

# Programmation par contraintes

En M2 Bases de données et IA (UE 7.1) et M2 Image et IA

Par Olivier Bailleux, Maître de conférences HDR, Université de Bourgogne

Dans ce document, les noms communs désignant des catégories de personnes, tels qu'étudiant, participant, enseignant, apprenant, etc. ne désignent pas spécialement des personnes de sexe masculin, mais sont utilisés dans un sens neutre vis-à-vis du genre. Par exemple, étudiant signifie « étudiante ou étudiant », étudiants signifie « étudiantes et étudiants ».

## Qu'est-ce que la programmation par contraintes ?

C'est l'art de résoudre des problèmes en les modélisant à l'aide de variables et de relations (appelées contraintes) entre ces variables, de manière à permettre leur résolution à l'aide de solveurs génériques sans avoir à développer des algorithmes spécifiques. D'après Eugene Freuder, « Constraint Programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: The user states the problem, the computer solves it ». La programmation par contraintes s'inscrit dans le champ de l'intelligence artificielle symbolique et utilise des techniques de raisonnement automatique.

## Prérequis

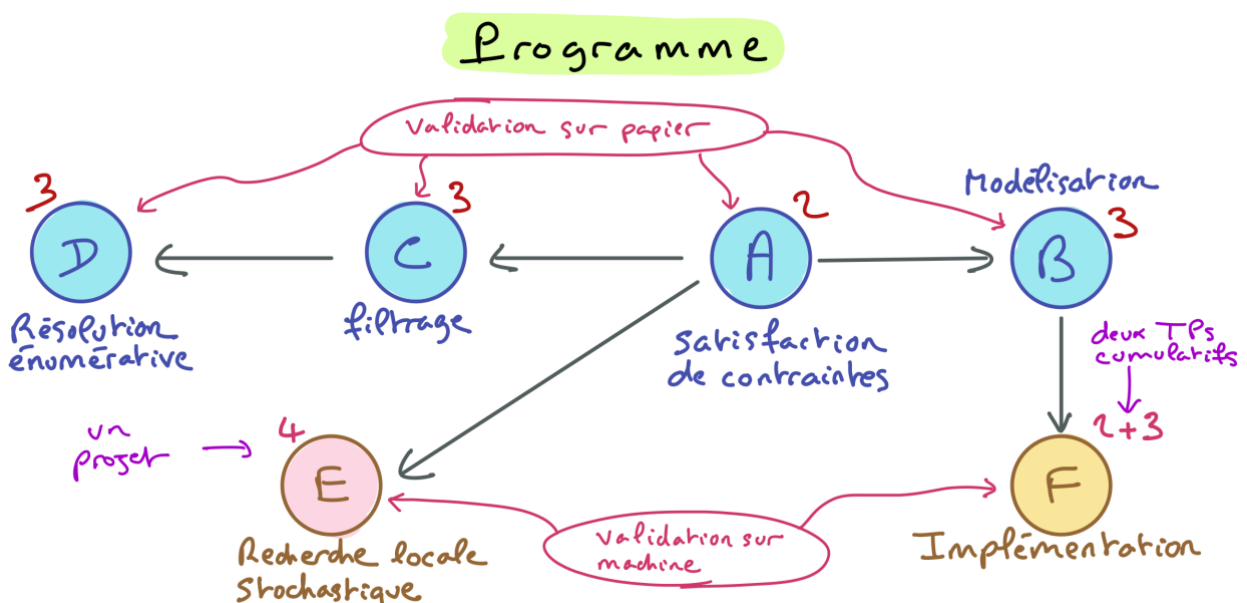
**Notions élémentaires de mathématiques** : savoir ce qu'est une relation, un ensemble, comprendre les notations utilisant les symboles somme et produits, les quantificateurs universels et existentiels, maîtriser les opérations Booléennes (et, ou, non).

**Notions de base en algorithmique** : avoir une compréhension limpide des comportements et utilisations des boucles, alternatives, variables, tableaux et opérations arithmétiques et logiques courantes.

Au moins une centaine d'heures d'**expérience de programmation** dans un langage impératif objet ou procédural, et quelques notions de programmation en langage JAVA (déclaration et instanciation de classes, créations de tableaux, définition et utilisation de méthodes de classe et de méthodes d'instances).

## Objectifs de l'enseignement

Acquérir les compétences désignées par les badges représentés dans la figure ci-dessous. Les flèches indiquent les prérequis de chaque badge.



Les badges A, B, C, D sont évalués deux fois et votre score pour chacun d'eux est le maximum des scores obtenus aux deux évaluations.

Dans le cas du badge F, il y a deux mini-projets facultatifs dont les notes s'ajoutent.

Le badge E est évalué par un projet facultatif.

## Détail des compétences pouvant être acquises

### A : satisfaction de contraintes

Savoir déterminer si une assignation de variables satisfait ou falsifie une contrainte, est ou non une solution d'une instance de problème (appelée réseaux de contraintes). Savoir déterminer si un réseau de contraintes simple, traitable sans ordinateur, est cohérent.

### B : modélisation

Être capable de modéliser une propriété simple, décrite en langage naturel, sous la forme d'une contrainte spécifiée en extension ou en compréhension. Être capable de traduire en extension une contrainte donnée en compréhension. Être capable de modéliser un problème sous la forme d'un réseau de contraintes définies en extension ou en compréhension.

### C : Filtrage

Savoir déterminer si une valeur d'un domaine a un support pour une contrainte, si un réseau de contraintes respecte la propriété de cohérence de domaine. Savoir restaurer la cohérence de domaine d'un réseau de contraintes. Être capable de concevoir un algorithme de restauration de cohérence de domaine pour tout type de contrainte ne nécessitant pas une analyse mathématique allant au-delà d'un raisonnement élémentaire. Être capable de comparer les puissances de déduction de la restauration de la cohérence de domaine sur différentes modélisations d'un type de contraintes ou de réseaux de contraintes.

### D : Résolution énumérative

Savoir simuler le déroulement de l'algorithme MAC (Maintaining Arc Consistency) sur un réseau de contraintes pour lequel l'arbre de recherche tient largement sur une page au format A4. Savoir déterminer le nombre de solutions d'un réseau de contraintes par un raisonnement basé sur le principe de la recherche énumérative.

### E : Recherche locale stochastique

Être capable de concevoir et d'implémenter un algorithme de recherche locale stochastique permettant de résoudre un problème spécifié à l'aide de contraintes en proposant un espace de recherche, une relation de voisinage, et un critère de transition pertinents.

### F : modélisation avancée

Être capable de modéliser un problème sous la forme d'un réseau de contraintes et d'implémenter ce réseau en utilisant la bibliothèque CHOCO.

## Approche pédagogique

L'enseignement dispensé sera autant que possible *différencié* et *asynchrone*, c'est-à-dire que chaque participant étudiera en autonomie et à son rythme, avec l'écoute et les conseils de l'enseignant, différentes compétences constituant l'objectif pédagogique de l'unité d'enseignement. En respectant une cohérence pédagogique validée par l'enseignant, chaque étudiant peut adapter son parcours – et la quantité de travail associée – à sa capacité de

travail et de progression. En particulier, le deuxième projet du badge F et le projet du badge E sont complètement facultatifs.

Les ressources pédagogiques permettant de préparer les différents badges sont accessibles en ligne. Une application collaborative permet aux étudiants de poser des questions auxquelles les autres étudiants et les enseignants peuvent répondre. Elle permet également à l'enseignant de poser des questions aux étudiants, de leur proposer des activités et de leur donner des conseils, et permet à tous de partager de nouvelles ressources pédagogiques. Chaque étudiant peut communiquer de manière privée avec l'enseignant, mais il est invité à poser toute question où à faire toute remarque pouvant être utile aux autres sur un canal de communication accessible à tous.

## Modalités d'évaluation

Plusieurs contrôles intermédiaires et un contrôle terminal sont organisés. Le contrôle terminal est une évaluation sommative couvrant l'ensemble du programme et produit une note CT sur 20. Chaque contrôle intermédiaire évalue une partie des badges. Certains badges sont évalués deux fois et rapportent un nombre de points calculé selon le barème présenté au début de ce document.

La note de l'unité d'enseignement est  $(2 \text{ CT} + \text{CI}) / 2$ .

## Temps de travail

Chaque étudiant peut doser son temps de travail en fonction de ses ambitions, de ses capacités et de sa motivation pour la matière et du temps qu'il choisit d'investir dans les autres unités d'enseignement. Par exemple, le parcours A, B, C, D, F plafonne la note de contrôle continu à 16/20 mais ne nécessite que quelques heures de travail personnel hors des séances programmées. Pour aller au-delà, il faut faire le projet de validation du badge E, ce qui ajoute entre 10 et 20 heures de travail.

## Bibliographie

### Livres

Handbook of Constraint Programming, Editors F. Rossi, P. Van Beek, T. Walsh, Elsevier, 2006.

Programmation par contraintes, Annick Fron, Addison-Wesley, 1994.

### Cours en ligne

Attention, ces cours peuvent aller nettement au-delà du programme de la présente unité d'enseignement.

[Introduction à la programmation par contraintes](#), Ruslan Sadykov, LIX, École Polytechnique.

[Introduction à la programmation par contraintes](#), Lionel Eyraud-Dubois, INRIA Bordeaux.