Projet de recherche

IFT-7020

Circuit d'entraînement pour cycliste

présenté à

M. Claude-Guy Quimper - Enseignant

par

Alexandre Gariépy
Numéro de dossier : 111 046 788

Jean-Samuel Bédard
Numéro de dossier : 111 043 631

Mathieu Carpentier
Numéro de dossier : 111 044 647

22 avril 2015

Département d'informatique et de génie
Faculté des sciences et de génie

Circuit d'entraînement pour cycliste

Alexandre Gariépy, Jean-Samuel Bédard et Mathieu Carpentier

14 avril 2015

Résumé

La qualité d'un parcours d'entraînement pour cycliste s'évalue par le fait qu'il répond aux besoins de l'athlète en terme de durée, de nombre de paliers d'efforts et de repos, etc. Dans cet article, nous nous intéressons à trouver un trajet d'entraînement optimal dans la carte d'un quartier donné qui respecte le mieux ce qui a été demandé par l'utilisateur. Pour ce faire, on modélise le problème dans un modèle de programmation linéaire et on compare les résultats de deux fonctions objectifs différentes.

1 INTRODUCTION

La recherche de chemin dans un graphe est un problème courant en informatique. Toutefois, le chemin recherché est pratiquement toujours le plus court. Quand ce n'est pas le cas, il s'agit souvent de trouver celui qui respecte une ou plusieurs contraintes.

Par exemple, dans [1], on recherche le chemin le plus court qui satisfait une contrainte de délai, tandis que dans [2], on cherche le meilleur trajet pour traverser rapidement et sécuritairement un terrain hostile en temps de guerre. Dans tous les cas, les chemins recherchés ne comportent pas de cycles et sont les plus courts possible.

Dans cet article, nous montrons un modèle linéaire permettant de trouver un parcours de cycliste dont la courbe d'effort correspond à celle qui était demandée. Dit autrement, nous trouvons un chemin de longueur variable comportant des cycles et dont l'effort réel pour effectuer ce parcours correspond le plus possible à ce qui était demandé au départ.

La section 2 présente plus en détail les caractéristiques du problème à faire résoudre par notre modèle. Par la suite, la section 3 montre le modèle utilisé pour résoudre le problème et les différentes heuristiques utilisés pour évaluer le chemin. La section 4 présente la façon dont nous avons testé celui-ci suivi de la section 5 qui présente les résultats que nous avons obtenus. Finalement, la section 6 fait part des conclusions auxquelles nous sommes arrivées suite aux expérimentations.

2 DESCRIPTION DU PROBLÈME

Le problème consiste à générer automatiquement un parcours d'entraînement pour un cycliste dans une zone prédéterminée et en fonction d'un graphique d'effort désiré dans le temps.

Plusieurs problèmes découlent de cette situation. Le premier est qu'on cherche un chemin de longueur approximative. En effet, celui-ci peut être légèrement plus long ou plus court que celui demandé. Le deuxième est que le parcours doit permettre les cycles. Cela est essentiel si on veut que le cycliste puisse revenir à son point de départ à la fin de l'entraînement. De plus, pour avoir un mouvement plus naturel, le chemin ne doit pas permettre de revenir sur ces pas. Cela oblige l'utilisation de cycles pour pouvoir visiter le même endroit plus d'une fois. Le modèle présenté dans la section 3 montre l'approche utilisée pour résoudre ces problèmes.

Pour résoudre cette tâche, il faut aussi discrétiser la zone dans laquelle le cycliste s'entraînera. Cela est primordial pour pouvoir effectuer la recherche de chemin dans un graphe. Nous avons résolu ce problème en mettant un noeud sur chaque intersection et en reliant les noeuds selon les routes. De plus, nous avons supposé que chaque arête du graphe prend un temps équivalent à parcourir. Si la distance entre deux noeuds est trop longue, on ajoute des noeuds intermédiaires pour que le temps entre chaque noeud soit équivalent. Cela nous permet aussi de résoudre le problème des pentes qui sont plus longues à monter que descendre. En effet, il suffit d'ajouter des noeuds lors des montées.

Finalement, il reste le problème d'évaluer la pertinence du chemin trouvé. Pour ce faire, nous utilisons deux techniques : une distance de Hamming modifiée et une distance d'édition. Les détails sur la façon dont ces deux techniques ont été implémentées sont présentés dans la section 3.

3 APPROCHE PROPOSÉE

Le problème revient à une recherche de chemin dans un graphe orienté. Le chemin commence à un noeud choisi α et se termine à un noeud choisi β. α et β peuvent être égaux. Au noeud β, on ajoute une boucle de coût nul permettant de considérer des chemins de plusieurs longueurs différentes.

Chaque arête est étiquetée de deux valeurs. La première est le coût en temps pour parcourir l'arête. Pour simplifier le problème, chaque arête du graphe a un coût égal. Ainsi, on peut omettre ces valeurs dans la représentation du graphe. Aussi, cela permet d'avoir une correspondance directe entre le nombre de noeuds parcourus et le temps écoulé. La deuxième étiquette est l'effort nécessaire pour parcourir l'arête. Cet effort est un nombre entier fixé qualitativement :

- 1 représente un effort léger, voire nul, lorsque le cycliste descend une pente.
- 2 représente un effort normal, c'est-à-dire l'effort nécessaire pour parcourir un terrain plat.
- 3 représente un effort légèrement plus élevé que la normale, qui peut être causé par exemple par un faux plat.
- 4 représente un effort élevé, nécessaire par exemple pour monter une pente peu abrupte.
- 5 représente un effort très élevé, nécessaire par exemple pour monter une pente abrupte.
- 0 signifie que l'entraînement est terminé. Cette étiquette n'est possible que sur la boucle autour du noeud β.

Soit a une chaîne de nombres entiers représentant l'effort désiré par unité de temps et b une autre chaîne d'entiers représentant l'effort réel du chemin trouvé dans le graphe, le but est de minimiser la distance entre a et b. Ces deux chaînes sont de longueur T, soit le nombre d'unités de temps maximal de l'entraînement. On remarque que l'entraînement peut être plus court que T, auquel cas la fin de la chaîne sera constituée de 0. Deux mesures de distance sont utilisées.

La première mesure est la distance de Hamming à laquelle on ajoute une pondération. Plus spécifiquement, $\sum_{i=1}^{L} |a_i - b_i|$. On remarque que l'on compare les éléments de même position des deux chaînes. On compte le nombre d'éléments différents. Si les éléments sont différents, à la place d'ajouter 1 à la distance comme pour la distance de Hamming standard, on ajoute la différence en valeur absolue entre a_i et b_i.

La deuxième mesure est la distance d'édition. La distance d'édition compte le nombre d'insertion, de suppression et de remplacements pour passer d'une chaîne a à une chaîne b. L'approche usuelle pour calculer la distance d'édition est un algorithme de programmation dynamique. Soit $d_{i,j}$ la distance d'édition entre la sous-chaîne formée des i premiers caractères de a et des j premiers caractères de b. On a relation de récurrence :

$$d_{i,j} = \begin{cases} i & \text{si } j = 0 \\ j & \text{si } i = 0 \\ min(d_{i-1,j}, d_{i,j-1}, d_{i-1,j-1}+) & \text{sinon} \end{cases}$$

Comme le montre [3], il est possible de modéliser la distance d'édition dans un programme linéaire.

Ainsi, deux approches différentes sont proposées pour comparer un trajet réel avec un trajet désiré. Par contre, dans les deux cas, les contraintes modélisant le mouvement du cycliste dans le graphe sont les mêmes ; seuls le calcul de distance et la fonction objectif changent. Donc, dans un premier temps, voici les contraintes communes aux deux modèles :

— Soit $Arcs$, l'ensemble de tous les arcs du graphe. Les éléments de cet ensemble contiennent 4 valeurs : le numéro du noeud de départ dep, le numéro du noeud de destination $dest$, le coût en temps de l'arc $cout$ et le coût en effort de l'arc $effort$.

— Soit T, la longueur maximale d'un chemin.

— Soit L , une matrice de booléens de dimension $|Arcs| \times T$ dont $L[i][j]$ indique si oui ou non l'arc i est parcouru au temps j

On a les contraintes :

$$\sum_{arc \in Arcs} L_{arc,k} = 1 \quad \forall 1 \leq k \leq T \tag{1}$$

$$\sum_{arc \in Arcs | arc.dep = \alpha} L_{arc,1} = 1 \tag{2}$$

$$\sum_{arc \in Arcs | arc.dest = \beta} L_{arc,1} = 1 \tag{3}$$

$$L_{arc,k} \leq L_{arc,k+1} \quad \forall arc \in Arcs \,|\, arc.dep = arc.dest = \beta \quad \forall 1 \leq k \leq T-1 \tag{4}$$

$$\sum_{x \in Arcs} L_{x,k} = \sum_{\substack{y \in Arcs \\ | x.dest = y.dep}} L_{y,k+1} \quad \forall 1 \leq k \leq T-1 \tag{5}$$

$$1 \geq 2 - L_{x,k} - L_{y,k+1} \quad \forall x, y \in Arcs \,|\, x.dep \neq x.dest \wedge y.dest = x.dep \quad \forall 1 \leq k \leq T-1 \tag{6}$$

$$1 \geq 2 - L_{x,k} - L_{y,k+2} \quad \forall x, y \in Arcs \,|\, x.dep \neq x.dest \wedge y.dest = x.dep \quad \forall 1 \leq k \leq T-2 \tag{7}$$

$$1 \geq 2 - L_{x,k} - L_{y,k+3} \quad \forall x, y \in Arcs \,|\, x.dep \neq x.dest \wedge y.dest = x.dep \quad \forall 1 \leq k \leq T-3 \tag{8}$$

$$b_i = \sum_{arc \in Arcs} L_{arc,i} \times arc.effort \quad \forall 1 \leq k \leq T \tag{9}$$

(1) assure que le cycliste n'est qu'à un seul endroit par unité de temps. (2) et (3) assurent que le trajet commence bien au noeud α et se termine bien au noeud β, respectivement. (4) assure qu'une fois qu'on entre dans la boucle autour de β, on y reste jusqu'au temps T. Autrement dit, une fois

l'entraînement terminé, le cycliste ne repart pas. La contrainte (5) assure après avoir parcouru un arc, le prochain arc à parcourir part bel et bien du noeud d'arrivée de l'arc précédent. Les contraintes (6), (7) et (8) assurent qu'on ne revienne pas au même noeud avant trois unités de temps. En pratique, cela s'avère suffisant pour éviter que le cycliste ne cesse de faire demi-tour. Une exception est ajoutée à ces trois contraintes pour la boucle autour de β. Finalement, (9) met les efforts réels dans le vecteur b.

3.1 Distance de Hamming

Regardons comment on trouve le trajet optimal en utilisant la distance de Hamming pondérée. On utilise les variables :

— Soit D un vecteur de longueur T contenant la différence entre a_i et b_i.

— Soit D^+ et D^- des vecteurs positifs de taille T utilisés pour représenter les valeurs absolues.

En utilisant la distance de Hamming, on souhaite minimiser les distances entre les efforts désirés et les efforts réels.

$$\min \sum_{k=1}^{T} D_k^+ + D_k^-$$

Pour ce faire, on ajoute les contraintes suivantes au modèle :

$$D_k = b_k - a_k \quad \forall 1 \leq k \leq T$$
$$D_k = D_k^+ - D_k^- \quad \forall 1 \leq k \leq T$$

3.2 Distance d'édition

Si on veut utiliser la distance d'édition, la fonction objectif change. Comme proposé dans [3], on construit un graphe g de $(T+1) \times (T+1)$ noeuds, notés $d_{i,j}$ avec des indices i et j allant de 0 à T. On organise les noeuds en une grille carrée. Pour chaque paire de noeuds adjacents horizontalement dans la grille, on ajoute ajoute un arc de coût 1 allant de $d_{i,j-1}$ à $d_{i,j}$ dans un ensemble H. Pour chaque paire de noeuds adjacents verticalement dans la grille, on ajoute ajoute un arc de coût 1 allant de $d_{i-1,j}$ à $d_{i,j}$ dans un ensemble V. Pour chaque paire de noeuds $d_{i-1,j-1}, d_{i,j}$, on ajoute ajoute un arc de coût 1 allant de $d_{i-1,j-1}$ à $d_{i,j}$ dans un ensemble D et un autre arc de coût 0 allant lui aussi de $d_{i-1,j-1}$ à $d_{i,j}$ dans un autre ensemble E'_q.

La distance d'édition entre les chaînes a et b correspond à la longueur au chemin le plus court entre $d_{0,0}$ et $d_{T,T}$ dans le graphe g contenant les arêtes des ensembles H, V, D et E'_q. Par contre, on ne peut emprunter un arc (de coût nul) de l'ensemble E'_q entre $d_{i-1,j-1}$ et $d_{i,j}$ uniquement que si $a_i = b_j$.

Le modèle linéaire en conforme à celui proposé par [3] :

— Soit Q une matrice de 0 et de 1 de taille T par T dont $Q_{i,j} = 1$ si et seulement si $a_i \neq b_j$.

— Soit $f(x)$ une fonction qui retourne 1 si et seulement si l'arête x du graphe g est sur le chemin le plus court ente $d_{0,0}$ et $d_{T,T}$.

— $\gamma = 5$ le plus grand nombre qu'on retrouve dans les chaînes a et b.

La fonction objectif est :

$$\min \sum_{x \in H \cup V \cup D} f(x)$$

Sujet à, pour tout $1 \leq i \leq T$ et $1 \leq j \leq T$:

$$a_i - b_j \leq \gamma Q_{i,j}$$
$$a_i - b_j \geq -\gamma Q_{i,j}$$
$$f(x) \leq 1 - Q_{i,j} \; \forall x \in E'_q \; tel \; que \; x.dest = d_{i,j}$$

et sujet à, pour tout noeud n sauf $d_{0,0}$ et $d_{T,T}$:

$$\sum_{\substack{w \in H \cup V \cup D \cup E'_q \\ |w.dep = n}} f(w) = \sum_{\substack{z \in H \cup V \cup D \cup E'_q \\ |z.dest = n}} f(z)$$

et sujet à

$$\sum_{\substack{z \in H \cup V \cup D \cup E'_q \\ |z.dest = d_{T,T}}} f(z) = 1$$

4 PROTOCOLE D'EXPÉRIMENTATION

Pour tester nos différents modèles, à l'aide de données géographiques provenant de Google Map, nous avons discrétisé une carte dans un graphe orienté en ne considérant que les principales voies cyclables. Nous avons pris la peine de prendre un quartier qui comporte beaucoup de pentes dans les différentes routes pour simuler une zone d'entraînement réaliste.

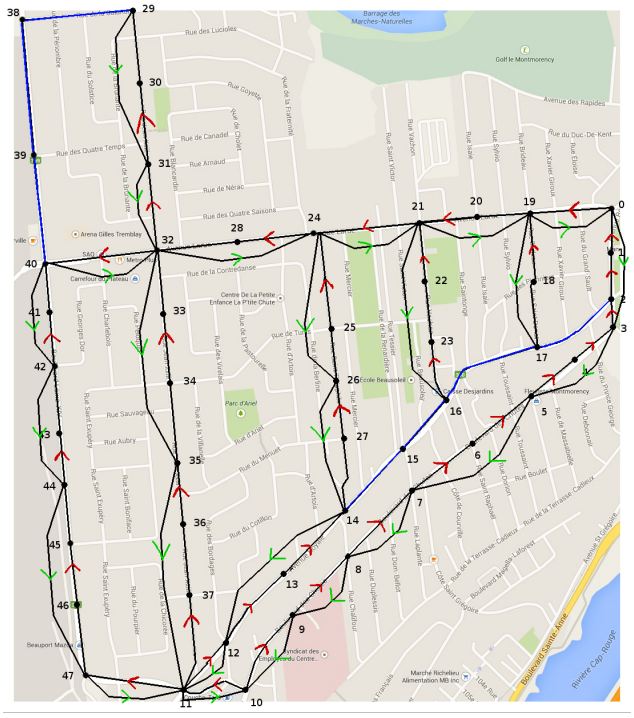

FIGURE 1 – Quartier résidentiel

Pour chaque route sur la carte, nous retrouvons au minimum 2 arêtes en sens opposé pour représenter les deux sens possibles de la route. Sur les différents arcs du graphe de la figure 1, les

flèches rouges représentent une ascension et les vertes une descente ; les arêtes en bleu représentent un chemin plat. De plus, chaque arc représente en moyenne un temps de déplacement de 1 minute. Nous retrouvons donc un graphe à 48 noeuds et 88 arêtes.

Comme instance de test, nous avons déterminé une courbe représentant l'effort désiré à chaque unité de temps de 1 minute pour un trajet de 40 minutes. Le problème revient à déterminer un cycle dans le graphe fourni en tentant de s'approcher le plus possible de l'effort désiré à chaque unité de temps. Les modèles présentés dans la section 3 ont été implémentés dans le solveur linéaire *CPLEX*.

5 RÉSULTATS

La figure 2 présente les résultats obtenus avec les deux différentes fonctions objectif présentées dans la section 3. Pour la distance de Hamming, la valeur optimale de la fonction objectif est 18. Cette solution optimale a été trouvée en 5.88 secondes sur un processeur Intel i7-3610QM.

Pour la distance d'édition, la valeur optimale de la fonction objectif est 13. Cette solution a été trouvée en 12 minutes et 19.85 secondes sur le même processeur Intel i7-3610QM.

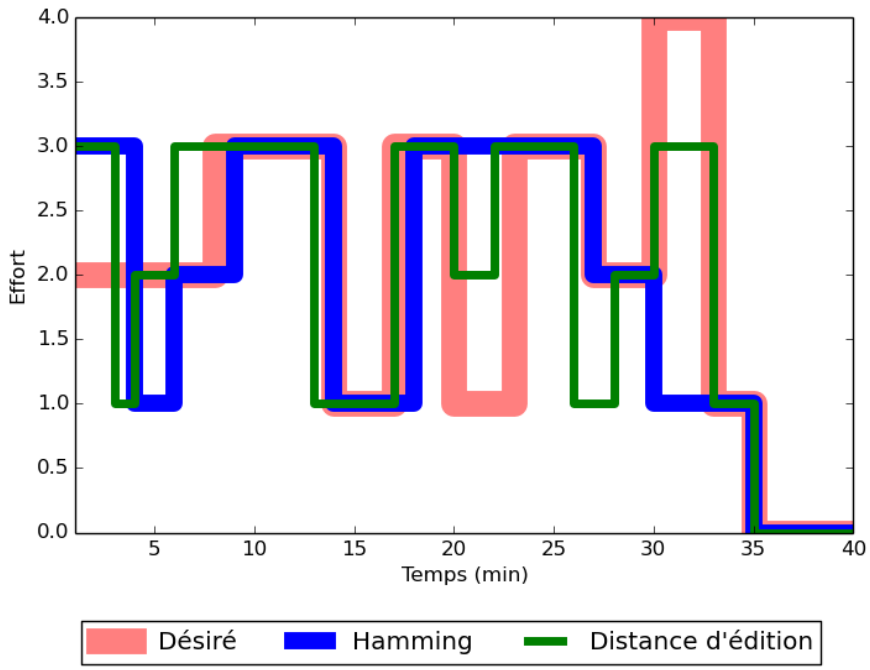

FIGURE 2 – Effort voulu et efforts obtenus en fonction du temps

La figure 3 montre visuellement les trajets obtenus. Puisqu'un trajet peut passer plusieurs fois par le même arc, trois couleurs sont utilisées pour tracer les trajets pour qu'on puisse bien distinguer les étapes du trajet. Le trajet commence en jaune, puis continue en orange, et se termine en vert.

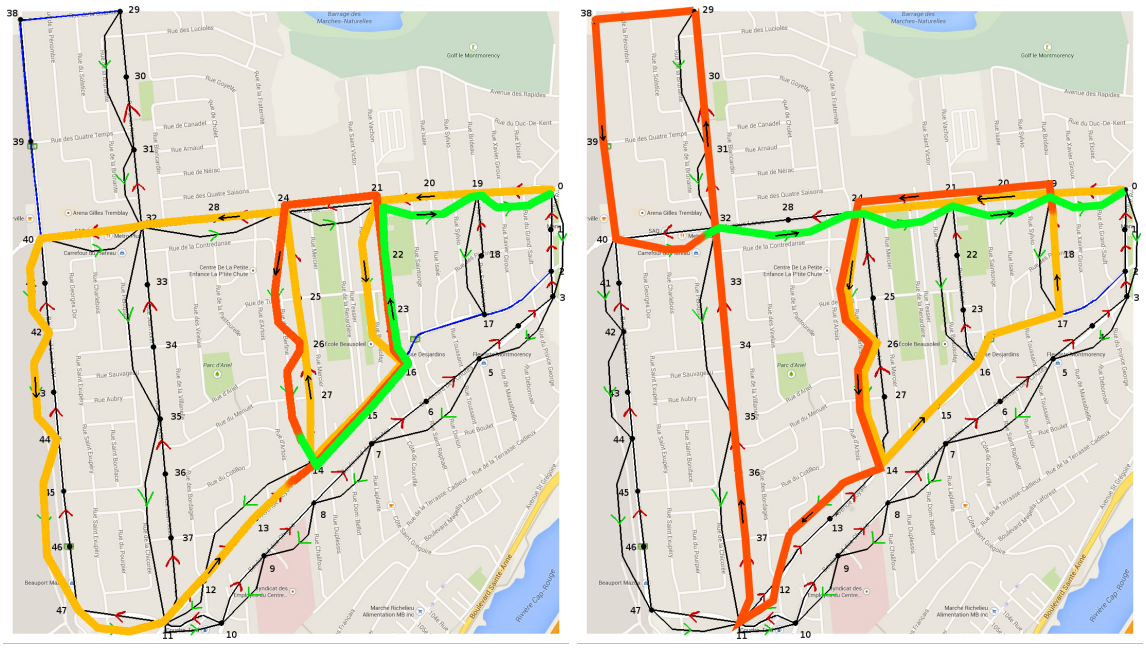

FIGURE 3 – Trajets obtenus, à gauche en utilisant la distance de Hamming, à droite en utilisant la distance d'édition

6 DISCUSSION

En observant la figure 2, on peut comparer les différences entre la distance de Hamming et la distance d'édition. Cela est essentiel, car on ne peut pas comparer directement les valeurs objectives retournées par les deux méthodes. En effet, la distance de Hamming est pondérée. Cela signifie qu'un effort demandé de 3 comparé à un effort réel de 1 compte pour 2. En utilisant la distance d'édition, une addition, suppression, etc. compte toujours uniquement pour un.

Notons que, dans les deux cas, le graphe d'effort obtenu suit relativement bien la courbe demandée. Toutefois, les deux méthodes ont aussi leurs inconvénients. Dans le cas de Hamming, la courbe d'effort reste collée sur ce qui était demandé ce qui est excellent. De plus, cette méthode est beaucoup plus rapide pour retourner une solution que la distance d'édition. Par contre, certains paliers d'efforts sont complètement ignorés. Par exemple, entre les minutes 30 et 33, un effort de 3 était demandé et le chemin trouvé n'en a pas tenu compte.

Dans le cas de la distance d'édition, la courbe d'effort ne suit pas exactement ce qui était demandé. Par contre, elle suit beaucoup plus « l'esprit » de ce que l'utilisateur souhaite faire. Par exemple, entre les minutes 17 et 27, il fallait faire un effort léger, suivi d'une descente avant de refaire un effort léger. La distance d'édition a trouvé un chemin qui demandait de faire un effort léger, suivi d'un plat avant de refaire un effort léger. De plus, cette méthode permet d'avoir une courbe d'effort décalée par rapport à ce qui était demandé. Par exemple, aux minutes 22 à 26, le même palier que celui demandé est présent, mais décalé d'une minute. Ceci est un avantage par rapport à la méthode de Hamming qui n'aurait pas pu le faire.

7 CONCLUSION

Dans cet article, nous avons présenté un modèle permettant de trouver un parcours d'entraînement dans un quartier donné. Celui-ci permet notamment de trouver un chemin de longueur

indéterminé et comportant des cycles qui correspond le mieux à ce qui était demandé au départ en terme d'effort. De plus, pour évaluer la pertinence du chemin trouvé, nous avons utilisé deux techniques différentes, soit une distance de Hamming modifiée et une distance d'édition.

Dans le futur, il serait intéressant de trouver le parcours d'entraînement qui fournit l'effort le plus semblable à ce qui était demandé et qui, en même temps, maximise la couverture du quartier. C'est-à-dire que le cycliste visite le plus de noeuds différents possibles afin de diversifier le parcours.

De plus, dans le cas des athlètes de haut niveau qui sont capables de parcourir de très grandes distances en peu de temps, il faut modifier la technique afin que le solveur puisse trouver une solution dans un temps raisonnable. On pourrait s'appuyer sur le travail de [4] pour résoudre ce problème. L'idée consisterait à trouver le chemin dans une zone très vaste, mais dont la majorité des routes ne sont pas marquées pour, par la suite, faire fonctionner le solveur sur les zones plus petites qui ont été traversées.

Références

[1] S. Chen, M. Song et S. Sahni, "Two techniques for fast computation of constrained shortest paths", *IEEE/ACM Trans. Netw.*, t. 16, n° 1, p. 105–115, fév. 2008, ISSN : 1063-6692. DOI : 10.1109/TNET.2007.897965. adresse : http://dx.doi.org.acces.bibl.ulaval.ca/10.1109/TNET.2007.897965.

[2] T. Oral et F. Polat, "A multi-objective incremental path planning algorithm for mobile agents", in *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 02*, sér. WI-IAT '12, Washington, DC, USA : IEEE Computer Society, 2012, p. 401–408, ISBN : 978-0-7695-4880-7. DOI : 10.1109/WI-IAT.2012.143. adresse : http://dx.doi.org.acces.bibl.ulaval.ca/10.1109/WI-IAT.2012.143.

[3] Moisan, Quimper, Gaudreault et Michaud, "Re-planning with minimal perturbation", in *Proceedings of the 10th International Conference on Modeling, Optimization and Simulation (MOSIM 2014)*, 2014, forthcoming. adresse : http://www2.ift.ulaval.ca/~quimper/publications/production_distance.pdf.

[4] M. Mekni, "Hierarchical path planning for situated agents in informed virtual geographic environments", in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, sér. SIMUTools '10, Torremolinos, Malaga, Spain : ICST (Institute for Computer Sciences, Social-Informatics et Telecommunications Engineering), 2010, 30 :1–30 :10, ISBN : 978-963-9799-87-5. DOI : 10.4108/ICST.SIMUTOOLS2010.8811. adresse : http://dx.doi.org.acces.bibl.ulaval.ca/10.4108/ICST.SIMUTOOLS2010.8811.

[5] J. Sun, Y. Meng et G. Tan, "An integer programming approach for the chinese postman problem with time-dependent travel time", *Journal of Combinatorial Optimization*, t. 29, n° 3, p. 565–588, 2015, cited By 0. DOI : 10.1007/s10878-014-9755-8. adresse : http://www.scopus.com/inward/record.url?eid=2-s2.0-84924224093&partnerID=40&md5=05f0a7f60dd738fa11e5625d74b470c9.

[6] A. Richards et J. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming", in *American Control Conference, 2002. Proceedings of the 2002*, t. 3, 2002, 1936–1941 vol.3. DOI : 10.1109/ACC.2002.1023918.

[7] M. Ziegelmann, "Constrained shortest paths and related problems", eng, Doctoral dissertation, Universität des Saarlandes, juil. 2001. adresse : http://scidok.sulb.uni-saarland.de/volltexte/2004/251.

[8] J. C. Sousa, H. A. Biswas, R. Brito et A. Silveira, "A multi objective approach to solve capacitated vehicle routing problems with time windows using mixed integer linear programming", *International Journal of Advanced Science and Technology*, t. 28, p. 1–8, 2011.

[9] P. Ji et K. Chen, "The vehicle routing problem : the case of the hong kong postal service", *Transportation Planning and Technology*, t. 30, n° 2-3, p. 167–182, 2007. doi : 10.1080/03081060701390841. eprint : http://dx.doi.org/10.1080/03081060701390841. adresse : http://dx.doi.org/10.1080/03081060701390841.