Jeux combinatoires et pensée informatique

Emmanuel Beffara

décembre 2020

Les situations de recherche favorisent une approche expérimentale de l'activité mathématique. Elles donnent du sens au raisonnement en le mettant en action sur des problèmes où l'enjeu n'est pas une connaissance particulière. Les problèmes combinatoires, comme ceux évoqués ici, se prêtent bien à cette approche. D'une part les questions posées sont souvent compréhensibles sans formalisme et on peut les explorer par manipulation, ce qui favorise la réflexion en évitant les contraintes du langage mathématique. D'autre part, la recherche exhaustive permet de traiter de petits cas particuliers mais n'est pas praticable en général sans développer des approches systématiques et abstraites. Nous allons voir que ces problèmes stimulent aussi des compétences de nature informatique, même quand on ne cherche pas à utiliser un ordinateur pour les résoudre.

Énumérer les cas

Prenons le classique *jeu des allumettes*: on part d'un tas de 20 allumettes puis chacun des deux joueurs, à tour de rôle, doit prendre une à trois allumettes. Celui qui prend la dernière a perdu. Dans ce jeu combinatoire (cf. encadré), on se demande lequel des deux joueurs a une stratégie gagnante.

> **Encart:** Un *jeu combinatoire* est un jeu à deux joueurs, à information complète et sans hasard. Les joueurs y jouent à tour de rôle, dans le but d'atteindre une position victorieuse. Parmi ces jeux, on trouve les échecs, le go, l'awalé, le puissance 4… L'étude mathématique de ces jeux montre que dans n'importe quelle position d'un tel jeu, l'un des deux joueurs peut théoriquement être certain de gagner s'il choisit bien ses coups. Bien sûr, en pratique, il est rarement possible de déterminer effectivement quel est ce joueur et de donner une *stratégie gagnante*, mais c'est faisable dans des jeux relativement simples comme ceux dont parle cet article.

Les positions du jeu correspondent aux nombres d'allumettes restantes à un tour donné. Du fait de la structure du jeu, il y a une méthode simple pour déterminer si une position est gagnante ou perdante:

- Énumérer les coups possibles.
- Pour chaque coup, déterminer le statut de la position à laquelle il mène. Si elle est perdante, la position de départ est gagnante.

- Si tous les coups ont mené à des positions gagnantes, la position de départ est perdante.

Quelqu'un qui étudierait le jeu des allumettes sans en connaître la solution appliquerait naturellement une telle méthode qui consiste essentiellement à énumérer toutes les parties possibles. Ceci ne peut se faire qu'en procédant de façon méthodique, c'est-à-dire selon un algorithme (même si ce n'est pas formulé en ces termes). Cet algorithme fait intervenir l'itération, la conditionnelle et la récursivité:

Pour résoudre n:
  Pour chaque a de 1 à 3 (tel que $n - a > 1$),
    résoudre $n - a$,
    si c'est perdant, répondre que n est gagnante.
  Répondre que n est perdante.

Éviter de se répéter

Appliquer cette méthode au jeu des allumettes conduit à se poser plusieurs fois la question du statut de la plupart des positions: pour résoudre la position 20, il faut résoudre les positions 17 à 19. Quand on aura traité la position 17, on passera à la position 18, dont l'étude demande de déterminer le satut des positions 15 à 17, or celles-ci auront déjà été vues. On ne va donc pas les explorer à nouveau.

Pour éviter de répéter des calculs, on en viendra à noter ses résultats partiels dans un tableau pour pouvoir s'y reporter. On verra que pour résoudre une position, il faut avoir résolu toutes les positions plus petites. La façon la plus efficace d'organiser son calcul est alors de traiter les positions en ordre croissant:

Pour chaque n de 2 à 20:
  Pour chaque a de 1 à 3,
    si $n - a > 1$ et $n - a$ est marquée perdante,
      marquer n gagnante et passer au n suivant.
  Marquer n perdante.

C'est le même algorithme qu'au dessus, mais au lieu de le formuler par récursivité, on dit explicitement dans quel ordre procéder et comment réutiliser des résultats intermédiaires.

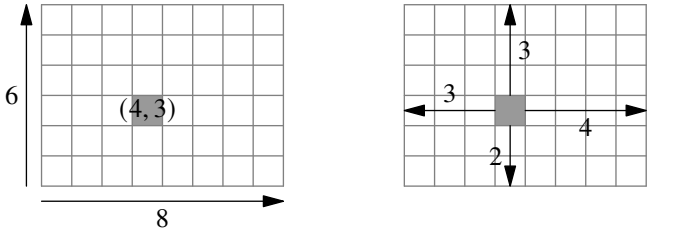
En termes informatiques, cela s'appelle la *programmation dynamique*. Cette technique s'applique à de nombreux problèmes d'optimisation, comme le rendu de monnaie ou la recherche de plus court chemin. Son étude est au programme de NSI en Terminale mais l'idée est accessible plus tôt sans bagage algorithmique. De fait, c'est une manifestation d'une forme de pensée informatique dans l'étude d'une question qui n'est pas de nature informatique.

Modéliser pour traiter l'information

Considérons à présent le *jeu du chocolat*. Il se joue à deux avec une tablette de chocolat rectangulaire dont un carré, connu des deux joueurs, est empoisonné. Chaque joueur à tour de rôle doit casser la tablette en deux et manger la partie qui ne contient pas le carré empoisonné. Le joueur qui ne peut plus jouer a perdu. On se demande, en fonction de la taille de la tablette et de la position du carré empoisonné, quel joueur a une stratégie pour gagner.

C'est aussi un jeu combinatoire et on peut explorer ses stratégies comme pour le jeu des allumettes. Mais pour mener l'exploration, un travail de modélisation est nécessaire: il faut représenter les positions du jeu sous une forme qui permet de s'organiser et d'identifier les positions déjà vues. Il faut donc *abstraire* les informations pertinentes du contexte du problème et les *structurer* pour les rendre manipulables, deux aspects fondamentaux de la pensée informatique.

Dans ce jeu, une position est caractérisée par la taille de la tablette restante et la position du carré empoisonné, c'est-à-dire un quadruplet (L, H, x, y). On voit facilement que des positions qui ne diffèrent que d'une rotation ou d'une symétrie axiale sont équivalentes.



$(L, H, x, y) = (8, 6, 4, 3)$ $\quad$ $(n_1, n_2, n_3, n_4) = (4, 3, 3, 2)$

deux représentations d'une même situation

L'étude du problème suggère pourtant une meilleure modélisation: représenter une position comme le nombre de carrés dans les quatre directions autour du carré empoisonné (cf. figure). L'intérêt est que jouer un coup consiste simplement à diminuer le nombre de carrés dans une direction sans modifier les autres. L'ordre des quatre nombres n'a donc pas d'importance pour le jeu et on peut identifier des positions qui ne diffèrent que par l'ordre, ce qui réduit plus l'espace de recherche que la première modélisation.

Ce travail de modélisation et de représentation est fondamental en informatique. En programmation, le choix des *structures de données* est crucial car il guide l'élaboration des algorithmes et ceux-ci seront plus faciles à étudier si les variables sont bien choisies.

Notons que pour une situation donnée, il y a rarement une seule modélisation adaptée. Pour le jeu du chocolat, la deuxième formulation est la plus efficace pour étudier les stratégies, mais pour dessiner une position, c'est la première qui est adaptée. En algorithmique classique, on retrouve ce phénomène avec les graphes: selon le cas, la représentation adaptée peut être la liste d'adjacence, la matrice d'adjacence ou la matrice d'incidence. Naviguer entre plusieurs représentations d'une information est une activité

omniprésente qui nécessite une bonne compréhension des objets et de leur sens: ceux qui ont déjà été confrontés à des problèmes de conversion de fichiers d'un logiciel vers un autre connaissent bien les effets de ce problème!

La programmation comme outil d'étude

L'approche informatique à l'œuvre dans les problèmes combinatoires peut être poursuivie jusqu'à programmer. Ce n'est pas utile dans le jeu des allumettes où la recherche exhaustive est accessible à la main, mais pour une situation plus élaborée comme le jeu du chocolat, cela peut être un outil d'étude pertinent: un programme permettra de traiter des cas pour lesquels l'étude manuelle n'est plus praticable.

Certes, traiter des cas particuliers par le calcul n'est pas suffisant pour résoudre un problème, mais il faut plutôt voir la programmation comme une instrumentation de la recherche mathématique. De même qu'on utilise des calculatrices quand on sait déjà compter, on utilise l'ordinateur pour réaliser une tâche devenue routinière qui s'intègre dans une démarche plus large.

Autrement dit, bien intégrée dans la démarche de recherche, l'informatique ne pense pas à notre place mais nous aide à penser!