

Propagation d'incertitudes dans un modèle de collecte de déchets post-inondations

Sofiane MARTEL

M2 *Analyse et probabilités*

Responsable : J. Féjoz

sous la direction de
Mickaël BRASEBIN
Arnaud LE GUILCHER

Table des matières

1	Contexte et enjeux	1
2	Modélisation	3
2.1	<i>The Capacitated Arc Routing Problem</i>	3
2.2	Incertitudes	6
3	Colonies de fourmis	8
3.1	Principe général	8
3.2	Le problème du voyageur de commerce	10
3.3	CARP	11
4	Méthode intrusive	13
4.1	Principe	13
4.2	Les quantités de déchets : distribution transversale des phéromones	14
4.3	Hauteur d'eau : stratégie de réparation	15
5	Algorithme et résultats	16
5.1	Environnement déterministe	16
5.2	Environnement aléatoire / Méthode intrusive	18
5.3	Conclusion et perspectives	20

Introduction

Ce mémoire a été réalisé lors d'un stage de fin de M2 au sein du laboratoire COGIT de l'Institut national de l'information géographique et forestière (IGN). Ce stage s'est réalisé dans le cadre du projet SIGOPT (Systèmes d'Information Géographique et OPTimisation Territoriale) qui est un projet de type PEPS¹ financé par le CNRS et par la ComUE UPE. L'objectif des PEPS est de favoriser la collaboration entre établissements et la mise en œuvre de projets de recherche interdisciplinaires et exploratoires. Dans cette optique, le projet SIGOPT a rassemblé des chercheurs spécialisés en géomatique, en géographie, et en mathématiques appliquées des laboratoires COGIT et Lab'Urba². Sa thématique se situe au confluent de la gestion territoriale, de la géomatique et de l'optimisation. Je profite de cette introduction pour remercier mes encadrants, Mickaël Brasebin et Arnaud Le Guilcher, pour leur accueil, leurs précieux conseils et leurs idées innovantes.

1 Contexte et enjeux

Le projet SIGOPT s'intéresse au ramassage de déchets post-crue et à la mise en place de programmes de collecte de ces déchets à travers des systèmes d'information géographique (SIG). Des récents travaux ont mis en évidence l'importance d'un système efficace de gestion de déchets dans la *résilience* d'un territoire sinistré, c'est-à-dire dans sa capacité à revenir vers un fonctionnement

1. Projet Exploratoire Premier Soutien

2. Université Paris Est Créteil

acceptable à la suite d'une catastrophe naturelle [5]. Les déchets générés par une catastrophe comprennent la végétation, les gravats de construction ou de démolition, les ordures ménagères, les appareils électroménagers, les véhicules motorisés, les animaux morts ainsi que les produits chimiques dangereux. La quantité des déchets post-catastrophe peut parfois correspondre à plusieurs années de déchets ménagers habituellement produits. À titre d'exemple, les déchets générés par l'ouragan Katrina en 2005 sont estimés à plus de 90 millions de mètres cube [33]. Immédiatement après la catastrophe (durant la phase de «réponse»), le principal objectif du système de gestion de déchets est de débloquer les routes en dégagant les débris vers les côtés afin que les opérations de secours puissent se dérouler correctement. Une fois les activités d'urgence terminées, durant la phase de «relèvement», les déchets sont collectés et transportés depuis le territoire sinistré vers des sites de traitement.

La mise en place d'une tournée de collecte efficace nécessite de connaître les quantités de déchets présentes sur les tronçons de route. Cependant, ces quantités sont souvent très difficiles à estimer. Très souvent, les seules données a posteriori dont on dispose concernant les catastrophes passées sont des données agrégées et aucune information n'est disponible sur les routes individuelles. Le projet MECaDePI [6] propose des méthodes de caractérisation et de quantification a priori des déchets ménagers post-inondations à partir de données INSEE (ameublement, budget des familles, logements inondables...). De telles méthodes permettent d'aider à planifier la gestion des déchets en cas d'inondation mais leur précision est par définition limitée et ne permet pas de lever les incertitudes inhérentes aux aléas naturels. Étant donnée la difficulté à leur attribuer une estimation fiable, nous proposons un modèle dans lequel les quantités de déchets sont aléatoires. De cette manière là, l'incertitude est directement prise en compte dans le modèle et les résultats peuvent être traités de manière adaptée à notre connaissance imparfaite des données.

Une partie importante de la littérature en recherche opérationnelle se consacre au développement de modèles dits de *logistique d'urgence* et à des méthodes de résolution efficaces des problèmes de gestion soulevés par les catastrophes naturelles. Ces problèmes présentent souvent des difficultés inconfortables. Les problèmes de tournées optimales relèvent de l'optimisation combinatoire et les méthodes de résolutions exactes sont presque toujours inefficaces en pratique. En effet, ces problèmes appartiennent à la classe des problèmes NP-complets (les problèmes les plus "difficiles" de la classe NP [20]) pour lesquels aucun algorithme de résolution exacte en temps de calcul polynômial, ni une réfutation de l'existence d'un tel algorithme, n'ont été trouvés à ce jour. Pour cette raison, on assiste à un développement massif des méthodes d'approximation de solutions qui permettent d'atteindre une solution acceptable en un temps de calcul raisonnable. Ces méthodes sont généralement qualifiées d'*heuristiques*. Certaines de ces méthodes sont suffisamment générales pour être adaptées à des problèmes de natures différentes, on parle dans ce cas-là de *métaheuristiques* [14].

Dans la section qui suit, on présente le modèle que nous avons mis en place pour la collecte optimale des déchets sous incertitudes. Il s'agit d'une extension stochastique d'un problème d'optimisation combinatoire déjà présent dans la littérature scientifique et appelé CARP. En section 3, on présente la métaheuristique sur laquelle on se base pour résoudre notre problème : les colonies de fourmis. On montre d'abord comment cette méthode s'applique à des problèmes déterministes de parcours sur graphes comme le problème du voyageur de commerce ou le CARP. En section 4, on étend le concept des colonies de fourmis pour l'adapter à notre représentation aléatoire des données du problème : la méthode mise en place est qualifiée de «méthode intrusive». Les résultats obtenus sont présentés en section 5.

2 Modélisation

Les problèmes d'optimisation combinatoire ont souvent la particularité d'être très faciles à énoncer mais très difficiles à résoudre. Dans la plupart des cas, le nombre de solutions possibles augmente exponentiellement avec la taille des données du problème. L'idée d'une recherche exhaustive de toutes les solutions est alors d'emblée écartée. En pratique, on recherche un algorithme dont le temps de calcul d'une solution optimale est polynômial, c'est-à-dire qu'on peut trouver un polynôme p tel que pour des données d'entrée de taille n , le calcul de la solution optimale nécessite un temps de l'ordre de $\mathcal{O}(p(n))$. Une importante théorie permettant de caractériser la difficulté des problèmes d'optimisation combinatoire est celle de la NP -complétude. Cette théorie distingue deux classes de problèmes d'un intérêt particulier : la classe P des problèmes pour lesquels on connaît un algorithme de résolution polynômial, et la classe NP des problèmes pour lesquels on sait déterminer en temps polynômial la valeur d'une solution donnée. S'il est facile de voir que $P \subseteq NP$, la question de savoir si $P = NP$ est encore ouverte aujourd'hui (l'hypothèse $P \neq NP$ est néanmoins considérée comme la plus vraisemblable par la communauté scientifique). Un problème est dit NP -difficile si tous les problèmes de la classe NP peuvent se transformer en ce premier via un algorithme polynômial, ce qui signifie qu'il est au moins aussi difficile que chacun des problèmes de la classe NP . Un problème NP -difficile appartenant à la classe NP est dit NP -complet. Les problèmes NP -complets sont donc les plus difficiles de la classe NP et leur difficulté est à l'origine du développement des algorithmes d'approximation qui ne garantissent pas la découverte de l'optimum mais dont l'efficacité à trouver des solutions suffisamment proches de l'optimum est vérifiée expérimentalement.

Le problème que nous présentons ci-dessous appartient à la famille des problèmes NP -difficiles.

2.1 *The Capacitated Arc Routing Problem*

Dans le but de favoriser l'efficacité du ramassage de déchets, on cherche à trouver les meilleures tournées des véhicules de collecte pour débayer le territoire de sorte que la distance totale parcourue soit la plus petite possible. La théorie des graphes fournit le cadre naturel pour résoudre ce genre de problèmes. En effet, il s'agit d'un problème typique de "parcours sur arcs" (*arc routing*) appelé CARP (*Capacitated Arc Routing Problem*). Fondée à partir du fameux *problème des sept ponts de Königsberg*, résolu par Leonhard Euler en 1736 [17], la théorie de l'«arc routing» est aujourd'hui utilisée massivement et dans une variété de situations dépassant largement le cadre de la gestion des déchets.

Le CARP a été introduit pour la première fois en 1981 par Golden & Wong [22]. On peut le formuler de la manière suivante. Dans un graphe connexe non-orienté $G = (V, E)$, à chaque arête $(i, j) \in E$ est associée une quantité $q_{ij} \geq 0$ (la demande) et un coût de traversée $c_{ij} \geq 0$. V contient un sommet spécifique d , appelé *dépôt*. Une flotte de N véhicules, tous dotés d'une capacité de stockage identique C et positionnés initialement au dépôt, doit traverser toutes les arêtes et collecter les quantités associées sans jamais dépasser la limite C , puis retourner au dépôt.

Les applications du CARP sont nombreuses (entretien des rues, ramassage de déchets, livraison de courrier, ramassage scolaire, lecture de compteurs, salage des routes en hiver...). Des détails sur ces applications peuvent être trouvés dans [15, 16]. En particulier, l'application à notre problème est triviale : chaque arête est un tronçon de route, chaque sommet est une intersection et les q_{ij} sont les quantités de déchets. Les coûts c_{ij} correspondent typiquement à la longueur ou au temps de parcours des tronçons (i, j) . On appelle *arêtes requises* les arêtes appartenant à l'ensemble

$E_R := \{(i, j) \in E : q_{ij} > 0\}$. Pour tous les $(i, j) \in S^2$ tels que $(i, j) \notin E$, on pose par défaut $c_{ij} = -1$. Les formulations mathématiques du CARP utilisent souvent la programmation linéaire en nombres entiers qui décrit des problèmes d'optimisation ayant une fonction objectif et des contraintes d'inégalités linéaires par rapport à des variables de décision qui sont entières. Un exemple typique de formulation du CARP est le suivant :

Pour tous $(i, j) \in E$, $k \in \llbracket 1, N \rrbracket$, les valeurs x_{ij}^k et y_{ij}^k caractérisent les tournées. x_{ij}^k représente le nombre de fois que le véhicule k passe par l'arête (i, j) . y_{ij}^k est une variable binaire valant 1 si l'arête (i, j) est déblayée par le véhicule k et 0 sinon. La fonction objectif à minimiser est

$$\sum_{k=1}^N \sum_{e \in E} x_e^k c_e$$

et les variables de décision sont soumises à une liste de contraintes linéaires :

Chaque tournée démarre et s'achève au dépôt :

$$\forall k \in \llbracket 1, N \rrbracket, \quad \sum_i x_{di}^k = 1, \quad \sum_i x_{id}^k = 1$$

Chaque tournée suit une trajectoire continue :

$$\forall k \in \llbracket 1, N \rrbracket, \forall s \in S, \quad \sum_i x_{is}^k = \sum_i x_{si}^k$$

Les arêtes parcourues "existent" :

$$\forall k \in \llbracket 1, N \rrbracket, \forall (i, j) \in S^2, \quad x_{ij}^k c_{ij} \geq 0, \quad y_{ij}^k c_{ij} \geq 0$$

Les déchets ramassés par un véhicule ne dépassent pas sa capacité :

$$\forall k \in \llbracket 1, N \rrbracket, \quad \sum_{e \in E} y_e^k q_e \leq C$$

Une arête déblayée est nécessairement parcourue :

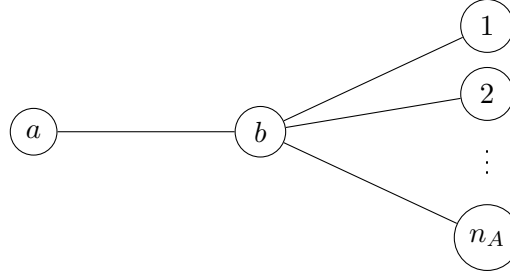
$$\forall k \in \llbracket 1, N \rrbracket, \forall e \in E, \quad x_e^k \geq y_e^k$$

Ce type de formulation est le plus répandu dans la littérature mais n'est pas unique. Par exemple, Belenguer et Benavent [3] proposent une formulation linéaire relaxée où les contraintes sont résumées en une seule ligne. Nous présenterons plus bas une formulation non-linéaire et plus intuitive, et à partir de laquelle nous développerons notre méthode.

L'avantage d'une formulation en programmation linéaire/entière est qu'il existe diverses méthodes heuristiques basées sur cette formulation pour résoudre le problème. Parmi celles-ci on peut citer la méthode des *plans sécants* [3, 32], des méthodes de *recherche locale* [7], une méthode d'*insertion parallèle* [9], la méthode du *Branch-and-Price* [11], des heuristiques *constructives* (path-scanning, construct and strike, augment-merge) [21, 37] et des méthodes de voisinage [25, 31, 38].

Plusieurs métaheuristiques ont également été appliquées au CARP, c'est le cas des algorithmes génétiques [18], des méthodes de recherche tabou [24], des colonies de fourmis [27, 30, 39], des algorithmes mémétiques [29]...

FIGURE 1 – Le problème de partition comme cas particulier du CARP



La difficulté à trouver des solutions approchées du CARP est accentuée par le fait que le problème consistant à trouver une solution 1,5-approchée au CARP est lui-même NP-difficile [22]. En effet, ce problème se réduit polynômialement au problème de partition que l'on définit ici :

Définition 1 (Problème de partition). Étant donné un ensemble fini A et une application $s : A \rightarrow \mathbb{N}$, existe-t-il un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$?

La réduction polynômiale du CARP au problème de partition consiste à montrer que n'importe quel problème de partition peut être transformé en un cas particulier du CARP (et que cette transformation s'effectue en un temps polynômial via une machine de Turing déterministe ; plus de détails sur la théorie de la complexité peuvent être trouvés dans [20]).

L'appartenance du CARP à la famille des problèmes NP-difficiles repose donc sur le résultat suivant :

Lemme 1 (Karp, [28]). *Le problème de partition est NP-complet.*

Théorème 1. *Soit O la valeur optimale du CARP. Le problème consistant à trouver une solution du CARP d'une valeur strictement inférieure à $1.5 \times O$ est NP-difficile.*

Démonstration. Étant donné un ensemble A et une fonction $s : A \rightarrow \mathbb{N}$, il s'agit de trouver un graphe sur lequel une solution 1,5-optimale du CARP détermine la réponse du problème de partition pour A et s . On construit donc le graphe constitué des sommets $\{a, b, 1, 2, \dots, n_A\}$, où a et b sont deux sommets quelconques, et où les sommets $1, 2, \dots, n_A$ indexent les éléments de A . Les sommets a et b sont reliés par une arête, et pour tout $i \in \llbracket 1, n_A \rrbracket$, les sommets b et i sont reliés par une arête (voir figure 1). L'arête $\{a, b\}$ a un coût $c_{ab} = 1$ et une demande $q_{ab} = 0$. Pour tout $i \in \llbracket 1, n_A \rrbracket$, on pose $c_{bi} = 0$ et $q_{bi} = s(i)$. Le sommet a correspond au dépôt. On attribue à chaque véhicule la capacité $C = \frac{1}{2} \sum_{i \in A} s(i)$.

La solution optimale du CARP sur ce graphe a un coût égal à 4 si la réponse du problème de partition est «oui» pour A et s . Elle a un coût supérieur ou égal à 6 si la réponse est «non». Ainsi, si une solution 1,5-optimale du CARP est < 6 , la réponse du problème de partition est «oui», sinon, la réponse est «non». On a donc montré que trouver une solution 1,5-approchée du CARP est au moins aussi difficile que de résoudre le problème de partition, i.e. c'est NP-difficile. \square

Ce résultat est un peu moins général que celui de Golden et Wong [22], mais permet de se rendre compte assez simplement que le CARP est NP-difficile.

2.2 Incertitudes

Comme nous le mentionnions dans la première section, une des principales difficultés des modèles de logistique d'urgence est d'attribuer une estimation précise de leurs paramètres. Lorsqu'il est appliqué à des opérations quotidiennes de ramassage de déchets, le CARP peut bénéficier d'estimations statistiques suffisamment précises pour les quantités de déchets. Ce n'est pas le cas dans une situation de catastrophe naturelle. Dans le modèle qu'on présente, les incertitudes portent sur deux éléments du graphe : les quantités de déchets et la disponibilité des tronçons de route. Fleury et al. [18] ont introduit un modèle de CARP stochastique (SCARP) dans lequel les demandes présentes sur les arêtes sont aléatoires. Mei et al. [35] ont rajouté au modèle SCARP l'incertitude sur la praticabilité des arêtes, et ont appelé UCARP le modèle qui en résultait. On présente ici une adaptation du modèle UCARP dans laquelle la disponibilité des tronçons de route pour les camions de ramassage correspond à un état de décrue post-inondation.

On présente d'abord notre formulation pour le CARP dans un environnement déterministe, inspirée de [35]. Contrairement à ce qu'on a présenté en section 2.1, cette formulation n'est pas linéaire et plus intuitive. En particulier, le concept de "tournée" est plus explicite, ce qui nous permet de nous faire directement une idée claire des solutions du problème. Cette formulation est par ailleurs suffisamment adaptée à la métaheuristique que nous introduirons plus loin. Pour cette raison, on conservera cette notation tout au long de ce rapport.

On se donne un graphe connexe non-orienté $G = (V, E)$, un coût de parcours c_{ij} (également noté $c(i, j)$) et une demande q_{ij} pour chaque arête $\{i, j\} \in E$. On note $E_R := \{e \in E : q_e > 0\}$ l'ensemble des arêtes requises, et on note $d \in V$ le dépôt. Une solution du CARP pour N véhicules de capacités C est un couple (X, Y) où $X = \{X_1, \dots, X_N\}$ est un ensemble de cycles et $Y = \{Y_1, \dots, Y_N\}$ un ensemble de vecteurs binaires associés à ces cycles. Une tournée X_k est une séquence de sommets $(x_{k1}, \dots, x_{kl_k})$ démarrant et terminant au dépôt, ie $x_{k1} = x_{kl_k} = d$. Le vecteur correspondant $Y_k = (y_{k1}, \dots, y_{kl_k-1}) \in \{0, 1\}^{l_k-1}$ détermine si une arête est servie ou non de la manière suivante : si $(x_{ki}, x_{k(i+1)})$ est une arête requise, alors $y_{ki} = 1$ si elle est servie par le véhicule k à la position courante, et $y_{ki} = 0$ sinon.

Le problème est alors le suivant :

$$\min_{X, Y} \sum_{k=1}^N \sum_{i=1}^{l_k-1} c(x_{ki}, x_{k(i+1)}) \quad (1)$$

sous les contraintes

$$\forall k \in \llbracket 1, N \rrbracket, \quad x_{k1} = x_{kl_k} = d \quad (2)$$

$$\sum_{k=1}^N \sum_{i=1}^{l_k-1} y_{ki} = |E_R| \quad (3)$$

$$\forall y_{ki} = 1, \quad (x_{ki}, x_{k(i+1)}) \in E_R \quad (4)$$

$$\forall y_{k_1 i_1} = 1, y_{k_2 i_2} = 1, (k_1, i_1) \neq (k_2, i_2), \quad \begin{cases} (x_{k_1 i_1}, x_{k_1(i_1+1)}) \neq (x_{k_2 i_2}, x_{k_2(i_2+1)}) \\ (x_{k_1 i_1}, x_{k_1(i_1+1)}) \neq (x_{k_2(i_2+1)}, x_{k_2 i_2}) \end{cases} \quad (5)$$

$$\forall k \in \llbracket 1, N \rrbracket, \quad \sum_{i=1}^{l_k-1} q(x_{ki}, x_{k(i+1)}) y_{ki} \leq C \quad (6)$$

$$\forall k \in \llbracket 1, N \rrbracket, \forall i \in \llbracket 1, l_k - 1 \rrbracket, \quad (x_{ki}, x_{k(i+1)}) \in E \quad (7)$$

La contrainte (2) permet de s'assurer que les tournées démarrent et s'achèvent bien au point de dépôt ; la contrainte (3) que toutes les arêtes sont bien servies ; la contrainte (4) que chaque arête servie est bien requise ; la contrainte (5) qu'une arête requise n'est pas servie plusieurs fois ; la contrainte (6) que les capacités des véhicules sont respectées ; et la contrainte (7) que les trajectoires sont continues.

Dans un problème de ramassage de déchets post-inondations, la collecte s'effectue une fois que la décrue a commencé et la hauteur d'eau courante de la crue est incertaine. La disponibilité d'une arête (i, j) dans le graphe E dépend de cette hauteur d'eau qui peut dépasser partiellement ou totalement le tronçon de route (i, j) correspondant. Dans ce cas, (i, j) n'est pas praticable. Lorsque plusieurs arêtes ne sont pas praticables, le CARP doit être résolu sur un sous-graphe G' de G (pourvu que ce sous-graphe soit toujours connexe ; la perte de la connexité impliquant la perte de toute solution admissible). En notant H la variable aléatoire représentant la hauteur d'eau au moment de la collecte et en attribuant à tout sommet $i \in V$ une altitude a_i , une réalisation $H(\omega)$ détermine le sous-graphe $G' = (V', E')$ de manière unique par :

$$V' = \{i \in V : a_i > H\} \quad ; \quad E' = \{(i, j) \in E : a_i, a_j > H\}$$

La variable aléatoire H suit donc une loi de probabilité sur \mathbb{R} dont on ne s'intéressera qu'aux réalisations entre l'altitude minimale ($\min_i a_i$) et l'altitude du dépôt (a_d). Chaque arête $(i, j) \in E$ est présente avec une probabilité

$$p_{ij} = \mathbb{P}(a_i \wedge a_j > H)$$

ce qui signifie qu'on peut parcourir le tronçon de route (i, j) si ses deux extrémités i et j sont au-dessus de la hauteur d'eau.

Lorsqu'une arête requise (i, j) est praticable, sa demande q_{ij} est déterminée par un paramètre environnemental aléatoire ξ . On note alors $q_{ij}(\xi)$ ou $q(i, j, \xi)$. Typiquement, ξ est un vecteur $(\xi_1, \dots, \xi_n) \in \mathbb{R}^n$ et $q_{ij}(\xi)$ se décompose ainsi :

$$q_{ij}(\xi) = \sum_{k=1}^n q_{ij}^k \xi_k$$

Les variables ξ_k peuvent représenter les facteurs de dégâts (par exemple la hauteur d'eau maximale atteinte pendant la crue, la durée de submersion, la vitesse du courant, la prévisibilité... [6]).

Les valeurs q_{ij}^k sont alors les coefficients de vulnérabilité des tronçons (i, j) aux différents facteurs associés. Ces coefficients dépendent donc directement du quartier dans lequel se trouve le tronçon (i, j) : ils peuvent être plus ou moins élevés suivant qu'il s'agisse d'une zone pavillonnaire, d'une zone industrielle...

L'introduction de variables aléatoires dans le modèle implique une remise en perspective du problème d'optimisation associé. L'approche traditionnelle lorsque le choix de la variable de décision est antérieur à la réalisation des paramètres aléatoires est l'*optimisation robuste*. Dans un cadre stochastique, les différentes réalisations possibles des paramètres mènent à plusieurs scénarii dont chacun a ses propres solutions admissibles et sa propre solution optimale. Le principe de l'optimisation robuste est de définir un *critère de robustesse* permettant de traiter le problème d'optimisation randomisé comme un problème d'optimisation classique.

L'approche que nous proposons est différente : au lieu de chercher une solution. On souhaite obtenir un ensemble de bonnes solutions associées à un ensemble de paramètres d'entrée. Cette approche sera détaillée en section 4.

3 Colonies de fourmis

Une métaheuristique est un ensemble de concepts algorithmiques qui peut être utilisé pour définir des méthodes heuristiques applicables à une grande variété de problèmes différents. Les métaheuristiques sont souvent inspirées de phénomènes d'optimisation observés dans la nature. On peut citer entre autres :

- le *recuit simulé* qui, à partir d'une solution initiale, effectue des modifications successives qui peuvent améliorer mais aussi, dans une certaine mesure, dégrader la solution courante afin d'explorer au mieux l'espace des solutions ;
- les *algorithmes génétiques* appliquent un principe artificiel de sélection naturelle à une population de solutions, cette population est soumise à des processus de mutation et de croisement puis donne lieu à une nouvelle population. Ces processus favorisent, au fur et à mesure des générations, la convergence de l'algorithme vers un "individu optimal" ;
- les *colonies de fourmis* sont quant à elles inspirées du comportement que les fourmis prennent pour trouver le plus court chemin entre leur nid et une source de nourriture. Les fourmis déposent des traces de phéromones sur leur passage et leur propension à choisir un chemin plutôt qu'un autre dépend positivement de la quantité de phéromones qui s'y trouve. L'intensité des phéromones sur le chemin le plus court augmente plus rapidement que sur les autres et, à terme, le plus court est celui qui est emprunté par la quasi-totalité des fourmis.

Les colonies de fourmis ont été employées dans le cadre de ce stage. On étudiera dans cette section comment le comportement naturel des fourmis a donné lieu à la mise en place d'algorithmes d'optimisation (3.1) et comment ceux-ci s'appliquent à des exemples significatifs tels que le *problème du voyageur de commerce* (3.2). On montrera ensuite comment adapter cette métaheuristique à un problème de type CARP (3.3).

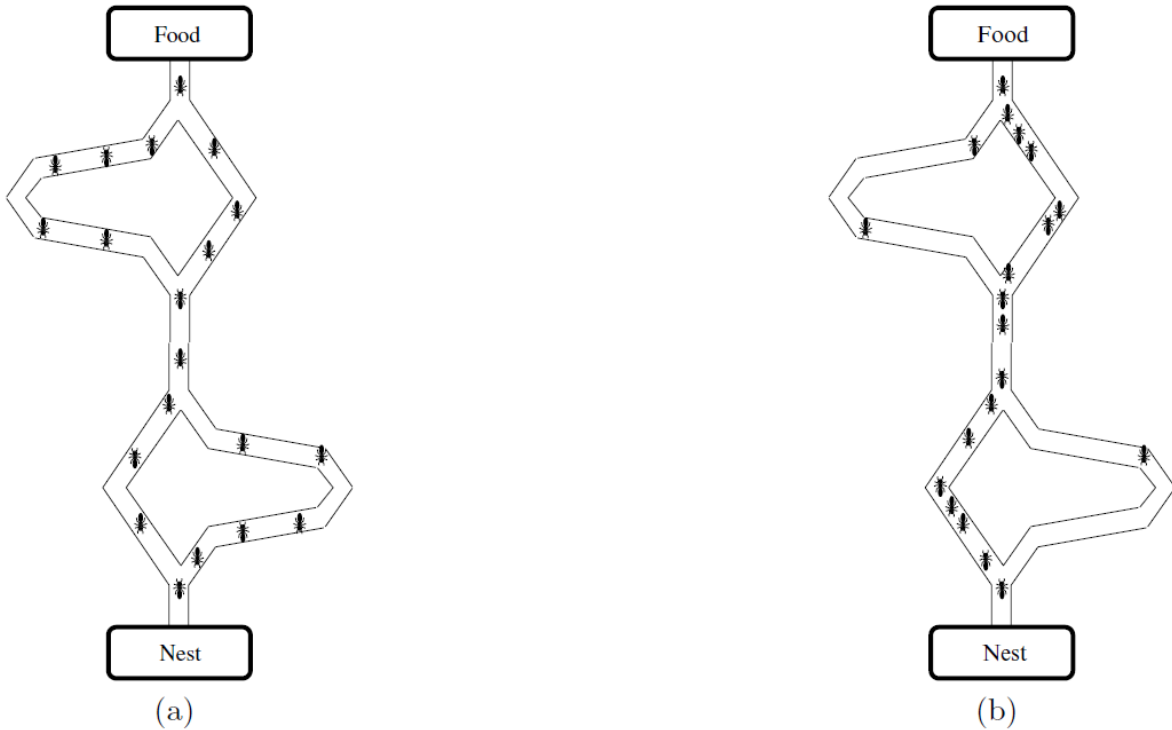
3.1 Principe général

Les algorithmes de colonies de fourmis ont été développés à partir d'une observation importante : les fourmis sont capables de résoudre collectivement des problèmes très complexes notamment lorsqu'il s'agit d'exploiter des ressources alimentaires [13]. Les fourmis communiquent entre elles via

des modifications de l'environnement, ce qu'on appelle la *stigmergie*. Elles déposent au sol une substance volatile appelée *phéromones* à partir d'une glande située sur leur abdomen. Elles perçoivent ces phéromones à l'aide de récepteurs situés dans leurs antennes. Chaque fourmi peut donc laisser sur un chemin qu'elle emprunte une *trace de phéromones* qui pourra être suivie par ses camarades.

Des expériences ont été réalisées pour étudier le comportement d'une colonie de fourmis lorsque deux chemins sont possibles entre la fourmilière et une source de nourriture [12, 23]. Ces expériences ont montré que lorsqu'il existe un chemin long et un chemin court entre le nid et la source, une fourmi arrivant à l'embranchement allait a priori (en l'absence de phéromones et dans l'impossibilité de se faire une idée globale de la longueur du chemin) choisir son chemin de manière arbitraire et équiprobable. En envoyant plusieurs fourmis successivement – étant donné qu'elles se déplacent à vitesse constante – les fourmis empruntant le chemin le plus court arrivent et repartent plus tôt de la source. Inexorablement, à partir d'un certain instant, le plus court chemin aura été visité plus souvent que le long (voir figure 2, tirée de [14]). Les traces de phéromones déposées sur ce chemin seront alors plus importantes, ce qui, in fine, induit la quasi-totalité des fourmis à suivre le chemin court.

FIGURE 2 – The Double Bridge Experiment. (a) Début de l'expérience ; (b) Fin de l'expérience.



Les expériences menées dans [12, 23] ont permis de déterminer un modèle stochastique décrivant le comportement des fourmis. À un instant donné, une fourmi située à l'embranchement choisit le chemin court avec probabilité

$$p_c = \frac{(t_c + \varphi_c)^\alpha}{(t_c + \varphi_c)^\alpha + (t_l + \varphi_l)^\alpha}$$

où t_c est le temps de parcours du chemin court, φ_c et φ_l sont les quantités de phéromones respecti-

vement sur les chemins court et long, et $\alpha > 0$ est une constante.

Un tel modèle établit les bases pour la mise en place d'algorithmes de plus courts chemins sur graphes, et plus généralement pour le développement de la métaheuristique des colonies de fourmis, applicable à une large variété de problèmes d'optimisation combinatoire. Un des exemples d'application les plus significatifs est le *problème du voyageur de commerce* (*traveling salesman problem* ou *TSP*) qui consiste à trouver le plus court cycle passant par tous les sommets d'un graphe. Dans la sous-section qui suit, on présente une application des colonies de fourmis au TSP, ce qui nous permettra d'introduire le formalisme général de ce type d'algorithme, puis de faire la transition vers le problème qui nous concerne : le CARP.

3.2 Le problème du voyageur de commerce

Le TSP est habituellement formulé sur un graphe complet, le principe étant de trouver le plus court chemin qui passe par chaque sommet et qui s'achève au sommet de départ (*cycle hamiltonien*). Ce problème est NP-complet, et de nombreuses méthodes heuristiques ont été formulées pour le résoudre. Formellement, le TSP se définit comme suit. Étant donné un graphe constitué des sommets $1, \dots, n$, et des distances $d(i, j)$ entre chaque paire de sommets (i, j) , l'objectif est de trouver une permutation $\sigma \in \mathcal{S}_n$ (\mathcal{S}_n étant le groupe symétrique de $\llbracket 1, n \rrbracket$) minimisant la quantité

$$\sum_{k=1}^{n-1} d(\sigma_k, \sigma_{k+1}) + d(\sigma_n, \sigma_1)$$

La méthode des colonies de fourmis pour le TSP va consister à envoyer successivement des fourmis artificielles parcourir tous les sommets du graphe. Autrement dit, chaque fourmi va construire une solution au TSP. Initialement, une quantité initiale de phéromones τ_0 est déposée entre chaque paire de sommets, ie $\forall (i, j) \in S^2, \tau_{ij} = \tau_0$. Une fourmi construit une solution en partant du dépôt $d \in S$ puis choisit le prochain sommet de manière aléatoire. Lors de la construction d'une solution, une double stratégie d'*intensification* et de *diversification* dirige la fourmi. L'intensification va orienter nos recherches vers des solutions prometteuses, elle va nous permettre notamment d'améliorer les bonnes solutions déjà trouvées. La diversification nous permet de mieux explorer l'espace des solutions. Supposons que la fourmi soit située sur un sommet i et qu'il reste un ensemble $J \in S$ de sommets non visités. Avec une probabilité $p_0 \in [0, 1]$, la stratégie d'intensification est choisie, c'est-à-dire que le prochain sommet $j \in J$ à visiter est celui qui vérifie

$$j = \arg \max_{k \in J} \left(\tau_{ik}^\alpha \eta_{ik}^\beta \right)$$

où η_{ik} est la *visibilité* du sommet k depuis le sommet i , qui est définie comme l'inverse de la distance entre i et k : $\eta_{ik} = 1/d(i, k)$.

Avec une probabilité $1 - p_0$, la stratégie de diversification est choisie, c'est-à-dire que le prochain sommet j est tiré parmi les éléments de J avec la probabilité suivant la *règle de transition proportionnelle aléatoire* suivante [8] :

$$p_{ij} = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{k \in J} \tau_{ik}^\alpha \eta_{ik}^\beta}.$$

À la fin de chaque itération, les traces de phéromones subissent deux modification. La première est l'évaporation : la matrice $(\tau_{ij})_{(i,j) \in S^2}$ subit la modification

$$\tau_{ij} \leftarrow \rho \tau_{ij}$$

où ρ est le *coefficient d'évaporation*. La deuxième est le *renouvellement global*, sur chaque arête visitée par la fourmi, la quantité de phéromones est renforcée en proportion de la qualité de la solution obtenue. Typiquement, ce renouvellement sur une arête (i, j) parcourue par la fourmi prend la forme

$$\tau_{ij} \leftarrow \tau_{ij} + \frac{C}{L}$$

où L est la longueur de la solution obtenue et $C > 0$ est une constante d'ajustement.

3.3 CARP

Une adaptation de cette méthode à un problème de parcours sur arcs comme le CARP, n'est pas naturelle mais est néanmoins possible. L'idée générale est la suivante : dans le TSP, chaque sommet n'était visité qu'une seule fois et on cherchait l'ordre optimal de visite de ces sommets ; dans le CARP, chaque arête requise n'est servie qu'une seule fois et on va rechercher l'ordre optimal de service de ces arêtes. (Pour bien concevoir cette idée d'"ordre", étant donné qu'on dispose de plusieurs véhicules, on peut imaginer que les véhicules effectuent leurs tournées successivement, i.e. le véhicule 2 quitte le dépôt une fois que le véhicule 1 y est revenu). Un ordre de service des arêtes ne définit cependant pas nécessairement une solution admissible : il est possible que pour un ordre donné, aucun enchaînement de véhicules ne satisfasse les contraintes de capacité. En effet, le CARP est en particulier un problème de *Bin Packing* (qui consiste à ranger des objets dans un certain nombre de boîtes). Pour cette raison, l'admissibilité des solutions fera l'objet d'un traitement particulier dans notre étude.

Contrairement au TSP on ne dispose plus les traces de phéromones explicitement sur le graphe mais de manière abstraite «entre les arcs requis» (où par "arc", on entend arête munie d'un sens d'orientation). Ainsi, lorsqu'un véhicule k effectue une tournée $X_k = (x_{k1}, x_{k2}, x_{k3}, x_{k4}, x_{k5}, x_{k6})$ avec le vecteur associé $Y_k = (1, 0, 0, 1, 1)$, cela implique un renforcement des phéromones entre les arcs (x_{k1}, x_{k2}) et (x_{k4}, x_{k5}) et entre les arcs (x_{k4}, x_{k5}) et (x_{k5}, x_{k6}) , ie les valeurs $\tau_{(x_{k1}, x_{k2}), (x_{k4}, x_{k5})}$ et $\tau_{(x_{k4}, x_{k5}), (x_{k5}, x_{k6})}$ sont renouvelées. De plus, on introduit une trace de phéromone initiale pour orienter le premier véhicule vers le tout premier arc à servir.

Comme pour le TSP, une itération correspond à la production d'une solution par une fourmi. C'est-à-dire qu'à chaque itération, une fourmi va construire N tournées qui correspondent aux N véhicules disponibles. La visibilité concerne le prochain arc à servir à partir du sommet courant sur lequel est située la fourmi. Elle dépend positivement de la demande sur les arcs non servis et négativement de la distance de ces arcs par rapport au sommet courant. On introduit une matrice $(D_{ij})_{i,j \in V}$ des distances des plus courts chemins entre chaque paire de sommets. On définit la visibilité de l'arc (i, j) depuis le sommet s par

$$\eta_{s,(i,j)} := \frac{q_{ij}}{1 + C \times D_{si}}$$

où $C > 0$ est une constante d'ajustement.

Une fourmi parcourt le un graphe de manière suivante. Lorsqu'elle vient de servir un arc (i, j) , le prochain arc à servir est choisi avec probabilité p_0 suivant une stratégie de diversification et

avec une probabilité $1 - p_0$ suivant une stratégie d'intensification. p_0 est le *coefficient d'équilibre intensification/diversification* et n'est pas nécessairement constant au fil des itérations. Lorsque la stratégie de diversification (resp. d'intensification) est choisie, les arcs (k, l) du graphe qui n'ont pas été servis et qui présentent une demande inférieure à la capacité courante du camion sont classés selon leurs valeurs $\eta_{j,(k,l)}$ (resp. $\eta_{j,(k,l)}^\alpha \tau_{(i,j),(k,l)}^\beta$). Les K meilleures valeurs de ce classement sont conservées, et l'arc (\bar{k}, \bar{l}) est choisi avec probabilité

$$\frac{\eta_{j,(\bar{k},\bar{l})}}{\sum_{\text{les } K \text{ meilleurs } (k,l)} \eta_{j,(k,l)}} \quad \left(\text{resp. } \frac{\eta_{j,(\bar{k},\bar{l})}^\alpha \tau_{(i,j),(\bar{k},\bar{l})}^\beta}{\sum_{\text{les } K \text{ meilleurs } (k,l)} \eta_{j,(k,l)}^\alpha \tau_{(i,j),(k,l)}^\beta} \right)$$

(où $\alpha, \beta > 0$).

Lorsque tous les arcs non servis ont une demande supérieure à la capacité courante du véhicule dirigé par la fourmi, celle-ci rentre au dépôt et entame la tournée du véhicule suivant pour laquelle le choix du premier arc à servir dépend des traces de phéromones qui démarrent au dernier arc servi de la tournée précédente.

Comme nous l'avons mentionné plus haut, lorsqu'une fourmi construit une solution, elle construit un ordre de service des arcs qui ne correspond pas nécessairement à une solution admissible. Pour qu'une solution soit admissible, il faut que toutes les arêtes requises soient servies. Cependant, deux solutions violant cette dernière contrainte peuvent avoir des qualités différentes au regard de leur mise en œuvre opérationnelle. De plus, des solutions non admissibles présentant un service "presque complet" du graphe peuvent s'avérer beaucoup plus prometteuses que d'autres solutions admissibles. Aussi, introduire des solutions non admissibles dans le processus de recherche permet d'atteindre des bonnes solutions plus efficacement qu'en se restreignant aux seules solutions admissibles, une telle restriction entravant notre liberté de mouvement dans l'espace des solutions. Pour cette raison, on propose une relaxation de la contrainte (3) : on s'autorise à considérer les solutions ne proposant pas un service complet du graphe. On introduit dans la fonction de coût une pénalité sur la perte de qualité due à des arêtes requises non servies dans la solution. La fonction de coût (1) devient alors :

$$\sum_{k=1}^N \sum_{i=1}^{l_k-1} c(x_{ki}, x_{k(i+1)}) + \gamma \sum_{(i,j) \text{ non servi}} q(i, j)$$

où $\gamma > 0$, et, de manière formelle,

$$\sum_{(i,j) \text{ non servi}} q(i, j) = \sum_{\{i,j\} \in E_R} \left(q(i, j) \times \left(1 - \sum_{k=1}^N \sum_{h=1}^{l_k-1} 1_{(x_{kh}, x_{k(h+1)})=(i,j), y_{kh}=1} \right) \right)$$

Finalement, le renouvellement des phéromones est très similaire avec ce qu'on a présenté pour le TSP. À la fin de chaque itération et pour toute paire d'arc (a, b) , la trace $\tau_{a,b}$ s'évapore avec un taux ρ (ie $\tau_{a,b} \leftarrow \rho \tau_{a,b}$). Et de plus, les traces de phéromones sur tous les arcs parcourus lors de l'itération courante sont renforcées de la manière suivante : $\tau_{a,b} \leftarrow \tau_{a,b} + f(L)$, où L est la valeur de la solution courante et f une fonction décroissante positive.

4 Méthode intrusive

4.1 Principe

La méthode des colonies de fourmis telle qu'on l'a présentée nous permet de résoudre de manière à peu près fiable le CARP dans un environnement déterministe. Autrement dit, elle nous fournit la *réponse* du modèle d'optimisation mis en place. Une identification des sources d'incertitudes nous permet de donner une description stochastique de l'aléa environnemental. Cette description a été présentée en section 2.2. On appelle *réponse aléatoire* du modèle la solution approchée et aléatoire du CARP, compte tenu de l'incertitude des paramètres d'entrée. Le principe de *propagation d'incertitudes* consiste à caractériser la réponse du modèle par rapport aux paramètres donnés en entrée [40]. Les méthodes de propagation d'incertitudes relèvent principalement du génie mécanique mais ont également été appliquées en génie chimique, en biologie, ou encore en électromagnétisme. À notre connaissance, elles n'ont jamais été adaptées à des problèmes d'optimisation combinatoire. Ces méthodes peuvent permettre d'estimer l'espérance et la variance de la réponse du modèle (*méthode de moments*), mais aussi d'étudier les caractéristiques de sa queue de distribution (*méthodes de fiabilité structurelle*), ou encore de déterminer complètement l'aléa de la réponse du modèle, i.e. sa densité de probabilité (*méthodes spectrales*).

L'approche que nous allons présenter peut s'associer à cette troisième catégorie. Comme nous l'avons formalisé en section 2, l'entrée du modèle est décrite par un vecteur aléatoire $(\xi_1, \dots, \xi_n, H) \in U \subset \mathbb{R}^{n+1}$. Ainsi, à chaque valeur (ξ, H) de U est associée une solution (approchée) $\mathcal{S}(\xi, H) = (X, Y)$ du CARP. On souhaite donc attribuer une solution à chaque élément de U . Une manière "brute" de calculer cette fonction est d'envoyer un à un à l'algorithme d'approximation les éléments d'un large échantillon de jeux de paramètres (fig. 3). Dans le sens où l'algorithme d'optimisation sera appliqué indépendamment pour chaque jeu de paramètre, on peut qualifier cette méthode de *non-intrusive*.

Une telle approche est en général coûteuse en temps de calcul. Des méthodes ont été développées pour approcher la surface de réponse le plus précisément possible tout en minimisant le nombre d'appels à l'algorithme [26]. Ces méthodes relèvent de l'optimisation bayésienne et sont très utiles lorsque l'algorithme en question est une « boîte noire ». Cependant, nous ne nous y intéresseront pas dans ce projet et nous proposons une approche différente qui repose en partie sur l'agrément que nous ne disposons pas d'une boîte noire mais d'un algorithme que nous avons développé. Notre approche part de l'observation suivante : les valeurs des solutions associées à deux jeux de paramètres d'entrée voisins sont souvent très proches l'une de l'autre. Cette idée est d'autant plus manifeste dans notre démarche d'optimisation par métaheuristique : on ne cherche pas spécifiquement des solutions optimales mais simplement des « bonnes solutions », l'hypothèse qu'on puisse trouver une bonne solution pour un jeu de paramètres fixé dans le voisinage d'une bonne solution trouvée pour un jeu de paramètres voisin paraît suffisamment vraisemblable. Concrètement, cela signifie que lorsque l'on dispose d'une tournée de véhicules efficace pour une certaine répartition des déchets, la conservation d'une efficacité raisonnable suite à légère modification de la disposition des déchets ne nécessite que de légères modifications dans les trajectoires des véhicules.

À partir de cette idée, on peut développer une méthode de calcul de la surface de réponse en prenant en compte, pour chaque jeu de paramètres, les solutions obtenues pour les jeux de paramètres à proximité. On qualifiera un telle démarche de *méthode intrusive*. L'objectif étant évidemment d'avoir un temps de calcul plus efficace qu'une méthode non-intrusive. Dans la figure 4, les jeux de paramètres sont représentés spatialement sur une grille dans \mathbb{R}^2 . L'idée est que dans

FIGURE 3 – Méthode non-intrusive

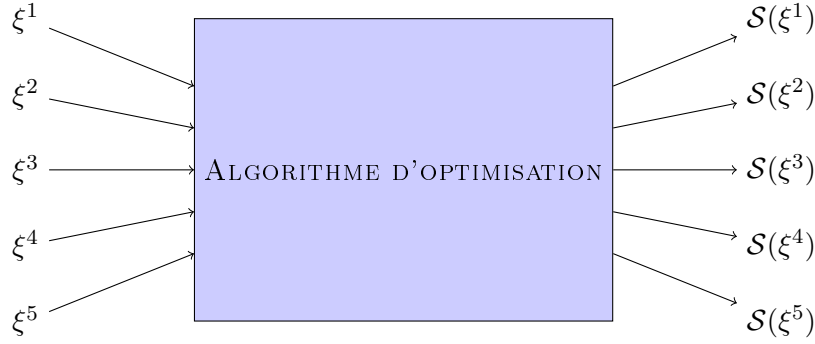
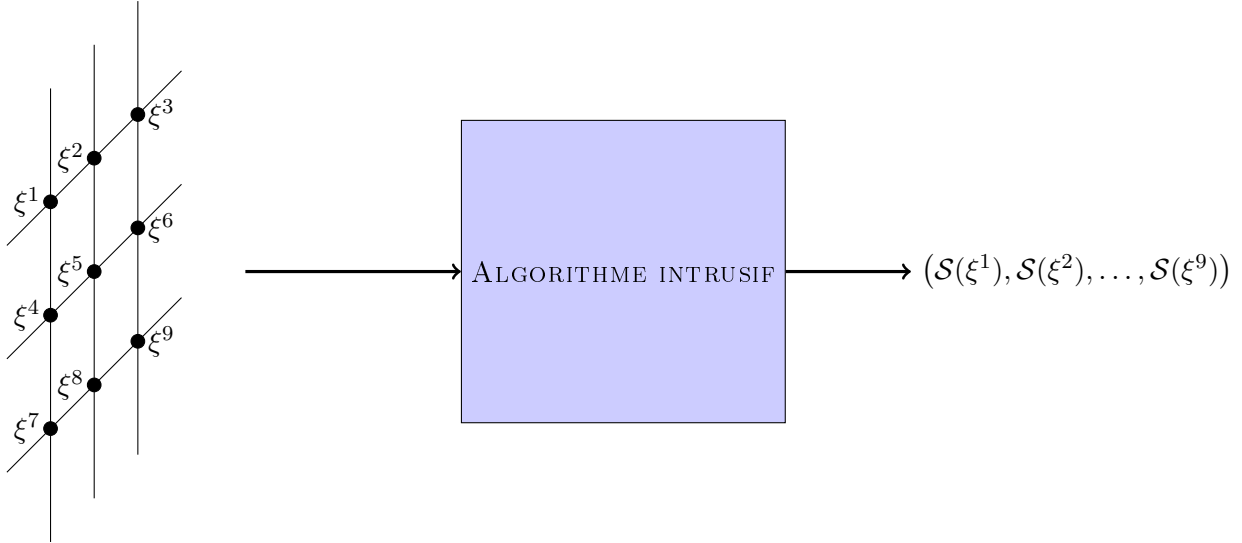


FIGURE 4 – Méthode intrusive



l'algorithme intrusif, la recherche d'une bonne solution pour le paramètre ξ^1 soit améliorée par la recherche des bonnes solutions pour les paramètres voisins ξ^2 , ξ^4 et ξ^5 . Réciproquement, ξ^1 va communiquer ses bons résultats à ses voisins et ainsi accélérer la recherche de leurs propres bonnes solutions.

Cette forme de communication doit alors être intégrée de manière efficace dans notre algorithme de colonies de fourmis qui, rappelons-le, est fondé sur le principe de la stigmergie, qui est elle-même une forme de communication entre les fourmis via des modifications de l'environnement. L'idée que nous proposons pour traiter l'incertitude – ou en tout cas une partie de cette incertitude – est de fusionner ces deux concepts de « communication ».

4.2 Les quantités de déchets : distribution transversale des phéromones

La répartition des déchets sur le territoire à la suite de l'inondation est l'incertitude la plus importante qui nous concerne. On utilise ici le modèle stochastique que nous avons introduit en section 2, à savoir que les quantités de déchets q_{ij} sur les arêtes sont décrites par un jeu de paramètres $\xi = (\xi_1, \xi_2, \dots, \xi_n)$ de manière linéaire : $q_{ij} = \sum_{k=1}^n q_{ij}^k \xi_k$. Chaque ξ_k est une variable aléatoire ayant

une fonction de répartition F_k . Toutes les variables ξ_k sont indépendantes entre elles. Une grille de jeux de paramètres convenable est alors

$$\left\{ \left(F_1^{-1} \left(\frac{i_1}{K} \right), F_2^{-1} \left(\frac{i_2}{K} \right), \dots, F_n^{-1} \left(\frac{i_n}{K} \right) \right) : (i_1, i_2, \dots, i_n) \in \llbracket 1, K-1 \rrbracket^n \right\} \quad (8)$$

où $K \in \mathbb{N}^*$ est un paramètre déterminant la taille de la grille.

Dans notre algorithme de colonies de fourmis dans un cadre déterministe, une itération correspondait à la construction d'une solution par une fourmi. L'adaptation de cette méthode à un environnement incertain se fait de la manière suivante : une itération correspond dorénavant à la construction de $(K-1)^n$ solutions pour les $(K-1)^n$ éléments de notre grille. À chaque itération, $(K-1)^n$ fourmis vont donc parcourir le graphe. Une trace de phéromones est mise en place pour chacun des $(K-1)^n$ paramètres. En construisant sa solution, chaque fourmi agit sur la trace de phéromones associée à son propre paramètre ainsi que – dans une moindre proportion – sur les traces associées aux paramètres voisins. Dit autrement, chaque fourmi est influencée par les résultats trouvés précédemment par les fourmis associées à son paramètre et par les fourmis associées à des paramètres voisins.

Formellement, lorsqu'une fourmi associée au paramètre ξ_i construit une solution ayant pour valeur $L(\xi_i)$ et dans laquelle les arcs a et b sont servis l'un après l'autre, la trace de phéromones τ^{ξ_i} subit un renouvellement

$$\tau_{ab}^{\xi_i} \longleftarrow \tau_{ab}^{\xi_i} + f(L(\xi_i))$$

et pour chaque paramètre ξ_j voisin de ξ_i , la trace de phéromones associée τ^{ξ_j} subit le renouvellement

$$\tau_{ab}^{\xi_j} \longleftarrow \tau_{ab}^{\xi_j} + \frac{1}{2} f(L(\xi_i))$$

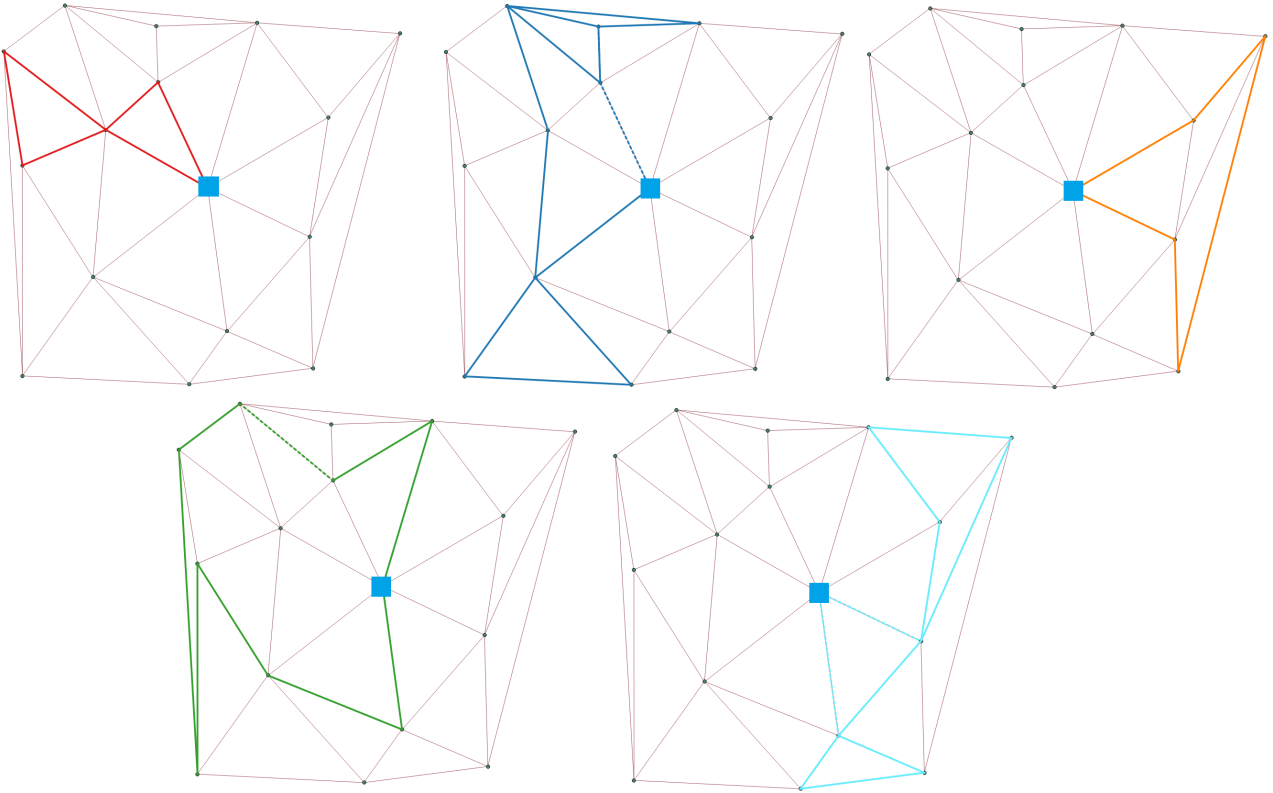
Ce partage des résultats intensifie le processus de recherche de solution pour chaque paramètre et permet de donner en un temps rapide une évaluation de la surface de réponse du modèle.

4.3 Hauteur d'eau : stratégie de réparation

La seconde source d'incertitudes est la hauteur d'eau courante H de la crue. Elle constitue une dimension de plus dans l'espace des données incertaine. La valeur H détermine la présence ou l'absence de certaines arêtes du graphe, et détermine ainsi le sous-graphe G' sur lequel notre problème doit finalement être résolu. Chronologiquement, H est la hauteur de crue au moment où la décision d'une tournée de véhicules doit être prise et sa réalisation est postérieure à celle des autres variables ξ_1, \dots, ξ_n qui agrègent les caractéristiques des dégâts causés par l'inondation.

Déterminer comment résoudre un problème d'optimisation sur un graphe inconnu est la problématique principale de l'optimisation combinatoire probabiliste [36]. En pratique, une fois obtenue la réalisation du sous-graphe G' , nous n'avons pas le temps de résoudre directement sur ce graphe le problème d'optimisation. L'idée est alors de résoudre le problème sur le graphe total G , ce qui nous fournit une solution a priori, puis de lui appliquer une stratégie de modification, i.e. un algorithme peu coûteux qui modifie la solution a priori pour la rendre admissible sur le sous-graphe effectif G' . Concrètement, il s'agit de trouver une tournée de véhicules optimale dans le cas où tous les tronçons de route sont accessibles, puis de trouver une manière de contourner les routes qui sont effectivement englouties. Dans notre cas, la stratégie de modification est assez naturelle et intuitive : les arcs à servir qui ne sont pas praticables sont retirés du trajet a priori, ceux qui restent sont reliés par les plus courts chemins possibles sur le graphe effectif G' .

FIGURE 5 – Une solution du CARP sur un graphe à 16 sommets.



Chaque couleur correspond à un véhicule. Le carré central est le point de dépôt. Les lignes en pointillés correspondent à un parcours sans service. Le graphe a été généré à partir des données du réseau des communes de l'Île-de-France. Ces données ont été construites dans le cadre du projet SIGOPT.

5 Algorithme et résultats

5.1 Environnement déterministe

L'algorithme de colonies de fourmis pour le CARP en environnement déterministe a été présenté en section 3.3, il est résumé dans le pseudo-code suivant :

```

pour chaque fourmi faire
  pour chaque véhicule faire
    tant que le véhicule n'est pas plein & il reste des arcs non servis faire
      Choix du prochain arc à servir suivant la règle de décision présentée en section 3.3;
    fin
  fin
  Renouvellement des phéromones;
fin
Retourner la meilleure solution obtenue

```

Algorithme 1 : Colonie de fourmis pour le CARP déterministe

La figure 5 représente une solution du CARP pour sur un graphe avec 16 sommets. Elle a été réalisée à l'aide du système d'information géographique QGIS.

TABLE 1 – Résultats pour l’algorithme déterministe

Fichier	V	E	N	C	Q	Optimum	Résultat	Itérations	R/O
val1A	24	39	2	200	358	173	173	6092	1
val1B	24	39	3	120	358	173	173	50000	1
val1C	24	39	8	45	358	245	294	50000	1.2
val2A	24	34	2	180	310	227	227	13038	1
val2B	24	34	3	120	310	259	260	50000	1.004
val2C	24	34	8	40	310	457	485	50000	1.06
val3A	24	35	2	80	137	81	81	9860	1
val3B	24	35	3	50	137	87	93	50000	1.07
val3C	24	35	7	20	137	138	140	50000	1.01
val4A	41	69	3	225	627	400	400	31887	1
val4B	41	69	4	170	627	412	430	50000	1.04

Pour le renouvellement des phéromones, nous avons choisi un renforcement de la forme $\tau_{a,b} \leftarrow \tau_{a,b} + ((L_{\text{seuil}} - L_{\min})_+)^2$, où L_{\min} est la valeur de la meilleure solution trouvée jusqu’à présent et L_{seuil} est une valeur au-dessus de laquelle les solutions sont considérées comme trop mauvaises pour engendrer un renouvellement des traces de phéromones. On a fixé $L_{\text{seuil}} := 1.2 \times L_{\min}$.

Le coefficient d’évaporation des phéromones ρ a été réglé à $\rho = 0.99$. Le coefficient de pénalisation de demande excessive γ est ajusté à la demande moyenne des arêtes requises : on a posé $\gamma = 4 \times \left(\frac{1}{|E_R|} \sum_{e \in E_R} q_e \right)^{-1}$. Autrement dit, ce coefficient est inversement proportionnel à la quantité de déchets moyenne présente sur les tronçons de route. Le coefficient d’équilibre intensification/diversification p_0 dépend de l’itération courante, il est réglé de sorte à ce que la diversification soit importante au début de la recherche d’une solution, puis il décroît rapidement afin de laisser place à un perfectionnement des bonnes solutions déjà découvertes. On a donc posé $p_0(n) = 0.01 \frac{n}{n_{\max}}$, où n est l’itération courante et n_{\max} est le nombre maximal d’itérations. De cette manière là, le degré de diversification est proche de 1 au début de la procédure et le degré d’intensification est proche de 1 à la fin de la procédure.

Pour évaluer les performances de l’algorithme, nous avons réalisé des tests sur des données «benchmark»³ régulièrement utilisées dans la littérature sur le CARP [29, 1, 30, 19, 2, 38, 39, 41, 34, 10]. Les graphes sur lesquels l’algorithme a été testé ont été proposés par Benavent et al. [4] ; ils ont un nombre de sommets allant de 24 à 41 et un nombre d’arêtes allant de 34 à 69. Les algorithmes ont été implémentés avec Scilab⁴ et exécutés sur un processeur Intel Core i5 à 3.30 GHz.

Les résultats obtenus ont été consignés dans le tableau 1. On donne pour chaque fichier le nombre de sommets du graphe, le nombre d’arêtes, le nombre de véhicules, la capacité de chaque véhicule, la valeur de l’optimum, la demande totale sur le territoire, la valeur de la meilleure solution trouvée, le nombre d’itérations réalisées, ainsi que le rapport Résultat / Optimum, qui donne une indication de la proximité notre meilleur résultat à la valeur optimale.

3. Ces données peuvent être téléchargées à l’adresse <http://www.uv.es/belenque/carp.html>.

4. Les codes réalisés lors de ce stage sont disponibles à l’adresse <https://github.com/IGNF/SIGOPT/tree/master/sigopt-uncertainty>

Dans cette procédure, nous avons fixé le nombre maximal d'itérations à $n_{\max} = 50000$, si l'optimum est atteint avant ce nombre d'itérations, l'algorithme s'arrête.

On peut constater que l'atteinte de l'optimum est plus difficile lorsque le ratio $N \times C/Q$ est faible. En effet, les contraintes de capacité sont d'autant plus importantes que $N \times C/Q$ est proche de 1. Les solutions admissibles sont moins nombreuses et par conséquent, l'exploration de l'espace des solutions est plus délicate. La problématique de distance minimale s'efface au profit de la problématique du Bin Packing.

Par ailleurs, il est à noter qu'à la connaissance de l'auteur, un seul algorithme est parvenu à atteindre l'optimum pour l'ensemble des jeux de données sur lequel nous avons fait nos tests. Il s'agit d'une métaheuristique hybride proposée par Chen et al. [10] en 2016.

Ces résultats permettent de valider notre algorithme en vérifiant empiriquement la convergence vers une valeur suffisamment proche de l'optimum. Cette validation nous offre une meilleure appréciation des tests d'efficacité de l'algorithme intrusif.

5.2 Environnement aléatoire / Méthode intrusive

L'algorithme «intrusif» pour le CARP en environnement incertain a été présenté en section 4. On le résume dans le pseudo-code suivant :

```

pour chaque paramètre  $\xi$  de la grille (8) faire
  pour chaque fourmi associée à  $\xi$  faire
    pour chaque véhicule faire
      tant que le véhicule n'est pas plein & il reste des arcs non servis faire
        Choix du prochain arc à servir suivant la règle de décision présentée en section 3.3;
      fin
    fin
    Renouvellement de la trace de phéromones  $\tau^\xi$ ;
    pour chaque paramètre  $\xi'$  voisin de  $\xi$  faire
      Renouvellement de la trace de phéromones  $\tau^{\xi'}$ ;
    fin
  fin
fin
Retourner les meilleures solution obtenues pour chaque paramètre  $\xi$ ;

```

Algorithme 2 : Algorithme intrusif pour le CARP en environnement incertain

L'objectif de l'algorithme intrusif étant d'accélérer la convergence vers l'optimum de tout un échantillon de paramètres d'entrée, une manière de mesurer son efficacité est de le comparer à une méthode où chaque élément de l'échantillon est calculé par l'algorithme «non-intrusif», i.e. l'algorithme standard de la section 5.1. On propose alors le test consistant à comparer les résultats obtenus par la méthode intrusive et la méthode brute pour un nombre d'itérations fixé.

Comme il n'existe pas de données pour le CARP en environnement incertain, nous avons développé une méthode de randomisation de benchmarks. Pour ces tests, seul l'aléa sur les demandes des arêtes du graphe a été pris en compte. En respectant les notations de la section 4.2, on note n le nombre de paramètres environnementaux, i.e. la dimension de l'espace stochastique. En prenant en entrée un graphe pour le CARP avec des demandes déterministes $q_{i,j}$ pour chaque arête $\{i, j\}$,

TABLE 2 – Test d’efficacité de la méthode intrusive. À gauche : la méthode intrusive. À droite : la méthode "brute".

(ξ_1, ξ_2)	$\xi_1^{(1)}$	$\xi_1^{(2)}$	$\xi_1^{(3)}$	$\xi_1^{(4)}$	$\xi_1^{(5)}$	$\xi_1^{(6)}$	(ξ_1, ξ_2)	$\xi_1^{(1)}$	$\xi_1^{(2)}$	$\xi_1^{(3)}$	$\xi_1^{(4)}$	$\xi_1^{(5)}$	$\xi_1^{(6)}$
$\xi_2^{(1)}$	173	173	173	173	173	173	$\xi_2^{(1)}$	179	183	177	186	183.87	186
$\xi_2^{(2)}$	173	173	173	173	173	173	$\xi_2^{(2)}$	185	173	182	198	179	181
$\xi_2^{(3)}$	173	173	173	173	173	173	$\xi_2^{(3)}$	187.96	192	181	186	188	199
$\xi_2^{(4)}$	173	173	173	173	173	173	$\xi_2^{(4)}$	187	187	194	181	196	200
$\xi_2^{(5)}$	173	173	173	173	173	173	$\xi_2^{(5)}$	198	197	185	195	201	199
$\xi_2^{(6)}$	173	173	173	173	173	173	$\xi_2^{(6)}$	174	173	186	186	195	185

on a attribué à chaque valeur ξ_i une loi gamma dont l’espérance est fixée de la manière suivante :

$$\forall i \in \llbracket 1, n \rrbracket, \quad \mathbb{E}[\xi_i] = \frac{1}{n \times |E_R|} \sum_{e \in E_R} q_e$$

et la variance $\text{Var}(\xi_i)$ de chaque paramètre ξ_i correspond à la variance statistique des des demandes sur les arêtes requises. Les ξ_1, \dots, ξ_n sont donc i.i.d et de loi $\Gamma\left(\frac{\mathbb{E}[\xi_1]^2}{\text{Var}(\xi_1)}, \frac{\mathbb{E}[\xi_1]}{\text{Var}(\xi_1)}\right)$, où les deux paramètres entre parenthèses sont respectivement le paramètre de forme et le paramètre d’échelle de la loi gamma. Les coefficients q_{ij}^k pour le graphe aléatoire sortant ont été fixés à $q_{ij}^k = \frac{q_{ij}}{n \times \mathbb{E}[\xi_1]}$. De cette manière là, on a

$$\mathbb{E}\left[\sum_{k=1}^n q_{ij}^k \xi_k\right] = \sum_{k=1}^n \frac{q_{ij}}{n \mathbb{E}[\xi_1]} \mathbb{E}[\xi_1] = q_{ij}$$

et l’espérance du graphe déterministe en entrée est donc conservée. Une grille de paramètres est alors construite à partir de la formule (8) en choisissant un paramètre K .

Le graphe que nous obtenons en sortie de cette procédure est donc un graphe sur lequel les demandes fluctuent autour de la valeur fixée par le graphe déterministe donné en entrée. Plus ces demandes deviennent importantes, moins les solutions admissibles sont nombreuses ; allant parfois jusqu’à une absence totale de solutions admissibles. Pour cette raison, on décide d’adapter le nombre de véhicules aux quantités de déchets disposées sur les routes. En notant Q la demande totale du graphe, on attribue $N = \lceil \frac{Q \times 1,1}{C} \rceil$ véhicules ; le coefficient 1,1 servant à donner une marge facilitant la collecte.

Dans le tableau 2, on compare les résultats obtenus par la méthode intrusive et la méthode brute appliquées au graphe `val1A` randomisé. On a posé $K - 1 = 6$ et $n = 2$. L’ensemble des paramètres peut donc se disposer sur une grille de taille $6 \times 6 = 36$. Dans cette grille, les variables aléatoires ξ_1 et ξ_2 prennent chacune six valeurs possibles qu’on note $\xi_1^{(1)}, \dots, \xi_1^{(6)}$ et $\xi_2^{(1)}, \dots, \xi_2^{(6)}$. Dans le tableau 2, on a reporté à gauche les résultats obtenus par la méthode intrusive et à droite les résultats obtenus par la méthode brute (la différence entre ces méthodes étant simplement la communication entre les différents paramètres) pour un nombre d’itérations fixé à $n_{\max} = 200$. Les résultats de l’algorithme intrusif, tous identiques, indiquent que la distance minimale a été atteinte

FIGURE 6 – Résultat de la méthode intrusive pour le paramètre $(\xi_1^{(1)}, \xi_2^{(1)})$.

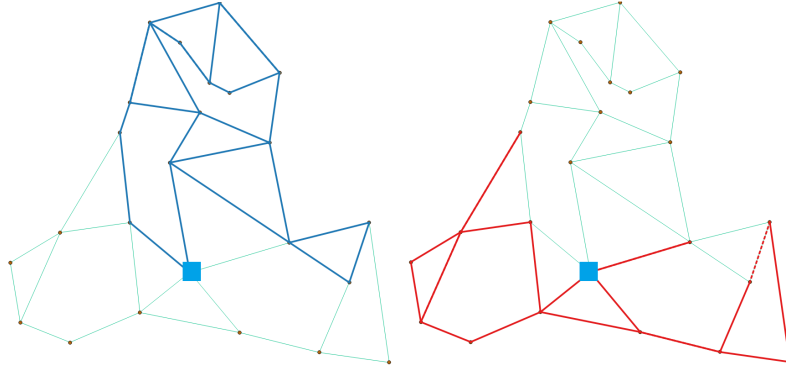
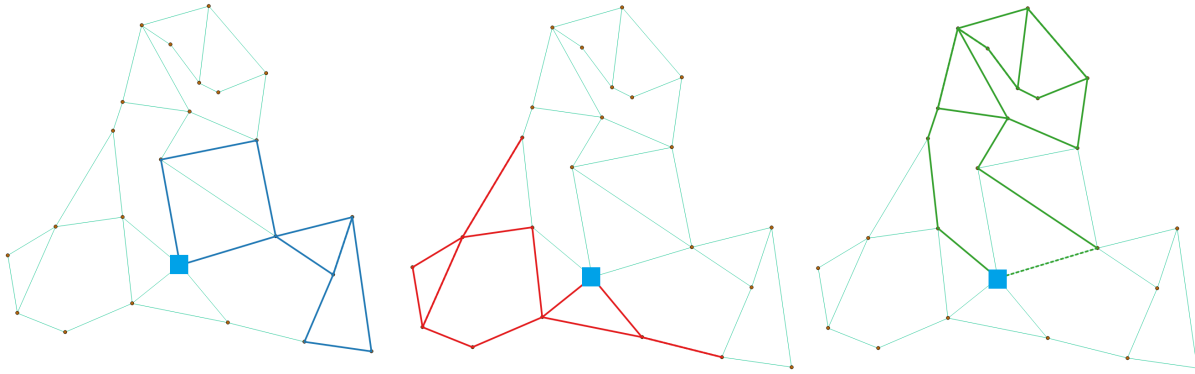


FIGURE 7 – Résultat de la méthode intrusive pour le paramètre $(\xi_1^{(6)}, \xi_2^{(6)})$.



(les figures 6 et 7 représentent les résultats obtenus pour les paramètres $(\xi_1^{(1)}, \xi_2^{(1)})$ et $(\xi_1^{(6)}, \xi_2^{(6)})$, les cases correspondantes sont en rouge dans le tableau 2, ces deux tournées ont des longueurs identiques). Les résultats de la méthode brute, dans le tableau de droite, soulignent en revanche une recherche plus laborieuse de valeur optimale.

Il est à noter que la méthode intrusive nécessite un temps de calcul supplémentaire par rapport à la méthode brute qui est dû à la propagation des traces de phéromones entre les paramètres. Cependant, ce temps supplémentaire consiste en quelques opérations d'affectation par itération et est considéré comme négligeable par rapport au temps qu'engendrerait une construction effective de solutions pour atteindre les mêmes résultats.

5.3 Conclusion et perspectives

Le CARP est un problème ayant un intérêt pratique important et représente un défi calculatoire significatif dû à sa NP-difficulté. Ce défi est encore plus considérable lorsque les données de ce problème sont incertaines, ce qui est notamment le cas lorsqu'il est appliqué à des événements difficiles à prévoir tels que des catastrophes naturelles. Dans ces circonstances, en fixant pour objectif de chercher un ensemble de solutions adapté à la variabilité des scénarios envisagés, et étant contraints par des temps de calcul potentiellement inconvenants, on propose une méthode d'opti-

misation «intrusive» consistant à optimiser globalement un échantillon de ces scénarios en prenant avantage de la similarité qui peut exister parmi eux. Les résultats obtenus, notamment ceux du tableau 2 montrent que l'idée d'un traitement intrusif des incertitudes peut effectivement mener à une économie substantielle de temps de calcul.

À l'issue de cette étude, des directions s'ouvrent à la fois dans une direction théorique et dans une direction applicative. En ce qui concerne la direction théorique, il faut remarquer que le caractère «intrusif» de la méthode que nous proposons est fortement dépendant de l'heuristique que nous avons choisi. Il peut être intéressant d'évaluer dans quelle mesure ce concept peut s'étendre à d'autres formes d'optimisation, et s'il est possible de développer des méthodes génériques de «communication de l'information des bonnes solutions» pour certains types de problèmes d'optimisation. Dans cette perspective, il peut être également intéressant de chercher quelle forme de communication est la plus efficace, de redéfinir les concepts de voisinage des paramètres, ou encore de déterminer la meilleure manière d'échantillonner ces paramètres selon le problème en question.

En ce qui concerne la direction applicative, plusieurs éléments du CARP en environnement incertain peuvent être révisés afin de convenir à une mise en œuvre opérationnelle. Un premier ajustement peut par exemple concerner la prise en compte des sens interdits et donc de considérer des graphes orientés, ce qui correspondrait à une représentation territoriale plus réaliste. Également, des indications plus précises sur les temps de parcours des véhicules peuvent être tirées du modèle notamment à partir d'informations sur l'ampleur des dégâts causés par le sinistre. Enfin, la stratégie de réparation relative à la praticabilité des voies – qui n'a pas fait l'objet d'une attention particulière lors de nos tests numériques du fait qu'elle consistait essentiellement à adapter des solutions non admissibles à l'environnement dégradé dans lequel elles se trouvaient – peut se considérer d'un point de vue pratique, en particulier si on l'adapte à un modèle dynamique dans lequel les rues impraticables sont de moins en moins nombreuses à mesure que la décrue progresse.

Références

- [1] S. AMPONSAH AND S. SALHI, *The investigation of a class of capacitated arc routing problems : the collection of garbage in developing countries*, Waste Management, 24 (2004), pp. 711 – 721.
- [2] R. BALDACC AND V. MANIEZZO, *Exact methods based on node-routing formulations for undirected arc-routing problems*, Networks, (2006), pp. 53–60.
- [3] J. M. BELENGUER AND E. BENAVENT, *A cutting plane algorithm for the capacitated arc routing problem*, Computers & Operations Research, 30 (2003), pp. 705–728.
- [4] E. BENAVENT, V. CAMPOS, A. CORBERÁN, AND E. MOTA, *The capacitated arc routing problem. A heuristic algorithm*, Qüestiió, 14 (1990), pp. 107–122.
- [5] H. BERAUD, *Initier la résilience du service de gestion des déchets aux catastrophes naturelles : le cas des territoires urbains et de l'inondation*, PhD thesis, Université Paris-Est, Marne-la-Vallée, Février 2013.
- [6] H. BERAUD, J. JADOT, B. BARROCA, G. HUBERT, AND N. BAUDUCEAU, *Mécadépi : «Méthode d'évaluation et de caractérisation des déchets post-inondation»*, rapport final, 2012.
- [7] P. BEULLENS, L. MUYLDERMANS, D. CATTRYSSE, AND D. V. OUDHEUSDEN, *A guided local search heuristic for the capacitated arc routing problem*, European Journal of Operational Research, 147 (2003), pp. 629–643.

- [8] E. BONABEAU, M. DORIGO, AND G. THERAULAZ, *Swarm intelligence*, Santa Fe Institute studies in the sciences of complexity, Oxford University Press, 1999.
- [9] L. CHAPLEAU, J. A. FERLAND, G. LAPALME, AND J.-M. ROUSSEAU, *A parallel insert method for the capacitated arc routing problem*, Operations Research Letters, 3 (1984), pp. 95–99.
- [10] Y. CHEN, J.-K. HAO, AND F. GLOVER, *A hybrid metaheuristic approach for the capacitated arc routing problem*, European Journal of Operational Research, 253 (2016), pp. 25–39.
- [11] C. H. CHRISTIANSEN, J. LYSGAARD, AND S. WØHLK, *A branch-and-price algorithm for the capacitated arc routing problem with stochastic demands*, Operations Research Letters, 37 (2009), pp. 392–398.
- [12] J.-L. DENEUBOURG, S. ARON, S. GOSS, AND J. M. PASTEELS, *The self-organizing exploratory pattern of the Argentine ant*, Journal of Insect Behavior, 3 (1990), pp. 159–168.
- [13] M. DORIGO AND T. STÜTZLE, *Ant Colony Optimization*, MIT Press, A Bradford book, 2004.
- [14] J. DRÉO, A. PÉROWSKI, P. SIARRY, AND E. TAILLARD, *Metaheuristics for Hard Optimization*, Springer Berlin Heidelberg, 2006.
- [15] H. A. EISELT, M. GENDREAU, AND G. LAPORTE, *Arc routing problems, Part I : The Chinese postman problem*, Operations Research, 43 (1995), pp. 231–242.
- [16] ———, *Arc routing problems, Part II : The rural postman problem*, Operations Research, 43 (1995), pp. 399–414.
- [17] L. EULER, *Solutio problematis ad geometriam situs pertinentis*, Commentarii academiae scientiarum Petropolitanae, 8 (1741), pp. 128–140.
- [18] G. FLEURY, P. LACOMME, AND C. PRINS, *Evolutionary algorithms for stochastic arc routing problems*, in Applications of Evolutionary Computing, G. R. R. et al., ed., Coimbra, Portugal, April 2004, EvoWorkshops 2004, Springer Berlin Heidelberg, pp. 501–512.
- [19] ———, *Stochastic capacitated arc routing problem*, Research Report RR-05-12, LIMOS, Aubière, Septembre 2005.
- [20] M. R. GAREY AND D. S. JOHNSON, *Computers and intractability*, A series of books in the mathematical sciences, W. H. Freeman and Company, New York, 1979.
- [21] B. L. GOLDEN, J. S. DEARMON, AND E. K. BAKER, *Computational experiments with algorithms for a class of routing problems*, Comput. & Ops. Res., 10 (1983), pp. 47–59.
- [22] B. L. GOLDEN AND R. T. WONG, *Capacitated arc routing problems*, Networks, 11 (1981), pp. 305–315.
- [23] S. GOSS, S. ARON, J. L. DENEUBOURG, AND J. M. PASTEELS, *Self-organized shortcuts in the Argentine ant*, Naturwissenschaften, 76 (1989), pp. 579–581.
- [24] P. GREISTORFER, *A tabu scatter search metaheuristic for the arc routing problem*, Computers & Industrial Engineering, 44 (2003), pp. 249–266.
- [25] A. HERTZ AND M. MITTAZ, *A variable neighborhood descent algorithm for the undirected capacitated arc routing problem*, Transportation Science, 35 (2001), pp. 425–434.
- [26] D. R. JONES, M. SCHONLAU, AND W. J. WELCH, *Efficient global optimization of expensive black-box functions*, Journal of Global Optimization, 13 (1998), pp. 455–492.
- [27] A. KANSOU AND A. YASSINE, *Splitting algorithms for the multiple depot arc routing problem : application by ant colony optimization*, International Journal of Combinatorial Optimization Problems and Informatics, 3 (2012), pp. 20–34.

- [28] R. M. KARP, *Reducibility among combinatorial problems*, Complexity of Computer Computations, (1972), pp. 85–103.
- [29] P. LACOMME, C. PRINS, AND W. RAMDANE-CHERIF, *Competitive memetic algorithms for arc routing problems*, Research Report LOSI-2001-01, LOSI, Université de Technologie de Troyes, Troyes, Novembre 2001.
- [30] P. LACOMME, C. PRINS, AND A. TANGUY, *First competitive ant colony scheme for the CARP*, Research Report RR-04-21, LIMOS, Clermont-Ferrand, Juillet 2004.
- [31] G. LAPORTE, R. MUSMANNO, AND F. VOCATURO, *An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands*, Transportation Science, 44 (2010), pp. 125–135.
- [32] A. N. LETCHFORD AND R. W. EGGLESE, *The rural postman problem with deadline classes*, European Journal of Operational Research, 105 (1998), pp. 390–400.
- [33] L. LUTHER, *Managing disaster debris : Overview of regulatory requirements, agency roles, and selected challenges*, CRS Report for Congress 7-5700, Congressional Research Service, January 2011.
- [34] R. MARTINELLI, M. POGGI, AND A. SUBRAMANIAN, *Improved bounds for large scale capacitated arc routing problem*, Computers & Operations Research, 40 (2013), pp. 2145–2160.
- [35] Y. MEI, K. TANG, AND X. YAO, *Capacitated arc routing problem in uncertain environments*, in Proc. of the 2010 IEEE Congress on Evolutionary Computation (CEC2010), Barcelona, Spain, July 2010, IEEE, pp. 1400–1407.
- [36] C. MURAT AND V. T. PASCHOS, *Probabilistic Combinatorial Optimization on Graphs*, Wiley-ISTE, March 2006.
- [37] W. L. PEARN, *Augment-insert algorithms for the capacitated arc routing problem*, Computers Ops Res., 18 (1991), pp. 189–198.
- [38] M. POLACEK, K. F. DOERNER, R. F. HARTL, AND V. MANIEZZO, *A variable neighborhood search for the capacitated arc routing problem with intermediate facilities*, J Heuristics, 14 (2008), pp. 405–423.
- [39] L. SANTOS, J. COUTINHO-RODRIGUES, AND J. R. CURRENT, *An improved ant colony optimization based algorithm for the capacitated arc routing problem*, Transportation Research Part B : Methodological, 44 (2010), pp. 246–266.
- [40] B. SUDRET, *Uncertainty propagation and sensitivity analysis in mechanical models – Contributions to structural reliability and stochastic spectral methods*, 2007. Habilitation à diriger des recherches, Université Blaise Pascal, Clermont-Ferrand, France.
- [41] F. L. USBERTI, P. M. FRANÇA, AND A. L. M. FRANÇA, *GRASP with evolutionary path-relinking for the capacitated arc routing problem*, Computers & Operations Research, 40 (2013), pp. 3026–3217.