Approximating the Optimal Weights for Discrete-time Generalized Processor Sharing

Jasper Vanlerberghe, Joris Walraevens, Tom Maertens, and Herwig Bruneel
Stochastic Modeling and Analysis of Communication Systems Research Group (SMACS)
Department of Telecommunications and Information Processing (TELIN)
Ghent University (UGent)
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
Email: {jpvlerbe,jw,tmaerten,hb}@telin.UGent.be

***Abstract*—Generalized Processor Sharing (GPS) is a simple, flexible and fair scheduling mechanism to achieve delay differentiation between several customer classes. The amount of delay differentiation is regulated by the weights given to the classes. In this paper we assume a discrete-time, two-class GPS queueing system. Our goal is to derive the optimal weights in order to minimize a weighted sum of functions of the mean delays of both classes. As analytical results are scarce we use an approximation method. The approximation is based on power series expansions of the mean queue length of each of the queues for certain weights. Padé approximants are used to extrapolate the approximation to the whole domain of possible weights, resulting in a set of approximations. An algorithm is proposed to filter out the infeasible solutions (with regard to monotonicity and other characteristics of the system) and aggregate the others, resulting in a single approximation. The result proves to be an accurate approximation of the optimal weights w.r.t. the cost function. For a load of 90% we have a maximum misprediction of 1% of the cost, in the case of a weighted sum of squares of the mean delays. The main contribution of this article is that power series approximations can be used effectively for optimization purposes.**

***Keywords*—Generalized Processor Sharing (GPS), optimization, queueing, approximation**

I. Introduction

Many queueing systems today require a means to differentiate service between different classes of customers. This provides each class with a different quality of service while using the system efficiently. Generalized Processor Sharing (GPS) [1], [2] is a scheduling mechanism which can provide this QoS in a fair and flexible manner. When using GPS, each customer class gets assigned a certain weight and the server distributes its available capacity according to these weights. The class with the highest weight amongst the backlogged classes, gets a bigger average share of the service capacity.

For practical applications one needs to determine the optimal weights in a given situation. It is widely known that a queueing system using GPS is very hard to analyze. For instance for a two-class system with weights β and $1-\beta$, very few analytical results are available for general β. In special cases that β equals 0 or 1 however, the scheduling simplifies to a strict priority system. Analytical resuls are available for a broad class of strict priority systems [3], [4]. For general β, results are obtained using an equivalent formulation as a boundary value problem [5]–[7], using approximations and/or simulation [1], [2], [8], [9].

ISBN 978-3-901882-58-6

A commonly used method to derive approximations in queueing theory is the use of Taylor series. With this technique one represents the function that needs to be approximated as a series in one of the parameters. Subsequently one tries to calculate expressions for the coefficients. The approximation is achieved by truncating the series (only calculating the first several coefficients); mostly this is dictated by computational constraints. This technique is for instance used for systems with complex arrival processes in [10], [11] and for systems with coupled queues in [12].

In this paper, we use the technique from Walraevens et al. [5] to find a set of approximations for certain performance measures. In Walraevens et al., it is shown that these approximations can be accurate. However, it is not known if the approximations are accurate enough to be used in optimizations. The latter is the topic of this paper.

We therefore study a rather simple queueing system with two classes of customers and a single server. We consider time to be discrete and call the time intervals slots. Each customer requires a deterministic service time of 1 single slot from the server. The server chooses a customer to serve in each slot. When customers of both classes are backlogged the server chooses a class 1 customer with probability β, with β the weight assigned to class 1. A customer of class 2 is chosen with probability $1-\beta$. When only one class is backlogged, this class is served, so the system is work conserving. We define the probability generating function (pgf) of the arrival process as $A(z_1, z_2) = E[z_1^{a_1} z_2^{a_2}]$ with a_i the number of arrivals of class i in a random slot in steady state. Furthermore λ_i denotes the arrival rate of class i and we also define $\lambda_T = \lambda_1 + \lambda_2$ as the total arrival rate and $\alpha = \frac{\lambda_1}{\lambda_T}$ as the fraction of class 1 arrivals. The queueing system is depicted in Figure 1. We assume a stable system.

We develop an algorithm to filter the set of approximations obtained using [5] by selecting the approximations that respect some known analytic properties of the performance measures. By using the average of the selected approximations, we find an accurate approximation for the optimal weights with respect to a given cost function. We validate this approximation with simulation results for several arrival processes and cost functions.

The paper is outlined as follows. In the following section

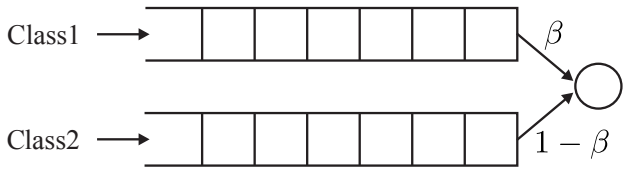


Fig. 1: Studied queueing model

we introduce the Power Series Approximation from Walraevens et al. [5]. We elaborate on the approximation of the mean queue lengths. In Section III, we introduce the cost function we want to minimize. The main contribution of this paper is presented in Section IV. We present some observations about the queueing system which leads to a small series of steps resulting in an accurate approximation of the optimal weight given the cost function of Section III. In Section V we evaluate the performance of our approximation by comparing with simulation results. In the last section we present the main conclusions to be drawn from this paper.

II. Power Series Approximation

We consider the Power Series Approximation from [5]. The more general GPS queueing system in Walraevens et al. is easily simplified to the one we use in this paper. Walraevens et al. decompose the bivariate pgf of the steady state queue contents $U(z_1, z_2) = E[z_1^{u_1} z_2^{u_2}]$ as a power series in β, namely

$$U(z_1, z_2) = \sum_{m=0}^{\infty} V_m(z_1, z_2)\beta^m. \tag{1}$$

Here u_i is the number of customers of class i ($i = 1, 2$) at the beginning of a random slot in steady state. The authors present an iterative procedure to calculate the expressions for V_m starting from V_0. The expression for V_0 is easily found from the case of strict priority scheduling [4], [13]; this can be seen by setting $\beta = 0$. For higher values of m the coefficients become progressively more computationaly intensive, which in practice bounds the number of coefficients that can be computed. By setting the higher coefficients to zero, the power series is truncated and represents an approximation of the actual pgf.

Using this technique we can calculate the exact mean number of customers present in the system and its derivatives in $\beta = 0$. To get the values for $\beta = 1$, we can exploit the symmetry of the system. This is done by interchanging the arrival process, i.e. send customers of type 2 to queue 1. We use the following formulas derived from [5]

$$\bar{u_i}^{(m)}(0) = \left.\frac{\partial V_m(z_1, z_2)}{\partial z_i}\right|_{z_1=z_2=1} \tag{2}$$

$$\bar{u_i}^{(m)}(1) = \left.\frac{\partial V_m^*(z_1, z_2)}{\partial z_{1-i}}\right|_{z_1=z_2=1}, \tag{3}$$

$i = 1, 2$. Here $\bar{u}_i^{(m)}(\beta)$ denotes the m-th derivative of $\bar{u}_i(\beta)$, this in turn represents the mean queue content of class i which depends on the weight parameter β of the GPS system. V_m^* equals V_m with the arrival process switched as discussed before, i.e. we substitute the pgf $A(z_1, z_2)$ of the number of arrivals in a slot by $A(z_2, z_1)$. The factor z_{1-i} in the denominator of Equation (3) also arises from this switching.

We calculate up to the m-th order approximation of $U(z_1, z_2)$ by calculating the coefficients V_0 up to V_m, resulting in $2(m + 1)$ parameters for $\bar{u}_i(\beta)$, namely up to the m-th derivative in $\beta = 0$ and $\beta = 1$. Note that this includes the 0-th order derivative or the actual value in $\beta = 0$ and $\beta = 1$. The values of these derivatives can be used to approximate $\bar{u}_i(\beta)$ with a polynomial of degree $2m + 1$. We extend this to Padé approximants resulting in a set of approximations including the aforementioned polynomial. Padé approximants are rational functions

$$[L/K](\beta) = \frac{\sum_{l=0}^{L} c_{1,l}\beta^l}{\sum_{k=0}^{K} c_{2,k}\beta^k}, \tag{4}$$

with $L + K = 2m + 1$ using the normalization $c_{1,0} = 1$. They are therefore the easiest extension of polynomials as approximating functions. The Padé approximants thus have $2m + 2$ unknown coefficients left; these need to be calculated from the known values of the derivatives in 0 and 1. This results in $2(m + 1)$ different functions ($L = 0, \ldots, 2m + 1$) of which the $[L/0]$ approximant represents the polynomial approximation. Due to the way the parameters are obtained, the approximations are exact in $\beta = 0$ and $\beta = 1$, as are all derivatives up to the m-th one in these points. This means that the Power Series Approximation is expected to be less accurate moving further away from the endpoints 0 and 1 as is also described in [5]. That is why it is not certain that these approximations are useful for optimization purposes, as we want to optimize β over the complete interval $[0, 1]$.

III. Cost Function

To determine the optimal weights for the GPS system we minimize a cost function F which is a weighted sum of a function (g) of the average delays of a random packet ($\bar{d}_j(\beta)$) of both classes:

$$F(\gamma, \beta) = \gamma g(\bar{d}_1(\beta)) + (1 - \gamma)g(\bar{d}_2(\beta)). \tag{5}$$

As we proved in [8] for *linear* or *concave* g the optimal weight is either 0 or 1. In these cases there is consequently no need for an approximation, nor for simulation as the exact values for 0 or 1 are known (from priority). For convex g however we proved that the optimal weight can differ from 0 or 1.

Note that Equation (5) could also be constructed with the average queue contents as is done in [8]. We however only use the delay in this paper to unclutter notations and because delay differentiation is most important in practice. The average delay can be calculated from the average queue length (discussed in Section II) using Little's law: $\bar{d}_i(\beta) = \frac{\bar{u}_i(\beta)}{\lambda_i}$. As is proven in theorem 2 of [8], $\beta_{\text{opt}}(\gamma) = \phi^{-1}(\gamma)$ with ϕ^{-1} the inverse function of

$$\phi(\beta) = \frac{\lambda_1 g'(\bar{d}_2(\beta))}{\lambda_1 g'(\bar{d}_2(\beta)) + \lambda_2 g'(\bar{d}_1(\beta))}. \tag{6}$$

This result can easily be seen from calculating the stationary points of F (Equation (5)) with respect to β and solving for γ. The function $\phi(\beta)$ will play a primordial role in our algorithm.

Taking the n-th derivative of $\phi(\beta)$ we see that only derivatives up to the n-th of $\bar{d}_i(\beta)$ appear in the resulting expression.

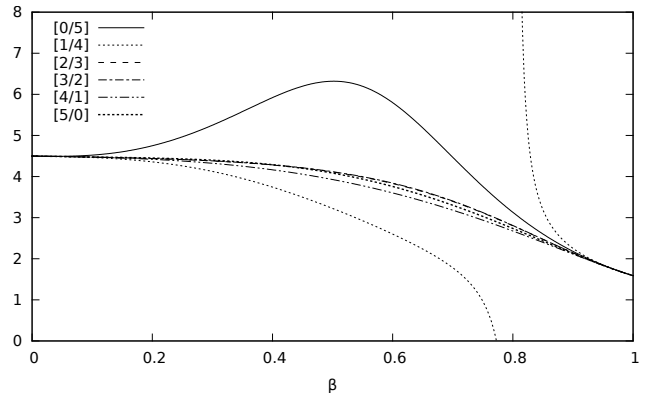


Fig. 2: $\bar{u}_1(\beta)$ for a binomial arrival process with $N = 16$, $\alpha = 0.8$ and $\lambda_T = 0.9$ and single slot service times

As a consequence, if we know the exact values of up to the n-th derivative of $\bar{d}_i(\beta)$ we can also calculate the exact values of up to the n-th derivative of $\phi(\beta)$. We know from [8] that the inverse of $\phi(\beta)$ exists. By using the chain rule and taking the derivative of both sides of the equation $\phi^{-1}(\phi(\beta)) = \beta$ we can easily see that having the n-th derivative of $\phi(\beta)$ also gives us the n-th derivative of $\beta_{\text{opt}}(\gamma) = \phi^{-1}(\gamma)$.

IV. Some Observations

We now have all the pieces to build an approximation for β_{opt}. This is done by calculating an approximation for both $\bar{u}_1(\beta)$ and $\bar{u}_2(\beta)$ using the Power Series Approximation. Using Little's law, we find approximations for the functions $\bar{d}_i(\beta)$ which are subsequently used to calculate $\phi(\beta)$. Inverting this last function eventually leads to $\beta_{\text{opt}}(\gamma)$. In general this easy approximation is however far from accurate. In the remainder of this section we make some observations about the mean delay in a GPS queue and the approximations via Padé approximants. Subsequently we use these observations to derive a method to improve the approximation.

The first observation is that Padé approximants are rational functions and can have singularities in the interval $[0, 1]$. This could be a reason to stick with polynomials as these cannot have any singularities, but our results show that at least one of the Padé approximants always leads to a better approximation. The difficulty is to filter out the good approximations from the bad. For instance the $[1/4]$ Padé approximant of $\bar{u}_1(\beta)$ in Figure 2 has a pole for $\beta = 0.8$. Obviously the actual function $\bar{u}_i(\beta)$ cannot have such poles as the system is assumed to be stable for all β. This means that we can a priori discard the approximants which have denominators with zeros in $[0, 1]$.

A second observation is that some of the approximants are non-monotone. An example can also be seen in Figure 2; the $[0/5]$ Padé approximant has a maximum. It can however be proven that $\bar{u}_1(\beta)$ is monotonically decreasing and $\bar{u}_2(\beta)$ is monotonically increasing (increasing β leads to a higher share of the server for class 1 customers, see also [14]). We can thus also filter out these approximants by calculating the zeros of the first derivative: zeros between 0 and 1 indicate a local extremum and therefore non-monotonicity.

Calculate:

1) $\bar{u}_T$
2) Power Series Approximation for $U(z_1, z_2)$
3) values for $\bar{u}_i^{(m)}(0)$ and $\bar{u}_i^{(m)}(1)$
4) Padé approximants for $\bar{u}_i(\beta)$ using the values from the previous step
5) filtered set of approximants (Omit the non-monotonic approximants and the approximants with singularities)
6) $\phi(\beta)$ as in (6), from each approximant of the filtered set using the relation $\bar{u}_T = \bar{u}_1(\beta) + \bar{u}_2(\beta)$
7) average all remaining $\phi(\beta)$ functions
8) inverse of this average to obtain approximation for $\beta_{\text{opt}}(\gamma)$

Fig. 3: Algorithm

Last but not least, we know that

$$\bar{u}_1(\beta) + \bar{u}_2(\beta) = \bar{u}_T, \tag{7}$$

following from the work conserving property of the queueing system. In this formula $\bar{u}_T$ signifies the mean total amount of customers in the system and is independent of the scheduling discipline (and thus of β). The latter can be calculated as the mean total system content of a system with only one queue [15] and aggregation of the two classes of customers. We however observe that independently calculating approximations for $\bar{u}_1(\beta)$ and $\bar{u}_2(\beta)$ does not sum to $\bar{u}_T$, and in a lot of cases is far off. As Equation (7) is not satisfied, this results in bad approximations. Our tests have shown that a better approximation for $\phi(\beta)$ is found when calculating the approximant for one of the queues and using Equation (7) to calculate the approximant for the complementing queue, as opposed to calculating the approximants for both queues independently.

An extra advantage of coupling the approximants for $\bar{u}_i(\beta)$ is that we obtain twice the amount of approximations for $\phi(\beta)$. For instance if we calculate the m-th order approximation for $U(z_1, z_2)$ we can calculate $2(m + 1)$ parameters for $\bar{u}_i(\beta)$, i.e. $2(m + 1)$ Padé approximants for $\bar{u}_1(\beta)$ and $2(m + 1)$ approximants for $\bar{u}_2(\beta)$. This leads to $4(m+1)$ approximations for $\phi(\beta)$. Most of these approximations are different except for both polynomial approximants, which already satisfy Equation (7) (this can easily be seen from the way they are constructed).

After filtering the invalid solutions (non-monotonic, singularities) we end up with a set of approximations for $\phi(\beta)$. The best approximation from this set depends on the parameters of the studied system. When no simulation is available (which is obviously the case when one wants to use the approximations in practice) it is consequently not a priori known which approximation performs best. Therefore, we propose as a heuristic to aggregate the set of approximations by taking the average over the set. Taking all this together we get the algorithm from Figure 3.

It should be noted that the expressions obtained for the Padé approximants are symbolic and that they do not have to be recalculated for each set of parameters. Filtering the approximations for singularities and non-monotonicity should be done for each parameter set as the appearance of these phenomena depends on the values for the parameters. We can thus use the approximation to do sensitivity analysis on the parameters of the system without the need for large amounts of computations.

V. Numerical results

For the numerical results we used Monte-Carlo simulations as a reference to evaluate the performance of our approximations. This was done by running a simulation for every β from 0 to 1 with intervals of 0.01. For different β we used the exact same arrival process and random variables to pilot the scheduling of the two types of customers (so called coupled random generators, see [16], [17]). Using these identical trajectories minimizes the variance within the simulation of a specific arrival process but with a different scheduling (i.e. weight value). Each simulation starts from an empty system. We simulate 10^9 slots, this is long enough to minimize bias from the transient period and the specific choice of the arrival process [8]. For 10 simulations the standard deviation averages to only 0.02% of the mean, with a maximum of 0.03%.

We study two different arrival processes. For both classes, the packets require a single slot of service.

a) Arrival process 1: The first arrival process we study is a two-dimensional binomial arrival process with pgf

$$A(z_1, z_2) = \left(1 - \frac{\lambda_1}{N}(1 - z_1) - \frac{\lambda_2}{N}(1 - z_2)\right)^N. \quad (8)$$

This is the arrival process in a queue of an $N \times N$ output-queueing switch with Bernoulli arrivals at its inlets and with independent and uniform routing towards the outlets [4]. The parameters λ_i denote the arrival rates of class i as defined before. We choose $N = 16$ throughout.

b) Arrival process 2: The second arrival process is a bivariate Bernoulli process with pgf

$$\begin{aligned} A(z_1, z_2) =& 1 - \lambda_1 - \lambda_2 + \rho + (\lambda_1 - \rho)z_1 + (\lambda_2 - \rho)z_2 \\ &+ \rho z_1 z_2. \end{aligned} \quad (9)$$

From this pgf we see that λ_i is again the arrival rate of class i in accordance with previous definitions. ρ is the probability that packets from both classes arrive in the same slot. No more than two packets can arrive in the same slot using this arrival process. If $\rho = 0$ then there is at most one arrival per slot as a consequence this packet can be served in the slot thereafter and no backlog occurs.

A. Influence of arrival process

For a cost function with a quadratic g function we plotted some results for various parameters of the first arrival process in Figure 4. The leftmost figures display the performance of increasing orders (m) of the approximation for $\beta_{\text{opt}}(\gamma)$ by plotting them together with results from simulation. In the remainder of this paper we will denote the real optimal β as β_{opt}, this value is obtained by simulation. Approximations for the optimal β are denoted by $\tilde{\beta}_{\text{opt}}$, these values are obtained by applying the algorithm of Figure 3. In the center figures, we have plotted the value of the cost function for β_{opt} as a function of γ with a solid line. The other curves are the real costs that would be obtained when using the suboptimal $\tilde{\beta}_{\text{opt}}$ value found from the respective approximations. The graphs on the right show the fractional difference between the approximation curves and the simulated curve in the center figures. This fractional difference is calculated as

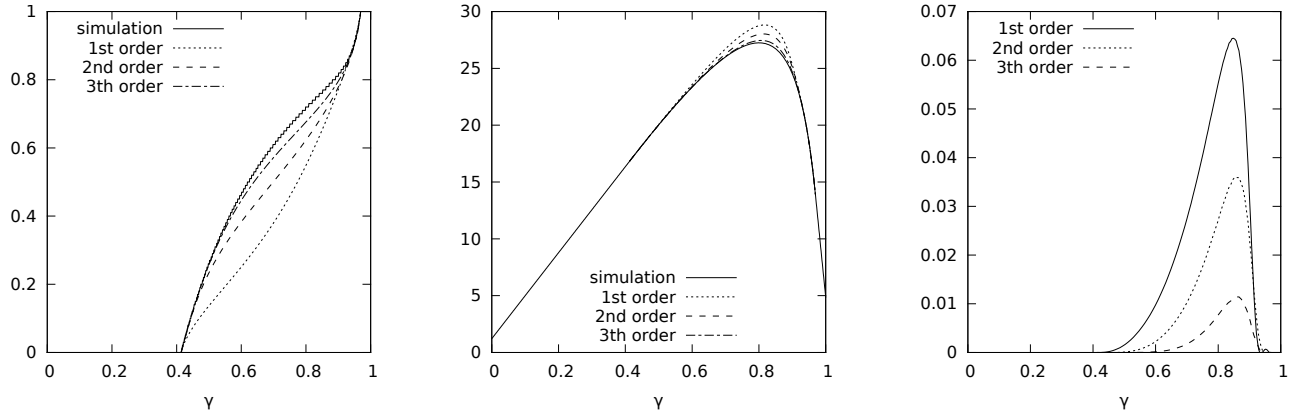
$$\frac{\left|F(\tilde{\beta}_{\text{opt}}) - F(\beta_{\text{opt}})\right|}{F(\beta_{\text{opt}})}.$$

As can be seen from the figures, increasing the order of the approximation m always leads to a better approximation of the real results. Specifically we see that the approximation at the endpoints (β equals 0 or 1) is progressively better. This could be expected by the way in which we constructed the approximation and the properties of the Power Series Approximation. It was already shown analytically in the last paragraph of Section III. We now also visually see that increasing the order matches more derivatives of $\beta_{\text{opt}}(\gamma)$ at the endpoints. The required extra computational effort thus pays off.

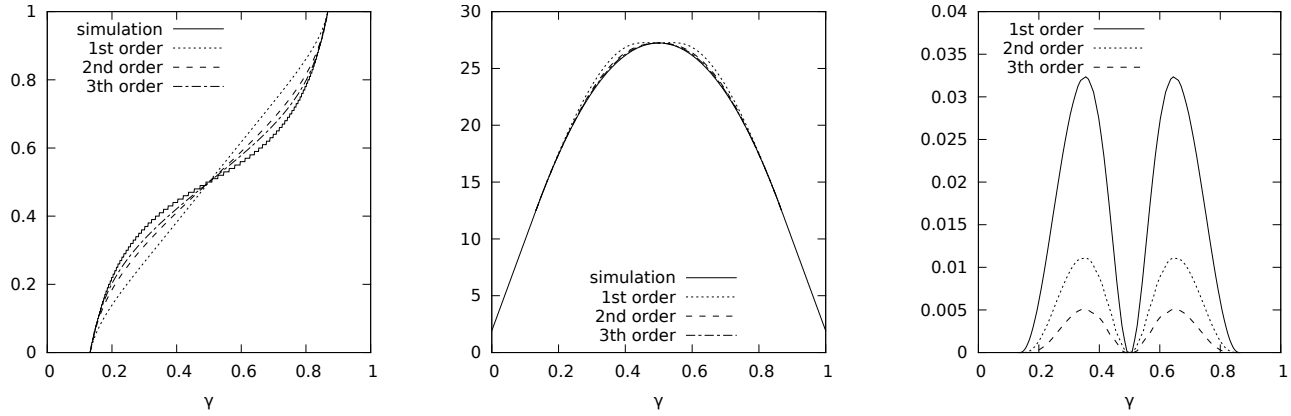
We also see that misprediction of β_{opt} does not lead to the same level of misprediction of the cost. For instance we see from the left graph of Figure 4a that the maximum misprediction of the optimal weight by the first order approximation is 33% at $\gamma = 0.7$. Looking at the right graph we however see that the misprediction of the optimal cost is about 3%. We see a maximum misprediction of the optimal cost of 6.5% for $\gamma = 0.85$ where the error on β_{opt} is 18%. We conclude that even with a far from optimal weight value we can get close to the optimal cost.

One other conclusion is that a higher load on the system reduces the accuracy of the approximation. The error (w.r.t. the real cost) for a load of 90% is at its worst about 1% using the third order approximation, but for a load of 70% the approximation is nearly perfect. Even when only the first order approximation is used the fractional difference is less than 0.2%. Using the third order approximaton we got an error of 30% for a load of 99% and for a load of 50% we had a maximum error of 0.05%. This follows from the fact that with a lower load there is less queuing, this results in lower values for $\bar{u}_i(\beta)$. As a consequence the difference between $\phi(0)$ and $\phi(1)$ is smaller, leaving less room (and thus error) for possible functions in between.

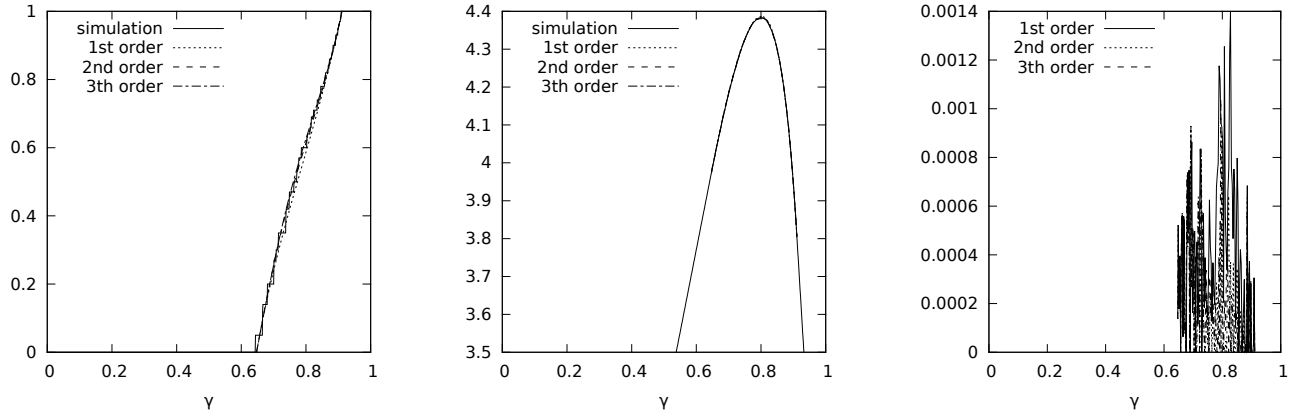
For the same cost function we studied the GPS queue with the second arrival process. We always used the third order approximation for tests with the bivariate Bernoulli arrival process. Figure 5 shows the mean absolute error on β_{opt} by the approximation. This mean is calculated by sampling over γ. For each gamma we calculate the absolute difference between β_{opt} from simulation and $\tilde{\beta}_{\text{opt}}$. Subsequently the mean is calculated over all samples. Visually it is the mean vertical distance between the simulated curve and an approximated curve for β_{opt} in the left graph of Figure 4. The mean fractional error on F when $\tilde{\beta}_{\text{opt}}$ is used, is shown in Figure 6. Visually this is the mean value for one of the curves in the rightmost graph in Figure 4, obtained by sampling. We did this experiment for $\alpha = 0.2$ and $\alpha = 0.5$ whereby ρ was



(a) $g(x) = x^2$, $N = 16$, $\alpha = 0.8$, $\lambda_T = 0.9$



(b) $g(x) = x^2$, $N = 16$, $\alpha = 0.5$, $\lambda_T = 0.9$



(c) $g(x) = x^2$, $N = 16$, $\alpha = 0.8$, $\lambda_T = 0.7$

Fig. 4: $\beta_{\text{opt}}(\gamma)$ (left), real F when using approximation (center), fractional difference (right) for several parameter combinations.

kept equal to $\frac{\lambda_1}{2}$. The same conclusions as in the previous paragraph can be drawn. A higher load does not increase the misprediction on β_{opt} whereas it does increase the relative induced error on F. For one dataseries we also included the standard deviation over the error samples as error bars in the graph. As the load increases so does the standard deviation. We only show error bars for one series, as to not overload the figure, but the same trend is seen for the other series. This all proves that the approximation is less accurate for higher loads, not only on average, there is also a higher variability on the error.

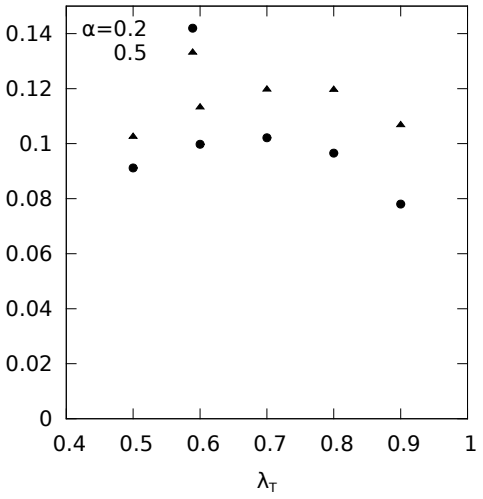


Fig. 5: Mean absolute error on β_{opt} for bivariate Bernoulli arrival process with increasing load.

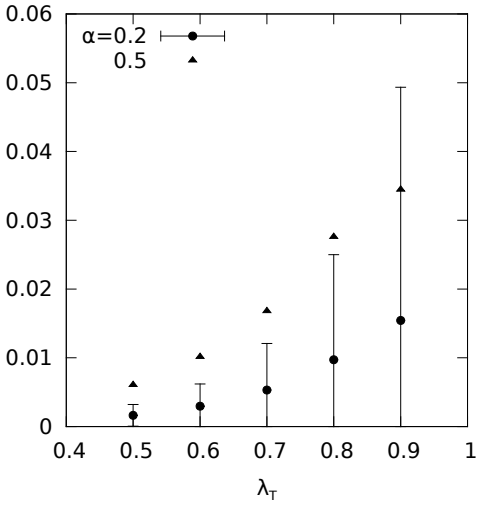


Fig. 6: Mean fractional error on F by using approximated β_{opt} for bivariate Bernoulli arrival process with increasing load.

To test the robustness of the approximation we ran a more involved simulation. The bivariate Bernoulli arrival process has three parameters $(\lambda_1, \lambda_2, \rho)$. We ran 108 separate simulations taking samples out of the possible parameter space for the arrival process. Our program chose λ_T uniformly in $[0.3, 0.99]$ (excluding very low loads), subsequently α in $[0.01, 0.99]$ and lastly ρ in $]0, \min(\lambda_1, \lambda_2)]$ (excluding 0 as there is never any backlog in that situation). From these parameters the conversion to the parameters $\lambda_1, \lambda_2, \rho$ is evident.

For each sample point we also calculated the third order approximation according to the algorithm in Figure 3. Subsequently the mean absolute error and standard deviation on the difference between β_{opt} and $\tilde{\beta}_{\text{opt}}$ were calculated. Secondly, we calculated the fractional difference between the value of the cost function at β_{opt} and the value of the cost function at $\tilde{\beta}_{\text{opt}}$ from the approximation. Lastly we calculated the mean and standard deviation over all samples for these values, they are displayed in Table I. For instance the top left cell 0.085 is the mean (column) over all 108 samples of the means (row) per sample over γ; the top right cell 0.025 is the standard deviation (column) over the 108 samples of the means (row) per sample over γ. We can see that the approximation performes well with a mean fractional error on F of less than 1% and a mean standard deviation of less than 2%.

TABLE I: Performance of the approximation over 108 random samples for the parameters of the second arrival process.

			Over all samples	
			Mean	STDEV
Over all γ	$\left\lvert\tilde{\boldsymbol{\beta}}_{\text{opt}} - \boldsymbol{\beta}_{\text{opt}}\right\rvert$	Mean	0.085	0.025
		STDEV	0.053	0.019
	$\frac{\left\lvert F(\tilde{\beta}_{\text{opt}}) - F(\beta_{\text{opt}})\right\rvert}{F(\beta_{\text{opt}})}$	Mean	0.89%	1.12%
		STDEV	1.77%	2.67%

Each of the 108 simulations took about 3 hours on a modern processor (using a single core). The calculations for the Padé approximants are done in less then 5 minutes and can be reused for each sample (steps 1-4 of the algorithm in Figure 3). In fact it can be reused by everyone for all possible parameters and arrival processes, so it really only needs to be calculated *once*. Filtering and averaging the approximations (steps 5-8) needs to be done for each sample, but this is done instantly. The time gains from using the approximation are self-evident.

B. Influence of cost function

Now we look at the influence of the cost function for a given arrival process; specifically we vary the g function. As mentioned, we keep the g functions convex as linear or concave functions do not yield optimal weights differing from 0 or 1. These results are given in Figure 7 in the same format as in Figure 4.

We see that with higher powers of the average delay (i.e.: higher n with $g(x) = x^n$) the amount of misprediction of β_{opt} stays roughly the same, but the fractional cost difference increases. This can be explained by making the following reasoning. Take two different cost functions one (F_1) with a low power, say $g_1(x) = x^2$, and one (F_2) using a high power, for instance $g_2(x) = x^{20}$. If the approximations for both cost functions indicate the same β_{opt} for a given γ in the cost function, both will have used the same Padé approximants for $\bar{d}_i(\beta)$. Consequently they will both have the same misprediction on $\bar{d}_i(\beta)$. Leading to roughly the same misprediction on β_{opt}. However in F_2 this misprediction on $\bar{d}_i(\beta)$ will get blown up because of raising the power, leading to a higher fractional difference.

The higher n, the more the $\beta_{\text{opt}}(\gamma)$ function approaches a constant function for $\gamma \in]0, 1[$, as explained in [8]. This

already makes it harder to approximate the real β_{opt} using the method we presented here (since we use continuous rational functions). For this class of g function we should develop other techniques to achieve reasonable approximations. One can also wonder how relevant these kinds of higher order g functions are in practice.

For the first order approximation of $\beta_{\text{opt}}(\gamma)$ we can see a kink in the graph which becomes more outspoken for higher n. A clear example hereof is the leftmost graph in Figure 7d at $\gamma = 0.7$. This is a consequence of the averaging over the set of $\phi(\beta)$ approximations, whereas the individual approximations in this set do not have this kink. We zoom in on the example of the first order approximation in Figure 7d. $\phi(\beta)$ approaches a step function with a very steep gradient at $\beta = 0.7$. Before averaging we have three clusters of approximations; each cluster approaches a step function with the step at another value for β. We see a clusters with steps at β equal to 0.4, 0.65 and 0.72. Averaging results in multiple steps at each of these values, where the height of the step is largely determined by the number of approximations in the cluster. Subsequently calculating the inverse gives us (quasi-)horizontal parts in the graph at 0.4, 0.65 and 0.72 and the resulting kinks at $\gamma = 0.7$ and $\gamma = 0.85$.

Another way to form an aggregation could be to take the average over the approximations for $\bar{d}_i(\beta)$ and construct one $\phi(\beta)$ from this average. As can be seen from Figure 8 we see that the alternative method does not present this kink. For the exotic case of $g(x) = x^{20}$ it even performs better, or at least more consistently bad. Where our proposed method performs worse for most γ it performs much better in a narrow range.

However overall our tests have shown that this alternative method leads to less accurate results as is indicated in Figure 9. We see that the method proposed in Figure 3 performs about 3% better at the maximum misprediction point using the first order approximation. For the second order approximation this reduces to 0.5%. Lastly for the third order approximation the difference between both methods further decreases, leading to quasi-identical results. For other parameters of the arrival process or other g functions we also see that both methods achieve the same results for the third order approximation. As this kink only presents itself when using g functions that are not used in practice, we suggest to use the method we proposed in Figure 3. The exotic case for $g(x) = x^{20}$ is only mentioned here for completeness and a clearer illustration of this kink phenomenon.

VI. Conclusion

In this paper we have shown that we can use the Power Series Approximation to derive an approximation for the optimal weights in a discrete-time Generalized Processor Sharing queueing system with two classes. We have validated our approximation against extensive simulation results over a wide parameter range. The optimal weights are found with respect to a general class of cost functions, considering the fact that for a range of functions it is already proven that priority scheduling is always optimal. The performance of the approximation depends on the specific (parameters of) the arrival process and cost function. For a weighted average of the squares of the mean delay of both classes and an arrival process offering a load of 90% we obtained a deviation of only 1% with respect to the real optimum, with much less computational effort than through simulations. Obvious future work is the extension to more than 2 customer classes, this is however non trivial as a Power Series Approximation for this kind of GPS system is not readily available.

Acknowledgment

This research has been co-funded by the Interuniversity Attraction Poles (IAP) Programme initiated by the Belgian Science Policy Office.

References

[1] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking (TON)*, vol. 1, no. 3, pp. 344–357, 1993.

[2] A. K. Parekh and R. G. Gallagher, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case," *IEEE/ACM Transactions on Networking (TON)*, vol. 2, no. 2, pp. 137–150, 1994.

[3] J. Walraevens, B. Steyaert, and H. Bruneel, "Delay characteristics in discrete-time GI-G-1 queues with non-preemptive priority queueing discipline," *Performance Evaluation*, vol. 50, no. 1, pp. 53–75, 2002.

[4] ——, "Performance analysis of a single-server ATM queue with a priority scheduling," *Computers & Operations Research*, vol. 30, no. 12, pp. 1807–1829, 2003.

[5] J. Walraevens, J. van Leeuwaarden, and O. Boxma, "Power series approximations for two-class generalized processor sharing systems," *Queueing systems*, vol. 66, no. 2, pp. 107–130, 2010.

[6] G. Fayolle and R. Iasnogorodski, "Two coupled processors: the reduction to a Riemann-Hilbert problem," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 47, no. 3, pp. 325–351, 1979.

[7] J. W. Cohen and O. J. Boxma, *Boundary value problems in queueing system analysis*. Elsevier, 2000.

[8] J. Vanlerberghe, T. Maertens, J. Walraevens, S. De Vuyst, and H. Bruneel, "A hybrid analytical/simulation optimization of generalized processor sharing," in *Proceedings of The 25th International Teletraffic Congress (ITC 25)*, Shanghai, September 2013.

[9] J. F. Kingman, "Two similar queues in parallel," *The Annals of Mathematical Statistics*, vol. 32, no. 4, pp. 1314–1323, 1961.

[10] S. Ouazine, K. Abbas, and B. Heidergott, "The Taylor series expansions for performance functions of queues: Sensitivity analysis," in *Analytical and Stochastic Modeling Techniques and Applications*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 7984, pp. 1–11.

[11] K. De Turck, D. Fiems, S. Wittevrongel, and H. Bruneel, "A Taylor series expansions approach to queues with train arrivals," in *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 447–455.

[12] K. De Turck, E. De Cuypere, S. Wittevrongel, and D. Fiems, "Algorithmic approach to series expansions around transient Markov chains with applications to paired queuing systems," in *Performance Evaluation Methodologies and Tools (VALUETOOLS), 2012 6th International Conference on*. IEEE, 2012, pp. 38–44.

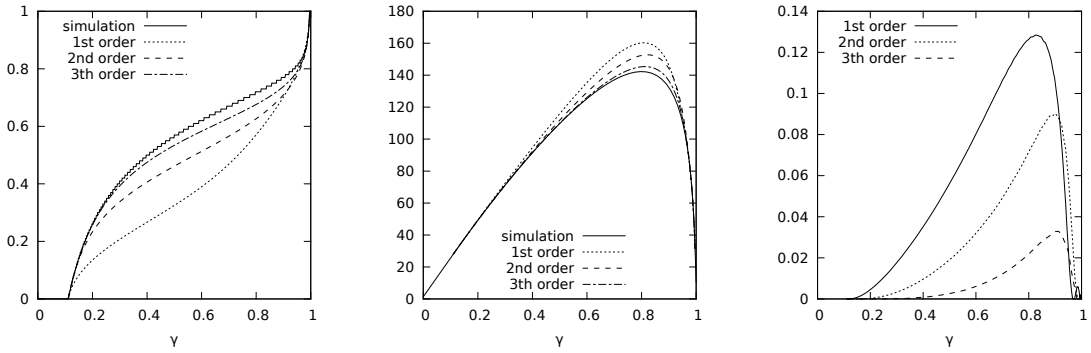
[13] K. Laevens and H. Bruneel, "Discrete-time multiserver queues with priorities," *Performance Evaluation*, vol. 33, no. 4, pp. 249–275, 1998.

[14] I. M. Verloop, U. Ayesta, and S. Borst, "Monotonicity properties for multi-class queueing systems," *Discrete Event Dynamic Systems*, vol. 20, no. 4, pp. 473–509, 2010.

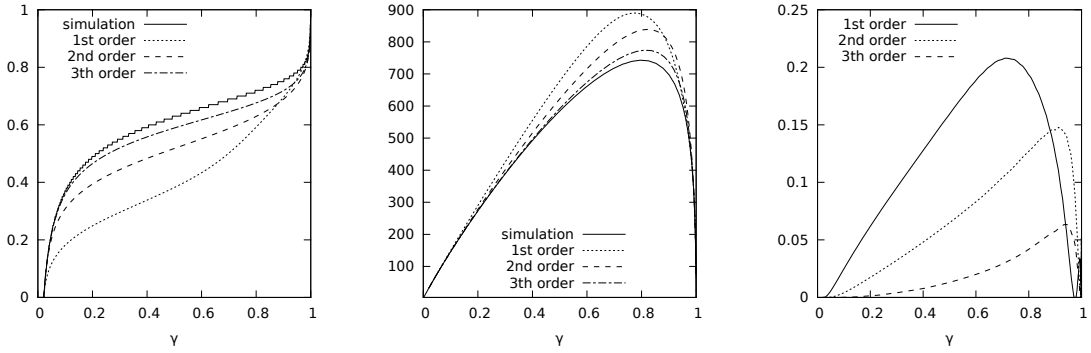
[15] H. Bruneel and B. G. Kim, *Discrete-time models for communication systems including ATM*. Kluwer Academic Publishers, 1992.

[16] S. Asmussen, "Stochastic simulation," 1999.

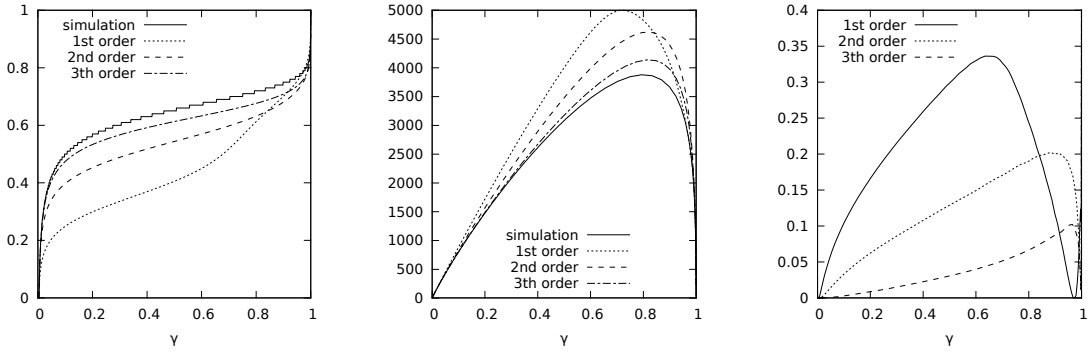
[17] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control*. Wiley-Interscience, 2005, vol. 65.



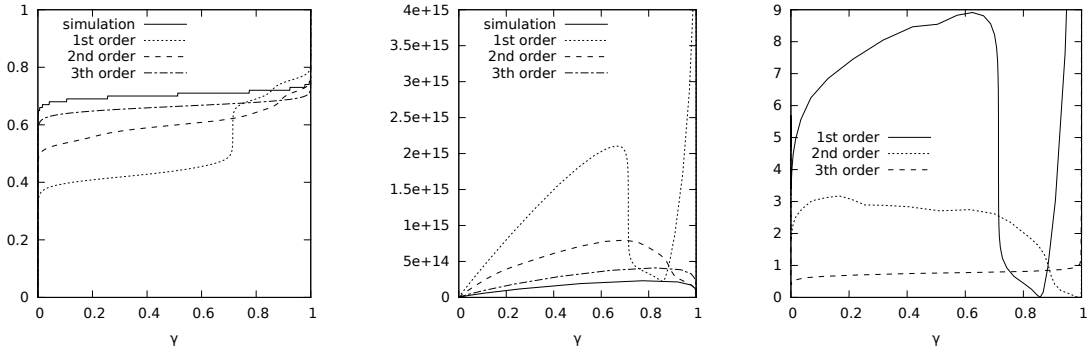
(a) $g(x) = x^3$



(b) $g(x) = x^4$



(c) $g(x) = x^5$



(d) $g(x) = x^{20}$

Fig. 7: $\beta_{\text{opt}}(\gamma)$ (left), real F when using approximation (center), fractional difference (right) for several cost functions. The arrival process uses the following parameters: $N = 16$, $\alpha = 0.8$, $\lambda_T = 0.9$

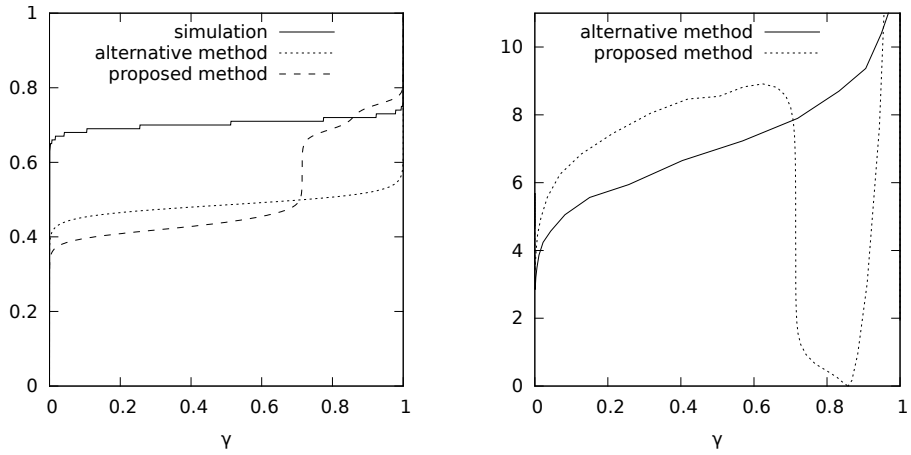
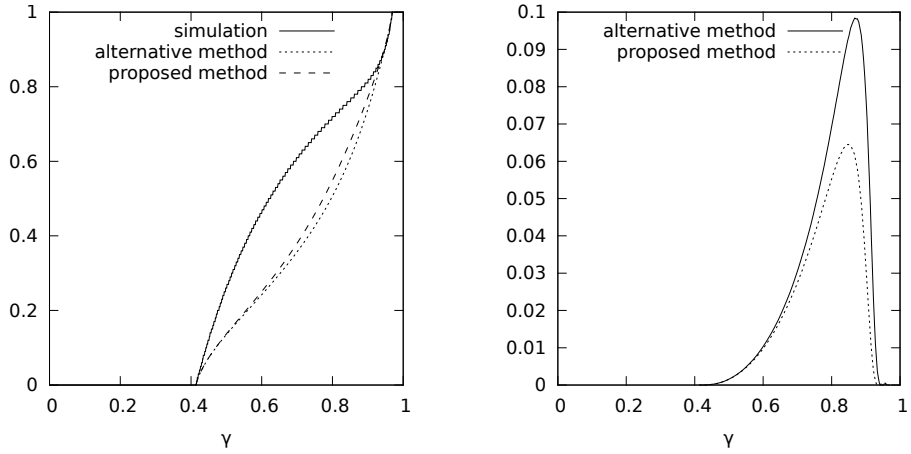
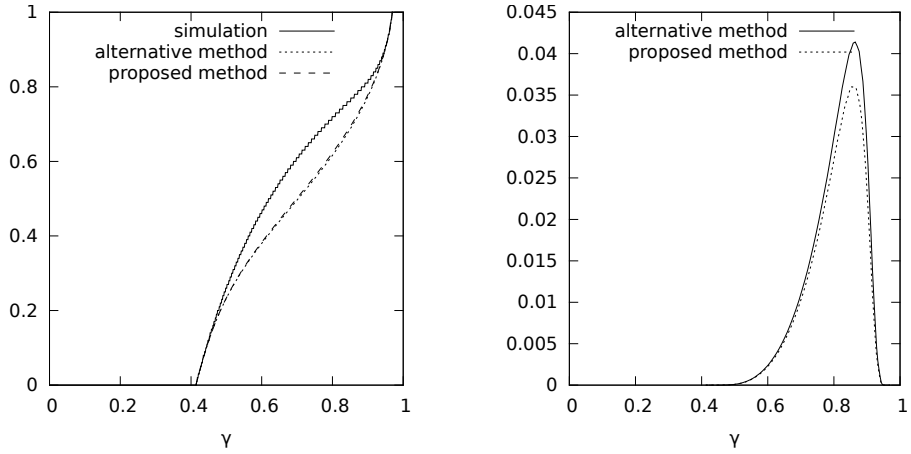


Fig. 8: Comparison of first order approximations of two methods: $\beta_{\text{opt}}(\gamma)$ (left), fractional cost difference (right) for $g(x) = x^{20}$. The arrival process uses the following parameters: $N = 16$, $\alpha = 0.8$, $\lambda_T = 0.9$



(a) First order approximation



(b) Second order approximation

Fig. 9: Comparison of two methods: $\beta_{\text{opt}}(\gamma)$ (left), fractional cost difference (right) for $g(x) = x^2$. The arrival process uses the following parameters: $N = 16$, $\alpha = 0.8$, $\lambda_T = 0.9$