Intelligence Artificielle : Logique et Contraintes - Examen - session n° 1

Date : 21/02/2023 Durée : 2h30

Documents autorisés : 1 copie double manuscrite (recto/verso soit 4 pages)

Barème : 2 + 6 + 6 + 6 (le barème est juste est indicatif)

Il est conseillé de lire le sujet dans son intégralité avant de commencer.

Les exercices sont indépendants. Vous pouvez les traiter dans l'ordre de votre choix.

Toute réponse doit être justifiée, sinon elle ne sera pas prise en compte.

Compte tenu des contraintes sanitaires, **ne pas cacheter vos copies** mais pensez à bien recopier votre numéro de copie sur chaque intercalaire.

Exercice 1 (Questions de cours)

- 1. Un algorithme comme AC-3 peut-il être utilisé pour produire toutes les solutions d'un CSP binaire ?
- 2. On considère différentes variantes du schéma général d'algorithme de recherche hybride, avec différentes procédures de filtrage. Classer les algorithmes suivants en fonction de leur pouvoir de filtrage (par ordre croissant) Full-Lookahead, BT, Partial-Lookahead, ForwardChecking, AC-L (i.e. Real Full Lookahead)

Exercice 2 (Formalisation de problème)

Quatre jeunes juges tout juste sortis de l'école nationale de la magistrature se retrouvent à l'occasion d'une rencontre amicale et échangent leurs points de vue sur leur première année d'expérience. Ils étaient tous quatre de très bons étudiants et occupaient les 4 premières places dans le classement de sortie de l'école, ce qui leur a plus ou moins permis de choisir le lieu de leur première affectation. On sait qu'au moment de leur candidature, ils étaient tous âgés entre 27 et 29 ans et que deux d'entre eux avaient 28 ans. On sait également que :

- a) André à 29 ans.
- b) Lucie était moins bien classée que le candidat affecté à Paris.
- c) Les deux premiers ont choisi une affectation dans des villes au sud de la France (Lyon ou Marseille).
- d) Didier, qui a 28 ans était moins bien classé que celui qui est allé à Lyon.
- e) Ce n'est pas Claire qui est allée à Lyon.
- f) Le plus jeune candidat était classé au moins 2 rangs derrière Claire et s'est retrouvé affecté à Lille.
- 1. Formaliser l'énoncé précédent sous forme d'un problème de satisfaction de contraintes. Bien préciser le sens de vos variables et comment vous modélisez les différentes parties de l'énoncé. Vous ferez attention à ne pas utiliser de contraintes disjonctives.
- 2. Trouver une solution pour ce problème. Vous pouvez procéder comme vous le souhaitez... mais expliquez rapidement votre manière d'obtenir cette solution.

Exercice 3 (Algorithmes Hybrides)

On considère le CSP $\mathcal{P} = (X, D, C)$ caractérisé par :

- $X = \{A, B, C, D, E, F\}$
- $D = \{d, d, d, d, d, d\}$, avec d = 1..5
- $C = \{ F \ge 2, E \ne D, D < C, C 2 \le B, A > B + 1, |A E| \le 3, F = B + 2, E \ne F, D \mod 2 = 1, F \ne D, E \le 3, D > B \}$
- 1. Dessinez le graphe de contraintes correspondant à \mathcal{P}

- 2. Construire l'arbre correspondant à l'exécution de l'algorithme de recherche avec backtrack (BT) pour trouver les solutions de ce problème mais uniquement dans le prolongement de la branche correspondant à l'affectation partielle $A=4,\ B=2.$ Vous affecterez les variables restantes dans l'ordre F,D,C,E.
- 3. On souhaite à présent comparer le résultat précédent avec une méthode hybride. Après avoir traité les contraintes unaires de façon adéquate, appliquez l'algorithme de recherche hybride du *forward checking* (FC) pour trouver **toutes** les solutions de ce problème. Vous affecterez les variables dans l'ordre D, B, F, C, A, E

Exercice 4 (Opl - Gappy)

Le jeu Gappy consiste à noircir certaines cases d'une grille de taille n*n, de façon à respecter certaines propriétés. Chaque ligne et chaque colonne doit contenir exactement deux cases noires. Deux cases noires ne peuvent être voisines, i.e. ne peuvent se toucher, latéralement ou diagonalement. Pour chaque instance une série d'indices est communiqué sous la forme d'entiers figurant à gauche de chaque ligne (resp. au sommet de chaque colonne). Le nombre indiqué caractérise le nombre de cases blanches séparant les deux cases noires sur cette ligne (resp. colonne).

Par exemple, une solution possible pour l'instance décrite figure 1a correspond à la figure 1b.

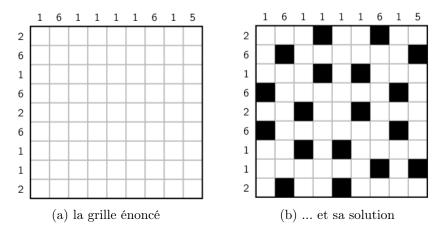


FIGURE 1 – Une instance du problème Gappy sur une grille 9×9 et sa solution

•

On souhaite écrire un modèle générique OPL permettant de résoudre n'importe instance pour ce jeu.

- 1. Proposer un moyen de décrire dans le langage OPL les données caractérisant une instance d'un problème Gappy (l'illustrer avec le code décrivant l'instance de la figure 1a).
- 2. Ecrire le code OPL permettant de modéliser ce problème (sauf les contraintes, traitées à la question suivante). Vous détaillerez la façon dont les données sont représentées (de façon externalisées), les pré-traitements éventuels, les variables de décision à déclarer et l'affichage de la solution.
- 3. Décrire le code modélisant les contraintes à respecter pour trouver une solution pour n'importe quel instance. Bien expliquer la façon dont chaque contrainte de l'énoncé est modélisée.