Controlling the supermarket service

José Antonio Montero and L. Enrique Sucar

[1]Instituto Tecnologico de Acapulco, Av. Inst. Tecnológico S/N, Acapulco, Mex. ,
[2]Inst. Nacional de Astrofisca, Optica y Electronica, Luis Enrique Erro #1
Tonantzintla, Puebla, México,
jamontero1@prodigy.net.mx, esucar@inaoep.mx
(Paper received on September 11, 2010, accepted on October 20, 2010)

Abstract. This work presents an approach that combines queueing the-ory and decision theory to optimize the number of open cashiers in a supermarket. The setting is a simulated supermarket service controlled by policies generated by Factored MDPs to minimize the long-term cost function. Preliminaries tests demostrates that the performance of the simulated model is according with the mathematical counterpart, and the actions selected by the controller are according to the optimal pol-icy.

Keywords: Decision-theoretic systems, Factored Markov decision pro-cess, Supermarket queue control.

1 Introduction

Currently, supermarkets understand that the overall service is the key for suc-cess, and service quality is the key for winning the competition in supermarkets with similar quality and price. For supermarkets, more cashiers mean more in-vestments, however, few cashiers may lead to serious waiting, a ecting service quality and causing loss of costumers. Some times, managers make decisions that not always result in the best option for the business e ciency, so the design of automatic systems that supports management decisions are of great help. In this work, we present the design of a simulator for the queue supermarket joined with a controller based on Factored Markov Decision Process (FMDPs) [1], for optimizing the supermarket service. The designed system suggests the optimal number of cashiers in a planning period, this decision is based on minimum cost approach. For developing the proposed system we integrate aspects of queueing theory and decision theory. In particular, queueing theory addresses problems related to the optimal design and control of queues, the goal is to find optimal values for parameters which, once determined, become fixed characteristics of the queueing system, such as the maximum number of allowed customers [2], maximum waiting time for customers in queue [2] or the total number of avail-able servers [3]. Problems dealing with queue (service) control are dynamic, the goal is to find the optimal operating policy, that is, rules for turning the server in occupied and idle, that result in the lowest long-run cost [4]. Then, this mod-els determine an optimal action to take when the queue is in a particular state.

Special Issue in Advances in Artificial Intelligence and Applications
Research in Computing Science 51, 2010, pp. 49-56

[5], [6]. Some previous works also proposed solutions to problems dealing with optimal queue design and queue control [7].

The problem we address in this paper[3] deals with both: queue design and service control, since we would like to determine, based on minimum total cost (cashier cost + waiting cost), the optimal number of cashiers that should be attending in the supermarket and how this cashiers should be selected during a planning horizon. For reaching this, we structured our work as follows: (i) using the queueing theory and based on [8], we designed a simulation model for the supermarket queue system with C servers, then we simulated the behaviour of the parameters (arrival rate and service rate) that occur when customers are waiting for service in the supermarket queue, and (ii) based on the FMDPs framework and using the parameters values of the supermarket queue model previously simulated, we obtain the policy for the controller. The optimal policy is constructed using Spudd [9] and the reward is based on a measurement of minimum cost reached during the planning period.

Preliminary tests indicate that selected actions by the controller are accord-ing to the observed states in the supermarket queue model.

2 Simulation of supermarket service

Computer simulation is a quite e ective way of analyzing the performance of queuing systems. For some aspects of the simulator, we performed a modifica-tion and extension to the simulation computer program for M/M/1 model by Law and Kelton [8]. Particulary we replaced the random number generator by another appropriate to our aim[4], also we added an interface to allow the interac-tion with users, after we included multiple threads to simulate multiple M/M/1 queues. In this work we are interested in how obtain the total cost (cashier cost + waiting cost). The waiting time cost is not easy to obtain, but there are works related with its study [6], [7] that give some ideas. In the proposed model, the simulation of multiple servers is carried out by varying the value of the arrival rate ( ) according to the number of cashiers attending, and maintaining con-stant the service ratio ( ), this is a valid assumption when there are not many interchanges of customers between the rows in the queue. Then, the performance measurements[5] used in this work correspond to the M/M/1 model [5].

---

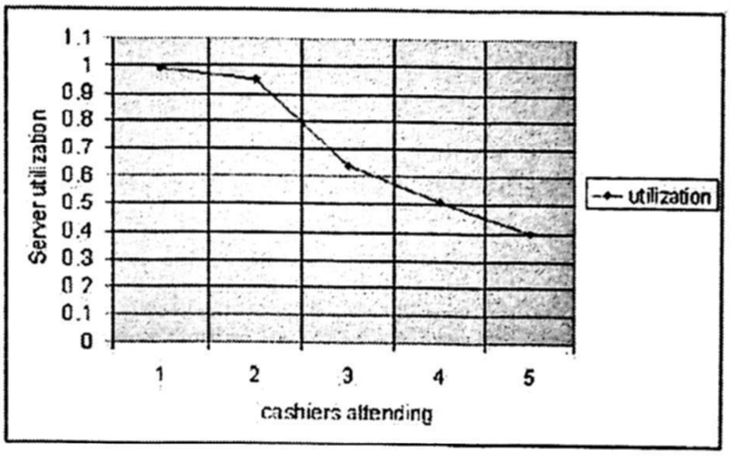
3 For a full description of this work download the technical report at: http://ccc.inaoep.mx/publicaciones/reportes tecnicos/CCC-10-006.pdf

4 We are using the Mersenne Twister [10] that generates periods long enough for many practical applications.

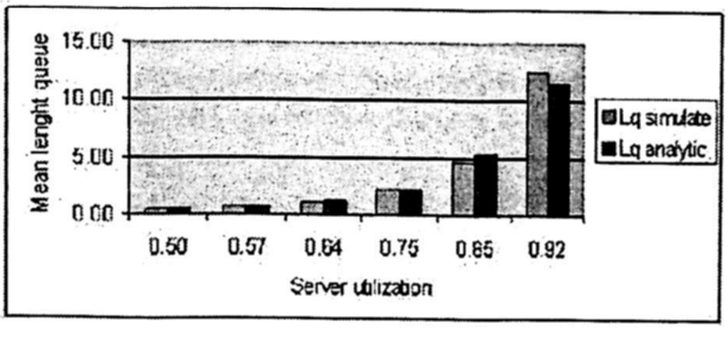
5 Server utilization ( ), average waitig time that customers spent in the queue (W_q)and the average lenght in the queue (L_q)

2.1 Results from the simulator

The simulator was tested varying the rate of arrivals ( ) and the rate of service ( ). In Figure 1 (a), we observe the variation between the server utilization and the cashiers attending (service rate). When there is only one cashier attending, the system crashes, if one cashier is added the server utilization is 95 percent (the queue may soon collapse under this condition), if a third cashier is added, the server utilization is 64 percent, and the system is under stable conditions. In another test, we realize the comparation between the simulated model $M{=}M{=}1$ and the analytical model $M{=}M{=}1$ (see Figure 1). The parameters used in this test are: (*a*) the value for the arrival rate is 60 customers per minute and is maintained fixed, (*b*) the values for the service rate are varying with the follow-ing values $= f120;\ 105;\ 90;\ 80;\ 70;\ 65g$ customers per hour. Results obtained for both models exhibited a variation ranked from 4% to 14%. Then, we conclude that results generated by the designed simulator are reliable.



(a)



(b)

***Fig. 1.** Operation of the simulator: (a) Behaviour of the server utilization and the number of actives cashiers C when arrival stream is bigger than service rate, (b) Com-paration between analytical results and simulated results for the queue service model.*

3 SuperMarket queue controller

Determining the optimal number of cashiers is one of the problems that addresses queueing theory. Basically, there are two approaches to solve this problem: (i) Decision cost-based model and (ii) Decision accept-based model [5]. In this work we applied the last approach, in which, the goal is to balance conflicting measurements, like the average waiting time (W_q) and idle server time (X). Generally the values for this variables are adjusted using a subjetive approach for each particular application [5]. In this work, the value for W_q is determined by the simulator queue model under steady operation conditions (<) . The value obtained for W_q variable must be less than 7 minutes (value obtained when the server utilization value is 81%). To determine the idle time value, some re-searchers [5], [8] suggest that optimal values for idle time server are ranking from 20% to 30%. In this work, we consider interval values computed by the simulator of the supermarket queue model under steady operation, then com-puted values for the X variable are ranking from 19% to 33% (very similar with the suggested values). However, it is not always possible to match these values, then the management must to decide the best action. For example, Figure 2 illustrates values for W_q and X variables, when the value of the arrival rate ( ) and service rate ( ) are varying. Taking into account the values observed in this Figure, the optimal number for cashiers are indicated by the shadowed areas. This means, that in the first row the average customer waiting time is 5.43 min-utes (less than 7 minutes) and the idle server time is 18.6% (very close to 19%), then the optimal number for cashier is 1, in the second row the values of waiting time and idle server time suggest that 2 cashiers are the optimal number under this operation conditions.

Next, we are interested in determining the optimal number of cashiers during a planning horizon (what action should be taken in a particular state of the queue), for getting it, we construct an optimal policy using Factored MDP's.

3.1 MDP for queue control

The policy for the supermarket service queue is defined using an MDP frame-work [11] . A Markov decision process (MDP) is a mathematical framework for sequential decision making problems in stochastic domains. The solution of a MDP provides an optimal policy, a decision rule for each decision time point of the process that optimizes the performance of the system measured by a utility function. This function assigns to each policy, a value according to optimal-ity criteria, which generally is a metrics of the total expected reward over an horizon (finite or infinite). Classical dynamic programming solving methods for MDPs present a big problem: they grow up exponentially when the number of domain features increasing, so, a considerable e ort has been devoted to devel-oping representational methods for MDPs that obviate the need to enumerate the state space, like abstraction techniques [12] and factorized representations [1]. FMDPs [9], [1] allow us to represent complex uncertain dynamic systems

	Cashiers attending					
	1	2	3	4	5	parameter values
Wq	5.43	0.88	0.27	0.26	0.19	μ=65, λ=60
X	18.6	54	73	77	83	
Wq	41.17	1.12	0.57	0.38	0.21	μ=65, λ=70
X	2	48	67	73	81	
Wq	43.5	1.9	0.78	0.56	0.38	μ=50, λ=60
X	1	35	63	71	78	
Wq	52.7	3.74	1.36	0.73	0.47	μ=40, λ=60
X	0.73	28	49	76	77	
Wq	59.7	18	1.85	0.97	0.63	μ=60, =120
X	0.17	6	34	47	63	

Fig. 2. *Illustration of the optimal cashiers attending, the parameter values are deter-minated by the simulator under steady operation conditions, recommended values in each case are appearing by shadowed boxes.*

very compactly by exploiting the problem specific structure. Specifically, the state of the system is described by a set of variables that evolve stochastically over time using a compact representation called a dynamic Bayesian network (DBN) [13].

We defined a FMDP model for the supermarket queues including states, ac-tions and rewards; which are detailed below.

States. The state space in the Supermarket queue model is characterized by three variables. Next, the interpretation of each one is presented:

nca (number of attending cashiers). The valid values[6] for this state variable were obtained from the simulator under steady operation and considering the goals of the designed system: (i) optimizing the number of cashiers and (ii) op-timizing the waiting time (see Figure 2). The optimal values for the *low* interval values were obtained from the variable, then for the interval [0:84 0:91], means that the server utilization is ranked from 84% to 91%.

tep (average waiting time). The valid values for this state variable under steady operation conditions are: *low* [1:8 3:8], *normal* [3:9 5:9] and, *high* [6:0 8:0].

npc (number of customers in the queue). The valid values for this state vari-able under steady operation conditions are: *low* [1:13 2:83], *normal* [2:84 4:54] and, *high* [4:55 6:25].

6 The unit considered for this variable is a real number indicating the number of cashiers attending.

Actions. In this work we are using an DBN approach to represent actions. At the moment, the actions defined in the MDP model are:

Continue action. The controller selects this action, when the operation of queue supermarket system is in stable conditions. Its e ect on the system is like a null action.

Add a cashier action. When the controller is observing during the planning period that the operation model is under unestable conditions, this means: (i) the average waiting time in queue is *high* is over the *normal* expected value , (ii) the expected number of customers in queue override the *normal* operation conditions, and (iii) the rate of performance is high -over 90%-, then the con-troller indicates that appropriate action is *Add a cashier*. Generally this action is sugested when during the planning horizon the arrival rate is bigger than the service rate, and maintained under that condition during several planning period.

Eliminate a cashier action. This action is suggested when the operation con-dition reported by the queue supermarket simulator is below the *normal* opera-tion or steady condition system is operating with many cashiers .

Rewards. In this work, the reward function represents a compromise be-tween the goal of the supermarket management (optimal cashier number with minimum cost) and the goal of the customers (minimum waiting time). These goals are reached based on the: (i) optimal number of cashiers using an accept-cost model approach (see Figure 2) and, (ii) optimal values for the queue param-eters (*nca, npc, tep*) based on the steady operation conditions of the simulated supermarket queue model (see Figure 3). *Normal* condition of operation is the combination of the two goals.

To solve the FMDP we have used the SPUDD (stochastic planning using decision diagrams) system. SPUDD implements classical value iteration, and uses ADDs to represent value functions and CPTs, this often yields substantial savings in both space and computational time. Figure 4 illustrates the policy obtained with Spudd and that our controller uses to select the best action.

4 Conclusions and future work

We presented the design of a simulator for the queue supermarket combined with a controller based on MDPs. The designed system selects the optimal num-ber of cashiers in a planning period, this decision is based on a minimum cost approach. For developing the proposed system we integrated two powerful tech-niques: queueing theory and decision theory. We provided experimental evidence illustrating that the performance of the simulated model was according with the mathematical counterpart, and the actions selected by the controller were ac-cording the optimal policy constructed by Spudd. FMDPs is an approach to derive an optimal control policy to maximize the expected utility for a planning period, this is a useful characteristic for a variety of industrial applications, like: machine maintenance, inventory control, plant operation, among others.

	npc			*nca*			*tep*		
	low	*normal*	*high*	*low*	*normal*	*high*	*low*	*normal*	*high*
Elimina-tes a cashier	0.7	0.3	0.0	0.0	0.3	0.7	0.7	0.3	0.0
Conti-nue	0.5	0.5	0.0	0.5	0.5	0.0	0.5	0.5	0.0
Add a cashier	0.0	0.3	0.7	0.7	0.3	0.0	0.0	0.3	0.7

Fig. 3. *Adequate balance between optimal number of cashiers and minimum waiting time. Customers preferences are minimum waiting time: then, many cashiers attend-ing. Supermarket preferences are: minimum cost, but, few cashiers attending. Then, to equilibrate these preferences, and to maintain the operation system under steady operation conditions, prefered action is Continue and prefered value for the state vari-ables is Normal. Prefered actions are represented by a value indicating an associate cost value in the interval 0 1, where 0 value indicates minimum preference.*

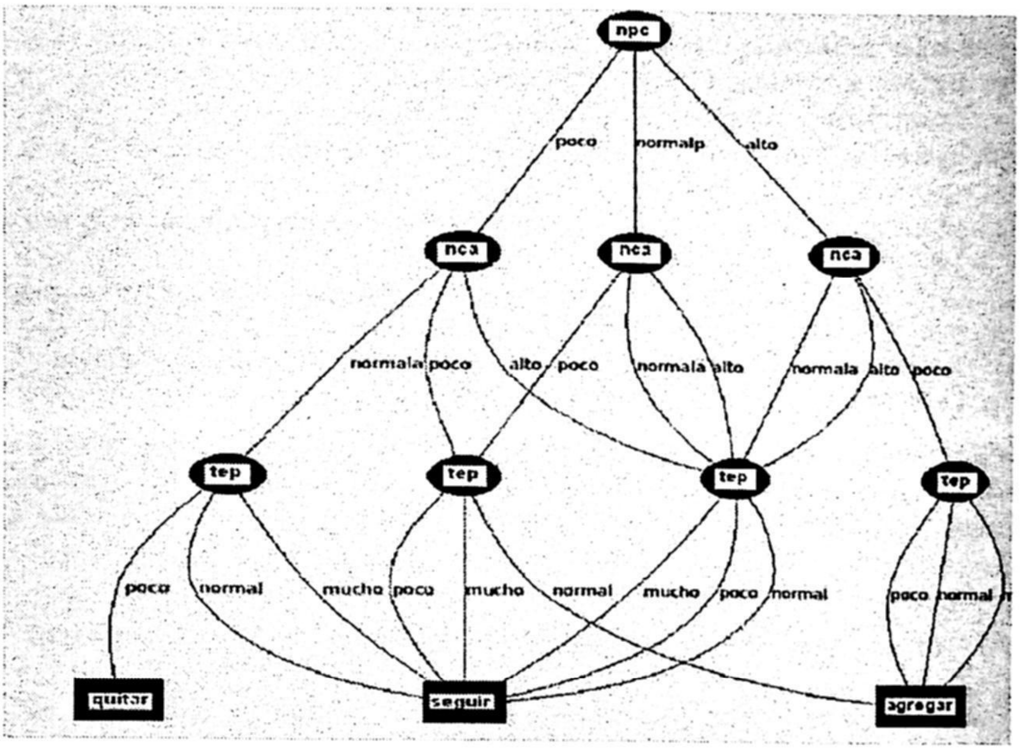


Fig. 4. *Optimal policy computed by Spudd. State variables are the following: npc represents the number of person waiting in queue, nca represents the number of cashiers attending customers, and tep represents the waiting time in the queue, each variable has three values. Actions are enclosed by rectangles, these are the following: eliminate a cashier (quitar), continue (seguir) and add a cashier (agregar).*

At the moment we have assumed that the underlying supermarket queue planning problem is fully observable, but in more general formulations, the con-troller may be able to make noisy observations about the world. Then, such planning problem can be formulated as a partially observable Markov decision process (POMDP) this is the target of future work.

References

1. C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy con-struction. *Proceedings IJCAI-95, Montreal*, pages 1104–1111, August 1995.
2. Brigham. On a congestion problem in an aircraft factory. *Operations Research*, 3:412–428, 1955.
3. W. K. Grassman, X. Chen, and Kashyap B.R.K. Optimal service rates for the state-dependent m/g/1 queues in steady state. *Operations Research Letters*, 29:57–63, 2001.
4. G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Arti cial Intelligence*, 42:393–405, 1990.
5. D. Gross and C. Harris. Fundamentals of queueing theory. *Wiley Sons, Inc.*, 1998.
6. Pen-Yuan Liao. Optimal sta ng policy for queuing systems with cyclic demands: waiting cost approach. *POMS 18th Annual Conference Dallas, Texas, U.S.A.*, pages 1–14, 2007.
7. Daria T., J. Christopher, and Kenneth N. A constraint programming approach for solving a queueing design and control problem. *Technical Report, Univ. of Toronto, Toronto, Ontario, Canada.*, pages 1–29, 2008.
8. M. Law and D. Kelton. Simulation modeling and analysis. *McGraw-Hill, New York*, 1991.
9. J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. Spudd: Stochastic planning using decision diagrams. *Proceedings of UAI 99, Stockholm Sweden*, pages 1–10, 1999.
10. Agner Fog. Instructions for the random number generator libraries. *GNU General Public License*, pages 1–30, December 2008.
11. M. L. Puterman. Markov decision process discrete stochastic dynamic program-ming. *New York, New York: John Wiley Sons, Inc.*, 1994.
12. D. Dearden and C. Boutilier. Abstraction and approximate decision theoretic planning. *Arti cial Intelligence*, 89:219–283, 1997.
13. T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, pages 142–150, 1989.