2-23-1999

A Simulated Annealing Approach to Communication Network Design

Stephen J. Sugden
Bond University, ssugden@bond.edu.au

Marcus Randall
Bond University, marcus_randall@bond.edu.au

Graham McMahon
Bond University

A Simulated Annealing Approach to Communication Network Design

Marcus Randall Graham McMahon
Stephen Sugden
School of Information Technology
Bond University, QLD 4299

February 23, 1999

Abstract

This paper explores the use of the meta-heuristic search algorithm Simulated Annealing for solving a minimum cost network synthesis problem. This problem is a common one in the design of telecommunication networks. The formulation we use models a number of practical problems with hop-limit, degree and capacity constraints. Emphasis is placed on a new approach that uses a knapsack polytope to select amongst a number of pre-computed traffic routes in order to synthesise the network. The advantage of this approach is that a subset of the best routes can be used instead of the whole set, thereby making the process of designing large networks practicable. Using simulated annealing, we solve moderately large networks (up to 30 nodes) efficiently.

1 Introduction

Increasingly, the most valuable commodity that societies possess is information. As such there is great interest in the efficient storage and transference of this information. Of great importance therefore is the design of efficient telecommunications networks. This paper investigates the network synthesis problem.

The aim of the design problem is to satisfy all traffic requirements at minimum cost between a set of nodes. For the network synthesis problem involving n nodes, there are n^{n-2} possible topologies, e.g. one billion possibilities for a network as small as 10 nodes. The information required to formulate the problem is the traffic demand between each origin and destination (O-D) pairs, and the cost function for carrying traffic on each (possible) link between nodes i and j.

Recent interest in the problem has been in the application of Genetic Algorithms (GAs). Work by Berry, Murtagh, Sugden and McMahon (1995, 1997) has demonstrated this approach in which a GA is used to synthesise the network while a Linear Program (LP) is used to allocate the traffic amongst the routes of the network. However, this approach has the potential to be inefficient due to the typically large LP component (though no empirical investigation was undertaken in Berry et al. (1995, 1997)). Other solution techniques have been described in Balakrishnan, Magnanti, Shulman and Wong (1991), Gavish (1985), Kershenbaum and Peng (1986), Minoux (1989) and Sharma, Mistra and Bhattacharji (1991).

2 Formulations

There are numerous ways that the network synthesis problem can be formulated mathematically. We first discuss a traditional linear programming approach that has been used in the literature (Berry et al. 1995, 1997). From this, a model is developed that uses a knapsack polytope to select amongst a number of pre-computed routes between nodes in the network.

Let $\mathbf{G}$ be the set of all undirected graphs on n nodes with G a member of this set. We represent G by its upper triangular node-node adjacency matrix B with elements b_{ij}. The problem is to find a member G^* which minimises the cost of transporting required origin-destination (O-D) flows subject to specified link, node capacity, node degree and chain hop-limit constraints. The total bandwidth (flow) requirement on virtual path connections between O-D pair $p-q$ is given by F^{pq} (without loss of generality represented as an element of an upper triangular matrix). The partial flow along the r^{th} route between node p and node q is denoted by h_r^{pq} and C_r^{pq} is the cost per unit flow on this route.

$$\underset{G \epsilon \mathbf{G}}{Minimise} \sum_{p=1}^{n} \sum_{q>p} \sum_{r} C_r^{qp} h_r^{pq} \tag{1}$$

s.t.

$$f_{ij} = \sum_{p=1}^{n} \sum_{q>p} \sum_{r} a_{ij,r}^{qp} h_r^{pq} \qquad \forall i, j \tag{2}$$

$$\sum_{r} h_r^{pq} = F^{pq} \qquad \forall p, q \tag{3}$$

$$\frac{1}{2}\left[\sum_{p=1}^{n}(F^{pi} + F^{ip}) + \sum_{j \neq i} f_{ij}\right] \leq u_i^{max} \tag{4}$$

$$0 \leq f_{ij} \leq f_{ij}^{max} \qquad \forall i, j \tag{5}$$

$$\sum_{(i,j)} a_{ij,r}^{pq} \leq h^{max} \qquad \forall p, q, r \tag{6}$$

$$\sum_{j=1}^{n}(b_{ij} + b_{ji}) \leq d_i^{max} \qquad \forall i \tag{7}$$

$$0 \leq h_r^{pq} \qquad \forall p, q, r \tag{8}$$

Where:

N is the number of nodes.

F^{pq} is the total bandwidth (flow) requirement between O-D pair $p - q$.

$a_{ij,r}^{pq}$ is 1 if the link $p - q$ exists on route r between nodes i, j, 0 otherwise.

f_{ij}^{max} is the upperbound on the available capacity (total flow) on the link (i, j).

u_i^{max} is the upperbound on total flow at node i. This flow comprises traffic originating at, terminating at and traversing node i.

H^{max} is the upperbound imposed on the number of links in a route (the hop limit).

h_r^{pq} is the amount of traffic routed between p and q on route r.

Equations 1 to 7 represent the model of the network synthesis problem. Equation 1 is the objective function in which the traffic costs along the routes is minimised. Equation 2 is used to calculate the total flow on each link. Constraint 3 ensures that the bandwidth is distributed among the appropriate routes. Constraint 4 ensures that the node capacities are not exceeded while 5 ensures that the link capacities are not exceeded. The hop-limit is preserved in 6 and ensures that the nodes meet the node degree constraints are modeled in 7.

2.1 Pre-Computed Routes Approach

In this paper, we propose the idea that a set of routes can be pre-computed before the application of an optimisation technique. A route is a sequence of nodes between an origin and destination node and has a length less that or equal to the hop-limit. For instance, a route between O-D pair (1,5) might be (1,2,4,5). The set of possible routes can be large for moderately size problems, therefore we have developed techniques for reducing the size of the set. These are discussed in Section 4.

The pre-computed routes approach has several distinct advantages which are: hop limits are automatically satisfied and routes are feasible (i.e. there are no circuits or repeated nodes in the routes). The problem becomes one of selection of routes in much the same way as items are selected for a knapsack in the knapsack problem (Petersen 1967). The selected routes then form the telecommunications network (the synthesis part of the problem). The only potential disadvantage is that it can be memory intensive if a large set of routes is used (especially if the routes have a high hop-limit). The subset approach discussed in Section 4 does lessen this problem.

Using this approach, we can apply common local search operators within simulated annealing (SA) (Osman and Kelly 1996) to locate feasible solution states. Another part of the problem is to allocate traffic amongst the selected routes. Both these topics are discussed below.

2.1.1 Appropriate Transition Operators

The most appropriate local search transition operators for the network synthesis problem are *add*, *drop* and *change*. These operators are expressed graphically in Figure 1 and are described as:

> *Add*: A route is added to the network.
>
> *Drop*: A route is removed from the network.
>
> *Change*: A route in the network is changed to another route. This is equivalent to adding a route and dropping another route.

Desired location of Figure 1

Multiple transition operator neighbourhoods can be explored in the course of solving a particular problem. In our SA implementation, *add*, *drop* and *change* are assigned an equal probability of being selected at each iteration of the algorithm. This is like the weighted probability wheel used in GA roulette wheel selection by Goldberg (1989).

2.1.2 Allocating Traffic Amongst Routes

The SA algorithm is used to *synthesise* a suitable network topology. As there may be more than one route per O-D pair, a subproblem is to determine an appropriate allocation of traffic for each route. There are two broad methods of achieving this: *exactly* - the optimal allocation of traffic to routes is determined; and *heuristically* - the allocation is determined by a special-purpose algorithm. The first approach has been used in Berry et al. (1995, 1997), however it can be time consuming for medium to large size problems. In contrast, the heuristic approach does not seek the optimal allocation, but one that satisfies Equation 3. We have adopted the latter approach in order to produce good solutions within a reasonable amount of computer time.

Our heuristic is very simple yet very efficient in terms of the computational time required to perform it. In essence, for each O-D pair and its set of routes (those in the current solution), the algorithm proportionally loads traffic on each route according to its cost. Cheaper routes will therefore be given a greater amount of traffic than more expensive routes. Further study could examine the relationship between the complexity of the heuristic and the affect on the objective function. The algorithm is given in Figure 2.

Desired location of Figure 2

3 Simulated Annealing

Simulated annealing (SA) is a general purpose meta-heuristic method that has been applied successfully to a number of combinatorial optimisation problems (Collins, Eglese and Golden 1988; Eglese 1990; Koulamas, Antony and Jansen 1994; van Laarhoven and Aarts (1987).

Figure 1: The transition operators *add*, *drop* and *change*. The diagrams demonstrate the effect of each operator on the original network.

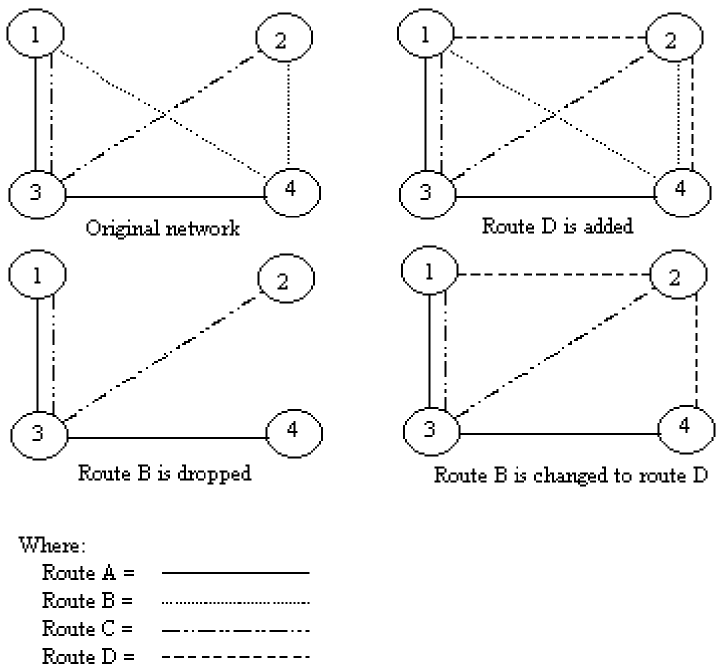

Figure 2: A heuristic algorithm to allocate traffic to each route between every O-D pair.

```
For each O-D pair (i, j)
   sc = 0;
   For(n ϵ routes in current solution between (i, j))
         sc = sc + rc(n);
   End for;
   If(number of routes in the solution between (i, j)>1)
         sca = 0;
         For(n ϵ routes in current solution between (i, j))
               sca = sca + rc(n);
         End for;
         For(n ϵ routes in current solution between (i, j))
               h(i, j, n) = d(i, j) × (sc − rc(n)) ÷ sca;
         End for;
   Else
         n =number of the single route between (i, j);
         h(i, j, n) = d(i, j);
   End if;
End for;
End.
```

Where:

$rc(n)$ is the cost of route n.

$h(i, j, n)$ is the allocation of traffic of route n between (i, j).

$d(i, j)$ is the demand between nodes i and j.

The theory of SA is derived from the physics of annealing substances. Simulated annealing seeks to minimise an energy function, which in combinatorial optimisation is the objective function. At the beginning of the annealing run there is a high likelihood of accepting any transition made in the search space (whether it improves the solution or not) rather than later in the run. Each transition consists of choosing a variable at random and giving it another value (also at random). This process is done in accordance with an exponential acceptance function based on a parameter called *temperature*. The temperature is decremented until it is quite small and hence very few uphill moves (where a worse solution may replace the current solution) are accepted. As SA can make these uphill moves, settling into a local optima is potentially avoided. The way the temperature is controlled is referred to as the *cooling schedule*.

4 Implementation

4.1 Route Subsets

The full set of routes for a practical size problem can be extremely large. Therefore our software allows the user to specify the percentage of routes that form the pre-computed route set. These routes are chosen according to their cost (i.e. the cost of including the route if it was fully loaded). For instance, if 10% is specified, only 10% of the best routes for each O-D pair would be included in the route set.

4.2 SA Algorithm

The SA search engine implements Connolly's (Connolly 1990) Q8-7 cooling schedule as it has been shown to be quite successful (Abramson and Randall 1999; Connolly 1990). The cooling schedule is based on reheating the temperature a number of times throughout the search (shown in Figure 3). Both the number of times that this reheating occurs as well as the interval between reheats can be altered.

Desired location of Figure 3

4.3 Generating Initial Feasible Solutions

As the problem is highly constrained, it can be difficult to form an initial feasible solution. However, the approach we have adopted consists of two phases. In the first phase, an initial solution is constructed so that at least one route per O-D pair (with non-zero demand requirements) is present.

The second phase attempts to modify the solution so that it becomes feasible. This is achieved by using SA in order to minimise the amount of constraint violation. When this becomes 0, a feasible solution has been found. The degree of violation is calculated according to the relational operator of each constraint. For instance, if the sign of the constraint is $\leq$ and the left hand side is larger than the right hand side, the net difference is the amount of constraint violation. The constraint violation of the other signs are calculated in a similar manner.

5 Computational Experience

A set of problem instances have been generated with which to test the SA solver (see Table 1). We have generated these instances with demand matrices that are 90% dense. This means that 90% of the O-D pairs have non-zero traffic requirements. Subsequently this should make it difficult for SA to solve these problems. These problems (as well as a problem generator) are available at `http:\\tide.it.bond.edu.au\mrandall\tele.exe`.

Desired location of Table 1

The size of the pre-computed route set for each problem is also varied in order to determine whether this factor has an effect on solution quality and runtime efficiency. We use route subsets of 10%, 50% and 100% of the total number of routes available for each problem instance. The hop limit has been set to 3.

The computer on which these tests are run is a Sun Ultra Sparc 1 Workstation. Each problem instance is run across 10 random seeds.

The results are given in Table 2. Each run for the 5, 10, 20 node problems is terminated after 600 seconds of CPU time has elapsed. The 30 node problems require a greater amount of time and as such each run is terminated after 1800 seconds of CPU time has elapsed. As the data are highly non-normally distributed, non-parametric descriptive statistics are used. The results table is divided into two sections, **Cost** and **Runtime**. For each section, the minimum (denoted **Min**), median (denoted **Med**), maximum (denoted **Max**) and inter-quartile range (denoted **IQR**) are used. The **Runtime** section records the amount of CPU time required to reach the best solution for that particular run.

Desired location of Table 2

6 Conclusions

We have shown that simulated annealing is capable of solving a complex network synthesis problem with difficult constraints. This problem has applications in the domain of telecommunication network design.

The problem is modeled in a novel manner using an approach that uses a set of suitable pre-computed routes between each O-D pair. The problem then becomes one of selecting an appropriate subset of routes in order to synthesise the network topology. A specialised heuristic is used to allocate the traffic for each of the routes between every O-D pair. This heuristic is a greedy method that works by allocating greater amounts of traffic to cheaper routes. However, it is very efficient (in terms of runtime) compared with the LP method used in Berry et al. (1995, 1997).

The experimental results indicate that using larger size route sets generally produce superior results to smaller sizes. In the case of the small problems (5 and 10 nodes), the use of subsets of routes (10% and 50%) means that initial feasible solutions can not be formed. This is because suitable combinations of routes (from the subsets) that satisfied all the constraints could not be found. In contrast, for the 20 and 30 node problems, it is impractical to use large route sets, as the performance of the algorithm is substantially degraded. For the large problems, it was better to use route set densities of 10% or 50%. Therefore, for very large network design, this method is very appropriate as only a small subset need be used in order to produce an efficient

Figure 3: SA re-heating schedule.

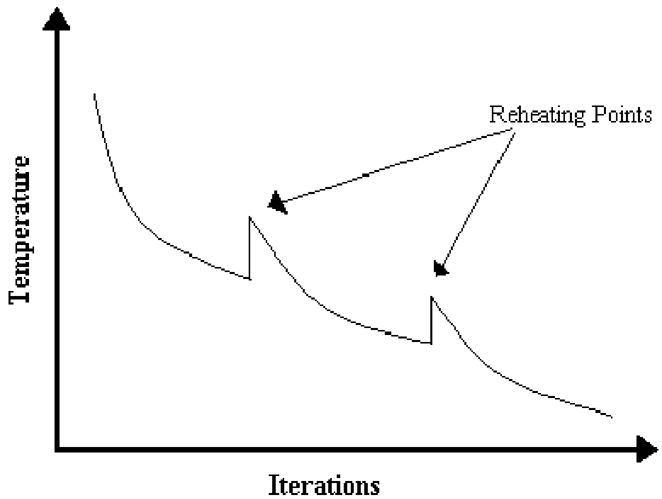

Table 1: Problem instances used in this study.

Name	Size(Nodes)
tele5-1	5
tele5-2	5
tele10-1	10
tele10-2	10
tele20-1	20
tele20-2	20
tele30-1	30
tele30-2	30

Table 2: Results of running the SA code on the test problems. CNS (Could Not Solve) indicates that a feasible solution could not be obtained.

Problem	Route	Cost				Runtime			
	Size	Min	Med	Max	IQR	Min	Med	Max	IQR
tele5-1	10%	CNS							
	50%	CNS							
	100%	1082	1101	1101	19	0.13	4.06	4.47	1.3
tele5-2	10%	902	902	902	0	0.06	0.11	0.25	0.02
	50%	902	902	902	0	0.18	0.26	0.39	0.03
	100%	656	656	656	0	0.08	0.095	0.13	0.03
tele10-1	10%	CNS							
	50%	3991	4175.5	4272	188.5	150.45	292.54	528.14	182.63
	100%	2982	3040	3130	79.85	78.23	381.69	573.98	344.26
tele10-2	10%	CNS							
	50%	1898	2038	2285	58.5	89.71	279.83	597.85	243.21
	100%	1712	1817	2026	57.25	82.16	236.94	397.55	110.3
tele20-1	10%	13647	13647	13647	0	0.1	0.16	0.43	0.04
	50%	7254	7633.5	7777	432.25	236.11	582	599.9	50.5
	100%	7921	8395.5	9286	787.75	347.25	565.8	598.67	71.78
tele20-2	10%	16535	16535	16535	0	0.13	0.24	0.67	0.06
	50%	7913	8099.5	8810	211.5	407.24	555.97	598.69	53.18
	100%	8094	8970	9785	773	529.57	557.98	589.22	45.8
tele30-1	10%	8729	8745.5	8762	12.5	1133.1	1300.6	1742.38	332.02
	50%	7069	7342	7617	219.5	1475.36	1499.75	1517.23	12.65
	100%	7968	8309	8648	160.5	1547.32	1586.32	1591.23	17.06
tele30-2	10%	10196	10196	10196	0	1257.21	1258.32	1477.51	110.15
	50%	9188	9330	9665	127.5	1498.32	1524.32	1554.39	28.04
	100%	9633	9807	10232	201	1569.25	1578.36	1578.42	4.6

network with a short amount of computational time.

The next stage of our research will focus on developing a representation method in which routes are dynamically built rather than being statically selected. In addition, we wish to develop a tabu search implementation to solve this problem. We are also investigating alternative methods for allocating traffic to routes as this may provide improvements to overall solution quality.

Acknowledgments The Authors acknowledge the financial support of the Australian Research Council under Grant #A69601975 and the assistance of Benn Sugden who wrote the software that generates the problem instances.

References

D. Abramson and M. Randall, "A Simulated Annealing Code for General Integer Linear Programs," to appear in Annals of Operations Research, 1999.

A. Balakrishnan, T. Magnanti, A. Shulman and R. Wong, "Models for Planning Capacity Expansion in Local Access Telecommunication Networks," Annals of Operations Research, vol. 33, pp. 239-284, 1991.

L. Berry, B. Murtagh, S. Sugden, and G. McMahon, "Application of a Genetic-based Algorithm for Optimal Design of Tree-structured Communications Networks," Proceedings of the Regional Teletraffic Engineering Conference of the International Teletraffic Congress, South Africa, 1995, pp. 361-370.

L. Berry, B. Murtagh, S. Sugden, and G. McMahon, "An Integrated GA-LP Approach to Communication Network Design," Proceedings of the 2^{nd} IFIP Workshop on Traffic Management and Synthesis of ATM Networks, Canada, 1997.

N. Collins, R. Eglese and B. Golden, "Simulated Annealing: An Annotated Bibliography," American Journal of Mathematical and Management Sciences, vol. 8, pp. 209-307, 1988.

D. Connolly, "An Improved Annealing for the QAP," European Journal of Operational Research, vol. 46, pp. 93-100, 1990.

R. Eglese, "Simulated Annealing : A Tool for Operational Research," European Journal of Operational Research, vol. 46, pp. 271-281, 1990.

B. Gavish, "Augmented Lagrangian based Algorithms for Centralised Network Design", IEEE Transactions on Communications, vol. 33, pp. 1247-1257, 1985.

D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley: Reading, MA, 1989.

Kershenbaum, A. and Peng, S. "Neighbor finding Algorithms for CMST Customer Calculations," IEEE INFOCOM, Miami, Florida, 1986.

S. Kirkpatrick, D. Gelatt and M. Vecchi, "Optimization by Simulated Annealing", Science, vol. 220, pp. 671-680, 1983.

C. Koulamas, S. Antony and R. Jansen, "A Survey of Simulated Annealing to Operations Research Problems," Omega International Journal of Management Science, vol. 22, pp. 41-56, 1994.

M. Minoux, "Network Synthesis and Optimum Network Design Problem: Models, Solution Methods and Applications", Networks, vol. 19, pp. 313-360, 1989.

I. Osman and J. Kelly, Metaheuristics: Theory and Applications, Kluwer Academic Publishers: Norwell, MA, 1996.

C. Petersen, "Computational Experience of the Balas Algorithm Applied to a Selection of R&D Projects," Management Science, vol. 13, pp. 736-750, 1967.

U. Sharma, K. Mistra and A. Bhattacharji, “Applications of an Efficient Search Techniques for Optimal Design of Computer Communication Networks,” Micro Electronics and Reliability, vol. 31, pp. 337-341, 1991.

L. van Laarhoven, and E. Aarts, Simulated Annealing: Theory and Applications, D Reidel Publishing Company: Dordecht, 1987.