Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds.

SENSITIVITY ANALYSIS OF LINEAR PROGRAMMING FORMULATIONS FOR G/G/M QUEUE

Wai Kin (Victor) Chan
Nowell Closser

Department of Industrial and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180, USA

ABSTRACT

Linear programming representations for discrete-event simulation provide an alternative approach for analyzing discrete-event simulations. This paper presents several formulations for $G/G/m$ queues and discusses the applications and limitations of these formulations. We derive the relationship between these formulations. We then demonstrate the applications of these formulations in sample-path gradient estimation.

1 INTRODUCTION

Mathematical programming representations (MPRs) for discrete-event dynamic systems (DEDSs) are mathematical programs modeling the sample paths of DEDSs (Chan and Schruben 2008). These representations provide a new means of analyzing DEDSs using mathematical programming techniques. Using the MPRs, (Chan and Schruben 2006) demonstrates the use of dual variables to compute infinitesimal perturbation analysis (IPA) estimators for $G/G/1$ queues. This paper extends the work in (Chan and Schruben 2006) to MPRs for $G/G/m$ queue. In particular, MPRs for $G/G/2$ queues are used to illustrate the relationship between these MPRs and the meaning of the dual variables. IPA estimators based on the dual variables are also developed.

IPA is a gradient estimation approach based on differentiating the sample path of a discrete-event simulation. It is computationally efficient and relatively easy to implement. Consistency property of IPA estimators has been proved for certain queueing systems. For example, (Suri and Zazanis 1988) give a strong consistency proof of IPA for $M/G/1$ queue. (Zazanis and Suri 1994) extends the consistency proof to the $GI/G/1$ queue under certain assumptions. (Fu and Hu 1991) provide the proof for the $GI/G/m$ queue. IPA algorithms have also been developed for generalized semi-Markov processes (GSMP) (see e.g., (Glasserman 1991)).

However, IPA can fail (e.g., biased) when the estimation involves discontinuities in the performance measure (Suri 1989). When IPA fails, various generalizations or alternatives of perturbation analysis techniques have been developed. One example is the smoothed perturbation analysis (SPA), which uses conditional probability to derive gradient estimators (Gong and Ho 1987). See (Fu and Hu 1997) for a detailed discussion and comparison of various extensions of perturbation analysis techniques. (Homem-de-Mello, Shapiro, and Spearman 1999) also makes use of max-plus algebra to obtain sample path gradient for production scheduling problems under continuous distributions.

One advantage of having MPRs is that perturbation analysis of DEDSs could be carried out by using sensitivity analysis of mathematical programming, where a rich theory and tools already exists. For example, (Gal 1979; Gal and Greenberg 1997) provide an extensive discussion on postoptimal analysis of

mathematical programming models. (Ward and Wendell 1990) review different approaches of sensitivity analysis in linear programming. Being the mathematical programming models for DEDSs, the MPRs could be considered as a bridge that allows these mathematical programming sensitivity analysis methods to be applied to perturbation analysis of DEDSs.

In this paper, we first review the MPR of the *G*/*G*/1 queue in Section 2. In Section 3, we develop a new MPR for the *G*/*G*/2 queue, which can be extended to the *G*/*G*/*m* queue. We show that this MPR can be decomposed into two separate linear programs, each representing the sample path of one server. Section 4 presents experimental results. Section 5 offers a conclusion.

2 BACKGROUND -- LINEAR PROGRAMMING REPRESENTATIONS OF G/G/1 QUEUE

We first review the *G*/*G*/1 queue linear programming formulation (Chan and Schruben 2008). We will also introduce the dual LP and its associated results.

Consider a discrete-event simulation model for a *G*/*G*/1 queue with *n* jobs to be processed. The linear programming formulation (or mathematical programming representation, i.e., MPR) for this simulation model is given in the following:

GG1-LP(*F*):

$$\begin{aligned} \min & \sum_{i=1}^{n} F_i \\ \text{s.t.} \quad & F_i - A_i \geq s_i, \qquad i = 1, \ldots, n \quad (U_i) \\ & F_i - F_{i-1} \geq s_i, \qquad i = 2, \ldots, n \quad (V_i) \\ & F_i \text{ free } \forall i \end{aligned}$$

where F_i is the finish time of the i^{th} customer in the sample path, U_i and V_j are the dual variables of the corresponding constraints, s_i is the service time of the i^{th} customer, and $A_i = a_1 + a_2 + \ldots + a_i$ is the arrival time of the i^{th} customer to the system with a_i being the inter-arrival time between the i–1 and i^{th} arrivals, i=1,…,n, j=2,…,n. There is no explicit restriction on the sign of F_i's since the constraints require that they are positive. The dual model is:

GG1-LP-Dual(*F*):

$$\begin{aligned} \max & \sum_{i=2}^{n} (A_i + s_i) U_i + \sum_{i=2}^{n} V_i \\ \text{s.t.} \quad & U_1 - V_2 = 1 \qquad\qquad\qquad\qquad\qquad\qquad (F_1) \\ & U_i + V_i - V_{i+1} = 1, \qquad i = 2, \ldots, n-1 \quad (F_i) \\ & U_n + V_n = 1, \qquad\qquad i = 2, \ldots, n-1 \quad (F_i) \\ & U_i, V_i \geq 0 \quad \forall i \end{aligned}$$

One note about these two formulations is that the primal is in the time domain while the dual is in the number domain. Using an induction argument, one can show that in the optimal solution the dual variables U_1 or $U_i + V_i$ (i = 2,…, n) are equal to the number of customers in the busy period seen by the i^{th} departing customer (Chan 2005).

It should be noted that, given the realizations of the input random variables, the optimal solution of GG1-LP(*F*) (or GG1-LP-Dual(*F*)) will be identical to the sample path of the corresponding simulation executed using the same set of random variable realizations. This optimal solution should *not* be mis-

understood as the "optimal processing schedule" for the customers, as the service discipline is always FCFS and services start as soon as feasible.

Example 1. Solving the GG1-LP(F) for $n = 5$, $(A_1, A_2, \ldots, A_5) = (0, 2, 4, 6, 8)$, and $(s_1, s_2, \ldots, s_5) = (3, 2, 5, 6, 3)$ gives the optimal solution: $(F_1, F_2, \ldots, F_5) = (3, 5, 10, 16, 19)$, $(U_1, U_2, \ldots, U_5) = (5, 0, 0, 0, 0)$ and $(V_2, V_3, \ldots, V_5) = (4, 3, 2, 1)$. One can verify this sample path by a manual calculation. It can be easily seen that there is only one busy period, and therefore, the number of customers seen by the i^{th} departing customer matches the dual variables U_1 and $U_i + V_i$. ■

Different MPRs can be developed for the same DEDS, just as different simulation models can be created to model the same system. (Chan 2005) gives several different MPRs for $G/G/1$ queue. These MPRs look distinct as they are based on different variables, such as finish times, start times, and waiting times. However, these MPRs are equivalent in the sense that they represent the same dynamics of the same system—a $G/G/1$ queue. The same idea applies to the *G/G/m* queue. In the next section, we present a formulation for the $G/G/2$ queue. We then modify this LP into another LP, which will finally be decomposed into two $G/G/1$ LPs.

3 LINEAR PROGRAMMING FORMULATIONS FOR G/G/*M* QUEUE

We now introduce an LP formulation for $G/G/m$ queues. For ease of exposition, we will focus on the $G/G/2$ queue, but the LP developed can be extended to model the $G/G/m$ queue (see detail in (Chan 2010)).

FIFO does not hold in general in a *G/G/*2 queue*,* although the service discipline is still FCFS. For example, the first-arriving customer may not be the first customer to leave if its processing time is sufficiently long to outlast the second-arriving customer's service and departure from the other server. This phenomenon is called "overtaking." To handle overtaking, the multiple-server LP will need additional variables and constraints. This makes it more complicated and harder to analyze than the single-server LP.

In particular, for a two-server LP, two additional sets of variables are needed: α_i and β_i. In particular, α_1 is used to store $\max\{F_1, F_2\}$ and β_1 to store $\min\{F_1, F_2\}$, and subsequently α_i to store $\max\{\alpha_{i-1}, F_{i+1}\}$ and β_i to store $\min\{\alpha_{i-1}, F_{i+1}\}$, $i = 2,\ldots,n-1$. The idea is to use these two variables to figure out which server is available first when a new customer arrives. For example, the 3rd customer will begin processing at the lesser of its arrival time and $\beta_1 = \min\{F_1, F_2\}$. The value stored in α_1, representing the later finishing customer of the 1st and 2nd, will be passed on to β_2, which will then evaluate whether this later finishing customer will finish before the 3rd customer. This process then repeats until the last customer departs the system.

We adopt the notation used in (Chan 2005) and denote the following LP for multiple-server queues as $G/G/m$-LP7(F), as it is the 7th formulation for multiple-server queues proposed in (Chan 2005).

GG2-LP7(F):

$$
\begin{aligned}
\min \sum_{i=1}^{n} F_i + \sum_{i=1}^{n-1} \alpha_i - \sum_{i=1}^{n-1} \beta_i & & & \\
\text{s.t.} \quad F_i & \geq A_i + s_i, & i = 1, \ldots, n & \quad (U_i) \\
F_i - \beta_{i-2} & \geq s_i, & i = 3, \ldots, n & \quad (V_i) \\
-\beta_i + F_{i+1} & \geq 0, & i = 1, \ldots, n-1 & \quad (X_i^{\beta F}) \\
-\beta_i + \alpha_{i-1} & \geq 0, & i = 1, \ldots, n-1 & \quad (X_i^{\beta \alpha}) \\
\alpha_i - F_{i+1} & \geq 0, & i = 1, \ldots, n-1 & \quad (X_i^{\alpha F}) \\
\alpha_i - \alpha_{i-1} & \geq 0, & i = 1, \ldots, n-1 & \quad (X_i^{\alpha \alpha}) \\
F_i, \alpha_i, \beta_i & \geq 0 \quad \forall i & &
\end{aligned}
$$

with $\alpha_0 = F_1$ and $X_i^{\beta F}, X_i^{\beta \alpha}, X_i^{\alpha F}, X_i^{\alpha \alpha}$ are the corresponding dual variables.

Unfortunately, it is pointed out in (Chan 2005) that the optimal solution of this LP is not necessarily identical to the simulation sample path. To see that, one can examine the interaction between the variables. The objective function above provides incentive to minimize all F_i's and α_i's and maximize all β_i's. The hope is to push α_i down (minimize) to $\max\{\alpha_{i-1}, F_{i+1}\}$ as modeled by the 5th and 6th constraints and pull β_i up (maximize) to $\min\{\alpha_{i-1}, F_{i+1}\}$ as governed by the 3rd and 4th constraints. However, the objective of pushing α_i down interferes with the objective of pulling β_i up. One can see, for instance, that the same objective value is obtained in the case where $\alpha_i = \max\{\alpha_{i-1}, F_{i+1}\} + 1$ and $\beta_i = \min\{\alpha_{i-1}, F_{i+1}\} - 1$ and the case where $\alpha_i = \max\{\alpha_{i-1}, F_{i+1}\}$ and $\beta_i = \min\{\alpha_{i-1}, F_{i+1}\}$. Therefore, the optimal solution is not guaranteed to be the same as the sample path. In essence, the above LP yields multiple optimal solutions, of which we are only interested in one.

(Chan 2005) gives several solutions to this problem, including a sample path approach and a disjunctive constraint approach. Here, we develop a new formulation that can circumvent this problem. The idea is to combine the 1st and 2nd constraints by using a max operator, the 3rd and 4th constraints by a min operator, and the 5th and 6th constraints by a max operator. With the introduction of the min and max operators, all the inequality constraints are converted into equality. The objective function can also be simplified by removing the sum of α_i's and β_i's. In fact, the objective function no longer plays a part in the primal solution. We retain the format below for the purpose of giving particular meaning to the dual variables. This results in the following formulation that we call GG2-LP10(*F*) to continue the numbering of formulations used in (Chan 2005).

GG2-LP10(*F*):

$$
\begin{aligned}
\min \sum_{i=1}^{n} F_i & & & \\
\text{s.t.} \quad F_i & = \max\{A_i + s_i, \beta_{i-2} + s_i\} & i = 1, \ldots, n & \quad (U_i) \\
\beta_i & = \min\{\alpha_{i-1}, F_{i+1}\} & i = 1, \ldots, n-1 & \quad (X_i^{\beta}) \\
\alpha_i & = \max\{\alpha_{i-1}, F_{i+1}\} & i = 1, \ldots, n-1 & \quad (X_i^{\alpha}) \\
F_i, \alpha_i, \beta_i & \geq 0 \quad \forall i & &
\end{aligned}
$$

with $\beta_{-2} = 0$, $\beta_{-1} = 0$, $\alpha_0 = F_1$, and X_i^{β} and X_i^{α} are the corresponding dual variables.

Strictly speaking, as this formulation has max and min operators in the constraints, it is not a linear program. However, the fact that all constraints are equality constraints essentially reduces the formulation into two linear programs that can be quickly solved by pre-solvers of optimization packages.

To see this, one can follow an inductive argument (see the formal proof in Proposition 1). Starting with β_1 and α_1, pre-solver can find their values from $\min\{F_1, F_2\}$ and $\max\{F_1, F_2\}$, respectively. $F_3 = \max\{A_3+s_3, \beta_1+s_3\}$ can then be readily calculated. This enables the next iteration of computation for $\beta_2 = \min\{\alpha_1, F_3\}$, $\alpha_2 = \max\{\alpha_1, F_3\}$, and $F_4 = \max\{A_4+s_4, \beta_2+s_4\}$. This may be repeated until F_n is computed.

This statement is formally presented in the following proposition.

Proposition 1. Given a set of input random numbers (A_1, …, A_n) and (s_1,…, s_n), the linear programming representation, GG2-LP10(F), can be pre-solved, decomposed, and transformed into two separate GG1-LP(F)'s. The optimal solutions of these two LPs are also optimal for the original GG2-LP10(F).

The proof is given in the Appendix. Because the resulting LPs have the same format as the regular GG1-LP(F) introduced in Section 2, we have the following corollary.

Corollary 1. Both decomposed and transformed GG1-LP(F)'s share the same duality property as the regular GG1-LP(F).

In essence, the pre-solve procedure is equivalent to computing a set of recursive min and max equations. Therefore, this LP can be pre-solved efficiently. Fortunately, pre-solving the LP does not eliminate the LP's applications. Indeed, there are several such applications. For the purposes of this paper, we shall focus on getting the shadow prices for sensitivity analysis. We use an example to illustrate this application.

Example 2. Consider a $G/G/2$ queue simulation model that simulates up to $n = 10$ customers using the sample path data ($A_1, A_2, \ldots, A_{10}$) = (0, 2, 4, 6, 8, 10, 12, 20, 21, 22), and ($s_1, s_2, \ldots, s_{10}$) = (3, 2, 5, 6, 3, 3, 2, 5, 6, 3). Plugging this data into GG2-LP10(F) and solving gives the optimal solution: ($F_1, F_2, \ldots, F_{10}$) = (3, 4, 9, 12, 13, 15, 15, 25, 27, 28), ($U_1, U_2, \ldots, U_{10}$) = (1, 1, 3, 2, 2, 1, 1, 2, 1, 1), ($X_1^{\beta}, X_2^{\beta}, \text{K}, X_9^{\beta}$) = (0, 0, 2, 1, 1, 0, 0, 1, 0), and ($X_1^{\alpha}, X_2^{\alpha}, \text{K}, X_9^{\alpha}$) = (0, 2, 1, 1, 0, 0, 1, 0, 0). The two decomposed LPs are:

GG1-LP(F,1):

$$\begin{aligned} &\min F_1 + F_3 + F_5 + F_7 + F_8 + F_10 \\ \text{s.t.}\quad & F_i - A_i \geq s_i \quad i = 1,3,5,7,8,10 && (U_i) \\ & F_i - F_j \geq s_i \quad (i,j) \in \{(3,1),(5,3),(7,5),(8,7),(10,8)\} && (V_i) \\ & F_i \text{ free } \forall i \end{aligned}$$

GG1-LP(F,2):

$$\begin{aligned} &\min F_2 + F_4 + F_6 + F_9 \\ \text{s.t.}\quad & F_i - A_i \geq s_i \quad i = 2,4,6,9 && (U_i) \\ & F_i - F_j \geq s_i \quad (i,j) \in \{(4,2),(6,4),(9,6)\} && (V_i) \\ & F_i \text{ free } \forall i \end{aligned}$$

After pre-solving the LP, all the constraints should contain at most two variables, one with a positive sign and one with a negative sign when both variables are moved to the left hand side of the corresponding constraint. The dual of this formulation is therefore a network flow LP. We plot the network dynamics in Figure 1. The first horizontal line shows the arrival times of customers. The second and third horizontal lines represent the service time dynamics of the first and second servers, respectively. Both servers have experienced three busy periods in this example. A graphical representation of this network is depicted in Figure 2. The dotted lines represent the inactive constraints, while the solid ones are active constraints.

Lines in blue are the activities for the first server and green the second server. The dual variables are also shown. ■

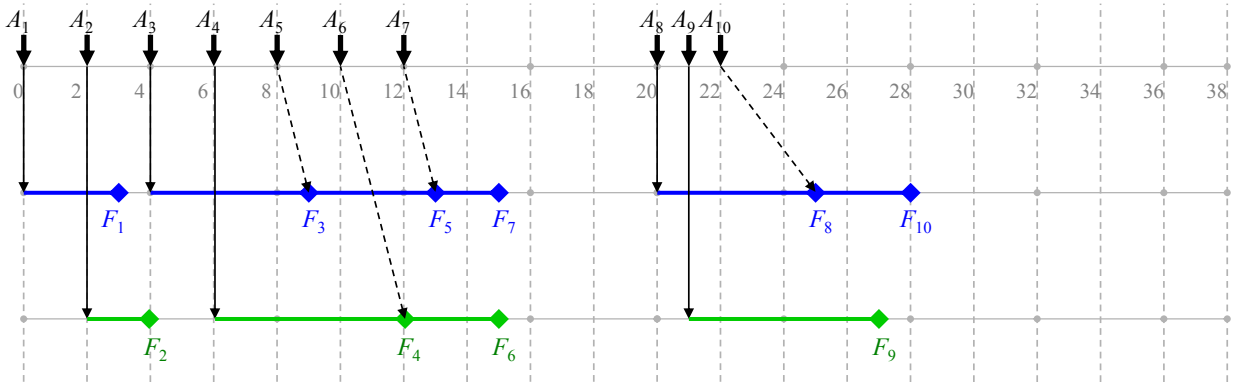

Figure 1: Network of GG2-LP10(F) with n = 10, (A_1, A_2, …, A_{10}) = (0, 2, 4, 6, 8, 10, 12, 20, 21, 22), and (s_1, s_2, …, s_{10}) = (3, 2, 5, 6, 3, 3, 2, 5, 6, 3)

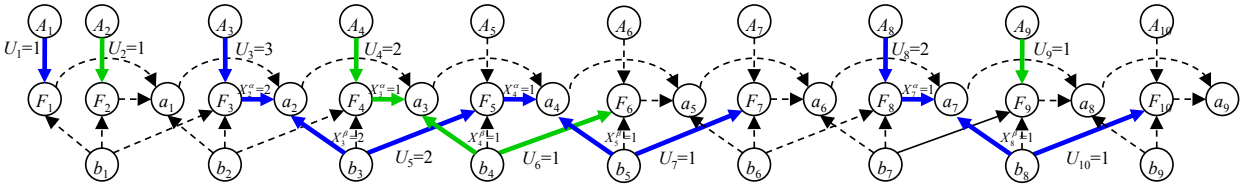

Figure 2: Graphical Representation of the Network of GG2-LP10(F) in Example 2.

4 EXPERIMENTAL RESULTS

In this section, we demonstrate the use of GG2-LP10(F) in obtaining finite-difference gradient estimators. In the limit, the finite-difference gradient estimator becomes the IPA estimator. We, therefore, shows how the dual variables of GG2-LP10(F) can be used to compute the IPA estimator for the service time parameter. The procedure here is similar to the one used in (Chan and Schruben 2006).

In an LP, the dual variables (shadow prices) represent how sensitive the objective function (system performance) is to changes in the right-hand-side random variables (input data) and therefore, provide information necessary for computing gradient estimators using the chain-rule as done in IPA gradient estimation. In fact, perturbations are propagated through all the binding constraints (constraints with zero excess) and the value of each dual variable represents the marginal effect of the corresponding right-hand-side random variable to the objective function. Therefore, all the binding constraints constitute an event-tree (the solution of the dual LP) similar to the one defined in (Suri 1987). As a matter of fact, the LP formulation is a generic event-tree: given a sequence of input random variables, a realization of the event-tree can be constructed by solving the LP and connecting all binding constraints to form the branches.

However, the LP solution obtained from running the simulation might provide more information for a single simulation run because other perturbed sample paths can be reached from the current sample path by some additional computation (pivots), which might be easier than running a new simulation. From the computational point of view, the LP representations would be a potentially effective tool for other types of sensitivity analysis (in particularly, finite difference gradient analysis) when IPA fails, for example, using the dual-simplex method to get new sample paths. Performance of such sensitivity analysis is under investigation.

We focus on the sensitivity of the service time parameter. The sensitivity of the arrival time parameter (arrival rate, $\lambda = 1/E[a_i]$) can be derived similarly. Let $\boldsymbol{b}(\theta) = (\boldsymbol{b}_1(\theta), \boldsymbol{b}_2(\theta))$ be the joint right-hand-sides of the two decomposed GG1-LP(F)'s. Dividing the sum of the two objective functions by n (this will not alter the optimal basis) and taking the limit $n \to +\infty$ gives a consistent estimator of the mean of event times. $\bar{F}(\theta) = \lim_{n\to+\infty} n^{-1} \sum_{i=1}^{n} F_i^*$ a.s., where F_i^*'s are the optimal primal solutions, or equivalently working with the dual,

$$\bar{F}(q) = \lim_{n\to\infty} n^{-1} z(q) = \lim_{n\to\infty} n^{-1} \boldsymbol{b}(q)\boldsymbol{U}^* = \lim_{n\to\infty} n^{-1} \left[\boldsymbol{b}_1(q)\boldsymbol{U}_1^* + \boldsymbol{b}_2(q)\boldsymbol{U}_2^*\right] \text{ a.s.},$$

where $\boldsymbol{U}^* = (\boldsymbol{U}_1^*, \boldsymbol{U}_2^*)$ is the optimal dual vector. For a small perturbation $\Delta\theta$ provided that the order of events remains unchanged—an usual assumption of IPA—the current dual variables remain optimal and therefore, the objective function is perturbed by an amount of $\Delta z = \Delta b \mathbf{U}^*$, where $\Delta\boldsymbol{b}$ is the amount of perturbation of the right-hand side due to the change in θ. The change to the mean event time is then

$$\bar{F}(q + \Delta q) - \bar{F}(q) = \lim_{n\to\infty} \frac{1}{n} \left[\boldsymbol{b}(q + \Delta q) - \boldsymbol{b}(q)\right]\boldsymbol{U}^* = \lim_{n\to\infty} \frac{1}{n} \Delta\boldsymbol{b}\boldsymbol{U}^* \qquad a.s.$$

Divided by $\Delta\theta$ and letting $\Delta\theta \to 0$ yields the derivative of the mean event time. Now, using the same assumptions typically made in IPA, i.e., the random variables $b_i(\theta)$'s are uniformly differentiable—a condition such that the random variables are smooth enough or well-behavior so that IPA works (see Cao 1985 or Ho and Cao 1991 for more details), we have

$$\begin{aligned} \frac{d\bar{F}(q)}{dq} &= \lim_{\Delta q\to 0} \lim_{n\to\infty} \frac{1}{n} \frac{\Delta \boldsymbol{b}(q)}{\Delta q} \boldsymbol{U}^* \qquad a.s. \\ &= \lim_{\Delta q\to 0} \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \frac{\Delta b_i(q)}{\Delta q} U_i^* \\ &= \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} b_i'(q) U_i^* \end{aligned}$$

where $b_i'(\theta)$ is the derivative of $b_i(\theta)$ w.r.t. θ (assume exists) and the last equation uses the uniform differentiability condition. Therefore, the dual variables provide a consistent estimator for the derivative of the mean event time under the usual IPA assumptions.

As discussed in the previous section, in the optimal solution the dual variables (U_i's) are equal to the numbers of customers in a busy period seen by the departing customers. This meaning matches the definition of the IPA estimator developed in the literature (Fu and Hu 1991).

Table 1 gives the experimental results of an $M/M/2$ queue, with mean arrival time $E[a_i] = 1/\lambda = 1$, at low ($\rho = \lambda\theta/2 = \theta/2 = 0.2$, i.e., $\theta = 0.4$), medium (ρ = 0.5, i.e., $\theta = 1$), and high (ρ = 0.8, i.e., $\theta = 1.6$) traffic intensities. Each number is the average of 40 independent replications. These traffic intensities settings are selected to compare the linear programming estimator (LPA) with those presented in (Fu and Hu 1991). The only difference is that the LPA only simulates 50,000 jobs in each replicaton while it was 100,000 busy periods in (Fu and Hu 1991). T is the system time ($F - A$).

Table 1: Gradient Estimators of $M/M/2$ Queue.

ρ	$E[T]_{LPA}$	$E[T]_{Fu,Hu}$	$E[T]$	$\left[\frac{\partial E[T]}{\partial\theta}\right]_{LPA}$	$\left[\frac{\partial E[T]}{\partial\theta}\right]_{Fu,Hu}$	$\frac{\partial E[T]}{\partial\theta}$	$\left[\frac{\partial E[T]}{\partial\lambda}\right]_{LPA}$	$\left[\frac{\partial E[T]}{\partial\lambda}\right]_{Fu,Hu}$	$\frac{\partial E[T]}{\partial\lambda}$
0.2	0.417±0.002	0.417±0.001	0.417	1.130±0.008	1.129±0.005	1.128	–0.035±0.001	–0.035±0.001	–0.035
0.5	1.330±0.011	1.334±0.006	1.333	2.204±0.039	2.222±0.021	2.222	–0.873±0.029	–0.890±0.016	–0.889

0.8	4.411±0.134	4.436±0.032	4.444	12.496±0.786	12.60±0.21	12.65	–15.582±1.141	–15.73±0.31	–15.80

5 CONCLUSION

We present a linear programming representation for $G/G/2$ queue and decompose it into to linear programming representations for $G/G/1$ queue. The decomposed LPs allow us to obtain IPA estimators for service time and arrival time parameters.

Linear programming formulations for closed and open tandem queueing networks have been given in (Chan 2005). There it is shown that the dual variables for different constraints have distinct physical meanings; for example, some of them represent the number of jobs in a busy period while the others equal the number of jobs in a local busy period (for definition of local busy period, see (Fu and Hu 1997)). Therefore, similar LP-based gradient estimators for queueing networks can also be computed using the dual variables.

ACKNOWLEDGMENTS

Part of this work is funded by the National Science Foundation through grant CMMI-0644959 to Rensselaer Polytechnic Institute.

A APPENDIX

Proof of Proposition 1.
The proof includes two parts. The first part is to show that the GG2-LP10(F) can be decomposed into two separated LPs with equality constraints. The second part is to argue that each of these two separated LPs can be transformed into an equivalent GG1-LP(F) with inequality constraints.

We now illustrate the first part of the proof. First, after the pre-solve, the right-hand-side of all the constraints will contain unique variables, i.e., there will not be two (or more than two) constraints sharing the same variable in their right-hand-sides. This one-to-one mapping assigns β_i, α_i, or F_i to one and at most one of β_j, α_j, or F_j. Cyclic assignments are not possible because the indexing of variables is strictly increasing (see also the following paragraph).

Second, this mapping constructs two separate event lists. At the beginning, Constraint 1 links $A_1 + s_1$ to F_1 and $A_2 + s_2$ to F_2. As the two servers are identical, without loss of generality we can assume that F_1 is the time of the first finish event at Server 1 and F_2 at Server 2. Each of the two events then starts a list of events for the corresponding server (with random assignments to break time ties). We discuss the event list created by F_1. The event list led by F_2 can be developed similarly. Either Constraint 2 or 3 will assign F_1 to β_1 or α_1. Suppose Constraint 2 assigns F_1 to β_1 (since $\alpha_0 = F_1$). Constraint 1 will then either link β_1 (plus s_3) to F_3 (which represents a busy period) or re-start the list from $A_3 + s_3$ (which means that the 1st busy period ends with only one customer and a new busy period begins with the 3rd customer). If a new busy period begins, then the argument can be repeated as starting at F_1. If the busy period continues, then F_3 will be linked to either β_2 or α_2 by Constraint 2 or 3. If F_3 is linked to β_2, then the argument can be repeated as starting at β_1. Otherwise, α_2 is linked to α_3 by Constraint 3. Next, α_3 will be linked to either β_4 or α_4 by Constraint 2 or 3. If α_3 is linked to β_4, then the argument can be repeated as starting at β_1. Otherwise, α_3 is linked to α_4 by Constraint 3 and the argument can be repeated as starting at α_3. As this procedure repeats, a set of event lists will be constructed for the services conducted by Server 1. A similar set of event lists will be created for Server 2.

Third, we need to show that the optimal solution for GG2-LP10(F) is also the optimal solution for the two decoupled GG1-LP(F)'s. Let $\boldsymbol{x} = (\boldsymbol{F}, \boldsymbol{\beta}, \boldsymbol{\alpha})^{\mathrm{T}}$ be a feasible solution for GG2-LP10(F), where $\boldsymbol{F}$, $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$ are the vectors of variables. Denote the right-hand-side vector as $\boldsymbol{b} = (\max\{\boldsymbol{A}+\boldsymbol{s}, \boldsymbol{\beta}+\boldsymbol{s}\}, \min\{\boldsymbol{\alpha}, \boldsymbol{F}\}, \max\{\boldsymbol{\alpha}, \boldsymbol{F}\})^{\mathrm{T}}$, where we have abused the vector notation inside the max and min operators to express the right-hand-sides in a compact form. Let $\boldsymbol{I} = (\mathbf{1},\ldots, \mathbf{1})^{\mathrm{T}}$ be the n x n identity matrix, where the i^{th} entry is a col-

umn vector with 1 in the i^{th} element and 0 in all other elements, i.e., $(0,\ldots, 1,\ldots, 0)^T$. We can write GG2-LP(F) using the matrix form:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} F_i \\ s.t. \quad & \boldsymbol{Ix} = \boldsymbol{b} \end{aligned}$$

To define the two decoupled GG1-LP(F)'s, we need the following notation. Let $\boldsymbol{x}_{(k)} = (\boldsymbol{F}_{(k)}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\alpha}_{(k)})^T$ be the set of F_i, β_i, and α_i's assigned to Server k by the pre-solver, $k = 1, 2$. Let $S(k)$ be the index set of all event indexes assigned to Server k, $k = 1, 2$. These assignments separate the constraints into two sets. Denote the corresponding right-hand-side as $\boldsymbol{b}_{(k)} = (\boldsymbol{A}_{(k)}+\boldsymbol{s}_{(k,1)}, \boldsymbol{\beta}_{(k,1)}+\boldsymbol{s}_{(k,2)}, \boldsymbol{\alpha}_{(k,1)}, \boldsymbol{F}_{(k,3)})^T$. In addition, the notation of $\boldsymbol{F}_{(k)}$, $\boldsymbol{\beta}_{(k)}$, $\boldsymbol{\alpha}_{(k)}$ are expanded to $(\boldsymbol{F}_{(k,1)}, \boldsymbol{F}_{(k,2)}, \boldsymbol{F}_{(k,3)}, \boldsymbol{F}_{(k,4)})$, $(\boldsymbol{\beta}_{(k,1)}, \boldsymbol{\beta}_{(k,2)})$, and $(\boldsymbol{\alpha}_{(k,1)}, \boldsymbol{\alpha}_{(k,2)}, \boldsymbol{\alpha}_{(k,3)}, \boldsymbol{\alpha}_{(k,4)})$, respectively, to allow us to match the assignments. With this notation and moving A_i, F_i, β_i, and α_i's from the right-hand-side to the left-hand-side, we can obtain the following LP for Server k, $k = 1, 2$:

$$\begin{aligned} \min \quad & \sum_{i \in S(k)} F_i \\ s.t. \quad & \begin{pmatrix} \boldsymbol{F}_{(k,1)} - \boldsymbol{A}_{(k)} \\ \boldsymbol{F}_{(k,2)} - \boldsymbol{\beta}_{(k)} \\ \boldsymbol{\beta}_{(k,1)} - \boldsymbol{\alpha}_{(k,3)} \\ \boldsymbol{\beta}_{(k,2)} - \boldsymbol{F}_{(k,3)} \\ \boldsymbol{\alpha}_{(k,1)} - \boldsymbol{\alpha}_{(k,4)} \\ \boldsymbol{\alpha}_{(k,2)} - \boldsymbol{F}_{(k,4)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{s}_{(k,1)} \\ \boldsymbol{s}_{(k,2)} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \end{aligned}$$

Fom the 3rd and 4th constraints in above formulation, we have $(\boldsymbol{\beta}_{(k,1)}, \boldsymbol{\beta}_{(k,2)}) = (\boldsymbol{\alpha}_{(k,3)}, \boldsymbol{F}_{(k,3)})$ and from the 5th and 6th constraint, $(\boldsymbol{\alpha}_{(k,1)}, \boldsymbol{\alpha}_{(k,2)}) = (\boldsymbol{\alpha}_{(k,4)}, \boldsymbol{F}_{(k,4)})$. As every α_i must obtain a value from the assignment, we have $(\boldsymbol{\alpha}_{(k,3)}, \boldsymbol{\alpha}_{(k,4)}) = (\boldsymbol{\alpha}_{(k,1)}, \boldsymbol{\alpha}_{(k,2)})$. Combining all the equalities, we have $\boldsymbol{\beta}_{(k)} = (\boldsymbol{\beta}_{(k,1)}, \boldsymbol{\beta}_{(k,2)}) = (\boldsymbol{\alpha}_{(k,3)}, \boldsymbol{F}_{(k,3)}) = (\boldsymbol{\alpha}_{(k,1)}, \boldsymbol{\alpha}_{(k,2)}, \boldsymbol{F}_{(k,3)})\backslash(\boldsymbol{\alpha}_{(k,4)}) = (\boldsymbol{\alpha}_{(k,4)}, \boldsymbol{F}_{(k,3)}, \boldsymbol{F}_{(k,4)})\backslash(\boldsymbol{\alpha}_{(k,4)}) = (\boldsymbol{F}_{(k,3)}, \boldsymbol{F}_{(k,4)})$, where the operator "$(\boldsymbol{X})\backslash(\boldsymbol{Y})$" gives the set of $\boldsymbol{X}$ excluding the set of $\boldsymbol{Y}$. Therefore, we can replace all β_i's in $\boldsymbol{\beta}_{(k)}$ of the 2nd constraint by using $F_i \in (\boldsymbol{F}_{(k,3)}, \boldsymbol{F}_{(k,4)}) = (\boldsymbol{F}_{(k,5)})$, resulting in the following simplified LP:

$$\begin{aligned} \min \quad & \sum_{i \in S(k)} F_i \\ s.t. \quad & \begin{pmatrix} \boldsymbol{F}_{(k,1)} - \boldsymbol{A}_{(k)} \\ \boldsymbol{F}_{(k,2)} - \boldsymbol{F}_{(k,5)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{s}_{(k,1)} \\ \boldsymbol{s}_{(k,2)} \end{pmatrix} \end{aligned}$$

We now begin the second part of the proof, which is to argue that this equality LP can be transformed into an inequality LP identical to GG1-LP(F).

Back to the 1st constraint of the original GG2-LP10(F), the pre-solver could assign A_i to F_i if $A_i \geq \beta_{i-2}$ for all i s.t. $A_i \in (\boldsymbol{A}_{(k)})$ (or equivalently for all i s.t. $F_i \in (\boldsymbol{F}_{(k,1)})$). From the derivation above, we have $\beta_{i-2} = F_{i-1}$ for all i s.t. $F_i \in (\boldsymbol{F}_{(k,1)})$. This means $s_i = F_i - A_i \leq F_i - \beta_{i-2} = F_i - F_{i-1}$ for all F_i's in $\boldsymbol{F}_{(k,1)}$. In other words, for all F_i's in $\boldsymbol{F}_{(k,1)}$ the constraint $F_i - F_{i-1} \geq s_i$ is valid for this LP and we can add this constraint to the LP. On the other hand (again in the 1st constraint of the original GG2-LP10(F)), the pre-solver could assign β_{i-2} to F_i if $A_i \leq \beta_{i-2}$ (time ties are broken arbitrary). These assignments result in the constraint $\boldsymbol{F}_{(k,2)} - \boldsymbol{\beta}_{(k)} = \boldsymbol{s}_{(k,2)}$ that is subsequently transformed into $\boldsymbol{F}_{(k,2)} - \boldsymbol{F}_{(k,5)} = \boldsymbol{s}_{(k,2)}$ in above LP. Because $A_i < \beta_{i-2} = F_{i-1}$, we have $s_i = F_i - F_{i-1} = F_i - \beta_{i-2} \leq F_i - A_i$ for all F_i's in $\boldsymbol{F}_{(k,2)}$. In other words, for all F_i's in $\boldsymbol{F}_{(k,2)}$ the con-

straint $F_i - A_i \geq s_i$ is valid for this LP and we can add this constraint to the LP. As a result, the above LP can be transformed into:

$$\begin{aligned} \min \quad & \sum_{i \in S(k)} F_i \\ s.t. \quad & \begin{pmatrix} \boldsymbol{F}_{(k,1)} - \boldsymbol{A}_{(k)} \\ \boldsymbol{F}_{(k,1)} - \boldsymbol{F}_{(k,5)} \\ \boldsymbol{F}_{(k,2)} - \boldsymbol{A}_{(k)} \\ \boldsymbol{F}_{(k,2)} - \boldsymbol{F}_{(k,5)} \end{pmatrix} \begin{pmatrix} = \\ \geq \\ \geq \\ = \end{pmatrix} \begin{pmatrix} \boldsymbol{s}_{(k,1)} \\ \boldsymbol{s}_{(k,1)} \\ \boldsymbol{s}_{(k,2)} \\ \boldsymbol{s}_{(k,2)} \end{pmatrix} \end{aligned}$$

We can combine the 1st and 3rd constraints and the 2nd and 4th constraints to simply the LP to:

$$\begin{aligned} \min \quad & \sum_{i \in S(k)} F_i \\ s.t. \quad & \begin{pmatrix} \boldsymbol{F}_{(k)} - \boldsymbol{A}_{(k)} \\ \boldsymbol{F}_{(k)} - \boldsymbol{F}_{(k,5)} \end{pmatrix} \geq \begin{pmatrix} \boldsymbol{s}_{(k)} \\ \boldsymbol{s}_{(k)} \end{pmatrix} \end{aligned}$$

Note that we have replaced all equalities by inequalities. Using the scalar notation, we can re-write the LP and obtain the following GG1-LP(F,k) for Server k, k = 1 and 2:

GG1-LP(F,k):

$$\begin{aligned} \min \quad & \sum_{i \in S(k)} F_i \\ s.t. \quad & F_i - A_i \geq s_i \quad \forall i \in S(k) \qquad (U_i) \\ & F_i - F_{i-1} \geq s_i \quad \forall i \in S(k) \qquad (V_i) \end{aligned}$$

The last step is to show that the optimal solutions of the two GG1-LP(F,k), k = 1, 2, are also the optimnal solution of the original GG2-LP10(F). Let $\boldsymbol{F_1}^*$ be the optimal solution for GG1-LP(F,1) and $\boldsymbol{F_2}^*$ for GG1-LP(F,2). Because both $\boldsymbol{F_1}^*$ and $\boldsymbol{F_2}^*$ are feasible to their corresponding constraints, they must also be feasible for the constraints of GG2-LP10(F). Let F_i, i = 1,…, n be an arbitrary solution of GG2-LP10(F). We have:

$$\sum_{i \in S(1)} F_i^* + \sum_{i \in S(2)} F_i^* \leq \sum_{i \in S(1)} F_i + \sum_{i \in S(2)} F_i = \sum_{i=1,\ldots,n} F_i$$

where the first inequality uses the fact that F_i^*'s give the minimum objective value of the corresponding objective function. Therefore, the optimal solutions $\boldsymbol{F_1}^*$ for GG1-LP(F,1) and $\boldsymbol{F_2}^*$ for GG1-LP(F,2) together is also optimal for GG2-LP10(F). □

REFERENCES

Chan, W. K. V. 2005. Mathematical programming representations of discrete-event system dynamics. IEOR Department. Berkeley, University of California.

Chan, W. K. V. 2010. Generalized lindley-type recursive representations for multi-server tandem queues with blocking. ACM Transactions on Modeling and Computer Simulation 20(4):1-19.

Chan, W. K. V., and L. W. Schruben. 2006. Response gradient estimation using mathematical programming models of discrete-event system sample paths. In *Proceedings of the 2006 Winter Simulation Conference.* eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters. 272 - 278. Piscataway, New Jersy: Institute of Electrical and Electronics Engineers.

Chan, W. K. V., and L. W. Schruben. 2008. Optimization models of discrete-event system dynamics. Operations Research 56:1218-1237.

Fu, M., and J. Q. Hu. 1997. *Conditional monte carlo: Gradient estimation and optimization applications.* Boston: Kluwer Academic Publishers.

Fu, M. C., and J. Q. Hu. 1991. Consistency of infinitesimal perturbation analysis for the gi/g/m queue. European Journal of Operational Research 54(1):121-39.

Gal, T. 1979. *Postoptimal analyses, parametric programming and related topics.* London: McGraw-Hill.

Gal, T., and H. J. Greenberg. 1997. *Advances in sensitivity analysis and parametric programming.* New York: Springer.

Glasserman, P. 1991. *Gradient estimation via perturbation analysis.* Boston: Kluwer Academic Publishers.

Gong, W. B., and Y. C. Ho. 1987. Smoothed (conditional) perturbation analysis of discrete event dynamic-systems. IEEE Transactions on Automatic Control 32(10):858-866.

Homem-de-Mello, T., A. Shapiro, and M. L. Spearman. 1999. Finding optimal material release times using simulation-based optimization. Management Science 45(1):86-102.

Suri, R. 1987. Infinitesimal perturbation analysis for general discrete event systems. Journal of the ACM 34(3):686-717.

Suri, R. 1989. Perturbation analysis - the state of the art and research issues explained via the gi/g/1 queue. Proceedings of the IEEE 77(1):114-137.

Suri, R., and M. A. Zazanis. 1988. Perturbation analysis gives strongly consistent sensitivity estimates for the m/g/1 queue. Management Science 34(1):39-64.

Ward, J. E., and R. E. Wendell. 1990. Approaches to sensitivity analysis in linear programming. Annals of Operations Research 27(1-4):3-38.

Zazanis, M. A., and R. Suri. 1994. Perturbation analysis of the gi/gi/1 queue. Queueing Systems 18(3-4):199-248.

AUTHOR BIOGRAPHIES

WAI KIN (VICTOR) CHAN is an Associate Professor of the Department of Industrial and Systems Engineering at the Rensselaer Polytechnic Institute, Troy, NY. He holds a Ph.D. in industrial engineering and operations research from University of California, Berkeley. His research interests include discrete-event simulation, agent-based simulation, and their applications in social networks, service systems, transportation networks, energy markets, and manufacturing. He is a member of INFORMS, IIE, and IEEE. His e-mail address is <chanw@rpi.edu>.

NOWELL CLOSSER is a first-year Ph.D. student in Statistics at the University of Washington, Seattle. He received bachelor's degrees in Mathematics and Computer Science from Rensselaer Polytechnic Institue, Troy, NY. He may be reached at <clossn@uw.edu>.