

-
- P. Canalda is with the Laboratory of Informatic of Franche-Comté, FRE CNRS 2661, 4 Place Tharradin, 25211 MONTBELLIARD Cédex France. E-mail: Philippe.Canalda@lifc.univ-fcomte.fr.
 - P. Chatonnay is with the Laboratory of Informatic of Franche-Comté, FRE CNRS 2661, 4 Place Tharradin, 25211 MONTBELLIARD Cédex France. E-mail: Pascal.Chatonnay@lifc.univ-fcomte.fr.
 - D. Josselin is with the the CNRS UMR 6012 of Avignon, 74, rue Louis pasteur 84029 AVIGNON France. E-mail Didier.Josselin@univ-avignon.fr.

Manuscript received (November, 30th, 2003).

Enumération d'arbres couvrants tentaculaires, une solution au problème de transport à la demande en convergence.

Philippe Canalda, Pascal Chatonnay, and Didier Josselin.

Abstract—The mono-convergent and on-demand transportation constitutes an offer of public transportation which allows to converge to a particular point of a city (train-station, hospital), at a fixed date, without requiring a priori the building of a dedicated transportation line. Based on the collection of such a public transportation service demand, it is feasible, at a given date, to determine the needs to satisfy the pick-up of users and to deliver them to the destination on time. In order to ensure and to maintain such a service at a minimizing cost, it is necessary to optimize the required needs to be requisitioned.

In this paper we present a system issued from a successful and multi-disciplinaries partnership involving geographic laboratory, informatic laboratory, organizational authority and transportation company. This system is fully experimented and has been plebiscited both by users and comanditors. We propose a method to prepare all the data (geographic and informatic), and to compute the pick-up scenarii allowing to minimize the number of required vehicles and to minimize the number of kilometers globally covered. First of all, our method consists in enumerating the spanning trees of a road network that connects the pick-up points. Secondly, it consists in evaluating the cost of each scenario with respect to the Quality of Service. This method provides a solution for the N-Pick-up and Delivery TSP problem. We invent 2 graph objects, a directed, acyclic, transitive and mono-convergent graph, and a tentaculous spanning tree (TST) in which the branches are only secant at the root. We show an algorithm which solves the optimal computing of TST. A top-down algorithm analyses the transitive and mono-convergent DAG and exhaustively computes the TST. The complexity of the algorithm remains np-complete but it is suitable and pragmatic when applied to the convergent and on-demand transportation problem. Our method has been experimented for two years time now, in a medium-size city (Besançon - France), to stop at the high-speed train station at times when the public transportation network is insufficient to satisfy the users' demand.

Index Terms— Computer-aided transportation system, GIS, On-demand Transportation, Graph, N-TSP, tentaculous spanning trees.



1 INTRODUCTION

Le transport à la demande (TAD) en convergence consiste à prendre en charge un ensemble d'usagers ayant réalisé une demande de transport, et à les acheminer tous en un point unique du territoire, à une heure donnée. Par exemple, les auditeurs d'une conférence, une fois arrivés dans la commune de la conférence, forment un groupe intéressant à desservir avec un TAD en convergence. En effet, en simplifiant, ils résident tous dans des hôtels plus ou moins proches du lieu de la conférence, et ils doivent tous arriver en ce lieu quelques minutes avant le début de la première allocution. Aussi, moyennant le fait qu'ils en aient exprimé le désir au préalable (la veille), il est possible de calculer le ou les itinéraires de un ou plusieurs véhicules convergeant vers le lieu de la conférence et desservant l'ensemble des demandes en réalisant du regroupement (mutualisation) le long d'un itinéraire. L'optimisation réside dans le nombre de véhicules mis en œuvre et le nombre de kilomètres parcourus globalement pour réaliser une convergence.

Les données sont :

- le réseau routier et ses conditions d'utilisation,
- l'ensemble des demandes de transport,
- et le contrat passé avec les usagers en terme de durée maximale de course, d'attente maximale à l'arrêt, et l'heure maximale avant le début de l'événement à laquelle nous devons arriver sur le lieu du dit événement (i.e. nous arrivons 5 minutes au moins avant le début de la conférence).

Le TAD en convergence est un moyen intéressant pour palier ponctuellement aux problèmes des réseaux de transport en commun sur les horaires de frange (très tôt ou très tard) ou sur les zones faiblement desservies. La contrainte principale est que les usagers doivent réserver leur place afin de permettre l'optimisation globale et la mise en place du système (voitures et chauffeurs).

Afin d'être efficace, c'est-à-dire de ne pas coûter trop cher tout en répondant bien aux besoins des usagers, un système de TAD en convergence doit travailler sur des données précises, construites grâce à une méthodologie rigoureuse. Les itinéraires proposés doivent être de qualité en terme de coût, mais également en terme de service aux usagers. C'est-à-dire qu'il ne faut pas, par exemple, donner l'impression à l'utilisateur que l'on ne converge pas directement depuis son lieu de prise en charge vers le point de convergence. En fait, la mise en œuvre du regroupement au sein d'un même véhicule ne doit pas donner l'impression que l'on fait un détour.

Dans cet article nous décrivons la mise en œuvre d'un TAD dans la commune de Besançon (25) en France. Ce TAD a pour objet la desserte de la gare TGV sur des horaires de frange. Ce travail est le fruit d'une collaboration pluridisciplinaire entre les laboratoires de géographie (THEMA) et d'informatique (LIFC) de l'université de Franche-Comté avec le soutien de la compagnie de transport de Besançon (CTB du groupe KEOLIS) et de la ville de Besançon et du CNRS.

Le problème a de multiples facettes. Nous n'aborderons pas ici ni la phase amont sur l'élaboration d'un transport correspondant au besoin des voyageurs, ni la phase plus avale sur les aspects de gestion des usagers (information et réservation) et déploiement des moyens (mise en œuvre

effective des itinéraires par des véhicules). Nous nous intéressons principalement à trois aspects, plus central, du problème :

La méthodologie d'élaboration des données de l'algorithme d'optimisation. C'est-à-dire, d'une part comment à partir du réseau routier nous construisons une matrice point à point des temps et des distances pour environ 550 points géographiques de Besançon. Et d'autre part comment à partir de cette matrice nous identifions et construisons un graphe de convergence vers la gare de Besançon. Ce graphe de convergence s'avère être orienté, acyclique et transitif. C'est-à-dire un DAG (Directed Acyclic graph) transitif.

L'exhibition d'une structure informatique sous-jacente au problème générique du TAD en convergence : les arbres couvrants tentaculaires. Nous montrons comment à partir d'un DAG transitif, qui est le cadre général d'un transport à la demande en convergence, nous construisons toutes les solutions de ramassage. Nous montrons que toutes ces solutions ont des propriétés morphologiques et structurelles. Nous montrons également que tout arbre couvrant du DAG transitif présentant ces propriétés est solution du problème de TAD en convergence.

Les résultats de la mise en œuvre de notre système et une analyse de son fonctionnement après deux ans d'exploitation et une prise en charge qui a dépassé les 5000 voyageurs cette année, et dont une grosse partie est fidélisée. Nous montrons le nombre d'usagers par desserte, le taux de remplissage des véhicules et comment les opérateurs statuent sur les solutions d'optimisation. Nous montrons également que le système a besoin de souplesse pour pouvoir réagir aux aléas de la circulation et aux imprécisions dans les données et/ou dans l'élaboration de ces données.

Dans la première partie nous décrivons le système de transport à la demande tel qu'il est mis en œuvre à Besançon et nous en exhibons les contraintes. La deuxième partie adresse le problème de la préparation des données d'un point de vue méthodologique. Nous montrons comment est construite la matrice d'accessibilité et comment est déduit le graphe de convergence c'est-à-dire le DAG transitif. La troisième partie est dédiée à l'énumération des solutions. Nous montrons comment construire tous les arbres couvrants tentaculaires (ACT) du graphe de convergence. Nous donnons l'algorithme naïf d'énumération des ACT. Dans la quatrième partie nous présentons les résultats de la mise en œuvre de notre solution dans le cadre du transport à la demande Bisontin. Nous terminons en concluant et en donnant des pistes d'extension du système.

2 LE TRANSPORT À LA DEMANDE EN CONVERGENCE À BESANÇON : EVOLIS.

Le transport à la demande mis en place à Besançon est nommé Evolis. Il a pour objet la desserte de la gare de Besançon pour les départs des Trains à Grande Vitesse (TGV) du matin en direction de Paris et pour les retours de ces TGV le soir. Les horaires de ces TGV sont : le matin avant 8h pour les trois départs concernés, le soir après 21h pour les deux retours concernés.

Sur ces créneaux horaires, la compagnie de bus, la CTB, ne dispose pas de lignes régulières avec des fréquences suffisantes pour répondre au besoin des usagers.

Après une enquête dans les trains concernés. Il apparaît que les deux pratiques les plus communes sont :

- le parage de la voiture qui a servi à venir à la gare sur le parking payant jusqu'au soir, cela pour une somme significative ;
- Le recours à un tiers disposant d'un véhicule, souvent le conjoint, pour être déposé à la gare le matin et être repris le soir.

D'autres techniques apparaissent dans les résultats de l'enquête mais de façon beaucoup plus anecdotique, comme le recours au taxi ou bien encore le recours à la marche.

L'enquête fait également apparaître un intérêt important pour un système de transport à la demande, à conditions de respecter un certain nombre de contraintes liées à la durée du transport.

A l'analyse des réponses des usagers, nous en avons déduit un mode de fonctionnement acceptable. La prise en charge des usagers est réalisée aux arrêts de bus. Le système reste un transport en commun, même si il est de nature spécifique. Se restreindre aux arrêts de bus à également l'avantage de limiter le nombre de points de prise en charge par rapport à un système qui réalise des courses porte à porte. Certains quartiers n'étant pas suffisamment doté en arrêt de bus, nous avons créé des arrêts virtuels uniquement dédiés au service évolis. Dans les faits, un usager du service évolis ne devra pas couvrir une distance de plus de 5mn à pied depuis son domicile jusqu'au point de prise en charge évolis le plus proche. Au final, nous travaillons sur une base de 550 lieux de prise en charge, arrêts virtuels compris.

Un usager qui demande à accéder au service évolis reçoit, lors de sa demande, un lieu et une heure de prise en charge. Il ne doit pas attendre plus de 3 minutes à l'arrêt. C'est-à-dire que les véhicules réalisant le service passent aux arrêts au maximum trois minutes après l'heure de prise en charge définie.

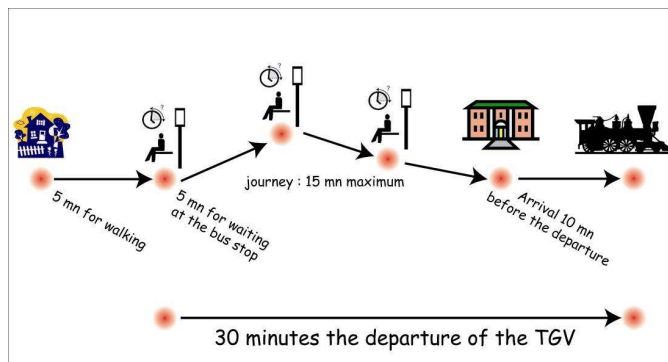
Les usagers sont déposés à la gare au moins 10 minutes avant le départ de leur train. Ces 10 minutes sont une sécurité qui a été demandée par les utilisateurs. Cela permet aussi aux personnes n'ayant pas encore leurs billets de se les procurer.

Le trajet en véhicule dure au maximum 20 minutes. Pour la commune de Besançon, la gare ayant une position relativement centrale, cette durée permet de desservir tous les quartiers de la ville.

Globalement, un usager quitte son domicile au maximum 35 minutes avant le départ de son train. Il se rend à son point de prise en charge, un arrêt de bus réel ou virtuel. Il attend au maximum 3 minutes après l'heure prévue de prise en charge. Il monte dans un véhicule et voyage pendant une durée maximum de 20 minutes. Le véhicule prend éventuellement en charge d'autres usagers durant le trajet. Le véhicule arrive devant la gare au moins 10 minutes avant le départ du train.

Pour pouvoir être pris en charge un usager doit téléphoner au plus tard la veille de son départ au service

évolis. Lors de cet appel il indique son lieu de résidence, la date et l'heure de son départ. En réponse il lui est donné un lieu et une heure de prise en charge. Le lieu de prise en charge peut être négocié avec l'opérateur évolis. En fonction de la connaissance du réseau de bus de l'utilisateur et de ses habitudes, il est possible de lui proposer un autre arrêt que celui qui est considéré comme le plus proche de l'adresse qu'il a indiquée. Si l'utilisateur revient avec un TGV du soir il lui également proposé de réserver son retour grâce au service évolis.



La veille du départ vers 18 heures les réservations sont considérées terminées. C'est à ce moment qu'est réalisé le calcul des feuilles de route des véhicules devant assurer les trajets du lendemain, matin et soir. Nous montrons dans la suite, sur quelles données s'appuie le calcul, comment ces données sont construites et comment elles sont utilisées.

3 PRÉPARATION DES DONNÉES

Les données nécessaires pour réaliser le calcul des feuilles de route des véhicules sont :

- un graphe ou une matrice de convergence évalué en temps et en distance,
- la liste des demandes de service avec les points de prise en charge,
- la fonction de coût des véhicules réalisant le service.

La fonction de coût est relativement simple elle consiste en une somme forfaitaire pour chaque véhicule mis en service puis en une somme proportionnelle au nombre de kilomètres globalement parcourus. Nous revenons sur ce point dans la dernière partie de l'article. Ici nous allons nous concentrer sur l'élaboration des graphes d'un point de vue qualitatif et méthodologique.

Les données concernant le réseau urbain sont connues dans un système d'information géographique (SIG). Le réseau routier est représenté sous forme de tronçon, où un tronçon est une section de voie sans intersection. Pour chacun des tronçons du réseau, de nombreuses informations sont connues, entre autre la longueur, les coordonnées de l'origine et de la destination, la forme, le dénivelé.

Grâce au SIG nous pouvons aisément construire le graphe d'accessibilité point à point. En fixant le point de convergence, nous déterminons le vecteur des heures de départ en chaque sommet du graphe puis le graphe orienté acyclique (DAG) qui est le graphe de convergence. Nous montrons ensuite quelques une des propriétés de ce DAG.

3.1 Le graphe d'accessibilité point à point

Soit un tronçon du réseau, nous pouvons déterminer le

temps théorique nécessaire pour le parcourir. Le Temps Théorique par Tronçon (TTT) est égal à la longueur de se tronçon multiplié par la vitesse maximale autorisée sur ce tronçon :

$$TTT = \text{longueur tronçon} * \text{vitesse maximale.}$$

Nous pouvons ensuite déterminer la rugosité de chaque tronçon. Dans un premier temps la rugosité physique (R_p) due aux paramètres physiques du tronçon tels que la sinuosité, la pente, la qualité de l'enrobé. Puis la rugosité d'usage (R_u) due à l'intensité d'usage de ce tronçon en fonction du créneau horaire d'utilisation. De ce fait nous pouvons calculer le temps théorique de parcours (TTP) d'un point à un autre dans le graphe. Nous notons M_{tp} la matrice des temps théoriques de parcours.

$$M_{tp} = M_{ttt} + M_{rp} + M_{ru}$$

Avec

- M_{ttt} la somme, le long des plus courts chemins, des temps théoriques par tronçon ;
- M_{rp} la somme, le long des plus courts chemins, des temps dus à la rugosité physique ;
- M_{ru} la somme, le long des plus courts chemins, des temps dus à la rugosité d'usage.

Les données utilisées pour déterminer M_{tp} , bien qu'elles soient de bonne qualité, ne permettent pas de construire une représentation exacte de la réalité. Lors de nos expérimentations il nous est arrivé d'observer une dérive importante sur certains parcours. Aussi pour palier à ce problème nous ajoutons un quatrième terme à la somme construisant M_{tp} . Ce dernier terme, noté M_{δ} , permet d'introduire une correction reflétant la réalité observée. Si les trois termes M_{ttt} , M_{rp} et M_{ru} sont positifs et représentent des temps, le dernier terme M_{δ} pourra être positif, nul ou négatif en fonction des écarts observés. Mais dans tous les cas chaque élément de M_{tp} sera un nombre positif représentant un temps de parcours.

$$M_{tp} = M_{ttt} + M_{rp} + M_{ru} + M_{\delta}$$

M_{tp} est la matrice d'incidence du graphe d'accessibilité point à point (GAPP). Dans notre cas, chaque point représente un lieu de prise en charge et chacun de ces points peut être atteint par n'importe lequel des autres points. GAPP est un graphe orienté complet.

3.2 Introduction du point de convergence

L'introduction du point de convergence permet de déterminer les heures de départ en chacun des arrêts. En effet,

- soit H l'heure de l'événement en C , le point de convergence ;
- soit $Offset$ une valeur décrivant combien de minutes avant l'événement les véhicules doivent être en C ;
- soit $Coeff$ un coefficient de dilatation décrivant la capacité à faire des détours le long d'un itinéraire (soit pour assurer des prises en charge, soit par mesure de sécurité).

Alors l'heure de prise en charge en un point P (HdP) est :

$$HdP = H - (M_{tp}(p \text{ vers } c) * Coeff) - Offset$$

Une fois les heures de départ définies pour chacun des arrêts il est facile de construire le graphe de convergence. Le graphe de convergence ne conserve que les arrêtes utilisables en terme de durée pour converger vers le point désigné. De plus le graphe de convergence est sans cycle et il possède la propriété de transitivité. Deux arrêtes sont connectés si le temps de trajet pour aller de l'un à l'autre est inférieur à la durée séparant leurs deux dates de départ. Autrement dit, une arrête existe si l'on dispose du temps matériel pour l'utiliser. Soit deux sommets $Orig$ et $Dest$, ils sont connectés si :

$$Hd_{dest} - Hd_{orig} > M_{tp}(Orig \text{ vers } Dest)$$

Le graphe de convergence est transitif. C'est-à-dire que si A et B sont connectés et B et C sont connectés alors A et C sont connectés. Cela se montre de la façon suivante :

$$A \text{ et } B \text{ sont connectés s'écrit : } Hd_B - Hd_A > M_{tp}(A \text{ vers } B)$$

$$B \text{ et } C \text{ sont connectés s'écrit : } Hd_C - Hd_B > M_{tp}(B \text{ vers } C)$$

En réalisant la somme des deux expressions

$$Hd_C - Hd_B + Hd_B - Hd_A > M_{tp}(A \text{ vers } B) + M_{tp}(B \text{ vers } C)$$

Or $M_{tp}(A \text{ vers } B) + M_{tp}(B \text{ vers } C)$ est un majorant de $M_{tp}(A \text{ vers } C)$ en effet il est possible de passer par B pour aller de A en C , mais il est également possible de trouver un chemin plus court. Donc en simplifiant on trouve :

$$Hd_C - Hd_A > M_{tp}(A \text{ vers } B) + M_{tp}(B \text{ vers } C) \geq M_{tp}(A \text{ vers } C)$$

Ce qui est bien l'expression de A est connecté à C . cela prouve la transitivité.

Le graphe de convergence est sans cycle. Un graphe sans cycle signifie qu'il n'existe pas de chemin permettant de passer deux fois par un même sommet. Donc il faut prouver que quelque soit A et B s'il existe un chemin de A vers B alors il n'existe pas de chemin de B vers A . En fait grâce à la propriété de transitivité, si il existe un chemin de A vers B alors il existe un arc de A vers B . Supposons qu'il existe un chemin de B vers A donc par transitivité qu'il existe un arc de B vers A , alors cela s'écrit :

$$A \text{ et } B \text{ sont connectés s'écrit : } Hd_B - Hd_A > M_{tp}(A \text{ vers } B)$$

$$B \text{ et } A \text{ sont connectés s'écrit : } Hd_A - Hd_B > M_{tp}(B \text{ vers } A)$$

Par construction nous savons que toutes les valeurs de M_{tp} sont positives. Donc $Hd_B - Hd_A > 0$ donc la valeur opposée $Hd_A - Hd_B < 0$. Cela contredit le fait que cette valeur soit supérieure à un élément de M_{tp} qui, lui, est supérieur à 0. Donc, ces deux inégalités ne peuvent être vérifiées simultanément. Donc le graphe de convergence est sans cycle.

Illustration 1. Considérons un réseau urbain de 10 points de ramassage (pdr) numérotés de 0 à c , et d'un point de convergence c pris parmi ces 10 pdr .

Considérons la matrice point à point des temps de trajet de ces 10 pdr , ainsi que le vecteur O des temps de

déplacement depuis les points de ramassage vers le point de convergence c. Alors, nous pouvons construire le graphe de convergence des pdrs vers le point de convergence c. Notons maintenant que si nous considérons une liste de demande de ramassage constituée d'une liste de demandes de ramassage, avec une demande qui est un 3-uplet (pdr, date, pdc) où pdr est l'identifiant du point de ramassage, date est l'horaire de ramassage en ce pdr pour acheminer le passager en un point de convergence pdc, alors nous pouvons construire le graphe de convergence G de la demande de ramassage.

$G=(S,E)$ est un DAG-transitif avec S, l'ensemble des sommets du graphe, et E l'ensemble des arcs (S_i,S_j) reliant des sommets de S :

$$S=\{2,3,4,5,c\}$$

$$E=\{(4,2);(2,c);(4,c); (5,2);(5,3);(3,c);(5,c),(2,3)\} \quad \square$$

Le graphe de convergence possède deux autres propriétés morphologiques intéressantes à souligner. D'une part le point de convergence reçoit un arc de chacun des sommets hormis lui-même. En effet, il est possible de venir à la gare depuis tous les arrêts. D'autre part, le point de convergence n'émet aucun arc. En effet, sinon il y aurait un cycle.

En nous appuyant sur le Graphe orienté acyclique transitif que nous venons de bâtir, nous allons maintenant montrer comment construire des dessertes et choisir les meilleures d'entre elles.

3.3 Les arbres couvrants tentaculaires

Afin de définir un arbre couvrant tentaculaire, nous introduisons les définitions nécessaires à la compréhension des algorithmes de calculs des arbres couvrants tentaculaires.

Nommons **chemin terminant** ct d'un graphe G, un chemin pour lequel il n'existe pas d'arc de G qui prolonge ct.

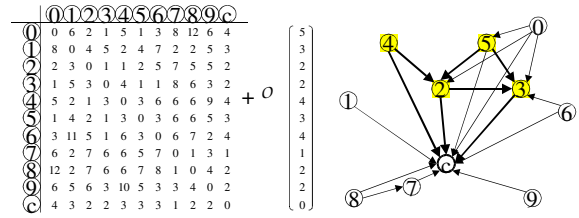
Nommons **chemin maximal** cm depuis un sommet s d'un graphe, le plus long chemin de ce graphe depuis ce sommet.

Définition 1. Un graphe G est convergent en un point O si, quelque soit le sommet s de $G / \{O\}$ alors tout chemin terminant depuis s se termine en O.

Définition 2. Un arbre couvrant tentaculaire (ACT) est un arbre couvrant dont tous les chemins terminants sont arcs-disjoints deux-à-deux et sommets disjoints deux à deux (hormis à la racine).

Propriété 1. Dans un graphe de convergence Gc, tout ACT de Gc est un recouvrement de Gc constitué de chemins terminaux arcs-disjoints deux-à-deux et sommets-disjoints deux-à-deux hormis le point de convergence, et réciproquement.

Preuve. On montre l'équivalence entre les points i) et ii) suivants : i) les chemins terminaux d'un ACT sont arcs-disjoints deux-à-deux ; ii) les chemins terminaux d'un ACT sont deux-à-deux sommets-joints au point de convergence (i.e. à la racine). Ensuite on montre qu'un ACT se décompose en une union de chemins maximaux



de cet ACT. Cette décomposition constitue un recouvrement. \square

4. Enumération de tous les arbres couvrants tentaculaires (ACT) d'un graphe de convergence Gc en 0.

Il s'agit de résoudre le problème complexe du N-TSP appliqué au transport à-la-demande par rabattement monopolaire sans fenêtre de temps. Cela revient à calculer la solution qui minimise le nombre N de véhicules nécessaires et le coût global des courses des véhicules tout en satisfaisant des contraintes (service, QoS, flotte disponible, demande).

Afin de réduire la complexité pratique de la résolution de ce problème, il s'agit tout d'abord de calculer statiquement tout ce qui peut l'être pour réduire la complexité dynamique du traitement. Pour réduire la complexité dynamique du traitement, il s'agit de déterminer des bornes sur le nombre N de véhicules nécessaires :

[les éléments minimaux du graphe de la demande, le nombre de points de demande de transport]

En reprenant l'exemple de l'illustration n°1, nous montrons dans la suite qu'il suffit de résoudre le N-TSP pour des valeurs de N compris entre 2 véhicules et 4 points de ramassage.

Une solution "naïve" à ce problème consiste à calculer tout d'abord tous les chemins terminaux (en tout point) du graphe Gc convergent en O (cf. algorithme 1) puis à calculer tous les recouvrements de Gc constitués de chemins terminaux arcs et sommets disjoints 2-à-2 hormis au point de convergence (cf. algorithme 2).

4.1 Algorithme 1 : Enumération de tous les chemins terminaux du graphe Gc convergent en O

Considérons le graphe convergent G de la demande de transport de la figure précédente. Soit donc $G = \{S=\{c, 2, 3, 4, 5\}, E=\{(4,c), (4,2), (5,c), (5,2), (5,3), (2,c), (2,3), (3,c)\}\}$.

Initialisation : $G_c = G$; CheminsTerminaux = {} ; CheminsPartiels = {} ;

Etape 1 = calcul de l'ensemble $EltMin(G_c) = \{4,5\}$ qui contient toutes les éléments minimaux de Gc, c'est-à-dire les nœuds de G n'ayant pas de nœuds précesseurs.

Lemme 1 : le nombre d'éléments minimaux d'un graphe de convergence constitue une borne minimale du nombre de branches de tout ACT de Gc.

Ces éléments minimaux sont aussi des feuilles de Gc. Un

élément minimal de G_c est un nœud de G_c qui n'est pas un nœud sortant d'aucun arc de G_c .

Lemme 2 : si l'ACT d'un graphe G_c est composé de chemins maximaux 2-à-2 arcs et sommets disjoints alors le nombre de feuilles de cet ACT est égal à son nombre de branches.

Preuves. Cf. [CCLJ'03] \square

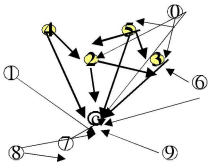
Étape 2 = pour chacun des nœuds de Feuille(G_c), on calcule les chemins partiels et terminaux depuis ce nœud et aussi ceux qui passent par ce nœud. Depuis ce nœud désigne que le sommet traité est une feuille du graphe G_c . Par ce nœud désigne que le sommet traité n'est pas une feuille du graphe initial G mais seulement du sous-graphe G_c de G .

Itération 1 : $G_c = G$; $\text{EltMin}(G_c) = \{4,5\}$;

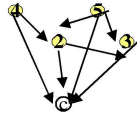
Pour 4 : CheminsPartiels += {4->2};

CheminsTerminaux += {4->c}

5 : CheminsPartiels += {5->2,5->3};

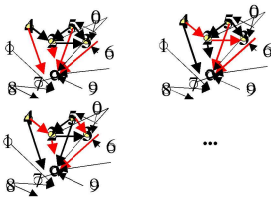


Dag transifif convergent complet



Dag transifif convergent de la demande de transport

Quelques exemples d'ACT



CheminsTerminaux += {5->c}

On commence par compléter les chemins partiels dont l'extrémité terminale est le sommet 4. Pour tous ces chemins on va construire autant de chemins partiels ou terminaux qu'il y a d'arc sortants de ce sommet. Mais ici l'ensemble des chemins partiels est vide alors on va procéder au calcul des chemins partiels et terminaux depuis ces nœuds. Pour le sommet 4 cela donne {4->2, 4->c}. Et CheminsPartiels += {4->2}; CheminsTerminaux += {4->c}. On retire alors 4 du graphe G_c et tous les arcs qui en partent. Pour le sommet 5 cela donne {5->2, 5->3, 5->c}. CheminsPartiels += {5->2, 5->3}; CheminsTerminaux += {5->c}. Et on retire le sommet 5 de G_c et tous les arcs de G qui partent de 5.

Ainsi à l'**itération 2** nous avons : $G_c = G - \text{EltMin}(G_c)$,

Soit $G_c = \{S, E\}$ avec $S = \{c, 2, 3\}$, $E = \{(2, c), (3, c)\}$;

$\text{EltMin}(G_c) = \{2\}$;

Pour 2 : CheminsPartiels qui passent par 2 += {4->2->3, 5->2->3};

CheminsPartiels qui partent de 2 += {2->3};

CheminsTerminaux qui pa(rt/ss)ent (de/par) 2 += {4->2->c, 5->2->c, 2->c}

On réitère une 3^{ème} fois l'étape 1 sur G_c sauf si $G_c = \{\text{point de convergence}\}$.

A la 3^{ème} **itération** des étapes 1 et 2 cela donne :

$G_c = G - \text{EltMin}(G_c) = \{S = \{c, 3\}, E = \{(3, c)\}\}$; $\text{EltMin}(G_c) = \{3\}$;

Pour 3 : CheminsTerminaux qui pa(rt/ss)ent (de/par) 3 += {5->3->c, 4->2->3->c, 5->2->3->c, 2->3->c, 3->c}

CheminsPartiels qui passent par 3 = {};

CheminsPartiels qui partent de 3 = {};

A la 4^{ème} **itération**, on s'arrête car $G_c = G - \text{EltMin}(G_c) = \{c\}$.

4.2 Algorithme 2 : Calcul de tous les recouvrements tentaculaires de G construits par union des chemins terminaux 2 à 2 arcs-disjoints.

Cet algorithme est assez simple lorsque l'on construit tous les recouvrements tentaculaires RT_i de G tq $RT_i = \{CT(G)_{i,1}, \dots, CT(G)_{i,j}\}$ avec les $CT(G)_{i,l}$ et $CT(G)_{i,k}$ tels que l et k in $[1, j]$, $l < k$, $S(CT(G)_{i,l}) \text{ inter } S(CT(G)_{i,k}) = \{O\}$ et $E(CT(G)_{i,l}) \text{ inter } E(CT(G)_{i,k}) = \{\}$. C'est dire que les chemins terminaux qui ont des arcs disjoints 2 à 2 et qui ont des sommets disjoints 2 à 2 hormis le point de convergence.

Calcul de complexité : Il faut parcourir la liste des chemins terminaux. Pour chacun de ces chemins, il faut agréger d'autres chemins qui respectent les contraintes de disjonction d'arêtes et de sommets (hormis le point de convergence). Si n est le nombre de chemins terminaux calculés alors la complexité est de $n!$ dans le pire cas. Le nombre de recouvrements de convergence peut atteindre $2^{(n-1)} * n!$

Sur l'exemple précédent, le nombre de chemins terminaux (CT) est de 10. Quand on constitue le recouvrement, pour chaque CT on multiplie par 2 le nombre de ACT, et ainsi de suite en progressant dans la liste des CT => $10! * 2^9 = c'$ est énorme !

Pour faire tomber cette complexité, on peut d'une part intégrer l'algorithme 2 au sein de l'algorithme 1 et ainsi réaliser les 2 traitements en une seule passe. D'autre part, on peut tenir compte d'un ordonnancement des nœuds du graphe de convergence.

4.3 Calcul de tous les ACT de G en une passe.

Reprenons l'exemple qui illustre les algorithmes précédents.

Initialisation : $G_c = G$; CheminsTerminaux = {} ;

CheminsPartiels = {} ;

Une 1ère amélioration consiste à calculer l'ordre total suivant sur les sommets de G_c , d'une part l'ordre partiel de dépendance et puis d'autre part sur l'ordre (total) décroissant des offset au point de convergence.

Étape 1 : calcul de l'ordre total sur les sommets de G . Cela consiste à

- initialiser $OT(G) = \{\}$ et $G_c = G$;
- déterminer les éléments maximaux de G_c . On trie ensuite les éléments de cet ensemble selon l'ordre décroissant des offset et on ajoute ces éléments triés à $OT(G) += OT(G_c)$;
- calculer $G_c = G_c - \text{EltMin}(G_c)$;
- réitérer l'étape b) jusqu'à ce que G_c soit vide.

On obtient $OT(G) = \{4,5,2,3,c\}$

Étape 2 : calcul de l'ensemble ACTPartiels = {} ; Pour chacun des éléments de $OT(G)$ on calcule l'ensemble partiel des ACT.

Itération 1 Pour 4, autant d'ACT que d'arcs qui en sortent :
ACTPartiels={{4->c},{4->2}}

Lors des itérations suivantes, l'étape 2 se sophistique.
L'ensemble des ACT partiels est calculé en déterminant l'ensemble des arcs sortants de s, puis en construisant autant d'ACT partiels qu'il y a de combinaisons de compléter les ACT partiels avec les arcs qui sortent de s.

Itération 2 Pour 5, les arcs qui en sortent sont 5->2, 5->3 et 5->c. Les ACTPartiels sont arêtes et sommets disjoints 2-à-2 sauf en c : ACTPartiels={{4->c,5->2}, {4->c,5->3}, {4->c, 5->c}, {4->2,5->2} non, {4->2,5->3}, {4->2, 5->c}}

Itération 3 Pour 2, les arcs qui en sortent sont 2->c et 2->3 ;
ACTPartiels={{4->c,5->2->c},{4->c,5->2->3},{4->c,5->3,2->c}, {4->c,5->3,2->3}non, {4->c, 5->c, 2->c}, {4->c, 5->c,2->3}, {4->2->c,5->3}, {4->2->3,5->3}non, {4->2->c, 5->c}, {4->2->3, 5->c}}

A cette itération, 2 combinaisons de chemins partiels ont été réfutées car elles ne respectaient pas la contrainte de sommets disjoints 2-à-2 (en dehors du point de convergence) : {{(4,c),(5,3),(2,3)}, ((4,2)(2,3),(5,3))}.

Itération 4 Pour 3, les arcs qui en sortent sont 3->c ;
ACTPartiels={{4->c,5->2->c,3->c},{4->c,5->2->3->c},{4->c,5->3->c,2->c},{4->c,5->c,2->c,3->c},{4->c, 5->c,2->3->c},{4->2->c,5->3->c},{4->2->c, 5->c,3->c},{4->2->3->c, 5->c}}

Itération 5 Fin car c est un point de convergence.

4 MISE EN ŒUVRE EVOLIS

Les **Transports à la Demande (TAD)** sont *des transports collectifs terrestres de personnes activés seulement à la demande*. Leur champ d'application est très large, de systèmes empiriques à large portée (de type « taxi-brousse » dans les pays en voie de développement) aux « shuttles » présents aux Etats-Unis autour des grandes villes et d'aéroports, intégrant des Technologies de l'Information (GPS et/ou logiciel d'optimisation de trajets).

Le TAD **Evolis-gare**, et son logiciel d'exploitation **RESAD2** conçu et mis en oeuvre pour la desserte de la gare de **Besançon** aux heures de frange, constitue dans ce contexte une expérience innovante alliant marketing, recherche et technologie.

Evolis-gare est le fruit d'une réflexion menée conjointement par l'équipe marketing de la **Compagnie des Transports de Besançon** (société de transport de Besançon, filiale de **Kéolis**, 35000 personnes en Europe, leader des transports terrestres de personnes en France, filiale de la SNCF), les informaticiens du **Laboratoire d'Informatique de Franche-Comté** (FRE CNRS 2661) et les géographes des laboratoires **THEMA** (UMR 6049) et **ESPACE** (UMR 6012) du CNRS. Le projet a été réalisé dans le cadre :

- du **GDR Systèmes d'Information Géographique Modélisation et Application (SIGMA)** n° 2340 du CNRS du Département Sciences et Techniques de l'Information et de la Communication (STIC),
- des programmes de recherche **PREDIT** et de l'**Institut des Sciences et des Technologies de l'Information de Franche Comté** (Contrat de Plan Etat-Région).

Lancé au mois d'octobre 2000, ce service, expérimental pendant 4 mois, est maintenant désigné comme un **produit commercial au service de plus de 4000 clients** (fin 2003).

La gestion du service est assurée par un logiciel d'aide à l'exploitation, baptisé **RESAD2**¹ et spécialement développé à cet effet. Ce logiciel comporte deux modules, l'un pour enregistrer les réservations des clients, l'autre, pour optimiser les dessertes en fonction de la demande. Le module de création de lignes virtuelles propose une série d'alternatives de desserte en fonction des véhicules disponibles et de leur capacité. Parmi ces différentes solutions, équivalentes d'un point de vue économique, le technicien choisira celle qui lui paraît la plus conforme aux pratiques de la société de transport.

En 2001, le service **Evolis-gare** a effectué 2 055 voyages correspondant à 10777 km. Il est viable économiquement (ratio Recettes / Dépenses d'environ 35 %), grâce à un bon taux de regroupement (moyenne de 1,8 voyageurs par course). Il a par ailleurs permis de capter de nouveaux clients ayant effectivement délaissé leurs véhicules personnels au profit de ce « transport collectif individualisé ». **Evolis-gare** et **RESAD2** ont su s'adapter aux besoins d'une clientèle exigeante, à des impératifs économiques et à des objectifs scientifiques d'organisation de la desserte de la gare TGV sous contraintes temporelles et spatiales.

Service de transport innovant dédié à un segment de marché particulier, **Evolis-gare** a été lauréat du concours du développement local de Franche-Comté 2001 et du prix **Mercurie d'Or 2001** du groupe **Kéolis**. Pour l'anecdote, le concept a été présenté à deux reprises dans l'émission « Du Nouveau » sur TF1, spot sponsorisé par le Gan, 20h50, les 7 et 22 octobre 2002, dans une rubrique « du nouveau dans le déplacement ».

5 CONCLUSION

Dans cet article nous avons présenté l'algorithme complet d'énumération des arbres couvrant tentaculaire d'un DAG transitif. C'est-à-dire nous donnons la solution générique du problème de TAD en convergence. Cette approche fait l'objet d'un dépôt de brevet conjoint entre l'université de Franche-Comté, Le CNRS, Le groupe **KEOLIS** et la communauté de communes du grand Besançon (**CAGB**).

Cependant, même si cette solution ouvre la voie à une détermination optimale des dessertes en convergence, un certains nombres de travaux peuvent et doivent être entrepris afin d'une part de traiter le passage à l'échelle, et d'autre part résoudre des problèmes encore plus généraux que celui de dessertes mono-pôlaires.

Le passage à l'échelle requière des adaptations graduelles. Nous prônons tout d'abord un algorithme de programmation dynamique qui réutilise les calculs

¹ *Réservation d'un Service à la Demande et Système d'Aide à la Décision*

communs en permettant et un temps de calcul moindre et une économie d'espace mémoire. Nous prôtons ensuite le calcul d'un sous-ensemble des meilleures solution à la manière du Tabu-Search ou bien encore d'algorithmes génétiques.

La résolution du N-TSP pour le rabattement monopolaire avec fenêtre de temps permettra d'améliorer le taux de remplissage des véhicules. Par contre la fenêtre de temps introduit des cycles dans le DAG-transitif.

Et la résolution du N-TSP pour le rabattement MULTIPOLAIRE sans ou AVEC fenêtre de temps permettra de traiter le cas des pôles d'échange tout en optimisant le remplissage des véhicules. L'ACT devient alors une forêt d'ACT.

Les travaux de recherche des équipes du CNRS (géographes) et de l'Université de Franche-Comté (informaticiens) dans le domaine des TAD en convergence ont produit des avancées significatives (méthodes de traitement des matrices Origine-Destination, graphes multiscalaires pour la gestion d'objets mobiles et des nouveaux systèmes de transports flexibles. Ils constituent une première étape vers des Transports à la Demande « temps réels » plus en phase avec les nouvelles technologies de l'information et les pratiques de mobilité et de consommation des usagers des transports.

Le CNRS, en collaboration avec l'Université de Franche-Comté, le Groupe Kéolis et la Communauté d'Agglomération du Grand Besançon dépose actuellement le logiciel RESAD2.

REMERCIEMENTS

Les auteurs remercient l'ensemble des collaborateurs et partenaires impliqués dans la mise en œuvre d'Evolis et son exploitation.

REFERENCES

- [1] AGARWAL Y., MATHUR K., SALKIN M., "Set Partitioning Approach to Vehicle Routing." *Networks* 7, pp.731-749, 1989.
- [2] BOLOT J., JOSSELIN D., THEVENIN T. (avril 2002), "Responsive Demand Transports in the mobilities and technologies evolution: context, concrete experience and perspectives" Agile conference, Majorca.
- [3] CANALDA P., CHATONNAY P., JOSSELIN D., "Optimizing the convergent and on-demand transportation. An enumerating algorithm of tentaculous spanning trees.", Abstracts book of the the ECCO XVI International Conference, The European Chapter on Combinatorial Optimisation, 5 - 7 June 2003, Molde University College, Norway, page 13, session Graphs.
- [4] CANALDA P., CHATONNAY P., JOSSELIN D., " Résolution du problème de ramassage à-la-demande par rabattement fondé sur le calcul d'arbres couvrants tentaculaires : application au transport à-la-demande multimodal.", Rapport de Recherche du Laboratoire Informatique de Franche-Comté (FRE CNRS 2661), à paraître, début 2004.
- [5] CHALASANI P., MOTWANI R., "Approximating Capacitated

routing and Delivery Problems.", *SIAM Journal on Computing*, Vol. 28 (6), pp. 2133-2149, 1999.

- [6] CHRISTOFIDES N., MINGOZZI A., TOTTH P., "Exact Algorithms for Solving the Vehicle Routing Problem Based on Spanning Trees and Shortest Path Relaxations", *Mathematical Programming* 20, pp. 255-282, 1981.
- [7] COLLETTE Yann, SIARRY Patrick, "Optimisation multiobjectif", Editions Eyrolles, 2002, Coll. Algorithmes ISBN 2-212-11168-1.
- [8] DESROCHERS M., DESROSIERS J, SOLOMON M., "A new optimization algorithm for the vehicle routing problem with time windows", *Operations Research* 40 (2), pp. 342-354.
- [9] C. Lang, M. Tréhel, and P. Baptiste, "A distributed placement algorithm based on process initiative and on a limited travel". In *Parallel and Distributed Applications and Activities: PDPTA'99*, volume 5, pages 2636-2641, Las Vegas (USA), 28 juin - 1 juillet 1999.
- [10] GENDREAU M., Hertz A., LAPORTE G., "A tabu search heuristic for the vehicle routing problem.", *Management Science* 40 (10), pp. 1276-1290.
- [11] HORN M., "Fleet scheduling and dispatching for demand-responsive passenger services", *Transportation Research Part C* 10, 2002, pp. 35-63.
- [12] THEVENIN T., "Quand l'information géographique se met au service des transports publics : une approche spatio-temporelle appliquée à l'agglomération Bisontine", Thèse, 260 p, 2002.