Proxel-based simulation of queuing systems with attributed customers

Claudia Krull*, Wenjing Xu and Graham Horton

Department of Computer Science,
Otto-von-Guericke-University of Magdeburg,
Universitätsplatz 2, 39106, Magdeburg, Germany
E-mail: claudia.krull@ovgu.de
E-mail: wenjing_de@yahoo.com
E-mail: graham.horton@ovgu.de
*Corresponding author

Abstract: This paper describes a state space-based simulation method for queuing systems with attributed customers. Motivation was the need for exact solutions of queuing systems where no analytical solution is available. The original Proxel-based queuing simulation method is extended to incorporate attributed customers, concentrating on efficient coding and storage strategies to dampen state space explosion. The attributes classes priority, deadline and processing time are implemented and tested. Some interesting result statistics obtained would be hard to obtain using traditional simulation methods. The presented method can yield deterministic results for a larger number of queuing systems that cannot easily be solved analytically.

Keywords: state space-based simulation; proxel-based simulation; proxels; queuing analysis; queuing systems; multiclass queuing systems; attributed customers.

Reference to this paper should be made as follows: Krull, C., Xu, W. and Horton, G. (2010) 'Proxel-based simulation of queuing systems with attributed customers', *Int. J. Simulation and Process Modelling*, Vol. 6, No. 1, pp.50–57.

Biographical notes: Claudia Krull received her German Diploma in Computer Science from the Otto-von-Guericke-University Magdeburg, in 2003. She spent an exchange year at the University of Wisconsin, Stevens Point, where she received her Bachelor's Degree in 2002. She has been working at the Department of Computer Science since October 2003 as a research and teaching assistant in the field of Simulation and Modelling. In April 2008 she successfully defended her PhD Thesis at the Otto-von-Guericke-University Magdeburg, titled "Discrete-Time Markov Chains: Advanced Applications in Simulation". Her research interests include fast solution methods for Markov chains and their application in the analysis of more realistic models.

Wenjing Xu studied Computer Science at Hefei Technology University, China, from 2000 to 2004. For her Bachelor's thesis, she developed a student management system for the university. In October 2004 she began her "Computational Visualistics" Master's study in the Computer Science Department at the Otto-von-Guericke- University Magdeburg and her major is focused on the research field of computer graphics, image processing and simulation. Her master thesis topic was the application of Proxel-based algorithm to attributed queuing system. In April 2008 she received her Master's Degree and presently lives in Germany.

Graham Horton studied Computer Science at the University of Erlangen, and received his Master's Degree ('Diplom') in 1989. He received his PhD in Computer Science in 1991 and his 'Habilitation' in 1998 from the same university, in the field of simulation. Since 2001, he has been a Professor of Simulation and Modelling in the Computer Science Department of the Otto-von-Guericke-University of Magdeburg. His research interests include efficient solution algorithms for Markov chains, state space-based simulation methods and idea engineering.

1 Introduction

Queues are a part of our everyday life. Besides the obvious queues of human customers, they are widely used in telecommunications and computer systems. Ensuring traffic speed and reliability (quality of service) in such networks is a major objective as traffic volume and network size increase. Most real world systems deal with different types of data traffic and thus have different types of packages to handle, each with different properties, which are represented by attributes. Some network analysis methods exist that decompose the network into individual nodes (queuing models) and analyse their properties before aggregation. These single queuing systems can now be analysed using classical queuing theory, which is a well-researched area of mathematics. The usual approach to a queuing problem is to devise an appropriate formal representation, find a match among the already known classes of queuing systems or create a new one and analytically derive measures for the queuing system's performance.

For a number of classes of queuing systems, no analytical expression (or other readily available solution method) for performance measures like throughput or server utilisation is available. Examples for these hard problems are queuing systems with a limited calling source size. In such cases, simulation is an alternative way to tackle the problem. However, discrete event-based simulation may need to perform many replications to obtain estimates for the results of a specific queuing system. These results are often more expensive to obtain and less accurate than the analytical ones.

The recently developed method of Proxel-based simulation (Horton, 2002; Lazarova-Molnar, 2005) can feasibly conduct state space-based analysis of discrete stochastic systems and yields accurate deterministic results with reasonable effort. The Proxel results can compete with typical analytical ones, and the method is not inherently limited in the type of system it can simulate.

Proxels have already been successfully applied to the simulation of queuing systems with one type of customer (Krull and Horton, 2007). However, many queuing problems model so-called multiclass queuing systems; they implement attributed customers to enable more advanced queuing strategies than FIFO. The attribute processing time could for example be assigned to the customers upon their arrival to enable a Shortest Job First (SJF) queuing strategy, thus maximising the throughput of the system.

The object of this paper is to introduce attributes into a Proxel-based queuing system simulation, test the resulting performance and show the suitability of the approach. If successful, more classes of queuing problems can be tackled using Proxel-based simulation. We believe that this could provide useful help for queuing analysts. Proxel-based simulation can provide deterministic solutions for a specific queuing system's performance measures, when no analytical solution is available. One can obtain rough estimates at low cost or results of arbitrary accuracy at higher computation cost. Furthermore, the method's performance is not sensitive to the occurrence of rare events and it yields a transient solution of the system, which can for example help detect the possibility of infinite postponement when implementing SJF as a queuing strategy.

2 State-of-the-art

2.1 Queuing system analysis

Classical queuing analysis takes a real queuing system and derives a queuing model by identifying the arrival process, the service process and other system specifications. The performance measures of the queuing model are then calculated using known formulae for the given class of queuing systems. The performance measures of a queuing system are usually expressed as scalar measures such as the system throughput or the average waiting time of the customers (Bolch et al., 2006; Gross and Harris, 1998).

Queuing systems that contain generally distributed processes are often hard to handle analytically. For most of these classes no general solution exists. As an alternative, discrete event-based simulation can be used to obtain stochastic estimates for these systems performance measures. However, unless a lot of computation cost is invested, the accuracy of these results is not comparable to analytical solutions.

An important application area of queuing analysis is the modelling of data traffic in networks, for example, in the internet. The models often consist of multiple queuing systems that are connected to form queuing networks. As stated in Cremonesi et al. (2002) the traffic consists of customers with very different resource requirements (attributes), resulting in so-called multiclass queuing networks. The paper introduces a modelling framework for approximate solutions, because it is not feasible to derive exact solutions for problems with large state spaces. Exact solutions to multiclass queuing networks are only possible for special classes (Casale, 2006). These two examples show the importance of attributed customers, and also their impact on the complexity of exact solution methods.

One approach to the analysis of queuing networks is the decomposition into single queuing systems and the combination of their solutions. In Heindl (2001) one such approach is described for a special class of queuing systems. Another decomposition approach for multiclass queuing networks is described in Whitt (1994). The restriction of the decomposition approaches lies in the goal of describing the performance measure results using mathematical analysis. This makes special solutions necessary for the many classes of queuing systems. However, when exact solutions are required, decomposition is the only feasible way to reduce the model's state space to a controllable size. Therefore, a generally applicable and exact solution method for single queuing systems is still of interest. Such a method is the long term goal of the current research.

2.2 Proxel-based queuing system simulation

Discrete event-based simulation is one way to derive estimates for the scalar performance measures, if no analytical solution exists. However, the results are of stochastic nature and only applicable to the specific queuing system investigated. The simulation itself can get very expensive for stiff queuing models.

Hasslinger and Kempken (2006) present an approach for the transient analysis of a queuing system. Transition equations for discrete points in time are derived and only the relevant state transitions at arrival and service instances are considered. The approach seems promising, and according to the authors it can be extended to a number of queuing problems. However, the example in the paper is small, considering only two discretisation points for the arrival and service distributions. The transition equations and the system state space can get complex when the problem gets larger, since all possible combinations of the arrival and service of a customer have to be considered.

The Proxel-based simulation of queuing systems proposed in Krull and Horton (2007) has several advantages. It can yield deterministic results for the performance measures of a given queuing system. The accuracy of the results can be controlled. In addition to the steady state performance measures, the Proxel method also yields a transient solution, containing the probability of every possible system state at each simulation time step investigated. This cannot be easily obtained using common simulation techniques.

Proxel-based simulation is a state space-based simulation method that scans all possible system development paths in discrete time steps (Horton, 2002; Lazarova-Molnar, 2005). In contrast to DES it follows all possible development paths, tracking their respective probabilities and creating a discrete-time Markov chain. This method ensures the discovery of rare development paths. The disadvantage of Proxels is that through expanding the discrete system states by supplementary age variables, the resulting Markov chain can be of immense size. This so-called state space explosion limits the applicability of Proxels to models with few discrete states. However, queuing models usually have a small discrete state space and seem a feasible model class.

The current implementation of the Proxel-based queuing system simulator is restricted to customers without attributes. A queuing Proxel (probability element) includes the following information: number of customers in the queue q, number of customers in service s, age of the arrival process τq, age of the service processes τs and the probability of that combination p.

$$\text{Proxel} = (q, s, \tau_q, \vec{\tau}_s, p).$$

The special purpose implementation enabled the specialisation of the arrival and service processes in separate methods. This speeds up the computation, since the queuing order only needs to be checked and possibly changed upon the arrival of a customer. The service process method only removes the first customer in the queue and starts the service process.

The restriction to only one class of customers limits the possible queuing strategies to FIFO and possibly other simple queuing strategies. However, many real systems use more sophisticated strategies for sorting their customers. To enable the Proxel-based simulation of more realistic queuing models, attributed customers are essential.

The basic idea of the extension to attributed customers and one example attribute implementation have been described in Xu (2008). The current paper attempts to generalise these results and conducts tests to gain an insight into the performance and the applicability of the method.

3 Adding attributed customers

This section describes the steps which are necessary to include attributed customers in the Proxel-based simulation of queuing systems.

In this paper, we selected three representative static attributes without pre-emption policy for implementation. Adding variable attributes or pre-emption would further enlarge the state space. We also decided to limit the approach to queuing strategies with only one attribute, again to dampen state space explosion.

3.1 Attribute choice

Common queuing strategies often sort the arriving customers by intrinsic priorities or some given time restrictions. This in mind, the following three attributes were chosen for implementation and test of the method: *priority*, *processing time* and *deadline*.

A customer's priority is a given measure of urgency or importance. It is usually represented by an integer where, in our case, a smaller number implies a higher priority.

A customer's processing time holds the information about how long the servicing of the customer will take. The processing time is usually given by a real value and can be distributed randomly. This real valued processing time can be converted into an integer by discretising it using the Proxel simulation time step Δt. Since the discretisation of time is already a necessary part of the Proxel algorithm, we do not lose information by also discretising the processing time. At the same time, this greatly reduces the possible system state space and thereby memory and computation cost.

The attribute deadline is a measure of urgency, which is again given by a real value and can be randomly distributed. The deadline can also be expressed as the difference between the current simulation time and the customer's deadline. Doing this limits the attribute's value set and thereby reduces the system state space. The 'time to deadline' can also be converted to an integer by discretising this distance using the discrete simulation time step Δt, analogous to the processing time.

3.2 Coding and storage strategies

As described above, all three attributes can be expressed as integer values. The processing time and time to deadline can have potentially large value sets depending on the discretisation time step Δt and the support of the distribution. The value set of the attributes will have a great influence on the number of discrete states the system can be in and therefore also affect the algorithm's performance. In order to include the attribute information the scalar numbers for customers in queue and service have to be replaced by integer arrays holding the attribute information.

$$\text{Proxel} = (\vec{q}, \vec{s}, \tau_q, \vec{\tau}_s, p).$$

By adding attributes to the customers in the queuing system, the discrete state space of the model will be increased significantly. This has two effects. First of all, the memory requirement of the algorithm will grow, because the number of Proxels increases with a larger state space and the size of a Proxel grows when one replaces a scalar queue length by an integer array holding the queued customers' attributes. Secondly, the computational bottleneck of the Proxel-based method is the retrieval of already created Proxels. A newly created Proxel has to be compared to a possibly large number of existing Proxels in the storage container, requiring a comparison of all elements of these Proxels including discrete state space and age vector. Therefore, a clever storage of Proxels and the connected discrete system states is necessary to counter these effects.

The attribute of a customer in service can be stored in an array the size of the number of servers. The attributes of the customers in the queue are also stored in an integer array with a maximum length given by the implementation, which will most likely not be reached. After testing several options (Xu, 2008), this has been shown to be the most efficient storage strategy for saving memory and retrieval time. The queue array contains the attributes of the customers in the order that they have in the queue.

As a further enhancement, the customer priority can also be coded by listing the number of customers of each priority class in the array instead of enumerating the whole queue. Thereby, the size of the array can be limited to the number of priority classes, which is most likely less than the maximum queue length; e.g., the queue content '111222233555' could be expressed by '34203'. This saves memory and also decreases the time needed to compare two Proxels in the retrieval bottleneck.

Processing time and time to deadline need to be enumerated for all queued customers, since we cannot expect as many customers with the same values.

3.3 Adapting the algorithm

The Proxel-based queuing system simulation algorithm (Krull and Horton, 2007) needs to be adapted slightly to the enhanced definition of a queuing Proxel.

The attributes of the customers in the queuing system are fixed upon their arrival. Therefore, the formerly single arrival Proxel will now be split into as many Proxels as possible values of the chosen attribute (e.g., number of priority classes, each possible discretised service time). In each of these new arrival Proxels, the queue has to be reordered, meaning the attribute of the arriving customer has to be inserted into the queue array at the appropriate position. This step needs to be embedded in the existing arrival procedure.

The time to deadline and priority attribute have no influence on the processing of the customer. However, fixing the processing time of a customer upon arrival turns the actual servicing into a deterministic process. This reduces the number of possible service process development paths to one, reducing the overall system state space. The appropriate handling is also embedded in the existing service procedure, not affecting the other algorithm elements. This decrease in expanded system states might compensate some of the increase caused by splitting each arrival Proxel.

The time to deadline is an integer multiple of the discretisation time step and is assigned upon the arrival of a customer. As the simulation time advances, the deadline approaches and the time to deadline decreases. This is realised by decrementing the time to deadline for each customer in the system during the processing of a Proxel.

3.4 Adapting the interface

The interface of the Proxel-based queuing simulation tool has to be changed to enable the selection and specification of customer attributes. In the adapted interface the user can choose to include customer attributes in the queuing system. A separate dialogue enables a selection of the attributes. The number of priority levels and a probability distribution for the time to deadline can also be specified. The specification of the service time distribution is done through the main interface.

The output part of the interface was not adapted to include statistics of attributed customers. It only holds the overall system performance measures. Due to complexity, the measures calculated with regard to the attributes (see next subsection) are put out into files for later external analysis.

3.5 Calculating performance measures

The inclusion of customer attributes into the queuing system specification enables the calculation of more sophisticated performance measures. Examples of newly possible measures are the following:

- *Priority attribute*: One can determine the average and extreme waiting times for each priority class from the Proxel simulation result. This could, for example, help decide whether low priority customers have to wait too long and should be treated otherwise.
- *Deadline attribute*: One can determine the number of late customers and also the cumulative lateness of all customers processed from the Proxel simulation result. This can yield information on the cost induced by customers processed after their deadline.
- *Processing time attribute*: One can determine the minimum and maximum waiting time for each possible processing time from the Proxel simulation result. This might help to detect infinite postponement, which can happen in SJF scheduling environments.

All of these performance measures can be calculated on the fly or using the steady state result of the Proxel algorithm. Their calculation requires only constant effort and does, therefore, not influence the algorithm's performance significantly. The next section will show results on the algorithm's performance and also on performance measure results obtained using attributed customers.

4 Experimental results

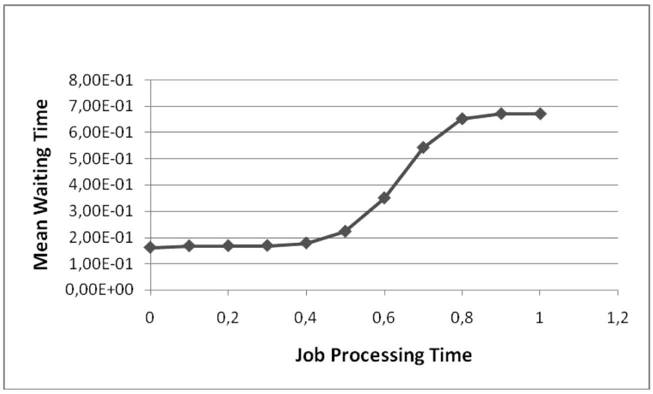
This section describes some experiments conducted to show that the inclusion of attributes into Proxel-based queuing system simulation is possible, and to test the effect of adding attributes on the algorithm's performance. Furthermore, we want to show some interesting results that can only be obtained by including attributes in the Proxel-based queuing system simulation.

4.1 Experiment on Shortest Job First strategy

The first experiment simulates a queuing system using the SJF scheduling strategy. The investigated queuing system has a Markovian arrival process with a rate of one customer per minute and a normally distributed service process with the parameters $\mu = 0.5$ and $\sigma = 0.1$.

As an interesting result measure, we obtain a distribution of the waiting time over the possible processing time values. This result is not easily computable using existing queuing analysis or simulation methods. Figure 1 shows the average waiting time for the different discretised values of the service time distribution. As expected, the waiting time of customers with short processing times is short, and the waiting time of customers with larger processing times is longer. The steeper increase around the expected value of the service time is due to the higher number of customers created. These customers with equal discrete processing time delay each other when they are ordered according to FIFO.

Figure 1 Waiting time distribution over processing time

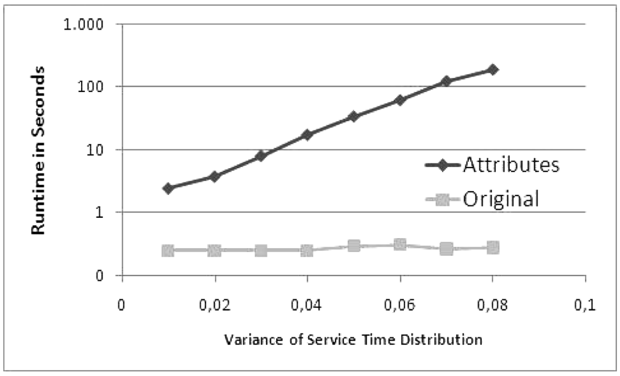


4.2 Investigating the effect of different processing time variances

The second experiment investigates the effect of adding different processing time distributions on the performance of the Proxel-based simulation method. The arrival process is set to be Markovian with a rate of one customer per minute. The service process is normally distributed with a mean of 0.5 and the standard deviation is varied from 0.01 to 0.08. This results in 2 to 12 discretisation time steps of the distributions as possible attribute values. Proxel simulations are performed using the same processing time distribution with and without explicit customer attributes.

Figure 2 shows the development of the runtime for the different service time distributions. The runtime of the algorithm without attributed customers (grey line) is hardly affected by the change in service time distribution. The runtime of the simulation incorporating attributed customers (black line) rises from two seconds to several hundred. The increasing number of discretisation time steps of the service time distribution increases the attributes value set and thereby increases the system state space already upon customer creation. The algorithm without customer attributes (Krull and Horton, 2007) splits the possible different service time values by successively reducing the probability of the Proxel representing the servicing of a customer. This experiment shows that adding the attribute service time to the Proxel simulation increases the complexity drastically. The combinatorial effect of enumerating every possible combination of customers waiting in the queue enlarges the state space by several orders of magnitude. The effect is more severe when a larger number of attribute values are possible. Therefore, only systems with small service time variances can be simulated feasibly.

Figure 2 Algorithm runtime over service time distribution with and without attribute



4.3 Investigating the effect of the number of priority classes

The third experiment investigates the effect on computation cost when adding the attribute priority and increasing the number of priority classes. The tested queuing system has, again, a Markovian arrival process with an arrival rate of one customer per minute and one server. The service time distribution was chosen to have a rate of two customers per minute. Two different distributions were tested for the service time: Exp($\lambda = 2$) and $N(\mu = 0.5, \sigma = 0.1)$. We want to investigate the effect of the attributes on different queuing systems and compare the storage strategies enumeration of priority classes and count of elements per class.

The resulting number of concurrent Proxel elements (as a measure of memory requirement) was the same for both experiments. Compared to enumeration, the overall storage space needed could be reduced to about one half by using the counting strategy, since the storage space needed by one Proxel was reduced. The development of the algorithm runtime for the M/M/1 queuing system with a growing number of priority classes can be seen in Figure 3 (logarithmic scaling). The grey line shows the experiment using enumeration of all queued customers and the black line shows the development for counting the elements per priority class. The drastic effect of this small change in the storage scheme is surprising. This shows that the handling of the Proxels and, especially, the comparison of two individual probability elements is one major bottleneck of the algorithm. When the attributes of all queue elements are stored, these also need to be compared individually when two Proxels are compared. This overhead can be reduced when only the number of elements per priority class needs to be compared. The same effect is also visible in Figure 4, where the runtime for the normal service time distribution is depicted.

Figure 3 Algorithm runtime for exponential service time distribution and different numbers of priority classes

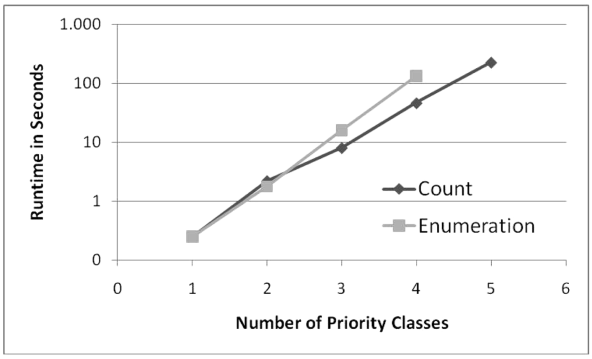


Figure 4 Algorithm runtime for normal service time distribution and different numbers of priority classes

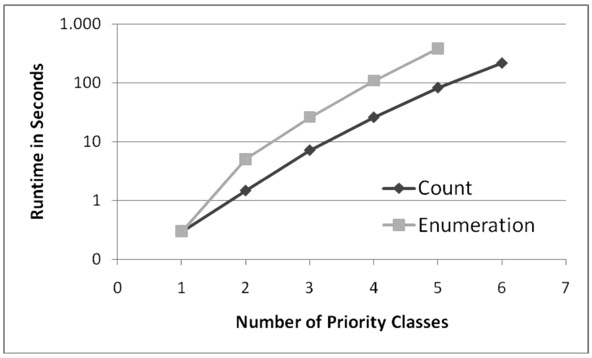
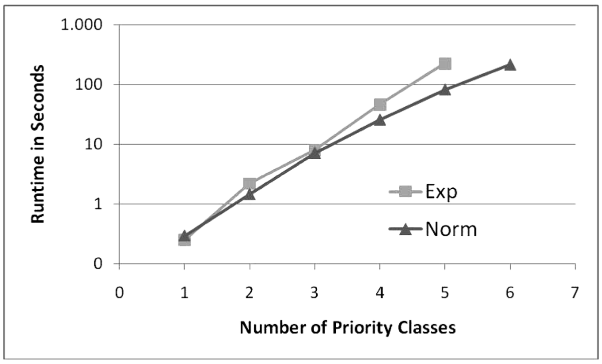


Figure 5 shows the runtime of the algorithm using attributed customers with normally distributed and exponential service time distributions. One can see that using normally distributed service times, the runtime is faster than when using exponential service times. This behaviour contrasts with that of the original Proxel algorithm, where exponential distributions do not expand the state space due to their constant rate function. The greater variance of the exponential distribution causes a larger possible queue length, resulting in a larger state space and higher runtime costs. This experiment shows that the increase in state space by adding age variables for a non-Markovian distribution is less severe than the combinatorial explosion caused by the greater queue length and the large variance of the exponential distribution.

Figure 5 Algorithm runtime for different service time distributions with priority attribute



Therefore, the effect of adding attributes to a queuing system is not so much dependent on the original state space of the system, but rather on the possible maximum number of customers in the system, which influences the number of possible combinations of customers' attributes.

4.4 *Investigating the effect of different deadline distributions*

The last experiment shows the effect of different deadline distributions on the computation cost of the method. The queuing system used has normally distributed inter-arrival times (~$N(0.7, 0.2)$) and normally distributed service times (~$N(0.5, 0.1)$). Figure 6 shows the development of the algorithm runtime using different deadline distributions with a mean of 0.5 and with the number of possible time-to-deadline values varying from 2 to 12. The initial runtime of 30 s is already large considering a maximum possible queue length of 5, but the runtime increase for a supposedly larger value set is not that severe. This happens because the value of the attribute time to deadline is not static. In the course of the simulation, the value is updated by decreasing it in every time step, so that it reaches zero when the deadline approaches, afterwards it becomes negative. Therefore, the actual value space of the attribute deadline ranges from the maximum distribution value to the longest delay and is larger than the initial number of discretised distribution values. In our experiment, the maximum delay was 1.7 min, which increases the maximum number of attribute values to almost 30 in the worst case. Figure 7 shows the throughput for the different final delays. The most frequent attribute values can be found around zero, as is expected when the service time and deadline distribution have the same mean. All other classes between –1.7 and 1.1 have much lower frequencies.

Figure 6 Algorithm runtime for different deadline distributions

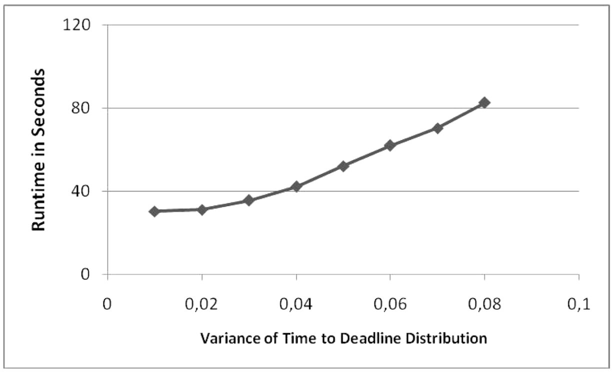
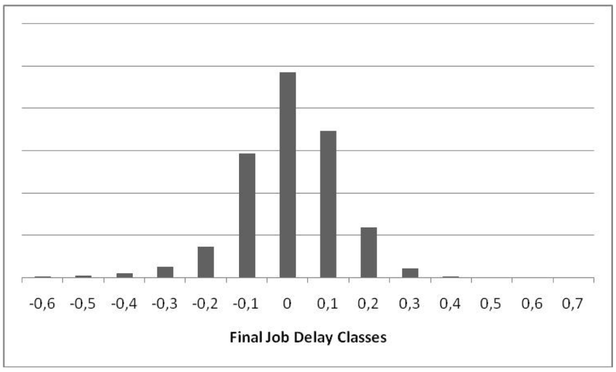


Figure 7 Throughput of customers with different delays



4.5 Result discussion

All experiments show a significant increase in memory and runtime cost due to adding attributes, as can be expected. However, clever storage and coding strategies help reduce these costs. The number of different attribute values that can be added feasibly differs between the attributes. The experiments suggest that the maximum number of priority classes is about 10, also depending on the maximum possible number of customers in the system. The discrete time steps of a distribution can be up to 20 or more, when the extreme values are less frequent than the average ones, such as when using a normal distribution. This happens because the simulation algorithm truncates Proxels below a certain probability threshold to limit the computation complexity.

Overall, the inclusion of attributes into Proxel-based queuing system simulation is feasible for attributes with a small value set or a small number of frequent attribute values.

5 Conclusion and outlook

The paper described how Proxel-based queuing system simulation can be improved by adding attributes to the customers. This increases the number of possible applications to multiclass queuing systems and the realism of the simulated queuing systems. Limiting the approach initially to only one static attribute was necessary to keep the model's size in check. Furthermore, efficient storage strategies are essential for handling the state space explosion.

As expected, adding attributes leads to an increase in computation cost. This currently limits the applicability to queuing systems with general distribution functions with small variances or a low mean value. The experiments suggest that adding one attribute with a small value set is feasible. Only then can the method be applied feasibly using the current implementation.

The developed approach can be useful to queuing analysts, if no analytical solution is available for a queuing system. Especially, rough estimates of the performance measures can be obtained very fast. By including attributes, interesting results can be obtained such as the distribution of waiting time over the possible values of processing time.

Since we were able to implement the attribute processing time at an acceptable cost, one future task is to implement round-robin as a queuing strategy. Implementing multiple attributes per customer is possible, but will only be feasible in conjunction with even more efficient storage strategies.

The long-term goal of this research is to provide a state space-based method for the analysis of queuing systems and networks. These networks should be decomposed and then the components should be analysed using Proxel-based simulation. The combination of the results is still a subject of future work. We think that the output of one queuing system could be easily transferred into the input of another queuing system. The resulting method could help network and queuing analysts in the absence of analytical solutions to get reasonably accurate results without having to employ discrete event-based simulation.

References

Bolch, G., Greiner, S., de Meer, H. and Trivedi, K.S. (2006) *Queueing Networks and Markov Chains*, 2nd ed., John Wiley & Sons, Hoboken, New York.

Casale, G. (2006) 'An efficient algorithm for the exact analysis of multiclass queueing networks with large population sizes', *Performance Evaluation Review*, Vol. 34, No. 1, pp.169–180.

Cremonesi, P., Schweitzer, P.J. and Serazzi, G. (2002) 'A unifying framework for the approximate solution of closed multiclass queuing networks', *IEEE Transactions on Computers*, Vol. 51, No. 12, pp.1423–1434.

Gross, D. and Harris, C.M. (1998) *Fundamentals of Queueing Theory*, 3rd ed., John Wiley & Sons, New York.

Hasslinger, G. and Kempken, S. (2006) 'Transient analysis of a single server system in a compact state space', *Proceedings of 13th International Conference on Analytical and Stochastic Modelling Techniques and Applications*, European Council for Modelling and Simulation, Bonn, Germany, pp.91–96.

Heindl, A. (2001) 'Decomposition of general tandem queueing networks with MMPP input', *Performance Evaluation*, Vol. 44, Nos. 1–4, pp.5–23.

Horton, G. (2002). 'A new paradigm for the numerical simulation of stochastic Petri nets with general firing times', *Proceedings of the European Simulation Symposium 2002*, SCS European Publishing House, Dresden, Germany, pp.129–136.

Krull, C. and Horton, G. (2007) 'Application of proxels to queuing simulation', *Proceedings of Simulation and Visualization 2007*, Magdeburg, Germany, pp.299–310.

Lazarova-Molnar, S. (2005) *The Proxel-Based Method: Formalisation, Analysis and Applications*, PhD Thesis, Otto-von-Guericke-University Magdeburg, Germany.

Whitt, W. (1994) 'Towards better multi-class parametric-decomposition approximations for open queueing networks', *Annals of Operations Research*, Springer Netherlands, Vol. 48, No. 3, pp.221–248.

Xu, W. (2008) *Application of Proxels to Queuing Simulation with Attributed Jobs*, Masters Thesis, Computer Science Department, Otto-von-Guericke-University Magdeburg, Germany.