Exact Solution to Bandwidth Packing Problem with Queuing Delays

Navneet Vidyarthi[a,b,*], Sachin Jayaswal[c], Vikranth Babu Tirumala Chetty[c]

[a] *Department of Supply Chain and Business Technology Management*
John Molson School of Business, Concordia University, Montreal, QC, H3G 1M8, Canada
[b] *Interuniversity Research Center on Enterprise Networks, Logistics, and Transportation (CIRRELT)*

[c] *Indian Institute of Management, Vastrapur, Ahmedabad, Gujarat, 380 015 India*

Abstract

The bandwidth packing problem seeks to select and route a set of calls from a given list, each with a pre-specified requirement for bandwidth, on an undirected communication network such that the revenue generated is maximized. In this paper, we present a model and an exact solution approach for the bandwidth packing problem with queuing delay costs under stochastic demand and congestion. We provide a more general model than available in the extant literature by assuming a general service time distribution on the links. The problem, under Poison call arrivals, is thus set up as a network of spatially distributed independent M/G/1 queues. However, the presence of delay cost in the objective function makes the resulting integer programming model nonlinear. We present an exact solution approach based on piecewise linearization and cutting plane algorithm. Computational results indicate that the proposed solution method provides optimal solution in reasonable computational times. Comparisons of our exact solution method with the Lagrangean relaxation based solution reported in the literature for the special case of exponential service times clearly demonstrate that our solution approach outperforms the latter, both in terms of the quality of solution and computational times. Using numerical examples, we demonstrate that the service time variability, if not correctly represented in the model, can result in a solution very different from the optimal.

Keywords: Bandwidth Packing; Telecommunications; Integer Programming; Queuing Delay; Linearization; Exact Approach; Cutting Plane Method

---

*Corresponding author, Phone: +001-514-848-2424x2990, Fax: +001-514-848-2824

Email addresses: `navneetv@jmsb.concordia.ca` (Navneet Vidyarthi), `sachin@iimahd.ernet.in` (Sachin Jayaswal), `tvikranth@iimahd.ernet.in` (Vikranth Babu Tirumala Chetty)

1. Introduction

Technological improvements in the telecommunications industry have led to a massive growth of services like video-conferencing, social networking, collaborative computing, etc. At the same time, the arrival of cheaper and smarter devices have resulted in demand for faster and better services from the providers. This has increased the pressure on telecommunication firms to efficiently manage their limited bandwidth to provide satisfactory end-user services. In this context, one of the fundamental problems that arises is the Bandwidth Packing Problem (BPP). The BPP can be stated as: given a set of calls, and their associated potential revenues and bandwidth requirements (demand), arising at an instant on a telecommunication network with limited bandwidth on its links, (i) decide which of these calls to accept/reject, and (ii) select a single path (sequence of links) to route each selected call, such that the total revenue generated from the accepted calls is maximized without violating the bandwidth capacities on the links (Cox et al., 1991).

Several variants of BPP have been studied in the literature. For example, Amiri and Barkhi (2000) present multi-hour BPP to account for the variation in traffic between peak and off-peak hours of the day. Another version of BPP that involves scheduling of the selected calls within given time windows is reported by Amiri (2005). Amiri and Barkhi (2012) present an extension of BPP that has applications in telecommunication services like video conferencing and collaborative computing. They consider a case wherein each request from users consists of a set of calls between various pairs of nodes, and a request cannot be partially accepted/rejected. Recently, Bose (2009) has studied another version of the problem wherein the calls belong to two priority classes: the calls belonging to the higher priority class are shorter in length and generate more revenue but consume more bandwidth compared to the calls belonging to the lower priority class.

Other extensions of BPP account for the delays arising as a result of calls waiting at nodes due to congestion on the links. Excessive delays may arise if the solution to BPP, or its variants, result in certain links getting utilized close to their bandwidth capacities. Explicit consideration of such delays in the modeling and solution of BPP is important to guarantee quality service to customers. Amiri et al. (1999), Rolland et al. (1999), and Han et al. (2012) explicitly account for such network delays due to congestion by incorporating queuing delay terms in their model. All of these papers model the links on the network as a network of independent M/M/1 queues with the implicit underlying assumption that call arrivals are Poisson and their service times on links have exponential distribution. Amiri et al. (1999) discourage such delays in their model by penalizing them in the objective function, while Rolland et al. (1999) and Han et al. (2012) impose a constraint to limit such delays. Bose (2009) extends the problem to a setting where calls may be classified into different priority classes. For this, he models each link as a preemptive priority M/M/1 queue. Amiri (2003) extends the multi-hour BPP, earlier studied by Amiri and Barkhi (2000), with delay guarantees. The problem presented by Gavish and Hantler (1983) is also related to BPP with delays due to congestion, although the acceptance/rejection of calls is not a decision in their problem.

The single path requirement in BPP, which arises in various telecommunication services like video teleconferencing, etc., makes the problem NP-hard (Parker and Ryan, 1993). As such, various solution methods are presented in the literature. Anderson et al. (1993); Laguna and Glover (1993), for instance, use Tabu Search metaheuristic, while Cox et al. (1991) apply Genetic Algorithms. Lagrangean relaxation has been a popular choice of solution method in the literature, reported by Gavish and Hantler (1983), Rolland et al. (1999), Amiri et al. (1999), Amiri and Barkhi (2000), Amiri (2003), Amiri (2005), and Amiri and Barkhi (2012). Branch-and-Price and Column Generation is used by Parker and Ryan (1993), while Park et al. (1996) and Villa and Hoffman (2006) report the use of Branch-and-Price-and-Cut and Column Generation. Han et al. (2012) use Branch-and-Price technique with their Dantzig-Wolfe decomposition based reformulation of their model.

From the review of literature, we observe that all the studies on BPP that account for delays on telecommunication links due to congestion are based on the simplifying assumption that call arrivals are Poisson and service times on links have exponential distribution (Gavish and Hantler, 1983; Amiri et al., 1999; Rolland et al., 1999; Amiri, 2003; Bose, 2009; Han et al., 2012). This is primarily to make the problem, which is already otherwise NP-hard, tractable. The current study is an attempt to overcome this limitation in the extant literature by presenting a more generalized model. Through this work, we make the following contributions to the literature on BPP:

1. We present a generalized model for BPP with queuing delay costs, where the links in the network are modeled as independent M/G/1 queues.
2. Using simple transformation and piecewise linearization of queuing delay cost function, we linearize the model and present an efficient and exact approach based on cutting plane algorithm to solve the resulting model.

The remainder of the paper is organized as follows. In Section 2, we formally describe the problem and present its non-linear integer programming formulation. Section 3 describes an approach to linearize the model, followed by an exact solution methodology to solve the resulting mixed integer linear programming problem (MILP). Illustrative example, computational results, and insights are reported in Section 4. Section 5 concludes with some directions for future research.

2. Problem Formulation

We introduce the following notations used to describe the problem.

N	:	Set of nodes in the network
i, j	:	Indices for nodes in the network; $i, j \in N$
E	:	Set of undirected links in the network
(i, j)	:	Undirected links in the network; $i < j$
M	:	Set of calls
m	:	Index for a call; $m \in M$
$O(m)$	:	Origin node of call m; $O(m) \in N$
$D(m)$	:	Destination node of call m; $D(m) \in N$
d^m	:	Demand (bits per unit time) of call m
r^m	:	Potential revenue from call m
Q_{ij}	:	Bandwidth capacity of link (i, j)
$1/\mu$	:	Mean of message length
σ	:	Standard deviation of message length
cv	:	Coefficient of variation of message length; $cv = \mu\sigma$
c	:	Unit queuing delay cost per unit time

In line with the literature (Gavish and Hantler, 1983; Amiri et al., 1999; Rolland et al., 1999; Han et al., 2012), we assume that the arrivals of calls/messages on the network occur according to a Poisson process. Further, links are assumed to have finite capacities Q_{ij} for transmission of messages, and that nodes have unlimited buffers to store messages waiting for transmission. However, unlike the existing literature, we allow the message lengths (in bits) to follow a general distribution with a mean $1/\mu$, standard deviation σ, and coefficient of variation $cv = \mu\sigma$. The service rate (in bits per second) of the link (i, j) is proportional to the capacity of the link Q_{ij}. Then, the service time per message on link (i, j) also follows a general distribution with a mean $1/\mu Q_{ij}$, standard deviation σ/Q_{ij}, and coefficient of variation $cv = \mu\sigma$. Each link is thus modeled as a single server M/G/1 queue, and the telecommunication network is modeled as a network of independent M/G/1 queues.

Assume the bits composing message $m \in M$ arrive at a rate d^m per unit time. Further, let X_{ij}^m (X_{ji}^m) $= 1$ if call m is routed through link (i, j) in the direction from i to j (j to i), 0 otherwise. Then, the arrival of bits on link (i,j), due to superposition of Poisson processes, follows a Poisson process with a rate $\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)$ per unit time, and the arrival rate of messages per unit time on link (i,j) is $\lambda_{ij} = \mu\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)$. The average utilization of link (i,j) is given by:

$$\rho_{ij} = \frac{\lambda_{ij}}{\mu Q_{ij}} = \frac{\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)}{Q_{ij}} \tag{1}$$

Under steady state conditions ($\rho_{ij} < 1$) and first-come first-serve (FCFS) queuing discipline, the *mean sojourn time* (waiting time in queue + service time) of a message on link (i,j), which is modeled as an M/G/1 queue, is given by the Pollaczek-Khintchine (PK) formula as: $E[w_{ij}] = \left(\frac{1+cv^2}{2}\right)\frac{\lambda_{ij}}{\mu Q_{ij}(\mu Q_{ij}-\lambda_{ij})} + \frac{1}{\mu Q_{ij}}$. The expected network delay can be estimated as the weighted average of the expected delays on links: $\frac{1}{\Lambda}\sum_{(i,j)\in E}\lambda_{ij}E[w_{ij}]$, resulting in the following:

$$E[W] = \frac{1}{\Lambda}\sum_{(i,j)\in E}\left\{\left(\frac{1+cv^2}{2}\right)\frac{(\lambda_{ij})^2}{\mu Q_{ij}(\mu Q_{ij}-\lambda_{ij})} + \frac{\lambda_{ij}}{\mu Q_{ij}}\right\}, \tag{2}$$

where $\Lambda = \mu\sum_{m\in M} d^m$ is the total arrival rate of messages in the network. Substituting $\lambda_{ij} = \mu\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)$, as defined above, this can be further expressed as:

$$E[W] = \frac{1}{\Lambda}\sum_{(i,j)\in E}\left\{\left(\frac{1+cv^2}{2}\right)\frac{(\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m))^2}{Q_{ij}(Q_{ij} - \sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m))} + \frac{\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)}{Q_{ij}}\right\} \tag{3}$$

Using the above notations, the problem BPP under queuing delay that we study can be stated as follows: given a set of calls M, their associated potential revenues $(r^m, m\in M)$ and bandwidth requirements $(d^m, m\in M)$, arising at an instant on an undirected telecommunication network consisting of nodes N and links E with fixed arc/link capacities $(Q_{ij}, (i,j)\in E)$, determine a subset of calls $M' \subseteq M$ and a subset of $E' \subseteq E$ for each $m \in M'$, such that the total net revenue minus queuing delay costs is maximized. Let $Y^m = 1$ if call m is accepted, 0 otherwise, then the mathematical model for BPP with queuing delays can be stated as:

$[PN]$:

$$\max\ Z(\mathbf{X},\mathbf{Y}) = \sum_{m\in M} r^m Y^m - C \sum_{(i,j)\in E} \left\{ \left(\frac{1+cv^2}{2}\right) \frac{(\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m))^2}{Q_{ij}(Q_{ij} - \sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)} + \frac{\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)}{Q_{ij}} \right\} \quad (4)$$

$$\text{s.t.} \sum_{j\in N} X_{ij}^m - \sum_{j\in N} X_{ji}^m = \begin{cases} Y^m & \text{if } i = O(m); \\ -Y^m & \text{if } i = D(m); \\ 0 & \text{otherwise} \end{cases} \qquad \forall (i,j)\in E, m\in M \quad (5)$$

$$\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m) \le Q_{ij} \qquad \forall (i,j)\in E \quad (6)$$

$$X_{ij}^m \in \{0,1\} \qquad \forall (i,j)\in E, m\in M \quad (7)$$

$$Y^m \in \{0,1\} \qquad \forall m\in M \quad (8)$$

The first term in the objective function (4) is the total revenue from accepted calls. The second term captures the average queuing delay cost due to all accepted calls, where $C = c/\Lambda$ (a constant). Constraint set (5) are the flow conservation equations on each link for each call. Constraint set (6) ensures that the total demand on each link is less than its bandwidth capacity, required for the stability of the queue ($\lambda_{ij} \le \mu Q_{ij}$). Constraint sets (7) and (8) are binary restrictions on the variables. For $cv = 1$, the above formulation reduces to the $M/M/1$ model studied by Amiri et al. (1999) and others.

The formulation $[PN]$ is a non-linear integer program. In the following section, we present an approach to transform the above model, using auxiliary variables, into an MILP, and a cutting plane based method to solve it.

3. Solution Methodology

After rearranging the terms in (2), $E[W]$ can be rewritten as:

$$\begin{aligned} E[W] =& \frac{1}{\Lambda} \sum_{(i,j)\in E} \frac{1}{2} \left\{ \left(1+cv^2\right) \frac{\lambda_{ij}}{\mu Q_{ij} - \lambda_{ij}} + \left(1-cv^2\right) \frac{\lambda_{ij}}{\mu Q_{ij}} \right\} \\ =& \frac{1}{\Lambda} \sum_{(i,j)\in E} \frac{1}{2} \left\{ \left(1+cv^2\right) \frac{\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)}{Q_{ij} - \sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)} + \left(1-cv^2\right) \frac{\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)}{Q_{ij}} \right\} \end{aligned}$$

We define non-negative auxiliary variables R_{ij}, such that:

$$R_{ij} = \frac{\lambda_{ij}}{\mu Q_{ij} - \lambda_{ij}} = \frac{\sum_{m \in M} d^m (X_{ij}^m + X_{ji}^m)}{Q_{ij} - \sum_{m \in M} d^m (X_{ij}^m + X_{ji}^m)} \tag{9}$$

Then,

$$\sum_{m \in M} d^m (X_{ij}^m + X_{ji}^m) = \frac{R_{ij}}{1 + R_{ij}} Q_{ij} \tag{10}$$

Substituting (9) in the expression for $E[W]$ above gives:

$$E[W] = \frac{1}{\Lambda} \sum_{(i,j) \in E} \frac{1}{2} \left\{ \left(1 + cv^2\right) R_{ij} + \left(1 - cv^2\right) \frac{\sum_{m \in M} d^m (X_{ij}^m + X_{ji}^m)}{Q_{ij}} \right\}$$

We use the following lemma to linearize $[PN]$.

Lemma 1: *The function* $f(R_{ij}) = \frac{R_{ij}}{1+R_{ij}}$ *is concave in* $R_{ij} \in [0, \infty)$.

Proof:

Differentiating the function w.r.t. R_{ij}, we get the first derivative $\frac{\delta f}{\delta R_{ij}} = \frac{1}{(1+R_{ij})^2} > 0$, and the second derivative $\frac{\delta^2 f}{\delta R_{ij}^2} = \frac{-2}{(1+R_{ij})^3} < 0$, which proves that the function is concave in R_{ij} for $R_{ij} > 0$.

■

Lemma 1 implies that the function $f(R_{ij}) = \frac{R_{ij}}{1+R_{ij}}$ can be approximated by a large set of piecewise linear functions that are tangent to $f(R_{ij})$ at points $\{R_{ij}^h\}_{h \in H}$, such that:

$$\frac{R_{ij}}{1 + R_{ij}} = \min_{h \in H} \left\{ \frac{1}{(1 + R_{ij}^h)^2} R_{ij} + \frac{(R_{ij}^h)^2}{(1 + R_{ij}^h)^2} \right\}$$

This is equivalent to the following set of constraints:

$$\frac{R_{ij}}{1 + R_{ij}} \leq \frac{1}{(1 + R_{ij}^h)^2} R_{ij} + \frac{(R_{ij}^h)^2}{(1 + R_{ij}^h)^2}, \qquad \forall (i,j) \in E h \in H$$

Using (10), the above set of constraints can be rewritten as:

$$\sum_{m \in M} d^m (X_{ij}^m + X_{ji}^m) - \frac{Q_{ij}}{(1 + R_{ij}^h)^2} R_{ij} \leq \frac{Q_{ij} (R_{ij}^h)^2}{(1 + R_{ij}^h)^2} \qquad \forall (i,j) \in E, h \in H \tag{11}$$

provided $\exists h \in H$ such that (11) holds with equality.

The above substitutions result in the following linear MIP model:

$[PL(H)]:$

$$\max \sum_{m\in M} r^m Y^m - \frac{C}{2}\sum_{(i,j)\in E}\left\{\left(1+cv^2\right)R_{ij} + \left(1-cv^2\right)\frac{\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)}{Q_{ij}}\right\} \tag{12}$$

s.t. $(5)-(8),(11)$

$$R_{ij} \geq 0 \qquad \forall (i,j)\in E \tag{13}$$

For equivalence between $[PN]$ and $[PL(H)]$, there should exist at least one $h \in H$ such that (11) holds with equality. Proposition 1 confirms that there indeed exists one such $h \in H$ at optimality.

Proposition 1: *At least one of the constraints (11) in* $[PL(H)]$ *will be binding at optimality.*
Proof:
After rearranging the terms, (11) can be rewritten as:

$$R_{ij} \geq (1+R_{ij}^h)^2 \frac{\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)}{Q_{ij}} - (R_{ij}^h)^2 \tag{14}$$

Since R_{ij} appears in the objective function with a negative coefficient, $[PL(H)]$ attains its optimum value only when R_{ij} is minimized. This implies that $\forall (i,j) \in E$, $\exists h \in H$ such that (14) holds with equality if $(1+R_{ij}^h)^2 \frac{\sum_{m\in M} d^m(X_{ij}^m+X_{ji}^m)}{Q_{ij}} - (R_{ij}^h)^2 \geq 0$, else $R_{ij} = 0$. Further,

$$\begin{aligned}
0 &\leq (1+R_{ij}^h)^2 \frac{\sum_{m\in M} d^m(X_{ij}^m + X_{ji}^m)}{Q_{ij}} - (R_{ij}^h)^2 \\
&= (1+R_{ij}^h)^2 \rho_{ij} - (R_{ij}^h)^2 \quad \text{(using (1))} \\
&= (\rho_{ij}-1)(R_{ij}^h)^2 + 2\rho_{ij}R_{ij}^h + \rho_{ij} \\
&\Leftrightarrow R_{ij}^h \in \left[0, \frac{\rho_{ij}+\sqrt{\rho_{ij}}}{1-\rho_{ij}}\right] \ \forall h \in H \quad \text{(since } \rho_{ij} \leq 1 \text{ and } R_{ij} \geq 0 \text{ using (9))}
\end{aligned}$$

Thus, to prove that $\exists h \in H$ such that (11) holds with equality, we need to show that $R_{ij}^h \in \left[0, \frac{\rho_{ij}+\sqrt{\rho_{ij}}}{1-\rho_{ij}}\right]$. Since R_{ij}^h is an approximation to R_{ij}, we obtain:

$$\begin{aligned}
0 \leq R_{ij}^h \approx R_{ij} &= \frac{\lambda_{ij}}{\mu Q_{ij} - \lambda_{ij}} \qquad \text{(using (9))} \\
&= \frac{\rho_{ij}}{1-\rho_{ij}} \\
&\leq \frac{\rho_{ij}+\sqrt{\rho_{ij}}}{1-\rho_{ij}}
\end{aligned}$$

This proves that $\forall(i,j) \in E$, $\exists h \in H$ such that, at optimality, (11) always holds with equality. ■

Proposition 2: *For every subset of points $\{R_{ij}^h\}_{h\in H^q\subseteq H}$, $v(PL(H^q))$ is an upper bound to $[PL(H)]$, and hence to $[PN]$, where $v(\bullet)$ is the optimal objective function value of the problem $(\bullet)$.*

Proof:

Suppose, at any iteration, we use a subset of tangent points $\{R_{ij}^h\}_{h\in H^q\subseteq H}$, and solve the corresponding problem $[PL(H^q)]$, which yields the solution $(\mathbf{X}^q, \mathbf{Y}^q, \mathbf{R}^q)$ with the objective function value $v(PL(H^q))$. Since $[PL(H^q)]$ is a relaxation of the full problem $[PL(H)]$, $v(PL(H^q)) \geq v(PL(H))$, and hence $v(PL(H^q))$ provides an upper bound, given by:

$$UB = v(PL(H^q)) = \sum_{m\in M} r^m Y^{mq} - \frac{C}{2}\sum_{(i,j)\in E}\left\{\left(1+cv^2\right)R_{ij}^q + \left(1-cv^2\right)\frac{\sum_{m\in M} d^m(X_{ij}^{mq}+X_{ji}^{mq})}{Q_{ij}}\right\} \tag{15}$$

■

Proposition 3: For every subset of points $\{R_{ij}^h\}_{h\in H^q\subseteq H}$, the lower bound to $[PN]$ is given by:

$$\begin{aligned} LB =& Z(\mathbf{X^q}, \mathbf{Y^q}) = \sum_{m\in M} r^m Y^{mq} - C\sum_{(i,j)\in E}\left\{\left(\frac{1+cv^2}{2}\right)\frac{(\sum_{m\in M} d^m(X_{ij}^{mq}+X_{ji}^{mq}))^2}{Q_{ij}(Q_{ij}-\sum_{m\in M} d^m(X_{ij}^{mq}+X_{ji}^{mq})}\right. \\ &\left. + \frac{\sum_{m\in M} d^m(X_{ij}^{mq}+X_{ji}^{mq})}{Q_{ij}}\right\} \end{aligned} \tag{16}$$

where $(\mathbf{X}^q, \mathbf{Y}^q, \mathbf{R}^q)$ is the optimal solution to $[PL(H^q)]$.

Proof:

For every subset of points $\{R_{ij}^h\}_{h\in H^q\subseteq H}$, the solution $(\mathbf{X}^q, \mathbf{Y}^q, \mathbf{R}^q)$ to $[PL(H^q)]$ is also a feasible solution to $[PN]$, and hence the objective function (4) evaluated at the solution $(\mathbf{X}^q, \mathbf{Y}^q, \mathbf{R}^q)$, which is given by (16), gives a lower bound to $[PN]$. ■

3.1. Solution Algorithm

The model $[PL(H)]$ consists of a large number of constraints (11). However, not all of them need to be generated a priori. The solution algorithm starts with an initial subset $H^1 \subset H$. H^1 may be empty. However, our preliminary computational experiments show that starting with a non-empty H^1 helps in faster convergence of the algorithm. The resulting $[PL(H^1)]$ is solved, giving a solution $(\mathbf{X}^1, \mathbf{Y}^1, \mathbf{R}^1)$. The upper bound (UB^1) and the lower bound (LB^1) are computed using (15) and (16) respectively. The better of the last and the new lower bounds is retained as the new LB^1. If UB^1 equals LB^1 within some accepted tolerance (ϵ), then $(\mathbf{X}^1, \mathbf{Y}^1)$ is an optimal solution to $[PN]$, and the algorithm terminates. Else, a new set of points $R_{ij}^{h_{new}}$ is generated using the current solution (X^1, Y^1, R^1) as follows: $R_{ij}^{h_{new}} = \frac{\sum_{m\in M} d^m (X_{ij}^{m1} + X_{ji}^{m1})}{Q_{ij} - \sum_{m\in M} d^m (X_{ij}^{m1} + X_{ji}^{m1})}$. New cuts of the form (11) are generated using these points, and added to $[PL(H^1)]$ to arrive at $[PL(H^2)]$. Next, $[PL(H^2)]$ is solved, giving a new solution $(\mathbf{X}^2, \mathbf{Y}^2, \mathbf{R}^2)$ and UB^2. The new lower bound is obtained as the greater of LB^1 and $Z(\mathbf{X}^2, \mathbf{Y}^2)\}$. If UB^2 equals LB^2 within the set tolerance (ϵ), then the algorithm terminates with $(\mathbf{X}^2, \mathbf{Y}^2)$ as an optimal solution. Else, the process is repeated until UB^q equals LB^q within the set tolerance for some iteration q. The complete algorithm is outlined below:

Algorithm 1 Solution Algorithm for $[PL(H)]$

$q \leftarrow 1; UB^{q-1} \leftarrow +\infty; LB^{q-1} \leftarrow -\infty;$
Choose an initial set of points $\{R_{ij}^h\}_{h\in H^q}$ to approximate the function $R_{ij}/(1 + R_{ij})$ $\forall (i, j) \in E$.
while $(UB^{q-1} - LB^{q-1})/UB^{q-1} > \epsilon$ **do**
    Solve $[PL(H^q)]$ to obtain $(\mathbf{X}^q, \mathbf{Y}^q, \mathbf{R}^q)$.
    Update the upper bound: $UB^q \leftarrow v(PL(H^q))$.
    Update the lower bound: $LB^q \leftarrow \max\{LB^{q-1}, Z(\mathbf{X}^q, \mathbf{Y}^q)\}$.
    Compute new points: $R_{ij}^{h_{new}} = \frac{\sum_{m\in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})}{Q_{ij} - \sum_{m\in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})} \quad \forall (i, j) \in E$
    $H^{q+1} \leftarrow H^q \cup \{h_{new}\}$
    $q \leftarrow q + 1$
end while

Proposition 4: *Algorithm 1 to solve $[PL(H)]$ terminates in a finite number of iterations.*

Proof:

Given that $X_{ij}^m \in \{0, 1\}$ and $R_{ij} = \frac{\lambda_{ij}}{\mu Q_{ij} - \lambda_{ij}} = \frac{\sum_{m\in M} d^m (X_{ij}^m + X_{ji}^m)}{Q_{ij} - \sum_{m\in M} d^m (X_{ij}^m + X_{ji}^m)}$, the number of values that R_{ij} can take is finite. Therefore, in order to prove that Algorithm 1 is finite, it is sufficient to prove that the generated values of R_{ij}^h are not repeated.

Consider an iteration q, where Algorithm 1 has not yet converged, that is, $UB^q > LB^q$. Further, suppose $(\mathbf{X}^q, \mathbf{Y}^q)$ is the solution to $[PL(H^q)]$. Then, the new points $R_{ij}^{h_{new}}$ generated at iteration q are given by:

$$R_{ij}^{h_{new}} = \frac{\sum_{m\in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})}{Q_{ij} - \sum_{m\in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})} \qquad \forall (i, j) \in E$$

Suppose the values of $R_{ij}^{h_{new}}$ were already generated in one of the earlier iterations

$\forall (i,j) \in E$. Then:

$$(11) \Rightarrow \frac{R_{ij}^{h_{new}}}{1+R_{ij}^{h_{new}}} \leq \frac{1}{1+R_{ij}^{h_{new}}} R_{ij}^q + \left(\frac{R_{ij}^{h_{new}}}{1+R_{ij}^{h_{new}}}\right)^2 \qquad \forall (i,j) \in E$$
$$\Rightarrow R_{ij}^{h_{new}} \leq R_{ij}^q \qquad \forall (i,j) \in E$$

We now have:

$$\begin{aligned}
UB^q = & \sum_{m \in M} r^m Y^{mq} - \frac{C}{2} \sum_{(i,j) \in E} \left\{ \left(1+cv^2\right) R_{ij}^q + \left(1-cv^2\right) \frac{\sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})}{Q_{ij}} \right\} \\
\leq & \sum_{m \in M} r^m Y^{mq} - \frac{C}{2} \sum_{(i,j) \in E} \left\{ \left(1+cv^2\right) R_{ij}^{h_{new}} + \left(1-cv^2\right) \frac{\sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})}{Q_{ij}} \right\} \\
= & \sum_{m \in M} r^m Y^{mq} - \frac{C}{2} \sum_{(i,j) \in E} \left\{ \left(1+cv^2\right) \frac{\sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})}{Q_{ij} - \sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})} \right. \\
& \left. + \left(1-cv^2\right) \frac{\sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})}{Q_{ij}} \right\} \\
= & \sum_{m \in M} r^m Y^{mq} - C \sum_{(i,j) \in E} \left\{ \left(\frac{1+cv^2}{2}\right) \frac{(\sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq}))^2}{Q_{ij}(Q_{ij} - \sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})} \right. \\
& \left. + \frac{\sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})}{Q_{ij}} \right\} \\
\leq & max\left(LB^{q-1}, \sum_{m \in M} r^m Y^{mq} - C \sum_{(i,j) \in E} \left\{ \left(\frac{1+cv^2}{2}\right) \frac{(\sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq}))^2}{Q_{ij}(Q_{ij} - \sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})} \right. \right. \\
& \left. \left. + \frac{\sum_{m \in M} d^m (X_{ij}^{mq} + X_{ji}^{mq})}{Q_{ij}} \right\} \right) \\
= & LB^q
\end{aligned}$$

This contradicts our initial assumption $UB^q > LB^q$. Therefore, at a given iteration, at least one of the values of R_{ij}^h generated is different from all the previously generated values. Furthermore, the number of values that R_{ij}^h can take is finite, and hence the algorithm terminates in a finite number of iterations.

■

4. Computational Study

We report our computational experience with the solution methodology described in Section 3. The exact solution algorithm is coded in Visual C++, while $[PL(H^q)]$ at every iteration q is solved using IBM ILOG CPLEX 12.4. The experiments are conducted on a machine with the following specifications: Intel Core i5-3230M, 2.60 GHz CPU; 4.00 GB RAM; Windows 64-bit Operating System. In Section 4.1, using an illustrative example (adopted from Laguna and Glover, 1993) with 10 nodes and 20 calls, we demonstrate the impact of variability in service times of the links on the optimal selection of calls and their routes in the network. The computational performance of proposed solution approach on networks with varying sizes are presented in Section 4.2.

4.1. Illustrative Example

Figure 1 shows the network topology for a problem instance with 10 nodes. The bandwidth capacities (Q_{ij}) of different links on the network are given in Table 1. The call table listing the bandwidth requirements (d^m) and potential revenues (r^m) for 20 calls is shown in Table 2. The optimal solution obtained using the method described in Section 3 is presented in Table 3, which displays the optimal routing (collection of links) for each call that is accepted, as well as the total gross revenue (GR) and the total delay cost (DC) , for different values of coefficient of variation cv and unit delay cost (C).

Table 3 demonstrates that the value of cv plays an important role in the call selection. For example, for $C = 5$, Call-9 is *rejected* at $cv = 0.5$, but gets *accepted* at higher values of $cv = 1, 1.5, 2$. On the other hand, for $C = 5$, Call-11 is *accepted* at $cv = 0.5$, but gets *rejected* at higher values of $cv = 1, 1.5, 2$. Call-16 exhibits an even more interesting pattern: for $C = 15$, it is *accepted* at $cv = 0.5$; *rejected* at $cv = 1, 1.5$; and again *accepted* at $cv = 2$. However, for $C = 20$, Call-16 is *accepted* at $cv = 0.5$; *rejected* at $cv = 1$; *accepted* at $cv = 1.5$; and again *rejected* at $cv = 2$. Table 3 further suggests that cv also plays a vital role in the route selection for the selected calls. For example, for $C = 5$, Call-16 is routed using links $0-8$; $7-0$; $8-4$ at $cv = 0.5, 1, 2$. However, the same call is routed using links $0-8$; $2-0$; $7-2$; $8-4$ at $cv = 2$. Similar observations can be made for $\{Call - 12; C = 15\}$ and $\{Call - 19; C = 20\}$. These results demonstrate the fact that service time variability plays a vital role in the optimal call and route selections in BPP, which, in turn, effect the total net revenue. This example thus illustrates the importance of accurately modelling service time variability for BPP.

Table 1: Bandwidth Capacities (Q_{ij}) of Links (i, j) for the Illustrative Example

i	0	0	0	0	0	1	2	4	5	5	6	7
j	1	2	7	8	9	3	7	8	7	8	7	8
Q_{ij}	25	35	40	20	15	10	20	15	10	15	10	10

Figure 1: Network Topology for a 10-Node Illustrative Example

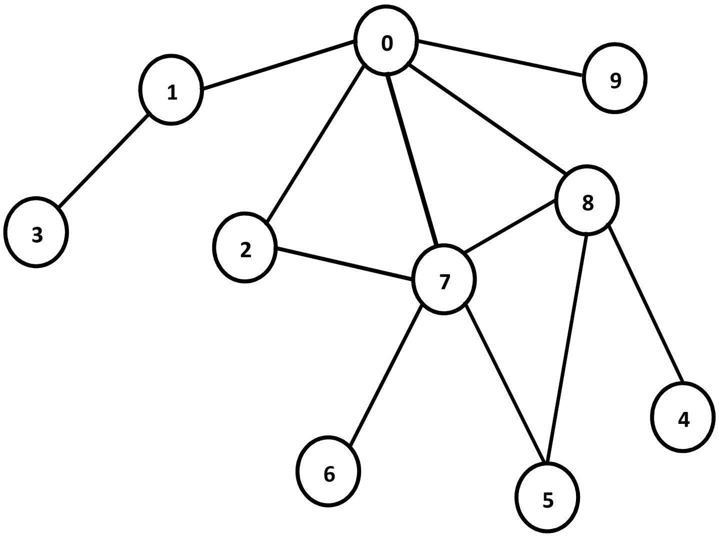

Table 2: Call Table for the Illustrative Example

Call m	Origin node $O(m)$	Destination node $D(m)$	Call demand d^m	Revenue r^m
1	0	2	10	420
2	0	7	7	380
3	0	5	6	400
4	0	4	6	390
5	1	6	5	500
6	1	5	5	490
7	1	4	7	400
8	2	9	2	150
9	2	3	4	450
10	2	4	8	500
11	3	5	6	850
12	5	2	3	200
13	6	9	5	370
14	7	1	6	500
15	7	9	5	340
16	7	4	2	120
17	8	1	6	460
18	8	2	8	450
19	9	5	5	360
20	9	1	5	170

Table 3: Solution Obtained for the Illustrative Example

	$C = 5$				$C = 10$				$C = 15$				$C = 20$			
	$\underline{cv}$				$\underline{cv}$				$\underline{cv}$				$\underline{cv}$			
Call m	0.5	1	1.5	2	0.5	1	1.5	2	0.5	1	1.5	2	0.5	1	1.5	2
1	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2	0-2
2	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7
3	0-7; 7-5	0-7; 7-5	0-7; 7-5	0-7; 7-5	0-7; 7-5	0-7; 7-5	0-8; 8-5	0-8; 8-5	0-7; 7-5	0-8; 8-5	0-8; 8-5	0-7; 7-5	0-7; 7-5	0-8; 8-5	0-7; 7-5	0-7; 7-5
4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4	0-8; 8-4
5	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6	0-7; 1-0; 7-6
6	0-8; 1-0; 8-5	0-8; 1-0; 8-5	0-8; 1-0; 8-5	0-8; 1-0; 8-5	0-8; 1-0; 8-5	0-8; 1-0; 8-5	0-7; 1-0; 7-5	0-7; 1-0; 7-5	0-8; 1-0; 8-5	0-7; 1-0; 7-5	0-7; 1-0; 7-5	0-8; 1-0; 8-5	0-8; 1-0; 8-5	0-8; 1-0; 8-5	0-8; 1-0; 8-5	0-8; 1-0; 8-5
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0	0-9; 2-0
9	-	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0	0-1; 1-3; 2-0
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	0-8; 1-0; 3-1; 8-5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	5-7; 7-2	5-7; 7-2	5-7; 7-2	5-7; 7-2	5-7; 7-2	5-7; 7-2	5-7; 7-2	5-7; 7-2	5-7; 7-2	5-7; 7-2	5-7; 7-2	0-2; 5-8; 8-0	5-7; 7-2	5-7; 7-2	0-2; 5-8; 8-0	0-2; 5-8; 8-0
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0	0-1; 7-0
15	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0	0-9; 7-0
16	0-8; 7-0; 8-4	0-8; 7-0; 8-4	0-8; 7-0; 8-4	0-8; 2-0; 7-2; 8-4	0-8; 7-0; 8-4	0-8; 7-0; 8-4	-	-	0-8; 7-0; 8-4	-	-	0-8; 2-0; 7-2; 8-4	0-8; 7-0; 8-4	-	0-8; 2-0; 7-2; 8-4	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7	7-2; 8-7
19	-	0-8; 8-5; 9-0	0-8; 8-5; 9-0	0-8; 8-5; 9-0	0-8; 8-5; 9-0	0-8; 8-5; 9-0	0-8; 8-5; 9-0	0-8; 8-5; 9-0	0-8; 8-5; 9-0	0-8; 8-5; 9-0	0-8; 8-5; 9-0	-	0-8; 8-5; 9-0	0-7; 7-5; 9-0	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Gross Revenue	5013	4948	4848	4707	4868	4747	4573	4368	4727	4563	4344	4073	4585	4407	4118	3842
Delay Cost	132	128	183	266	190	256	306	429	285	328	459	451	380	437	442	537

Figure 2 shows the effect of varying cv and C, over a wider range of values, on the total gross revenue, total delay cost and the total net revenue (NR = GR - DC) for the above illustrative example. The figures suggest that as cv or C increases, the total net revenue decreases. This is expected since a higher cv or C causes either (i) a higher congestion related cost if the set of accepted calls remains unchanged; or (ii) calls with lower potential revenue getting accepted if they are associated with lower bandwidth demands. In either case, the total net revenue is expected to decrease. Further, when a higher cv or C causes the former (i), then the total gross revenue is expected to remain unchanged. However, when it causes the latter (ii), then the total gross revenue is also expected to decrease with an increase in cv or C. Hence, the total gross revenue in Figure 2 either remains unchanged or decreases with an increase in cv or C. However, the change in the total delay cost, as cv or C increases, is non-monotonic. This, although appears counter-intuitive, can be explained as follows. When a higher cv or C does not cause any change to the set of accepted calls, then the delay cost is expected to increase. However, when a higher cv or C causes calls with lower bandwidth demands getting accepted, then the total delay cost is expected to decrease due to a decrease in congestion in the network.

4.2. Computational Results

For our computational study, we adopt the data generation scheme as reported by Amiri et al. (1999) to generate 10 sets of networks for each value of $|N| = \{10, 20, 30, 40, 50\}$. For each of these networks, a call table is generated for $P = \{50, 60, 70, 80, 90\}$, where P is the percentage of the maximum possible types of calls (a call type is specified by an origin-destination node pair) for the given network that are included in the call table. Thus, we have $10 \times 5 \times 5 = 250$ different problem sets. Each of these sets is solved for 4 different values of cv ($cv = 0, 0.5, 1, 1.5$) and for 5 different values of C ($C = 0.5, 1, 5, 10, 15, 20$), which together result in $250 \times 4 \times 5 = 5000$ problem instances. For each of the test instances, we start with a priori set (H^1) of points to approximate the function $f(R_{ij}) = R_{ij}/(1 + R_{ij})$ using its tangents $\widehat{f}(R_{ij})$ at these points. These points are generated such that the approximation error measured by $\widehat{f}(R_{ij}) - f(R_{ij})$ is at most 0.001 (Elhedhli, 2005). Our initial computational experiments reveal that starting with an a priori set of points (H^1) significantly improves the performance of the solution algorithm as it then requires fewer iterations/cuts and hence smaller CPU time for the algorithm to converge. The value of ϵ used in the convergence criterion is set at 10^{-6} in all the experiments.

The results of the computational experiments, which are averages over 10 different networks, are presented for each combination of $|N|$, P, C and cv in Table 4. The table reports

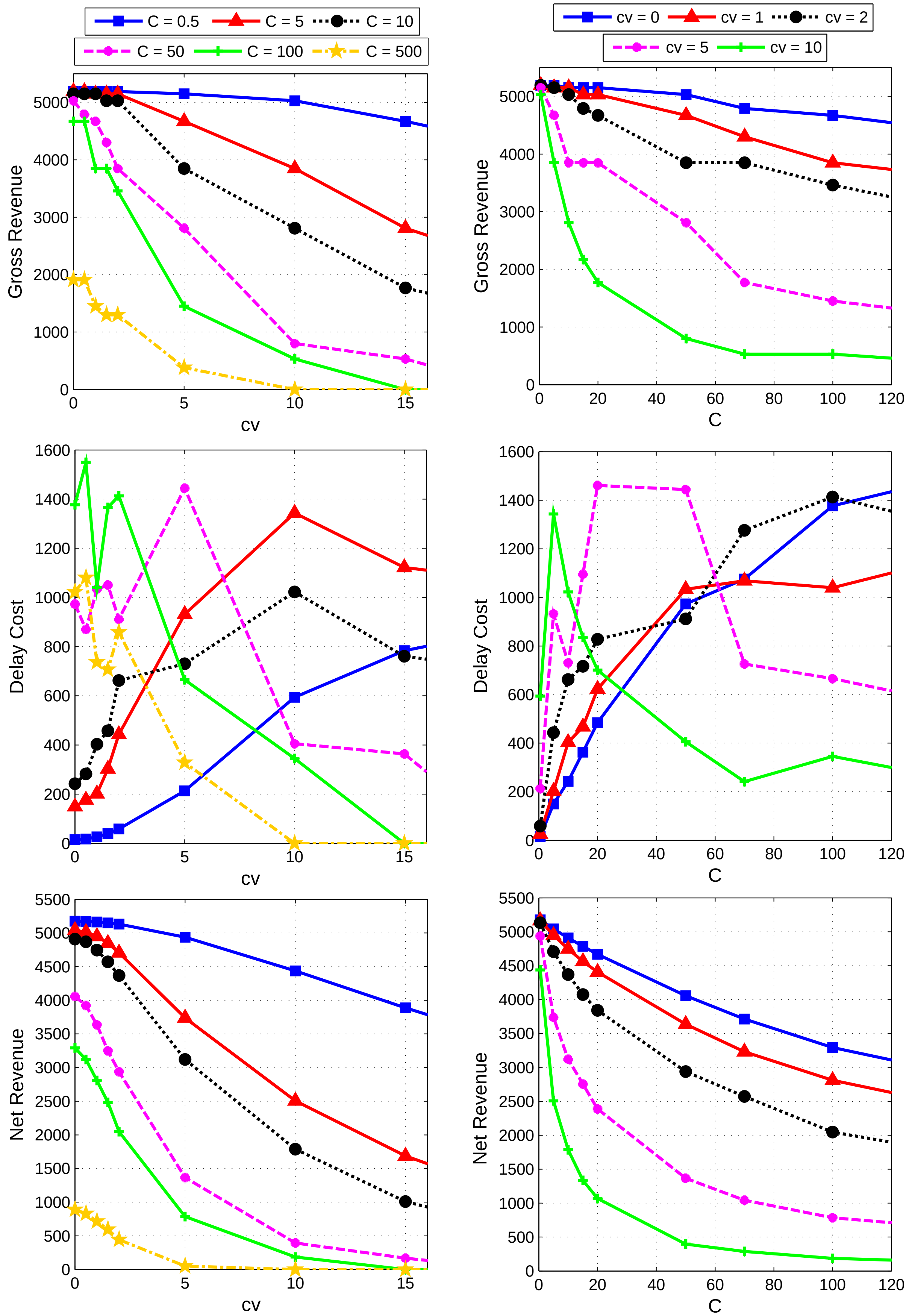

Figure 2: Gross Revenue (GR), Delay Cost (DC) and Net Revenue (NR) versus Coefficient of Variation of Service Times (cv) and Unit Delay Cost (C) for the Illustrative Example

the total gross revenue (GR), delay cost (DC) expressed as percentage of the total gross revenue, CPU time (in seconds), and the minimum, maximum, and the average link utilizations. The results clearly demonstrate the stability and the efficiency of our proposed exact solution method over a wide range of problem instances: it succeeds in finding optimal solutions to several instances with different unit delay costs and service time variability within a couple of minutes, with the maximum CPU time being 2153 seconds (for $|N| = 50$; $P = 90$; $C = 0.5$; $cv = 1$).

The efficiency of our solution algorithm is best highlighted by comparing its results, both the optimal objective function values and CPU times, with those from the Lagrangean relaxation based solution method reported by Amiri et al. (1999) for the special case of $cv = 1$ ($M/M/1$ queue model for the links). For the completeness of the paper, the mathematical model and the Lagrangean relaxation based solution algorithm reported by Amiri et al. (1999) are briefly presented in the appendix. The comparison of the the results are presented in Table 5, which demonstrates that our proposed solution method is, on an average, 3 to 10 times faster than the Lagrangean relaxation approach. Moreover, our proposed method solves the problem to optimality whereas the Lagrangean relaxation leaves an optimality gap of 2 to 7% on an average, and 11% in the worst case.

Table 4: Computational Results with Networks of Different Sizes, Unit Delay Cost, and Coefficient of Variation of Service Times

			$cv = 0$						$cv = 0.5$						$cv = 1$						$cv = 1.5$					
N	P	C	GR	DC(%)	CPU	% Util			GR	DC(%)	CPU	% Link Util			GR	DC(%)	CPU	% Link Util			GR	DC(%)	CPU	% Link Util		
						Min	Avg	Max				Min	Avg	Max				Min	Avg	Max				Min	Avg	Max
10	50	0.5	381	1.9	0.4	7	36	93	381	2.2	0.4	7	36	93	381	3.1	0.4	15	37	93	381	4.8	0.3	15	37	93
		5	361	10.4	0.2	7	28	83	361	11.8	0.2	7	28	83	360	16.1	0.3	7	28	83	336	15.9	0.5	7	25	78
		10	360	20.8	0.2	7	28	83	346	20.0	0.3	7	26	83	334	22.5	0.3	6	25	77	299	17.7	0.2	5	20	60
		15	335	23.7	0.2	5	25	78	330	25.1	0.2	6	24	77	299	20.9	0.2	5	20	60	298	26.3	0.1	6	20	59
		20	298	21.9	0.2	5	20	60	298	23.4	0.2	5	20	60	298	27.7	0.1	6	20	59	292	33.2	0.2	5	19	58
10	60	0.5	416	2.9	0.5	10	41	95	410	1.7	0.5	12	35	86	410	2.4	0.4	12	35	86	408	2.9	0.4	11	34	86
		5	405	11.4	0.1	11	34	86	405	13.0	0.2	11	34	86	403	17.4	0.3	11	34	85	357	11.6	0.3	10	25	69
		10	403	22.4	0.3	11	34	85	361	15.0	0.3	10	26	70	357	17.6	0.2	10	25	69	355	22.7	0.2	9	25	68
		15	357	19.6	0.2	10	25	69	357	21.3	0.1	10	25	69	355	25.8	0.1	9	25	68	318	25.4	0.3	6	21	59
		20	357	26.1	0.1	10	25	69	355	27.9	0.1	9	25	68	352	33.9	0.2	10	25	68	316	33.4	0.2	6	21	58
10	70	0.5	507	3.5	0.3	11	39	98	497	2.4	0.6	11	39	93	496	3.1	0.4	11	39	92	496	4.8	0.4	7	39	92
		5	474	9.9	0.4	7	34	86	464	9.2	0.3	11	31	82	461	11.6	0.3	10	31	81	456	15.1	0.4	10	30	80
		10	461	15.6	0.2	10	31	81	459	17.1	0.3	10	30	81	426	15.0	0.3	6	25	69	421	19.0	0.2	5	25	68
		15	428	16.9	0.3	6	25	70	426	18.1	0.2	6	25	70	421	21.6	0.2	5	25	68	384	21.0	0.3	6	21	59
		20	426	22.1	0.2	6	25	70	421	23.2	0.1	5	25	68	419	28.2	0.2	6	24	68	383	27.6	0.3	6	21	58
10	80	0.5	604	1.4	0.2	18	41	93	604	1.7	0.2	18	41	93	602	2.0	0.2	18	41	92	602	3.1	0.3	13	41	92
		5	584	9.4	0.4	14	39	87	582	10.4	0.4	13	38	86	567	11.3	0.3	13	37	80	549	12.1	0.3	8	35	70
		10	567	15.2	0.3	13	37	80	551	14.3	0.3	12	35	70	545	17.1	0.3	12	34	70	541	22.4	0.3	8	34	68
		15	549	19.1	0.3	12	35	70	545	20.3	0.2	12	34	70	541	25.0	0.3	12	34	68	463	21.1	0.4	6	24	59
		20	545	24.7	0.2	12	34	70	545	27.1	0.3	12	34	70	497	27.2	0.3	7	27	68	459	27.4	0.3	6	24	58
10	90	0.5	614	1.4	0.3	18	42	93	613	1.6	0.3	18	42	93	613	2.3	0.4	18	42	93	610	3.0	0.2	14	42	93
		5	594	8.5	0.3	14	39	82	593	9.4	0.3	12	39	81	593	12.7	0.3	12	39	81	559	11.9	0.3	8	35	70
		10	589	15.9	0.3	12	38	81	574	15.6	0.3	12	36	81	556	17.0	0.3	13	35	69	554	22.6	0.4	8	35	69
		15	559	18.8	0.3	12	35	70	556	20.2	0.3	13	35	69	554	25.2	0.3	13	35	69	469	20.3	0.4	6	23	59
		20	555	24.4	0.3	13	35	69	554	26.6	0.3	13	35	69	499	25.6	0.4	6	26	68	466	26.4	0.3	6	23	58
20	50	0.5	1131	4.6	8.3	15	54	98	1122	4.7	11.1	12	51	98	1097	3.8	6.4	14	49	94	1084	4.6	6.0	12	48	92
		5	1044	12.9	3.2	12	43	87	1028	13.3	4.2	12	42	84	1007	15.8	3.8	11	40	84	978	19.1	3.1	11	38	81
		10	1006	21.4	2.5	11	40	84	990	22.5	2.2	11	39	81	927	22.9	2.7	10	34	74	905	28.7	2.6	10	33	70
		15	943	26.3	1.9	10	35	75	925	27.2	2.1	10	33	74	906	32.5	2.3	10	33	70	826	34.8	2.4	8	28	66
		20	925	33.2	1.9	10	33	74	909	35.1	2.1	10	33	70	835	36.7	2.2	8	28	67	698	34.0	2.8	5	21	49
20	60	0.5	1326	3.7	12.7	17	55	98	1315	3.6	9.9	16	55	98	1297	3.6	9.1	16	53	94	1293	5.2	7.4	16	53	93
		5	1236	12.1	5.3	13	47	86	1216	12.1	4.3	13	45	84	1207	15.3	6.0	13	45	82	1170	17.9	4.1	15	42	78
		10	1182	18.9	3.0	13	42	81	1179	20.8	2.8	15	42	81	1129	22.8	4.8	12	39	73	1079	26.5	3.7	11	36	66
		15	1142	25.3	3.0	12	39	73	1116	26.0	3.1	10	37	73	1079	30.0	4.5	11	36	67	994	32.1	3.2	11	30	66
		20	1108	31.1	2.4	10	37	72	1077	32.0	3.4	10	35	71	994	33.1	4.1	11	30	66	880	33.2	3.7	8	24	52
20	70	0.5	1462	3.9	15.1	18	57	98	1453	4.0	13.4	18	57	98	1434	4.3	19.6	16	56	97	1417	5.1	10.5	16	54	95
		5	1349	11.0	7.1	16	48	87	1344	12.1	6.2	16	48	87	1309	13.2	6.0	16	44	80	1283	16.4	4.4	14	42	78
		10	1297	17.3	4.4	15	43	79	1296	19.2	4.3	15	43	79	1256	21.9	5.5	14	41	73	1211	26.3	6.4	13	38	70
		15	1271	24.2	3.0	14	41	78	1249	25.3	4.2	14	40	73	1212	29.6	5.9	11	37	72	1094	30.0	4.4	11	31	66
		20	1229	29.4	3.8	10	38	73	1214	31.3	4.0	11	37	72	1109	31.7	3.6	11	31	66	989	31.8	3.8	11	26	56

GR = Gross Revenue; DC(%) = Delay Cost expressed as a percentage of GR; CPU = Computation Time in seconds; % Link Util = % Utilization of Bandwidth Capacities on the Links

Table 4: Continued.

			cv = 0						cv = 0.5						cv = 1						cv = 1.5					
N	P	C	GR	DC(%)	CPU	% Util			GR	DC(%)	CPU	% Link Util			GR	DC(%)	CPU	% Link Util			GR	DC(%)	CPU	% Link Util		
						Min	Avg	Max				Min	Avg	Max				Min	Avg	Max				Min	Avg	Max
20	80	0.5	1636	4.3	29.3	17	63	98	1608	3.4	34.3	16	60	97	1602	4.6	40.7	15	60	97	1579	5.3	25.1	15	58	95
		5	1494	10.4	7.1	16	50	86	1475	10.6	7.5	16	48	82	1461	13.2	8.6	16	46	81	1418	15.5	5.7	14	44	75
		10	1457	17.6	5.4	16	46	81	1439	18.4	4.4	13	45	79	1386	20.2	5.3	14	41	73	1350	25.1	5.1	14	39	71
		15	1393	21.9	4.0	14	41	75	1386	23.7	3.9	14	41	73	1345	27.7	4.8	11	39	72	1225	28.4	5.9	11	32	66
		20	1375	28.1	3.5	11	40	72	1343	29.2	4.9	11	38	72	1249	30.5	4.3	11	33	66	1112	30.2	4.2	11	27	60
20	90	0.5	1762	3.6	29.5	20	65	98	1748	3.5	32.9	19	64	97	1733	4.3	40.3	18	62	97	1711	4.9	30.2	18	61	95
		5	1617	10.1	9.6	19	52	85	1603	10.7	7.6	19	51	83	1575	12.7	9.0	18	48	82	1540	15.5	6.5	17	46	76
		10	1575	17.1	5.4	19	48	82	1542	17.3	5.9	19	46	76	1510	20.5	6.9	17	43	75	1472	25.4	6.1	13	41	75
		15	1518	22.0	4.9	17	43	76	1490	22.8	4.1	15	42	75	1465	27.8	6.0	13	41	75	1327	28.3	6.0	13	34	66
		20	1490	27.6	4.4	15	41	75	1472	29.5	4.6	13	41	75	1349	30.0	6.3	14	35	66	1212	30.4	5.4	13	29	59
30	50	0.5	1599	3.2	28.7	5	48	97	1597	3.7	35.4	5	47	97	1582	4.3	43.0	5	47	96	1568	5.5	50.6	5	46	95
		5	1496	13.7	17.0	5	41	88	1478	14.6	18.4	5	39	88	1412	15.2	23.7	5	36	86	1350	16.3	16.2	4	33	78
		10	1406	20.2	17.2	5	35	84	1355	19.2	14.5	4	32	79	1297	20.1	16.7	4	30	72	1268	24.3	11.4	4	28	70
		15	1321	23.8	13.0	4	30	77	1291	23.9	9.5	4	29	72	1249	26.8	13.4	4	27	70	1157	28.0	21.4	4	23	57
		20	1263	27.7	9.0	4	27	70	1254	29.5	8.8	4	27	70	1161	29.7	13.8	4	23	59	1084	31.9	13.3	5	20	56
30	60	0.5	1758	2.5	44.4	7	48	96	1758	2.9	37.1	7	48	96	1753	4.1	47.6	7	48	95	1740	5.4	57.2	6	47	94
		5	1673	13.7	22.0	6	42	92	1638	13.7	21.8	6	41	88	1558	13.5	36.0	6	37	86	1507	15.2	26.6	6	34	79
		10	1552	18.1	18.7	6	35	86	1520	18.2	21.4	7	33	80	1465	19.7	21.7	7	31	73	1407	22.7	18.0	6	29	70
		15	1466	21.6	17.8	7	31	74	1461	23.3	10.9	6	30	73	1394	25.3	20.3	6	28	72	1294	26.1	19.8	5	23	57
		20	1454	28.1	10.8	7	30	73	1398	27.7	14.3	6	28	72	1314	28.4	17.8	5	24	60	1218	29.9	17.2	5	20	55
30	70	0.5	1945	2.9	40.6	7	52	96	1937	3.1	42.6	6	51	95	1932	4.4	87.7	6	51	95	1905	5.0	63.5	7	48	94
		5	1825	12.4	28.4	8	43	92	1792	12.5	30.2	8	42	87	1747	14.2	42.3	6	39	85	1689	16.5	35.3	6	37	84
		10	1732	18.3	21.6	7	38	85	1698	18.8	30.5	7	36	84	1611	19.3	31.2	6	33	74	1554	22.4	24.3	5	31	70
		15	1624	21.6	18.5	6	33	76	1600	22.4	18.9	6	32	73	1541	24.9	26.6	5	30	72	1428	25.5	25.1	6	25	58
		20	1586	26.7	11.9	6	31	73	1538	26.7	13.3	5	29	72	1447	27.6	26.2	5	25	59	1344	28.7	25.5	6	21	56
30	80	0.5	2120	2.6	69.9	8	52	98	2120	3.2	70.8	8	52	98	2106	3.9	97.8	10	52	95	2092	5.1	65.5	9	51	95
		5	2002	12.0	42.3	8	45	92	1967	12.1	34.7	8	44	87	1907	13.2	58.7	7	41	85	1871	16.4	47.9	7	40	82
		10	1898	17.4	34.4	8	40	85	1878	18.4	27.7	7	39	82	1770	18.3	53.5	7	34	74	1698	20.5	21.0	7	32	66
		15	1793	20.8	29.0	7	35	74	1770	21.8	28.3	7	34	74	1692	23.2	29.1	7	31	66	1600	25.2	23.9	7	28	58
		20	1741	25.2	17.6	7	32	74	1706	25.8	14.4	7	31	71	1620	27.2	27.9	7	28	61	1508	28.7	35.5	7	24	57
30	90	0.5	2362	3.0	165.4	9	57	98	2349	3.0	194.2	10	55	98	2335	3.8	197.1	11	55	95	2324	5.2	163.8	11	55	94
		5	2215	11.3	76.7	9	47	91	2156	10.5	60.0	8	45	86	2123	12.5	64.9	7	43	85	2076	15.0	70.0	8	41	83
		10	2116	16.5	37.0	8	42	85	2083	17.0	31.2	8	40	83	1995	17.8	42.5	8	37	75	1910	19.6	24.4	8	33	69
		15	2020	20.2	27.0	8	37	79	1989	20.9	22.4	7	36	75	1905	22.2	31.1	8	33	69	1800	24.0	30.1	7	29	60
		20	1952	23.9	23.9	8	34	73	1912	24.4	18.7	8	33	69	1827	26.1	32.0	7	30	64	1729	28.7	25.9	7	27	58
40	50	0.5	2285	3.6	140.9	9	58	98	2282	4.1	110.4	7	57	98	2256	4.8	240.0	8	56	97	2222	5.6	130.3	8	54	95
		5	2118	13.2	58.9	6	48	91	2081	13.3	49.2	6	46	89	2038	15.7	87.2	6	44	83	1954	17.7	47.8	7	40	78
		10	2016	20.3	39.3	7	42	83	1985	21.3	32.2	7	41	81	1920	24.3	59.1	7	39	77	1787	26.3	37.8	5	34	73
		15	1935	26.5	24.4	7	39	77	1899	27.8	26.0	7	38	77	1776	29.4	49.4	5	33	73	1569	28.1	36.0	5	25	63
		20	1850	31.7	25.5	6	35	77	1775	31.6	25.2	5	32	73	1584	30.0	60.0	5	25	64	1478	32.7	31.0	5	22	57

GR = Gross Revenue; DC(%) = Delay Cost expressed as a percentage of GR; CPU = Computation Time in seconds; % Link Util = % Utilization of Bandwidth Capacities on the Links

Table 4: Continued.

N	P	C	cv = 0 GR	DC(%)	CPU	% Util Min	Avg	Max	cv = 0.5 GR	DC(%)	CPU	% Link Util Min	Avg	Max	cv = 1 GR	DC(%)	CPU	% Link Util Min	Avg	Max	cv = 1.5 GR	DC(%)	CPU	% Link Util Min	Avg	Max
40	60	0.5	2455	3.6	181.1	10	59	98	2442	3.7	156.1	9	59	97	2414	4.3	295.4	10	56	96	2395	5.6	157.1	9	54	95
		5	2283	13.3	97.2	6	49	91	2249	13.8	72.6	6	47	89	2194	16.0	143.4	6	45	85	2086	17.2	106.7	7	41	79
		10	2161	19.9	62.8	7	43	84	2120	20.7	58.1	7	42	81	2024	22.3	58.7	7	38	76	1929	25.9	44.7	6	35	71
		15	2052	25.0	33.3	7	39	78	2026	26.5	28.7	7	38	77	1920	28.9	66.3	6	34	71	1714	28.4	50.2	5	27	64
		20	1974	30.2	26.9	6	35	76	1928	31.3	33.3	6	34	72	1719	29.8	65.5	5	27	65	1605	32.5	48.4	5	24	56
40	70	0.5	2706	3.0	282.7	10	61	97	2700	3.3	273.9	10	60	97	2675	3.9	280.4	11	58	96	2657	5.1	163.2	11	58	95
		5	2540	12.7	100.3	8	51	89	2502	13.1	97.6	9	50	87	2431	15.0	155.9	8	47	84	2339	17.2	65.8	7	44	78
		10	2406	19.2	72.7	7	46	84	2371	20.3	70.7	7	45	83	2280	22.7	84.1	6	42	76	2151	25.3	57.2	5	37	72
		15	2315	25.2	54.9	6	43	78	2262	26.0	50.3	6	41	77	2140	28.1	102.9	5	36	72	1947	29.0	58.1	6	30	65
		20	2204	29.4	46.6	5	38	77	2145	30.2	40.2	5	36	72	1968	30.7	95.8	6	30	67	1791	32.0	55.2	5	25	60
40	80	0.5	2911	2.9	208.9	12	63	97	2906	3.3	237.0	13	62	97	2884	4.0	324.4	13	61	96	2865	5.2	178.1	13	60	95
		5	2723	12.0	98.8	10	53	89	2706	13.1	102.1	11	52	88	2628	14.9	176.1	10	49	85	2534	17.3	96.4	9	46	80
		10	2609	19.1	96.8	9	48	85	2570	20.2	95.0	8	47	82	2464	22.4	135.4	8	43	78	2301	24.1	68.2	8	38	71
		15	2505	24.8	79.4	8	44	80	2427	25.0	48.2	8	41	77	2292	26.8	107.1	8	37	72	2127	29.1	61.0	7	33	65
		20	2387	28.9	54.2	7	40	77	2315	29.4	47.9	7	38	74	2149	30.6	106.7	7	33	67	1934	30.9	68.2	6	27	59
40	90	0.5	3139	3.1	324.3	12	65	98	3125	3.3	436.2	14	64	97	3101	3.9	441.4	14	63	96	3073	4.8	214.2	13	61	94
		5	2935	11.9	148.5	12	55	90	2903	12.6	168.8	12	54	88	2824	14.2	192.9	13	51	85	2729	16.4	120.9	11	47	80
		10	2799	18.1	97.8	12	49	84	2751	18.8	85.3	11	48	81	2659	21.3	148.9	10	45	78	2505	23.5	77.8	8	40	72
		15	2705	23.7	64.6	10	46	80	2628	24.0	61.3	8	43	76	2510	26.6	129.9	8	40	73	2324	28.5	73.8	8	35	65
		20	2602	28.3	50.0	8	42	76	2526	28.8	69.5	8	40	75	2357	30.2	109.2	8	35	67	2121	30.6	66.5	7	28	60
50	50	0.5	2927	3.7	1041.9	5	57	97	2916	4.1	1044.3	5	57	97	2885	4.9	1629.6	5	56	96	2846	6.0	670.6	5	55	95
		5	2692	14.4	323.6	6	48	90	2649	15.0	296.5	6	47	88	2550	16.3	372.8	5	43	86	2452	18.6	207.5	5	40	80
		10	2523	21.1	145.1	5	42	84	2464	21.5	162.6	5	41	80	2369	24.0	186.1	5	38	75	2234	26.7	121.3	6	34	69
		15	2408	27.1	84.4	5	38	78	2324	27.1	103.9	5	36	76	2212	29.8	163.3	6	33	70	2043	32.5	105.6	4	29	62
		20	2276	31.5	63.9	5	34	74	2223	32.6	72.3	6	33	72	2079	35.0	194.4	5	29	64	1867	36.7	102.4	5	24	59
50	60	0.5	3150	3.4	1111.6	8	59	97	3144	3.9	847.3	8	59	97	3117	4.9	1380.6	8	58	96	3074	5.9	757.3	8	56	95
		5	2903	13.9	345.4	7	49	89	2857	14.5	358.5	6	47	88	2739	15.3	584.3	5	44	85	2646	17.6	199.8	7	42	78
		10	2704	19.6	209.8	6	43	84	2672	20.8	194.2	6	42	82	2574	23.2	252.4	6	39	75	2428	26.0	167.5	6	35	70
		15	2596	25.7	142.9	6	39	76	2551	26.8	128.8	6	38	73	2423	29.4	313.2	6	34	71	2216	31.2	135.5	5	30	62
		20	2492	30.8	102.4	6	36	74	2441	32.2	105.0	6	35	72	2276	34.3	309.1	5	31	63	2028	35.1	224.1	5	25	60
50	70	0.5	3484	3.1	1196.1	8	60	97	3482	3.6	749.7	8	60	97	3458	4.6	1532.8	8	59	96	3417	5.8	688.0	8	58	95
		5	3232	13.6	420.7	7	51	90	3179	14.0	402.9	7	50	88	3068	15.4	635.8	8	47	85	2928	16.7	398.8	7	43	79
		10	3036	19.6	335.5	7	45	85	2969	20.0	278.3	6	44	81	2844	21.7	386.9	7	41	76	2738	25.7	202.5	6	38	72
		15	2889	24.6	207.8	6	41	78	2833	25.5	167.1	6	40	76	2735	29.0	332.2	7	37	72	2511	31.0	174.7	6	32	63
		20	2783	29.6	119.3	6	38	75	2741	31.2	114.8	7	37	74	2550	33.1	344.1	6	33	65	2291	34.4	203.8	6	27	60

GR = Gross Revenue; DC(%) = Delay Cost expressed as a percentage of GR; CPU = Computation Time in seconds; % Link Util = % Utilization of Bandwidth Capacities on the Links

Table 4: Continued.

N	*P*	C	cv = 0						cv = 0.5					
			GR	DC(%)	CPU	% Util			GR	DC(%)	CPU	% Link Util		
						Min	Avg	Max				Min	Avg	Max
50	80	0.5	3766	2.8	764.1	9	62	97	3764	3.4	735.1	9	62	97
		5	3524	13.3	432.4	6	54	90	3470	13.9	449.7	7	53	89
		10	3314	19.3	302.5	7	48	85	3253	20.0	282.5	7	46	82
		15	3180	25.0	266.1	6	44	80	3100	25.6	232.5	6	42	76
		20	3062	30.0	179.0	6	41	77	3002	31.4	182.6	6	40	74
50	90	0.5	4038	3.2	946.3	10	65	98	4033	3.7	1094.0	10	65	98
		5	3734	12.5	516.3	8	55	90	3702	13.6	444.9	8	54	90
		10	3547	19.2	325.3	7	50	86	3466	19.5	304.7	7	48	82
		15	3395	24.4	285.6	7	46	80	3316	25.2	268.0	7	44	78
		20	3250	28.8	228.8	7	42	77	3175	29.9	282.0	7	40	75

N	*P*	C	cv = 1						cv = 1.5					
			GR	DC(%)	CPU	% Link Util			GR	DC(%)	CPU	% Link Util		
						Min	Avg	Max				Min	Avg	Max
50	80	0.5	3743	4.4	1452.1	9	62	96	3702	5.6	1000.7	9	60	95
		5	3338	15.0	639.8	7	49	85	3219	17.3	361.6	7	46	81
		10	3126	22.3	479.5	7	43	77	2975	25.8	311.9	6	40	72
		15	2957	28.5	448.4	6	39	72	2706	30.2	330.9	6	34	64
		20	2763	32.6	518.4	6	34	66	2466	33.1	261.8	6	28	61
50	90	0.5	4003	4.7	2152.8	10	64	97	3939	5.3	914.5	10	62	95
		5	3572	15.0	698.1	8	51	86	3430	17.0	399.3	7	47	81
		10	3332	21.8	517.5	7	45	77	3150	24.5	408.2	7	40	72
		15	3142	27.4	565.4	7	40	72	2891	29.2	406.0	7	34	65
		20	2955	31.5	441.0	7	35	67	2672	32.7	309.6	6	29	62

GR = Gross Revenue; DC(%) = Delay Cost expressed as a percentage of GR; CPU = Computation Time in seconds; % Link Util = % Utilization of Bandwidth Capacities on the Links

Table 5: Comparison between the Proposed Exact Solution Method and Lagrangean Relaxation Method (for $cv = 1$)

		$\|N\| = 10$					$\|N\| = 20$					$\|N\| = 30$					$\|N\| = 40$					$\|N\| = 50$				
		Exact		LR			Exact		LR			Exact		LR			Exact		LR			Exact		LR		
P	C	NR	CPU_e	NR	Gap	$\frac{CPU_{lr}}{CPU_e}$	NR	CPU_e	NR	Gap	$\frac{CPU_{lr}}{CPU_e}$	NR	CPU_e	NR	Gap	$\frac{CPU_{lr}}{CPU_e}$	NR	CPU_e	NR	Gap	$\frac{CPU_{lr}}{CPU_e}$	NR	CPU_e	NR	Gap	$\frac{CPU_{lr}}{CPU_e}$
50	0.5	482	0.3	482	3.7	6.1	1385	5.8	1360	4.5	4.6	1360	64.4	1289	11.0	1.9	2638	336.8	2557	6.5	1.4	2728	927.4	2641	7.5	1.2
	5	373	0.1	373	0.4	10.4	1103	3.6	1101	1.6	7.1	1035	25.1	1027	6.0	4.6	2092	49.6	2079	2.9	9.8	2172	386.6	2125	5.9	3.1
	10	293	0.2	291	4.0	7.4	918	2.9	918	2.3	8.6	878	18.7	868	5.0	6.0	1778	55.9	1773	1.5	8.6	1866	276.7	1819	5.0	4.2
	15	247	0.1	247	1.6	9.7	790	2.7	787	3.3	11.5	766	14.5	763	3.9	7.7	1548	34.3	1548	1.5	14.2	1617	195.3	1599	4.2	5.9
	20	205	0.2	205	3.1	8.2	696	3.0	691	3.4	8.8	676	14.4	674	4.4	8.1	1371	32.2	1368	1.9	15.7	1430	200.9	1423	3.8	5.7
	Min		0.1		0.4	6.1		2.7		1.6	4.6		14.4		3.9	1.9		32.2		1.5	1.4		195.3		3.8	1.2
	Avg.		0.2		2.6	8.3		3.6		3.0	8.1		27.4		6.1	5.7		101.8		2.9	9.9		397.4		5.3	4.0
	Max.		0.3		4.0	10.4		5.8		4.5	11.5		64.4		11.0	8.1		336.8		6.5	15.7		927.4		7.5	5.9
60	0.5	550	0.3	541	6.3	5.4	1589	9.9	1559	5.7	3.4	1571	78.5	1502	9.4	1.8	2655	454.8	2525	9.4	1.3	3007	918.7	2926	6.9	1.6
	5	413	0.3	407	4.1	7.2	1281	4.2	1276	2.0	7.3	1220	73.7	1218	5.8	2.0	2083	275.4	2074	2.8	2.2	2403	512.0	2388	4.9	2.9
	10	334	0.2	332	3.3	9.1	1097	4.1	1094	1.6	7.4	1074	22.5	1064	3.8	6.7	1767	54.1	1752	2.7	11.1	2093	282.6	2057	4.7	5.0
	15	276	0.2	276	3.8	10.4	965	3.8	965	1.6	7.8	955	16.6	951	2.9	9.3	1539	54.5	1538	1.9	11.5	1847	225.2	1833	3.9	6.4
	20	224	0.2	224	7.1	7.5	855	3.3	853	2.4	9.0	853	14.5	853	3.1	11.1	1358	61.6	1355	2.8	10.2	1648	225.4	1634	4.1	6.3
	Min		0.2		3.3	5.4		3.3		1.6	3.4		14.5		2.9	1.8		54.1		1.9	1.3		225.2		3.9	1.6
	Avg.		0.2		4.9	7.9		5.1		2.7	7.0		41.2		5.0	6.2		180.1		3.9	7.2		432.8		4.9	4.5
	Max.		0.3		7.1	10.4		9.9		5.7	9.0		78.5		9.4	11.1		454.8		9.4	11.5		918.7		6.9	6.4
70	0.5	561	0.5	561	4.7	4.7	1719	16.3	1683	6.1	2.3	1689	53.3		8.0	3.3	2925	349.4	2869	6.5	2.1	3581	1634.4	3477	7.4	1.0
	5	423	0.3	418	3.5	8.3	1402	5.2	1402	1.4	6.9	1364	50.1	1359	3.4	3.5	2310	134.5	2298	3.0	5.9	2936	529.2	2901	4.3	3.2
	10	337	0.2	335	3.5	9.8	1207	4.8	1205	1.4	7.8	1167	26.2	1151	4.5	6.7	1961	71.1	1947	2.8	10.7	2564	267.4	2549	2.9	6.5
	15	276	0.2	276	4.0	9.5	1059	5.1	1059	1.7	7.1	1009	22.2	988	7.7	8.3	1697	64.2	1695	3.1	11.9	2258	282.5	2242	3.4	6.2
	20	224	0.3	224	7.1	7.8	939	5.5	932	3.1	6.4	881	25.9	871	9.9	7.2	1493	83.6	1485	4.0	9.7	2000	348.3	1988	4.7	4.9
	Min		0.2		3.5	4.7		4.8		1.4	2.3		22.2		3.4	3.3		64.2		2.8	2.1		267.4		2.9	1.0
	Avg.		0.3		4.6	8.0		7.4		2.7	6.1		35.5		6.7	5.8		140.6		3.9	8.1		612.3		4.5	4.4
	Max.		0.5		7.1	9.8		16.3		6.1	7.8		53.3		9.9	8.3		349.4		6.5	11.9		1634.4		7.4	6.5
80	0.5	597	0.5	594	5.0	5.0	1811	25.4	1762	7.0	1.7	1808	72.6	1795	6.6	2.8	3241	386.8	3177	6.2	2.1	3981	1857	3864	7.0	1.1
	5	450	0.3	444	3.2	7.0	1474	11.5	1473	1.7	3.6	1471	76.8	1466	3.5	2.7	2614	126.1	2614	2.3	6.4	3258	839.0	3219	4.2	2.3
	10	356	0.3	356	3.5	9.6	1282	4.6	1280	1.0	8.9	1262	29.9	1254	3.7	6.8	2216	187.0	2213	3.1	4.2	2820	807.4	2801	3.3	2.7
	15	293	0.3	293	4.2	11.0	1123	4.1	1122	1.3	10.1	1102	27.4	1080	6.7	7.4	1935	83.6	1930	4.1	9.4	2481	368.2	2456	4.3	5.4
	20	242	0.3	242	6.0	9.0	994	5.6	988	2.5	7.3	973	26.2	963	7.9	7.8	1725	78.5	1725	4.1	10.0	2215	532.9	2193	5.0	3.6
	Min		0.3		3.2	5.0		4.1		1.0	1.7		26.2		3.5	2.7		78.5		2.3	2.1		368.2		3.3	1.1
	Avg.		0.3		4.4	8.3		10.3		2.7	6.3		46.6		5.7	5.5		172.4		4.0	6.4		880.8		4.7	3.0
	Max.		0.5		6.0	11.0		25.4		7.0	10.1		76.8		7.9	7.8		386.8		6.2	10.0		1856.8		7.0	5.4
90	0.5	691	0.5	685	3.9	6.3	2010	47.3	1977	6.2	1.0	1897	193.1	1859	9.1	1.3	3476	519.4	3402	6.1	1.8	4141	1908	3978	7.8	1.2
	5	541	0.3	541	1.8	9.3	1665	6.8	1658	1.7	7.0	1537	49.2	1526	4.9	4.7	2815	160.3	2793	2.9	5.8	3345	786.2	3290	5.0	3.0
	10	451	0.3	450	2.2	8.8	1450	5.2	1445	1.2	9.0	1338	34.7	1335	2.7	6.8	2426	127.4	2418	2.3	7.4	2906	549.5	2875	3.8	3.9
	15	382	0.3	380	4.7	8.0	1281	5.0	1279	1.4	9.2	1184	26.1	1182	2.5	9.0	2149	88.4	2149	1.5	11.0	2585	589.4	2564	3.3	3.7
	20	332	0.3	332	5.3	8.6	1148	5.8	1141	2.2	7.9	1043	33.7	1033	4.9	7.2	1930	86.0	1929	1.4	11.5	2323	439.4	2305	3.4	5.1
	Min		0.3		1.8	6.3		5.0		1.2	1.0		26.1		2.5	1.3		86.0		1.4	1.8		439.4		3.3	1.2
	Avg.		0.3		3.6	8.2		14.0		2.5	6.9		67.3		4.8	5.8		196.3		2.8	7.5		854.5		4.7	3.4
	Max.		0.5		5.3	9.3		47.3		6.2	9.2		193.1		9.1	9.0		519.4		6.1	11.5		1908.2		7.8	5.1

NR = Net Revenue; Gap = % Gap between the Lagrangean Upper and Lower Bounds = $\frac{UB-LB}{LB} \times 100$; CPU_{lr} = Computation Time in seconds for the Lagrangean Method; CPU_e = Computation Time in seconds for the Proposed Exact Method

5. Conclusion

In this paper, we presented a model to analyze the impact of service time variability on the optimal call selection and routings in communication networks, commonly known as the bandwidth packing problem. We formulated a more generalized model of BPP with queuing delays by modelling the links, which process the calls arriving on the network, as $M/G/1$ queues. We presented a non-linear integer programming model, and linearized it using simple transformation and piecewise linear approximation. We further proposed an efficient solution approach, based on the cutting plane method, to solve the resulting linearized model to optimality. Through a computational study, we demonstrate the efficiency and the stability of the proposed solution algorithm in solving within minutes problem instances of the size of 50 nodes with varying service time variability delay costs. The proposed method also outperforms the Lagrangean relaxation approach, reported in the literature for the special case when services times on links are exponentially distributed.

The work reported in this paper can be extended in several ways. One such extension is to model the links as GI/G/1 queues, although the solution method for it is not immediately obvious. Another possible extension is to consider giving different priorities to calls from different classes of customers.

APPENDIX

We briefly present the mathematical model and the Lagrangian relaxation based solution approach reported by Amiri et al. (1999) for the special case when $cv = 1$ such that the links in the network are modeled as $M/M/1$ queues. For this, we introduce an additional set of variables W_{ij}^m as defined below:

$$W_{ij}^m = \begin{cases} 1 & \text{if call } m \text{ uses link}(i,j)\text{in either direction;} \\ 0 & \text{otherwise.} \end{cases}$$

The non-linear integer programming model of this problem is :

$$[P_{M/M/1}] : \max \sum_{m \in M} r^m Y^m - C \sum_{(i,j) \in E} \frac{\sum_{m \in M} d^m W_{ij}^m}{Q_{ij} - \sum_{m \in M} d^m W_{ij}^m} \tag{17}$$

$$\text{s.t. } X_{ij}^m + X_{ji}^m \leq W_{ij}^m \qquad \forall (i,j) \in E, m \in M \tag{18}$$

$$\sum_{m \in M} d^m W_{ij}^m \leq Q_{ij} \qquad \forall (i,j) \in E \tag{19}$$

$$W_{ij}^m, \in \{0,1\} \qquad \forall (i,j) \in E, m \in M \tag{20}$$

$$(5), (7), (8)$$

On dualizing the constraint set (18) using non-negative lagrangean multipliers α_{ij}^m $\forall (i,j) \in E$ and $m \in M$, the problem $[P_{M/M/1}]$ decomposes into two sets of subproblems: (i) $[L1_{LR}^m]$ $\forall m \in M$; and (ii) $[L2_{LR}^E]$ $\forall (i,j) \in E$, as given below:

$$[L1_{LR}^m] : \max \; r^m Y^m - \sum_{(i,j) \in E} \alpha_{ij}^m (X_{ij}^m + X_{ij}^m) \tag{21}$$

$$s.t. \; (5), (7), (8)$$

$$[L2_{LR}^E] : \max \sum_{m \in M} \alpha_{ij}^m W_{ij}^m - C \frac{\sum_{m \in M} d^m W_{ij}^m}{Q_{ij} - \sum_{m \in M} d^m W_{ij}^m} \tag{22}$$

$$s.t \; (19), (20)$$

The solution algorithms to solve $[L1_{LR}^m]$, LP relaxation of $[L2_{LR}^E]$ and to generate feasible solutions are presented below:

The pseudocode to solve the BPP using Lagrangian Relaxation method is outlined below:

Algorithm 2 Solution Algorithm for $[L1^m_{LR}]$

Solve $[L1^m_{LR}]$ a shortest path problem with α^m_{ij} as the link costs

if $(r^m > \sum_{(i,j)\in E} \alpha^m_{ij}(X^m_{ij} + X^m_{ij})$ **then**

$(Y^m = 1)$

else

$Y^m = 0$ and $X^m_{ij} = 0\ \forall (i,j) \in E$

end if

Algorithm 3 Solution Algorithm for LP Relaxation of $[L2^E_{LR}]$ for link (i,j)

Sort the calls $(m \in M)$ in non-increasing order of α^m_{ij}/d^m. Use index m' to represent the calls in this order.

$m' \leftarrow 0$

while $m' < |M|$ **do**

$m' \leftarrow m' + 1$

$S \leftarrow \sum_{k<m'} d^k W^k_{ij}$

$W_0 \leftarrow \min\left\{1, \frac{1}{d^{m'}}\left[(Q_{ij} - S) - \left(\frac{Cd^{m'}Q_{ij}}{\alpha^{m'}_{ij}}\right)^{1/2}\right]\right\}$

if $\alpha^{m'}_{ij} > 0$ and $W_0 > 0$ **then**

$W^{m'}_{ij} \leftarrow W_0$

else

$W^{m'}_{ij} \leftarrow 0$

end if

if $W^{m'}_{ij} < 1$ **then**

$W^{m'}_{ij} \leftarrow 0\ \ \forall \{k : m' < k \leq |M|\}$, and stop.

end if

end while

Algorithm 4 Solution Algorithm for Generating a Feasible Solution

$A_{ij} \leftarrow Q_{ij}\ \forall (i,j) \in E$
$DC \leftarrow 0$; $\Delta \leftarrow 0$
Sort the calls ($m \in M$) in non-increasing order of $v(L1^{m}_{LR})$ obtained from Algorithm 2. Use m' to represent the calls in this order.
Get the the values of $X^{m'}_{ij}\ \ \forall m' \in M,\ (i,j) \in E$ obtained using Algorithm 2
$m' \leftarrow 0$
while $m' < |M|$ **do**
  $m' \leftarrow m' + 1$
  if $d^{m'}(X^{m'}_{ij} + X^{m'}_{ji}) < A_{ij}\ \forall (i,j) \in E$ **then**
    $\Delta \leftarrow C \sum_{(i,j)\in E} \frac{\sum_{k' \le m'} d^{k'}(X^{k'}_{ij} + X^{k'}_{ji})}{Q_{ij} - d^{k'}(X^{k'}_{ij} + X^{k'}_{ji})} - DC$
    if $r^{m'} > \Delta$ **then**
      $Y^{m'} \leftarrow 1$
      $A_{ij} \leftarrow A_{ij} - d^{m'}(X^{m'}_{ij} + X^{m'}_{ji})\ \forall (i,j) \in E$
      $DC \leftarrow DC + \Delta$
    else
      $Y^{m'} \leftarrow 0$ and $X^{m'}_{ij} \leftarrow 0\ \forall (i,j) \in E$
    end if
  else
    $Y^{m'} \leftarrow 0$ and $X^{m'}_{ij} \leftarrow 0\ \forall (i,j) \in E$
  end if
end while

Algorithm 5 Lagrangean Relaxation Based Solution Method

$\alpha^{m}_{ij} \leftarrow 0\ \forall (i,j) \in E$ and $m \in M$; $UB \leftarrow +\infty; LB \leftarrow -\infty; iter \leftarrow 1; max\_iter \leftarrow 500; \epsilon \leftarrow 10^{-6}$
while $(UB - LB)/LB > \epsilon$ AND $iter < max\_iter$ **do**
  Solve $L1^{m}_{LR}\ \forall m \in M$ using Algorithm 2.
  Solve $L2^{E}_{LR}\ \forall (i,j) \in E$ using Algorithm 3.
  $UB \leftarrow \sum_{m \in M} v(L1_{LR}) + \sum_{(i,j)\in E} v(L2_{LR})$
  Generate a feasible solution using Algorithm 4
  $LB \leftarrow v(P_{M/M/1})$
  Update α^{m}_{ij} using sub-gradient method.
  $iter \leftarrow iter + 1$
end while

Acknowledgements

This research was supported by the National Science and Engineering Research Council of Canada (NSERC) grant to the first author, and by the Research & Publication Grant, Indian Institute of Management Ahmedabad, to the second author.

References

Amiri, A., 2003. The multi-hour bandwidth packing problem with response time guarantees. Information Technology and Management 4, 113–127.

Amiri, A., 2005. The selection and scheduling of telecommunication calls with time windows. European Journal of Operational Research 167, 243–256.

Amiri, A., Barkhi, R., 2000. The multi-hour bandwidth packing problem. Computer and Operations Research 27, 1–14.

Amiri, A., Barkhi, R., 2012. The combinatorial bandwidth packing problem. European Journal of Operations Research 208, 37–45.

Amiri, A., Rolland, E., Barkhi, R., 1999. Bandwidth packing with queuing delay costs: Bounding and heuristic procedures. European Journal of Operational Research 112, 635–645.

Anderson, C., Fraughnaugh, K., Parkner, M., Ryan, J., 1993. Path assignment for call routing: An application of tabu search. Annnals of Operations Research 41, 301–312.

Bose, I., 2009. Bandwidth packing with priority classes. European Journal of Operational Research 192, 313–325.

Cox, L., Davis, L., Qui, Y., 1991. Dynamic anticipatory routing in circuit-switched telecommunications networks, in: Davis, L. (Ed.), Handbook of Genetic Algorithms, Van Norstrand/Reinhold, New York. pp. 229–340.

Elhedhli, S., 2005. Exact solution of a class of nonlinear knapsack problems. Operations Research Letters 33, 615–624.

Gavish, B., Hantler, S., 1983. An algorithm for optimal route selection in sna networks. IEEE Transactions on Communications 31, 1154–1161.

Han, J., Lee, K., Lee, C., Park, S., 2012. Exact algorithms for a bandwidth packing problem with queueing delay guarantees. INFORMS Journal on Computing doi:10.1287/ijoc.1120.0523.

Laguna, M., Glover, F., 1993. Bandwidth packing: A tabu search approach. Management Science 39, 492–500.

Park, K., Kang, S., Park, S., 1996. An integer programming approach to the bandwidth packing problem. Management Science 42, 1277–1291.

Parker, M., Ryan, J., 1993. A column generation algorithm for bandwidth packing. Telecommunication Systems 2, 185–195.

Rolland, E., Amiri, A., Barkhi, R., 1999. Queueing delay guarantees in bandwidth packing. Computers and Operations Research 26, 921–935.

Villa, C., Hoffman, K., 2006. A column-generation and branch-and-cut approach to the bandwidth-packing problem. Journal of Research of the National Institute of Standards and Technology 111, 161–185.