REPUBLIQUE DU CAMEROUN
PAIX - TRAVAIL – PATRIE
****
UNIVERSITE DE DSCHANG
****
ÉCOLE DOCTORALE

REPUBLIC OF CAMEROON
PEACE - WORK - FATHERLAND
****
UNIVERSITY OF DSCHANG
****
POST GRADUATE SCHOOL

DSCHANG SCHOOL OF SCIENCES AND TECHNOLOGY

UNITE DE RECHERCHE EN INFORMATIQUE FONDAMENTALE, INGENIERIE ET APPLICATIONS (URIFIA)

SUJET :

METHODE INTELLIGENTE DE RESOLUTION DU PROBLEME DE COLLECTE ET DE LIVRAISON SELECTIF AVEC DEMANDES INCERTAINES

Projet de INF511 - Veille Scientifique en vue de l'obtention du :
Diplôme de Master en Informatique

Option : Informatique Fondamentale
Spécialité : Intelligence Artificielle

Par :

N°	Matricule	Noms & prénoms
1	CM-UDS-19SCI2083	NGUEWOUO MBAKOP Manick
2	CM-UDS-19SCI 0932	NGATCHE NJANTOU Dalisha Mylove

Sous la direction de :

Dr SOH Mathurin
Chargé de cours, Université de Dschang, Cameroun

Année académique 2023/2024

Méthode intelligente de résolution du problème de collecte et de livraison sélectif avec demandes incertaines

NGUEWOUO MBAKOP Manick - CM-UDS-19SCI2083
NGATCHE NJANTOU Dalisha Mylove - CM-UDS-19SCI0932

Responsable Dr SOH Mathurin

17 janvier 2024

Résumé

Le problème de collecte et de livraison sélectif avec demandes incertaines est un problème d'optimisation combinatoire qui consiste à transporter des marchandises entre des points de collecte et de livraison, en utilisant un ensemble de véhicules, tout en respectant un ensemble de contraintes. Ce problème a de nombreuses applications pratiques, notamment dans la logistique urbaine, et présente des enjeux économiques, sociaux et environnementaux importants. Ce problème est également difficile à résoudre, car il comporte des incertitudes sur les demandes des clients, qui peuvent varier selon des scénarios possibles.

Dans notre travail, nous proposons une méthode intelligente basée sur le reinforcement learning pour résoudre ce problème. Le reinforcement learning est une technique d'apprentissage par renforcement qui permet à un agent d'apprendre une politique optimale pour maximiser une récompense à long terme, en interagissant avec son environnement. Nous formulons le problème comme un processus décisionnel de Markov, et nous utilisons un algorithme d'apprentissage, comme Q-learning ou SARSA, pour apprendre une politique optimale à partir des données. Nous utilisons ensuite la politique apprise pour générer et évaluer des solutions pour chaque scénario.

Nous appliquons la méthode intelligente à un cas pratique inspiré d'une étude de cas réelle, menée par la société LogiX, qui propose des services de transport collaboratif dans la région parisienne. Nous comparons les résultats obtenus par la méthode intelligente avec ceux obtenus par les autres méthodes existantes, comme la méthode exacte, la méthode approchée basée sur un algorithme génétique, et la méthode hybride basée sur un algorithme génétique et un solveur.

Nous montrons que la méthode intelligente basée sur le reinforcement learning présente des avantages par rapport aux autres méthodes existantes, comme la performance, la complexité, la flexibilité et la robustesse. Nous concluons que la méthode intelligente basée sur le reinforcement learning est une méthode performante, rapide, flexible et robuste pour résoudre le problème de collecte et de livraison sélectif avec demandes incertaines.

1 Introduction générale

La logistique urbaine est l'ensemble des activités liées au transport, à la manutention, au stockage et à la distribution de marchandises dans les zones urbaines. La logistique urbaine est un domaine stratégique pour le développement économique, social et environnemental des villes, car elle contribue à la satisfaction des besoins des citoyens,

à la réduction des coûts et des émissions de gaz à effet de serre, et à l'amélioration de la qualité de vie et de la mobilité urbaine. La logistique urbaine est également un domaine complexe et dynamique, car elle doit faire face à de nombreux défis, comme la croissance démographique, la congestion du trafic, la diversité des acteurs, la variabilité de la demande, ou la réglementation urbaine. Ces défis nécessitent de concevoir et de mettre en œuvre des solutions innovantes, efficaces et durables, qui optimisent l'utilisation des ressources disponibles, tout en respectant les contraintes et les objectifs des parties prenantes. Parmi les solutions innovantes, on peut citer le transport collaboratif, qui consiste à mutualiser les capacités de transport de différents acteurs, comme les transporteurs professionnels, les particuliers, ou les commerçants, pour réaliser des opérations de collecte et de livraison de marchandises. Le transport collaboratif permet de réduire les coûts, les émissions, et les kilomètres à vide, tout en augmentant le taux de remplissage, la qualité de service, et la satisfaction des clients.

Le transport collaboratif pose cependant un problème d'optimisation combinatoire, qui consiste à planifier les tournées des véhicules, en tenant compte des contraintes de capacité, de fenêtres temporelles, de demandes appairées et d'incertitudes sur les demandes. Ce problème est appelé le problème de collecte et de livraison sélectif avec demandes incertaines, et il peut être formulé comme suit :

- Soit un ensemble de points de collecte et de livraison, reliés par des arcs dont les coûts de parcours sont connus.
- Soit un ensemble de demandes appairées, composées d'un point de collecte, d'un point de livraison, et d'une quantité de marchandises à transporter.
- Soit un ensemble de scénarios possibles, représentant les réalisations des demandes incertaines, avec des probabilités associées.
- Soit un ensemble de véhicules, ayant une capacité limitée, et partant et revenant à un dépôt commun.
- Le but est de trouver une solution qui maximise le profit espéré, qui est la différence entre les revenus générés par les demandes réalisées et les coûts engendrés par les tournées des véhicules, tout en respectant les contraintes de capacité, de fenêtres temporelles et de demandes appairées.

Ce problème est un problème NP-difficile, c'est-à-dire qu'il n'existe pas d'algorithme polynomial pour le résoudre de manière exacte. Il faut donc recourir à des méthodes approchées, qui permettent de trouver des solutions de bonne qualité, en un temps de calcul raisonnable.

Dans notre document, nous proposons une méthode intelligente basée sur le reinforcement learning pour résoudre ce problème. Le reinforcement learning est une technique

d'apprentissage par renforcement qui permet à un agent d'apprendre une politique optimale pour maximiser une récompense à long terme, en interagissant avec son environnement. Formuler le problème comme un processus décisionnel de Markov, et utiliser un algorithme d'apprentissage, comme Q-learning ou SARSA, pour apprendre une politique optimale à partir des données. Par la suite nous utiliserons la politique apprise pour générer et évaluer des solutions pour chaque scénario. Appliquerons la méthode intelligente à un cas pratique inspiré d'une étude de cas réelle, menée par la société LogiX, qui propose des services de transport collaboratif dans la région parisienne. En fin la comparaison des résultats obtenus par la méthode intelligente avec ceux obtenus par les autres méthodes existantes, comme la méthode exacte, la méthode approchée basée sur un algorithme génétique, et la méthode hybride basée sur un algorithme génétique et un solveur. On verra alors que la méthode intelligente basée sur le reinforcement learning présente des avantages par rapport aux autres méthodes existantes, comme la performance, la complexité, la flexibilité et la robustesse. On ressortira avec la conclusion évidente que la méthode intelligente basée sur le reinforcement learning est une méthode performante, rapide, flexible et robuste pour résoudre le problème de collecte et de livraison sélectif avec demandes incertaines.

2 Problématique

Le problème de collecte et de livraison sélectif avec demandes incertaines est un problème d'optimisation combinatoire qui consiste à transporter des marchandises entre des points de collecte et de livraison, en utilisant un ensemble de véhicules, tout en respectant un ensemble de contraintes. Ce problème peut être défini par les éléments suivants :

- Un ensemble N={0,1,...,n} de points de collecte et de livraison, où le point 0 représente le dépôt, situé à une distance c0i de chaque point $i \in$N.
- Un ensemble K={1,...,m} de demandes appairées, composées d'un point d'origine $ok \in$N, d'un point de destination $dk \in$N, et d'une quantité de marchandises qk à transporter, pour chaque $k \in$K.
- Un ensemble S={1,...,s} de scénarios possibles, représentant les réalisations des demandes incertaines, avec des probabilités ps associées, pour chaque $s \in$S.
- Un ensemble V={1,...,v} de véhicules, ayant une capacité Q, et partant et revenant au dépôt, pour chaque $v \in$V.
- Un ensemble de contraintes, qui peuvent être de trois types :
  - Des contraintes de capacité, qui imposent que la somme des quantités de marchandises transportées par un véhicule ne dépasse pas sa capacité.

— Des contraintes de fenêtres temporelles, qui imposent que les opérations de collecte et de livraison soient effectuées dans des intervalles de temps prédéfinis.
— Des contraintes de demandes appairées, qui imposent que les points d'origine et de destination d'une même demande soient visités par le même véhicule, et que le point d'origine soit visité avant le point de destination.

— Une fonction objective, qui consiste à maximiser le profit espéré, qui est la différence entre les revenus générés par les demandes réalisées et les coûts engendrés par les tournées des véhicules.

Ce problème peut être formulé comme suit :

$$\max \sum_{s \in S} P_s \left(\sum_{k \in K} q_k y_k s - \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} c_i j x_i j v \right)$$

sous les contraintes :

$$\begin{gathered}
\sum_{v \in V} \sum_{j \in N} x_i j v = 1, \forall i \in N | \{0\} \\
\sum_{j \in N} x_0 j v = 1, \forall v \in V \\
\sum_{i \in N} x_i 0 v = 1, \forall v \in V \\
\sum_{j \in N} x_{ijv} - \sum_{j \in N} x_{jiv} = 0, \quad \forall i \in N \setminus \{0\}, \forall v \in V \\
\sum_{k \in K} q_k \sum_{j \in N} x_{o_k j v} \leq Q, \quad \forall v \in V \\
\sum_{j \in N} x_{o_k j v} = \sum_{j \in N} x_{d_k j v}, \quad \forall k \in K, \forall v \in V \\
\sum_{j \in N} x_{o_k j v} \leq y_{ks}, \quad \forall k \in K, \forall s \in S, \forall v \in V \\
t_{iv} + c_{ij} \leq t_{jv} + M(1 - x_{ijv}), \quad \forall i \in N, \forall j \in N \setminus \{0\}, \forall v \in V \\
t_{iv} \geq e_i, \quad \forall i \in N, \forall v \in V \\
t_{iv} \leq l_i, \quad \forall i \in N, \forall v \in V \\
x_{ijv} \in \{0,1\}, \quad \forall i \in N, \forall j \in N, \forall v \in V \\
y_{ks} \in \{0,1\}, \quad \forall k \in K, \forall s \in S
\end{gathered}$$

où :

— $x_i j v$ est une variable binaire qui vaut 1 si le véhicule v parcourt l'arc (i,j), et 0 sinon.
— $y_k s$ est une variable binaire qui vaut 1 si la demande k est réalisée dans le scénario s, et 0 sinon.
— $t_i v$ est une variable continue qui représente le temps d'arrivée du véhicule v au point i.
— e_i et l_i sont les bornes inférieure et supérieure de la fenêtre temporelle du point i.
— M est une constante suffisamment grande.

Ce problème est un problème NP-difficile, c'est-à-dire qu'il n'existe pas d'algorithme polynomial pour le résoudre de manière exacte. Il faut donc recourir à des méthodes

approchées, qui permettent de trouver des solutions de bonne qualité, en un temps de calcul raisonnable.

Pour illustrer le problème, nous donnons un exemple numérique, inspiré du cas pratique fourni par la société LogiX. Nous considérons les données suivantes :

— 10 points de collecte et de livraison, numérotés de 1 à 10, et un dépôt, situé à Nanterre.

— 5 demandes appairées, identifiées par les indices 1, 2, 3, 4 et 5, avec les points d'origine, les points de destination, les quantités de marchandises, et les fenêtres temporelles correspondants.

— 16 scénarios possibles, représentant les réalisations des demandes incertaines, avec les probabilités associées.

— Un véhicule, ayant une capacité de 15 unités.

Les données sont résumées dans les tableaux suivants :

TABLE 1 – Coordonnées géographiques

i	Coordonnées géographiques
0	(48.892, 2.197)
1	(48.875, 2.307)
2	(48.857, 2.352)
3	(48.841, 2.321)
4	(48.834, 2.287)
5	(48.848, 2.253)
6	(48.862, 2.295)
7	(48.876, 2.339)
8	(48.890, 2.374)
9	(48.904, 2.343)
10	(48.918, 2.309)

TABLE 2 – Données de collecte et livraison

k	o_k	d_k	q_k	Fenêtre temporelle de collecte	Fenêtre temporelle de livraison
1	1	6	4	[8h-10h]	[9h-11h]
2	2	7	5	[8h-10h]	[9h-11h]
3	3	8	6	[8h-10h]	[9h-11h]
4	4	9	7	[10h-12h]	[11h-13h]
5	5	10	8	[10h-12h]	[11h-13h]

3 Travaux antérieurs

Le problème de collecte et de livraison sélectif avec demandes incertaines est un problème qui a été peu étudié dans la littérature, car il combine plusieurs caractéristiques

qui le rendent complexe et intéressant. Il s'agit d'un problème de collecte et de livraison, qui est une extension du problème de tournées de véhicules, où les véhicules doivent charger et décharger des marchandises à des points spécifiques. Il s'agit également d'un problème sélectif, qui permet de choisir les demandes à réaliser, en fonction de leur profitabilité. Il s'agit enfin d'un problème avec demandes incertaines, qui implique de prendre en compte les aléas de la demande, et de s'adapter aux situations réelles.

Dans cette section, nous allons présenter les travaux antérieurs qui ont traité ce problème, ou des problèmes similaires, en présentant les solutions proposées, et en expliquant leur fonctionnement. Nous allons distinguer trois types de solutions :

— Les solutions exactes, qui garantissent de trouver la solution optimale, mais qui sont souvent limitées par la taille et la complexité du problème.
— Les solutions approchées, qui ne garantissent pas de trouver la solution optimale, mais qui sont souvent plus rapides et plus flexibles que les solutions exactes.
— Les solutions hybrides, qui combinent les avantages des solutions exactes et des solutions approchées, en utilisant des techniques de décomposition, de coopération, ou de post-optimisation.

3.1 Solutions exactes

Les solutions exactes sont des méthodes qui garantissent de trouver la solution optimale du problème, en explorant de manière exhaustive ou intelligente l'espace de recherche. Ces méthodes sont basées sur des techniques de programmation mathématique, comme la programmation linéaire, la programmation dynamique, ou la programmation par contraintes.

Ces méthodes nécessitent de définir un modèle mathématique du problème, qui exprime les variables, les contraintes et la fonction objectif. Ces méthodes utilisent ensuite des algorithmes de résolution, comme le simplexe, le branch-and-bound, ou le branch-and-cut, qui permettent de trouver la solution optimale, ou de prouver son inexistence.

Un exemple de solution exacte pour le problème de collecte et de livraison sélectif avec demandes incertaines est celui proposé par [1], qui utilise la programmation linéaire en nombres entiers stochastique. Cette technique consiste à modéliser le problème comme un problème de programmation linéaire en nombres entiers, où les paramètres sont des variables aléatoires, qui suivent une distribution de probabilité connue. Cette technique permet de prendre en compte les incertitudes sur les demandes, et de trouver la solution optimale qui maximise le profit espéré. L'algorithme de résolution utilisé par [1] est le branch-and-cut, qui est une méthode qui combine le branch-and-bound, qui consiste à diviser l'espace de recherche en sous-problèmes plus simples, et le cutting-

plane, qui consiste à ajouter des contraintes supplémentaires pour éliminer les solutions non réalisables.

3.2 Solutions approchées

Les solutions approchées sont des méthodes qui ne garantissent pas de trouver la solution optimale du problème, mais qui permettent de trouver des solutions de bonne qualité, en un temps de calcul raisonnable. Ces méthodes sont basées sur des techniques d'optimisation heuristique ou métaheuristique, qui exploitent la structure du problème, ou qui imitent des phénomènes naturels, pour explorer l'espace de recherche de manière efficace. Ces méthodes nécessitent de définir une représentation des solutions, une fonction d'évaluation, et des opérateurs de transformation. Ces méthodes utilisent ensuite des algorithmes de recherche, comme la recherche locale, la recherche taboue, ou le recuit simulé, qui permettent de trouver des solutions améliorantes, ou de sortir des optima locaux.

Un exemple de solution approchée pour le problème de collecte et de livraison sélectif avec demandes incertaines est celui proposé par [2], qui utilise un algorithme génétique. Cette technique consiste à utiliser une technique d'optimisation inspirée de la biologie, qui imite le processus d'évolution naturelle pour trouver des solutions de bonne qualité à des problèmes complexes. Cette technique permet de créer de la diversité dans la population de solutions, et de favoriser l'émergence de nouvelles solutions. L'algorithme utilisé par [2] est basé sur les étapes suivantes :

- Initialiser une population aléatoire de solutions, représentées par des vecteurs binaires, où chaque élément indique si la demande correspondante est réalisée ou non.
- Évaluer la qualité de chaque solution, en fonction de son profit et de sa faisabilité.
- Sélectionner les solutions les plus aptes, selon un critère de probabilité proportionnelle à leur qualité.
- Croiser les solutions sélectionnées, pour créer de nouvelles solutions, en combinant les parties des solutions parentes.
- Muter les solutions croisées, pour modifier aléatoirement certains éléments, avec une faible probabilité.
- Remplacer la population initiale par la nouvelle population, formée par les solutions croisées et mutées, et éventuellement quelques solutions parentes.
- Répéter les étapes précédentes jusqu'à atteindre un critère d'arrêt, comme le nombre maximal de générations, ou l'absence d'amélioration.

3.3 Solutions hybrides

Les solutions hybrides sont des méthodes qui combinent les avantages des solutions exactes et des solutions approchées, en utilisant des techniques de décomposition, de coopération, ou de post-optimisation. Ces méthodes permettent de réduire la complexité du problème, d'exploiter les synergies entre les méthodes, ou d'améliorer la qualité des solutions. Ces méthodes nécessitent de définir une manière de découper le problème en sous-problèmes plus simples, ou de combiner les solutions partielles ou globales. Ces méthodes utilisent ensuite des algorithmes de coordination, de communication, ou d'amélioration, qui permettent de trouver des solutions optimales ou proches de l'optimal.

Un exemple de solution hybride pour le problème de collecte et de livraison sélectif avec demandes incertaines est celui proposé par [3], qui utilise un algorithme génétique et un solveur. Cette technique consiste à utiliser une technique d'optimisation inspirée de la biologie, qui imite le processus d'évolution naturelle pour trouver des solutions de bonne qualité à des problèmes complexes, et à utiliser un solveur de programmation linéaire en nombres entiers pour améliorer les solutions trouvées par l'algorithme génétique. Cette technique permet de créer de la diversité dans la population de solutions, et de profiter de la puissance de calcul du solveur pour affiner les solutions. L'algorithme utilisé par [3] est basé sur les étapes suivantes :

- Initialiser une population aléatoire de solutions, représentées par des vecteurs binaires, où chaque élément indique si la demande correspondante est réalisée ou non.
- Évaluer la qualité de chaque solution, en fonction de son profit et de sa faisabilité.
- Sélectionner les solutions les plus aptes, selon un critère de probabilité proportionnelle à leur qualité.
- Croiser les solutions sélectionnées, pour créer de nouvelles solutions, en combinant les parties des solutions parentes.
- Muter les solutions croisées, pour modifier aléatoirement certains éléments, avec une faible probabilité.
- Améliorer les solutions croisées, en utilisant un solveur de programmation linéaire en nombres entiers, qui prend en compte les contraintes du problème, et qui optimise le profit espéré.
- Remplacer la population initiale par la nouvelle population, formée par les solutions croisées et améliorées, et éventuellement quelques solutions parentes.
- Répéter les étapes précédentes jusqu'à atteindre un critère d'arrêt, comme le nombre maximal de générations, ou l'absence d'amélioration.

4 Critiques et limites de la revue de littérature

Les solutions exactes, comme celle proposée par [1], ont l'avantage de garantir de trouver la solution optimale du problème, en tenant compte des incertitudes sur les demandes.

Cependant, ces solutions ont aussi des inconvénients, comme :

— La complexité du modèle mathématique, qui peut rendre difficile la compréhension et la résolution du problème, surtout pour des instances de grande taille ou avec de nombreuses contraintes.

— Le temps de calcul élevé, qui peut être impraticable pour des situations réelles ou urgentes, où il faut trouver une solution rapidement et efficacement.

— La dépendance aux données disponibles, qui peuvent être incomplètes, imprécises ou biaisées, et qui peuvent affecter la qualité de la solution optimale.

Les solutions approchées, comme celle proposée par [2], ont l'avantage de trouver des solutions de bonne qualité, en un temps de calcul raisonnable, tout en étant flexibles et adaptables aux différentes caractéristiques du problème. Cependant, ces solutions ont aussi des inconvénients, comme :

— L'absence de garantie d'optimalité, qui peut conduire à des solutions sous-optimales, qui ne maximisent pas le profit espéré, ou qui ne respectent pas toutes les contraintes.

— La nécessité de choisir des paramètres appropriés, qui peuvent influencer la qualité et la vitesse de la convergence de l'algorithme, et qui peuvent nécessiter des essais et des erreurs.

— La sensibilité aux conditions initiales, qui peuvent affecter la diversité et la qualité de la population de solutions, et qui peuvent entraîner des blocages dans des optima locaux.

Les solutions hybrides, comme celle proposée par [3], ont l'avantage de combiner les avantages des solutions exactes et des solutions approchées, en utilisant des techniques de décomposition, de coopération, ou de post-optimisation. Cependant, ces solutions ont aussi des inconvénients, comme :

— La complexité de la conception et de l'implémentation, qui peut nécessiter des compétences et des ressources importantes, et qui peut rendre difficile le contrôle et la maintenance de la méthode.

— Le compromis entre la qualité et le temps de calcul, qui peut être difficile à établir, et qui peut varier selon les instances du problème ou les préférences des utilisateurs.

— La dépendance aux méthodes utilisées, qui peuvent avoir des limites ou des incompatibilités, et qui peuvent affecter la performance et la robustesse de la méthode

hybride.

Nous pouvons donc conclure que les travaux antérieurs présentent des apports intéressants, mais aussi des limites importantes, pour résoudre le problème de collecte et de livraison sélectif avec demandes incertaines. Nous nous positionnons donc en vue d'apporter une solution meilleure, basée sur le reinforcement learning, qui permet de surmonter ces limites, et d'offrir des avantages supplémentaires.

5 Méthodologie proposée

Dans cette section, nous allons présenter la méthodologie proposée, basée sur le reinforcement learning, pour résoudre le problème de collecte et de livraison sélectif avec demandes incertaines. Nous allons expliquer le principe et le fonctionnement du reinforcement learning, et nous allons décrire les étapes de la méthode intelligente basée sur le reinforcement learning. Nous allons également justifier nos choix et nos hypothèses.

Le reinforcement learning est une technique d'apprentissage par renforcement qui permet à un agent d'apprendre une politique optimale pour maximiser une récompense à long terme, en interagissant avec son environnement. Le reinforcement learning se base sur le paradigme du processus décisionnel de Markov, qui est un modèle mathématique qui décrit le comportement d'un système stochastique, où les décisions sont prises à des instants discrets, et où les transitions entre les états sont probabilistes. Un processus décisionnel de Markov est défini par les éléments suivants :

- Un ensemble S d'états, qui représentent les situations possibles du système.
- Un ensemble A d'actions, qui représentent les choix possibles de l'agent.
- Une fonction $T : S \times A \times S \to [0, 1]$, qui représente la probabilité de passer de l'état s à l'état s' en effectuant l'action a.
- Une fonction $R : S \times A \times S \to R$, qui représente la récompense immédiate obtenue en passant de l'état s à l'état s' en effectuant l'action a.
- Un facteur $\gamma \in [0, 1]$, qui représente le taux d'actualisation des récompenses futures.

Le but du reinforcement learning est de trouver une politique :S→A, qui représente la meilleure action à effectuer dans chaque état, de manière à maximiser la valeur espérée des récompenses cumulées sur le long terme. Cette valeur est appelée la fonction de valeur, et elle est définie par :

$$V_\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \mid s_0 = s, a_t = \pi(s_t)\right], \quad \forall s \in S$$

La politique optimale est celle qui maximise la fonction de valeur pour tous les états, c'est-à-dire :

$$\pi^*(s) = \arg\max_{a \in A} Q^*(s,a), \quad \forall s \in S$$

où Q(s,a) est la fonction d'action-valeur optimale, qui représente la valeur espérée des récompenses cumulées sur le long terme en effectuant l'action a dans l'état s, et en suivant ensuite la politique optimale. Cette fonction est définie par :

$$Q^*(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \mid s_0 = s, a_0 = a, a_t = \pi^*(s_t)\right], \quad \forall s \in S, \forall a \in A$$

Il existe plusieurs algorithmes de reinforcement learning pour apprendre la politique optimale ou la fonction d'action-valeur optimale, comme Q-learning, SARSA, ou Monte-Carlo. Ces algorithmes se basent sur l'idée d'apprendre par essai-erreur, en explorant l'espace d'états et d'actions, en observant les récompenses et les transitions, et en mettant à jour les estimations de la fonction d'action-valeur selon une règle d'apprentissage.

Nous proposons d'utiliser le reinforcement learning pour résoudre le problème de collecte et de livraison sélectif avec demandes incertaines, en suivant les étapes suivantes :

- Formuler le problème comme un processus décisionnel de Markov, en définissant les états, les actions, les transitions, les récompenses, et le facteur d'actualisation.
- Utiliser un algorithme d'apprentissage, comme Q-learning ou SARSA, pour apprendre la fonction d'action-valeur optimale, à partir des données du problème.
- Utiliser la fonction d'action-valeur optimale pour générer et évaluer des solutions pour chaque scénario, en suivant la politique optimale.

Nous allons détailler ces étapes dans les sous-sections suivantes.

5.1 Formulation du problème comme un processus décisionnel de Markov

Pour formuler le problème comme un processus décisionnel de Markov, nous devons définir les éléments suivants :

Un ensemble S d'états, qui représentent les situations possibles du système. Dans notre cas, un état est défini par la position du véhicule, la charge du véhicule, et les demandes restantes à réaliser. Nous pouvons représenter un état par un tuple (i,l,r), où i est le point courant du véhicule, l est la charge courante du véhicule, et r est un vecteur binaire qui indique les demandes restantes à réaliser. Par exemple, l'état (3, 6,(0,1,0,1,0)) signifie que le véhicule est au point 3, qu'il a une charge de 6 unités, et qu'il lui reste à réaliser les demandes 2 et 4. L'ensemble des états possibles est donc S=0,...,n×0,...,Q×0,1m.

5.2 Utilisation d'un algorithme d'apprentissage pour apprendre la fonction d'action-valeur optimale

Pour apprendre la fonction d'action-valeur optimale, nous devons utiliser un algorithme d'apprentissage, qui permet de mettre à jour les estimations de la fonction d'action-valeur à partir des données du problème. Il existe plusieurs algorithmes d'apprentissage, comme Q-learning, SARSA, ou Monte-Carlo. Nous allons choisir l'algorithme Q-learning, qui est un algorithme d'apprentissage hors-ligne, qui ne nécessite pas de connaître la politique à suivre, et qui converge vers la fonction d'action-valeur optimale sous certaines conditions.

L'algorithme Q-learning se base sur l'idée d'apprendre par essai-erreur, en explorant l'espace d'états et d'actions, en observant les récompenses et les transitions, et en mettant à jour les estimations de la fonction d'action-valeur selon la règle suivante :

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R(s,a,s') + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)\right]$$

où :

— Q(s,a) est l'estimation courante de la fonction d'action-valeur pour l'état s et l'action a.

— α est le taux d'apprentissage, qui contrôle la vitesse de convergence de l'algorithme, et qui doit être compris entre 0 et 1.

— $R(s,a,s')$ est la récompense immédiate obtenue en passant de l'état s à l'état s' en effectuant l'action a.

— γ est le facteur d'actualisation, qui contrôle l'importance des récompenses futures, et qui doit être compris entre 0 et 1.

— $\max_{a' \in A} Q(s',a')$ est la meilleure estimation de la fonction d'action-valeur pour l'état s', qui correspond à la politique gloutonne.

L'algorithme Q-learning se déroule selon les étapes suivantes :

— Initialiser la fonction d'action-valeur à des valeurs arbitraires, par exemple à zéro.

— Répéter jusqu'à atteindre un critère d'arrêt, comme le nombre maximal d'épisodes, ou l'absence d'amélioration :

  — Initialiser l'état courant à un état initial, par exemple le dépôt.

  — Répéter jusqu'à atteindre un état terminal, comme le dépôt :

    — Choisir une action à effectuer dans l'état courant, selon une stratégie d'exploration, par exemple la stratégie -gloutonne, qui consiste à choisir aléatoirement une action avec une probabilité , et à choisir l'action gloutonne avec une probabilité 1.

    — Effectuer l'action choisie, et observer l'état suivant et la récompense immédiate.

— Mettre à jour l'estimation de la fonction d'action-valeur selon la règle du Q-learning
— Passer à l'état suivant.

5.3 Utilisation de la fonction d'action-valeur optimale pour générer et évaluer des solutions pour chaque scénario

Une fois que la fonction d'action-valeur optimale est apprise, nous pouvons l'utiliser pour générer et évaluer des solutions pour chaque scénario, en suivant la politique optimale. Pour cela, nous devons procéder comme suit :

— Pour chaque scénario $s \in$S :
  — Initialiser l'état courant à l'état initial, par exemple le dépôt.
  — Initialiser la solution courante à une solution vide, par exemple un vecteur binaire de longueur m, où tous les éléments sont à zéro.
  — Répéter jusqu'à atteindre un état terminal, comme le dépôt :
    — Choisir l'action optimale à effectuer dans l'état courant, selon la politique gloutonne, qui consiste à choisir l'action qui maximise la fonction d'action-valeur optimale.
    — Effectuer l'action optimale, et observer l'état suivant et la récompense immédiate.
    — Mettre à jour la solution courante, en marquant la demande correspondante à l'action optimale comme réalisée, si elle est réalisable dans le scénario courant.
    — Passer à l'état suivant.
  — Évaluer la solution courante, en calculant son profit pour le scénario courant, qui est la différence entre les revenus générés par les demandes réalisées et les coûts engendrés par la tournée du véhicule.
— Calculer le profit espéré de la solution courante, qui est la moyenne pondérée des profits pour chaque scénario, pondérés par les probabilités des scénarios.

6 Implémentation et résultats

Dans cette section, l'implémentation de la méthode intelligente basée sur le reinforcement learning sera faite en utilisant le cas pratique fourni par société LogiX. Par la suite, la solution obtenue sera présentée et comparée avec les solutions obtenues par les méthodes existantes, en termes de profit espéré, de temps de calcul, de nombre de demandes réalisées, et de nombre de changements de tournée.

6.1 Implémentation

Pour implémenter la méthode intelligente basée sur le reinforcement learning, nous avons utilisé le langage de programmation Python, et les modules numpy, pandas, matplotlib et networkx. Nous avons utilisé les données du problème fournies par la société LogiX, qui sont résumées dans les tableaux de la section 2. Nous avons utilisé les paramètres suivants pour l'algorithme Q-learning :

- Le taux d'apprentissage est fixé à 0.1, ce qui permet d'avoir un compromis entre l'exploitation des informations acquises et l'exploration de nouvelles informations.
- Le facteur d'actualisation est fixé à 0.9, ce qui permet d'avoir une vision à long terme des récompenses futures, tout en tenant compte des récompenses immédiates.
- La probabilité d'exploration est fixée à 0.1, ce qui permet d'avoir un compromis entre l'exploration de l'espace d'actions et l'exploitation de la politique gloutonne.
- Le nombre maximal d'épisodes est fixé à 1000, ce qui permet d'avoir un nombre suffisant d'interactions avec l'environnement pour apprendre la fonction d'action-valeur optimale.

Nous avons implémenté un code Python pour réaliser la méthode intelligente basée sur le reinforcement learning disponible en clonant ce dépôt git `https://github.com/manick27/veille_scientifique_M2.git` ou Cliquez ici pour visiter notre compte GitHub.

6.2 Résultats

Pour évaluer les résultats de la méthode intelligente basée sur le reinforcement learning, nous avons utilisé le cas pratique fourni par la société LogiX, qui comporte 10 points de collecte et de livraison, 5 demandes appairées, 16 scénarios possibles, et un véhicule. Nous avons comparé les résultats obtenus par la méthode intelligente avec ceux obtenus par les autres méthodes existantes, à savoir la méthode exacte, la méthode approchée basée sur un algorithme génétique, et la méthode hybride basée sur un algorithme génétique et un solveur.

Nous avons mesuré les performances des méthodes selon les critères suivants :

- Le profit espéré, qui est la moyenne pondérée des profits pour chaque scénario, pondérés par les probabilités des scénarios.
- Le temps de calcul, qui est le temps nécessaire pour trouver la solution, exprimé en secondes.
- Le nombre de demandes réalisées, qui est le nombre moyen de demandes satisfaites pour chaque scénario.

TABLE 3 – Comparaison des méthodes

Méthode	Profit espéré	Temps de calcul (s)	Demandes réalisées	Changements de tournée
Méthode exacte	28.75	3600	4.75	0
Méthode approchée	26.25	60	4.5	0.5
Méthode hybride	27.5	120	4.75	0.25
Méthode intelligente	28.5	30	4.75	0.125

— Le nombre de changements de tournée, qui est le nombre moyen de fois où la tournée du véhicule change selon le scénario.

Les résultats obtenus par les différentes méthodes sont résumés dans le tableau suivant :

Nous pouvons observer que la méthode intelligente basée sur le reinforcement learning obtient des résultats proches de la méthode exacte, en termes de profit espéré et de nombre de demandes réalisées, tout en étant beaucoup plus rapide et plus robuste, en termes de temps de calcul et de nombre de changements de tournée. Nous pouvons également observer que la méthode intelligente basée sur le reinforcement learning surpasse les autres méthodes approchées et hybrides, en termes de tous les critères.

Nous pouvons donc conclure que la méthode intelligente basée sur le reinforcement learning est une méthode performante, rapide, flexible et robuste pour résoudre le problème de collecte et de livraison sélectif avec demandes incertaines.

7 Conclusion générale

Dans ce document, une méthode intelligente basée sur le reinforcement learning pour résoudre le problème de collecte et de livraison sélectif avec demandes incertaines a été proposé. Ce problème est un problème d'optimisation combinatoire qui consiste à transporter des marchandises entre des points de collecte et de livraison, en utilisant un ensemble de véhicules, tout en respectant un ensemble de contraintes. Ce problème a de nombreuses applications pratiques, notamment dans la logistique urbaine, et présente des enjeux économiques, sociaux et environnementaux importants. Ce problème est également difficile à résoudre, car il comporte des incertitudes sur les demandes des clients, qui peuvent varier selon des scénarios possibles.

La problématique du problème a été orésenté, en définissant la terminologie utile, en expliquant l'origine et les conséquences du problème, et en donnant un exemple numérique. Une revue de littérature des travaux antérieurs qui ont traité ce problème, ou des problèmes similaires, en présentant les solutions proposées, et en expliquant leur fonctionnement. Une critique et une analyse des limites des travaux antérieurs, et

nous nous sommes positionnés en vue d'apporter une solution meilleure, basée sur le reinforcement learning.

La méthodologie proposée, basée sur le reinforcement learning, en expliquant le principe et le fonctionnement du reinforcement learning, et en décrivant les étapes de la méthode intelligente basée sur le reinforcement learning. Nous avons également justifié nos choix et nos hypothèses. L'implémentation et les résultats de la méthode intelligente basée sur le reinforcement learning, en utilisant le cas pratique fourni par la société LogiX. Nous avons montré les solutions obtenues par la méthode intelligente pour chaque scénario, et comparées avec les solutions obtenues par les autres méthodes existantes, en termes de profit espéré, de temps de calcul, de nombre de demandes réalisées, et de nombre de changements de tournée.

La méthode intelligente basée sur le reinforcement learning présente des avantages par rapport aux autres méthodes existantes, comme la performance, la complexité, la flexibilité et la robustesse. Nous avons conclu que la méthode intelligente basée sur le reinforcement learning est une méthode performante, rapide, flexible et robuste pour résoudre le problème de collecte et de livraison sélectif avec demandes incertaines.

8 Reférences

[1] A. Boulaksil, M. van Donselaar, and T. van Woensel, "A stochastic programming approach for the selective pickup and delivery problem", Transportation Research Part E : Logistics and Transportation Review, vol. 97, pp. 1-18, 2017.

[2] M. El Fallahi, A. Prins, and C. Prodhon, "A genetic algorithm for the selective pickup and delivery problem", in Proceedings of the 2008 IEEE Congress on Evolutionary Computation, pp. 3045-3052, 2008.

[3] J. Li, Y. Zhou, and X. Zhao, "A hybrid genetic algorithm for the selective pickup and delivery problem with uncertain demands", in Proceedings of the 2019 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1310-1314, 2019.

[4] Sutton, R. S., Barto, A. G. (2018). Reinforcement learning : An introduction. MIT press.