Chapitre 1

Introduction

Les colonies de fourmis dans la nature montrent une fascinante capacité à trouver des « chemins » entre des sources de nourriture et la fourmilière. A l'échelle d'une fourmi, le problème résolu est complexe : la fourmi n'a qu'une vision très locale de son environnement, et elle n'utilise pas de carte ni de GPS ! L'intelligence qui permet la résolution du problème est « collective » et est en fait le résultat émergeant d'un grand nombre d'interactions élémentaires.

Ce mécanisme de résolution collective de problèmes a inspiré une méta-heuristique, c'est-à-dire une méthode générique de résolution de problèmes, appelée « optimisation par colonies de fourmis ». Le premier algorithme a été initialement proposé par Dorigo en 1992 pour résoudre le problème du voyageur de commerce, qui consiste à trouver le plus court circuit passant par un certain nombre de villes. Depuis ces premiers travaux, l'optimisation par colonies de fourmis a été appliquée à un très grand nombre de problèmes d'optimisation combinatoires « difficiles ».

On entend par « difficile » le fait que la résolution de ces problèmes nécessite l'examen d'un très grand nombre (exponentiel) de combinaisons. Tout un chacun a déjà été confronté à ce phénomène d'explosion combinatoire qui transforme un problème apparemment très simple en un véritable casse-tête dès lors que l'on augmente la taille du problème à résoudre.

> *Considérons par exemple les jeux de la famille du pentamino, consistant à recouvrir entièrement une figure donnée à l'aide d'un ensemble de pièces de formes variées sans qu'elles ne se chevauchent ni ne débordent de la figure.*

Lorsque la figure à recouvrir est petite, ces problèmes sont très facilement résolus par un examen systématique des différentes combinaisons possibles. Cependant, une augmentation légère de la taille de la figure à recouvrir peut augmenter si fortement le nombre de combinaisons à explorer que la résolution devient impossible par une simple énumération et, pour les problèmes de plus grande taille encore, même l'ordinateur le plus puissant qui soit ne peut énumérer toutes les combinaisons en un temps raisonnable.

L'enjeu pour résoudre ces problèmes dépasse le cadre ludique des puzzles : un très grand nombre de problèmes industriels sont également confrontés à ce phénomène d'explosion combinatoire comme, par exemple, les problèmes consistant à établir des emplois du temps, à planifier une production en minimisant les pertes de temps, à découper des formes dans des matériaux en minimisant les chutes, etc. Il est donc crucial de concevoir des algorithmes permettant de résoudre effectivement ces problèmes combinatoires difficiles.

Ce livre étudie les capacités de l'optimisation par colonies de fourmis pour résoudre ces problèmes combinatoires difficiles. Cette étude est plus particulièrement menée dans le contexte de la programmation par contraintes qui permet de décrire les problèmes combinatoires de façon déclarative, en termes de contraintes.

Organisation du livre

Le livre est structuré en trois parties qui sont présentées dans les paragraphes suivants.

En préambule à ces trois parties, le chapitre 2 positionne les problèmes combinatoires dans le contexte de travaux théoriques sur la complexité des problèmes. Il s'agit là de bien mesurer le défi à relever : le caractère inévitable de l'explosion combinatoire impose la conception d'approches « intelligentes », capables de contenir ou de contourner cette explosion.

Programmation par contraintes

La première partie de cet ouvrage dresse un panorama des différentes approches existantes pour résoudre les problèmes fortement combinatoires dans le contexte de la programmation par contraintes.

Le chapitre 3 introduit tout d'abord la notion de « problème de satisfaction de contraintes », qui offre un cadre structurant pour la modélisation de problèmes combinatoires en termes de contraintes.

On présente ensuite les deux principales familles d'approches pouvant être utilisées pour résoudre ces problèmes de satisfaction de contraintes.

Les approches « exactes », présentées au chapitre 4, explorent de façon systématique l'espace des combinaisons jusqu'à trouver une solution ou prouver l'absence de solution. Afin de (tenter de) contenir l'explosion combinatoire, ces approches structurent l'espace des combinaisons en arbre et utilisent des techniques d'élagage, pour réduire cet espace, et des heuristiques, pour déterminer l'ordre dans lequel il est exploré.

Les approches « heuristiques », présentées au chapitre 5, contournent le problème de l'explosion combinatoire en faisant délibérément des impasses et n'explorent qu'une partie des combinaisons. Il existe essentiellement deux grandes familles d'approches heuristiques :

– les approches par voisinage construisent des combinaisons en modifiant des combinaisons existantes (par croisement et mutation pour les algorithmes génétiques, par application de transformations élémentaires pour la recherche locale, par déplacement en fonction d'une vitesse pour les algorithmes par essaims de particules) ;

– les approches constructives génèrent des combinaisons de façon incrémentale en utilisant pour cela un modèle stochastique. Dans le cas des algorithmes gloutons aléatoires, ce modèle est statique ; dans le cas des algorithmes par estimation de distribution et des algorithmes par colonies de fourmis, il évolue en fonction des expériences passées.

Le chapitre 6 conclut cette première partie en présentant quelques langages de programmation par contraintes. Ces langages permettent de décrire les problèmes combinatoires de façon déclarative, en termes de contraintes, et intègrent des algorithmes de résolution tels que ceux décrits aux chapitres 4 et 5.

Optimisation par colonies de fourmis

La deuxième partie de cet ouvrage présente l'optimisation par colonies de fourmis. Il s'agit là d'une approche heuristique qui, comme les autres approches décrites au chapitre 5, n'explore délibérément qu'une partie de l'espace des combinaisons et utilise des (méta-)heuristiques pour se guider dans cet espace vers les zones les plus prometteuses.

L'optimisation par colonies de fourmis s'inspire du comportement collectif des colonies de fourmis pour trouver un plus court chemin entre deux points. Aussi commençons-nous cette deuxième partie, au chapitre 7, par une description des mécanismes permettant aux fourmis dans la nature de converger sur un plus court chemin.

Le chapitre 8 décrit ensuite la méta-heuristique d'optimisation par colonies de fourmis qui constitue un cadre générique pour la résolution de problèmes d'optimisation combinatoires. Au-delà de la métaphore des colonies de fourmis, ce chapitre explicite les mécanismes mis en œuvre pour permettre à l'algorithme de converger vers la solution, et notamment de trouver un bon équilibre entre la diversification et l'intensification :

– la diversification vise à limiter les impasses et garantir un bon échantillonnage de l'espace des combinaisons ; elle est assurée par le mode de construction des combinaisons, basé sur un modèle stochastique ;

– l'intensification vise à guider l'effort de recherche vers les meilleures combinaisons ; elle est assurée par un mécanisme de renforcement qui exploite les expériences passées pour biaiser progressivement le modèle stochastique.

Les algorithmes à base de colonies de fourmis se sont avérés particulièrement performants pour la résolution de nombreux problèmes d'optimisation combinatoires. On s'intéresse plus particulièrement dans cet ouvrage aux problèmes de satisfaction de contraintes, qui constituent une classe générique de problèmes combinatoires.

Nous illustrons tout d'abord au chapitre 9 les capacités des colonies de fourmis sur un problème industriel difficile, très souvent utilisé pour évaluer les langages de programmation par contraintes, à savoir un problème d'ordonnancement de voitures. Ce problème consiste à séquencer des voitures le long d'une chaîne de montage en respectant des contraintes d'espacement entre les voitures en fonction des options devant y être installées. Il s'agit d'un problème crucial pour l'industrie automobile. Il permet d'optimiser les charges de travail des ouvriers installant les options, dans un contexte où chaque acheteur peut choisir, lors de la commande, les options devant être installées sur sa voiture, et où le délai entre la commande et la livraison de la voiture doit être réduit au maximum. On montre que l'optimisation par colonies de fourmis permet d'obtenir des résultats compétitifs avec les meilleures approches connues pour ce problème. Il est également intéressant de noter la simplicité de mise en œuvre de cette approche sur ce problème.

Nous étudions ensuite au chapitre 10 les capacités des colonies de fourmis pour la résolution de problèmes de satisfaction de contraintes quelconques, pour lesquels on ne dispose d'aucune connaissance particulière sur la nature des contraintes. On montre là encore que l'optimisation par colonies de fourmis permet de résoudre très efficacement des instances difficiles.

Programmation par contraintes avec des colonies de fourmis

La dernière partie de cet ouvrage concerne la mise en œuvre du paradigme de programmation par contraintes avec des colonies de fourmis.

On décrit tout d'abord au chapitre 11 un projet de recherche, mené en partenariat avec la société ILOG, visant à intégrer des algorithmes à base de colonies de fourmis dans une bibliothèque de programmation par contraintes. Cette intégration permet de bénéficier des procédures pour la modélisation, la vérification et la propagation de contraintes intégrées dans la bibliothèque. En revanche l'exploration par construction d'un arbre de recherche traditionnellement utilisée en programmation par contraintes est remplacée par une exploration stochastique guidée par les expériences passées, selon le principe de la méta-heuristique d'optimisation par colonies de fourmis.

On conclut au chapitre 12 sur quelques pistes à approfondir pour une meilleure intégration de l'optimisation par colonies de fourmis dans un langage de programmation par contraintes.