Design of Cellular Networks with Diversity and Capacity Constraints

Peter Kubat, *Member, IEEE*, J. MacGregor Smith, and Calvin Yum

***Abstract*—This paper presents three mathematical models for network design of cellular networks. The models reflect varying degrees of complexity.**

Model #1 is a 1-period fixed-link capacity model. Three heuristics are used for solving this problem. All the heuristics first use linear programming relaxations to yield the near-optimal integer solution, then use then clever rounding-schemes to find the final solution. The three heuristics are compared with an integer branch-and-bound algorithm to show the efficacy of the heuristics and the speed with which they achieve their solution. The first heuristic is the best. An Appendix presents a detailed algorithmic description of the first heuristic.

Model #2 allows the capacities of the links to vary. This is a much more difficult mathematical programming problem, yet certain features of the problem reveal valuable characteristics of the linear programming relaxations. Two heuristics are generated; the first heuristic is superior to the second. The heuristics are compared with an optimal branch-and-bound algorithm.

Model #3 presents a multi-period demand problem. This is a very complex problem and, while no heuristics are developed and no computational experiments are shown, the structure of the final problem is similar to models #1 & #2; thus linear programming relaxations should be a viable strategy for its solution.

***Index Terms*—Branch and bound, cellular, heuristics, telecommunications network design.**

I. Introduction

Acronyms

CDMA	code division multiple access
DSL	digital signal level
DS0	DSL 0–64 Kb/s
DS1	DSL 1–1.544 Mb/s, DS1 = 24DS0
DS3	DSL 3–44.736 Mb/s, DS3 = 28DS1
GSM	global system for mobile communication
H1-1	Heuristic 1-1, *etc*
IP	integer programming
IP*	optimal solution of IP
LIUB	least integer upper bound
LP	linear programming
MSC	mobile switching center
PSTN	public switched telephone network
SONET	synchronous optical network
$T1$	DS1
TDMA	time division multiple access
WAN	wide area network

A typical cellular system has 3 main parts:

- mobile units,
- cell sites,
- a MSC.

Calls originating and/or terminating at the mobile units, are connected to a cell site over a radio link (analog, or digital for newer systems). From the cell, the call proceeds to the MSC for processing and switching. The switch transfers the call to its destination—either another mobile unit, or to a PSTN for delivery to a wired telephone terminal. Between the cell site and the MSC, the connection is over terrestrial, digital transmission lines (usually $T1$ lines), or over private digital microwave. The $T1$ lines are typically leased from a telephone company, the microwave transmission systems are usually purchased and owned by the cellular operator.

The cell sites, containing radio transceivers, antennae, and interconnection transmission equipment, are strategically located to cover the geographic area served by the cellular service provider. The cell locations depend upon several factors, e.g.:

- technology (analog, TDMA, GSM, CDMA),
- cellular radio system frequency band (900 MHz, 1800 MHz),
- geography of the site (these factors influence radio propagation characteristics and thus radio coverage),
- the regional demographic,
- census data,
- highway systems and commuting habits (these are driving the system traffic engineering requirements),
- location of neighboring cells,
- location of cells of competing providers,
- cost of real estate.

The radio capacity of each cell (in terms of number of calls that it can handle simultaneously) is determined by

- subscriber "demographics and calling patterns," which, in conjunction with the radio technology and antenna system, eventually translate into the number of radio modules engineered to guarantee a specific grade of service (for example, 2% blocking during cell's busy hour),
- the number of DS0 channels assigned to carry the voice payload and radio control to the MSC.

Since every cell must be connected to the MSC, the logical connection must be a star, and historically, the early interconnect (physical) networks for cellular systems had a star topology as well. However, as more cell sites are added to the network and

Manuscript received September 8, 1998; revised September 8, 1999.

P. Kubat is with GTE Laboratories Inc, 40 Sylvan Rd, Waltham, MA 02451-1128 USA (e-mail: PKubat@gte.com).

J. M. Smith and C. Yum are with the Department of Mechanical & Industrial Engineering, University of Massachusetts, Amherst, MA 01003 USA (e-mail: JMSmith@ecs.umass.edu).

Responsible editor: J. D. Healy.

Publisher Item Identifier S 0018-9529(00)10125-3.

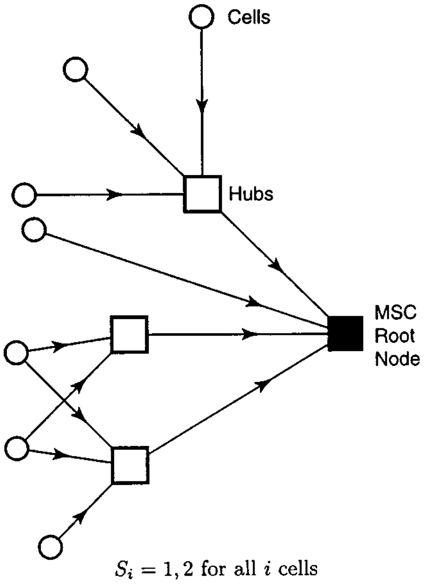

Fig. 1. Example topologies with diversity constraints.

the cost of the facilities increases, this star architecture is no longer cost effective because it does not provide for any sharing of transmission facilities. To reduce the transmission costs for large networks, hubbing centers are created at some suitable location (typically a cell site) to provide traffic concentration, and subsequently sharing bigger transmission pipes (e.g., DS3, large microwave), to use the economies of scale.

Hubbing is clearly more cost efficient; however, there has been a concern about the reliability of the interconnect network, since when the network is a tree. A failure of a high capacity trunk could wreak havoc on the entire cellular network. To avoid outages, cellular companies have adopted a variety of alternatives, including the use of—

- extensive backup facilities,
- physical diversity of paths,
- leasing transmission capacity on the self-healing SONET rings [2].

The problem in this paper addresses the partial traffic protection in the interconnect network for a cellular system.

Assumptions

1) The location of current (existing) and future cells and a backbone network of large capacity are known.
2) The topology of the backbone is known. It is a tree topology (or a forest); the tree(s) root is at the MSC.
3) Cells are connected either to the backbone or directly to the MSC.
4) To guarantee a partial survivability, some cells could be required to split equally their traffic over at least 2 completely-disjoint paths.
5) D_i/s_i is an integer.
6) Hubs do not generate demand.
7) Hubs are Steiner points.

The problem is to find

1) The optimal homing of the cells to the backbone hubs.
2) The optimal backbone link capacity to meet the required survivability criterion and minimize the total network cost.

Section II briefly reviews the literature on cellular network design for general network topologies.

Section III defines the fundamental network design problem addressed in this paper for tree topologies and its decomposition into 3 separate sub-problems.

Section IV addresses the first problem:

P1) Single-period demand, fixed link capacity.

It develops a linear programming relaxation approach for solving P1 and presents computational experiments to demonstrate the effectiveness of our heuristic approach.

Section V addresses the second problem:

P2) Single-period demand, variable link capacity.

It develops a heuristic approach (related to the approach developed in Section IV) for solving P2.

Section VI extends P1 & P2 to P3, and addresses some open questions.

II. Literature Review

The proliferation of corporate WAN handling voice, data, e-mail, Internet, and other telecommunication applications, extensive choice of service providers, selection from many complex and exotic transmission technologies and systems, with an ever increasing pressure to contain costs' provides a real challenge to network designers. This, in turn attracts attention to modeling and cost optimization of networks in the private setting. Some models and cost optimizations for private networks have been already considered in the literature, e.g., [1], [6], [10]. More details on the design of telecommunication networks using SONET rings and the corresponding reliability considerations are in [9], [11], [12].

Interconnection-access networks for cellular systems are private networks, but they are unique in many ways. The cells must deliver the traffic to the MSC; this implies that the optimal topology must be a tree (which makes the problem easier). There is a complex choice of transmission technologies (e.g., leased $T1$ *vs.* private microwave), cost of concentrators, and multi-year planning horizon (which makes the design problem harder). The cellular network design is further complicated, by expanding the network to accommodate a phenomenal growth of cellular subscribers (growth about 100% a year is not unheard off), yet the cell demand is not always increasing (the reason for this is the cell splitting and cell moving, which in turn, can cause a decrease in traffic load for some cells).

In contrast to the traditional telephone networks, the cellular network topology can be very flexible and dynamic—both in the topology and link capacity, which can change periodically. This is because the leased transmission facility is simple to add, terminate, or move to another destination; it is also relatively easy to add and relocate a microwave facility. Refs. [5], [8] consider modeling and solving a capacity expansion of a star-star network over a given planning time horizon to meet projected traffic demand at minimum cost. The expansion plan specifies:

- where and when to place concentrators in the network,

- what type of concentrator to use at each of the hubs,
- which cell sites to connect to each of the hubs,
- which cells to connect directly to the MSC,
- what facilities to use on each of the links.

III. Problem Description

Notation

N	$V \cup H$
V	$\{v_1, v_2, \cdots, v_N\}$: cell nodes of cardinality N
H	$\{h_1, h_2, \cdots, h_M\}$: hub nodes of cardinality M (not including the root node)
Q	trees: $\{T_1, T_2, \cdots, T_Q\}$; for any two trees, $T_i \cap T_j = \emptyset$, Trees are rooted at node #0 in MSC
R	$H + 0$: all hub nodes plus the root node
Z	annual cost of interconnect network
$c_{i,j}$	fixed cost of connecting cell i to hub j ($/year)
$x_{i,j}$	decision variables
$a_{j,l}$	predefined data coefficients; $a_{j,l} = 1$ if link l is on the path from the root to hub j, $a_{j,l} = 0$ otherwise
D_i	fixed demand in DSO circuits for each cell i to be delivered to root node #0
s_i	diversity requirement at cell i; $s_i = 1, 2, 3$
K_l	capacity (in DS0 circuits) for link l
$G(N, A)$	directed graph with two set of nodes

This paper addresses the partial traffic protection in the interconnect network for a cellular system. The topology of the backbone is given and it is a tree topology (or a forest); the tree(s) root at node 0—the MSC. Cells, which generate D_j (in DS0), send this demand to one or more tree nodes (excluding the root) or to the MSC (root) directly using leased $T1$ (DS1) circuits. Each cell is assigned protection or routing diversity requirement $1, 2$. For diversity 2, the paths are required to be completely disjoint and the demand is split equally. Equivalently, diversity 2 means that 50% of the traffic will be routed to 1 backbone node and the other 50% will go to some other node (or MSC). This makes sense, since most of the cell sites are unmanned, and the DS0 circuits are directly hardwired to the outgoing transmission facilities. When a cell-node link fails, only 50% of the traffic is lost. If a cell connects to 2 hub nodes, the nodes must be on 2 different trees. Diversity 1 means no diversity at all.

The objective then is to find a cell-to-hub interconnection network and backbone link capacities so that the diversity requirement is met and the cost of interconnection is minimized.

Design of cellular networks with SONET rings and partial survivability was studied in [2]. This paper examines a problem that is different from the design problem [2] in both its modeling aspects and the proposed heuristic solution. Specifically, [2] addresses the problem of the optimal homing arrangement where there is a single reliable SONET ring of given capacity and the MSC is attached to the ring.

- Cells with diversity 1 can home either directly 100% of the traffic demand to the MSC (not using the ring capacity at all) or to a ring node.
- Cells with diversity 2 must home to either 2 distinct ring nodes (using the ring capacity for 100% of the traffic demand), or to home 50% of the traffic directly to the MSC and the other 50% of the demand to a ring node (loading the ring with only 50% of the cell traffic demand). Thus, there is only a single capacity constraint for the ring.

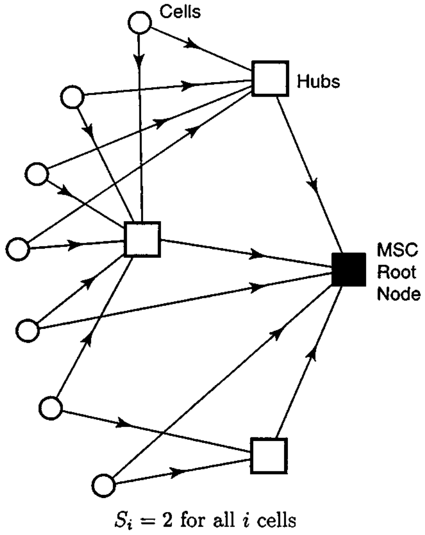

Fig. 2. Example topologies with diversity constraints.

This paper considers an optimal homing arrangement without the ring restriction and under a more general setting. Cells can home to any node, including the root (MSO), but for cells with diversity 2 (or higher), the cells must home not only to the two distinct nodes but these nodes must be on different trees. This restriction adds a complicating set of constraints. Different amounts of capacity are allowed on each tree branch. This makes the problem quite realistic, because the backbone can carry some other traffic as well, and only a certain portion of the total link capacity can be allocated to the cellular application. Thus, the formulation must consider that the diversified traffic traverses different trees, and all the capacity constraints are met. There are very many constraints; thus the problem is much more complicated.

Problem Decomposition

Three basic mathematical models/problems are considered in their order of complexity:

P1) Single-Period Demand, Fixed Link Capacity.
P2) Single-Period Demand, Variable Link Capacity.
P3) Multi-Period Demand, Variable Link Capacity.

IV. P1: Single-Period Demand, Fixed Link Capacity

The capacity in each link is fixed.

By convention, #0 is the site index for the MSC. In practice, the most likely values for diversity s_i are 1 and 2. The $c_{i,j}$ is derived using relevant tariffs and could depend on D_i.

The $x_{i,j}$ indicate the homing assignments and $R = H + 0$. The data coefficients, $a_{j,l}$, are predefined; see *Notation*. Also, *nota bene*, since we deal only with trees, then there are at most s_i paths emanating from each cell node to the root. The problem is stated formally as:

$$P1\text{: Minimize } Z = \sum_{i \in V} \sum_{j \in R} c_{i,j} \cdot x_{i,j} \tag{1}$$

subject to

$$\sum_{j\in R} x_{i,j} = s_i \quad \text{for all } i \in V \tag{2}$$

$$\sum_{i\in V}\sum_{j\in R} \frac{D_i}{s_i}\cdot a_{j,l}\cdot x_{i,j} \le K_l \quad \text{for all } l \in H \tag{3}$$

$$\sum_{j\in T_q} x_{i,j} \le 1 \quad \text{for all } i,\, q = 1, \cdots, Q \tag{4}$$

$$x_{i,j} = \{0, 1\} \quad \text{for all } i,\, j. \tag{5}$$

We want to pick homing assignments so that Z is minimized. Since the link capacity is fixed, the cost of the link capacity is irrelevant for the homing assignments. Under diversity, D_i is divided into s_i equal parts, rounding up, if necessary.

- Constraint set (2) ensures that the diversity requirement of each cell is met.
- Constraint set (3) ensures that the link capacity is not exceeded.
- Constraint set (4) are tree diversity constraints—ensuring that if $s_i \ge 2$ then only one branch is homed to a given tree (thus, if the link connecting to the root fails, not all the traffic is lost).

A. Heuristic Design Concept

While seemingly innocuous, P1 is very difficult to solve to optimality for a large problem, because it is NP-Complete [3]. Thus, to provide a reasonable, powerful approach, use

- LP relaxations to calculate a noninteger solution, then
- clever rounding schemes to ensure a reasonable integer solution.

It can be shown (but is not done here) that the LP relaxation is equivalent to a Lagrangean relaxation of the problem [4].

LP relaxation is a vital means of providing a tight lower bound on the integer programming problem, and we designed our heuristic through a series of LP subproblems. We also explored the development of a heuristic approach using minimum spanning trees, but it was inferior to the LP relaxations.

B. Heuristic Approach

To develop some heuristics based on LP relaxations, a process is defined whereby these relaxations lead to efficient integer solutions in a reasonable computation time. Thus we developed 3 closely related heuristics, all based on LP relaxations: H1-1, H1-2, H1-3.

1) Heuristic 1-1: An Overview: The purpose of H1-1 is to find a completely feasible solution for P1, then improve the solution. A completely feasible solution in this case is one that contains only integer $x_{i,j}$; fractional $x_{i,j}$ are not acceptable.

- $x_{i,j} = 1 \Rightarrow$ a complete link between v_i and h_j.
- $x_{i,j} = 0 \Rightarrow$ an empty (free) link between v_i and h_j.

This heuristic is constructed using *Mathematica*, which was chosen because of its

- innate ability to allow random generation of the data and problem parameters,
- graphic capabilities,
- incorporation of LP.

This heuristic consists of essentially 4 phases:

- Phase 1: Construction of direct links to the MSC.
- Phase 2: Link switching process.
- Phase 3: Capacity searching process.
- Phase 4: Cost minimization.

We use a matrix data structure, $M[i, j]$, which has i cell rows and $j+1$ hub columns to track

- each $x_{i,j}$, and
- the feasibility of the solution.

During each of the 4 phases of the heuristic, matrix M_k is augmented. The matrix built at phase k is then carried to phase $k+1$.

Phases 1–3 are designed to convert a noninteger solution obtained from solving the associated LP into an integer (or a completely feasible) solution. By moving and cutting some of the links in the current solution, the link capacities are freed by a certain amount for the possible addition of new but complete links, while maintaining both capacity & diversity constraints. However, there is a trade-off with a higher total cost of the homing assignment. After changing the link assignment in phases 1–3, an integer solution might be obtained. If so, phase 4 is then used to decrease the current total cost by changing a cell-MSC link to a cell-hub link whose cost is cheaper, while preserving the constraints.

C. Algorithm of Heuristic 1-1

This algorithm finds a feasible solution and then improves it for P1. Appendix A contains a detailed pseudo-code of the algorithm steps. A Mathematica program of these 4 phases is available from the authors upon written request.

1) Phase 1: Construction of Direct Links to MSC: The purpose of phase 1 is to group and cut 1 or $(1 - x_{i,0})$ units of incomplete links (demand) in row i of M_0, and then transfer this amount of demand to the new link, $[i, 0]$, a link connecting cell i and the MSC, given that link $[i, 0]$ is free (i.e., $x_{i,0} = 0$) or partially free. Without a capacity constraint in any link $[i, 0]$, this process applies to any cell regardless of its diversity degree. This Sub-Algorithm-1 begins the process from the first cell until the last cell.

Table I is an example in this section to illustrate the heuristic process. The first matrix M_0 is a noninteger solution of a Problem 1 in which there are 12 cells and 5 hubs (hub 0 is the MSC).

In matrix M_0, Table I, rows 4, 7, 10 have fractional $x_{i,j}$ and among links [4, 0], link [7, 0], link [10, 0], only link [10, 0] has room for extra demand. Therefore, by sending extra $(1 - 3/8)$ units of cell 10 demand to link [10, 0], then link [10, 0] becomes complete (i.e., $x_{10,0} = 1$) while the fractional $x_{10,3}$ becomes 0. A new matrix, M_1, Table I is formed.

2) Phase 2: Link Switching Process: The purpose of phase 2 is to free up the link capacity of hub j^* by transferring 1 unit demand of cell i' from link $[i', j^*]$ to link $[i', 0]$, given that link $[i', 0]$ is free ($x_{i',0} = 0$); i.e., change x_{i',j^*} from 1 to 0 while $x_{i',0}$ becomes 1. The goal is to complete an originally fractional link $[i, j^*]$ by putting extra demand into the link. When the new remaining link capacity of hub j^* is good enough to fit the extra amount of demand in link $[i, j^*]$, i.e.,

TABLE I
MATRIX M_0 ON THE LEFT—MATRIX M_1 ON THE RIGHT

	Hubs				
Cells	#0	#1	#2	#3	#4
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	0	7/15	8/15	0
5	1	0	0	0	1
6	1	0	0	0	0
7	1	0	61/75	0	14/75
8	1	0	1	0	0
9	1	0	0	1	0
10	3/8	0	0	5/8	0
11	1	0	0	0	0
12	0	0	0	0	1

	Hubs				
Cells	#0	#1	#2	#3	#4
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	0	7/15	8/15	0
5	1	0	0	0	1
6	1	0	0	0	0
7	1	0	61/75	0	14/75
8	1	0	1	0	0
9	1	0	0	1	0
10	1	0	0	0	0
11	1	0	0	0	0
12	0	0	0	0	1

TABLE II
MATRIX M_1 ON THE LEFT—MATRIX M_2 ON THE RIGHT

	Hubs				
Cells	#0	#1	#2	#3	#4
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	0	7/15	8/15	0
5	1	0	0	0	1
6	1	0	0	0	0
7	1	0	61/75	0	14/75
8	1	0	1	0	0
9	1	0	0	1	0
10	1	0	0	0	0
11	1	0	0	0	0
12	0	0	0	0	1

	Hubs				
Cells	#0	#1	#2	#3	#4
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	0	7/15	8/15	0
5	1	0	0	0	1
6	1	0	0	0	0
7	1	0	0	0	1
8	1	0	1	0	0
9	1	0	0	1	0
10	1	0	0	0	0
11	1	0	0	0	0
12	1	0	0	0	0

$(1 - x_{i,j^*})\cdot$ (unit demand of cell i), then group $(1 - x_{i,j^*})$ unit demand of cell i from other fractional link(s) in row i of M_1, cut the link(s) and form a complete link $[i, j^*]$. This Sub-Algorithm-2:

1) Determines all the possible candidates and calculates the additional cost of cutting & constructing the links for each candidate.
2) Assigns the extra cost to the associated variable which is identified as $\Delta c_{i,j}$.
3) Chooses the minimum $\Delta c_{i,j}$ and performs the link assignment & removal, if possible, by checking the capacity constraints.
4) Updates $\Delta c_{i,j}$ with a large nonzero value, $\approx \infty$.
5) However, some link assignments and removal might not be performed due to the previous changes of links, i.e., a link $[i, 0]$ has been constructed during any previous iteration in phase 2.
6) Repeat steps 1–5 until all $\Delta c_{i,j}$ are ∞.
7) Form a new matrix M_2.

End_Sub-Algorithm-2

Return to the example in Table II. In M_1, the link [12, 4] is complete, and link [12, 0] is free. Because the cut of link [12, 4] provides enough capacity for an increase of $(1 - 14/75)$ or 61/75 units of cell 7 demand in the link [7, 4], then link [12, 4] is removed, link [12, 0] is built, and link [7, 4] becomes complete while fractional $x_{7,2}$ becomes 0. M_2 is then formed; see Table II.

3) Phase 3: Capacity Searching Process: The purpose of phase 3 is to search for a hub link j whose remaining capacity is good enough for 1 unit demand of cell i'. If so, without violating tree constraints and with a minimum extra cost, 1 unit of fractional link $[i', j]$ are cut and link $[i', j]$ is constructed. This Sub-Algorithm-3

1) Starts searching at cell 1,
2) Changes the link assignment if possible.
3) Repeats steps 1–2 until the last cell has been checked.
4) Forms matrix M_3; see Table III.

End_Sub-Algorithm-3

Return to the example in Table III. Cell 4 has fractional links. Since the remaining capacity of hub 4 is larger than 1 demand unit of cell 4, link [4, 4] is built and fractional $x_{4,2}$ and $x_{4,3}$ are erased.

4) Phase 4: Cost Minimization: The purpose of phase 4 is to compensate the increase of the homing assignment cost which resulted from processing phases 1–3. When the linking cost of link $[i, j]$ is cheaper than the linking cost of link $[i, 0]$ and without violating both diversity & capacity constraints, then by moving the demand of cell i from link $[i, 0]$ to link $[i, j]$, there is a cost saving $cs_{i,j}$.

This idea is obtained from the 1-for-1 (swing) heuristic [1] and the 1-optimal heuristic [7]. Sub-Algorithm-4 functions in a way similar to the one of phase 2. It first determines all possible candidates and assigns each one with a variable $cs_{i,j}$. It then picks the maximum $cs_{i,j}$ and performs the link assignment and removal. One or more $cs_{i,j}$ are then changed to 0. Sub-Algorithm-4 repeats the selection process until all $cs_{i,j} = 0$. A final matrix M_4, Table IV, is obtained.

To complete this example, see matrix M_3 Table IV, cost reductions of homing assignments are gained by moving

- link [4, 0] to link [4, 3],

TABLE III
MATRIX M_2 ON THE LEFT—MATRIX M_3 ON THE RIGHT

	Hubs				
Cells	#0	#1	#2	#3	#4
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	0	7/15	8/15	0
5	1	0	0	0	1
6	1	0	0	0	0
7	1	0	0	0	1
8	1	0	1	0	0
9	1	0	0	1	0
10	1	0	0	0	0
11	1	0	0	0	0
12	1	0	0	0	0

	Hubs				
Cells	#0	#1	#2	#3	#4
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	0	0	0	1
5	1	0	0	0	1
6	1	0	0	0	0
7	1	0	0	0	1
8	1	0	1	0	0
9	1	0	0	1	0
10	1	0	0	0	0
11	1	0	0	0	0
12	1	0	0	0	0

TABLE IV
MATRIX M_3 ON THE LEFT—MATRIX M_4 ON THE RIGHT

	Hubs				
Cells	#0	#1	#2	#3	#4
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	0	0	0	1
5	1	0	0	0	1
6	1	0	0	0	0
7	1	0	0	0	1
8	1	0	1	0	0
9	1	0	0	1	0
10	1	0	0	0	0
11	1	0	0	0	0
12	1	0	0	0	0

	Hubs				
Cells	#0	#1	#2	#3	#4
1	0	0	0	1	0
2	1	0	0	0	0
3	1	0	0	0	0
4	0	0	0	1	1
5	1	0	0	0	1
6	0	0	1	0	0
7	1	0	0	0	1
8	1	0	1	0	0
9	0	0	1	1	0
10	1	0	0	0	0
11	1	0	0	0	0
12	1	0	0	0	0

- link [6, 0] to link [6, 2],
- link [9, 0] to link [9, 2].

M_4 is the final solution of H1-1, see Table IV. The value of the noninteger LP solution of model 1 is $39 625, and H1-1 provides an integer solution with a value of $42 083. Section IV-F provides more detail on the performance of this heuristic.

D. Heuristic 1-2

As mentioned in Section IV-B, H1-2 is based on H1-1. The major difference between these two alternative heuristics is in phase 1. In H1-1, phase 1 constructs direct cell-MSC links. The process continues until there are no more cells that simultaneously have incomplete links to the hubs and an empty (or incomplete) link to the MSC. During each iteration a complete direct cell-hub link is built. The idea is equivalent to adding a new constraint, $x_{i,0} = 1$, at every iteration. Rather than performing the link assignment statically, phase 1 of H1-1 is refined by re-solving the linear program every time a new constraint is included in the problem. By doing this, a more dynamic change of link assignment is gained from the previous noninteger solution and the bounding property of the LP relaxation is tightened more strongly than with the original phase 1. Hence, this concept, compared with phase 1, can lead faster to the integer solution. This re-solving process is repeated until there are no more new constraints that can be added: same process termination criterion as the original one in phase 1. H1-2 is again a 4-phase algorithm which performs the LP re-solving in its first phase, and is followed by phases 2–4 of H1-1.

TABLE V
PROBLEM SIZES

Problem Data	Class I	Class II	Class III
Number of cells	8 – 15	16 – 30	31 – 50
Number of hubs	2 – 4	4 – 6	6 – 9
Number of trees	1 – 4	1 – 2	1 – 3

E. Heuristic 1-3

One important issue about H1-2 is its algorithm running time. As one anticipates, due to the iterative LP re-solving process, when the network instance is large, the running time is appreciably longer. The trade-off between the ability of re-capturing the LP bounding property and the heuristic processing time requires attention. H1-3 is, therefore, developed as a reasonable compromise to H1-1 and H1-2. H1-3 re-solves the LP after only iteration #1 of phase 1, but does not re-solve the LP after every iteration in phase 1. The rest of the heuristics steps follow from H1-1.

F. Experimental Results

To test the heuristics, randomly generated problem instances were created within the Mathematica program. Because of the nature of Mathematica, it was relatively straightforward to perturb the cost parameters and location of the cell & hub nodes so that random problems were generated which also allowed for a graphic output of the network topology.

Table V illustrates the general experimental design where three classes of problems were generated. The relatively small instances & topologies assure that the Integer Programming

TABLE VI
CLASS I

	a. Performance Results							
#	i	j	Q	Z(LP)	H1-1	H1-2	H1-3	Z(IP*)
1	9	3	1	27487	27586	27586	27586	27586
2	10	3	1	38381	38625	38625	38625	38625
3	8	4	1	20493	21846	21370	21846	21314
4	11	4	1	32003	32046	32046	32046	32046
5	10	4	1	25374	26001	26001	26001	25917
6	13	4	1	37627	39174	39174	39174	38646
7	9	3	1	15521	15863	15863	15863	15863
8	10	4	4	18839	19677	19677	19677	19159
9	12	4	1	39714	40720	40356	40356	40356
10	8	3	1	26580	26688	26688	26688	26688
11	11	2	2	25203	25258	25258	25258	25258
12	14	3	1	44049	44268	44268	44268	44268

	b. Percent Deviation from Optimum						
	Time (sec)				Heuristic (%)		
#	1-1	1-2	1-3	IP*	1-1	1-2	1-3
1	0.54	0.52	0.61	0	0.	0.	0.
2	0.51	0.86	0.72	0	0.	0.	0.
3	0.63	1.36	0.70	1	2.5	0.3	2.5
4	0.71	1.46	1.12	0	0.	0	0
5	0.67	2.22	0.93	0	0.3	0.3	0.3
6	1.07	2.67	1.62	1	1.4	1.4	1.4
7	0.45	0.50	0.60	0	0.	0	0
8	0.47	1.33	0.69	1	2.7	2.7	2.7
9	0.94	1.21	1.30	1	0.9	0.	0.
10	0.47	0.78	0.56	1	0.	0.	0.
11	0.39	0.31	0.45	1	0.	0.	0.
12	0.74	1.88	1.14	0	0.	0.	0.
$\bar{x}$	*0.63*	*1.36*	*0.87*	*0.5*	*0.7*	*0.4*	*0.6*

TABLE VII
CLASS II

	a. Performance Results							
#	i	j	Q	$Z(LP)$	H1-1	H1-2	H1-3	Z(IP*)
1	24	5	1	64508	66633	66633	66633	65276
2	16	5	2	56452	59428	59267	59428	58278
3	22	6	2	38600	41422	39750	41384	38952
4	25	6	1	64754	67886	71758	66082	65092
5	24	6	1	59237	60195	61995	60843	59499
6	25	6	1	66859	71904	70320	69192	68352
7	26	6	2	56807	60151	60087	60087	58599
8	18	4	1	51004	51546	52668	52668	51546
9	17	5	1	73310	73766	73584	73766	73584
10	20	6	2	37899	38418	38924	38924	38418
11	18	5	2	69191	71460	70974	70380	70380
12	24	5	2	44925	46116	45376	46096	45196

	b. Percent Deviation from Optimum						
	Time (sec)				Heuristic (%)		
#	1-1	1-2	1-3	IP*	1-1	1-2	1-3
1	5.06	4.86	5.45	1	2.1	2.1	2.1
2	2.18	3.03	2.63	4	2.0	1.7	2.0
3	7.29	39.64	14.39	4	6.3	2.1	6.2
4	11.33	68.16	18.21	5	4.3	10.2	1.5
5	7.16	42.67	14.01	3	1.2	4.2	2.3
6	8.50	46.36	16.52	4	5.2	2.9	1.2
7	8.95	42.86	15.22	508	2.7	2.5	2.5
8	1.69	14.22	3.13	7	0.	2.2	2.2
9	2.48	13.93	4.26	1	0.3	0.	0.3
10	4.76	12.52	8.72	2	0.	1.3	1.3
11	4.05	13.77	6.19	7	1.5	0.8	0.
12	6.23	20.60	10.59	7	2.0	0.4	2.0
$\bar{x}$	*5.81*	*26.89*	*9.94*	*46.08*	*2.3*	*2.5*	*2.0*

model allows computing an optimal solution for comparison. All experiments were carried out on a Dell Dimension XPS H266MHz computer with Windows NT operating system.

G. *Class I Results*

Table VI(a) represents the value of the heuristic solutions for the problems in class I along with the value of the optimum integer solution. Table VI(b) illustrates the fraction deviation from the optimum solution for each of the heuristics. For this class of problems, H1-2 fares pretty well on solution quality, yet experiences a doubling of CPU time over H1-1. The CPU times for the optimal integer LINDO solution were not precise enough to yield values lower than 1 second, which explains the 0's in column 5 of Table VI(b). In later table runs, with larger problem instances, this discrepancy does not appear.

H. *Class II Results*

Table VII(a) represents the value of the heuristic solutions for the problems in class II along with the value of the optimum integer solution. Table VII(b) illustrates the percentage deviation from the optimum solution for each of the heuristics. For this second class of problems, H1-3 does very well, compared to H1-1 and H1-2; yet H1-1 runs the fastest.

I. *Class III Results*

Table VIII(a) represents the value of the heuristic solutions for the problems in class III along with the value of the optimum integer solution. The IP solution was not computable within a reasonable amount of time for P1. Table VIII(b) illustrates the percentage deviation from the optimum solution for each of the heuristics. For this final class of examples, H1-1 emerges as the best approach.

Table IX presents the summary experimental results across all 3 heuristics and all problem classes.

V. P2: SINGLE-PERIOD DEMAND VARIABLE LINK CAPACITY

This P2 formulation allows for variable link capacities. The variable link capacity problem is stated formally as:

$$P2\text{: Minimize } Z = \sum_{i\in V}\sum_{j\in R} c_{i,j}\cdot x_{i,j} + \sum_{l\in H} B_l \cdot y_l \qquad (6)$$

subject to

$$\sum_{j\in R} x_{i,j} = s_i \qquad \text{for all } i \in V; \qquad (7)$$

$$\sum_{i\in V}\sum_{j\in R} \frac{D_i}{s_i}\cdot a_{j,l}\cdot x_{i,j} \le K\cdot y_l \qquad \text{for all } l \in H; \qquad (8)$$

$$\sum_{j\in T_q} x_{i,j} \le 1 \qquad \text{for all } i,\ q = 1, \cdots, Q; \qquad (9)$$

$$x_{i,j} = \{0, 1\} \qquad \text{for all } i,\ j; \qquad (10)$$

TABLE VIII
CLASS III

a. Performance Results

#	i	j	Q	$Z(LP)$	H1-1	H1-2	H1-3	Z(IP*)
1	41	7	2	95986	98213	99605	99509	96029
2	50	7	2	120096	123750	130374	125622	120606
3	41	9	2	130018	131403	131241	130944	130512
4	33	8	2	108391	109158	109245	109245	109158
5	45	7	3	109700	113188	110520	113188	109899
6	41	9	3	56257	56313	56313	56313	—
7	33	9	2	48116	48422	48422	48422	—
8	34	8	2	56932	57448	57304	57304	—
9	41	8	2	106898	108780	110010	110130	—
10	38	9	2	96101	99744	97314	97314	—
11	44	8	2	95027	99496	103432	99640	—
12	35	7	1	94599	95922	97674	95922	—

b. Percent Deviation from Optimum

	Time (sec)				Heuristic (%)		
#	1-1	1-2	1-3	IP*	1-1	1-2	1-3
1	46.29	178.19	89.65	3	2.3	3.7	3.6
2	80.96	457.25	152.96	302	2.6	8.1	4.2
3	85.093	702.13	186.21	445	0.7	0.6	0.3
4	28.95	293.83	49.43	125	0.	0.1	0.1
5	57.28	291.15	113.22	75	3.0	0.6	3.0
6	75.20	72.19	74.99	—	—	—	—
7	38.31	107.96	73.17	—	—	—	—
8	48.23	93.31	93.58	—	—	—	—
9	57.87	277.79	113.44	—	—	—	—
10	64.98	319.45	126.82	—	—	—	—
11	2.46	700.16	157.89	—	—	—	—
12	32.19	249.51	63.83	—	—	—	—
$\bar{x}$	*58.15*	*311.91*	*107.93*	*190*	*1.7*	*2.6*	*2.2*

TABLE IX
SUMMARY—AVERAGES OF THE EXPERIMENTAL RESULTS

Problem Data	H1-1	H1-2	H1-3	IP*
Running Time	21.53	113.35	39.58	78.86
Deviation from Optimum (%)	1.55	1.84	1.59	

$$y_l \geq 0 \quad \text{for all } l \in H. \tag{11}$$

In this P2, K is a constant, e.g., $K = 672$, the capacity of the DS3 link but any nonnegative integer will do; y_l is the number of units to purchase; B_l is the fixed cost per unit.

A. Heuristic

As seen in the problem formulation, P2 has the integral property [4]; thus a series of LP relaxations can be used to generate effective lower bounds as in P1. Section V-D demonstrates that the LP relaxation always yields an integer LP lower bound on the $x_{i,j}$ variables, which proves to be a very effective heuristic procedure.

Running some of the computation experiments provided some very interesting insights for P2; those 3 properties are presented here without proof:

Property 1: P2 guarantees all integer $x_{i,j}$, but the most likely variables, y_l, are nonintegers.

This property is related to the integrality property of the LP relaxation, but unfortunately, we do not see a straightforward proof. In all our experiments to date, the $x_{i,j}$ were 0, 1 which eventually makes the heuristic solution process based on LP relaxations very powerful.

Property 2: Among a set of y_l, any y_l which is the closest to its LIUB, $\lceil\cdot\rceil$, tends to be (but not necessarily) rounded up to that LIUB in the optimal solution.

This property has a fortunate side-effect for our LP-based heuristics.

Property 3: If we round up the sum of the (fractional) y_l, the resulting integer value always equals the sum of integer y_l in the optimal solution.

This property is interesting and led us to explore a valid inequality to include with the LP relaxation process.

Table X illustrates these 3 properties we have found experimentally.

Two alternative heuristics (H2-1, and H2-2) for P2 are proposed, to yield an integer solution for y_l.

B. Heuristic 2-1

A straightforward approach to obtain a feasible solution for P2 is to take the fractional y_l from model 2 and get their LIUB: $\lceil\cdot\rceil$. When a fractional y_l is increased its LIUB, extra link capacity is added into the Hub-MSC link j, the constraint of integer y_l is no longer violated. This gives an easy upper bound on the optimal integer solution value. Using H2-1 one can anticipate that the homing assignment will remain unchanged while the monthly interconnection cost becomes higher.

An alternate approach is to pattern a heuristic similar to H1-1 where the closest y_l is replaced by its LIUB, then resolve the LP, sequentially solve LP relaxations in $O(N)$ time until an integer solution is achieved. We did not implement this heuristic modification because the simplest version worked well enough.

C. Heuristic 2-2

Property 3 in Section V-A shows how to determine the total number of link units that should be purchased. To use this insight well, a new constraint (cutting plane) is constructed which indicates that the sum of y_l must equal the rounded-up sum of fractional y_l from model 2. In addition, similar to the re-solving idea in H1-1, another new constraint is added into the problem every time before the re-solving process takes place. This is a constraint of forcing one y_l to take an integer value, where y_l is chosen based on the idea of Property 2 (the y_l, whose value is its LIUB). The re-solving process terminates when all y_l are integers.

D. P2: Experimental Results

Table XI(a) represents the value of the heuristic solutions for the problems in class I along with the value of the optimum integer solution. Table XI(b) illustrates the fraction deviation from the optimum solution for each of the heuristics.

Table XII(a) represents the value of the heuristic solutions for the problems in Class II along with the value of the optimum integer solution. Table XII(b) illustrates the fraction deviation from the optimum solution for each of the heuristics. H2-1 performs better than H2-2.

TABLE X
EXAMPLE—PROPERTY RESULTS

	Trial #	Hub Capacity Values: Y y_1	y_2	y_3	y_4	y_5	y_6	Total	⌈Total⌉
LP	1	3.75	3.0833	0.8333	1.6667	0		9.3333	10
Integer		4	1	1	4	0		10	10
LP	2	0	0	0.9167	4.8333			5.75	6
Integer		0	0	1	5			6	6
LP	3	5.1667	2.7083	2.7083	0			10.5833	11
Integer		3	3	5	0			11	11
LP	4	1.3333	0	0	2.75	0		4.0833	5
Integer		2	0	0	3	0		5	5
LP	5	3.5420	2.75	0	1.5417	6.8333		14.667	15
Integer		5	3	0	0	7		15	15
LP	6	0	3.5417	3.7917	2.5417	0.5417		10.4168	11
Integer		0	2	4	5	0		11	11
LP	7	2.8333	1.9167	0	2.5	4.5		11.75	12
Integer		7	2	0	0	3		12	12
LP	8	2.9167	0	1.3333	2.0417	0	0	6.2917	7
Integer		3	0	2	2	0	0	7	7
LP	9	4.75	0.6667	0	0.6667	0		6.0834	7
Integer		5	0	2	0	0		7	7
LP	10	2.5833	0	0	4.0417			6.625	7
Integer		5	0	0	2			7	7

TABLE XI
CLASS I (HEURISTIC SOLUTIONS)

	a. Performance Results						
#	i	j	Q	Z(LP)	H2-1	H2-2	Z(IP*)
1	9	4	1	30251	31388	31321	31010
2	9	3	1	19070	20344	20303	19664
3	12	3	1	44121	45038	44765	44374
4	8	3	1	25271	25812	25812	25812
5	9	3	3	39389	40152	41384	39984
6	10	4	1	27112	27534	27729	27444
7	13	3	1	55169	56430	57916	56386
8	14	3	1	61044	62249	65093	61789
9	14	4	1	30906	31370	31370	31370
10	11	4	4	35130	35658	35658	35658
11	11	4	1	25832	27379	27283	26476
12	12	4	1	24312	25274	N/A	24876

	b. Percent Deviation from Optimum				
	Time (sec)			Heuristic (%)	
#	2-1	2-2	IP*	2-1	2-2
1	0.22	1.11	1	1.2	1.0
2	0.16	0.61	1	3.5	3.3
3	0.32	1.01	2	1.5	0.9
4	0.12	0.57	1	0.0	0.0
5	0.40	0.82	1	0.4	3.5
6	0.30	1.41	1	0.3	1.0
7	0.34	0.95	1	0.1	2.7
8	0.42	1.33	6	0.7	5.4
9	0.63	1.80	3	0.0	0.0
10	0.62	2.18	0	0.0	0.0
11	0.37	1.41	1	3.4	3.1
12	0.48	N/A	2	1.6	N/A
$\bar{x}$	*0.37*	*1.20*	*1.67*	*1.1*	*1.9*

TABLE XII
CLASS II: (HEURISTIC SOLUTIONS)

	a. Performance Results						
#	i	j	Q	Z(LP)	H2-1	H2-2	Z(IP*)
1	26	5	1	89507	90622	92469	90358
2	28	6	1	103256	104973	105423	104433
3	25	5	2	96718	97475	97938	97475
4	24	5	2	99490	101248	100861	100438
5	29	6	1	105844	107820	109970	107010
6	24	6	1	58290	61184	59807	59282
7	19	4	1	51503	53233	53710	52614
8	24	5	2	63406	65141	65739	64018
9	19	6	1	66524	68382	69471	67723
10	18	5	1	61427	64574	64685	63098
11	20	4	1	47958	49385	49240	48357
12	25	5	2	83772	85196	84877	84678

	b. Percent Deviation from Optimum				
	Time (sec)			Heuristic (%)	
#	2-1	2-2	IP*	2-1	2-2
1	5.93	26.72	25	0.3	2.3
2	9.88	54.27	990	0.5	1.0
3	5.83	27.87	16	0.0	0.5
4	4.77	21.50	47	0.8	0.4
5	9.46	51.84	1748	0.8	2.8
6	6.75	35.84	128	3.2	0.9
7	1.12	4.73	7	1.2	2.1
8	5.34	21.66	4	1.8	2.7
9	4.83	27.11	10	1.0	2.6
10	1.65	8.36	7	2.3	2.5
11	1.47	5.72	6	2.1	1.8
12	4.67	20.70	24	0.6	0.2
$\bar{x}$	*5.14*	*25.53*	*251*	*1.2*	*1.7*

Table XIII(a) represents the value of the heuristic solutions for the problems in Class III along with the value of the optimum integer solution. Table XIII(b) illustrates the fraction deviation from the optimum solution for each of the heuristics. H2-1, again, performs better than H2-2.

E. Large-Scale Experiments for P2 *and* H*2-1*

Three large cellular network problems are examined; only H2-1 is applied. H2-2 is not considered due to the anticipated long run times. The results are in Table XIV.

TABLE XIII
CLASS III (HEURISTIC SOLUTIONS)

	a. Performance Results						
#	i	j	Q	Z(LP)	H2-1	H2-2	Z(IP*)
1	40	6	2	136492	137693	139112	137019
2	48	6	2	112247	114699	114201	113231
3	37	7	3	66092	67921	67439	66909
4	36	8	2	121741	123610	N/A	122564
5	49	9	1	106635	109933	113486	—
6	45	8	1	150248	152958	161997	—
7	36	7	2	99814	102211	102450	—
8	42	6	2	150679	152109	154155	—
9	48	8	2	113141	116230	116920	—
10	40	7	2	80532	83239	82120	—
11	35	7	2	95389	97543	10309	—
12	34	8	2	121108	124037	130257	—

	b. Percent Deviation from Optimum				
	Time (sec)			Heuristic (%)	
#	2-1	2-2	IP*	2-1	2-2
1	27.58	151.2	2580	0.5	1.5
2	37.09	205.1	7800	1.3	0.9
3	29.01	195.4	780	1.5	0.8
4	37.39	N/A	5400	0.9	N/A
5	99.91	813.8	—	—	—
6	65.79	468.7	—	—	—
7	27.38	114.5	—	—	—
8	5.22	142.9	—	—	—
9	81.12	564.0	—	—	—
10	29.52	179.4	—	—	—
11	25.15	156.2	—	—	—
12	31.34	222.5	—	—	—
$\bar{x}$	*43.04*	*292.2*	*4140*	*1.2*	*1.1*

TABLE XIV
CLASS III: LARGE-SCALE PROBLEM RESULTS

						Time (sec)	
#	i	j	Q	Z(LP)	H2-1	LP	H2-1
1	50	10	3	84387	86872	248.13	249.20
2	67	14	3	102294	106777	479.37	479.41
3	92	15	2	195217	200691	1730.55	1731.00

VI. P3: CONCLUDING REMARKS

This last section

- discusses P3, but does not attempt to solve it,
- addresses some open questions,
- presents further insights.

A. Multi-Period Demand, Variable Link Capacity

P1 & P2 are generalized; demand varies over time. The main difference between P2 and P3 is the addition of the discrete time-period index t which unfortunately increases the dimensions of the problem. The formal statement is:

$P3$: Minimize Z

$$= \sum_{i \in V} \sum_{j \in R} \sum_{t} c_{i,\, j,\, t} \cdot x_{i,\, j,\, t} + \sum_{l \in H} \sum_{t} B_{l,\, t} \cdot y_{l,\, t}; \tag{12}$$

subject to

$$\sum_{j \in R} x_{i,\, j,\, t} = s_{i,\, t} \quad \text{for all } i \in V,\, t; \tag{13}$$

$$\sum_{i \in V} \sum_{j \in R} \frac{D_{i,t}}{s_{i,t}} \cdot a_{j,\, l} \cdot x_{i,\, j,\, t} \le K \cdot y_{l,\, t} \quad \text{for all } K_l \in H,\, t \tag{14}$$

$$\sum_{j \in T_q} x_{i,\, j,\, t} \le 1 \quad \text{for all } i,\, t,\, q = 1, \cdots, Q \tag{15}$$

$$x_{i,\, j,\, t} = \{0,\, 1\} \quad \text{for all } i,\, j,\, t; \tag{16}$$

$$y_{l,\, t} \ge 0 \quad \text{for all } l \in H,\, t. \tag{17}$$

Again, by examining the constraint structure it seems that the LP relaxation process will be beneficial in providing lower bounds on the optimal integer solution to the multi-period programming problem.

B. Problem Variations

Another variation on P2 is incorporating a mix of infinite and finite capacities on the links. The link costs can be incorporated from one hub j to another hub k; the problem then becomes a nice Steiner problem.

APPENDIX
P1 HEURISTIC DETAILS

A. Phase 1

- Step 1: Given a noninteger solution of P1. Form the matrix M_0 to record the current noninteger solution obtained by solving the linear programming problem. Element $M_0[i,\, j]$ is 1, 0 or a fractional value less than 1; these represent respectively, a complete, empty, or incomplete "link between cell i and hub j is constructed." MSC is hub 0. Set $i = 1$.
- Step 2: While $i \le n$, if cell i has incomplete links to hubs (fractional $x_{i,\, j}$ in row i of matrix M_0) and if $M_0[i,\, 0] \ne 1$ (link between cell i and MSC is free or partially free), then construct a complete link $[i,\, 0]$ ($x_{i,\, 0}$ becomes 1) and subtract (1 - original $x_{i,\, 0}$) units of fractional $x_{i,\, j}$ in row i (remove incomplete links). The idea is to move (1 - original $x_{i,\, 0}$) units of cell i demand from those incomplete links to link $[i,\, 0]$. If the original $x_{i,\, 0}$ is 0, then the total demand transfer is 1. Set $i = i + 1$.
- Step 3: A new matrix M_1 is formed.

B. Phase 2

- Step 1: Bring in matrix M_1 from phase 1. Set $i = 1$.
- Step 2: While $i \le n$, if cell i has incomplete links to the hubs ($x_{i,\, j^*}$) then set counter = 0.
    - Step 2.1: counter = counter + 1.
        - Step 2.1.1: Set $i' = 0$.
        - Step 2.1.2: $i' = i' + 1$. If link $[i,\, 0]$ is free and if link $[i,\, 'j^*]$ is complete and if one demand unit of cell i' is larger than 1 demand unit of cell i then multiply i by $(1 - x_{i,\, j^*})$, and

          $$\text{xc}_{i',\, j^*} \equiv c_{i',\, 0} - c_{i',\, j^*}.$$

        - Step 2.1.3: Process until $i' = n$. Repeat step 2.1.2.
    - Step 2.2: If counter < number of fractional $x_{i,\, j}$'s in row 1, then go to Step 2.1, else $i = i + 1$.

- Step 3: While $\min_{i,j}\,[\mathrm{xc}_{i,\,j}] \neq \infty$ go to step 3.1.
    - Step 3.1: Choose $\min[\mathrm{xc}_{i,\,j},\ \mathrm{xc}_{i',\,j^*}]$. If link $[i,\,0]$ is free and if link $[i',\,j^*]$ is complete, then go to step 3.2; else go to step 3.3.
    - Step 3.2: Construct link $[i',\,0]$, remove link $[i',\,j^*]$, complete link $[i,\,j^*]$ by setting it to 1, and subtract $(1 - x_{i,\,j^*})$ from the other fractional $x_{i,\,j}$ in row i.
    - Step 3.3: Set $\mathrm{xc}_{i',\,j^*} = \infty$.
- Step 4: Matrix M_2 is then formed.

C. Phase 3

- Step 1: Bring in matrix M_2. Calculate the remaining link capacities.
- Step 2: Set $\mathrm{xc}_j = \infty$. Set $i = 1$.
- Step 3: While $i \leq n$, if cell i has incomplete links to hubs $(x_{i',\,j})$ set $j = 0$.
    - Step 3.1: $j = j + 1$. If the remaining capacity of hub j_o is larger than 1 demand-unit of cell i' and if the remaining capacity of any hub between hub j_o and MSC (on the same tree of hub j_o) is larger than 1 demand unit of cell i', then calculate xc_j:

$$\mathrm{xc}_{j_o} = c_{i',\,j_o} - \sum_{k=1}^{n} c^k_{i',\,j} \cdot x^k_{i',\,j},$$

k is the index of a fractional number,

n is the number of fractional $x_{i,\,j}$

-
    - Step 3.2: Proceed until $j = m$ (not including MSC). Go to step 3.1.
    - Step 3.3: If $\min_j\,[\mathrm{xc}_j] \neq \infty$, then go to step 3.4, else go to step 3.5.
    - Step 3.4: If the remaining link capacity of hub j' is larger than 1 demand unit of cell i', then set fractional $x_{i,\,j}$ in row i' to 0. Construct link $[i',\,j']$. Update the remaining link capacities. Reset $\mathrm{xc}_\mathrm{j} = \infty$.
    - Step 3.5: Set $i = i + 1$.

D. Phase 4

- Step 1: Bring in M_3. If at least 1 element in M_3 is not an integer, then Stop; else go to step 2.
- Step 2: Calculate the current remaining link-capacities. Set $\mathrm{cs}_{i,\,j} = 0$, $i = 1$.
- Step 3: While $i \leq n$, $(i = i')$. Set $j = 0$.
    - Step 3.1: $j = j + 1$ $(j = j')$. If link $[i',\,0]$ is complete, and if link $[i',\,j']$ is free, and if the remaining link capacity of hub j' is larger than the unit demand of cell i', and if $c_{i,\,0} > c_{i',\,j'}$, then calculate:

$$\mathrm{cs}_{i',\,j'} = c_{i,\,0} - c_{i',\,j'}$$

-
    - Step 3.2: Proceed until $j = m$ (not the MSC). Go to step 3.1.
    - Step 3.3: Set $i = i + 1$.
- Step 4: While maximum $\mathrm{cs}_{i,\,j} \neq 0$. Find maximum $\mathrm{cs}_{i,\,j}$, $\mathrm{cs}_{i^*,\,j^*}$.
    - Step 4.1: If the remaining capacity of hub j^* is larger than 1 demand-unit of cell i^*, then go to step 4.2; else go to step 4.4.
    - Step 4.2: If the remaining capacity of any hub between hub j^* and MSC (on the same tree of hub j^*) is larger than 1 demand-unit of cell i^*, then go to step 4.3; else go to step 4.4.
    - Step 4.3: If the construction of link $[i^*,\,j^*]$ does not violate the tree constraints, then cut link $[i^*,\,0]$ and construct link $[i^*,\,j^*]$ and update the link capacity of hub j^* and set all $\mathrm{cs}_{i,\,j}$ in row i^* to 0; else go to step 4.4.
    - Step 4.4: Set $\mathrm{cs}_{i^*,\,j^*} = 0$.
- Step 5: Matrix M_4 is formed. End of P1 Heuristic

References

[1] L. W. Clarke and G. Anandalingam, “An integrated system for designing minimum cost survivable telecommunications networks,” *IEEE Trans, Systems, Man, Cybernetics—Part A*, vol. 26, pp. 856–862, 1996.

[2] A. Dutta and P. Kubat, “Design of partially survivable networks for cellular telecommunication systems,” *European J. Operational Research*, vol. 118, pp. 52–64, 1999.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*: Freeman & Company, 1979.

[4] A. Geoffrion, “Lagrangean relaxation for integer programming,” *Math Programming Study*, vol. 2, pp. 82–114, 1974.

[5] P. Kubat and R. Vachani, “Facilities planning for cellular networks,” in *13 ITC*, Copenhagen, 1990.

[6] T. L. Magnanti, P. Mirchandani, and R. Vachani, “Modeling and solving the two-facility capacitated network loading problem,” *Operations Research*, vol. 43, no. 1, pp. 142–157, 1995.

[7] C. Monma and D. Shallcross, “Methods for designing communications networks with certain two-connected survivability constraints,” *Operations Research*, vol. 37, pp. 531–541, 1989.

[8] A. Ruiz, J. A. Herrera, and A. J. Gomez, “Optimization of the interconnection links in cellular telephone networks,” *4th INFORMS Telecommunication Conf.*, Mar. 1998.

[9] I. Saniee, “Optimal routing designs in self-healing communications networks,” *ITOR*, vol. 3, no. 2, pp. 187–195, 1996.

[10] R. Vachani, “Design of private telecommunication networks,” GTE Laboratories, TM0027-07-88-446, 1988.

[11] O. J. Wasem, T.-H. Wu, and R. H. Cardwell, “Survivable SONET networks—Design methodology,” *IEEE Selected Areas in Communications*, vol. 12, no. 1, pp. 205–212, Jan. 1994.

[12] T.-H. Wu, *Fiber Network Service Reliability*: Artech House, 1992.

Peter Kubat (M’90) is a Principal Member of the Technical Staff at the Analysis and Planning Systems Department, GTE Laboratories. Prior to joining GTE Labs in 1984, he was a Faculty Member at the W. E. Simon Graduate School of Business Administration, University of Rochester. He graduated in 1969 from the Charles University of Prague (mathematics and physics) and received his D.Sc. (1976) in management science from the Technion-Israel Institute of Technology. His research interests include applied optimization, design & planning of cellular systems and SONET/ATM/FR networks, traffic engineering, and reliability of communication systems. He has published over 50 papers and technical reports.

J. MacGregor Smith received his Bachelors and Masters of Architecture from the University of California at Berkeley, and his Ph.D. in mechanical & industrial engineering from the University of Illinois at Champaign-Urbana. He is a Registered Architect. His academic & professional interests include the design of algorithms for combinatorial optimization and stochastic network problems, the optimal design of facilities, networks, and the design & analysis of manufacturing systems. He has performed sponsored research for the US National Science Foundation, Digital Equipment Corporation, IBM, and US Army Natick R&D Labs. He is a Member of INFORMS, ACM, SIAM, Mathematical Programming Society, and Society of Sigma Xi.

Calvin Yum graduated from the University of Massachusetts at Amherst with a B.S. (1995) and M.S. (1997) in industrial engineering and operations Research. He works in a Massachusetts industry. This paper is partially based on his Masters thesis topic.