7-1-2008

Finding Optimal Solutions to Backbone Minimisation Problems using Mixed Integer Programming

Mike J. Morgan
m.j.morgan@glyndwr.ac.uk

Vic Grout
Glyndwr University, v.grout@glyndwr.ac.uk

Recommended Citation

This paper was presented at the Seventh International Network Conference (INC 2008), 8-10 July 2008, which was held in Plymouth, UK. It was published by the University of Plymouth and details of the conference are available at http://www.cscan.org

Finding Optimal Solutions to Backbone Minimisation Problems using Mixed Integer Programming

Abstract

Attempts to evaluate heuristic algorithms are often hampered by the lack of known exact solutions with which to compare results. This is true, in particular, in the study of network backbone design - to date, a fairly undeveloped area in mathematical optimisation. This paper uses a Mixed Integer Programming (MIP) approach to find optimal solutions to the problem of backbone minimisation in mesh networks. A simple model is formulated and then adapted to reduce the number of variables and constraints. Network reliability issues are then considered and a more complex model introduced. Finally the model is solved using a commercial solver to generate test instances with which to test the accuracy of a simulated annealing (SA) heuristic. The heuristic is shown to be accurate to within a very small error margin and the strengths and weaknesses of the two approaches are discussed.

Keywords

wireless mesh networks, network backbones, mixed integer programming, heuristics

Disciplines

Computer and Systems Architecture | Digital Communications and Networking | Hardware Systems | Systems and Communications

Comments

This paper was presented at the Seventh International Network Conference (INC 2008), 8-10 July 2008, which was held in Plymouth, UK. It was published by the University of Plymouth and details of the conference are available at http://www.cscan.org

Finding Optimal Solutions to Backbone Minimisation Problems using Mixed Integer Programming

Mike Morgan and Vic Grout

Centre for Applied Internet Research (CAIR)
NEWI, Plas Coch Campus, Mold Road, Wrexham. LL12 2AW
mi.morganlv.grout@newi.ac.uk

Abstract

Attempts to evaluate heuristic algorithms are often hampered by the lack of known exact solutions with which to compare results. This is true, in particular, in the study of network backbone design - to date, a fairly undeveloped area in mathematical optimisation. This paper uses a Mixed Integer Programming (MIP) approach to find optimal solutions to the problem of backbone minimisation in mesh networks. A simple model is formulated and then adapted to reduce the number of variables and constraints. Network reliability issues are then considered and a more complex model introduced. Finally the model is solved using a commercial solver to generate test instances with which to test the accuracy of a simulated annealing (SA) heuristic. The heuristic is shown to be accurate to within a very small error margin and the strengths and weaknesses of the two approaches are discussed.

Keywords

Wireless mesh networks, Network backbones, Mixed Integer Programming, Heuristics

1. Introduction

The problem of finding minimum sized backbones in networks is well known. Applications include self-organisation in mobile ad-hoc networks (Guha and Khuller, 1998), relay location in fixed broadband wireless systems (Sen and Raman, 2007) and the positioning of cross-connects (OXCs) in optical fibre networks (Savasini et al, 2007). All of these problems have the same underlying structure, which may be adapted to accommodate particular scenarios by use of additional constraints.

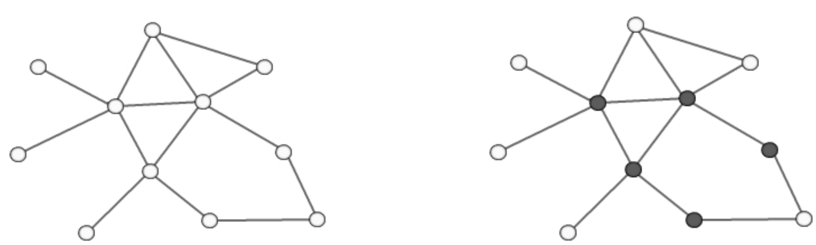

Figure 1a Figure 1b

Figure 1. Network backbones

Consider the example in Figure 1a. A number of locations, represented by nodes in the figure, require connection to one another. Feasible links between these nodes are indicated. In Figure 1b, a subset of these nodes (marked grey) has been selected to relay data on behalf of the others. This subset forms the network *backbone*. Observe that it is possible to send and receive data between any node pair along a path using only backbone nodes (or *relays*) and the two end node-backbone links. It is often advantageous to minimise the number of relays in the backbone whilst still maintaining connectivity between all node pairs. This is equivalent to solving the graph-theoretical *Minimum Connected Dominating Set* (*MCDS*) problem.

At this point we may indicate the relevance of MCDS to the practical examples listed above. By solving this problem, we minimise the number of broadcast nodes in the mobile ad-hoc network or the number of relays/OXCs, and consequently the implementation cost of the fixed broadband and optical fibre networks. Unfortunately, this problem is NP-complete (Garey and Johnson, 1979) and so we are reliant upon exponential time algorithms or heuristics if we wish to solve it.

The MCDS/SA heuristic (Morgan and Grout, 2006) has already been shown to outperform a number of established heuristics. It has been tested against some small, dense problem instances with a known optimum solution, discovered by an exhaustive search method (Morgan and Grout, 2007). The heuristic was found to be accurate in all test cases so it is necessary to find larger and more sparse problem instances with known optimum solutions. This paper introduces a Mixed Integer Programming method of finding such solutions before using them to test the accuracy of MCDS/SA.

2. A Simple MIP Model

Formally, a *dominating set* of a graph $\boldsymbol{G}$=($\boldsymbol{V}$,$\boldsymbol{E}$) is a subset $\boldsymbol{S}$ of the n nodes $\boldsymbol{V}$ such that every node in $\boldsymbol{V}$-$\boldsymbol{S}$ is adjacent, via one of the m edges in $\boldsymbol{E}$, to at least one node in $\boldsymbol{S}$. $\boldsymbol{S}$ is said to be a *connected* dominating set if the subgraph of $\boldsymbol{G}$ induced by $\boldsymbol{S}$ is connected.

The minimum connected dominating set (MCDS) problem involves finding a connected dominating set of minimal size.

To summarize, the MCDS problem involves minimising the number of relays in a subset $\boldsymbol{S}$ of $\boldsymbol{V}$, subject to connectivity and domination constraints. As the connectivity constraint is by far the more difficult to implement, we will begin with a model for the *minimum dominating set* (*MDS*) problem, where the connectivity constraint is ignored.

We define a Boolean *relay vector* $\underline{r} = (r_i)$, of length n, as r_i=1 if node i is a relay and 0 otherwise.

Also, the *adjacency matrix*, $A = (a_{ij})$, of $\boldsymbol{G}$ is an $n \times n$ Boolean matrix, whereby a_{ij}=1 if there is an edge linking nodes i and j in $\boldsymbol{E}$ and 0 otherwise.

Our initial model for Minimum Dominating sets is as follows:

Minimise: $$\sum_{i=1}^{n} r_i \qquad (1)$$

Subject to: $$r_i + \sum_{j=1}^{n} a_{ij} r_j \geq 1 \qquad i = 1..n \qquad (2)$$

r_i binary

The objective function (1) states that the number of relays is to be minimised whilst constraint (2) guarantees that all non-relay nodes have an adjacent relay.

The connectivity constraint requires that we introduce *flow* variables into the problem. We define a *flow matrix*, $F = (f_{ij})$, whereby f_{ij} defines the flow from node i to node j. If the subgraph is connected, we should be able to send flow from a source relay node to all other relay nodes (using only relay nodes as intermediaries). To begin, we shall arbitrarily use node 1 as the source node, requiring that this node be a relay in all candidate solutions. This constraint will then be relaxed by a further adaptation of the model. The additional flow constraints are:

$$\sum_{j=2}^{n} f_{1j} - \sum_{j=2}^{n} f_{j1} = n - 1 \qquad (3)$$

$$\sum_{j=2}^{n} f_{ij} - \sum_{j=2}^{n} f_{ji} \geq r_i \qquad i=1..n \qquad (4)$$

$$0 \leq f_{ij} \leq n r_i a_{ij} \qquad i=1..n,\ j=1..n \qquad (5)$$

$$r_1=1 \qquad (6)$$

Constraint (3) ensures that the source node produces sufficient flow to supply at least one unit of flow to all other relays. Constraint (4) forces each relay to consume at minimum one unit of flow. Additionally, flow may only originate from a relay and travel along a valid edge. This is expressed in (5) along with the more trivial constraint that flows must be positive. The source relay is fixed in (6).

Now it is necessary to modify the model to deal with the situation where node 1 is not a relay. In this case, node 1 must transfer all its flow to one (and only one) of its relay neighbours, effectively making the neighbour node the source of flow for the relay component. We know that node 1 must have a relay neighbour because the domination constraint requires it to. To achieve this, constraint (5) must be replaced by the following:

$$0 \leq f_{1j} \leq n a_{1j}(r_1 + r_j) \qquad \text{j}=2..n \qquad (7)$$

$$0 \leq f_{ij} \leq n a_{ij} r_i \qquad i=2..n,\ j=2..n \qquad (8)$$

Inequality (7) makes it possible for node 1 to transmit flow if it is not a relay, so long as the receiving node is a relay. The original constraint from (5) is then replaced by (8), only this time it applies to all nodes *except* 1. It remains necessary to limit the number of neighbours node 1 can transfer its flow to. To do this, a binary vector $\underline{q} = (q_i)$ is created such that q_j=1 if flow is permitted from node 1 to node j and 0 otherwise. Constraints are added as follows:

$$\sum_{j=2}^{n} q_j \leq 1 + r_1 n \tag{9}$$

$$f_{1j} \leq q_j n \qquad j=2..n \tag{10}$$

q_j binary

Constraint (10) ensures that flow can only be transferred from node 1 to j if q_j=1, while (9) forces q_j to be one for only one node j, if node 1 is not a relay.

Therefore, the simple model for the MCDS problem is as follows:

$$\text{Minimise:} \quad \sum_{i=1}^{n} r_i \tag{11}$$

$$\text{Subject to:} \quad r_i + \sum_{j=1}^{n} a_{ij} r_j \geq 1 \qquad i = 1..n \tag{12}$$

$$\sum_{j=2}^{n} f_{1j} - \sum_{j=2}^{n} f_{j1} = n - 1 \tag{13}$$

$$\sum_{j=1}^{n} f_{ji} - \sum_{j=1}^{n} f_{ij} \geq r_i \qquad i=2..n \tag{14}$$

$$0 \leq f_{1j} \leq n a_{1j} (r_1 + r_j) \qquad j=2..n \tag{15}$$

$$0 \leq f_{ij} \leq n a_{ij} r_i \qquad i=2..n,\ j=2..n \tag{16}$$

$$\sum_{j=2}^{n} q_j \leq 1 + r_1 n \tag{17}$$

$$f_{1j} \leq q_j n \qquad j=2..n \tag{18}$$

r_i, q_j binary

3. A Refined MIP Model for the MCDS problem

The model outlined in section 2 is accurate but also time and space inefficient. This situation is improved by replacing the adjacency matrix by a combination of lists, reducing the number of flow variables and constraints from $O(n^2)$ to $O(m)$. In practice, this constitutes a massive decrease in memory and processor time consumption on all but the most dense problem instances. We define two adjacency vectors a' and a'' along with an index vector y. The length of a' and a'' is m, and the length of y is n+1. a' lists the destination nodes of each edge in ascending order, beginning with those edges with a source at node 1, followed by node 2 etc. Each element y_i denotes the position in a' of the beginning of node i's edge list. The last value y_{n+1} is a delimiting index equal to m+1. a'' denotes the position of the reverse edge for all edges in a'.

The number of flow variables can now be reduced to m. A vector $f' = (f_i')$ denotes the flow across each edge in a'. As each edge (i,j) is listed twice in a', once with i as the source and once with j, the vector y can be used to index the flows out of each node whilst a'' indexes the flows into each node. The problem can be rewritten thus:

Minimise: $$\sum_{i=1}^{n} r_i \qquad (19)$$

Subject to: $$r_i + \sum_{j=y_i}^{y_{i-1}-1} r_{a'_j} \geq 1 \qquad (20)$$

$$\sum_{j=1}^{y_2-1} f'_j - \sum_{j=1}^{y_2-1} f'_{a''_j} = n-1 \qquad (21)$$

$$\sum_{j=y_i}^{y_{i+1}-1} f'_{a''_j} - \sum_{j=y_i}^{y_{i+1}-1} f'_j \geq r_i \qquad i=2..n \qquad (22)$$

$$0 \leq f'_j \leq n(r_1 + r_{a'_j}) \qquad j=1..y_2\text{-}1 \qquad (23)$$

$$0 \leq f'_j \leq nr_i \qquad i=2..n,\ j=y_i..y_{i+1}\text{-}1 \qquad (24)$$

$$\sum_{j=1}^{y_2-1} q_j = 1 + nr_1 \qquad (25)$$

$$f'_j \leq q_j n \qquad j=1..y_2\text{-}1 \qquad (26)$$

r_i, q_j binary

Constraints (20) to (26) are direct replacements for the constraint set given in section 2. The most significant alteration is in (24), where the number of constraints is reduced from n^2 (8) to m. The length of the binary vector q is also reduced from n to the degree of node 1, significantly reducing the number of integer variables in the model for most cases.

4. 2-Connected Backbones

The MCDS problem is a good basis for backbone minimisation problems in general but it does not provide redundancy to the backbone in the event of failure. To introduce a constraint which handles this, we offer the following definition: A graph is said to be k-connected if k or more node (or edge) deletions are required to disconnect it. In other words, a k-connected network may survive k-1 node or edge failures. A k-connected dominating set (k-*CDS*) is a CDS in which the backbone (the subgraph induced by $\boldsymbol{S}$) is k-connected.

The following theory is due to Kleitman (1969) and can be used to produce a constraint set for the Minimum 2-CDS problem:

G=(***V***,***E***) is k-connected if for any node v in ***V***, there are k node-independent paths from v to each other node and the graph ***G'*** formed by removing v and all its incident edges from ***G*** is (k-1)-connected

We present a model for 2-connected backbones using this theorem. Firstly, we must test that there are two vertex independent paths from some relay node to all other relays on the backbone and secondly, we must establish that the backbone remains connected when this node is deleted. The second part may be established using the same connectivity constraint as in the previous two sections, with the alteration that flow may not pass through the source node for the vertex-independent paths calculation.

The vertex independent path constraints may be implemented using multicommodity flows (Ahuja et al, 1993). The source node (which we fix as node 1 in this model) must transmit 2 units of a commodity to each relay; subject to the constraint that only 1 unit of each commodity may be transmitted through the intermediate relays. Notice that this increases the number of variables and constraints to O(nm) as we must measure flow across each edge for all n-1 commodities.

In addition to this, fixing node 1 as a relay means that the connectivity constraints from the previous section need a new source. This can either be achieved by fixing a second relay or by using the same method above using some node s as the source, where s is not adjacent to node 1 (in which case s might have no neighbouring relay besides 1 to transmit its flow to). In the model that follows we fix two nodes, although it should be noted that fixing one node could be more acceptable in practice.

We introduce an $n \times m$ matrix $F'' = (f_{ij}'')$ where f_{ij}'' denotes the flow of the commodity destined for node i across edge j. The following constraints ensure that there are two vertex independent paths from node 1 to all other relays:

$$\sum_{j=1}^{y_2-1} f''_{ij} - \sum_{j=1}^{y_2-1} f''_{ia''_j} = 2r_i \qquad i=2..n \qquad (27)$$

$$\sum_{j=y_i}^{y_{i+1}-1} f''_{ia''_j} - \sum_{j=y_i}^{y_{i+1}-1} f''_{ij} = 2r_i \qquad i=2..n \qquad (28)$$

$$\sum_{j=y_i}^{y_{i+1}-1} f''_{kj} - \sum_{j=y_i}^{y_{i+1}-1} f''_{ka''_j} = 0 \qquad i=2..n,\ k=2..n,\ k \neq i \qquad (29)$$

$$\sum_{j=y_i}^{y_{i+1}-1} f''_{kj} <= 1 \qquad i=2..n,\ k=2..n,\ k \neq i \qquad (30)$$

$$\sum_{j=y_i}^{y_{i+1}-1} f''_{ka''_j} <= 1 \qquad i = 2..n,\ k = 2..n,\ k \neq i \tag{31}$$

$$f''_{kj} \leq r_i \qquad i = 1..n, j = y_i..y_{i+1}\text{-}1, k\text{=}1..n \tag{32}$$

Equations (27) and (28) state that two units of flow for commodity i must be sent from node 1 and received at node i if node i is a relay. Equation (29) is the flow conservation constraint; flow for commodity i cannot be lost or gained at nodes other than 1 or i. Equations (30) and (31) state that no more than one unit of a commodity's flow may pass into or out of a node unless it is the source (node 1) or sink (node i) for that commodity. Flows are constrained only to originate from relays in (32).

Finally we need to modify the original flow equations to use node 2 as source (33),(34). We must also restrict these flows from entering or leaving node 1 as it has been 'deleted' at this stage of Kleitman's theorem (35),(36).

$$\sum_{j=y_2}^{y_3-1} f'_j - \sum_{j=y_2}^{y_3-1} f'_{a''_j} \leq n-2 \tag{33}$$

$$\sum_{j=y_i}^{y_{i+1}-1} f'_{a''_j} - \sum_{j=y_i}^{y_{i+1}-1} f'_j \geq r_i \qquad i\text{=}3..n \tag{34}$$

$$\sum_{j=1}^{y_2-1} f'_j = 0 \tag{35}$$

$$\sum_{j=1}^{y_2-1} f'_{a''_j} = 0 \tag{36}$$

These equations replace (21) and (22) from the model in section 3. Equations (23), (25) and (26) are removed.

Finally, if we wish to give all terminal nodes two adjacent relays, making our topology 2-connected throughout rather than just across the backbone, we may change the domination constraint (20) to:

$$r_i + \sum_{j=y_i}^{y_{i-1}-1} r_{a'_j} \geq 2 \tag{37}$$

This creates a 2-connected, 2-dominating set.

5. Results

To generate problem instances, nodes were positioned randomly within a unit square. Two parameters were defined: a Maximum Transmission Distance (MTD) and a

Line of Sight probability (LOS). Nodes were given an edge between them with probability LOS if they lay within MTD of one another. The instances were then solved using the CPLEX solver (ILOG, 2008).

Initially the SA algorithm was tested for problem instances ranging from 50-100 nodes with MTD and LOS set so that the graphs were as sparse as they could be whilst still being connected. As a result the values of MTD and LOS changed as the node count increased. This is demonstrated by Table 1.

Table 1: Parameters for initial set of instances

Nodes	**50**	**60**	**70**	**80**	**90**	**100**
MTD	0.3	0.3	0.3	0.3	0.3	0.2
LOS	0.4	0.4	0.4	0.3	0.3	0.5

The results are shown in Table 2. It can immediately be seen that the performance of the SA algorithm is not always optimal for these problems. The difference in performance between the SA algorithm and the MIP approach is very small, however, remaining below 2.5% for all these instances. It is also encouraging to see a maximum error margin of only 1 node, indicating that the SA result was never more than 1 relay above optimal. Figure 2 shows the maximum and mean runtimes for SA and MIP, demonstrating the time saving of SA and the unpredictability in runtimes for the MIP model.

Table 2: Mean results for variable *n* (10 runs)

Nodes	**50**	**60**	**70**	**80**	**90**	**100**
SA	15.5	15.6	17	22	21.2	27.9
Optimum	15.5	15.6	16.6	21.9	20.7	27.8
Difference (%)	0	0	2.41	0.46	2.42	0.36
Exact Solutions (%):		81.67	**Max Error Margin:**			1

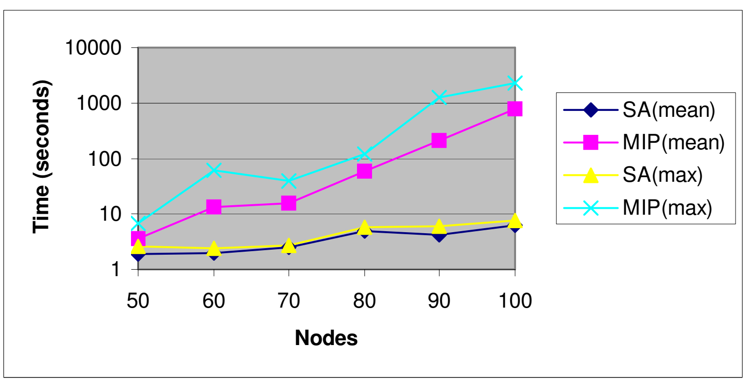

Figure 2: Mean and Maximum runtimes for tests in Table 2

Table 3 shows the performance of SA with increasing problem density. Here we can see very little change in the algorithm's accuracy, along with the same tight error margin. Figure 3 shows that runtimes were consistent once again compared to the unpredictability of MIP.

Table 3: Mean results for variable LOS with n=70, MTD=0.3 (10 runs)

LOS	0.3	0.5	0.7	0.9
SA	20.7	13.7	10.6	8.5
Opt	20.6	13.6	10.5	8.5
Diff	0.485437	0.735294	0.952381	0
Exact Solutions (%):		92.5	Max Error Margin: 1	

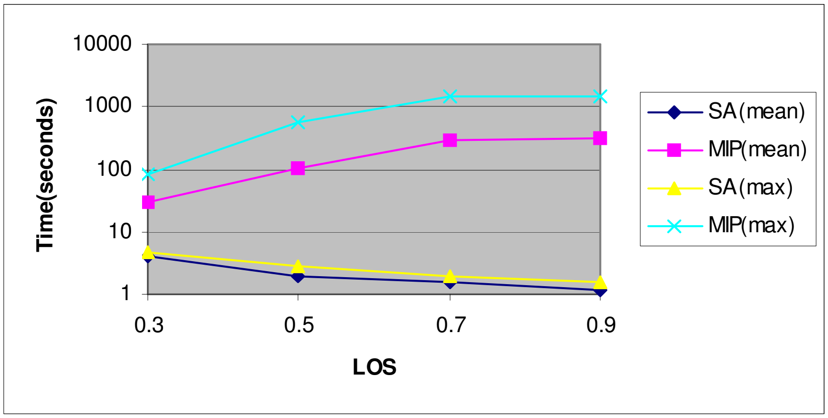

Figure 3: Mean and maximum runtimes for tests in Table 3

Finally, problem instances were tested using the 2-connectivity model from section 4. Size was restricted here due to the increase in the number of variables and constraints. Again MTD and LOS were set so that the graphs would be sparse (Table 4), but this time they would need to be 2-connected. Table 5 shows the results, which indicate a decrease in the SA algorithm's accuracy along with an increase in maximum error margin to 2 relays. Runtimes are given in Figure 4, indicating a similar trend to that shown in Figures 2 and 3.

Table 4: Parameters for 2-connected backbone tests

Nodes	30	40	50	60	70
MTD	0.5	0.4	0.4	0.4	0.4
LOS	0.4	0.4	0.4	0.4	0.3

Table 5: Mean results for 2-connected backbones with variable *n*

Nodes	30	40	50	60	70
SA	10.5	12.8	13.3	12.8	16
Opt	10.2	12.3	12.5	12.6	15.6
difference	2.941176	4.065041	6.4	1.587302	2.564103
Exact Solutions (%): 64		Max Error Margin: 2			

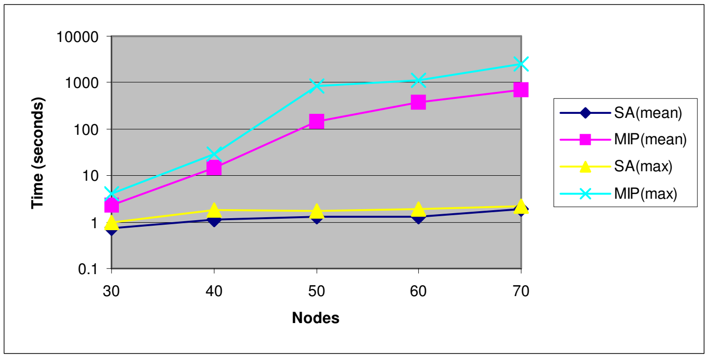

Figure 4: Mean and maximum runtimes for tests in Table 5

6. Conclusions

MCDS/SA has shown itself to be an accurate heuristic for finding connected dominating sets in the majority of cases tested. The time saving for this method is clear, although the MIP model may be preferred in cases where the problem is small as it guarantees to find the optimal solution. The results present a strong case for the use of heuristics and establish the limitations of exact methods more clearly. An outline for future work would include testing the heuristic with still larger problem instances using graphs designed with regular features so that the optimum CDS size can be known at the outset. Also the potential for a faster IP formulation of the model would need to be investigated.

7. References

Ahuja, R., Magnanti, T. and Orlin J. (1993) *Network Flows,* Prentice Hall.

Garey, M. and Johnson, D. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman.

Guha, S. and Khuller, S. (1998) "Approximation Algorithms for Connected Dominating Sets" *Algorithmica,* 20: 374-387.

ILOG (2008) *CPLEX*, available from http://www.ilog.com/products/cplex/ (January 2008).

Kleitman, D. (1969) "Methods of Investigating Connectivity of Large Graphs" *IEEE Transactions on Circuit Theory (Corresp.),* CT-16, pp232-233.

Morgan, M. and Grout, V. (2006), "Optimisation Techniques for Wireless Networks", *Proceedings of the Sixth International Network Conference: INC 2006,* Plymouth, pp339-346.

Morgan, M. and Grout, V. (2007), "Virtual Backbone Configuration in Wireless Mesh Networks" *Proceedings of the Third Collaborative Research Symposium on Security, E-learning, Internet and Networking: SEIN 2007,* Plymouth, pp231-237.

Savasini, S. Monti, P. Tacca, M. Fumagalli, A. and Waldman, H. (2007) "Regenerator Placement with Guaranteed Connectivity in Optical Networks", *Proceedings of the 11th International IFIP TC6 Conference, ONDM 2007,* Athens, Greece, pp438-447.

Sen, S. and Raman, B. "Long Distance Wireless Mesh Network Planning: Problem Formulation and Solution", *Proceedings of the Sixth International World Wide Web Conference (WWW2007),* Banff, Canada, pp893-903.