Conception d'une matheuristique pour la résolution du Pickup and Delivery Problem with Time Windows

Mehdi LATIF

Stage réalisé au sein de l'Équipe SLP (LS2N)
Université de Nantes - Licence 3 Informatique parcours Mathématiques

mehdi.latif@etu.univ-nantes.fr

28 juillet 2019

Plan de la présentation

L'entreprise DELIVREMOITOUT

L'entreprise DELIVREMOITOUT est spécialisée dans la livraison de repas gastronomiques préparés par des restaurateurs renommés, livrés directement chez des particuliers par une flotte de cyclistes.

Grâce à son site internet, les clients renseignent leurs commandes en spécifiant :

- Le nom du restaurant dans lequel ils souhaitent commander.
- La quantité de plats qu'ils souhaitent recevoir.
- Une fenêtre horaire durant laquelle ils souhaitent être livrés.

L'entreprise DELIVREMOITOUT

La politique de gestion des livraisons spécifie que les commandes doivent être passées une demi-journée avant la livraison.

Une fois les commandes effectuées, l'entreprise DELIVREMOITOUT transmet les commandes aux restaurateurs qui vont quant à eux, définir une fenêtre de temps durant laquelle les cyclistes pourront venir retirer les repas.

L'entreprise DELIVREMOITOUT

Ce jour là, tous les clients ont effectués leurs commandes dans des restaurants distincts.

Quelques informations supplémentaires :

- On prévoit un temps de service qui indique la durée nécessaire pour effectuer les retraits des repas chez les restaurateurs et les livraisons chez les clients.
- La flotte de l'entreprise DELIVREMOITOUT est composée d'un nombre limité de cyclistes.
- Les cyclistes ont un sac à dos dont la capacité est limitée.
- Les cyclistes doivent venir récupérer leurs vélos au siège de l'entreprise et venir les redéposer après les livraisons.

L'entreprise DELIVREMOITOUT

Quelques informations supplémentaires :

- Une commande est composée d'un point de collecte et d'un point de livraison.
- Une commande doit être servie par le même cycliste.
- Un cycliste peut prendre en charge plusieurs commandes sur un même voyage.
- Un cycliste peut arriver en avance dans un restaurant ou chez un client mais il devra attendre le début de la fenêtre de temps pour récupérer ou livrer un repas.

L'entreprise DELIVREMOITOUT

Objectif de l'entreprise

Optimiser les itinéraires *i.e.* une suite de points de collecte et de livraison de repas en :

1. Minimisant la distance totale parcourue par les cyclistes
2. Minimisant le temps total passé par chaque cycliste

Sous les contraintes

- Les itinéraires proposés par l'entreprise doivent respecter les fenêtres de temps imposées par les clients et les restaurateurs.
- Un cycliste ne doit pas se retrouver avec plus de repas qu'il ne peut en transporter.

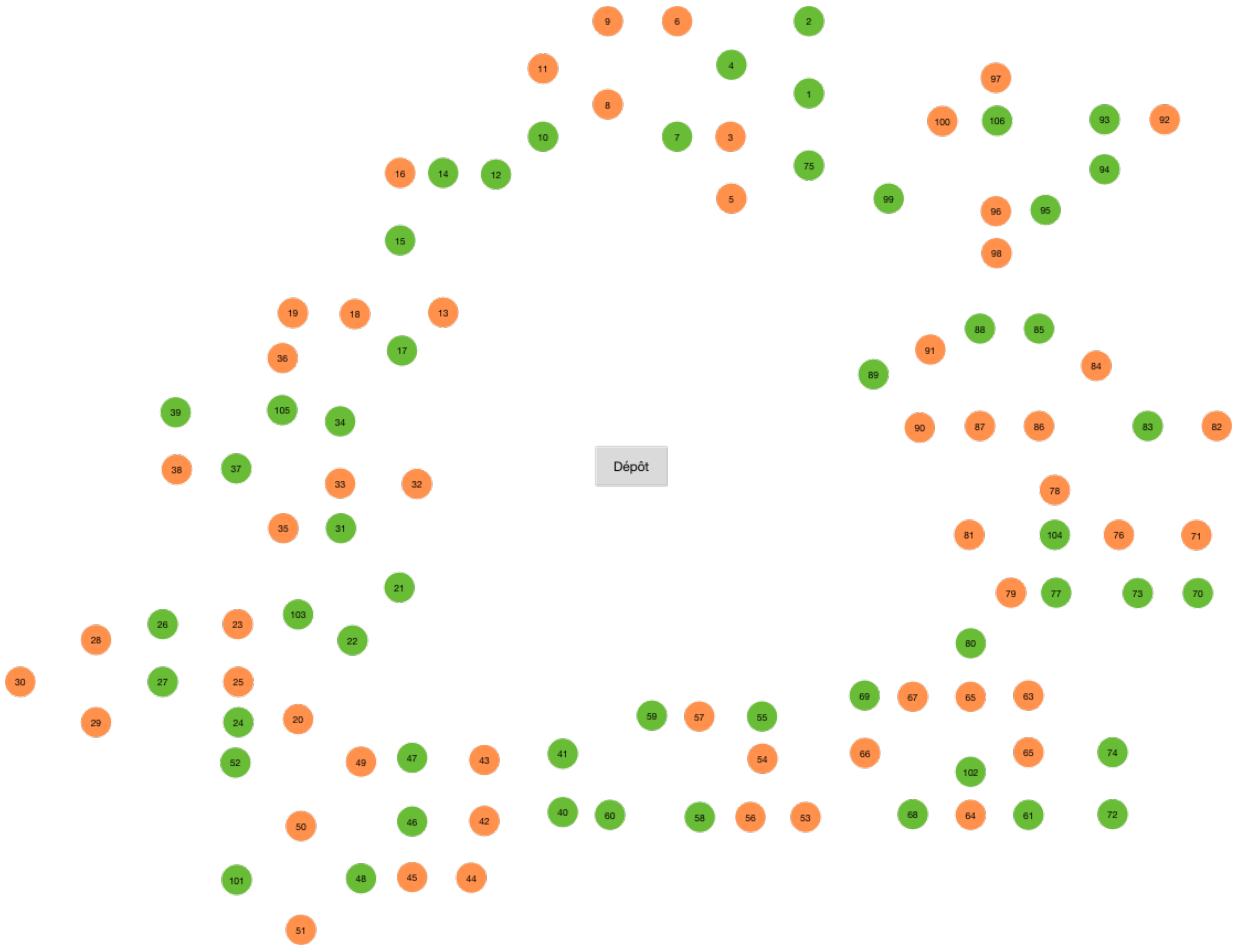
L'entreprise DELIVREMOITOUT

Ce soir-là, l'entreprise a reçu 53 commandes clients qui sont détaillées ci-dessous :

n° Commande	n° Restaurant	n° Client	Quantité à livrer	n° Commande	n° Restaurant	n° Client	Quantité à livrer
1	4	76	10	28	52	102	10
2	6	8	10	29	54	59	20
3	7	3	20	30	55	61	40
4	9	11	20	31	57	60	30
5	10	5	10	32	58	56	40
6	12	2	10	33	63	69	20
7	14	18	30	34	64	75	50
8	17	15	40	35	65	103	10
9	19	13	20	36	66	73	10
10	20	16	10	37	67	70	10
11	21	25	10	38	68	62	10
12	24	104	10	39	72	78	20
13	26	28	40	40	77	74	10
14	29	23	20	41	79	105	20
15	30	27	10	42	80	81	10
16	31	22	10	43	82	71	30
17	33	32	30	44	83	86	20
18	34	38	40	45	85	90	20
19	36	40	10	46	87	92	10
20	37	106	10	47	88	84	20
21	39	35	30	48	91	89	10
22	43	41	20	49	93	94	20
23	44	42	10	50	97	95	10
24	45	47	10	51	98	107	30
25	46	49	10	52	99	96	20
26	50	48	10	53	101	100	20
27	51	53	10				

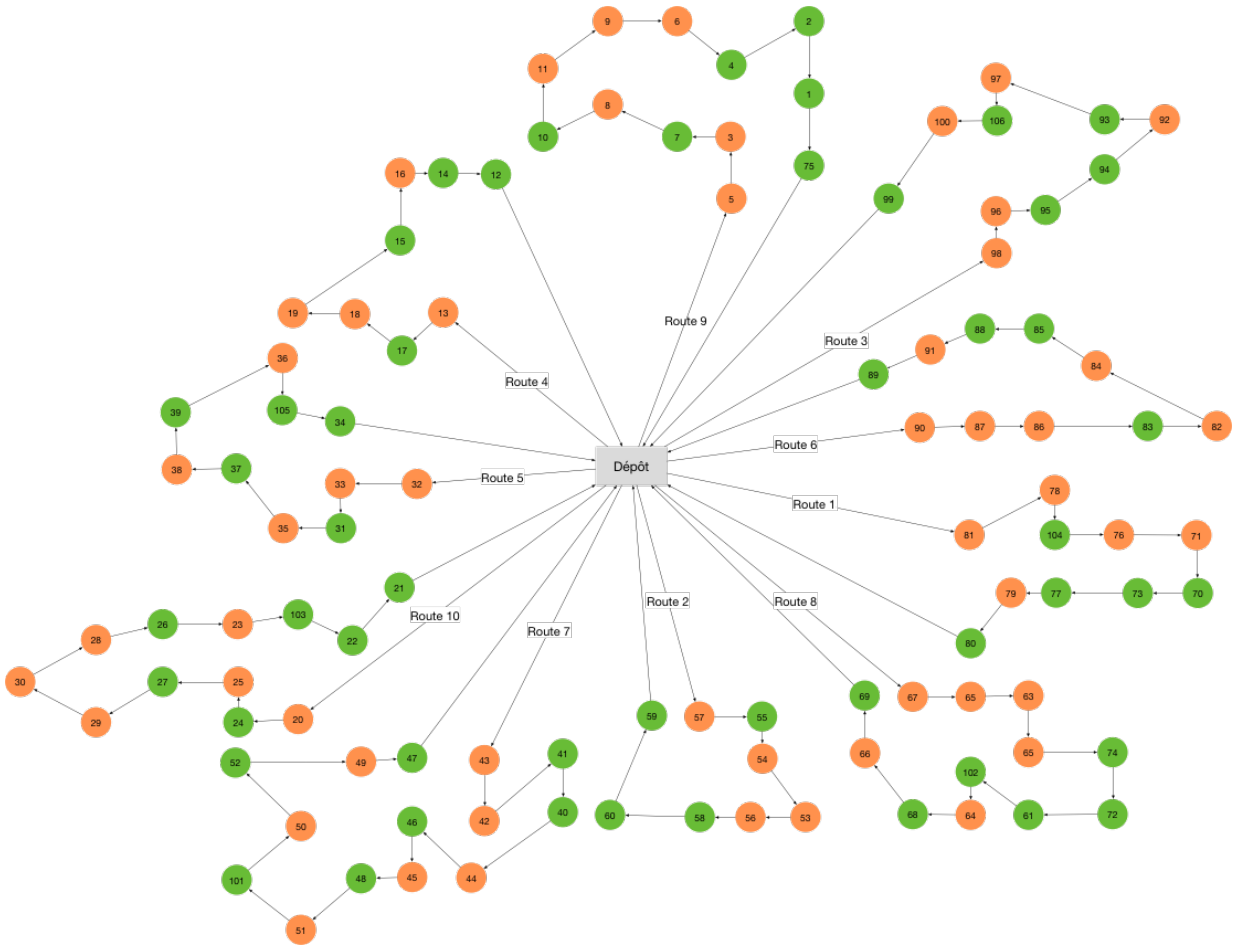
L'entreprise DELIVREMOITOUT

Analyse des coordonnées géographiques



L'entreprise DELIVREMOITOUT

Résultats de l'optimisation : 10 cyclistes nécessaires ayant parcouru une distance totale 82.894 km.



Présentation de la problématique - PDPTW

Le problème rencontré par l'entreprise DELIVREMOITOUT s'appelle en réalité

Pickup and **D**elivery **P**roblem with **T**ime **W**indows

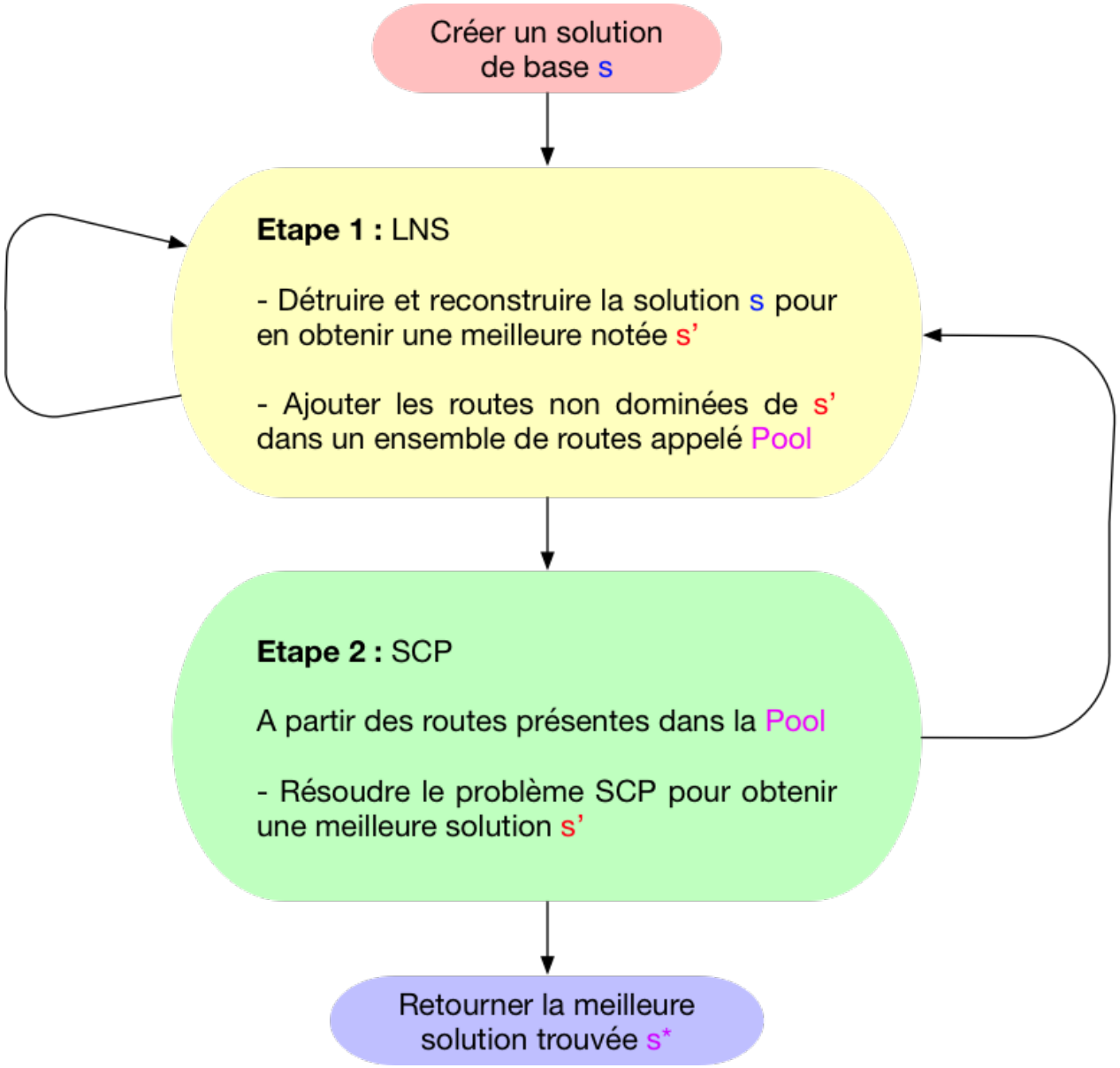
Le PDPTW est un problème $\mathcal{NP}$-Difficile.

⇒ Nécessite l'utilisation de **méthode de résolution approchée** pour des grandes instances du problème pour obtenir **un résultat en un temps raisonable**.

L'objectif de ce stage

Créer une *matheuristique* pour résoudre le problème du PDPTW

PDPTW - Présentation de la matheuristique



PDPTW - Travail réalisé - Test de faisabilité d'insertion

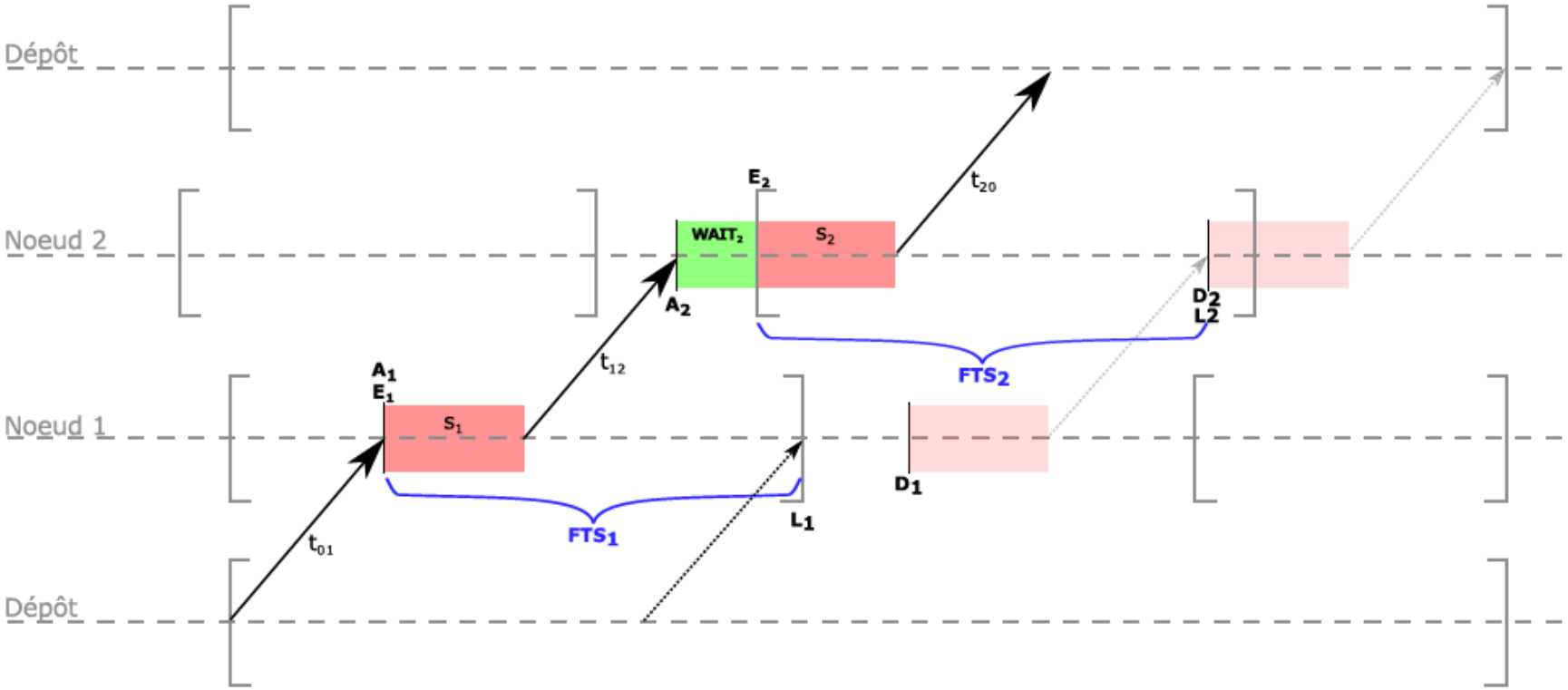
Objectif : Pour la métaheuristique de LNS, proposer un indicateur permettant de tester la faisabilité de l'insertion d'une requête dans une route et ceci en $O(1)$.

Notations :

On note :

- a_i la date d'ouverture de la fenêtre de temps du noeud i
- b_i la date de fermeture de la fenêtre de temps du noeud i
- $t_{i,j}$ le temps de parcours entre les noeuds i et j
- s_i le temps de service au noeud i

PDPTW - Travail réalisé - Test de faisabilité d'insertion



PDPTW - Travail réalisé - Test de faisabilité d'insertion

On définit les indicateurs temporels suivants :

- A_i la date d'arrivée et E_i la date de début de service au plus tôt au noeud i

$$A_i = \max\{A_{i-1}, a_{i-1}\} + s_{i-1} + t_{i-1,i}$$

$$E_i = \max\{A_i, a_i\}$$

- L_i la date d'arrivée au plus tard et D_i la date de début de service au plus tard au noeud i

$$D_i = \min\{D_{i+1}, b_{i+1}\} - s_i - t_{i,i+1}$$

$$L_i = \min\{D_i, b_i\}$$

- FTS_i Forward Time Slack *i.e.* la durée durant laquelle la visite d'un noeud i peut être repoussée sans occasionner de dépassement de fenêtre de temps et ceci, pour le reste de la tournée

$$FTS_i = L_i - E_i$$

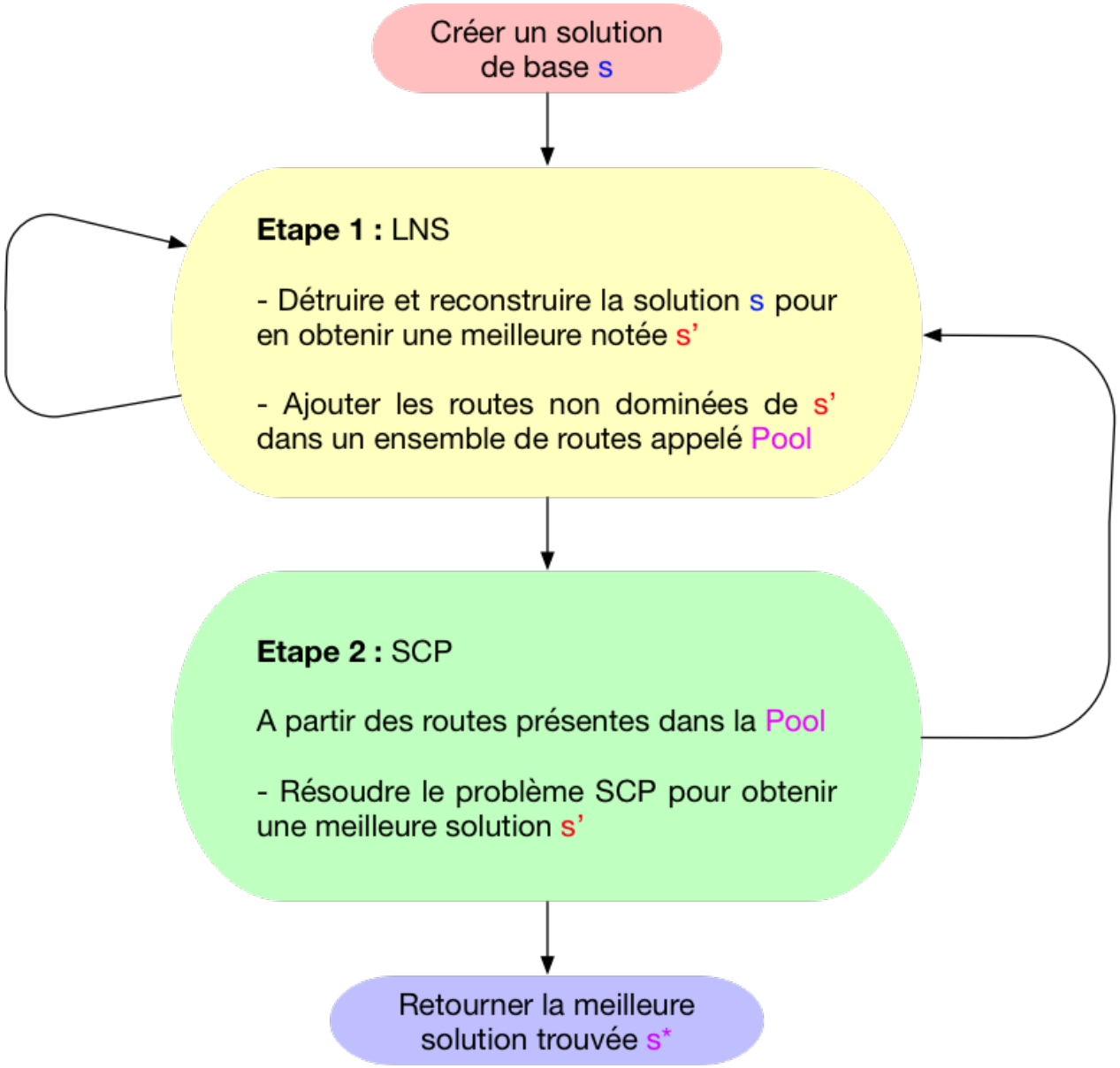
Algorithme 1 : Algorithme de vérification de la faisabilité de l'insertion d'une requête r à des i et j positions données

Entrées : ω, une route, r, une requête, i et j, des indices de position de noeuds dans S

Sorties : p, l'indice d'insertion du pickup, d, l'indice d'insertion du delivery, c le coût après insertion

```
1  p ← -1,d← -1, c← 0
2  Si q_r ≤ charge_ω(i+1, j) alors
3      h ← E_k,∀k ∈ {1,...,|ω|}      /* h est un vecteur contenant
         l'ordonancement au plus tôt du service pour toutes
         les tâches de ω */
4
5      x_1 ← max(h[i] + s_i + t_{i,p_r}, a_{p_r})
6      Si x_1 ≤ b_{p_r} alors
7          x_2 ← max(a_{i+1}, x_1 + s_{p_r} + t_{p_r,i+1})
8          δ ← x_2 − h[i+1]
9          Si δ ≤ FTS_{i+1} alors
10             x_3 ← max(h[j] + s_j + d_{j,d_r}, a_{d_r}) + δ
11             Si x_3 ≤ b_{d_r} + δ alors
12                 x_4 ← max(a_{j+1}, x_3 + s_{d_r} + t_{d_r,j+1})
13                 δ' ← x_4 − h[j+1]
14                 Si δ' ≤ FTS_{j+1} alors
                       /* On obtient ω' en ajoutant r dans ω en i
                          et j                                  */
15                     p ← i,d ← j, c ← |Δ(dist(ω'), dist(ω))|   /* Δ
                         est la variation de coût entre deux
                         routes */
16                 fin
17             fin
18         fin
19     fin
20 fin
21 retourner p,d,c
```

PDPTW - Retour sur la matheuristique



PDPTW - Matheuristique - Composante SCP

Une solution au PDPTW

Une **solution** au PDPTW comme un ensemble d'itinéraires (ou routes) *i.e.* une séquence de visites de noeuds, où chaque itinéraire est assigné à un véhicule et chaque noeud visité durant sa fenêtre fenêtre de temps.

Définition - Domination de routes

Soient ω, ω' deux routes issues de $\Omega = \{1, \ldots, |\Omega|\}$ et $\Pi(\omega)$ le coût d'une route ω. On dit que **la route ω domine la route ω'** si et seulement si :

1. ω et ω' servent les mêmes requêtes.
2. Le coût de ω est inférieur à ω' *i.e.* $\Pi(\omega) \leq \Pi(\omega')$

L'objectif du SCP

Trouver parmi les itinéraires présents dans la Pool, des routes couvrant l'ensemble des noeuds et dont les distances cumulées seront minimum.

Définition - Application du SCP au PDPTW

On définit :

- $\Omega = \{1, \ldots, |\Omega|\}$ un ensemble de routes non dominées
- c_ω le coût de ω $\forall \omega \in \Omega$
- R un ensemble de requête
- K le nombre maximal de véhicules
- $\Omega_r \subset \Omega$ l'ensemble des routes servent la requête r $\forall r \in R$

Soit x_ω, la variable de décision telle que :

$$x_\omega = \begin{cases} 1 & \text{Si la route } \omega \text{ est selectionée dans une solution} \\ 0 & \text{sinon} \end{cases}$$

PDPTW - Matheuristique - Composante SCP

Le programme linéaire du Set Covering s'écrit :

$$\min \sum_{\omega \in \Omega} c_\omega x_\omega \tag{1}$$

$$\sum_{\omega \in \Omega_r} x_\omega \geq 1 \qquad \forall r \in \mathcal{R} \tag{2}$$

$$\sum_{\omega \in \Omega} x_\omega \leq K \tag{3}$$

$$x_\omega \in \{0,1\},\ \omega \in \Omega \tag{4}$$

Suppresion de requête doublon

Dans cette modélisation, on autorise une requête r à être couverte par plusieurs routes. Il sera donc possible d'avoir à simuler le coût des routes après suppression de r et retirer r dans les routes présentant un meilleur gain.

PDPTW - Conclusions et perspectives

Etat actuel des expérimentations

La première version de notre implémentation du LNS n'arrive pas à fournir un ensemble de routes suffisamments variées pour couvrir toutes les requêtes avec un nombre réduit de véhicules. **L'impact du SCP est donc assez limité.**

Améliorations possibles :

- La diversification des opérateurs de reconstruction.
- La pondération de certains opérateurs de reconstruction suivant leurs efficacités face aux situations rencontrées par l'heuristique.
- Une modification du nombre d'itérations de LNS requises avant le déclenchement du SCP.
- ...

Conclusion personnelle

La découverte du domaine de la recherche académique.

Merci pour votre attention