

O.R. Applications

# The network design problem with relays

Edgar Alberto Cabral <sup>a</sup>, Erhan Erkut <sup>b,\*</sup>, Gilbert Laporte <sup>c</sup>,  
Raymond A. Patterson <sup>a</sup>

<sup>a</sup> School of Business, University of Alberta, Edmonton, Canada T6G 2R6

<sup>b</sup> Faculty of Business Administration, Bilkent University, 06800 Bilkent, Ankara, Turkey

<sup>c</sup> Canada Research Chair in Distribution Management, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Received 29 June 2005; accepted 20 April 2006

Available online 25 September 2006

## Abstract

The *network design problem with relays* (NDPR) is defined on an undirected graph  $G = (V, E, K)$ , where  $V = \{1, \dots, n\}$  is a vertex set,  $E = \{(i, j) : i, j \in V, i < j\}$  is an edge set. The set  $K = \{(o(k), d(k))\}$  is a set of *communication pairs* (or *commodities*):  $o(k) \in V$  and  $d(k) \in V$  denote the origin and the destination of the  $k$ th commodity, respectively. With each edge  $(i, j)$  are associated a cost  $c_{ij}$  and a length  $d_{ij}$ . With vertex  $i$  is associated a fixed cost  $f_i$  of locating a relay at  $i$ . The NDPR consists of selecting a subset  $\bar{E}$  of edges of  $E$  and of locating relays at a subset  $\bar{V}$  of vertices of  $V$  in such a way that: (1) the sum  $Q$  of edge costs and relay costs is minimized; (2) there exists a path linking the origin and the destination of each commodity in which the length between the origin and the first relay, the last relay and the destination, or any two consecutive relays does not exceed a preset upper bound  $\lambda$ . This article develops a lower bound procedure and four heuristics for the NPDR. These are compared on several randomly generated instances with  $|V| \leq 1002$  and  $|E| \leq 1930$ .

© 2006 Elsevier B.V. All rights reserved.

**Keywords:** Network design; Telecommunications; Column generation; Heuristic

## 1. Introduction

The *network design problem with relays* (NDPR) is defined on an undirected graph  $G = (V, E, K)$ , where  $V = \{1, \dots, n\}$  is a vertex set,  $E = \{(i, j) : i, j \in V, i < j\}$  is an edge set. The set  $K = \{(o(k), d(k))\}$  is a set of *communication pairs* (or *commodities*):  $o(k) \in V$  and  $d(k) \in V$  denote the origin and the destination of the  $k$ th commodity, respectively. With each edge

$(i, j)$  are associated a cost  $c_{ij}$  and a length  $d_{ij}$ . With vertex  $i$  is associated a fixed cost  $f_i$  of locating a relay at  $i$ . The NDPR consists of selecting a subset  $\bar{E}$  of edges of  $E$  and of locating relays at a subset  $\bar{V}$  of vertices of  $V$  in such a way that: (1) the sum  $Q$  of edge costs and relay costs is minimized; (2) there exists a path linking the origin and the destination of each commodity for which the length between the origin and the first relay, the last relay and the destination, or any two consecutive relays does not exceed a preset upper bound  $\lambda$ . This constraint distinguishes the NDPR from standard network design problems. We believe

\* Corresponding author. Tel.: +90 312 290 1276.

E-mail address: [erkut@bilkent.edu.tr](mailto:erkut@bilkent.edu.tr) (E. Erkut).

network design problems with relays have never previously been addressed, apart from two exceptions described below.

Fig. 1 depicts a graph where  $K = \{(1,7), (1,8), (2,8), (3,9)\}$  and  $\lambda = 5$ . The length of each edge is indicated. A feasible NDPR solution consists of selecting the bold edges  $(1,4)$ ,  $(1,5)$ ,  $(2,5)$ ,  $(3,6)$ ,  $(4,7)$ ,  $(5,8)$ , and  $(6,9)$  and of locating a relay at vertex 5.

This study is motivated by a telecommunications network design project in which the aim is to connect 422 communities in Alberta. In this problem there is a single origin and several destinations, so that the solution is a tree. The relays are *repeaters* which regenerate the signal(s) entering a vertex, and the value of  $\lambda$  is equal to 70 km.

The most recent reviews in network design problems are provided by Balakrishnan et al. (1997) and Raghavan and Magnanti (1997). Costa (2005) focuses his survey on Benders decomposition applied to fixed-charge network design problems, providing important insights in network design problem decomposition. One can trace a weak parallel between the NDPR and the *hop constrained network design problem* (HCNDP), where the number of arcs between the origin and the destination must respect a given threshold. Gouveia (1998) provides a survey on tree topology network design with hop-constraints. Voss (1999) considers a variant of the *Steiner tree problem* where hop-constraints are present, and proposes a solution heuristic based on tabu search. Gouveia and Requejo (2001) solve the hop-constrained minimum spanning tree problem using Lagrangean relaxation. Soni (2001) considers the HCNDP with partial survivability. De Giovanni et al. (2004) present a study on how hop constraints can impact network design solutions.

Generally, network design problems focus on arc costs and do not take into account the cost of installing equipment at the nodes. Tcha and Yoon (1995) constitute an exception: they discuss fixed

costs both along the edges and at the nodes and provide for signal bundling and switching. To model the problem, they assume that each region must have one hub assigned to it. This leads to a facility location model which is solved by a dual-based heuristic. Yoon et al. (1998) present an extension of this work. Examples of network design applications, but without optimization, are provided by Cosares et al. (1995), Cortes et al. (2001), and Davis et al. (2001).

The NDPR generalizes the *shortest path problem with relays* (SPPR) (Cabral et al., 2005) and the *weight constrained shortest path problem* (WCSPP) (Dumitrescu and Boland, 2003), both of which are NP-hard. The SPPR is a special case of the NDPR with  $|K| = 1$ , while the WCSPP consists of determining on a network a least cost path of length not exceeding  $\lambda$ .

The aim of this article is to develop a lower bound and four heuristics for the NDPR. The lower and upper bounding procedures are described in Sections 2 and 3, followed by computational results in Section 4 and conclusions in Section 5.

## 2. Lower bounding procedure

The lower bounding procedure we have developed for the NDPR is based on an integer linear programming formulation of the problem. As is often the case in combinatorial optimization, several formulations are available for the NDPR. We experimented with four different formulations (an undirected and a directed flow formulation; an undirected and a directed column generation formulation). After preliminary computational tests, we opted for a directed column generation formulation which seems easier to solve.

Because the problem is naturally undirected, it is necessary to double the number of commodities in order to arrive at a directed formulation. Hence, we redefine the set of commodities as  $K' = \{(o'(k), d'(k)), (o''(k), d''(k))\}$ , where  $(o'(k), d'(k)) = (o(k), d(k))$ , and  $(o''(k), d''(k)) = (d(k), o(k))$ , with  $(o(k), d(k)) \in K$ . Each edge  $(i, j) \in E$  is replaced with two opposite arcs  $(i, j)$  and  $(j, i)$ , with respective costs  $c'_{ij} = c'_{ji} = c_{ij}/2$  and respective lengths  $d'_{ij} = d'_{ji} = d_{ij}$ . Denote the set of arcs by  $A$ . The problem definition is otherwise unchanged.

To formulate the NDPR within a column generation framework, define for each commodity  $k \in K'$  the set  $P(k)$  of feasible paths from  $o(k)$  to  $d(k)$ ; given a path  $p \in P(k)$ , let  $R(p)$  denote the set of feasible relay patterns, i.e. an ordered subset of vertices on

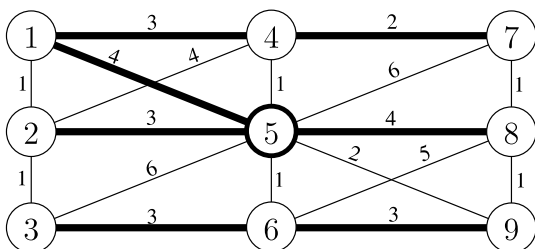


Fig. 1. Feasible NDPR solution.

$p$  separated by at most  $\lambda$  distance units, and let  $r \in R(p)$  be a feasible relay pattern for path  $p$ . Define the binary variables

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ belongs to the solution,} \\ 0 & \text{otherwise,} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if a relay is located at vertex } i, \\ 0 & \text{otherwise,} \end{cases}$$

$$z_k^{pr} = \begin{cases} 1 & \text{if path } p \text{ with relay pattern } r \text{ is used} \\ & \text{by commodity } k, \\ 0 & \text{otherwise,} \end{cases}$$

and the binary coefficients

$$a_{ij}^p = \begin{cases} 1 & \text{if arc } (i, j) \text{ belongs to path } p, \\ 0 & \text{otherwise,} \end{cases}$$

$$b_i^r = \begin{cases} 1 & \text{if a relay is located at vertex } i \text{ in relay pattern } r, \\ 0 & \text{otherwise.} \end{cases}$$

The master problem is then formulated as follows:

$$(MP) \text{ Minimize } \sum_{(i,j) \in A} c'_{ij} x_{ij} + \sum_{i \in V} f_i y_i \quad (1)$$

subject to

$$\sum_{\substack{p \in P(k) \\ r \in R(p)}} z_k^{pr} = 1 \quad (k \in K'), \quad (2)$$

$$\sum_{\substack{p \in P(k) \\ r \in R(p)}} a_{ij}^p z_k^{pr} \leq x_{ij} \quad ((i, j) \in A, k \in K'), \quad (3)$$

$$\sum_{\substack{p \in P(k) \\ r \in R(p)}} b_i^r z_k^{pr} \leq y_i \quad (i \in V, k \in K'), \quad (4)$$

$$x_{ij} - x_{ji} = 0 \quad ((i, j) \in A : i < j), \quad (5)$$

$$x_{ij} \geq 0 \quad ((i, j) \in A), \quad (6)$$

$$y_i \geq 0 \quad (i \in V), \quad (7)$$

$$z_k^{pr} \geq 0 \quad (k \in K', p \in P(k), r \in R(p)). \quad (8)$$

Denote by DMP the dual of MP. Now consider the restricted master problem RMP obtained by considering, for a given  $k \in K'$ , only a subset  $\bar{P}(k) \subseteq P(k)$  of feasible paths and, for a given  $p \in \bar{P}(k)$ , a subset  $\bar{R}(p) \subseteq R(p)$  of feasible relay patterns. The dual of RMP is denoted by DRMP and can be written as:

$$(DRMP) \text{ Maximize } \sum_{k \in K'} u_k \quad (9)$$

subject to

$$u_k - \sum_{(i,j) \in A} a_{ij}^p v_{ijk} - \sum_{i \in V} b_i^r w_{ik} \leq 0, \quad (k \in K', p \in \bar{P}(k), r \in \bar{R}(p)), \quad (10)$$

$$\sum_{k \in K'} v_{ijk} + \sum_{k \in K'} v_{jik} + q_{ij} \leq c'_{ij} \quad ((i, j) \in A : i < j), \quad (11)$$

$$\sum_{k \in K'} w_{ik} \leq f_i \quad (i \in V), \quad (12)$$

$$u_k \text{ unrestricted } (k \in K'), \quad (13)$$

$$v_{ijk} \geq 0 \quad ((i, j) \in A, k \in K'), \quad (14)$$

$$w_{ik} \geq 0 \quad (i \in V, k \in K'), \quad (15)$$

$$q_{ij} \text{ unrestricted } ((i, j) \in A : i < j). \quad (16)$$

Denote by  $Q(P)$  the optimal objective function value of a linear program  $P$ . Let  $(x, y, z)$  and  $(\bar{x}, \bar{y}, \bar{z})$  be the optimal solution vectors of MP and RMP, respectively, and let  $(\bar{u}, \bar{v}, \bar{w})$  be the associated optimal solution to the DRMP. Because the RMP is a restriction of MP, a feasible MP solution can be obtained by setting  $z_k^{pr} = \bar{z}_k^{pr}$  for  $p \in \bar{P}(k)$  and  $r \in \bar{R}(p)$ , and  $z_k^{pr} = 0$  for  $p \in P(k) \setminus \bar{P}(k)$  and  $r \in R(p) \setminus \bar{R}(p)$ . This solution is optimal for MP if and only if  $(\bar{u}, \bar{v}, \bar{w})$  is feasible for DMP, i.e.,

$$\bar{u}_k - \sum_{(i,j) \in A} a_{ij}^p \bar{v}_{ijk} - \sum_{i \in V} b_i^r \bar{w}_{ik} \leq 0 \quad (k \in K', p \in \bar{P}(k), r \in \bar{R}(p)). \quad (17)$$

```

1. Set  $\bar{E} := \emptyset, \bar{V} := \emptyset$  and  $Q = 0$ .
2. for each  $k \in K$  in a random sequence do {
    call SPPR( $k$ ) to find a path  $p(k)$  and a relay pattern  $r(k)$  over updated edges and
    relay fixed costs
    for each  $(i, j) \in p(k)$  do
        {  $\bar{E} := \bar{E} \cup (i, j); Q := Q + c_{ij}; c_{ij} = 0;$  }
    for each  $i \in r(k)$  do
        {  $\bar{V} := \bar{V} \cup i; Q := Q + f_i; f_i = 0;$  }
}
```

Fig. 2. Pseudo-code of the construction heuristic CH1.

Therefore, if for every commodity  $k \in K'$  and associated  $p \in P(k)$  and  $r \in R^p$  the following inequality holds:

$$u_k - \sum_{(i,j) \in A} a_{ij}^p v_{ijk} - \sum_{i \in V} b_i^r w_{ik} \leq 0, \quad (18)$$

then  $(\bar{x}, \bar{y}, \bar{z})$  is optimal for MP. To verify this condition, it suffices to solve the following subproblem for each  $k \in K'$ :

$$(SPPR(k)) \quad \min_{p \in P(k), r \in R(p)} \left\{ \sum_{(i,j) \in A} a_{ij}^p \bar{v}_{ijk} + \sum_{i \in V} b_i^r \bar{w}_{ik} \right\} \quad (19)$$

```

1. Set  $\bar{E} := \emptyset, \bar{V} := \emptyset$  and  $Q = 0$ .
2. while  $K \neq \emptyset$  {
    find  $k \in K$  with minimum SPPR( $k$ ) solution, with path  $p(k)$  and a relay pattern  $r(k)$ 
    for each  $(i, j) \in p(k)$  do
         $\{\bar{E} := \bar{E} \cup (i, j); Q := Q + c_{ij}; c_{ij} = 0;\}$ 
    for each  $i \in r(k)$  do
         $\{\bar{V} := \bar{V} \cup i; Q := Q + f_i; f_i = 0;\}$ 
     $K := K \setminus k;$ 
}
    
```

Fig. 3. Pseudo-code of IOH.

```

1. Set  $\bar{E} := \emptyset, \bar{V} := \emptyset$  and  $Q = 0$ .
2. while  $K \neq \emptyset$  {
    find  $k \in K$  with maximum SPPR( $k$ ) solution, with path  $p(k)$  and a relay pattern  $r(k)$ 
    for each  $(i, j) \in p(k)$  do
         $\{\bar{E} := \bar{E} \cup (i, j); Q := Q + c_{ij}; c_{ij} = 0;\}$ 
    for each  $i \in r(k)$  do
         $\{\bar{V} := \bar{V} \cup i; Q := Q + f_i; f_i = 0;\}$ 
     $K := K \setminus k;$ 
}
    
```

Fig. 4. Pseudo-code of DOH.

```

1. Set  $\tilde{E} := \emptyset, \tilde{V} := \emptyset$  and  $\tilde{Q} = 0$ .
2. calculate CH1;  $Q^*$  equals CH1 solution cost.
3. for each  $(i, j) \in E$  do {
     $aux := c_{ij}; c_{ij} := 0;$ 
     $z$  equals best solution cost of 10 calls to CH1;
    if  $z(CH1) + \tilde{Q} + aux < Q^*$ , then
         $\{ Q^* := z(CH1) + \tilde{Q} + aux; \tilde{E} := \tilde{E} \cup (i, j); \tilde{Q} := \tilde{Q} + aux; \}$ 
    else
         $c_{ij} := aux;$ 
}
4. for each  $i \in V$  do {
     $aux := f_i; f_i := 0;$ 
     $z$  equals best solution cost of 10 calls to CH1;
    if  $z(CH1) + \tilde{Q} + aux < Q^*$ , then
         $\{ Q^* := z(CH1) + \tilde{Q} + aux; \tilde{V} := \tilde{V} \cup i; \tilde{Q} := \tilde{Q} + aux; \}$ 
    else
         $f_i := aux;$ 
}
5. if  $\tilde{E}$  or  $\tilde{V}$  where updated, goto step 3.
    
```

Fig. 5. Pseudo-code of the heuristic CH2.

which follows from

$$\bar{u}_k - \sum_{(i,j) \in A} a_{ij}^p \bar{v}_{ijk} - \sum_{i \in V} b_i^r \bar{w}_{ik} \leq 0,$$

$$\max_{p \in P(k), r \in R(p)} \left\{ \bar{u}_k - \sum_{(i,j) \in A} a_{ij}^p \bar{v}_{ijk} + \sum_{i \in V} b_i^r \bar{w}_{ik} \right\} \leq 0,$$

$$\bar{u}_k - \min_{p \in P(k), r \in R(p)} \left\{ \sum_{(i,j) \in A} a_{ij}^p \bar{v}_{ijk} + \sum_{i \in V} b_i^r \bar{w}_{ik} \right\} \leq 0.$$

The problem defined by (19) is a *shortest path problem with relays* (Cabral et al., 2005) for which fast exact pseudo-polynomial algorithms exist. As

is commonly done in column generation algorithms (Lübbecke and Desrosiers, 2005), each column defined by (19) is added to RMP, the dual values are computed, and SPPR(k) is solved for new commodities k. This procedure is repeated as long as promising new columns can be identified. Alternatively, a computing time limit δ can be set as an alternative stopping criterion since generating all columns may be too time consuming. A lower bound Q(RMP) = ∑\_{k ∈ K'} Q(SPPR(k)) on Q(RMP) is then used if the limit is reached. In this case, we can state that the optimality gap attained is

Table 1  
Computational time average in seconds based on 10 test problems per parameter set

K	a	b	V	E	RMP	RMPI	CH1	IOH	DOH	CH2
5	4	5	22	31	0.84	0.62	0.03	0.00	0.00	1.64
	5	5	27	40	2.28	2.01	0.04	0.00	0.00	2.46
	6	5	32	49	14.16	7.98	0.04	0.00	0.00	3.47
	7	5	37	58	30.66	7.49	0.05	0.01	0.00	4.53
	8	5	42	67	256.71	33.65	0.05	0.00	0.01	5.91
	9	5	47	76	182.47	20.85	0.06	0.00	0.01	7.37
	10	5	52	85	1079.25	46.12	0.07	0.00	0.01	9.12
	11	5	57	94	3643.76	33.78	0.07	0.00	0.01	11.12
10	12	5	62	103	7193.12	549.22	0.08	0.01	0.01	13.26
	4	5	22	31	5.79	3.74	0.05	0.00	0.00	2.54
	5	5	27	40	19.99	30.45	0.06	0.00	0.01	3.97
	6	5	32	49	189.81	87.80	0.07	0.01	0.01	5.89
	7	5	37	58	378.51	183.60	0.08	0.01	0.01	7.92
	8	5	42	67	1048.27	523.47	0.09	0.01	0.01	10.34
	9	5	47	76	1926.54	339.31	0.10	0.01	0.01	13.00
	10	5	52	85	12232.46	1419.28	0.12	0.01	0.01	16.72
	11	5	57	94	16994.08	3496.11	0.14	0.01	0.01	20.39
	12	5	62	103	30922.21	4656.37	0.15	0.01	0.02	24.68

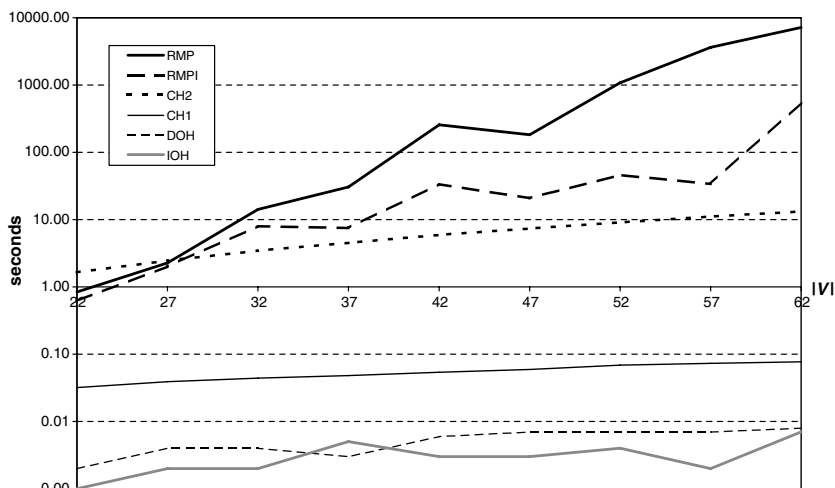


Fig. 6. Computational time for all six algorithms (|K| = 5).

$[Q(\text{RMP}) - \underline{Q}(\text{RMP})]/Q(\text{RMP})$ . A feasible solution to the NDPR can be then derived by replacing constraints (6)–(8) with:

$$x_{ij} = 0 \text{ or } 1 \quad ((i, j) \in A), \tag{20}$$

$$y_i = 0 \text{ or } 1 \quad (i \in V), \tag{21}$$

$$z_k^{pr} = 0 \text{ or } 1 \quad (k \in K', p \in P(k), r \in R(p)), \tag{22}$$

and calling the MIP solver in CPLEX. This is a heuristics solution as there is no guarantee that all optimal paths and relay patterns were included when

solving the RMP. We call this solution *reduced master problem in integers* (RMPI).

### 3. Heuristics

We developed four construction heuristics for the NDPR. The first heuristic, called *construction heuristic 1* (CH1), consists of constructing the network by considering one commodity at a time. The second heuristic, called *increasing order heuristics* (IOH), consists of constructing the network one

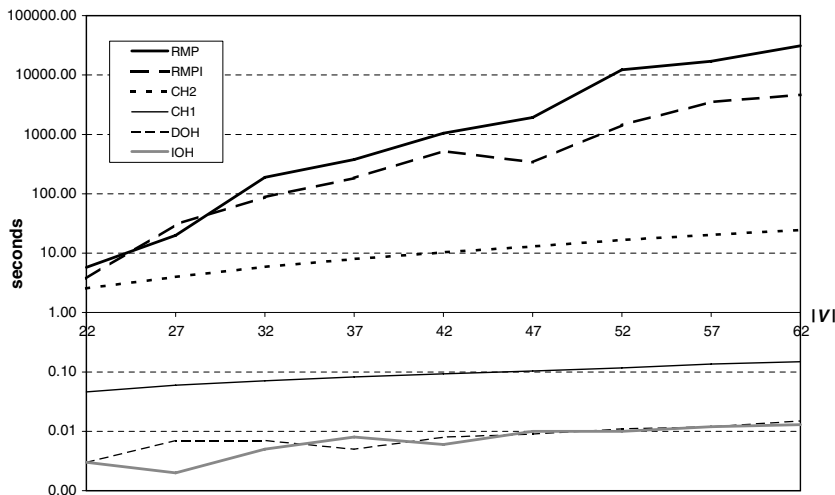


Fig. 7. Computational time for all six algorithms ( $|K| = 10$ ).

Table 2  
Optimality gap based on 10 test problems per parameter set

$ K $	$a$	$b$	$ V $	$ E $	RMPI (%)	CH1 (%)	IOH (%)	DOH (%)	CH2 (%)	
5	4	5	22	31	103.45	103.49	111.84	109.21	103.10	
	5	5	27	40	105.02	109.74	123.36	116.17	104.74	
	6	5	32	49	105.58	106.49	116.60	108.44	105.59	
	7	5	37	58	103.53	108.55	125.69	116.87	103.67	
	8	5	42	67	107.57	110.72	125.95	113.50	107.00	
	9	5	47	76	106.49	110.55	124.10	113.22	105.58	
	10	5	52	85	107.41	108.40	123.44	110.75	106.50	
	11	5	57	94	105.29	105.94	113.45	109.78	105.13	
	12	5	62	103	106.61	108.31	119.84	111.15	105.16	
	10	4	5	22	31	107.83	110.30	124.88	119.99	107.21
		5	5	27	40	110.40	116.25	128.89	123.53	109.80
		6	5	32	49	116.41	117.04	138.15	123.35	114.55
7		5	37	58	112.99	118.94	137.70	125.92	112.06	
8		5	42	67	114.01	116.92	136.59	125.19	111.74	
9		5	47	76	113.24	116.24	136.97	121.99	111.96	
10		5	52	85	114.58	118.09	132.10	120.54	112.75	
11		5	57	94	119.16	117.99	131.62	124.59	111.32	
12		5	62	103	121.79	122.08	137.34	129.88	118.33	

Gap from RMP lower bound.

commodity at a time, ranking them in increasing cost of implementation. The third heuristic, called *decreasing order heuristic* (DOH), consists of constructing the network one commodity at a time, ranking them by their decreasing cost of implementation. Finally, the fourth and last heuristic, called *construction heuristic 2* (CH2), uses CH1 as subroutine and implements the network by exploiting arcs and vertices impact on CH1 execution.

### 3.1. Construction heuristic 1 (CH1)

Heuristic CH1 is inspired by the algorithm of Takahashi and Matsuyama (1980) for the *Steiner tree problem* (STP). In the STP the aim is to con-

struct a minimum cost tree spanning a set of *mandatory vertices* in a graph. The Takahashi and Matsuyama (1980) algorithm gradually builds a tree by connecting an additional communication pair  $(o(k), d(k))$  at each iteration. This is done by connecting  $o(k)$  and  $d(k)$  to the current network by means of shortest paths. Whenever a path is added, its edge costs are set equal to zero in order to incite future shortest paths to make use of the edges currently used in the network. We have used the same idea except that in the NDPR each  $(o(k), d(k))$  path must be feasible with respect to relay locations and its cost is the sum of edge costs and relay fixed costs. The pseudo-code description of the construction algorithm CH1 is provided in Fig. 2. The order in

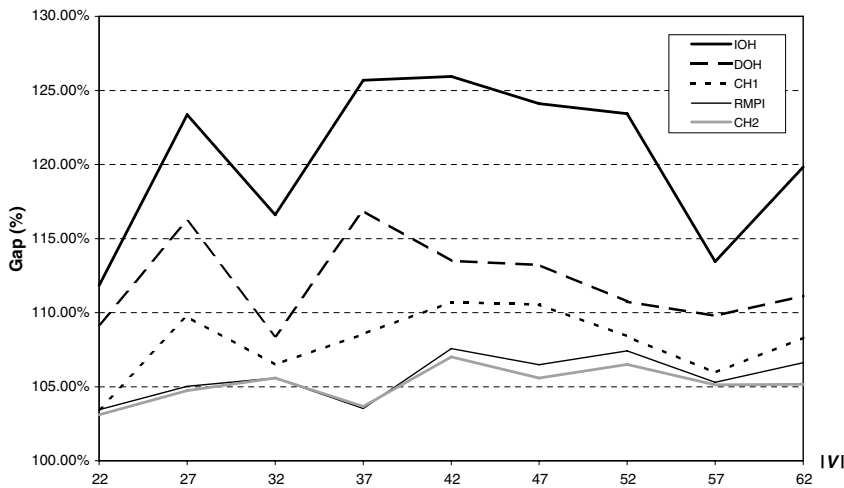


Fig. 8. Optimality gap for all five heuristics ( $|K| = 5$ ).

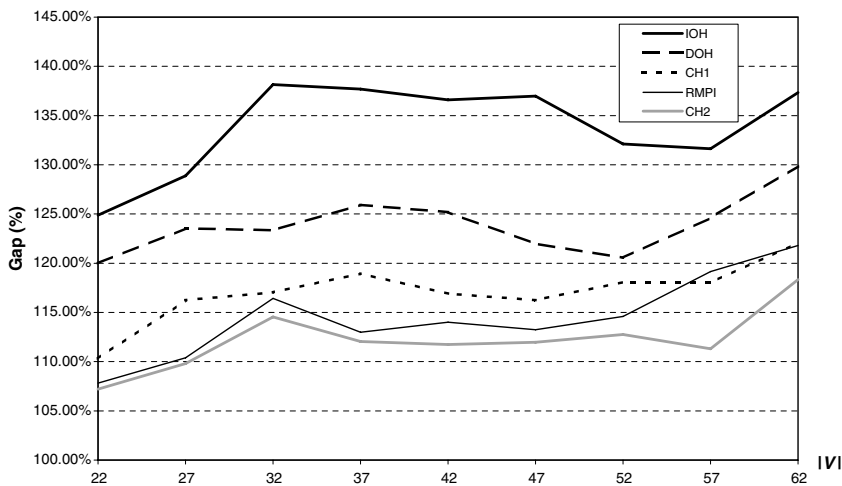


Fig. 9. Optimality gap for all five heuristics ( $|K| = 10$ ).

which each commodity is inserted impacts the final design. Because each call to CH1 orders  $K$  in a randomly generated sequence, multiple calls to CH1 may return up to  $|K|!$  different solutions.

### 3.2. Increasing order heuristic (IOH)

The idea explored in the IOH is that communication flows with the smallest implementation costs should be built first, independent of the other communication flows. Starting with an empty NDP solution (i.e.,  $\bar{V} := \emptyset$  and  $\bar{E} := \emptyset$ ), the algorithm identifies the communication pair  $(o(k), d(k))$  with minimum SPPR solution cost, and incorporates the edges and vertices with relays from this SPPR solution into  $\bar{V}$  and  $\bar{E}$ , while setting their costs  $c_{ij}$  and  $f_i$  to zero. The communication pair is then eliminated from  $K$ , and the above construction algorithm is repeated until  $K$  is empty. The pseudo-code for algorithm IOH is provided in Fig. 3.

### 3.3. Decreasing order heuristic (DOH)

The idea explored in the DOH is the opposite of the principle behind the IOH: communication flows with the highest implementation costs should be built first. We present its pseudo-code in Fig. 4.

### 3.4. Construction heuristic 2 (CH2)

The main weakness of CH1 lies on its blindness towards edges and vertices that are not in any path and relay pattern provided by SPPR solutions. To compensate for this weakness, CH2 holds a pool of edges  $\tilde{E}$  and vertices  $\tilde{V}$  that are considered as included in the final design, which amounts to stating that  $c_{ij} = 0$  for each  $(i, j) \in \tilde{E}$  and  $f_i = 0$  for each  $i \in \tilde{V}$ . We use the variable  $\tilde{Q}$  to represent the fixed cost of implementing pools  $\tilde{E}$  and  $\tilde{V}$ . The algorithm starts with  $\tilde{E} := \emptyset$  and  $\tilde{V} := \emptyset$ . In step 2, CH1 is calculated ( $E$  and  $V$ ), and the best solution value is stored in  $Q^*$ . Then, in an edge scanning loop (step 3), the cost  $c_{ij}$  of each edge  $(i, j) \in E$  is temporarily set to 0, and  $z$  is equal to the best solution cost from 10 calls to CH1. If  $z + \tilde{Q} + c_{ij}$  is smaller than  $Q^*$ , this solution is stored as the best solution encountered so far, i.e.,  $Q^* = z + \tilde{Q} + c_{ij}$ , and edge  $(i, j)$  is added to  $\tilde{E}$ . The same scanning loop is executed for vertices (step 4), where the cost  $f_i$  of each vertex  $i \in V$  is temporarily set to 0, and the best of 10 calls to CH1 is stored in  $z$ . If  $z + \tilde{Q} + f_i < Q^*$ , the

solution is stored as the best solution so far,  $Q^*$  is updated, and vertex  $i$  is added to  $\tilde{V}$ . If either  $\tilde{E}$  or  $\tilde{V}$  were updated in steps 2 and 3, the algorithm proceeds to step 2, otherwise it stops. The pseudo-code for algorithm CH2 is provided in Fig. 5.

## 4. Computational results

We carried out all computational tests out on a Sun Fire 480R station with four 900 MHz processors, 16 gigabytes of RAM and a Sun Solaris 5.7

Table 3  
Computational time average in seconds based on 10 test problems per parameter set

$ K $	$a$	$b$	$ V $	$ E $	CH1	IOH	DOH	CH2
5	10	5	52	85	0.1	0.0	0.0	9.7
		10	102	180	0.2	0.0	0.0	45.2
		15	152	275	0.3	0.0	0.0	110.7
		20	202	370	0.4	0.0	0.0	220.6
	20	5	102	175	0.2	0.0	0.0	42.2
		10	202	370	0.4	0.0	0.0	229.7
		15	302	565	0.6	0.1	0.1	553.1
		20	402	760	0.9	0.1	0.1	992.5
	30	5	152	265	0.3	0.0	0.0	108.4
		10	302	560	0.6	0.1	0.1	553.1
		15	452	855	1.1	0.1	0.1	1397.4
		20	602	1150	1.5	0.1	0.2	2619.1
40	5	202	355	0.4	0.0	0.0	228.8	
	10	402	750	1.0	0.1	0.1	1119.4	
	15	602	1145	1.5	0.1	0.1	2541.4	
	20	802	1540	2.2	0.2	0.2	5294.2	
50	5	252	445	0.5	0.0	0.0	320.4	
	10	502	940	1.2	0.1	0.1	1732.5	
	15	752	1435	2.2	0.2	0.2	4863.2	
	20	1002	1930	3.1	0.3	0.3	9008.6	
10	10	5	52	85	0.1	0.0	0.0	19.0
		10	102	180	0.3	0.0	0.0	89.6
		15	152	275	0.5	0.1	0.1	231.0
		20	202	370	0.8	0.1	0.1	467.6
	20	5	102	175	0.3	0.0	0.0	86.8
		10	202	370	0.8	0.1	0.1	480.2
		15	302	565	1.4	0.1	0.1	1173.7
		20	402	760	2.0	0.2	0.2	2375.3
	30	5	152	265	0.5	0.1	0.1	227.4
		10	302	560	1.3	0.1	0.1	1185.8
		15	452	855	2.3	0.2	0.2	2949.6
		20	602	1150	3.4	0.3	0.3	5999.0
40	5	202	355	0.8	0.1	0.1	463.1	
	10	402	750	2.1	0.2	0.2	2412.7	
	15	602	1145	3.4	0.3	0.3	5979.5	
	20	802	1540	5.0	0.4	0.5	11832.7	
50	5	252	445	1.0	0.1	0.1	692.6	
	10	502	940	2.8	0.3	0.3	4030.9	
	15	752	1435	5.1	0.5	0.5	11101.1	
	20	1002	1930	7.0	0.8	0.7	20811.3	



operational system. We coded the algorithms in C++ and compiled them with a *Sun Forte Developer 7* C++ compiler. We used CPLEX 8.0 to solve the linear programs in the column generation procedure.

The test graphs follow a grid structure, with  $a$  rows and  $b$  columns and randomly (uniformly) generated integer values for costs and lengths. Cost and edge length values are selected from  $[10, 30]$ . Communication pairs have a common origin node, and were randomly chosen. The relay fixed costs are randomly generated as integers in the interval  $[\lambda, 2\lambda]$ .

Table 1 provides a representative sample of our computational experiments for all algorithms presented, for  $a$  values between 4 and 12, and  $|K|$  equals to 5 and 10. Parameters  $\lambda$  and  $b$  were fixed to 70 and 5, respectively. Each row contains average computational time in seconds for 10 instances. The RMP and RMPI algorithms had a time limit  $\delta$  of 10 hours, which affected one instance for  $a = 12$  and  $|K| = 5$ , one instance for  $a = 10$  and  $|K| = 10$ , three instances for  $a = 11$  and  $|K| = 10$ , and seven instances for  $a = 12$  and  $|K| = 10$ . One can observe the increased computational effort required for both RMP and

Table 4  
Gap from best heuristic solution based on 10 test problems per parameter set

$ K $	$a$	$b$	$ V $	$ E $	CHI (%)	IOH (%)	DOH (%)	CH2 (%)
5	10	5	52	85	101.80	115.88	103.97	100.00
		10	102	180	105.00	112.67	113.36	100.00
		15	152	275	103.56	112.91	113.69	100.00
		20	202	370	102.57	111.00	115.18	100.00
	20	5	102	175	102.12	110.28	104.92	100.16
		10	202	370	101.27	110.90	108.89	100.00
		15	302	565	103.14	114.48	108.24	100.02
		20	402	760	102.24	107.91	112.38	100.00
	30	5	152	265	102.20	109.31	107.91	100.10
		10	302	560	104.41	106.27	109.86	100.00
		15	452	855	102.98	106.25	109.92	100.00
		20	602	1150	103.25	108.10	107.76	100.00
	40	5	202	355	100.50	105.67	101.16	100.00
		10	402	750	102.02	104.79	107.17	100.00
		15	602	1145	103.24	106.79	108.93	100.00
		20	802	1540	103.86	109.05	110.38	100.00
	50	5	252	445	102.07	107.28	104.26	100.02
		10	502	940	102.53	106.20	105.32	100.00
		15	752	1435	103.10	106.18	106.47	100.01
		20	1002	1930	102.65	106.75	110.13	100.00
10	10	5	52	85	104.79	117.09	106.88	100.04
		10	102	180	106.80	118.74	114.49	100.00
		15	152	275	105.63	114.78	113.83	100.00
		20	202	370	104.75	114.61	115.63	100.00
	20	5	102	175	103.97	113.68	109.79	100.00
		10	202	370	104.24	116.75	113.58	100.00
		15	302	565	103.29	113.70	110.07	100.00
		20	402	760	103.66	109.31	109.09	100.00
	30	5	152	265	102.15	112.86	109.68	100.12
		10	302	560	104.10	110.40	113.75	100.00
		15	452	855	104.44	112.70	110.86	100.00
		20	602	1150	104.01	107.29	113.86	100.00
	40	5	202	355	102.03	110.10	104.98	100.00
		10	402	750	103.29	107.86	111.20	100.00
		15	602	1145	104.53	110.37	110.16	100.00
		20	802	1540	104.11	109.32	113.35	100.21
	50	5	252	445	102.34	111.40	105.12	100.05
		10	502	940	103.79	108.49	109.54	100.00
		15	752	1435	102.79	107.29	107.45	100.00
		20	1002	1930	104.12	109.06	108.93	100.00

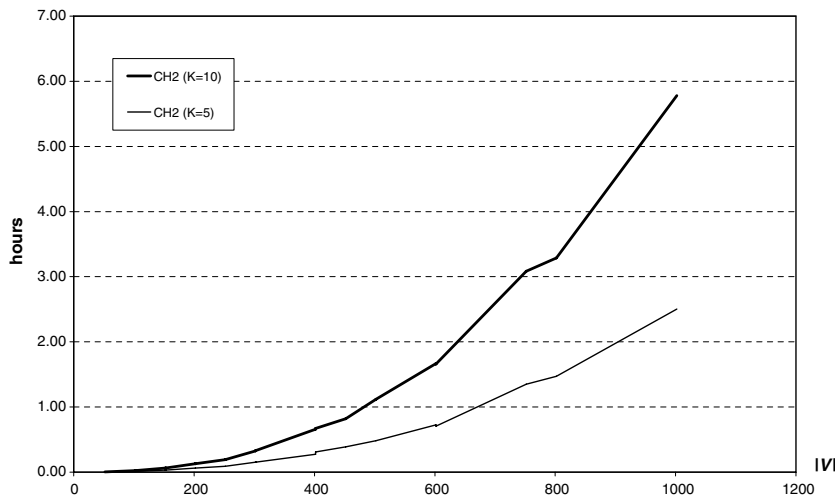


Fig. 10. Computational time for CH2 for larger test networks.

RMPI in Figs. 6 and 7 (logarithmic time axis), limiting its usefulness for applications of the NDPR.

Table 2 provides the optimality gap (in percent) for each one of the heuristics developed for the NDPR, computed as heuristic solution cost divided by lower bound generated by RMP. As one can observe in Figs. 8 and 9, the most promising heuristics are CH2 and RMPI, with an obvious preference for the first one given its lower computational time.

Tables 3 and 4 present an extended computational analysis for algorithm CH1, IOH, DOH, and CH2. Table 3 reports the average computational time required to solve larger instances, and Table 4 reports the gap between the heuristics values and the best solution found cost. On the one hand, one can observe that in general CH2 provides the best solutions (see Fig. 10). On the other hand, as computational effort required by the other heuristics is insignificant compared to CH2, one should run all heuristics. If one cannot afford the computational time required by CH2, CH1 seems to be a good alternative, as it produces solutions that are within 5% gap from CH2. We note that the computational times reported in Table 3 are not directly comparable to those in Table 1 since we were forced to use a different compiler to solve the larger problems.

## 5. Conclusions

We introduced the *network design problem with relays* and proposed a lower bound as well as four heuristics for its solution. The lower bound is obtained through column generation using the SPPR as a subproblem. The first heuristic is a mod-

ification of the construction heuristic by Takahashi and Matsuyama (1980) for the *Steiner tree problem*. The second and third heuristics check whether biasing the first construction heuristic could improve its solution. Finally, the fourth heuristic explores the inclusion of edges and vertices into the final design to improve the results provided by the first heuristic. Overall the fourth heuristic is the best; we applied it to instances with up to 1002 vertices and 1930 edges within reasonable computational times.

## Acknowledgments

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council under Grants CRD 268431, OGP 25481 and OGP 39682. This support is gratefully acknowledged. Thanks are due to Fatma Gzara, Osman Alp and the referees for their valuable comments.

## References

- Balakrishnan, A., Magnanti, T.L., Mirchandani, P., 1997. Network design. In: Dell'Amico, M., Maffioli, F., Martello, S. (Eds.), *Annotated Bibliographies in Combinatorial Optimization*. Wiley, Chichester, pp. 311–334.
- Cabral, E.A., Erkut, E., Laporte, G., Tjandra, S.A., 2005. The shortest path problem with relays, Working paper.
- Cortes, P., Onieva, L., Larraneta, J., Garcia, J.M., 2001. Decision support system for planning telecommunication networks: A case study applied to the Andalusian region. *Journal of the Operational Research Society* 52, 283–290.
- Cosares, S., Deutsch, D.N., Saniee, I., Wasem, O.J., 1995. Sonet toolkit: A decision support system for designing robust and cost-effective fiber-optic networks. *Interfaces* 25 (1), 20–40.

- Costa, A.M., 2005. A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research* 32, 1429–1450.
- Davis, R.D., Kumaran, K., Liu, G., Sanjee, I., 2001. Spider: A simple and flexible tool for design and provisioning of protected lightpaths in optical networks. *Bell Labs Technical Journal* 6, 82–97.
- De Giovanni, L., Della Croce, F., Tadei, R., 2004. On the impact of the solution representation for the internet protocol network design problem with max-hop constraints. *Networks* 44, 73–83.
- Dumitrescu, I., Boland, N., 2003. Improved preprocessing labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks* 42, 135–153.
- Gouveia, L., 1998. Using hop-indexed models for constrained spanning and Steiner tree models. In: Sanso, B., Soriano, P. (Eds.), *Telecommunications Network Planning*. Kluwer Academic Publishers, Boston, pp. 21–31.
- Gouveia, L., Requejo, C., 2001. A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem. *European Journal of Operational Research* 132, 539–552.
- Lübbecke, M.E., Desrosiers, J., 2005. Selected topics in column generation. *Operations Research* 53, 1007–1023.
- Raghavan, S., Magnanti, T.L., 1997. Network connectivity. In: Dell'Amico, M., Maffioli, F., Martello, S. (Eds.), *Annotated Bibliographies in Combinatorial Optimization*. Wiley, Chichester, pp. 335–354.
- Soni, S., 2001. Hop constrained network design problem with partial survivability. *Annals of Operations Research* 106, 181–198.
- Takahashi, H., Matsuyama, A., 1980. An approximate solution for the Steiner problem in graphs. *Mathematica Japonica* 24, 573–577.
- Tcha, D.-W., Yoon, M.-G., 1995. Conduit and cable installation for a centralized network with logical star–star topology. *IEEE Transactions on Communications* 43, 958–967.
- Voss, S., 1999. The Steiner tree problem with hop constraints. *Annals of Operations Research* 86, 321–345.
- Yoon, M.G., Baek, Y.H., Tcha, D.W., 1998. Design of a distributed fiber transport network with hubbing topology. *European Journal of Operational Research* 104, 510–520.