[8,1][9,1][10,1][18,1]
8	[1,1][3,1]	[7,1][13,1][14,1][15,1][18,1]

Summary Statistics window:

- [Emploi du temps initial] : Nombre de conflits : 13
- [Emploi du temps amélioré] : Initialisation en cours...
- [Emploi du temps amélioré] : Amélioration en cours...
- [Emploi du temps amélioré] : Calculé...
- [Emploi du temps amélioré] : Emploi du temps final affiché...
- [Emploi du temps amélioré] : Nombre de conflits : 0
- [Emploi du temps amélioré] : Nombre (minimum) de séances affectées : 201 sur 201

Programme

Etape 1 sur 5

1) Nombre de groupes de séances et l'ensemble des groupes pouvant être en conflit

2) Nombre de séances pour chaque groupe de séances

3) Nombre de types de salles et de périodes

4) Nombre de salles disponibles de chaque type et à chaque période

5) Possibilité d'affectation des séances aux types de salles et aux périodes

Avancé....

Nombre de groupes de séances: 18

Groupes de séances qui ne peuvent se dérouler en même temps

1er Groupe: 1, 2, 3, 4, 5, 6, 7, 8

2ème Groupe: 4, 5, 6, 7, 8, 9, 10, 11

Premier groupe	Deuxième groupe
1	5
1	6
2	7
2	8
3	9
3	10

Ajouter >>

<< Supprimer

< Précédent

Suivant >

Terminer

Annuler

14