

Lefebvre, Mario

Article

A controlled discrete-time queueing system as a model for the orders of two competing companies

Games

Provided in Cooperation with:

MDPI – Multidisciplinary Digital Publishing Institute, Basel

Suggested Citation: Lefebvre, Mario (2024) : A controlled discrete-time queueing system as a model for the orders of two competing companies, Games, ISSN 2073-4336, MDPI, Basel, Vol. 15, Iss. 3, pp. 1-8,
<https://doi.org/10.3390/g15030019>

This Version is available at:

<https://hdl.handle.net/10419/330088>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>

Article

A Controlled Discrete-Time Queueing System as a Model for the Orders of Two Competing Companies

Mario Lefebvre

Department of Mathematics and Industrial Engineering, Polytechnique Montréal, C.P. 6079, Succursale Centre-ville, Montréal, QC H3C 3A7, Canada; mlefebvre@polymtl.ca

Abstract: We consider two companies that are competing for orders. Let $X_1(n)$ denote the number of orders processed by the first company at time n , and let $\tau(k)$ be the first time that $X_1(n) < j$ or $X_1(n) = r$, given that $X_1(0) = k$. We assume that $\{X_1(n), n = 0, 1, \dots\}$ is a controlled discrete-time queueing system. Each company is using some control to increase its share of orders. The aim of the first company is to maximize the expected value of $\tau(k)$, while its competitor tries to minimize this expected value. The optimal solution is obtained by making use of dynamic programming. Particular problems are solved explicitly.

Keywords: dynamic programming; difference equations; linear equations; first-passage time; homing problem

AMS Subject Classification: Primary 93E20; Secondary 60K25

1. Introduction

Many papers have been published on the optimal control of queueing systems. Often the authors assume that it is possible to control the service rate; see, for example, Laxmi et al. [1], Chen et al. [2] and Tian et al. [3]. In other papers, the aim is to control the arrival of customers into the system; see Wu et al. [4]. Sometimes, it is assumed that there are different types of customers; see Wen et al. [5] and Dudin et al. [6]. Another problem that is often considered is to determine the optimal number of servers in the system; see Asadzadeh et al. [7].

In this paper, we consider a single-server queueing model in discrete time. Orders arrive one at a time. We assume that the time needed for an order to arrive is a random variable A that follows a geometric distribution with parameter $2p_A$, so that

$$P[A = n] = (1 - 2p_A)^{n-1} 2p_A \quad \text{for } n = 1, 2, \dots \quad (1)$$

Moreover, the service time is a random variable S having a geometric distribution with parameter p_S .

There are two competing companies. In equilibrium, orders arrive at each company according to a random variable A_i having a geometric distribution with parameter p_A , for $i = 1, 2$.

Instead of solving an optimal control problem, our aim in this paper is to present a stochastic dynamic game that can serve as a model for the behavior of companies competing for orders. We must, of course, make simplifying assumptions in order to obtain a mathematically tractable problem. However, we believe that the model is realistic enough to be useful.

Let $X_1(n)$ be the number of orders being processed by the first company at time n , and let

$$\tau(k) [= \tau(k, j, r)] := \inf\{n \geq 0 : X_1(n) < j \text{ or } X_1(n) = r \mid X_1(0) = k\}, \quad (2)$$



Citation: Lefebvre, M. A Controlled Discrete-Time Queueing System as a Model for the Orders of Two Competing Companies. *Games* **2024**, *15*, 19. <https://doi.org/10.3390/g15030019>

Academic Editor: Ulrich Berger

Received: 30 April 2024

Revised: 26 May 2024

Accepted: 27 May 2024

Published: 29 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

where $j \leq k \leq r$ and $j, k, r \in \mathbb{N}$.

Now, suppose that each company can try to use some form of control in order to increase its share of the orders. More precisely, we assume that the first company uses the control $u_n \in \{0, p_A\}$ at time n to increase the arrival rate of its orders. Similarly, the second company uses $v_n \in \{-p_A, 0\}$ at time n to decrease the order arrival rate of its competitor.

Remark 1. Writing that $v_n = -p_A$ means that Company 2 is making some efforts (for instance, by lowering its prices) in order to reduce to zero the rate at which Company 1 receives orders, unless this company also tries to increase or at least keep its share of the orders.

Let

$$p_A(n) := p_A + u_n + v_n \quad \text{for } n = 1, 2, \dots \quad (3)$$

We define the cost function

$$J(k) = \sum_{n=0}^{\tau(k)-1} \left\{ q_1 u_n^2 + q_2 v_n^2 + \lambda (v_n - u_n) \right\}, \quad (4)$$

where q_1, q_2 and λ are positive constants. We look for the values u_n^* and v_n^* of the control variables that are such that the expected value of $J(k)$ is minimized.

The idea behind the cost function is as follows: since the parameter λ is positive, if r is large, then Company 1 wants to maximize the time it stays in business. To do this, it would like to use the control $u_n = p_A$; however, this leads to quadratic control costs. Company 2, for its part, would like to use $v_n = -p_A$ in order to bankrupt Company 1 as quickly as possible, but this also entails quadratic control costs.

Problems in which the optimizers try to minimize or maximize the expected value of a certain cost function until a given random event occurs are known as *homing problems*. Whittle [8] considered the case when the optimizer controls an n -dimensional diffusion process until it leaves a given subset of \mathbb{R}^n . Rishel [9] also treated the homing problems for n -dimensional diffusion processes; these processes were more realistic models for the wear of devices than those proposed by various authors.

The author has recently extended homing problems to the optimal control of queueing systems in continuous time; see [10–12]. In these three papers, the aim was to determine the optimal number of servers working at time t . He also published a paper ([13]) on a homing problem for a controlled random walk with two optimizers.

Next, we define the *value function*

$$F(k) = \min_{\substack{(u_n, v_n) \\ 0 \leq n \leq \tau(k) - 1}} E[J(k)]. \quad (5)$$

The function $F(k)$ is the expected cost incurred (which can sometimes be a reward) if both optimizers choose the optimal value of u_n and v_n between the initial time $n = 0$ and time $\tau(k) - 1$.

In Section 2, the *dynamic programming equation* satisfied by the value function $F(k)$ will be derived, and a particular problem will be solved explicitly. In Section 3, the problem formulation will be modified. We will then assume that the value of u_n is known, and we will look for the value of v_n that maximizes the expected value of a certain cost function. Concluding remarks will be made in Section 4.

2. Dynamic Programmic Equation

We will derive the dynamic programming equation satisfied by $F(k)$. We have

$$F(k) = \min_{\substack{(u_n, v_n) \\ 0 \leq n \leq \tau(k) - 1}} \left\{ q_1 u_0^2 + q_2 v_0^2 + \lambda(v_0 - u_0) + E \left[\sum_{n=1}^{\tau(k)-1} \left\{ q_1 u_n^2 + q_2 v_n^2 + \lambda(v_n - u_n) \right\} \right] \right\}. \tag{6}$$

Then, making use of Bellman’s principle of optimality (see [14]), we can write that

$$F(k) = \min_{(u_0, v_0)} \left\{ q_1 u_0^2 + q_2 v_0^2 + \lambda(v_0 - u_0) + E[F(X_1(1))] \right\}. \tag{7}$$

Indeed, whatever the two optimizers decide to do at time $n = 0$, the decisions they make from time $n = 1$ to time $\tau(k) - 1$ must be optimal.

Remark 2. Equation (7) is valid because of our assumptions that A_i , for $i = 1, 2$, and S have a geometric distribution. Indeed, as is well known, this distribution possesses the memoryless property. If we assume instead that these random variables have any other (discrete) distribution, then we will have to take the past into account, rendering the optimization problem almost intractable.

Furthermore, we have

$$E[F(X_1(1))] = F(k + 1) p_A(0) (1 - p_S) + F(k - 1) [1 - p_A(0)] p_S + F(k) [p_A(0) p_S + [1 - p_A(0)] (1 - p_S)]. \tag{8}$$

Hence, we can state the following proposition.

Proposition 1. The value function $F(k)$ satisfies the dynamic programming equation (DPE)

$$F(k) = \min_{(u_0, v_0)} \left\{ q_1 u_0^2 + q_2 v_0^2 + \lambda(v_0 - u_0) + F(k + 1) p_A(0) (1 - p_S) + F(k - 1) [1 - p_A(0)] p_S + F(k) [p_A(0) p_S + [1 - p_A(0)] (1 - p_S)] \right\}. \tag{9}$$

Moreover, we have the boundary condition $F(k) = 0$ if $k < j$ or $k = r$.

Remark 3.

- (i) We take for granted that each optimizer does not know what the other has decided to do. In Section 3, we will assume that Company 2 knows the decision made by Company 1.
- (ii) There are four possibilities for (u_0, v_0) : $(0, 0)$, $(0, -p_A)$, $(p_A, 0)$ and $(p_A, -p_A)$. If we solve the difference equation corresponding to each possible value of (u_0, v_0) , we actually obtain the value of the function $F(k)$ if the optimizers choose the same value of (u_0, v_0) for any k . Hence, we cannot obtain the value function and/or the optimal controls for any value of k by comparing the four expressions for $F(k)$ obtained by solving the four difference equations.
- (iii) We can write that $(u_n, v_n) = (u_n(k), v_n(k))$. The number of possible pairs $(u_0(k), v_0(k))$ for $j \leq k < r$ is equal to 4^{r-j} . If we have the values of $(u_0(k), v_0(k))$ for $k = j, \dots, r - 1$, we can solve a system of $r - j$ linear equations to obtain the corresponding values of $F(k)$ for any k . If $r - j$ is small, it is a simple matter to consider all the possible values of $(u_0(k), v_0(k))$ and compute the function $F(k)$ for $k = j, \dots, r - 1$. We can then determine the optimal controls and the associated value function.

In the following subsection, a particular problem will be solved explicitly.

An Example

Suppose that $q_1 = q_2 = \lambda = 1, j = 1$ and $r = 3$. Moreover, we take $p_A = 1/4$ and $p_S = 1/2$. The values of $F(1)$ and $F(2)$ for the $4^2 = 16$ possible choices for $(u_0(1), v_0(1))$ and $(u_0(2), v_0(2))$ are presented in Table 1.

Table 1. Values of $F(1)$ and $F(2)$ for the 16 possible choices for the control variables.

No.	$u_0(1)$	$v_0(1)$	$u_0(2)$	$v_0(2)$	$F(1)$	$F(2)$
1	0	0	0	0	0	0
2	p_A	0	0	0	-0.6	-0.45
3	0	$-p_A$	0	0	-0.375	-0.28125
4	0	0	p_A	0	-0.107	-0.4286
5	0	0	0	$-p_A$	-0.125	-0.5
6	p_A	$-p_A$	0	0	-0.923	-0.692
7	p_A	0	p_A	0	-0.75	-0.75
8	p_A	0	0	$-p_A$	-1.125	-1.5
9	0	$-p_A$	p_A	0	-0.375	-0.5625
10	0	$-p_A$	0	$-p_A$	-0.375	-0.75
11	0	0	p_A	$-p_A$	-0.231	-0.923
12	p_A	$-p_A$	p_A	0	-0.964	-0.857
13	p_A	$-p_A$	0	$-p_A$	-1.125	-1.5
14	p_A	0	p_A	$-p_A$	-1.2	-1.65
15	0	$-p_A$	p_A	$-p_A$	-0.375	-1.03125
16	p_A	$-p_A$	p_A	$-p_A$	-1.154	-1.615

For example, if $(u_0(1), v_0(1)) = (0, 0)$ and $(u_0(2), v_0(2)) = (0, -p_A)$ (Case no. 5), then we have $p_A(0) = p_A$ if $k = 1$ and $p_A(0) = 0$ if $k = 2$. We must solve the system of linear equations

$$F(1) = \frac{1}{8}F(2) + \frac{1}{2}F(1) \tag{10}$$

$$F(2) = -\frac{3}{16} + \frac{1}{2}F(2) + \frac{1}{2}F(1), \tag{11}$$

whose solution is $F(1) = -1/8$ and $F(2) = -1/2$.

We see that the optimal strategy is to choose Case no. 14; that is, we take $(u_0(1), v_0(1)) = (p_A, 0)$ and $(u_0(2), v_0(2)) = (p_A, -p_A)$.

Remark 4. The four difference equations that must be solved, subject to the boundary conditions $F(0) = F(3) = 0$, are given below. We denote their solutions by $F_i(k)$, for $i = 1, 2, 3, 4$.

(1) $(u_0, v_0) = (0, 0)$:

$$F(k) = \frac{1}{8}F(k+1) + \frac{3}{8}F(k-1) + \frac{1}{2}F(k). \tag{12}$$

We easily find that the solution is $F_1(k) \equiv 0$.

(2) $(u_0, v_0) = (p_A, 0)$:

$$F(k) = -\frac{3}{8} + \frac{1}{4}F(k+1) + \frac{1}{4}F(k-1) + \frac{1}{2}F(k). \tag{13}$$

We find that

$$F_2(k) = \frac{3}{8}k(-3+k). \tag{14}$$

(3) $(u_0, v_0) = (0, -p_A)$:

$$F(k) = -\frac{3}{8} + \frac{1}{2}F(k-1) + \frac{1}{2}F(k). \tag{15}$$

Because $p_A(0) = 0$, $X_1(n)$ cannot reach the value 3 if $k = 1$ or 2. The solution that satisfies the boundary condition $F(0) = 0$ is

$$F_3(k) = -\frac{3}{8}k. \tag{16}$$

(4) $(u_0, v_0) = (p_A, -p_A)$:

$$F(k) = -\frac{3}{8} + \frac{1}{8}F(k+1) + \frac{3}{8}F(k-1) + \frac{1}{2}F(k). \tag{17}$$

We have

$$F_4(k) = \frac{3}{52}(-3 + 3^{k+1} - 26k). \tag{18}$$

The functions $F_i(k)$, for $i = 2, 3, 4$, are shown in Figure 1. Moreover, the values of $F_i(k)$ for $k = 1, 2$ and $i = 1, 2, 3, 4$ are presented in Table 2.

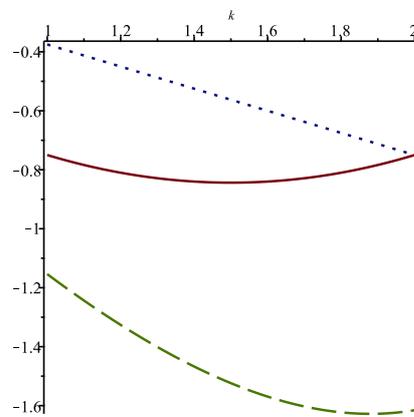


Figure 1. Functions $F_2(k)$ (solid line), $F_3(k)$ (dotted line) and $F_4(k)$ (dashed line) for $k \in [1, 2]$.

Table 2. Values of $F_i(k)$ for $k = 1, 2$ and $i = 1, 2, 3, 4$.

i	$F_i(1)$	$F_i(2)$
1	0	0
2	-0.375	-0.75
3	-0.75	-0.75
4	-1.154	-1.615

Notice that $F_1(k)$ corresponds to Case no. 1, while $F_4(k)$ corresponds to Case no. 16. We observe that none of the functions $F_1(k), \dots, F_4(k)$ is the value function. However, $F_4(k)$ yields values which are quite close to those obtained with the value function. Therefore, if $r - j$ is large, so that the number of equations to consider is also large, then a suboptimal solution can be obtained by assuming that $(u_0(k), v_0(k))$ will be the same for any k .

3. Optimal Control When u_n Is Known

In this section, we assume that Company 2 knows the strategy of Company 1, and tries to maximize the expected value of the following cost function:

$$C(k) = \sum_{n=0}^{\tau(k)-1} \{q_1 u_n + q_2 v_n\} + K[X_1(\tau(k))]. \tag{19}$$

The terminal cost function $K(\cdot)$ is defined by

$$K[X_1(\tau(k))] = \begin{cases} K_1 & \text{if } X_1(\tau(k)) = r, \\ K_2 & \text{if } X_1(\tau(k)) = j - 1, \end{cases} \tag{20}$$

where $K_1 < 0$ and $K_2 > 0$. The constant $r \in \mathbb{N}$ could be the maximum number of orders that Company 1 can process at the same time.

Suppose that Company 1 uses the control $u_n(k) = p_A$ for any n and any k . Company 2 must decide whether to choose $v_n(k) = 0$ or $-p_A$. We define the value function

$$V(k) = \max_{v_n, 0 \leq n \leq \tau(k)-1} E[C(k)]. \tag{21}$$

Proceeding as in the previous section, we can prove the following proposition.

Proposition 2. *The value function $V(k)$ satisfies the dynamic programming equation*

$$V(k) = \max_{v_0 \in \{-p_A, 0\}} \left\{ q_1 p_A + q_2 v_0 + V(k+1) p_A(0) (1 - p_S) + V(k-1) [1 - p_A(0)] p_S + V(k) [p_A(0) p_S + [1 - p_A(0)] (1 - p_S)] \right\},$$

where $p_A(0) = 2p_A + v_0$. Furthermore, the function $V(k)$ is such that $V(r) = K_1$ and $V(j-1) = K_2$.

There are now 2^{r-j} possible strategies for Company 2. If $r - j$ is small, we can proceed as in the previous section and calculate the expected value of $C(k)$ for each possible strategy.

Assume that $q_1 = q_2 = 1$, $p_A = 1/4$, $p_S = 1/2$, $K_1 = -10$, $K_2 = 10$, $r = 5$ and $j = 1$. We present in Table 3 the value of $V(1), \dots, V(4)$ for each of the 16 possible strategies that Company 2 can choose. These values are obtained by solving the system of four linear equations

$$V(k) = \frac{1}{4} + v_0(k) + V(k+1) \left(\frac{1}{2} + v_0(k) \right) \frac{1}{2} + V(k-1) \left(\frac{1}{2} - v_0(k) \right) \frac{1}{2} + V(k) \frac{1}{2}, \tag{22}$$

for $k = 1, 2, 3, 4$, together with the boundary conditions $V(0) = -10$ and $V(5) = 10$. We conclude that the optimal strategy is the one that corresponds to Case no. 15; that is, $v_0(1) = 0$ and $v_0(k) = -p_A$ for $k = 2, 3, 4$.

Table 3. Values of $V(1), \dots, V(4)$ for the 16 possible choices for the control variables $v_0(1), \dots, v_0(4)$.

No.	$v_0(1)$	$v_0(2)$	$v_0(3)$	$v_0(4)$	$V(1)$	$V(2)$	$V(3)$	$V(4)$
1	0	0	0	0	8	5	1	-4
2	$-p_A$	0	0	0	8.92	5.69	1.46	-3.77
3	0	$-p_A$	0	0	9.36	7.73	2.82	-3.09
4	0	0	$-p_A$	0	9.56	8.11	5.67	-1.67
5	0	0	0	$-p_A$	9.29	7.57	4.86	1.14
6	$-p_A$	$-p_A$	0	0	9.45	7.81	2.81	-3.06
7	$-p_A$	0	$-p_A$	0	9.52	8.08	5.64	-1.68
8	$-p_A$	0	0	$-p_A$	9.42	7.68	4.95	1.21
9	0	$-p_A$	$-p_A$	0	10.13	9.26	6.65	-1.17
10	0	$-p_A$	0	$-p_A$	10	9	6	2
11	0	0	$-p_A$	$-p_A$	10.47	9.93	8.40	3.80
12	$-p_A$	$-p_A$	$-p_A$	0	9.72	8.87	6.31	-1.34
13	$-p_A$	$-p_A$	0	$-p_A$	9.67	8.69	5.76	1.82
14	$-p_A$	0	$-p_A$	$-p_A$	9.84	9.35	7.86	3.40
15	0	$-p_A$	$-p_A$	$-p_A$	10.49	9.98	8.44	3.83
16	$-p_A$	$-p_A$	$-p_A$	$-p_A$	9.83	9.34	7.85	3.39

Remark 5. As in the previous section, we can obtain at least a suboptimal solution which is close the optimal one by solving the difference equations obtained by assuming first that $v_0(k) \equiv 0$, namely

$$V(k) = q_1 p_A + V(k+1) 2 p_A (1 - p_S) + V(k-1) (1 - 2 p_A) p_S + V(k) [2 p_A p_S + (1 - 2 p_A) (1 - p_S)], \tag{23}$$

and then that $v_0(k) \equiv -p_A$:

$$V(k) = q_1 p_A - q_2 p_A + V(k+1) p_A (1 - p_S) + V(k-1) (1 - p_A) p_S + V(k) [p_A p_S + (1 - p_A) (1 - p_S)]. \tag{24}$$

The first equation corresponds to Case no. 1, and the second one to Case no. 16. We find that the solutions to these equations in the particular case considered above are, respectively,

$$V_1(k) = 10 - \frac{3}{2} k - \frac{1}{2} k^2 \tag{25}$$

and

$$V_{16}(k) = \frac{1220}{121} - \frac{10}{121} 3^k. \tag{26}$$

See Figure 2.

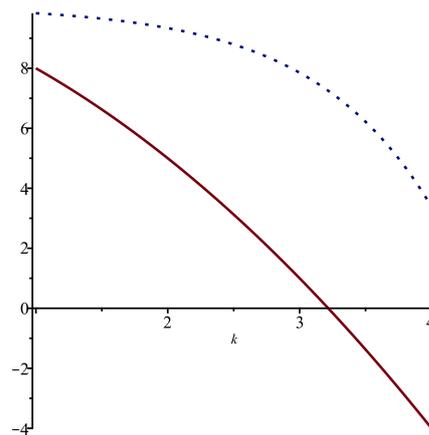


Figure 2. Functions $V_1(k)$ (solid line) and $V_{16}(k)$ (dotted line) for $k \in [1, 4]$.

Notice that the solution obtained when $v_0(k) \equiv 0$ (Case no. 1) actually gives the minimum of the expected value of the cost function $C(k)$. Moreover, we see in Table 3 that the choice that corresponds to Case no. 11 almost yields the optimal solution that we are looking for.

4. Conclusions

In this paper, a homing problem for a queueing model in discrete time has been considered. The problem can be seen as a dynamic game because there are two optimizers with opposing objectives.

In Section 2, dynamic programming was used to derive the equation satisfied by the value function. From this equation, one can deduce the optimal values of the control variables. However, we have seen that, in order to do this, one has to solve a possibly large number of systems of linear equations, subject to the appropriate boundary conditions. Although solving each system is straightforward, repeating this procedure a large number of times can become tedious. We have also seen that it is possible to obtain a good suboptimal solution to our problem fairly quickly.

In Section 3, the problem formulation was modified. We assumed that the strategy of Company 1 was known, and we looked for the strategy that Company 2 should adopt to

maximize the expected value of a certain cost function. We treated the case when the control variable u_n is always equal to p_A . However, the same type of analysis could be carried out for any choice of u_n . In particular, we could find the optimal strategy of Company 2 if $u_n \equiv 0$.

In theory, we could easily extend the problems considered to the case when each optimizer can choose between more than two possible values for the variable it controls. The calculations would, however, become quite complex. One could possibly use numerical simulations to determine the optimal solutions. Indeed, instead of solving a large number of difference equations, simulating the proposed model can enable us to determine the optimal solution by computing the value function for each simulation. Simulating geometric random variables is not a difficult task.

Finally, it is also possible to consider optimal control problems for queueing models in continuous time with two optimizers.

Funding: This research was supported by the Natural Sciences and Engineering Research Council of Canada.

Data Availability Statement: No data was used for this research.

Acknowledgments: The author wishes to thank the anonymous reviewers of this paper for their constructive comments.

Conflicts of Interest: The author reports that there are no competing interests to declare.

References

1. Laxmi, P.V.; Bhavani, E.G.; Jyothsna, K. Analysis of Markovian queueing system with second optional service operating under the triadic policy. *OPSEARCH* **2023**, *60*, 256–275. [[CrossRef](#)]
2. Chen, G.; Liu, Z.; Xia, L. Event-based optimization of service rate control in retrial queues. *J. Oper. Res. Soc.* **2023**, *74*, 979–991. [[CrossRef](#)]
3. Tian, R.; Su, S.; Zhang, Z.G. Equilibrium and social optimality in queues with service rate and customers' joining decisions. *Qual. Technol. Quant. Manag.* **2024**, *21*, 1–34. [[CrossRef](#)]
4. Wu, C.-H.; Yang, D.-Y.; Yong, C.-R. Performance evaluation and bi-objective optimization for F -policy queue with alternating service rates. *J. Ind. Manag. Optim.* **2023**, *19*, 3819–3839. [[CrossRef](#)]
5. Wen, J.; Geng, N.; Xie, X. Optimal insertion of customers with waiting time targets. *Comput. Oper. Res.* **2020**, *122*, 105001. [[CrossRef](#)]
6. Dudin, A.; Dudin, S.; Dudina, O. Analysis of a queueing system with mixed service discipline. *Methodol. Comput. Appl. Prob.* **2023**, *25*, 57. [[CrossRef](#)]
7. Asadzadeh, S.; Akhavan, B.; Akhavan, B. Multi-objective optimization of Gas Station performance using response surface methodology. *Int. J. Qual. Reliab. Manag.* **2021**, *38*, 465–483. [[CrossRef](#)]
8. Whittle, P. *Optimization over Time*; Wiley: Chichester, UK, 1982; Volume I.
9. Rishel, R. Controlled wear process: Modeling optimal control. *IEEE Trans. Automat. Control* **1991**, *36*, 1100–1102. [[CrossRef](#)]
10. Lefebvre, M. An optimal control problem for a modified $M/G/k$ queueing system. In Proceedings of the Workshop on Intelligent Information Systems, Chişinău, Republic of Moldova, 19–21 October 2023; pp. 141–148.
11. Lefebvre, M. Reducing the size of a waiting line optimally. *WSEAS Trans. Syst. Control* **2023**, *18*, 342–345. [[CrossRef](#)]
12. Lefebvre, M.; Yaghoubi, R. Optimal control of a queueing system. 2024, *Submitted for publication*.
13. Lefebvre, M. A discrete-time homing problem with two optimizers. *Games* **2023**, *14*, 68. [[CrossRef](#)]
14. Bellman, R. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.