Optimizing the throughput, service rate, and buffer allocation in finite queueing networks

F. R. B. Cruz [1,2]

Department of Statistics, Federal University of Minas Gerais,
31270-901 - Belo Horizonte - MG, Brazil.

Abstract

We examine the problem of maximizing the throughput of an acyclic network of general-service time queueing network while reducing the total number of buffers and the overall service rate. These are conflicting objectives and we utilize an original multi-objective genetic algorithm to tackle this problem. Promising preliminaries results show the efficacy of the approach.

Keywords: Operations research, genetic algorithms, queueing networks, buffer allocation, service allocation.

1 Introduction

The problem of maximizing Θ, the throughput (that is, the number of jobs, parts, clients, and so on, served per unit of time), in an acyclic network (for an example, see Fig. 1) of general-service time queueing network is examined here. The problem may be stated simply as to find the minimum number of buffers, $\mathbf{K} = \{K_1, K_2, \ldots, K_n\}$, and service rates, $\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_n\}$, that

[1] This research has been partially funded by CNPq, CAPES, and FAPEMIG

[2] Email: `fcruz@est.ufmg.br`

must be allocated to a queueing network in a given topology and for a given external arrival rate, $\mathbf{\Lambda} = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_n\}$, in order to provide maximum Θ.

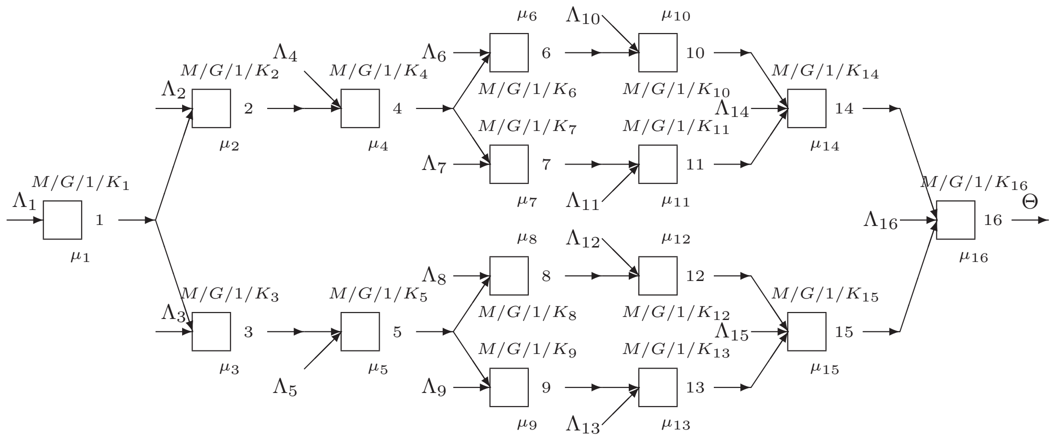


Fig. 1. A complex network [8].

From a modeling point of view, the throughput maximization problem can be defined by a mixed-integer mathematical programing formulation in which the total buffer and server costs are minimized and the throughput is maximized subject to an integer buffer allocation and a non-negative service rate. Defining a queueing network as a digraph $G(N, A)$, where N is a finite set of nodes and A is a finite set of arcs, the mixed-integer mathematical programming formulation follows

$$\text{minimize}\, F(\mathbf{K}, \boldsymbol{\mu}) = \Big(f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), -f_3(\mathbf{K}, \boldsymbol{\mu})\Big)^{\mathrm{T}}, \tag{1}$$

subject to

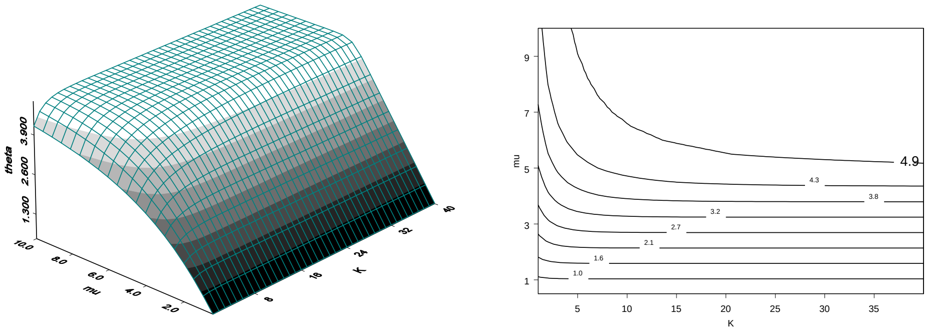
$$K_i \in \{1, 2, \ldots\},\ \forall i \in N, \tag{2}$$

$$\mu_i \geq 0,\ \forall i \in N, \tag{3}$$

in which the decision variables K_i and μ_i are, respectively, the total capacity *including* those in service and the service rate, for the ith $M/G/1/K$ queue. The objective functions are the total buffer allocation, $f_1(\mathbf{K}) = \sum_{\forall i \in N} K_i$, the overall service allocation, $f_2(\boldsymbol{\mu}) = \sum_{\forall i \in N} \mu_i$, and, finally, the overall throughput, $f_3(\mathbf{K}, \boldsymbol{\mu}) = \Theta(\mathbf{K}, \boldsymbol{\mu})$. These three objectives are in conflict because buffer and service allocation are expensive but low buffer sizes and low service rates usually lead to low throughput and consequently less profit [6].

Fig. 2 shows a typical surface for the throughput in a single finite queue, as a function of the buffer size and the service rate, and the respective contour plot. In a network of queues, one will observe a similar behavior, as we will show shortly. It is worthwhile mentioning that although the surface seen in

Fig. 2 is smooth and convexity seems to hold in this case, as it holds for simpler queueing networks studied in the past (see, for instance, [6]), for optimization purposes, the flatness in the top of the surface, around which is the maximum throughput, creates difficulties for traditional methods. For instance, multiple starts had to be used for the Powell method to avoid premature convergence and to derive a successful optimization algorithm for buffer allocation in single server general-service time queueing networks [8].



(a) Θ *versus* service rate and buffer size

(b) contour plot

Fig. 2. Results for a single $M/G/1/K$ queue for $\Lambda = 5.0$

In this paper, we propose a different approach and determine an approximation for the whole Pareto set, which is the set of optimal solutions for more than one objective in the objective functions. We use a multi-objective genetic algorithm (MOGA) approach in combination with the generalized expansion method (GEM), a well-known method to obtain accurately approximations for performance measures of a queueing network [2].

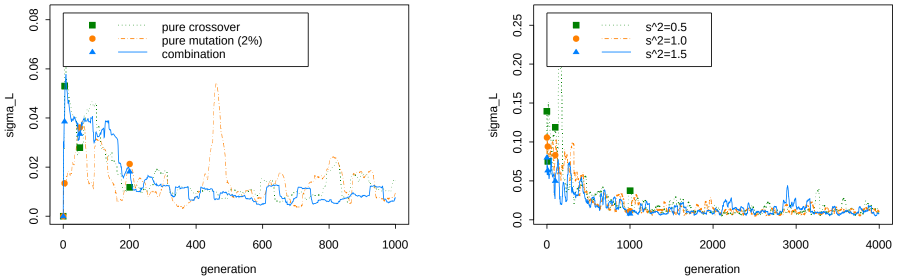
2 Proposed Algorithm

In order to solve the throughput maximization problem, one needs to have a good estimate for $\Theta(\mathbf{K}, \boldsymbol{\mu})$. In a *single* $M/G/1/K$ queue, such a problem is well solved by means of a computationally efficient and accurate closed-form approximate expression for the blocking probability p_K, proposed by Smith [7]. For a *network* of queues, an algorithm that is available is the GEM [2]. Firstly, a pre-evaluation is performed. An arbitrary node j is chosen from set V (initially, $V = N$) but not from set Q (in which Q is the set of nodes already evaluated, initially, $Q = \emptyset$), such that for all arc $(i, j) \in A$, vertex i has been evaluated already. Then, vertex j has computed its blocking probability $p_K^{(j)}$ and its arrival rate, from $\theta_j = \lambda_j \times (1 - p_K^{(j)})$. These service rates are then forwarded as arrival rates to the downstream nodes (if they exist), and vertex j is included in set Q. Note that the pre-evaluation step is a variant of Dijkstra's minimum path algorithm [5]. The GEM includes also an evaluation step, which seeks flow conservation, that is $\theta_j \leq \lambda_j + \sum_{\forall\, i|(i,j)\in A} \theta_i p_{ij},\ \forall j \in V$. The evaluation algorithm is a Dijkstra's labeling algorithm working in reverse. Notice that the performance evaluation algorithm must have available the routing probabilities p_{ij} before it can compute all the performance measures.

For the multi-objective throughput maximization problem under consideration, a multi-objective genetic algorithm (MOGA) appear to be suitable a choice. The efficiency of MOGAs is well established for dealing with multi-objective problems [1]. MOGAs are optimization algorithms to perform an approximate global search relying on the information obtained from the evaluation of several points in the search space and obtaining a population of these points that converges to the optimum through the application of the genetic operators *mutation*, *crossover*, *selection*, and *elitism*. The selection and elitism operators used were the standard for the NSGA-II version [4]. For crossover, we choose a mechanism known as *uniform*, popular for multivariable coding [9]. With regards to the mutation scheme, it happens with probability `rateMut`, for each one of the genes of the individuals (the decision variables K_i and μ_i). As suggested by Deb & Agrawal [3], Gaussian perturbations are added to the decision variables. Notice that after crossover and mutation, constraints (2) and (3) may no longer hold. In order to guarantee feasibility, the values are accordingly rounded, for the integer variables, and readjusted, by means of the reflection operators, $K_{i,r} = 1 + |K_i - 1|$ and $\mu_{i,r} = \mu_{\text{lowlim}_i} + |\mu_i - \mu_{\text{lowlim}_i}|$, in which 1 is the lower limit for buffer allocation, μ_{lowlim_i} is the lower limit for service allocation (in order to make sure that $\rho < 1$ will hold), K_i and μ_i are the resulting values after crossover and mutation, and $K_{i,r}$ $\mu_{i,r}$ are the result after reflection, being $|x|$ the absolute value of x. The above scheme warrants only feasible solutions without avoiding or privileging any particular solution.

3 Computational Results and Discussion

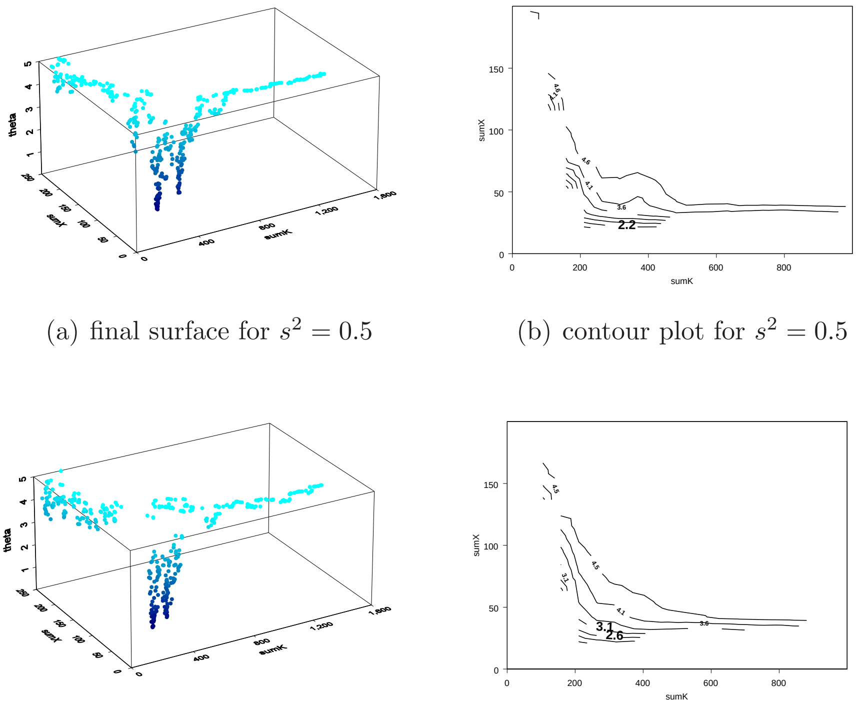
In order to make use of a GEM implementation previously developed in FORTRAN [8], the optimization algorithm was implemented in the same language. The code is available from the authors upon request. The complex network from Fig. 1, previously presented in the literature, was analyzed. For an arrival rate $\Lambda_1 = 5.0$, three different squared coefficient of variations were analyzed $s^2 = \{0.5, 1.0, 1.5\}$. The maximum number of generations was fixed in 4,000. Firstly, we can infer from Fig. 3 that a combined use of crossover and mutation is effective in speeding-up the convergence and that the convergence, measured in terms of maximal crowding distance [4], is independent on the squared coefficient of variation.



(a) crossover and mutation effect (b) squared coefficient of variation effect

Fig. 3. Convergence for the 16-node network

The profile can be seen in Fig. 4, which present the final surfaces and respective contour plots. For comparison purposes, an exact contour plot for a single-node queue is presented in Fig. 2-(b). The resemblance is very encouraging.



(a) final surface for $s^2 = 0.5$

(b) contour plot for $s^2 = 0.5$

(c) final surface for $s^2 = 1.5$

(d) contour plot for $s^2 = 1.5$

Fig. 4. Final results for the 16-node complex network

4 Conclusions and Final Remarks

In this paper, we briefly described results for a multi-objective approach for the throughput maximization problem for finite single server general queueing networks. Combining the generalized expansion method, as the performance evaluation tool, with a multi-objective genetic algorithm may disclose insightful Pareto curves. These curves explicitly show the trade-off between throughput, total buffer allocation, and overall service allocation. Open questions include how well this methodology would apply to slightly different trade-off problems in finite queueing networks.

References

[1] Coello, C. A. C., *An updated survey of GA-based multiobjective optimization techniques*, in: *Proceedings of the ACM Computing Surveys* **32** (2000), pp. 109–143.

[2] Cruz, F. R. B. and J. M. Smith, *Approximate analysis of $M/G/c/c$ state-dependent queueing networks*, Computers & Operations Research **34** (2007), pp. 2332–2344.

[3] Deb, K. and R. B. Agrawal, *Simulated binary crossover for continuous search space*, Complex Systems **9** (1995), pp. 115–148.

[4] Deb, K., A. Pratap, S. Agarwal and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation **6** (2002), pp. 182–197.

[5] Dijkstra, E. W., *A note on two problems in connection with graphs*, Numerical Mathematics **1** (1959), pp. 269–271.

[6] Meester, L. E. and J. G. Shanthikumar, *Concavity of the throughput of tandem queueing systems with finite buffer storage space*, Advances in Applied Probability **22** (1990), pp. 764–767.

[7] Smith, J. M., *Optimal design and performance modelling of $M/G/1/K$ queueing systems*, Mathematical and Computer Modelling **39** (2004), pp. 1049–1081.

[8] Smith, J. M. and F. R. B. Cruz, *The buffer allocation problem for general finite buffer queueing networks*, IIE Transactions **37** (2005), pp. 343–365.

[9] Sywerda, G., *Uniform crossover in genetic algorithms*, in: *Proceedings of the third international conference on Genetic algorithms* (1989), pp. 2–9.