Global Optimization in Communication Networks

W. Y. Lin
G. L. Bilbro

Center for Communications and Signal Processing
Department of Electrical and Computer Engineering
North Carolina State University

TR-93/10
April 1993

Abstract

Simulated Annealing and a fast pipeline optimization algorithm are introduced. Their applications in communication networks are presented. Two networks optimization problems are solved by extending the pipeline algorithm. The experimental results are given.

1 Introduction

Simulated annealing is a probabilistic method which based on the physical process whereby a molten solid is slowly cooled so that it eventually "freezes" into a minimum energy configuration. Researchers have used SA extensively in the last decade with excellent results in many cases [9] [11] [6] . There are two kinds of problems which are very difficult and time consuming for traditional optimization methods: one is the continuous problems having multiple local minima (without loss of generality, we can consider optimization to be the search for minima of cost function f (or -f)), the other is NP hard combinatorial optimization problems. SA allows 'hill climbing' moves to climb out of local minima and ultimately terminates at the global minimum. The cooling schedule and the transition mechanism of SA are based on the observation that combinatorial optimization problems exhibit properties similar to physical processes with many degrees of freedom. Experimental results [3] [10] show that SA produces very good solutions for combinatorial optimization problems. SA provides an approach to both the above problems, at the expense, however, of large computation time.

Many problems belonging to the above two kinds of problems exist in communication networks. Computer/communication components are often discrete. For instance it is possible to lease communication lines with 2400-Bd or 4800-Bd bandwidths but not one in between. The problem of selecting optimal capacities for communication lines is therefore combinatorial . Multichain queueing networks model results in a cost function [1] with multiple minima. On the other hand, even in the convex problem which has unique optimum and can be solved by mathematical techniques, when the cost function cannot be computed by a closed form solution, the problem must be solved by iterations. But the iterative procedure diverges for poor initialization. Applying SA to this kind of problems can avoid the divergence problem and get the good result.

Starting with the recently introduced pipeline algorithm [2], we extend this strategy to solve constrained optimization of continuous variables as well as constrained combinatorial problems. We introduce our optimization algorithm in section 2 and give two examples of applying the algorithm to optimization problems in communication networks in section 3 and section 4.

2 The optimization algorithm

In SA, guaranteed convergence requires a cooling schedule of the form $T_k \geq C/ln(k+1)$, therefore requires a computational effort that is exponential in C to attain a specified temperature. The pipeline algorithm works much faster than traditional SA.

The algorithm considers an array of K samplers, each operating at its own fixed temperature T_k for $k = 1, 2, ..., K$ where

$$T_k = \frac{\Delta_f}{\frac{1}{3}k} \tag{1}$$

The number of samplers K in the array is calculated by the equation

$$K = \frac{\Delta_f}{\frac{1}{3}T_{fin}} \tag{2}$$

where

$$\Delta_f = \max f(x) - \min f(x) \tag{3}$$

$f(x)$ is the value of the cost function. T_{fin} is the last temperature which is low enough so that the corresponding Boltzmann distribution is dominated by solutions acceptably close to the global optimum, where " acceptable" depends on the problem.

In each time step, each unit in the pipeline array updates its current state twice: first with the state of the preceding sampler, then with a candidate generated uniformly in a small region around the current state by corresponding different acceptance probability. More details about the algorithm are presented in [2]

3 BCMP Queueing Networks Throughput Optimization

For BCMP [7] queueing networks with N stations containing single or infinite servers and **K** jobs, the Bard/Schweitzer [12] [8] algorithm provides the following solution
The mean response time of the i^{th} station ($i = 1, 2, ..., N$) is:

$$\overline{t_i} = \begin{cases} \frac{1}{\mu_i}(1 + \frac{K-1}{K}\overline{k_i}) & Type(i) \neq IS \\ \frac{1}{\mu_i} & Type(i) = IS \end{cases} \tag{4}$$

where μ_i is the service rate of the i^{th} station. The throughput of the network is obtained by Little's law:

$$\lambda = \frac{K}{\sum_{i=1}^{N} e_i \overline{t_i}} \tag{5}$$

The mean number of jobs at the i^{th} station is also obtained by Little's law:

$$\overline{k_i} = \lambda e_i \overline{t_i} \tag{6}$$

where e_i is the mean number of visits that a job makes to station i.

Our objective here is to find the optimal service rates $\mu_i{}^*$ which provide the maximum throughput λ^* within the available budget. The cost constraint of the network is calculated by a family of functions suggested by Kleinrock [5] of the form:

$$COST = \sum_i C_i \cdot \mu_i{}^{\beta_i} \tag{7}$$

This family of functions is extremely rich: it includes functions which are concave ($\beta \geq 1$) , convex ($\beta < 1$) , and neither concave nor convex (mixed $\beta \geq 1$ and $\beta < 1$).

The optimal service rates $\mu_i{}^*$ which determine the maximum throughput λ for a given cost constrain cannot be computed by a closed form solution. Akyildiz and Bolch used the method of Lagrange multipliers to solve the optimization problems [4] . They noted that the iteration converged for $0 < \beta < 2$ but not $\beta \geq 2$. They attributed failure to values for μ_i that were not initiated appropriately.

We use our algorithm to solve a problem given in [4] again. The new algorithm converges at any feasible starting point for any value of β.

At each evaluation of the algorithm (at each particular $\mu_i^{(n)}$), we start with $\overline{k_i} = \frac{K}{N}$ and iterate sequentially through (4) - (6) until convergence is observed at a fixed point of the $\overline{k_i}$ when the current $\overline{k_i}$ values given by equation (6) deviate less than $\epsilon = 10^{-4}$ from $\overline{k_i}$ values of the previous iteration.

In Table 1, we show our results compared with the results of Akyildiz and Bolch (λ). Our results are negligibly smaller than theirs due to our coarse move set. If necessary, their method of Lagrange multipliers will converge for any value of β, if it starts at our results.

β	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.5	3.0	4.0	6.0
μ_1^*	37.10	12.62	7.20	5.07	4.00	3.34	2.91	2.39	2.08	1.75	1.46
μ_2^*	7.73	2.69	1.59	1.11	0.89	0.79	0.73	0.53	0.70	0.57	0.73
μ_3^*	18.90	6.51	3.79	2.69	2.09	1.71	1.52	1.33	1.16	1.01	0.85
μ_4^*	1.16	0.60	0.49	0.39	0.36	0.84	0.25	0.45	0.61	0.33	0.27
μ_5^*	1.67	0.73	0.29	0.52	0.35	0.19	0.83	0.23	0.66	0.79	0.72
λ^*	35.954	12.379	7.096	5.016	3.959	3.297	2.891	2.377	2.071	1.745	1.458
λ	36.007	12.395	7.109	5.028	3.960	3.322					

Table 1: ($K = 70, COST = 100$, the missing cells indicate failure of the Akyildiz and Bolch's algorithm)

4 Selecting Bandwidths for Communication Lines in Networks with Window Flow Control

Most packet networks nowaday provide virtual channels that are end to end flow controlled. The maximum number of packets that can be in transit within the virtual channel is referred to as the virtual channel window size. We consider the modeling of flow controlled virtual channels as multiple closed chains. In the queueing network model considered, FCFS servers are used to model communication channels and IS servers are used to model random delays associated with acknowledgements and timeouts. The Bard/Schweitzer [12] [8] algorithm provides the following solution for the analysis of such networks.

Assume that there are N servers and R different (routing) chains in the network. The mean delay of a packet in chain r at server i (for $i = 1, 2..., N$ and $r = 1, 2..., R$) is

$$\overline{t_i} = \begin{cases} \frac{1}{\alpha_{ir}\mu_i}(1 + \frac{K_r - 1}{K_r}\overline{k_i}) + \sum_{s \neq r}^{R} \overline{k_{is}} \frac{1}{\alpha_{is}\mu_i} & \text{Server}(i) \neq IS \\ \frac{1}{\alpha_{ir}\mu_i} & \text{Server}(i) = IS \end{cases} \tag{8}$$

where μ_i is the nominal service rate of the i^{th} server. For communication channels, it equals the bandwidth. For acknowledgements and timeouts delays, it is a constant. α_{ir} characterize the relative service rates for different message lengths in virtual channels, it is also a constant for IS servers. K_r is the window size of chain r.

The throughput of the network for chain r is obtained by Little's law:

$$\lambda_r = \frac{K_r}{\sum_{i=1}^{N} e_{ir}\overline{t_{ir}}} \tag{9}$$

The mean number of chain r packets at the i^{th} server is also obtained by Little's law:

$$\overline{k_{ir}} = \lambda_r e_{ir}\overline{t_{ir}} \tag{10}$$

where e_{ir} is the mean number of visits that a packet of chain r makes to server i.

The solution is similar to equation (4) – (6) in section 3 with two differences. All jobs in section 3 are in the same class, but here there are different packet classes. The service rates in section 3 are real and continuous, but here the bandwidths can only be selected from certain values. The first difference results in multiple local maxima while the second difference changes the continuous problem to a discrete problem. It is hard to use any mathematical programming methods to solve the problem. Our algorithm is better suited to this kind of problem.

Figure 1 shows the topology of a communication network model which we will analyze. The queueing model is shown in Figure 2. Servers 1-6 represent FCFS queues and 7-11 represent IS servers. We also admit the rich class of cost constraint function introduced in section 3 (equation (7)).

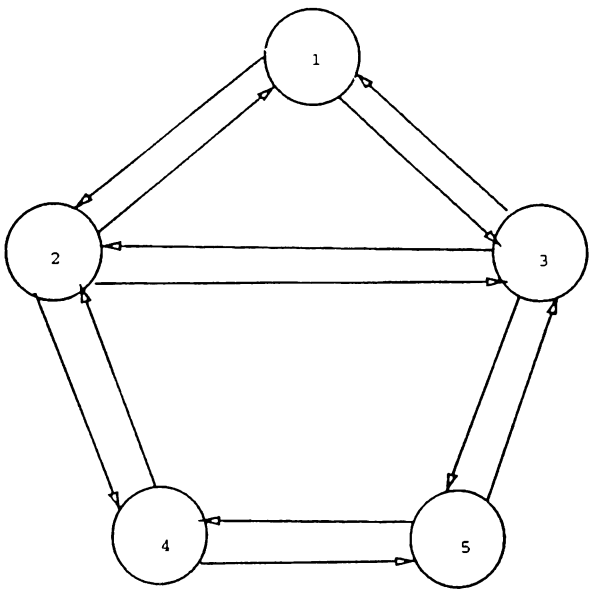

Figure 1: Communication Network

Cost rates (C_i), routing chains and service rate factors are specified in Tables 2, 3 and 4 as input parameters. We select μ_i from ten different bandwidths: $n \cdot 64kbps, n = 1, 2..., 10$. In table 5 we just show the values of n as our results. We set window size of each chain with vector $\underline{K} = (4, 4, 4, 4, 4, 6, 2)$ and constrain the budget cost as 2700.

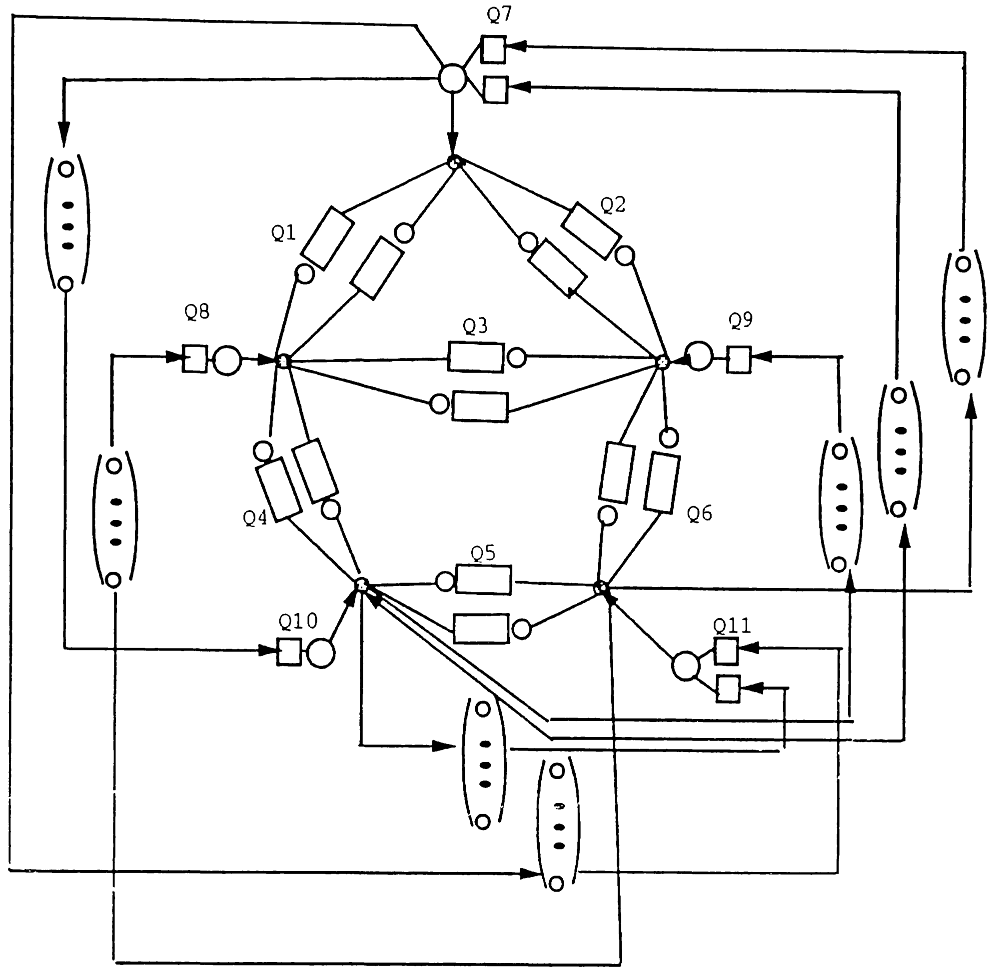

Figure 2: Queueing Model

server	rate C_i	server type
1	250	$\neq IS$
2	125	$\neq IS$
3	125	$\neq IS$
4	50	$\neq IS$
5	50	$\neq IS$
6	100	$\neq IS$
7	110	$= IS$
8	110	$= IS$
9	110	$= IS$
10	110	$= IS$
11	110	$= IS$

Table 2: Cost Rates

e_{ir}							
	chain						
server	1	2	3	4	5	6	7
1	1.0	0.0	0.0	0.0	1.0	1.0	0.0
2	0.0	1.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	1.0	1.0	0.0	1.0	0.0
4	1.0	0.0	0.0	1.0	1.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	1.0
6	0.0	1.0	1.0	0.0	0.0	1.0	0.0
7	1.0	1.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	1.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	1.0	0.0	0.0	0.0
10	0.0	1.0	0.0	0.0	1.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	1.0	1.0

Table 3: Visit Ratios

α_{ir}							
	chain						
server	1	2	3	4	5	6	7
1	5000	6000	7000	8000	9000	10000	11000
2	5000	6000	7000	8000	9000	10000	11000
3	5000	6000	7000	8000	9000	10000	11000
4	5000	6000	7000	8000	9000	10000	11000
5	5000	6000	7000	8000	9000	10000	11000
6	5000	6000	7000	8000	9000	10000	11000
7	10	10	10	10	10	10	10
8	10	10	10	10	10	10	10
9	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
11	10	10	10	10	10	10	10

Table 4: Service Rate Factors = message lengths (bits)

β	0.5	1.0	1.25	1.75	2.0	2.5	3.0
$\mu_1{}^*$	10	1	1	1	1	1	1
$\mu_2{}^*$	10	1	1	1	2	1	1
$\mu_3{}^*$	1	6	3	2	1	1	1
$\mu_4{}^*$	10	4	2	2	1	2	2
$\mu_5{}^*$	10	10	10	5	4	3	2
$\mu_6{}^*$	10	3	2	2	2	2	2
λ^*	223.4	110.9	89.3	56.9	47.5	40.2	34.4

Table 5: Optimal Throughput and Optimal Service Rates

5 Conclusion

A fast pipeline optimization algorithm has been extended to treat constraints in optimization problems and has been applied to two communication networks problems. Experimental results show that the new algorithm is more robust than other techniques. We suggest that the new algorithm is useful for solving realistic optimization problems in communication networks.

Acknowledgement
The authors thank Dr. A. Nilsson for helpful discussions and guidance.

References

[1] Ian F. Akyildiz and Ron Shonkwiler. Simulated annealing for throughput optimization in communication networks with window flow control. Georgia Institute of Techology.

[2] Griff Bilbro. A new optimization strategy. Technical Report TR-92/10, CCSP, Department of Electrical and Computer Engineering, NCSU, 1992.

[3] C.D. Gelatt, S. Kirkpatrick, and M. P. Vecch. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[4] I.F.Akyildiz and G. Bolch. Throughput and response time optimization in queueing network models of computer systems. In *Proc. of Int. Seminar on Distributed and Parallel Systems*, North-Holland, December 1988.

[5] L. Kleinrock. Analytic and simulation methods in computer network design. In *AFIPS conf. Proc.*, June 1969.

[6] M.Lundy. Applications of the annealing algorithm to combinatorial problems in statistics. *Biometrika*, 72:191–198, 1985.

[7] R.R.Muntz, F.Baskett, K. M. Chandy, and F.G.Palacios. Open, closed and mixed queues of networks with different types of customers. *Journal of the ACM*, 22:248–260, April 1975.

[8] P.J. Schweitzer. Approximate analysis of multiclass closed networks of queues. In *Int. Conference Stochastic Control and Optimization*, Amsterdam, 1979.

[9] C. Sechen, F. Romeo, and A. Sangiovanni-Vincentelli. Simulated annealing research at berkeley. In *Proc. IEEE Int. Conf. Computer Design*, pages 652–657, 1984.

[10] C. Sechen and A. Sangiovanni-Vincentelli. The TimberWolf placement and routing package. In *Proc. 1984 Custom Intergrated Circuit Conference*, Rochester, 1984.

[11] M. P. Vecchi and S. Kirkpatrick. Global wiring by simulated annealing. *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst.*, CAD-2:215–222, Oct. 1983.

[12] Y.Bard. Some extension to multiclass queueing network analysis. In *Proc. of Performance 79*, Vienna, Feb. 1979.