



Un problème de cueillette et livraison avec plaques tournantes

Jean-François Cordeau

Jean-François Côté

Jean-Yves Potvin

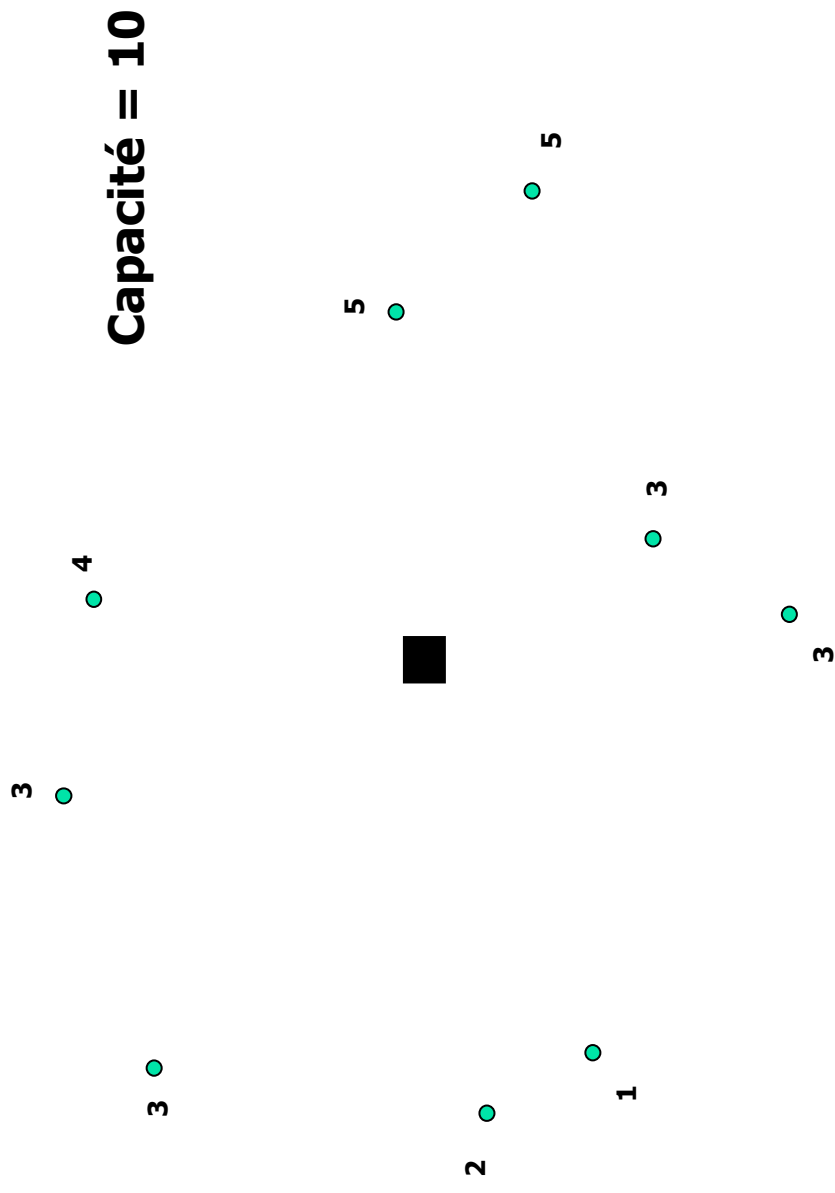
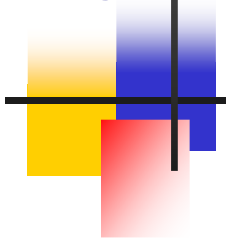
Marie-Ève Rancourt

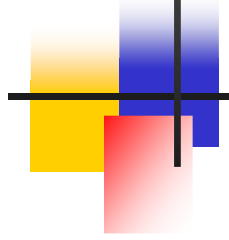
GIRO

Charles Fleurent

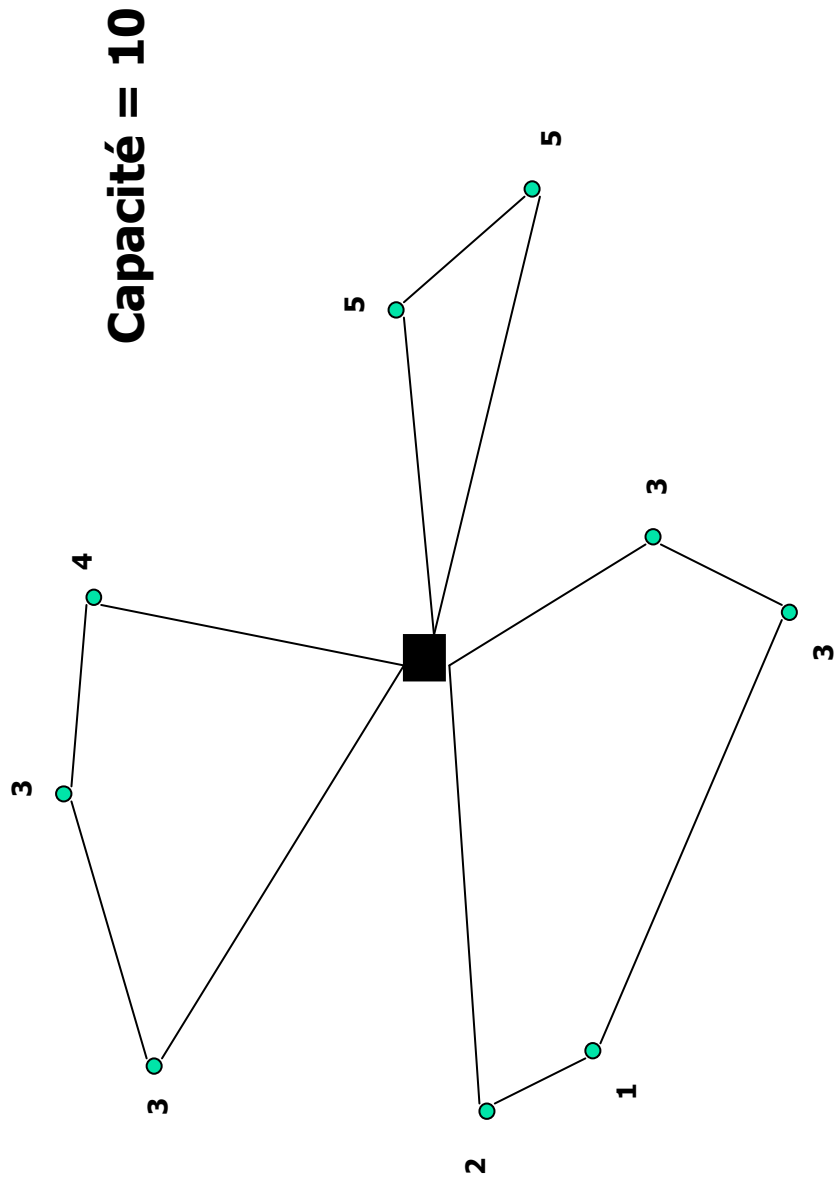
Patrick St-Louis

Problème de tournées de véhicules avec contraintes de capacité





Problème de tournées de véhicules avec contraintes de capacité

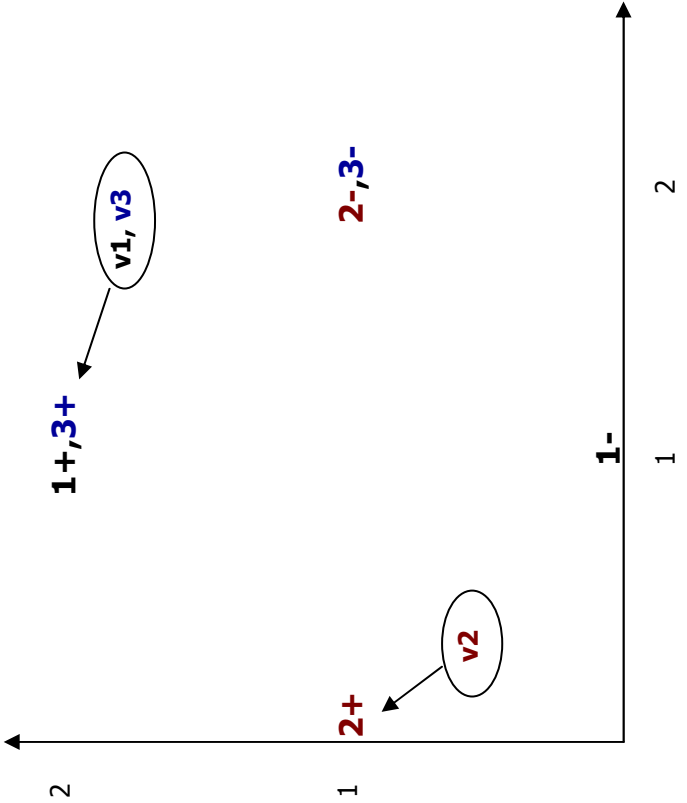


Problèmes de cueillette et

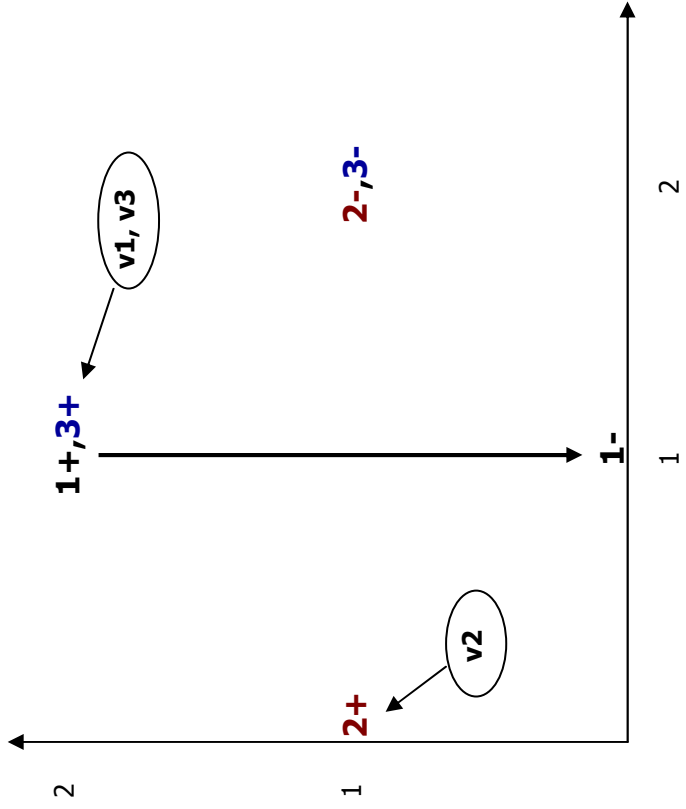
livraison

- Chaque client (requête) correspond à une paire de sommets
 - Collecte
 - Livraison
- Contrainte de précedence
- Contrainte de couplage

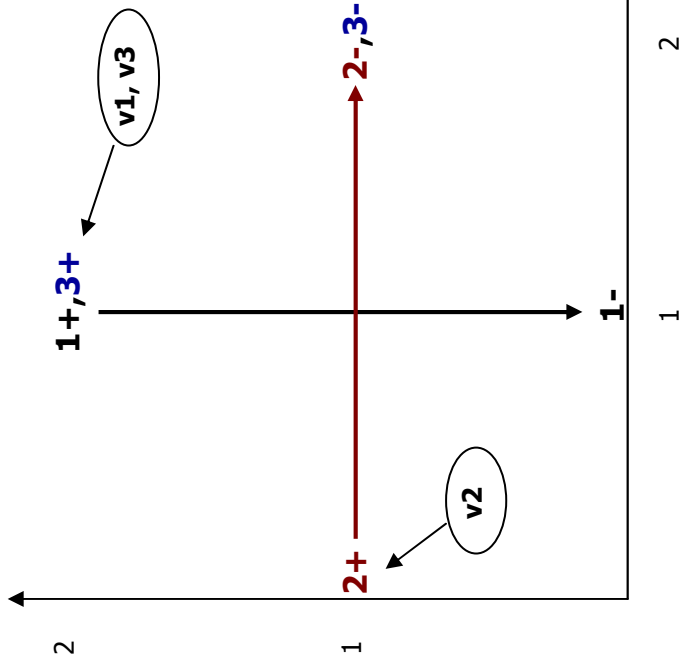
client	i+	i-	$[a_{i+}, b_{i+}]$	$[a_{i-}, b_{i-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]



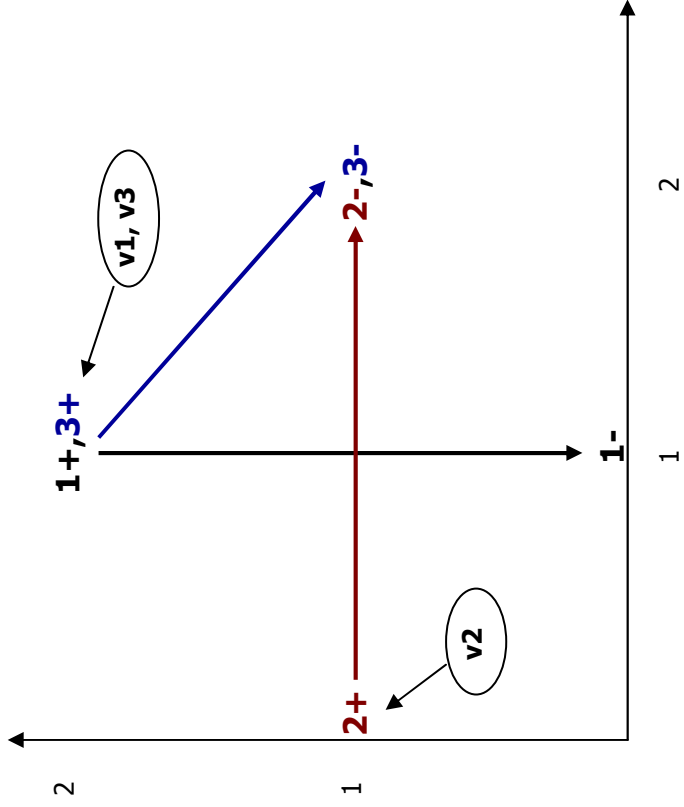
client	i_+	i_-	$[a_{i_+}, b_{i_+}]$	$[a_{i_-}, b_{i_-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]



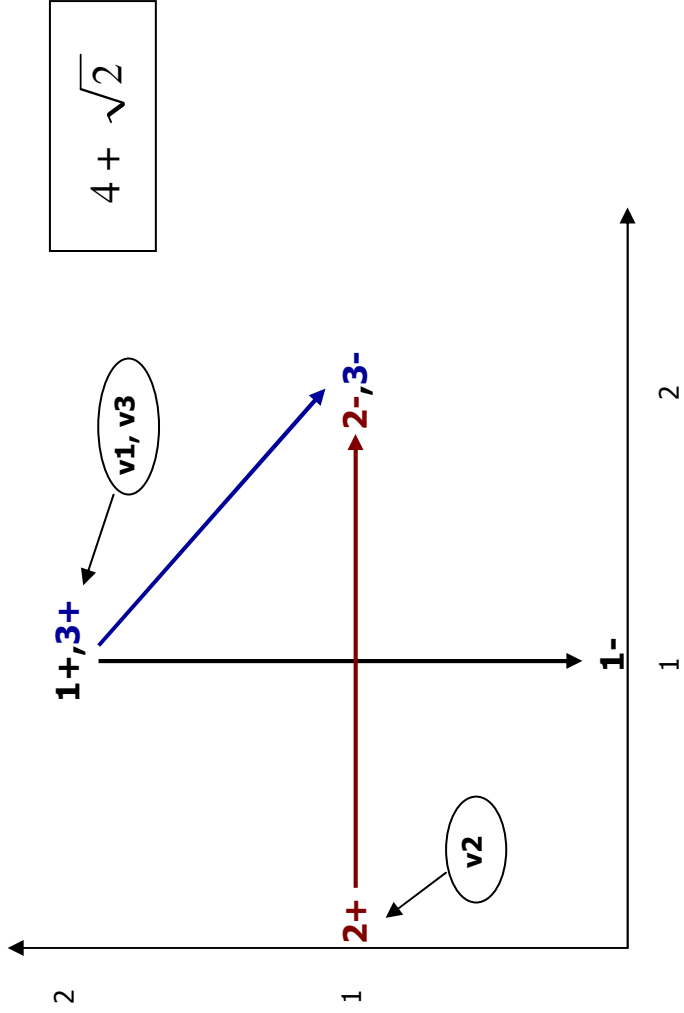
client	$i+$	$i-$	$[a_{i+}, b_{i+}]$	$[a_{i-}, b_{i-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]



client	i_+	i_-	$[a_{i_+}, b_{i_+}]$	$[a_{i_-}, b_{i_-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]

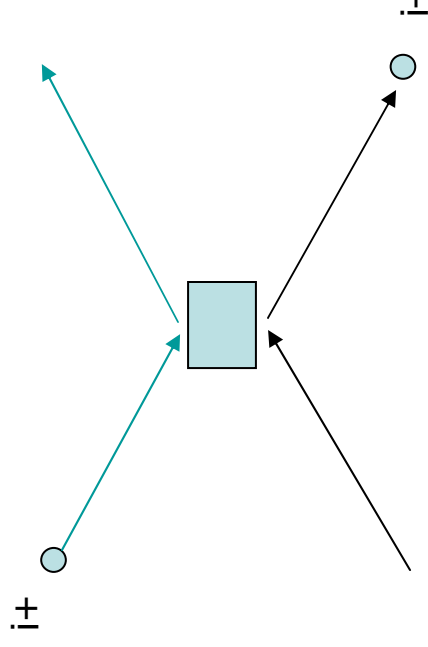


client	i_+	i_-	$[a_{i_+}, b_{i_+}]$	$[a_{i_-}, b_{i_-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]

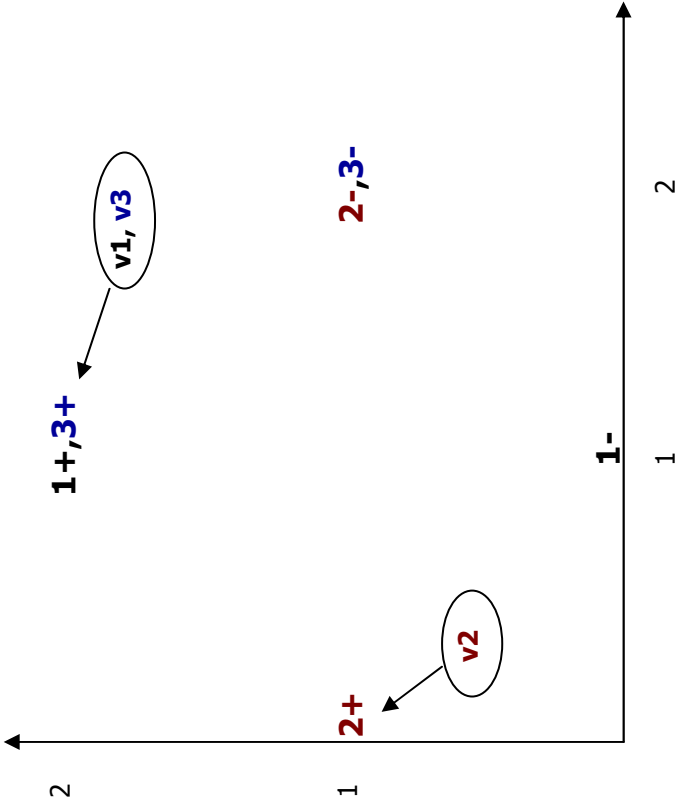


Plaques Tournantes

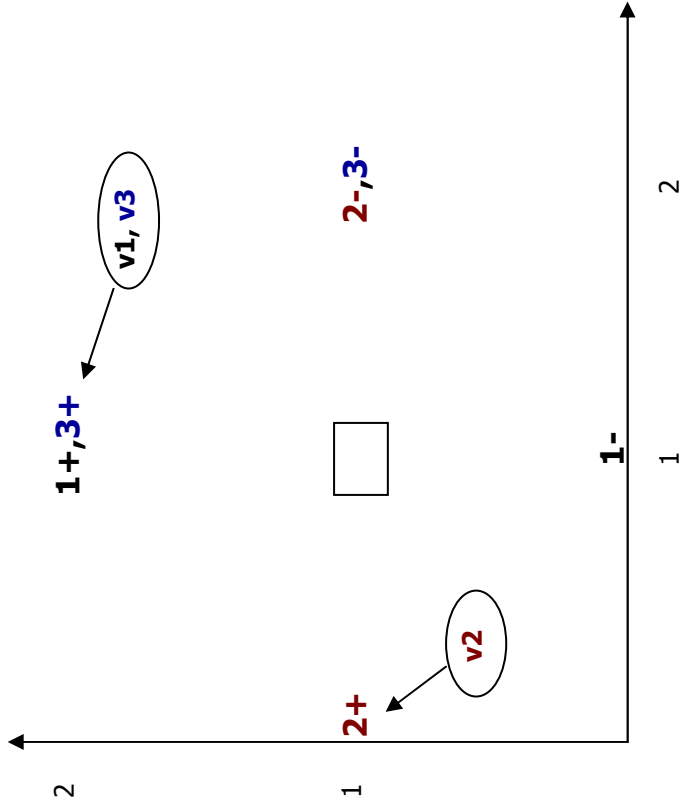
- Permettent de relaxer la contrainte de couplage



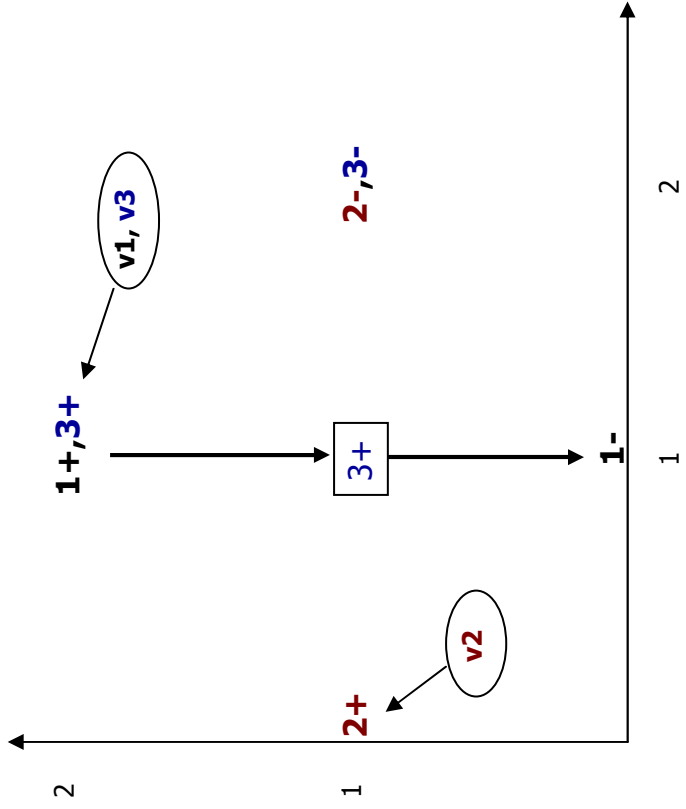
client	i+	i-	$[a_{i+}, b_{i+}]$	$[a_{i-}, b_{i-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]



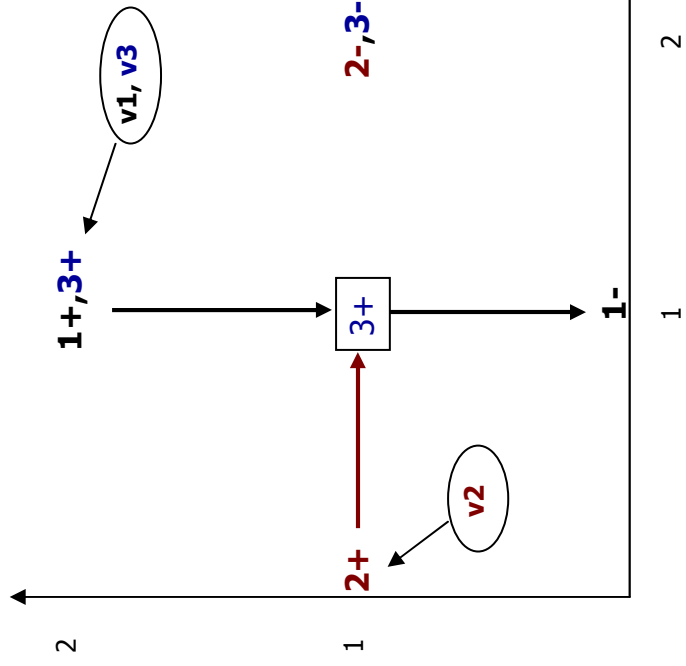
client	$i+$	$i-$	$[a_{i+}, b_{i+}]$	$[a_{i-}, b_{i-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]



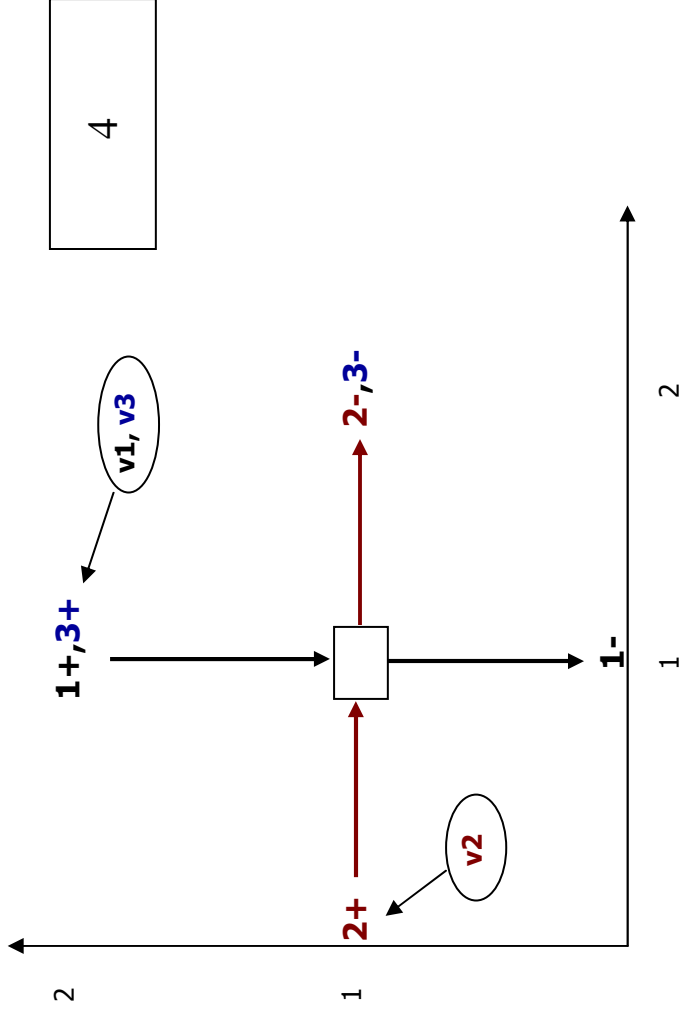
client	i_+	i_-	$[a_{i_+}, b_{i_+}]$	$[a_{i_-}, b_{i_-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]



client	$i+$	$i-$	$[a_{i+}, b_{i+}]$	$[a_{i-}, b_{i-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]



client	i_+	i_-	$[a_{i_+}, b_{i_+}]$	$[a_{i_-}, b_{i_-}]$
1	(1,2)	(1,0)	[0,0]	[0,2]
2	(0,1)	(2,1)	[0,0]	[0,2]
3	(1,2)	(2,1)	[0,0]	[0,2]



4



Revue de littérature

- Cortés C.E., Matamala M., Contardo C., « The Pick-Up and Delivery Problem with Transfers: Formulation and a Branch-and-Cut Solution Method », soumis à *European Journal of Operational Research*.
- Grünert T., Sebastian H.-J., « Planning Models for Long-Haul Operations of Postal and Express Shipment Companies », *European Journal of Operational Research* 122, 289-309, 2000.
- Irnich S., « A Multi-Depot Pickup and Delivery Problem with a Single Hub and Heterogeneous Vehicles », *European Journal of Operational Research* 122, 310-332, 2000.
- Laporte G., Mitrović-Minić S., « The Pickup and Delivery Problem with Time Windows and Transshipment », *INFOR*, 217-227, 2006.
- Mues C., Pickl S., « Transshipment and Time Windows in Vehicle Routing », in Proceedings of the 8th Int. Symposium on Parallel Architectures, Algorithms and Networks, IEEE Computer Society Press, 113-119, 2005.



Problème proposé par GIRO

- Problème de cueillette et livraison
- Requêtes
 - Demande
 - Fenêtres de temps souples au point de cueillette et de livraison (pénalité si violation de la borne supérieure)
- Plaques tournantes
 - Capacité
 - Fenêtres de temps



Problème proposé par GIRO

- Flotte de véhicules
 - Hétérogène
 - Disponibilité
 - Capacité
 - Compatibilité véhicule-requête
- Routes
 - Distance maximale
 - Durée maximale
 - Autres considérations (convention collective)



Problème proposé par GIRO

- Objectif
 - Minimiser le nombre de véhicules
 - Minimiser la distance totale (temps total) de parcours des véhicules + pénalités pour la violation des fenêtres de temps des requêtes.
- Problème de grande taille
 - 5 000 requêtes
 - Plus de 100 véhicules
 - 5-10 plaques tournantes



Outils disponibles chez GIRO

- Algorithme pour problèmes de cueillette et livraison standard (pas de plaque tournante)
 - Construction par insertion
 - Amélioration (à l'aide de déplacement de requêtes ou d'échanges de requêtes)
- La procédure d'amélioration est beaucoup plus coûteuse que la procédure de construction.

Nouvelle approche algorithmique



- Construction de solutions différentes à l'aide d'une version stochastique de la procédure d'insertion.
 - En perturbant l'ordre d'insertion des requêtes.
 - En choisissant, pour chaque requête, une insertion quelconque parmi les x meilleures insertions.
- On applique la procédure d'amélioration à partir de la meilleure solution obtenue.
 - En partant d'une meilleure solution, on réduit le temps de calcul.
 - La solution obtenue à la fin de la procédure d'amélioration risque aussi d'être meilleure.



Algorithmes proposés

- Exploiter au mieux les algorithmes déjà développés chez GIRO.



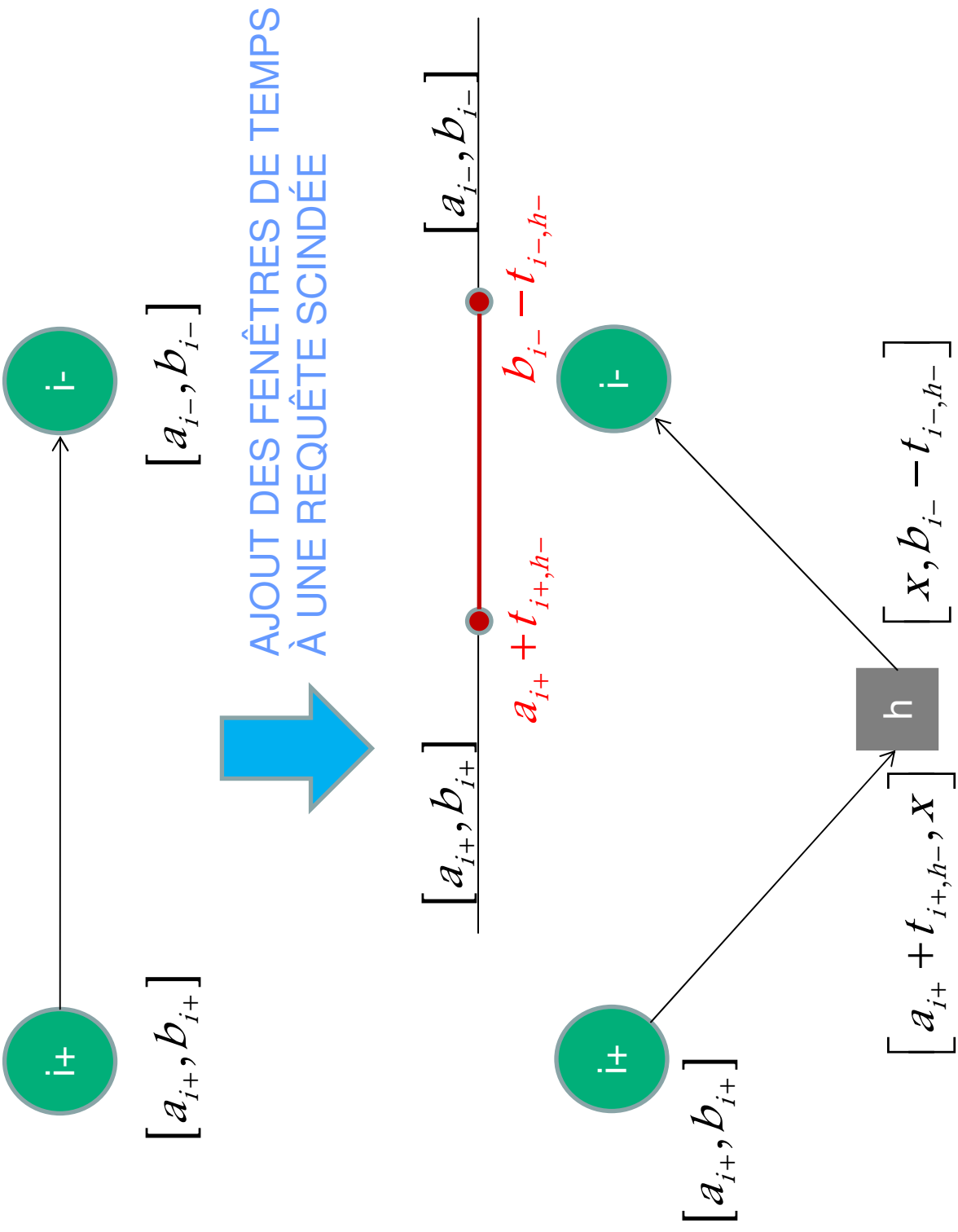
Algorithmes proposés

- Première approche:
 - Appliquer l'algorithme de construction pour cueillette et livraison.
 - Post-optimisation (avec inclusion de plaques tournantes dans la solution)
 - Les détails de cette étape seront explicités ultérieurement



Algorithmes proposés

- Deuxième approche
 - L'idée de cette approche est de transformer les données afin que l'algorithme de construction pour cueillette et livraison produise une solution qui utilise des plaques tournantes.
- Modification de certaines requêtes
 - Déterminer *a priori* les requêtes qui doivent être scindées (i.e. qui doivent passer par une plaque tournante).
 - $i+ \rightarrow i-$ deviendra $(i+ \rightarrow h- \text{ et } h+ \rightarrow i-)$ avec $h- = h+$.
 - Fenêtre de temps en $h-$ et $h+$: $[a_{i+} + t_{i+h-}, x]$ et $[x, b_{i-} - t_{h+i-}]$.



Scission des requêtes

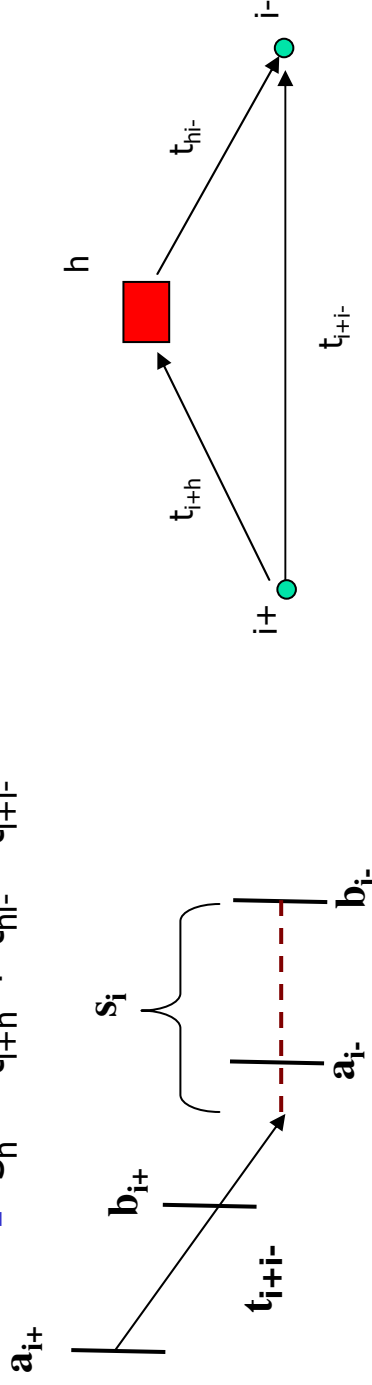
- Déterminer a priori les requêtes qui doivent être scindées:

- "slack" pour la livraison

- $s_i = b_{i-} - (a_{i+} + t_{i+i-})$

- Temps du détour

- $S_h = t_{i+h} + t_{hi-} - t_{i+i-}$





Scission des requêtes

- Soit H l'ensemble des plaques tournantes
- Soit S l'ensemble des requêtes à scinder (initialement vide)
- Soit $r_c \geq 1$ un ratio critique
- Pour chaque requête i calculer:

$$r_i = \frac{S_i}{S_{h^*}} = \max_{h \in H} \frac{S_i}{S_h}$$

- Si $r_i \geq r_c$ alors
 - $S \leftarrow S \cup \{i\}$
 - $h_j \leftarrow h^*$
 - $n_{h^*}^0 \leftarrow n_{h^*}^0 + 1$



Scission des requêtes

- Pour $k = 1, \dots, K$ faire

- Pour $i \in S$

$$r_i = \frac{s_i}{s_{h^*}} = \max_{h \in H} \frac{s_i}{s_h - f(n_h^{k-1})}$$

$$h_j \leftarrow h^*$$

$$n_{h^*}^k \leftarrow n_{h^*}^k + 1$$

- Scinder chaque requête $i \in S$ à la plaque tournante h_j



Deuxième approche (suite)

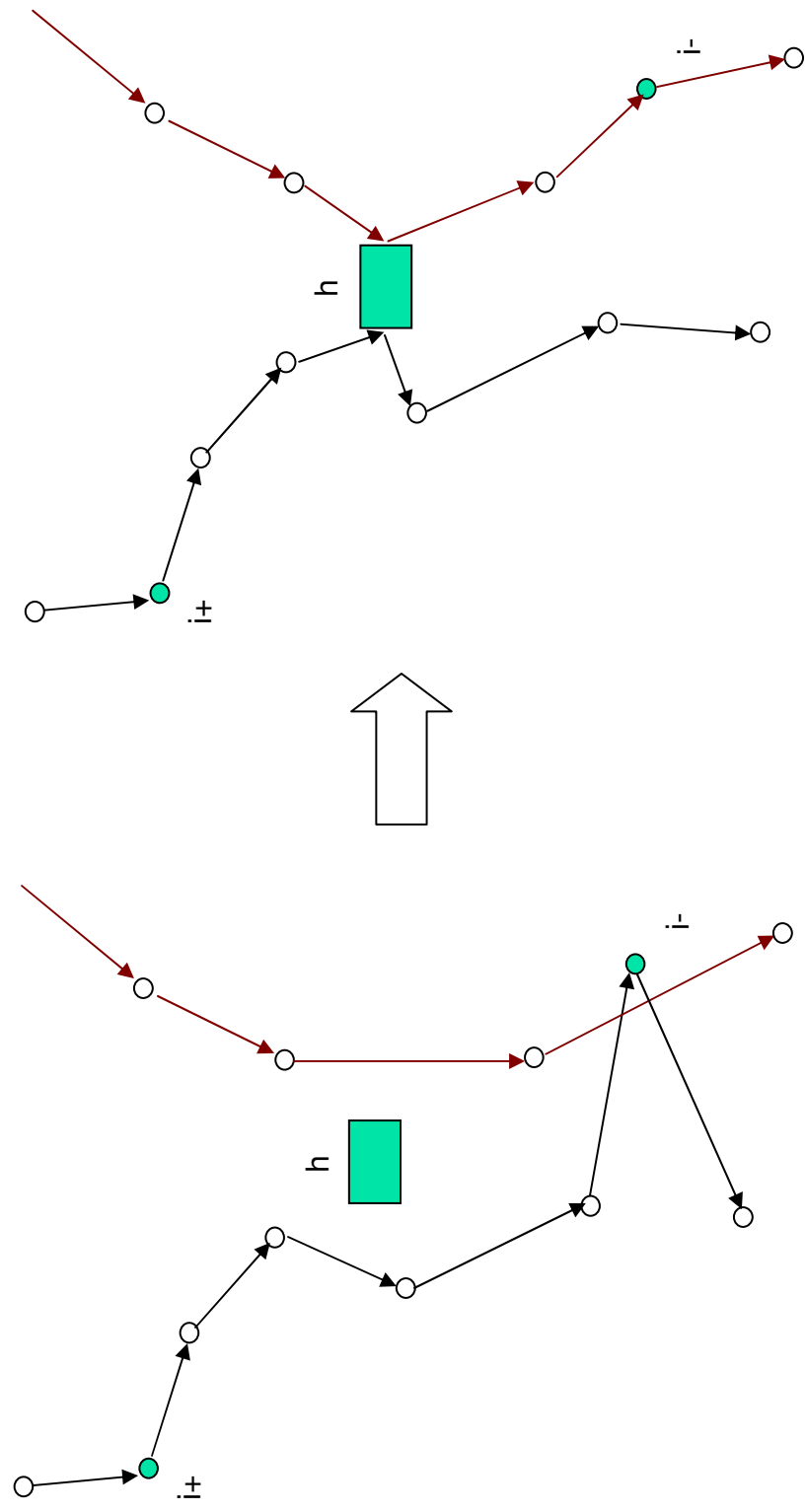
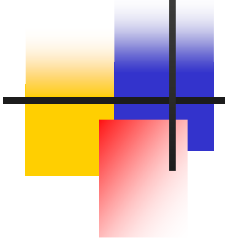
- Appliquer l'algorithme de construction pour cueillette et livraison standard.
- Post-optimisation (les détails suivent)



Post-optimisation

- Utilisation de plaques tournantes dans la solution
 - Scission simple
 - Scission multiple
 - Insertion forcée de plaques tournantes
- Fusion
- Déplacement ou échange de requêtes intra-route.

Scission simple





Scission simple

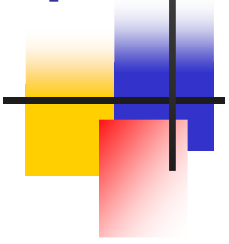
- Soit une requête i non scindée et une solution s
 - Retirer $i+$ et $i-$ de la solution s pour obtenir s' et calculer le gain obtenu G_i
 - $C^* \leftarrow \infty$
 - Pour chaque $h \in H$
 - scinder i en h : $(i+,h-)$ et $(h+,i-)$
 - Déterminer le meilleur coût d'insertion C_i de $(i+,h-)$ et $(h+,i-)$ dans la solution s'
 - Si $C_i < C^*$ alors $C^* \leftarrow C_i$ et $h^* \leftarrow h$
- Si $C^* < G_i$ alors insérer $(i+,h^*-)$ et $(h^*+,i-)$ dans s' ,
- Sinon conserver la solution s .



Scission multiple

- Généralisation de scission simple à un nombre variable de plaques tournantes.
- Exemple pour 2 plaques tournantes
 - on cherche la meilleure paire h' et $h'' \in H$ permettant la scission de la requête i selon:

$(i+, h'-), (h'+, h''-)$ et $(h''+, i-)$



Insertion de plaques tournantes

- On force certaines routes à passer par une même plaque tournante h .
- pour un certain nombre d'itérations appliquer l'opérateur de scission en h seulement.
- S'il y a des routes qui n'utilisent pas h , on retire h de ces routes.



Fusion

- Soit une requête i scindée et une solution s
 - Retirer i de la solution s pour obtenir s' et calculer le gain obtenu G_i .
 - Déterminer le meilleur coût d'insertion C_i de la requête i dans une même route de s' .
 - Si $C_i < G_i$ alors insérer la requête i dans s' ,
 - Sinon conserver la solution s .



Opérateurs multi-requêtes

- Exploration plus large de l'espace des solutions (perturbation).
- Fusion multi-requêtes
 - On retire plusieurs requêtes scindées de la solution s pour obtenir s' .
 - On réinsère chacune des requêtes à moindre coût dans s' de telle façon que chaque requête se retrouve dans une seule route.
- Scission multi-requêtes



Autres idées

- Création de tournées entre les plaques tournantes
 - favorise la scission multiple des requêtes (i.e., plusieurs plaques tournantes intermédiaires entre le point de cueillette d'une requête et son point de livraison)
- Modification des fenêtres de temps pour les requêtes scindées (qui passent par une plaque tournante)
- Considérer la possibilité d'agréger plusieurs requêtes « similaires » afin de réduire la taille du problème.



Conclusion

- L'introduction de plaques tournantes s'est révélée plus complexe que nous le croyions au départ.
- Cette présentation n'a décrite que quelques unes des idées élaborées au cours de la semaine.
- Nous envisageons des sujets de thèse en collaboration avec GIRO qui s'inspireront du problème proposé.