Problème 1 : Planification de rencontres sportives

(http://www.csplib.org/ => Scheduling => prob026: round robin tournaments)

Description du problème

Nous considérons les spécifications suivantes :

- un tournoi regroupe un ensemble T d'équipes (|T| pair) ;
- chaque équipe rencontre toutes les autres exactement une fois. La notion de matches aller/retour n'est pas traité dans ce problème ;
- le tournoi se déroule sur |T|-1 semaines. Chaque équipe joue une et une seule fois chaque semaine ;
- la programmation d'une rencontre se déroule sur un terrain, à une certaine date. Nous emploierons le terme générique de période, au nombre de |T|/2. A chaque période de chaque semaine correspond un match. Aucune équipe ne peut jouer plus de deux fois sur une période.

Le problème est de déterminer une programmation qui respecte l'ensemble de ces contraintes.
Le Tableau 1 montre un exemple de planification vérifiant l'ensemble des contraintes pour |T|=8 équipes numérotées de 0 à 7 (il y a donc 7 semaines et 4 périodes).

	Période 0	Période 1	Période 2	Période 3
Semaine 0	0 vs 1	2 vs 3	4 vs 5	6 vs 7
Semaine 1	0 vs 2	1 vs 7	3 vs 5	4 vs 6
Semaine 2	4 vs 7	0 vs 3	1 vs 6	2 vs 5
Semaine 3	3 vs 6	5 vs 7	0 vs 4	1 vs 2
Semaine 4	3 vs 7	1 vs 4	2 vs 6	0 vs 5
Semaine 5	1 vs 5	0 vs 6	2 vs 7	3 vs 4
Semaine 6	2 vs 4	5 vs 6	0 vs 7	1 vs 3

Tableau 1 - *Exemple de planification pour 8 équipes*.

Comme dans le tableau précédent, une configuration peut être représentée sous la forme d'un tableau à deux dimensions avec les semaines en lignes et les périodes en colonnes.

Chaque colonne vérifie la contrainte de cardinalité pour laquelle aucune équipe n'apparaît plus de deux fois. Sur chaque ligne chaque équipe apparaît exactement une fois. Ce qui est équivalent à dire que toutes les équipes d'une ligne sont différentes. Enfin, il existe une contrainte globale sur tout le tableau : chaque match n'y apparaît qu'une seule fois, *i.e.* tous les matches sont différents.

Problème 3 : Somme coloration

Soit $G = (V, E)$ un graphe non orienté, simple et sans boucle, avec V l'ensemble des sommets de G et $E \subset V \times V$ l'ensemble de ses arêtes. Une k-coloration de G est une partition de V en k sous-ensembles $V_1, \ldots, V_k$ tels que $\{v, v'\} \in V_i \times V_i \, (1 \leq i \leq k) \Rightarrow \{v, v'\} \notin E$. Le problème de somme coloration (SumCOL) d'un graphe G consiste à trouver une k-coloration c de G telle que $f(c) = \sum_{i=1}^{i=k} i \cdot |V_i|$ soit minimum, avec $|V_i|$ la cardinalité de V_i et $|V_1| \geq \cdots \geq |V_k|$.

Pour les graphes Dimacs, consulter le site : http://cedric.cnam.fr/~porumbed/graphs/

Problème 2 : Progressive party problem

Spécification et description du problème dans : http://www.info.univ-angers.fr/pub/hao/papers/MetaH2.pdf
Informations complémentaires sont sur : http://www.csplib.org/ => problems organised by subject area and problem number => prob013: progressive party problem).

Travail demandé

1. Choisir un de ces problèmes et implémenter une ou plusieurs algorithmes heuristiques.
2. Préparer un compte de rendu de 5 pages minimum incluant la description de méthode(s) développée(s), les résultats obtenus et incluant les meilleurs résultats de l'état de l'art et les références.
3. Présentation orale de 10-15 minutes à la dernière séance du cours.