qu'elles arrêtent leurs déplacements aléatoires et qu'elles rejoignent le chemin marqué, en renforçant le marquage à leur retour, s'il mène bien vers de la nourriture.

Dans le même temps, le chemin le plus court sera davantage parcouru, et donc plus renforcé et plus attractif, on parle de rétroaction positive (*positive feedback*). En considérant que la phéromone s'évapore, les chemins les moins renforcés finissent par disparaître, ce qui amène toutes les fourmis à suivre ce chemin le plus court.

L'algorithme de colonies de fourmis a été à l'origine principalement utilisé pour produire des solutions quasi-optimales au problème du voyageur de commerce, puis, plus généralement, aux problèmes d'optimisation combinatoire. On observe, depuis ses débuts, que son emploi se généralise à plusieurs domaines, depuis l'optimisation continue jusqu'à la classification, ou encore le traitement d'image [74], [125], [129], [109], [103], [69], [92] et [75].

Les variantes combinatoires apportent un avantage, par rapport aux autres métaheuristiques, dans le cas où le graphe étudié peut changer dynamiquement au cours de l'exécution : la colonie de fourmis s'adaptera en temps réel aux changements. Ceci est par exemple intéressant pour le routage d'un réseau.

Le problème du voyageur de commerce TSP Les fourmis partent initialement de la source et parcourent les différents chemins possibles (figure 5.1) jusqu'à la fin. Les quantités de phéromone sont initialement égales. Sur leur chemin de retour, les fourmis déposent de la phéromone sur les différents chemins. Les valeurs des quantités de phéromone sont renouvelées selon l'équation :

$$\tau_i = \tau_i + \frac{Q}{l_i}$$

avec Q une constante et l_i la longueur du chemin parcouru. Donc la nouvelle valeur de la quantité de phéromone est inversement proportionnelle à la longueur du chemin.
Un aller-retour d'une seule fourmi constitue une construction complète de la solution. Le processus est réitéré, et les fourmis repartent à nouveau de la source. Une fourmi donnée va choisir entre les chemins selon la probabilité :

$$P_{e_i} = \frac{\tau_i}{\sum_{i=1}^{n} \tau_i}$$

Dans notre exemple (figure 1.3) les probabilités seront les suivantes :

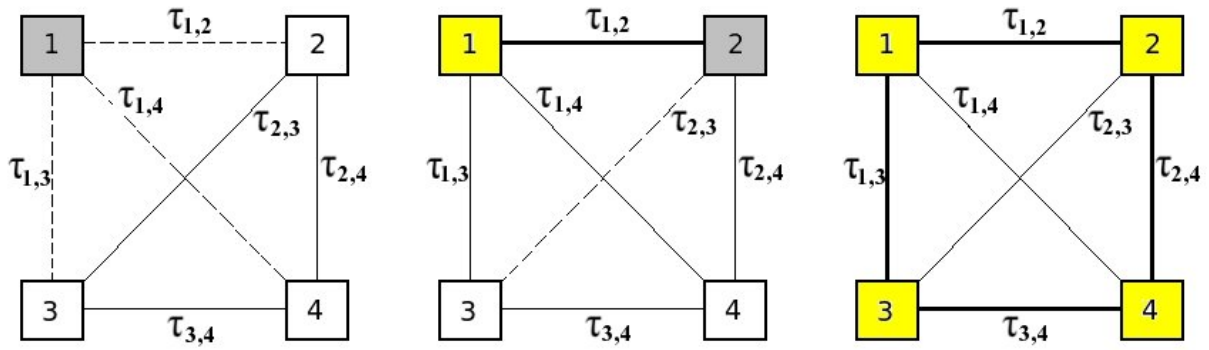
$$P_{e_1} = \frac{\tau_1}{\tau_1 + \tau_2 + \tau_3} \qquad P_{e_2} = \frac{\tau_2}{\tau_1 + \tau_2 + \tau_3} \qquad P_{e_3} = \frac{\tau_3}{\tau_1 + \tau_2 + \tau_3}$$

Donc, si nous supposons que $l_1 < l_2 < l_3$, alors $\tau_1 > \tau_2 > \tau_3$. Par conséquent, $P_{e_1} > P_{e_2} > P_{e_3}$. Le chemin choisi sera alors e_1.

D'autres changements affectent la phéromone, l'évaporation. Ce dernier changement a pour objectif d'empêcher une augmentation continue d'une seule solution, et de réduire l'importance d'autres solutions :

$$\tau_i = \tau_i.(1 - \rho) \qquad \text{avec} \quad \rho \in [0, 1]$$

ρ est le paramètre qui va régler la quantité de phéromone déposée, donc c'est le paramètre qui va influencer la vitesse de convergence de l'algorithme vers une solution donnée.



(a) Le début. (b) Le choix d'un chemin. (c) Un chemin complet.

FIG. 1.3 – Un exemple représentant la construction d'une solution dans le cas d'un problème discret, pour un problème de TSP avec quatres villes, N=4.

La méthode de colonies de fourmis a été initialement dédiée aux problèmes discrets (i.e. le cas où les valeurs possibles des variables d'un problème sont en nombre fini). Dans ce cas, nous pouvons définir un ensemble de composantes de la solution ; la solution optimale d'un problème donné sera alors une série ordonnée de ces composantes. Et un problème donné possède un ensemble de solutions. Pour fixer les idées, nous prenons l'exemple du problème du voyageur de commerce TSP. Le TSP peut être représenté par un graphe complètement connecté (un graphe où tous les noeuds sont inter-connectés). Les villes sont représentées par N_i noeuds avec $i = \{1, ..., n\}$ et n désigne le nombre total des noeuds. Un trajet entre deux noeuds (exemple : N_i et N_j) est désigné par e_{ij}. Par conséquent, une solution à un problème est un ensemble ordonné de $n - 1$ composantes $\{e_{ij}\}$ avec $i, j = \{1, ..., n\}$. Une construction d'une solution dans le cas du TSP consiste à partir d'une solution vide, à choisir une ville de départ et à ajouter les villes l'une après l'autre, puis à réitérer le processus jusqu'à ce que le nombre total de composantes ajoutées soit égal au nombre total de villes n. Dans le cas discret, les solutions ne sont pas

connues à l'avance, donc la valeur de la quantité de phéromone ne peut plus être attribuée à une solution complète, mais plutôt aux composantes de la solution.

Essaims particulaires

L'optimisation par essaims particulaires (OEP) est une métaheuristique d'optimisation, inventée par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995. Cet algorithme s'inspire à l'origine du monde du vivant [16]. Il s'appuie notamment sur un modèle développé par le biologiste Craig Reynolds à la fin des années 1980, permettant de simuler le déplacement d'un groupe d'oiseaux. Une autre source d'inspiration, revendiquée par les auteurs, est la socio-psychologie.

Cette méthode d'optimisation se base sur la collaboration des individus entre eux. Elle a d'ailleurs des similarités avec les algorithmes de colonies de fourmis, qui s'appuient eux aussi sur le concept d'auto-organisation. Cette idée veut qu'un groupe d'individus individuellement peu intelligents puisse posséder une organisation globale complexe.

Ainsi, grâce à des règles de déplacement très simples (dans l'espace des solutions), les particules peuvent converger progressivement vers un minimum local. Cette métaheuristique semble cependant mieux fonctionner pour des espaces en variables continues.

Au départ de l'algorithme, chaque particule est donc positionnée, aléatoirement ou non, dans l'espace de recherche du problème. Chaque itération fait bouger chaque particule, en fonction de trois composantes :

1. sa vitesse actuelle V_k,
2. sa meilleure solution P_i,
3. la meilleure solution obtenue dans son "voisinage" P_g.

Cela donne l'équation de mouvement suivante :

$$V_{k+1} = a.V_k + b_1(P_i - X_k) + b_2(P_g - X_k)$$

$$X_{k+1} = X_k + V_{k+1}$$

avec :

b_1 tiré aléatoirement dans $[0, \psi 1]$

b_2 tiré aléatoirement dans $[0, \psi 2]$