European Journal of Operational Research 219 (2012) 738–750

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

# Multi-period traffic routing in satellite networks ☆

Ioannis Gamvros [a], S. Raghavan [b,*]

[a] IBM, 4400 North First Street, San Jose, CA 95134, United States

[b] The Robert H. Smith School of Business and Institute for Systems Research, University of Maryland, College Park, MD 20742-1815, United States

A R T I C L E  I N F O

*Article history:*
Available online 11 November 2011

*Keywords:*
OR in telecommunications
Routing
Integer programming
Large scale optimization
Branch-and-price-and-cut

A B S T R A C T

We introduce a traffic routing problem over an extended planning horizon that appears in geosynchronous satellite networks. Unlike terrestrial (e.g., fiber optic) networks, routing on a satellite network is not transparent to the customers. As a result, a route change is associated with significant monetary penalties that are usually in the form of discounts (up to 40%) offered by the satellite provider to the customer that is affected. The notion of these rerouting penalties requires the network planners to explicitly consider these penalties in their routing decisions over multiple time periods and introduces novel challenges that have not been considered previously in the literature. We develop a branch-and-price-and-cut procedure to solve this problem and describe an algorithm for the associated pricing problem. Our computational work demonstrates that the use of a multi-period optimization procedure as opposed to a myopic period-by-period approach can result in cost reductions up to 13% depending on problem characteristics and network size considered. These cost reductions correspond to potential savings of several hundred million dollars for large satellite providers.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

Satellite communications forms a large part of the telecommunications industry. The Satellite Industry Association reports (SIA, 2010) that the commercial satellite industry generated $160.9 billion in revenues in 2009, of which $93 billion or 58% is attributable to the satellite services sector. Satellite communication providers operate large fleets of satellites and are able to provide a multitude of different services to retail customers, government agencies, and companies in geographically diverse locations throughout the world. Some of the products that companies in the satellite industry currently offer include temporary and permanent video connections that usually carry traffic for cable and television networks, internet trunking services that are used by internet service providers (ISPs), telecom carriers, global enterprises, government agencies, and the military to connect remote locations to existing high-speed backbones (e.g., in the United States or Europe).

Nowadays, companies in the satellite industry usually operate satellites in geosynchronous (GEO) orbit at different longitudes that remain over the same regions of the earth constantly. Typically a GEO satellite, depending on the complexity of the on-board equipment and its size, will cost hundreds of millions of dollars to design and launch into orbit. Moreover, these satellites have an expected life-span of approximately 15 years. Therefore, satellite companies are under constant pressure to generate as much revenue as possible and utilize their capacity with the utmost efficiency.

In general, a satellite provider will receive service requests from customers that wish to transmit a specific amount of traffic (or lease a certain amount of bandwidth) between two locations. The provider will then have to route this request over a satellite that has available capacity and is directly visible from both locations. Satellites usually have multiple antennas (or equivalently beams) that can either receive or transmit (or both) telecommunications signals from and to earth, respectively (for a nice introduction to satellite technology see Maral and Bousquet, 1998). These beams can communicate with specific regions of the world that are visible from orbit and depend on the satellite's design. Fig. 1 presents a typical situation for a GEO satellite (positioned over the Atlantic ocean) with a characteristic beam layout. In the industry lingo beams that receive communications from the ground are called up-beams while those that transmit signals back are called down-beams. Also, it is important to note that on-board the satellite there is a specific, number of connections (i.e., transponders) between up-beams and down-beams. The transponders receive signals from the up-beam to which they are connected and after processing them they transmit them towards the earth through the down-beam. Each transponder has a specific bandwidth and processing characteristics which make it suitable for certain types of traffic. For example high-definition video broadcasting requires the use of transponders with enough capacity and transmitting power, while voice trunks can be allocated to transponders with relatively limited power. As a result, in order to connect two

☆ This paper was awarded the EURO Management Science Strategic Innovation Prize 2010.

* Corresponding author.
*E-mail addresses:* igamvros@us.ibm.com (I. Gamvros), raghavan@umd.edu (S. Raghavan).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.11.004

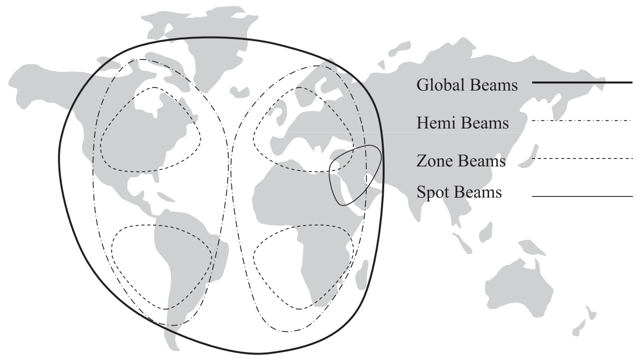


**Fig. 1.** Typical beam footprint for a GEO satellite over the Atlantic ocean.

distinct locations requested by a customer the satellite provider must decide on the satellite and more importantly the up-beam, down-beam pair, connected with an appropriate transponder, that will handle the request. In Fig. 1 for example, in order to connect Europe to North America one could use the eastern-hemi beam together with the western-hemi beam, or alternatively the north-eastern-zone beam together with the north-western-zone beam provided that these beams are connected with transponders on-board the satellite. Furthermore, in some satellites, network operators can change the connectivity between the up-beams and down-beams. More precisely, for these satellites network planners can choose amongst a limited set of given configuration possibilities.

Up to this point one could deduce, and rightfully so, that routing service requests over a satellite network presents similar challenges to those that terrestrial carriers face when trying to utilize their networks more efficiently. In the terrestrial case a carrier could optimize the utilization of the network by finding the best possible routing solution for all its customers and since routing in such networks is transparent to the customers (i.e., the customers are unaware of the route their traffic follows) this solution could be readily implemented. On-board configuration decisions in a terrestrial context could resemble leasing of additional capacity or installing new facilities in the long term. Notice that the frequency of routing reconfigurations depends on the underlying technology and the network layer at which they are implemented but, in general, can take place in real-time. However, in satellite networks rerouting a customer, has a very direct impact on the customer. This occurs because in most cases it is the customers who own the equipment (satellite dishes) that point to specific locations in the sky and communicate with predetermined satellites. Rerouting a customer to a different satellite would entail a discontinuation of service until the satellite dish can be re-pointed to the new satellite location and the channel parameters would have to be reconfigured. As a result, service contracts for satellite services define financial penalties for the satellite provider in case of a rerouting. These penalties are usually defined as discounts on the price that the customer pays for the service and in the case of a satellite change can be as high as 40%. Even in cases when the customer is routed over the same satellite but a different up-beam to down-beam channel (i.e., transponder) there is still a rerouting penalty imposed to the satellite provider, albeit a smaller one. The reason for this penalty is that network engineers still have to reconfigure the connection on the customer's side to deal with a possibly different frequency, signal strength and/or polarity. Finally, on-board configuration changes on a satellite require significant programmatic changes on the satellite from ground based controllers, and affect all of the traffic routed through the satellite. Thus, they result in some rerouting penalties for all of the traffic through the satellite and they are only considered when demand patterns change significantly and the realignment of a satellite's capacity with current demand trends outweighs the rerouting penalties imposed by the reconfiguration. Given the significant programmatic changes required and the significant engineering resources in providing such capabilities on a satellite; it is common for satellites to be designed with only (or for ground based engineers controlling the satellite to only consider) a very limited number of configuration possibilities.

The notion of rerouting penalties is the critical characteristic of satellite networks that sets them apart from their terrestrial counterparts and has significant implications as to the way in which one addresses the problem. Because of these penalties it is not enough to plan traffic routing and satellite reconfigurations based on a static picture of the network's current status, since that would ignore the significant cost effect of possible reroutings. As a result one needs to introduce a time dimension into the problem and plan for the routes that service requests are going to use over an extended time horizon. The consideration of routing decisions over time makes it necessary to take into account the highly dynamic nature of satellite networks. Because satellites have a limited life expectancy it is not uncommon to have launches of new satellites, discontinuation of service of old satellites, and (for companies with large fleets) planned relocations of existing satellites to new longitudes. All of these events as well as the possible reconfigurations of the on-board connectivity cause significant changes to the physical topology of the network and affect the routing decisions for all service requests.

In this paper, based on our interactions and a project with one of the world's largest satellite services provider, we tackle the traffic routing problem for existing and future service requests on a satellite network with multiple GEO satellites over several periods of time. Network planners in this context forecast the amount of traffic demanded by service requests between different origin and destination regions based on historical data and strategic decisions for the entire planning horizon. Consequently, the service requests that we consider in this problem have a time dimension and a given traffic component that is a function of time. Additionally, even though the state of the network is dynamic, changes caused by launches of new satellites, relocations of existing spacecraft, and discontinuation of service for old satellites result from high-level strategic decisions and are known with certainty. Therefore, the state of the network can change, but it is predetermined, over the entire planning horizon. Naturally, we wish to route as much demand as possible while minimizing the sum of the routing and penalty costs. Thus, the objective of the *multi-period network planning* (MPNP) problem is to minimize the overall cost of routing "multi-period" traffic requests while making onboard configuration decisions on a dynamic network topology for an extended time horizon.

### 1.1. Related literature

Multi-period network planning and routing problems have been previously considered in the literature. In the absence of any capacity expansion (or installation) decisions or rerouting penalties, the multi-period problem can be reduced to a series of single-period problems. A single-period problem, while challenging, can be posed as an integer multi-commodity flow (IMCF) problem. The IMCF problem has been studied previously by researchers (Alvelos and Valério de Carvalho, 2001, Barnhart et al., 2000, Holmberg and Yuan, 2003) who developed branch-and-price or branch-and-price-and-cut techniques to solve it. Earlier papers on multi-period network planning have largely focused on network dimensioning or capacity planning decisions when demands vary over time (Girard et al., 1991, Medhi and Tipper, 2000). Further, integrality on flows is generally not required or ignored, though it is necessary in the satellite problem. A large number of researchers have focused on the capacity planning or capacitated network design problem (Bienstock and Günlük, 1996, Günlük, 1999).

Optical network design and local access network design problems sometimes address multi-period problems and reconfiguration concerns as traffic patterns change over time (Banerjee and Mukherjee, 2000, Frantzeskakis and Luss, 1999, Labourdette, 1998, Labourdette et al., 1994). However, the approaches taken usually focus on finding the best possible reconfiguration of the network as long as the starting and ending states meet a previously computed optimal criterion. In other words, the goal is to minimize changes while targeting an already known network configuration. In this sense the reconfiguration analysis takes a secondary role and is not the main driving force behind the planning decisions.

The closest related problem in the literature is the multi-period network design problem with incremental routing studied by Lardeux et al. (2007). Here, in the context of the design of the transmission layer of an optical network, the authors consider a network dimensioning (or configuration) problem over multiple time periods where (i) demands increase over time, (ii) fractional routing of demands is permitted, and (iii) no rerouting of past demands is allowed (i.e., once a demand is routed it follows the same path over the entire time horizon). This can be viewed as an extreme case of the problem posed within this paper where the route change penalties are infinite, and as such can easily be incorporated as a special case of the MPNP presented in this paper. The focus in Lardeux et al. (2007) is a polyhedral study, and is closely related to works on the capacitated network design polytope. In contrast, the MPNP problem seeks to minimize the overall cost of routing traffic over an extended planning horizon while taking into account the cost of rerouting traffic (which is permitted at a cost). To the best of our knowledge multi-period routing with the notion of well-defined and significant (in terms of their effect on the objective function) *path-based rerouting penalties* has not been previously examined in the literature.

The rest of this paper is organized as follows. In the next section, we model the multi-period network planning problem on a directed graph and present a path-based formulation with an exponential number of variables. In Section 3, we focus on the novel issues of a branch-and-price-and-cut (BPC) procedure that is specifically designed for our problem. In Section 4, by means of a large set of computational experiments, we evaluate the strengths and weaknesses of our solution approach and compare it to a myopic period-by-period approach that can potentially be used for multi-period planning in the satellite industry. We also compare the BPC procedure with a heuristic strategy that generates columns only at the root node of the branch-and-bound (B&B) tree. We briefly discuss our experience with real-world instances. Finally, in Section 5, we provide concluding remarks and discuss possible directions for future work and problem extensions.

## 2. Problem formulation

We model our problem on a directed graph $G = (N,A)$. The node set $N$ and arc set $A$ consist of disjoint sets $N_t$ and $A_t$, respectively, each one representing the state of the network at time period $t = 1,\ldots,T$, where $T$ is the length of the planning horizon. Each of the node sets $N_t$ contains one set of nodes that represents all origin regions, a different set that represents all destination regions and one node for each up-beam (this node can receive signals from origin nodes) and each down-beam (this node can send signals to destination nodes) on all satellites for the given period. The reason for having two disjoint node sets representing the origin and destination locations of possible customers is that in satellite networks it is not uncommon for services to originate and terminate in the same region. The arcs in our graph represent connections between the origin nodes and up-beams, destination nodes and down-beams, and on-board connectivity for satellites (i.e., up-beam to down-beam connections). In the satellite context, the provider owns the satellites while the customer owns the equipment at the origin and destination nodes. Thus, the only arcs in this representation to have a nonzero cost and capacity associated with them are the ones representing the connections on-board the satellites. We denote the cost per unit of bandwidth of arc $(i,j) \in A$ by $c_{ij}$ and its capacity by $b_{ij}$. Fig. 2 provides an example of this graph for a two-period problem. Notice that $G$ is not connected and it is comprised of distinct components that represent the state of the network at a specific time period $t$. We will refer to the component that is associated with time period $t$, as $G_t = (N_t,A_t)$.

When considering alternative on-board configurations (i.e., connections between up-beams and down-beams) for each satellite, $G$ needs to be augmented by additional node sets and arcs sets. Specifically, for each satellite for which we wish to consider an alternative configuration we replicate the node sets that represent the up-beams and down-beams of the satellite for as many times as the number of different configurations. Each replication of the up-beam and down-beam node sets will represent the state of a spacecraft in one of its possible configurations. As a result we need to connect the up-beam and down-beam sets of each of these replicas according to onboard connections of the configuration they represent. Naturally, we also make the appropriate connections between the origin nodes and the up-beam nodes from all replications and similarly between the down-beam nodes and the destination nodes. We denote the set of satellites in period $t$ as $S_t$ and the set of configurations for each satellite $s \in S_t$ as $U^s$. Fig. 3 shows how the graph $G$ will be augmented to accommodate two configurations for two different satellites.

We denote the set of service requests that we wish to route with $L$. Each service request, $l$, has an origin, a destination and a demand $d^l$ that is a function of time and can be positive only for consecutive time periods (if periods of positive demand are non-consecutive, the request can be treated as multiple requests consisting of positive demands with consecutive time periods). For notational convenience each service request is treated as a separate customer. Further, all demand for each request must be routed on a single path (i.e., no demand splitting is allowed). Our problem resembles a *series* of IMCF problems on each of the $G_t$ components with some additional edge selection decisions. While we discuss the MPNP problem in the context of the satellite communications application where it arose, we should *emphasize* that our model and solution technique is *quite general* and applies to MPNP problems on general graphs with (*any type of*) route change penalties.

A flow based formulation for graph $G$ would require an extremely large number of flow variables $f_{ij}^{lt}$ (i.e., one for each arc $(i,j)$,

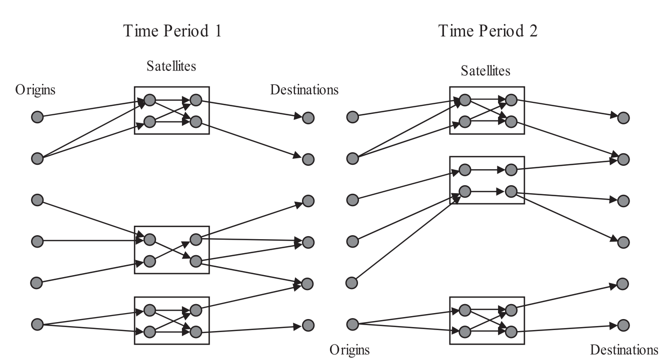


**Fig. 2.** Graph $G$ for two time periods.

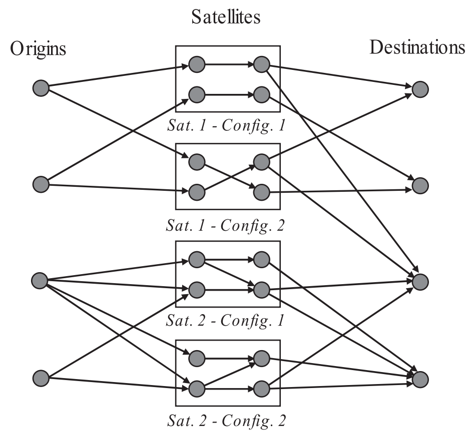


**Fig. 3.** Graph $G_t$ for a specific time period and two satellites, each one having two alternative configurations.

for each customer/service request $l$ and time period $t$). Moreover, tracking the rerouting penalties with the use of flows would require additional decision variables and constraints that would be able to capture the differences $\left|f_{ij}^{l(t-1)} - f_{ij}^{lt}\right|$ for each arc $(i,j)$ and each time period $t = 2,\ldots,T$. These extra variables and constraints make the flow-based approach intractable even for a small number of time periods. Instead, we use a path-based formulation quite similar to those discussed previously in the literature (Alvelos and Valério de Carvalho, 2001, Barnhart et al., 2000, Holmberg and Yuan, 2003) for the IMCF problem.

We introduce decision variables $x_p^l$ that denote whether path $p$ will be used to route service request $l$. Path $p$ is not a path in the conventional sense. It can be thought of as a "super-path" representing the entire sequence of paths across the different time periods over which customer $l$'s traffic will flow. We denote the set of all such "super-paths" $p$ that can be used to carry customer $l$'s traffic with $P^l$. Specifically,

$$x_p^l = \begin{cases} 1, & \text{if super-path } p \text{ will be used to carry customer } l\text{'s traffic,} \\ 0, & \text{otherwise.} \end{cases}$$

If we ignore the possibility of reconfiguration decisions we can formulate a simpler version of the MPNP that deals exclusively with the routing decisions of the service requests over the time horizon. We call this simpler version, the multi-period traffic routing (MPTR) problem and model it by the following integer programming formulation.

$$\text{(MPTR)} \qquad \min \sum_{l \in L} \sum_{p \in P^l} c_p^l x_p^l$$

$$\text{subject to} \quad \sum_{l \in L} d_t^l \left( \sum_{p \in P^l} \delta_{ij}^p x_p^l \right) \leqslant b_{ij}, \quad t = 1, \ldots, T, (i,j) \in A_t, \tag{1}$$

$$\sum_{p \in P^l} x_p^l = 1, \quad \forall l \in L, \tag{2}$$

$$x_p^l \in \{0, 1\}, \quad \forall l \in L, p \in P^l. \tag{3}$$

In this model $d_t^l$ represents the traffic demand for customer $l$ in time period $t$. $\delta_{ij}^p$ is one if super-path $p$ uses arc $(i,j)$ and is zero otherwise. $c_p^l$ denotes the cost of path $p$ for customer $l$ and includes the arc costs as well as the rerouting penalties for path $p$. Specifically,

$$c_p^l = \sum_{t=1}^{T} \sum_{(i,j) \in A_t} \delta_{ij}^p d_t^l c_{ij} + \sum_{t=1}^{T} e_t^l \gamma_t^p, \tag{4}$$

where $\gamma_t^p$ is one if there is a rerouting for super-path $p$ from period $t-1$ to period $t$ (zero otherwise) and $e_t^l$ is the rerouting penalty cost for customer $l$ in period $t$. Notice that for the first time period for which a customer is routed and after the last time period in which a customer has non-zero traffic demand there are no reroutings. For example, for $t = 1$ there are no reroutings and therefore $\gamma_1^p = 0,\ \forall p \in P^l$. In this model, the objective is to minimize the overall cost of routing the demand while taking into account the rerouting penalties. Constraint set (1) ensures that the capacity of an arc is not exceeded. Constraint set (2) ensures that exactly one of all the possible super paths for each customer is selected.

In order to model the MPNP problem we introduce additional decision variables $y_t^{su}$ that indicate the chosen configuration $u$, for satellite $s$, at time period $t$. Specifically,

$$y_t^{su} = \begin{cases} 1, & \text{if satellite } s \text{ is using configuration } u \text{ during time period } t, \\ 0, & \text{otherwise.} \end{cases}$$

The multiperiod network planning problem in satellite networks can now be modeled by the following integer programming formulation.

$$\text{(MPNP)} \qquad \min \sum_{l \in L} \sum_{p \in P^l} c_p^l x_p^l$$

$$\text{subject to} \quad (1), (2), (3),$$

$$\sum_{p \in P^l} x_p^l \beta_{pt}^{su} \leqslant y_t^{su}, \quad t = 1, \ldots, T, s \in S_t, u \in U^s, l \in L, \tag{5}$$

$$\sum_{u \in U^s} y_t^{su} = 1, \quad t = 1, \ldots, T, s \in S_t, \tag{6}$$

$$y_t^{su} \in \{0, 1\}, \quad t = 1, \ldots, T, s \in S_t, u \in U^s. \tag{7}$$

Here $\beta_{pt}^{su}$ is a coefficient which is set to one if super-path $p$ is using satellite $s$'s configuration $u$ at time period $t$ and zero otherwise. As with MPTR the objective of MPNP is to minimize all routing costs, including possible rerouting penalties. Constraint (5) ensures that if a configuration for a particular satellite and time period is not selected then all paths that use that configuration cannot be selected either. Constraint (6) forces exactly one configuration to be selected for each satellite and each time period and constraint (7) defines the configuration selection variables as binary. Before we discuss our solution approaches we note that the integrality constraints on the configuration variables (i.e., constraint (7)) can be relaxed (as long as the $x_p^l$ variables are binary). This follows in a straightforward fashion by observing that each $y_t^{su}$ variable occurs in exactly two constraints (once in constraint (5) and once in constraint (6)) resulting in a totally unimodular constraint matrix on the $y_t^{su}$ variables.

## 3. Solution approach

We now describe our solution approach for the multi-period network planning problem that uses the MPNP formulation in conjunction with a branch-and-price-and-cut procedure.

### 3.1. Overview

Branch-and-price or IP column-generation has been known as a theoretical solution technique for integer programming problems, with an exponential number of variables, for the past 40 years. However, it has only found computational success recently over the past 15–20 years. Some applications, surveys and discussions on specific issues relating to branch-and-price can be found in Barnhart et al. (1998), Desrosiers et al. (1995), Savelsbergh (1997), Vance et al. (1997) and Vanderbeck and Wolsey (1996).

More recently Vanderbeck (2000) discusses and compares general branching strategies for branch-and-price procedures and the book edited by Desaulniers et al. (2005) consists of a collection on practical applications in different contexts and theoretical results. Given the vast literature on IP column generation, in what follows we assume familiarity with the basic concepts involved in the BPC framework.

Our problem differs significantly from those studied previously in the literature (Alvelos and Valério de Carvalho, 2001, Barnhart et al., 2000, Holmberg and Yuan, 2003) due to the rerouting penalties involved. Consequently, while the structure of the MPNP and MPTR super-path based formulations is virtually identical to the path based formulation for the IMCF problem, the BPC algorithms developed for the IMCF problem cannot be applied to the formulations presented here. The reason is that the solution to the pricing problem for the IMCF problem no longer applies when there are route change penalties. Instead, we now present a novel algorithm for solving the pricing problem of the MPTR and MPNP formulations. We also discuss (in the Appendix) how our procedure can be generalized to deal with a much broader class of rerouting penalties. After discussing pricing, we elaborate on the branching, cutting, and initialization components of our BPC implementation.

### 3.2. Pricing in MPTR

For ease of exposition we first discuss the MPTR problem. In the typical IMCF problem setting the pricing problem can be solved with the use of a shortest-path algorithm on the original graph with slightly modified costs. Specifically, the cost structure is usually defined in a way that allows the path costs $c_p'^l$ for commodity $l$ (we use the notation $c_p'^l$ to distinguish the path costs in the IMCF problem from the path costs $c_p^l$ in the MPTR problem) to be represented as the sum of the arc costs on the path, $\sum_{(i,j)\in A}\delta_{ij}^p c_{ij}$. This in turn leads to the computation of the reduced cost for path $p$ and commodity $l$ as,

$$\overline{c'}_p^l = \sum_{(i,j)\in A} d^l(c_{ij}+\pi_{ij})\delta_{ij}^p - \sigma^l,$$

where $\pi_{ij}$ is the dual of the capacity constraints (1) when written as a $\geqslant$ constraint and $\sigma^l$ is the dual of the path selection constraints (2). As a result, the cost of an arc $(i,j)$ can be updated as $d^l(c_{ij}+\pi_{ij})$ and a shortest path algorithm can be used to find a path $p$ for commodity $l$ with the smallest possible cost (notice that $\pi_{ij} \geqslant 0$ and $\sigma^l$ is unrestricted in sign). If that cost is less than $\sigma^l$, then the reduced cost of this path is negative and the path is added to the restricted master problem (RMP) and the updated RMP is reoptimized.

In the satellite routing problem the super-path variables $x_p^l$ in MPTR represent a *series* of paths that commodity $l$ will follow across the different time periods in the planning horizon. Therefore the cost of each super-path consists of an arc-cost component and a rerouting component, as seen in Eq. (4). Specifically, the reduced cost for path $p$ and commodity $l$ is given by,

$$\bar{c}_p^l = \sum_{t=1}^{T}\sum_{(i,j)\in A} d_t^l(c_{ij}+\pi_{ij})\delta_{ij}^p + \sum_{t=1}^{T} e_t^l\gamma_t^p - \sigma^l. \tag{8}$$

Unfortunately, the reduced cost defined in (8) *cannot* be calculated by using the traditional approach that finds a shortest path on the original graph with updated costs, even if we separate it by period. This is because, any approach that uses only the updated arc costs will fail to capture the rerouting penalties (that are dependent on path changes across time periods) associated with some of the super-paths. Therefore, in order to find the super-path $p$ with the lowest reduced cost for each commodity $l$ we develop a technique that calculates the minimum cost routing across all time periods while taking into account rerouting penalties.

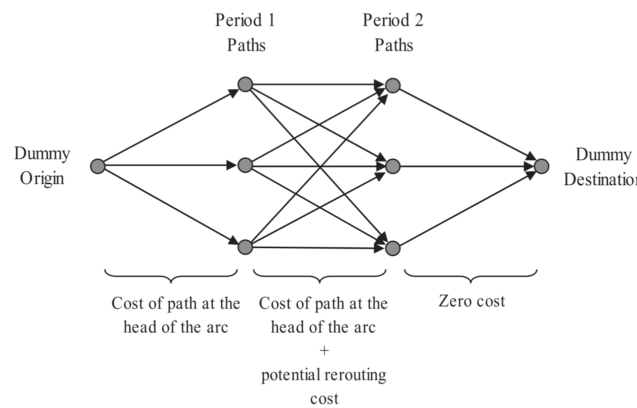


**Fig. 4.** Pricing graph, $G'$, for a problem with 2 time periods and 3 paths per period.

The first step in this approach involves solving a $K$-shortest path problem on $G_t$, between the customer's origin and destination, for each time period $t$ in which that customer has positive demand. The arc costs, on graph $G_t$ are updated with the dual values of the capacity constraints $\pi_{ij}$ in exactly the same way as in the traditional pricing problem approach (i.e., $c_{ij}+\pi_{ij}$). The number of paths $K_t$ that we need to find in time period $t$ is not fixed and can be different for different commodities and time periods. We will specifically discuss how $K_t$ is determined later in this section. Once we have found the $K_t$ shortest paths for each time period we then construct a "pricing graph" $G' = (N', A')$ in which the node set consists of a dummy origin node, a dummy destination node, and one node for each of the shortest paths found in each time period. We define the arcs of this graph as follows. We create arcs from the origin node to all first period nodes (i.e., nodes that represent paths in the first period that a customer has positive demand) and arcs from the last period nodes (i.e., nodes that represent paths in the last period that a customer has positive demand) to the destination node. Furthermore we connect all nodes from period $t-1$ to the nodes in period $t$ and set the cost, $h_{ij}$, of an arc $(i,j)$ equal to $h(q_j)+e_t^l$, where $h(q_j)$ is the cost of the path, $q_j$ represented by node $j$ taking into account the demand; $e_t^l$ is the penalty cost introduced only if the path $q_j$ represented by node $j$ is different from the path $q_i$ represented by node $i$. In the satellite planning context two paths in two different time periods are considered to be different when any of the arcs they include represent different communication links (i.e., origin to up-beam, on-board, or down-beam to destination links) or they represent the same links but the satellite configuration has been changed (configuration changes result when there are changes in the onboard connectivity on a satellite or when the satellite network itself changes; e.g., when a satellite is relocated to a new longitude). For arcs $(i,j)$ where $i$ is the dummy origin node we introduce no penalty cost[1] (i.e., $h_{ij} = h(q_j)$ and $e_1^l = 0$) and when $j$ is the dummy destination node we set $h_{ij} = 0$ (i.e., $e_{T+1}^l = 0$). Notice that a path in the pricing graph represents a super-path $p$ in the MPTR problem. Specifically, the nodes that are used in the path on $G'$ (apart from the dummy origin and destination nodes) represent paths in $G$. Consequently, when we create a node in time period $t$ of $G'$ for each path in $G_t$ there is a one-to-one correspondence between the paths in $G'$ and the super-paths in $G$. Fig. 4 presents the multi-period routing graph for a problem with 2 time periods in which 3 shortest paths have been calculated for each period.

[1] In practice, we might want to introduce penalties even when $i$ is the dummy origin node so that we can account for reroutings of existing service requests.

Once the construction of the pricing graph is complete we solve a shortest path problem from the dummy origin node to the dummy destination node. The cost of this path is then compared to the dual variable $\sigma^l$ and if it is smaller we add the corresponding super-path $p$ to our model. If the cost of the path is larger than the dual variable of the path selection constraints, then there are no super-paths for commodity $l$ that can improve the current solution. Naturally, we have to repeat the same process for all commodities $l$ in our model.

Clearly the size of the pricing graph is exponential if all paths for each time period (i.e., all paths of $G_t$) are generated and included as nodes of $G'$. Instead of generating all paths for a time period, we specify the following sufficient condition that can be used to determine whether a specific choice of $\{K_1, K_2, \ldots, K_T\}$ ensures that we have found the lowest cost super-path. Let $q_n^t$ denote the $n^{\text{th}}$ shortest path in time period $t$.

**Proposition 1.** *The pricing graph $G'$ contains a lowest cost super-path $p$, if $h\left(q_{K_t}^t\right) - h(q_1^t) \geqslant e_t^l + e_{t+1}^l$, for $t = 1,\ldots,T$.*

**Proof.** Suppose not. Let $p^*$ be a lowest cost super-path. Then for some time period $t$, $p^*$ contains a path $q_j^t$ distinct from $q_1^t, \ldots, q_{K_t}^t$, (i.e., $j > K_t$) and therefore $h\left(q_j^t\right) \geqslant h\left(q_{K_t}^t\right)$. By replacing path $q_j^t$ by path $q_1^t$ in super-path $p^*$ we can get a super-path with cost less than or equal to $p^*$, since $h\left(q_{K_t}^t\right) - h(q_1^t) \geqslant e_t^l + e_{t+1}^l$ and in the worst case we will incur one penalty $(e_t^l)$ going from time period $t-1$ to $t$ and another one $(e_{t+1}^l)$ going from time period $t$ to $t+1$. Consequently this new super-path is also optimal. Repeating this procedure for each time period $t$ in which $p^*$ contains paths distinct from $q_1^t, \ldots, q_{K_t}^t$, we obtain a lowest cost super path that belongs to $G'$. □

It is actually possible to generate significantly fewer paths in each time period. This is critical, since the time spent in pricing will be a function of the number of paths we generate and include as nodes in the pricing graph $G'$. To explain how, we need some additional notation. Let $R^t = \left\{q_1^t, q_2^t, \ldots, q_{K_t}^t\right\}$ denote the set of $K_t$-shortest paths in time period $t$. Let $P^t$ denote the set of all feasible paths in time period $t$. For each time period $t$, we define four quantities $t_a$, $t_b$, $t_c$, and $t_d$. Let

$$t_a = \begin{cases} T-t, & \text{if } h\left(q_{K_i}^i\right) - h(q_1^i) \leqslant e_i^l + e_{i+1}^l \text{ and } R^i \neq P^i, \text{ for } i = t, t+1, \ldots, T, \\ \min\left\{i \in [0, T-t] : h\left(q_{K_{t+i}}^{t+i}\right) - h(q_1^{t+i}) > e_{t+i}^l + e_{t+i+1}^l \text{ or } R^{t+i} = P^{t+i}\right\}, & \text{otherwise.} \end{cases}$$

Here $t_a$ tells us the first occurrence, in terms of the number of time periods after $t$, of a time period where either the cost of the $K$th-shortest path (actually $K_{t+t_a}$-shortest path in time period $t + t_a$) is greater than the cost of the shortest path for that time period plus the rerouting penalty from the previous time period and the rerouting penalty to the next time period, or the time period has generated all possible paths between the origin and destination. If no such time period exists then $t_a$ is defined as $T-t$, the largest possible value it could take. Similarly, let

$$t_b = \begin{cases} t-1, & \text{if } h\left(q_{K_i}^i\right) - h(q_1^i) \leqslant e_i^l + e_{i+1}^l \text{ and } R^i \neq P^i, \text{ for } i = 1, 2, \ldots, t, \\ \min\left\{i \in [0, t-1] : h\left(q_{K_{t-i}}^{t-i}\right) - h(q_1^{t-i}) > e_{t-i}^l + e_{t-i+1}^l \text{ or } R^{t-i} = P^{t-i}\right\}, & \text{otherwise.} \end{cases}$$

Here $t_b$ is similar to $t_a$ except that we are now looking for the first time period prior to (and including) time period $t$. Let

$$t_c = \begin{cases} T-t, & \text{if } h\left(q_{K_i}^i\right) - h(q_1^i) \leqslant e_i^l + e_{i+1}^l \text{ and } R^i \neq P^i, \text{ for } i = t, t+1, \ldots, T, \\ 0, & \text{if } t_a = 0, \\ t_a - 1, & \text{otherwise.} \end{cases}$$

Here $t_c$ tells us the number of consecutive time periods after $t$ for which $h\left(q_{K_i}^i\right) - h(q_1^i) \leqslant e_i^l + e_{i+1}^l$ and $R^i \neq P^i$. Similarly, let

$$t_d = \begin{cases} t-1, & \text{if } h\left(q_{K_i}^i\right) - h(q_1^i) \leqslant e_i^l + e_{i+1}^l \text{ and } R^i \neq P^i, \text{ for } i = 1, 2, \ldots, t, \\ 0, & \text{if } t_b = 0, \\ t_b - 1, & \text{otherwise.} \end{cases}$$

Here $t_d$ is similar to $t_c$ except that we are looking for the number of consecutive time periods prior to $t$ for which $h\left(q_{K_i}^i\right) - h(q_1^i) \leqslant e_i^l + e_{i+1}^l$ and $R^i \neq P^i$.

For a given path $q_j^r$ in time period $r$, we are interested in knowing whether this path is feasible in another time period $t$. Let $F^t\left(q_j^r\right) = \emptyset$ if path $q_j^r$ does not exist in time period $t$, and $F^t\left(q_j^r\right) = q_k^t$ for some positive $k$ if the path exists in time period $t$ (i.e., $F^t\left(q_j^r\right) \in P^t$). In other words $F^t(\cdot)$ is a mapping of a path to time period $t$, that tells us whether that path is feasible in time period $t$. When $F^t(\cdot)$ is applied to a set of paths $A = \{a_1, a_2, \ldots, a_n\}$, it outputs the set of paths obtained by applying $F^t(\cdot)$ to each of the paths in $A$. That is, $F^t(A) = \{F^t(a_1), F^t(a_2), \ldots, F^t(a_n)\}$. Let $R_t^s$ be the set of paths from $R^s$ that are valid for time period $t$. That is, $R_t^s = F^t(R^s)$.

We now describe two methods to generate significantly fewer paths in each time period. Let $Q^t = \bigcup_{r=t-t_b}^{r=t+t_a} R_t^r \setminus R^t$. $Q^t$ includes all the $K_r$ shortest paths from time periods $r = t - t_b$ to $r = t + t_a$ that are distinct from $R^t$ (the $K_t$ shortest paths in time period $t$) and valid for time period $t$. Notice, when $h\left(q_{K_t}^t\right) - h(q_1^t) > e_t^l + e_{t+1}^l$ or $R^t = P^t$, $Q^t = \emptyset$. Also, observe that the cost of any path in $Q^t$ is greater than or equal to the cost of all of the paths in $R^t$. Let $S^t = R^t \cup Q^t$.

We construct the "pricing graph" $G'$ as described before, except that the set of nodes (i.e., paths) in time period $t$ are created from the set $S^t$ (i.e., we create one node in time period $t$ for each of the paths in $S^t$). We now show that if we ensure

$$\bigcap_{r=t-t_d}^{r=t+t_c} F^t(R^r) \neq \emptyset \quad \text{for } t = 1, \ldots, T, \tag{9}$$

then the pricing graph $G'$ is guaranteed to contain a lowest cost super-path. Condition (9) says that when there is at least one common path for every maximal set of consecutive time periods that satisfy $h\left(q_{K_i}^i\right) - h(q_1^i) \leqslant e_i^l + e_{i+1}^l$ and $R^i \neq P^i$, the pricing graph $G'$ contains a lowest cost super-path.

**Theorem 1.** *When Condition (9) is satisfied, the pricing graph $G'$ contains a lowest cost super-path.*

**Proof.** Suppose not. Let $p^*$ be a lowest cost super-path. Then there is some time period $r$ in which $p^*$ contains a path $q_j^r$ that does not belong to $S^r$. If $r_a = r_b = 0$, then either $R^r = P^r$ or $h(q_{K_r}^r) - h(q_1^r) > e_r^l + e_{r+1}^l$. In the former case $q_j^r \in R^r = S^r$ and we have a contradiction. In the latter case, replacing path $q_j^r$ by path $q_1^r$ strictly reduces the cost of the super-path yielding a contradiction.

Consequently, assume $r_a + r_b > 0$. Further, consider the subcase where $r_a = r_c + 1$ and $r_b = r_d + 1$. The proofs of the other three subcases: (1) $r_a = r_c = T - r$ and $r_b = r_d = r - 1$, (2) $r_a = r_c + 1$ and $r_b = r_d = r - 1$, and (3) $r_a = r_c = T - r$ and $r_b = r_d + 1$, follow analogously. Let

$$j_\alpha^r = \max\left\{i : 0 \leqslant i \leqslant r_a \text{ and } q_j^r, F^{r+1}\left(q_j^r\right), \ldots, F^{r+i}\left(q_j^r\right) \in p^*\right\},$$

Loosely speaking, starting at time period $r$, $j_\alpha^r$ denotes the number of time periods after time period $r$ for which the path $F^t\left(q_j^r\right)$ appears consecutively in the super-path $p^*$. Similarly, let

$j^r_\beta = \max\left\{i : 0 \leqslant i \leqslant r_b \text{ and } F^{r-i}\left(q^r_j\right), \ldots, F^{r-1}\left(q^r_j\right),\ q^r_j \in p^*\right\}.$

In other words, the super-path $p^*$ consists of path $q^r_j$ repeatedly between time periods $r - j^r_\beta$ and $r + j^r_\alpha$ with no route change penalty. Specifically,

$F^{r-j^r_\beta}\left(q^r_j\right), F^{r-j^r_\beta+1}\left(q^r_j\right), \ldots, F^{r-1}\left(q^r_j\right), q^r_j, F^{r+1}\left(q^r_j\right), \ldots, F^{r+j^r_\alpha}\left(q^r_j\right) \in p^*.$

Note that $F^t\left(q^r_j\right) \notin S^t$ for $t = r - r_b, \ldots, r + r_a$. Otherwise, the path $F^t\left(q^r_j\right)$ would be in $R^t$ for some $t = r - r_b, \ldots, r + r_a$, and as a result (whenever it is feasible) it would also be in $S^t$ for all $t = r - r_b, \ldots, r + r_a$. If $j^r_\alpha = r_a$, then in time period $r + r_a, F^{r+r_a}\left(q^r_j\right)$ belongs to $p^*$. But, since replacing $F^{r+r_a}\left(q^r_j\right)$ by $q^{r+r_a}_1$ strictly decreases the cost of the super-path (because for $t = r + r_a,\ h\left(F^t\left(q^r_j\right)\right) \geqslant h\left(q^t_{K_t}\right) > h(q^t_1) + e^l_t + e^l_{t+1}$) this is not possible (the other possibility $R^{r+r_a} = P^{r+r_a}$ is eliminated since $F^t\left(q^r_j\right) \notin S^t$ for $t = r + r_a$). Thus $j^r_\alpha < r_a$ (and $j^r_\alpha \leqslant r_c$). Arguing similarly, $j^r_\beta < r_b$ (and $j^r_\beta \leqslant r_d$). Let $q^r_k$ be a common path across $R^{r-r_d}, \ldots, R^r, \ldots, R^{r+r_c}$. Specifically, $q^r_k \in \bigcap_{t=r-r_d}^{t=r+r_c} F^r(R^t)$. Observe that $h(F^t(q^r_k)) \leqslant h\left(F^t\left(q^r_j\right)\right)$ for $t = r - j^r_\beta, \ldots, r + j^r_\alpha$. By replacing path $F^t\left(q^r_j\right)$ by path $F^t(q^r_k)$ in time periods $t = r - j^r_\beta, \ldots, r + j^r_\alpha$ we get a super-path with cost less than or equal to $p^*$. Consequently, this new super-path is also optimal. Repeating this argument for time periods where $p^*$ contains a path $q^r_j$ that does not belong to $S^r$ completes the proof. □

In our second method to reduce the number of paths in $G'$ we define, $Q^t = \bigcup_{r=1}^{r=T} R^r_t \setminus R^t$. $Q^t$ now includes all the $K_r$ shortest paths from time periods $r = 1$ to $r = T$ that are distinct from $R^t$ and valid for time period $t$. Observe that the cost of any path in $Q^t$ is greater than or equal to the cost of all of the paths in $R^t$. Like the first method, the "pricing graph" $G'$ is constructed as before, with the set of nodes in time period $t$ created from the set $S^t = R^t \cup Q^t$. We now show that if we ensure

$$h\left(q^t_{K_t}\right) - h(q^t_1) \geqslant e^l_t \text{ or } R^t = P^t \quad \text{for } t = 1, \ldots, T, \tag{10}$$

then the multi-period graph $G'$ is guaranteed to contain the lowest cost super-path.

**Theorem 2.** *When Condition (10) is satisfied, the pricing graph $G'$ contains the lowest cost super-path.*

**Proof.** Suppose not. Let $p^*$ be a lowest cost super-path. Then for some time period $r$, $p^*$ contains a path $q^r_j$ not in $S^r$. Let

$$j^r_\alpha = \max\left\{i : 0 \leqslant i \leqslant T - t \text{ and } q^r_j, F^{r+1}\left(q^r_j\right), \ldots, F^{r+i}\left(q^r_j\right) \in p^*\right\},$$

and

$$j^r_\beta = \max\left\{i : 0 \leqslant i \leqslant t - 1 \text{ and } F^{r-i}\left(q^r_j\right), \ldots, F^{r-1}\left(q^r_j\right), q^r_j \in p^*\right\}.$$

Observe, that the paths $F^{r+i}\left(q^r_j\right)$ for $i = 0, \ldots, j^r_\alpha$ and the paths $F^{r-i}\left(q^r_j\right)$ for $i = 0, \ldots, j^r_\beta$ do not belong in $S^i$ for $i = r - j^r_\beta, \ldots, r, \ldots, r + j^r_\alpha$. We construct a new super-path by replacing the path $F^i\left(q^r_j\right)$ by $q^i_1$ in time periods $i = r - j^r_\beta, \ldots, r, \ldots, r + j^r_\alpha$. Notice that by using the new paths we may incur up to $j^r_\alpha + j^r_\beta$ extra penalties (specifically $\sum_{i=r-j^r_\beta+1}^{r+j^r_\alpha} e^l_i$). However, $h\left(F^i\left(q^r_j\right)\right) \geqslant h(q^i_K) \geqslant h(q^i_1) + e^l_i$, for all $i = r - j^r_\beta, \ldots, r + j^r_\alpha$ and thus $\sum_{i=r-j^r_\beta}^{r+j^r_\alpha} h\left(F^i\left(q^r_j\right)\right) \geqslant \sum_{i=r-j^r_\beta}^{r+j^r_\alpha} (h(q^i_1) + e^l_i) > \sum_{i=r-j^r_\beta}^{r+j^r_\alpha} h(q^i_1) + \sum_{i=r-j^r_\beta+1}^{r+j^r_\alpha} e^l_i$. As a result the new super-path has a cost that is strictly lower than the cost of $p^*$ which contradicts our initial claim that $p^*$ is the lowest cost super-path. □

In our implementation we generate a small number of paths, say $\kappa_t$, for each commodity and each period $t$ and then check to see whether Condition (9) or Condition (10) is satisfied. If neither of these two conditions are satisfied, we then generate the next set of $\kappa_t$ shortest paths for all time periods in which $R^t \neq P^t$ (i.e., we haven't generated all feasible paths). This is repeated, until either Condition (9) or Condition (10) is satisfied and the appropriate pricing graph is constructed. Notice that by saving the state of the $K$-shortest path algorithm in each time period it is possible to determine the next set of $\kappa_t$ shortest paths without having to recompute paths that were already found.

### 3.3. Pricing in MPNP

We now look at the pricing problem for the MPNP formulation. The reduced cost of an $x^l_p$ variable in the MPNP formulation is given by,

$$\bar{c}^l_p = \sum_{t=1}^{T} \sum_{(i,j)\in A_t} d^l_t (c_{ij} + \pi_{ij}) \delta^p_{ij} + \sum_{t=1}^{T} e^l_t \gamma^p_t + \sum_{t=1}^{T} \sum_{s\in S_t} \sum_{u\in U^s} \beta^{su}_{pt} \theta^{su}_{lt} - \sigma^l, \tag{11}$$

where $\theta^{su}_{lt} (\geqslant 0)$ is the dual variable associated with constraint set (5), again when written as a $\geqslant$ constraint. Observe, that compared to the reduced cost Eqs. (8), (11) has an additional term associated with constraint set (5). There are two possible approaches that we can take when solving the pricing problem and computing the reduced cost of the super-path variables.

The first approach for the pricing problem becomes apparent if we rewrite Eq. (11) as,

$$\bar{c}^l_p = \sum_{t=1}^{T} \sum_{(i,j)\in A_t} d^l_t (c_{ij} + \pi_{ij} + \sum_{s\in S_t} \sum_{u\in U^s} \zeta^{su}_{ij} \frac{\beta^{su}_{pt} \theta^{su}_{lt}}{d^l_t}) \delta^p_{ij} + \sum_{t=1}^{T} e^l_t \gamma^p_t - \sigma^l, \tag{12}$$

where $\zeta^{su}_{ij}$ is a coefficient which is one if arc $(i,j)$ belongs to configuration $u$ of satellite $s$ and zero otherwise. With this rewriting of Eq. (11) the reduced cost of a super-path is composed of an arc dependent term and a path dependent term as in Eq. (8). Thus we can directly apply the approach in the previous section and use Theorems 1 and 2 (which is what we do in our computational work).

In other words, when solving the pricing problem for commodity $l$ we have to update the cost of all arcs $(i,j) \in A$ both by $\pi_{ij}$ and $\theta^{su}_{lt}$ if the arc is part of configuration $u$ of satellite $s$. This way when we solve the $K$-shortest path problems for each time period $t$ in order to determine the nodes of the pricing graph we implicitly take into account the dual information from constraints (5) that enforce the configuration selections. Other than that the procedure remains the same.

The second approach deals directly with the third term in Eq. (11) as part of a more general rerouting penalty that depends not only on the time period $t$ and the commodity $l$ but also upon the path $p$ followed in time period $t - 1$. We discuss this approach (that takes into account general rerouting penalties) in the Appendix.

### 3.4. Cutting

Barnhart et al. (2000) observe that IMCF problems exhibit symmetry effects that make them hard to solve when using solely a branch-and-price approach. Therefore, it is necessary to use strong cuts for the problem polytope that help eliminate the symmetry.

For the standard IMCF problem the capacity constraints in the flow-based formulation when translated to a node-arc representation define 0–1 knapsack inequalities. For the MPNP problem the arc flow variables $f_{ij}^l$, represent the fraction of commodity $l$ flow that uses arc $(i,j)$. Using these arc-flow variables the capacity constraints can be written as,

$$\sum_{l \in L} d_{ij}^l f_{ij}^l \leqslant b_{ij}, \quad \forall (i,j) \in A.$$

We can now use lifted cover inequalities (LCIs) to strengthen the formulation and reduce the symmetry effects. The general form of a LCI with respect to the arc flow variables is,

$$\sum_{l \in C} f_{ij}^l + \sum_{l \in \overline{C}} \alpha_l f_{ij}^l \leqslant |C| - 1, \quad \forall (i,j) \in A,$$

where the set $C$ defines a minimal cover[2] and $\overline{C} = L \setminus C$. By using the flow decomposition theorem (see Ahuja et al., 1993), that states $f_{ij}^l = \sum_{p \in P^l} \delta_{ij}^p x_p^l$ we can go from the LCI written in terms of flow variables to the LCI written in terms of the path variables (since we use a path based model) as,

$$\sum_{l \in C} \sum_{p \in P^l} \delta_{ij}^p x_p^l + \sum_{l \in \overline{C}} \alpha_l \sum_{p \in P^l} \delta_{ij}^p x_p^l \leqslant |C| - 1, \quad \forall (i,j) \in A.$$

In practice after we solve the linear programming (LP) relaxation of the Master Problem (MP) to optimality using column-generation and the pricing procedure presented earlier we look at all the arcs of $G$ that are saturated (i.e., have zero slack). We then create a cover $C$ (similar to Gu et al., 1998) by inserting into $C$ first the commodities for which $f_{ij}^l = 1$ and then $f_{ij}^l < 1$ so that $\sum_{l \in C} d_{ij}^l > b_{ij}$. We then delete any commodities from the cover so as to make it minimal and then use the sequence independent lifting procedure proposed by Gu et al. (2000) to find the lifting coefficients $\alpha_l$. Using this approach we generated one LCI for each saturated arc and added all such LCIs into our model. The RMP is then re-solved and the pricing procedure generates any necessary additional columns. Notice that the cost of the arcs in $G$ will now have to be updated by taking into account the dual variables of the LCI constraints as well. Specifically, when solving the pricing problem for commodity $l$ the cost of arc $(i,j)$ is updated as,[3]

$$c_{ij} + \pi_{ij} + \sum_{m \in M} \alpha_l^m \frac{\eta^m}{d_{ij}^l},$$

where $M$ is the set of all LCIs that refer to arc $(i,j)$, $\alpha_l^m$ is the lifting coefficient of commodity $l$ in the $m^{\text{th}}$ inequality and $\eta^m$ ($\geqslant 0$) is the dual of that inequality when written as a $\geqslant$ constraint.

In our BPC approach we generate cuts with the procedure described above whenever possible and add them to the current model. Since these cuts are globally valid we also add all of these cuts to a global cut pool. At the start of each node in the BPC tree the cut pool is compared against the cuts currently in the node and any cuts that are missing from the node are added before solving the RMP.

### 3.5. Branching

In branch-and-price procedures branching presents an additional challenge since branching rules should not be allowed to interfere with the structure of the pricing problem. Barnhart et al. (2000) have developed a very successful branching rule for IMCF problems, which we applied in our procedure.

The branching rule finds the first node (called the divergence node) for which the path of a commodity $l$ splits in two (or more) fractional paths. The rule then partitions the arcs emanating from this divergence node into two sets, $B$ and $\overline{B}$. The two sets are constructed in such a way that one of the arcs emanating from the divergence node and used by a fractional path for commodity $l$ belongs in set $B$; while a second arc emanating from the divergence node and used by a fractional path for commodity $l$ belongs in set $\overline{B}$. The remaining arcs emanating from the divergence node are assigned to either $B$ or $\overline{B}$ while ensuring the cardinality of these two sets is roughly equal. In the first branch commodity $l$ is not allowed to use the arcs in $B$, while in the second branch commodity $l$ is not allowed to use the arcs in $\overline{B}$. In our case this branching rule can be easily enforced by deleting the arcs from the appropriate set when solving the pricing problem for commodity $l$. By deleting these arcs we ensure that when finding the $K$-shortest paths for commodity $l$ we will consider no paths (and as a result no superpaths) that use these arcs.

We noted earlier that in the MPNP formulation the configuration variables could have been defined as continuous and non-negative, $y_t^{su} \in \mathbb{R}_+$. Naturally, a model with fewer integer (or binary) variables might be preferred since they do not require that we branch on them and usually result in a smaller B&B tree and a faster solution time. However, in our case, at each node of our BPC tree we are solving a restricted problem that does not contain all variables. As a result by branching on the configuration variables first we can impose restrictions on which path variables will be considered. This has a twofold effect. First it can significantly reduce the number of branches required in the BPC tree since branching on a configuration variable will reduce the number of paths considered for all commodities. Additionally, when solving the pricing problem there are fewer columns that could potentially have negative reduced costs and this could lead to faster solutions of the restricted LPs. We found a very significant computational benefit by keeping the configuration variables binary and branching upon them first, followed by branching on the fractional path variables; and thus followed this approach in our computational experiments.

### 3.6. Initialization

Another issue that arises with the use of column-generation procedures is the initial feasibility of the RMP (since it does not include all possible variables). The standard practice that is used to ensure feasibility of the RMP at all nodes in the B&B tree is the inclusion of auxiliary columns with appropriate coefficients for the constraints and costs in the objective function (see Barnhart et al., 1998, Desrosiers et al., 1995, Vanderbeck and Wolsey, 1996). In the case of the MPNP problem we add one path for each commodity. These "feasibility" paths will have a coefficient equal to one for the constraints which ensure that exactly one path is chosen and a cost that must be greater than the cost of all the other paths for that commodity.

With the addition of the "feasibility" paths we have ensured that we are going to find a feasible (integer) solution to the problem when all demand can be met. However, in the case of the MPNP problem and practical applications of the IMCF problem we would still like to get a (integer feasible) solution when some of the requests cannot be routed because of traffic congestion in the network. For this reason we need to add special paths that carry flow directly (i.e., bypassing the network) from the origin to the destination. We refer to these as "unmet demand" paths and we construct them with the help of the pricing graph $G'$. Specifically, we augment graph $G'$ with one node for each time period. The new nodes represent unmet demand paths in their respective period and we refer to them as "unmet nodes". There are two ways in

[2] A set $C \subseteq L$ is a cover if $\sum_{l \in C} d_{ij}^l > b_{ij}$. A cover is minimal if $C \backslash \{l\}$ is not a cover for any $l \in C$.

[3] In Barnhart et al. (2000) the new cost of arcs $(i,j)$ are mistakenly updated by $\alpha_l^m \eta^m$ instead of $\alpha_l^m \eta^m / d_{ij}^l$.

which we can use these nodes depending on how we wish to model unmet demand in our problem. The first option is to connect the unmet node in period $t-1$ only with the unmet node in period $t$, for all time periods. Naturally, the unmet node in the first period is connected to the dummy origin node in $G'$ and the unmet node in the last period is connected to the dummy destination node. This way we allow for one super-path in the RMP (for each commodity) that will represent unmet demand across all time periods and will result in our model either routing customers or denying their service for the entire planning horizon. The second option is to introduce arcs that will connect all the nodes in period $t-1$ with the unmet node in period $t$, and the unmet node in $t$ with all the nodes in $t+1$, for all time periods. Under this scenario we will be able to consider super-paths that allow a commodity to be routed for some periods, then dropped and then possibly routed again. The first option is probably better suited for actual planning purposes since a satellite provider is usually unwilling to stop servicing an existing customer because of the associated, high ill-will costs. However, the second option provides us with the possibility of considering these ill-will costs in the model (if we desire to do so) and is more appropriate when comparing the results of the MPNP formulation with a period-by-period optimization approach (as we do in Section 4). In our implementation we assign a cost to the arc that leads to the unmet node in period $t$ equal to the revenue generated by the service request at period $t$ (plus the appropriate penalty and ill-will costs). With this technique even in cases when all demand cannot be routed we still get a feasible solution that maximizes our profits.

## 4. Computational results

We now present several computational experiments on various data sets. Due to the confidentiality of real-world instances, our computational analysis is mainly focused on simulated data sets. Most of the characteristics of our problem sets are designed to replicate the key attributes of real-life satellite networks and are pertinent to the MPNP problem. We were able to obtain the attributes of actual satellite networks in the course of our project with the satellite communications services company. Our computational work is split into two main directions. First we look at the benefits of applying a multi-period optimization procedure as opposed to a period-by-period optimization process for varying problem characteristics. Then we compare the full blown BPC procedure with a "Root-Node" procedure that uses column-generation only at the root node of the B&B tree and only generates cuts (as opposed to cuts and columns) during the entire search. The BPC and Root-Node procedures were coded in C++with the use of ILOG CPLEX v9.0 and the ILOG Maestro libraries, while the period-by-period process uses only ILOG CPLEX v9.0. All computational work was conducted on a Pentium IV Xeon processor, with 3 GHz clock speed and 2 GB of RAM.

Our computational analysis is done on randomly generated problem sets. Each problem set contains 10 instances. The problems correspond to a network with 2 satellites each with 3 onboard configurations (approximately 100 nodes and 280 arcs in each time period) and a planning horizon of 5 time periods. The arcs representing the on-board connections of the satellites have an average capacity of 1.5 traffic units (one traffic unit is equivalent to 36 MHz of bandwidth) and an average cost of \$150,000 (per traffic unit per time period). The network consists of 10 regions that can act as origins and destinations for each of the 50 customers that have average demands of 0.8 traffic units. The demand for each customer is drawn in each period from a uniform distribution on the interval [0.75,0.85]. A customer that is generated in period $t$ has a 90% chance of "surviving" in the next period and in each period after the first we generate 5 new customers. The unmet demand cost was set to \$750,000 (per traffic unit per time period), which approximates the average revenue generated by a satellite customer (leasing 1 traffic unit) over a one year period. The rerouting penalties in the satellite industry are usually defined as discounts that are offered to the affected customers and are typically set to 40%. The rerouting penalty was therefore set to \$300,000 per traffic unit per time period (i.e., $e_t^l = \$300,000 \times \min\left(d_{t-1}^l, d_t^l\right)$). In order to replicate the dynamic topology of satellite networks we also define a survival probability for the on-board connections (instead of modeling launches, relocations and discontinuation of service for entire satellites) which we set to 90%, so roughly 10% of the on-board links in each configuration will be changed in each period. This survival probability for onboard connections (90%) is actually set artificially low compared to the situation in practice. By setting this probability artificially low we can increase the difficulty of the simulated problems considered and make their difficulty (even with 2 satellites) approach those of real-world instances that contain a larger number of satellites. The set of attributes that we have defined comprise a baseline problem scenario. Individual characteristics of this baseline are then altered so as to explore different aspects of the multi-period traffic routing problem.

For all problem sets presented in the following sections the running time of the BPC procedure was limited to 4 hours. Additionally, following the example of Danna and Pape (2005) our BPC procedure is augmented by heuristics that allow our approach to quickly find integer feasible solutions. The first heuristic that we use is to provide the BPC approach with the Period-by-Period solution and the associated super-paths. Also, at the root node of the BPC tree and every 100 nodes we provide all the super-paths and cuts found up to that point to a regular branch-and-bound procedure and ask for an integer feasible solution within a 10 minute time limit.

### 4.1. Multi-period versus period-by-period

From a practical standpoint it is important to provide tangible proof to executives in the satellite industry as to the benefits of a multi-period approach over a period-by-period optimization process. By period-by-period optimization we refer to the process of myopically routing all of the commodities in period $t$ and then looking at the routing problem for the next period, $t+1$, without being able to change any of the routes in period $t$. We achieve this solution with the use of a typical flow-based formulation. The formulation uses the variables $f_{ij}^{lt}$, which represent the fraction of flow of commodity $l$, in time period $t$, that uses arc $(i,j)$ and the $y_t^{su}$ variables. We now present the period specific network design (PSND) formulation for period $t$,

$$(\mathrm{PSND}_t) \qquad \min \sum_{l\in L} \sum_{(i,j)\in A_t} c_{ij}^{lt} f_{ij}^{lt} \tag{13}$$

$$\text{subject to} \quad \sum_{j:(j,i)\in A_t} f_{ji}^{lt} - \sum_{j:(i,j)\in A_t} f_{ij}^{lt} = \begin{cases} -1 & \text{if } i = O^{lt}, \\ 1 & \text{if } i = D^{lt}, \\ 0 & \text{otherwise}, \end{cases} \quad \forall i \in N_t,\ l \in L, \tag{14}$$

$$\sum_{l\in L} d_{ij}^{l} f_{ij}^{lt} \leqslant b_{ij}, \quad \forall (i,j) \in A_t, \tag{15}$$

$$\sum_{(i,j)\in A_t} f_{ij}^{lt} \gamma_{ij}^{su} \leqslant y_t^{su}, \quad \forall t, s \in S_t,\ u \in U^s,\ l \in L, \tag{16}$$

$$\sum_{u\in U^s} y_t^{su} = 1, \quad \forall t,\ s \in S_t, \tag{17}$$

$$y_t^{su} \in \{0,1\}, \quad \forall t,\ s \in S_t,\ u \in U^s, \tag{18}$$

$$f_{ij}^{lt} \in \{0,1\}, \quad \forall l \in L, (i,j) \in A_t. \tag{19}$$

where $O^{lt}$, $D^{lt}$ are the origin and destination nodes of commodity $l$ at time period $t$ and $\gamma_{ij}^{su}$ is one if arc $(i,j)$ is part of configuration $u$ on satellite $s$ and zero otherwise. Note that we avoid infeasibility when all demand cannot be met by augmenting the set $A_t$ with arcs $(O^{lt},D^{lt})$ that have cost equal to the unmet demand cost, for each commodity $l$ and time period $t$. In order to take into account the rerouting penalties we have two options. Solve the PSND problem for each time period without penalties and then add the penalties based on the solution. Observe that this approach might be *the only course* in the MPNP problem on general graphs. However, in the case of the satellite network the rerouting penalties are effectively applied only when the onboard connection used changes. Thus we can incorporate the cost of a route change penalty by modifying the cost $c_{ij}$ of the arcs corresponding to the onboard connections. Therefore we set $c_{ij}^{lt}$ equal to $c_{ij}d_{ij}^{l} + e_{t}^{l}$ (where $e_{t}^{l}$ is the rerouting penalty) if commodity $l$ has not used arc $(i,j)$ in period $t-1$ and $(i,j)$ represents an on-board connection. We make this change in order to allow the period-by-period approach to take into account, at some level, the route change penalties.

Table 1 presents computational results for the BPC and period-by-period approaches on five different problem sets. These problem sets are characterized by a varying load-factor, which we define as the ratio of the total demand over the aggregate capacity in the network (in each period). The first column in the table specifies the load-factor generated for each problem set. The second and third columns present the solution found by the period-by-period approach and the time required (in seconds) to reach that solution, respectively. The three columns under the heading Multi-Period (BPC) specify the best primal solution found by the BPC procedure, the gap of that solution to the best dual bound, and the running time (all running times are reported in seconds). The last column in the table gives the average percentage gap between the solution of the period-by-period approach and the BPC procedure over all 10 instances. The runs were conducted using the second of the two options for dealing with unmet demand (see Section 3.5) and for both procedures customers were not allowed to be routed in the future once they had been denied service at some point in the past. In the period-by-period approach we achieve this by setting $f^{lt}_{O^{lt}D^{lt}} = 1$ for each time period, $t$, after the one in which the customer was routed over an unmet demand arc. The same effect can be achieved in the multi-period procedure when constructing the multi-period pricing graph $G'$ by allowing only one outgoing arc from the node that represents the unmet path in period $t$ to the node that represents the unmet path in $t+1$. Also, for both procedures we imposed an extra penalty when a customer that was routed is dropped in some future time period. This penalty represents the ill-will cost associated with denying service to an existing customer and we set it equal to the unmet demand cost. The results show that as the load factor changes the percentage gap between the two procedures varies from a low of 5.40% to a high of 7.66%. What is important to note is that in all cases the BPC procedure achieves gaps that are within 1% of optimality.

Table 2 provides another comparison between the multi-period and period-by-period approaches on four different problem sets. The table has the same structure as before and the problem sets are characterized by different rerouting penalties. This table can provide some insight as to when the rerouting penalty value is high enough to make the use of a multi-period approach beneficial. For this comparison we disabled the heuristic that provides the BPC procedure with the solution generated by the period-by-period approach. Observe that for the extreme case in which the rerouting penalty is zero (e.g., on terrestrial fiber optic traffic routing) the period-by-period approach can be, in theory, as good as a multi-period approach. However, other restrictions, such as the fact that we do not allow for customers that have been dropped to be routed in future time periods and the fact that we impose an extra penalty for dropping customers will always allow a multi-period approach to maintain the advantage. In Table 2 the first column gives us the penalty value as a percentage of the unmet demand cost used (i.e., \$750,000). Note that for high values of the rerouting penalty the difference between the two approaches becomes as high as 10.06%. For a rerouting penalty equal to zero the period-by-period approach actually does better when looking at the average over all the instances. Obviously, this is due to the fact that we used a 4 hour time limit for all problem runs and the fact that the runs of the BPC procedure did not converge (as is evident from the 24.91% gap observed). From this comparison we can have a clear indication as to the effect of the rerouting penalty size and gauge the potential benefits of a multi-period versus a period-by-period approach as that penalty changes.

Tables 3 and 4 compare the two approaches as the number of time periods and the number of alternative configurations increases. Observe that the running time for the BPC algorithm does not change significantly from the baseline (5 time periods) as the number of periods increases. The same is not true however for increasing the number of configurations considered from the baseline (3 configurations). Notice that when considering multiple configurations the size of the underlying network increases significantly. This seems to suggest that our BPC procedure is not

**Table 1**
Comparison of Period-by-Period to Multi-Period optimization for different load-factors.

| Load factor | Period-by-period | | Multi-period (BPC) | | | Primal gap (%) |
|---|---|---|---|---|---|---|
| | Primal | Time (s) | Primal | Gap (%) | Time (s) | |
| 0.4 | 43,260,909 | 1.2 | 40,950,651 | 0.02 | 4232 | 5.64 |
| 0.5 | 59,658,995 | 1.3 | 55,862,551 | 0.16 | 5404 | 6.67 |
| 0.6 | 73,755,286 | 1.5 | 70,025,220 | 0.16 | 10,954 | 5.40 |
| 0.7 | 95,354,132 | 2.2 | 88,573,897 | 0.13 | 10,563 | 7.66 |
| 0.8 | 119,884,983 | 1.2 | 112,817,535 | 0.18 | 10,126 | 6.23 |

**Table 2**
Comparison of period-by-period to multi-period optimization for different rerouting penalty values.

| Penalty (% of unmet cost) | Period-by-period | | Multi-period (BPC) | | | Primal gap (%) |
|---|---|---|---|---|---|---|
| | Primal | Time (s) | Primal | Gap (%) | Time (s) | |
| 0 | 55,998,362 | 164.8 | 66,142,850 | 24.91 | 14,454 | −10.12 |
| 10 | 63,138,975 | 3.2 | 61,583,609 | 0.21 | 14,407 | 2.37 |
| 40 | 73,755,286 | 1.5 | 70,025,220 | 0.16 | 10,954 | 5.40 |
| 100 | 95,421,076 | 1.4 | 86,668,356 | 0.10 | 8,695 | 10.06 |

**Table 3**
Comparison of period-by-period to multi-period optimization for different time periods.

| No. of time periods | Period-by-period | | Multi-period (BPC) | | | Primal gap (%) |
|---|---|---|---|---|---|---|
| | Primal | Time (s) | Primal | Gap (%) | Time (s) | |
| 3 | 34,740,282 | 1.2 | 33,147,239 | 0.09 | 3136 | 4.85 |
| 5 | 73,755,286 | 1.5 | 70,025,220 | 0.16 | 10,954 | 5.40 |
| 7 | 118,963,137 | 3.2 | 108,809,792 | 0.03 | 9242 | 9.36 |

**Table 4**
Comparison of Period-by-Period to Multi-Period optimization for different number of alternative on-board configurations.

| No. of onboard configurations | Period-by-period | | Multi-period (BPC) | | | Primal gap (%) |
|---|---|---|---|---|---|---|
| | Primal | Time (s) | Primal | Gap (%) | Time (s) | |
| 1 | 79,549,049 | 0.3 | 75,699,962 | 0.20 | 6605 | 5.08 |
| 3 | 73,755,286 | 1.5 | 70,025,220 | 0.16 | 10,954 | 5.40 |
| 5 | 73,888,278 | 4.5 | 72,542,398 | 9.38 | 13,783 | 2.02 |

adversely affected by the size of the planning horizon but larger number of configurations can require significantly more time to solve. Luckily, in practice network operators prefer to make this decision outside our model (i.e., for the perspective of our model there is only a single given configuration in each time period) or limit the number of configurations severely to at most 2 or 3 for each satellite. Also, observe that the percentage gap between the solutions for the two procedures remains relatively constant in Tables 1, 3 and 4. This is a another indication of the considerable effect of the rerouting penalty on MPNP problems and the value of applying a multi-period approach when that penalty is significant.

Table 5 provides a comparison of the two approaches as the number of customers that are routed increases. For these problems we kept the load factor constant while increasing the number of customer which resulted in individual customers with smaller demands on average. We observe that the difference between the two procedures is higher for smaller number of customers. This could be attributed to the fact that the rerouting penalties are proportional to customer demands and problems with larger demands are likely to result in a larger number of reroutings in the period-by-period optimization.

Table 6 presents a comparison of the period-by-period approach to the BPC procedure for larger problem instances with a single satellite configuration per time period. The problem set name SxCy specifies that x satellites were available in the network and y customers were routed. The problems were generated in exactly the same way as the earlier random instances with satellite and customer characteristics drawn from the same distributions as before. Each row in this table corresponds to a single instance. The results show that the savings in the operational costs for large networks can reach up to $50 million over 5 time periods for the largest network considered. Notice that even for these larger networks the BPC procedure can still achieve gaps that are in most cases within 1% of optimality within 4 hours.

### 4.2. BPC versus Root-Node

It is common in the mathematical programming literature to compare BPC procedures with heuristic approaches that use column-generation only at the root node and then go through the B&B tree without introducing new variables. These comparisons are usually indicative of the potential benefits of generating columns throughout the B&B tree but can also suggest that column-generation at the root node only can be used as a heuristic in practice without a significant disadvantage.

In Table 7 we compare the BPC and Root-Node procedures when varying the number of alternative onboard configurations. This table has a similar structure to Table 4 and provides the primal solution, IP-LP percentage gap (the lower bound used for computing this gap is from the BPC procedure as the lower bound from the root node procedure is not a valid lower bound for the problem) and computational time (in seconds) required for both procedures. Also, it provides the percentage difference between the primal solutions found by the two approaches over all instances. These

**Table 5**
Comparison of period-by-period to multi-period optimization for different number of customers.

| No. of customer requests | Period-by-period | | Multi-period (BPC) | | | Primal gap (%) |
|---|---|---|---|---|---|---|
| | Primal | Time (s) | Primal | Gap (%) | Time (s) | |
| 25 | 127,884,592 | 0.3 | 118,805,325 | 0.00 | 912 | 7.66 |
| 50 | 73,755,286 | 1.5 | 70,025,220 | 0.16 | 10,954 | 5.40 |
| 75 | 58,167,396 | 388.2 | 54,655,210 | 0.51 | 14,406 | 5.21 |

**Table 6**
Comparison of period-by-period to multi-period optimization for larger problem instances.

| Problem set | Period-by-period | | Multi-period (BPC) | | | Primal gap (%) |
|---|---|---|---|---|---|---|
| | Primal | Time (s) | Primal | Gap (%) | Time (s) | |
| S5C100 | 154,528,944 | 1.4 | 135,960,568 | 0.72 | 14,404 | 13.66 |
| S10C100 | 505,948,001 | 1.4 | 470,800,971 | 0.20 | 14,404 | 7.47 |
| S10C200 | 248,211,923 | 1,303.7 | 224,283,824 | 3.52 | 14,409 | 10.67 |
| S20C200 | 1,021,336,240 | 6.8 | 970,311,929 | 0.58 | 14,445 | 5.26 |

**Table 7**
Comparison of Root-Node to branch-and-price-and-cut for different number of alternative onboard configurations.

| No. of onboard configurations | Root-Node | | | BPC | | | Primal gap (%) |
|---|---|---|---|---|---|---|---|
| | Primal | Gap (%) | Time (s) | Primal | Gap (%) | Time (s) | |
| 1 | 75,720,657 | 0.6 | 11.9 | 75,699,962 | 0.20 | 6,605 | 0.02 |
| 3 | 94,960,430 | 64.0 | 629.3 | 70,025,220 | 0.16 | 10,954 | 35.99 |
| 5 | 98,801,734 | 77.1 | 4,715.5 | 72,542,398 | 9.38 | 13,783 | 36.37 |

gaps are computed as the difference of the primal bound of the Root-Node procedure minus the primal bound of the BPC procedure over the primal bound of the BPC procedure. What we observe from the results is that the Root-Node approach does significantly worse than the BPC procedure for all but one of the problem sets. The only situation in which we could potentially use the Root-Node approach is the one in which we only consider one alternative configuration. In all other cases the Root-Node approach discovers super-paths at the root node that use specific configurations. However once the root node is solved because of branching we end up using only a single configuration which might not be associated with many of the super-paths found at the root node. Therefore in the end the Root-Node procedure drops many of the customers because it lacks any feasible paths over the configurations chosen. Obviously this problem is absent when we only consider satellites with a single configuration and the approach can do significantly better. This behavior is confirmed in additional experiments described in Gamvros (2006) for different problem settings where satellites have multiple configurations. Consequently, we do not recommend the use of the root node procedure except possibly in situations where the satellite onboard configurations are all determined a priori.

### 4.3. Real-world instances

In the satellite industry large providers can have over 20 satellites and more than 10,000 service requests to route over the planning horizon they are considering. Our Root-Node procedure[4] has been successfully tested on real-world instances with up to 30 satellites, 1500 service requests (the requests were aggregated in order to reduce their number to a manageable size) and 5 time periods (typically one time period was equivalent to one year). Since it is current practice at the satellite company that we worked with to make configuration decisions a priori (i.e., outside the model), in these cases we considered only one onboard configuration for each satellite in each time period. In all cases, our procedure achieved results that were between 40% and 60% better than previous period-by-period practices (which might have been poorer than the period-by-period approach specified here). These improvements represented a potential operational cost reduction equivalent to roughly $200 million.

## 5. Concluding remarks

In this paper we described a multi-period traffic routing and planning problem that appears in geosynchronous satellite networks. Because of the rerouting penalties that are imposed when a customer's route through the network changes the problem presents new challenges that, to our knowledge, have not been examined previously in the literature. We developed a BPC procedure that uses a path-based multi-commodity formulation to solve this problem. The key challenge in this procedure is the solution of the pricing problem. Standard techniques for column-generation in IMCF problems could not be used because of the rerouting penalties involved. Therefore, we devised a novel solution technique capable of generating new multi-period super-paths while taking into account rerouting penalties. The technique involves the solution of a $K$-shortest path problem for each time period in which a commodity has non-negative demand and the computation of a shortest path on a specially generated "pricing graph".

Our computational analysis focuses on the comparison of multi-period optimization with a period-by-period approach and the differences between a BPC algorithm and a "Root-Node" approach. Based on our interactions with a leading company in the satellite services industry we were able to generate problem sets that mimic the characteristics of real-life networks. Our results indicate that a multi-period optimization algorithm can result in cost savings up to 13%. Even for networks with two satellites this corresponds to savings of millions of dollars. For large satellite providers with networks that consist of dozens of satellites the potential cost reduction could reach several hundred million dollars.

One possible extension to the MPNP problem for satellite networks would be to incorporate additional network design and planning decisions, such as satellite relocations and launches, in the optimization process. It is not uncommon for large satellite communication providers to own more orbital locations than satellites. Therefore, some operators have the ability to move satellites between longitudes in order to satisfy more demand and generate more revenue. Also, because of the limited lifespan of satellites service providers in the satellite industry constantly consider new launches. New launches or relocations of satellites can potentially have a dramatic impact on the routing of existing and future demands. Therefore it would be beneficial to consider these additional decisions in the same model when planning for satellite networks. This is part of our current research.

## Appendix. General rerouting penalties

We generalize the approach we developed in Section 3.2 to situations where the route change penalty is a function of the path followed in period $t-1$ and the path taken in period $t$. This allows us to address the situation of a much more general route change penalty cost. Unfortunately, Proposition 1 and Theorems 1 and 2 are not valid for these types of penalties.

We use the same notation as before, where $q_n^t$ denotes the $n^{\text{th}}$ shortest path in time period $t$, $R^t$ denotes the set of $K_t$-shortest paths in time period $t$ and $P^t$ denotes the set of all feasible paths in time period $t$. Also, let $e^l\left(q_i^{t-1}, q_j^t\right)$ denote the rerouting penalty of commodity (customer) $l$ that depends on path $q_i^{t-1}$ (the $i$th path in period $t-1$) and path $q_j^t$ (the $j$th path in period $t$). We specify the following sufficient condition, which is a generalization of the condition defined in Proposition 1, and can be used to determine whether a specific choice of $\{K_1, K_2, \ldots, K_T\}$ ensures that we have found the lowest cost super-path with the new rerouting penalties. This generalization states that if we can find a path $q_{n_t}^t$ in period $t$ that is lower in cost than the most expensive path (i.e., $q_{K_t}^t$) in that period by an amount greater than or equal to the greatest two penalties incurred when switching to $q_{n_t}^t$ from the previous and the next period then $G'$ will contain the lowest cost super-path.

[4] We were only able to use our Root-Node procedure on real-world instances because of licensing restrictions with the Maestro libraries.

**Theorem 3.** *The pricing graph $G'$ contains a lowest cost super-path p, if $\exists n_t \in \{1, \ldots, K_t - 1\}$, such that $h\left(q^t_{K_t}\right) - h(q^t_{n_t}) \geqslant \max_{i=1,\ldots,K_{t-1}} e^l(q^{t-1}_i, q^t_{n_t}) + \max_{i=1,\ldots,K_{t+1}} e^l\left(q^t_{n_t}, q^{t+1}_i\right)$ or $R^t = P^t$, for each $t = 1, \ldots, T$.*

**Proof.** Suppose not. Then for some time period $t$, $R^t \neq P^t$ because otherwise the pricing graph $G'$ will contain all feasible paths and therefore the lowest cost super-path. Let $p^*$ be a lowest cost super-path. Then for some time period $r$ (for which $R^r \neq P^r$), $p^*$ contains a path $q^r_{j_{p^*}}$ distinct from $q^r_1, \ldots, q^r_{K_r}$, and therefore $h\left(q^r_{j_{p^*}}\right) \geqslant h(q^r_{K_r})$. Here $q^r_{j_{p^*}}$ tells us the path used in time period $r$ by the lowest cost super-path $p^*$; with $j_{p^*}$ providing the path number. Let

$$j^r_\alpha = \max\left\{i : 0 \leqslant i \leqslant T - t \text{and } q^r_{j_{p^*}}, F^{r+1}\left(q^r_{j_{p^*}}\right), \ldots, F^{r+i}\left(q^r_{j_{p^*}}\right) \in p^*\right\},$$

and

$$j^r_\beta = \max\left\{i : 0 \leqslant i \leqslant t - 1 \text{and } F^{r-i}\left(q^r_{j_{p^*}}\right), \ldots, F^{r-1}\left(q^r_{j_{p^*}}\right), q^r_{j_{p^*}} \in p^*\right\}.$$

By replacing paths $F^i\left(q^r_{j_{p^*}}\right)$ by path $q^i_{n_i}$ for $i = r - j^r_\beta, \ldots, r, \ldots, r + j^r_\alpha$ in super-path $p^*$ we can get a super-path with cost less than or equal to $p^*$. Specifically, by switching to paths $q^i_{n_i}$ we incur the following penalty costs,

$$e^l\left(q^{r-j^r_\beta-1}_{j_{p^*}}, q^{r-j^r_\beta}_{n_{r-j^r_\beta}}\right) + e^l\left(q^{r+j^r_\alpha}_{n_{r+j^r_\alpha}}, q^{r+j^r_\alpha+1}_{j_{p^*}}\right) + \sum_{t=r-j^r_\beta+1}^{r+j^r_\alpha} e^l\left(q^{t-1}_{n_{t-1}}, q^t_{n_t}\right) \tag{20}$$

(i.e., $j^r_\alpha + j^r_\beta + 2$ penalties). Since,

$$h\left(q^t_{K_t}\right) - h\left(q^t_{n_t}\right) \geqslant \max_{i=1,\ldots,K_{t-1}} e^l\left(q^{t-1}_i, q^t_{n_t}\right) + \max_{i=1,\ldots,K_{t+1}} e^l\left(q^t_{n_t}, q^{t+1}_i\right) \quad \text{for } t+1, \ldots, T$$

and $h\left(q^t_{j_{p^*}}\right) \geqslant h\left(q^t_{K_t}\right)$ for $t = r - j^r_\beta, \ldots, r + j^r_\alpha$ we can write for the difference between the cost of the old and the cost of the new paths on the super-path the following,

$$\sum_{t=r-j^r_\beta}^{r+j^r_\alpha} \left(h\left(q^t_{j_{p^*}}\right) - h\left(q^t_{n_t}\right)\right) \geqslant \sum_{t=r-j^r_\beta}^{r+j^r_\alpha} \left(max_{i=1,\ldots,K_{t-1}} e^l\left(q^{t-1}_i, q^t_{n_t}\right) + \max_{i=1,\ldots,K_{t+1}} e^l\left(q^t_{n_t}, q^{t+1}_i\right)\right)$$

which amounts to $2\left(j^r_\alpha + j^r_\beta + 1\right)$ penalties that are the *greatest* possible penalties (when switching to path $q^t_{n_t}$) between any two periods and is therefore *greater than or equal to* the $j^r_\alpha + j^r_\beta + 2$ *specific* penalties stated previously in Eq. (20). Therefore the new super path will have cost less than or equal to $p^*$. □

For the MPNP problem we can define a general penalty as,

$$\tilde{e}^l_{tp} = e^l_t \gamma^p_t + \sum_{s \in S_t} \sum_{u \in U^b} \beta^{su}_{pt} \theta^{su}_{lt}.$$

Notice that $\tilde{e}^l_{tp}$, unlike the rerouting penalties we considered in Section 3.2, can assume different values even when solving the pricing problem of a specific commodity and for the same time period $t$. Therefore for $\tilde{e}^l_{tp}$ we would have to use Theorem 3 to solve the pricing problem.

## References


Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms and Applications. Prentice Hall.

Alvelos, F., Valério de Carvalho, J.M., 2001. Solving multicommodity flow problems with branch-and-price. Investigação Operational 21 (1), 71–92.

Banerjee, D., Mukherjee, B., 2000. Wavelength-routed optical networks: linear formulation, resource budgeting tradeoffs, and a reconfiguration study. IEEE/ACM Transactions on Networking 8 (5), 598–607.

Barnhart, C., Hane, C.A., Vance, P.H., 2000. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. Operations Research 48 (2), 318–326.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H., 1998. Branch and price: column generation for solving huge integer programs. Operations Research 46 (3), 316–329.

Bienstock, D., Günlük, O., 1996. Capacitated network design – polyhedral structure and computation. INFORMS Journal on Computing 8 (3), 243–259.

Danna, E., Pape, C., 2005. Branch-and-price-heuristics: a case study on the vehicle routing problem with time windows. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation. Springer, pp. 99–129.

Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), 2005. Column Generation. Springer.

Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F., 1995. Time constrained routing and scheduling. In: Ball, M.O. et al. (Eds.), Handbooks in Operations Research and Management Science, vol. 8. Elsevier, pp. 35–139.

Frantzeskakis, L.F., Luss, H., 1999. The network redesign problem for access telecommunications networks. Naval Research Logistics 46 (5), 487–506.

Gamvros, I., 2006. Satellite network design, optimization and management. Ph.D. Thesis, Univeristy of Maryland, College Park.

Girard, A., Lansard, P.D., Liau, B., Mongeon, A.M., Thibault, J.L., 1991. Multihour engineering and traffic routing optimization in hierarchical telephone networks. Annals of Telecommunications 46 (5–6), 335–350.

Gu, Z., Nemhauser, G.L., Savelsbergh, M.W.P., 1998. Lifted cover inequalities for 0–1 integer programs: computation. INFORMS Journal on Computing 10 (4), 427–437.

Gu, Z., Nemhauser, G.L., Savelsbergh, M.W.P., 2000. Sequence independent lifting in mixed integer programming. Journal of Combinatorial Optimization 4 (1), 109–129.

Günlük, O., 1999. A branch-and-cut algorithm for capacitated network design. Mathematical Programming 86 (1), 17–39.

Holmberg, K., Yuan, D., 2003. A multicommodity network-flow problem with side constraints on paths solved by column generation. INFORMS Journal on Computing 15 (1), 42–57.

Labourdette, J.-F.P., 1998. Traffic optimization and reconfiguration management of multiwavelength multihop broadcast ligthwave networks. Computer Networks and ISDN Systems 30 (9-10), 981–998.

Labourdette, J.-F.P., Hart, G.W., Acampora, A.S., 1994. Branch-exchange sequences for reconfiguration of ligthwave networks. IEEE Transactions in Communications 42 (10), 2822–2832.

Lardeux, B., Nace, D., Geffard, J., 2007. Multperiod network design with incremental routing. Networks 50 (1), 109–116.

Maral, G., Bousquet, M., 1998. Satellite Communications Systems, third ed. Wiley.

Medhi, D., Tipper, D., 2000. Some approaches to solving a multihour broadband network capacity design problem with single-path routing. Telecommunication Systems 13 (2–4), 269–291.

Savelsbergh, M.W.P., 1997. A branch-and-price algorithm for the generalized assignment problem. Operations Research 45 (6), 831–841.

SIA, 2010. State of the satellite industry report. Tech. Rep., Satellite Industry Association. <http://www.sia.org/>.

Vance, P.H., Barnhart, C., Johnson, E., Nemhauser, G., 1997. Airline crew scheduling: a new formulation and decomposition algorithm. Operations Research 45 (2), 188–200.

Vanderbeck, F., 2000. On Dantzig–Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. Operations Research 48 (1), 111–128.

Vanderbeck, F., Wolsey, L.A., 1996. An exact algorithm for IP column generation. Operations Research Letters 19 (4), 151–159.