Journal of Systems Engineering and Electronics
Vol. 33, No. 1, February 2022, pp.144 – 155

Optimal policy for controlling two-server queueing systems with jockeying

LIN Bing[1,*], LIN Yuchen[2], and BHATNAGAR Rohit[3]

1. School of Business, Jiangsu Normal University, Xuzhou 221116, China;
2. Wenzheng College, Soochow University, Suzhou 215104, China;
3. Nanyang Business School, Nanyang Technological University, Nanyang 639798, Singapore

Abstract: This paper studies the optimal policy for joint control of admission, routing, service, and jockeying in a queueing system consisting of two exponential servers in parallel. Jobs arrive according to a Poisson process. Upon each arrival, an admission/routing decision is made, and the accepted job is routed to one of the two servers with each being associated with a queue. After each service completion, the servers have an option of serving a job from its own queue, serving a jockeying job from another queue, or staying idle. The system performance is inclusive of the revenues from accepted jobs, the costs of holding jobs in queues, the service costs and the job jockeying costs. To maximize the total expected discounted return, we formulate a Markov decision process (MDP) model for this system. The value iteration method is employed to characterize the optimal policy as a hedging point policy. Numerical studies verify the structure of the hedging point policy which is convenient for implementing control actions in practice.

Keywords: queueing system, jockeying, optimal policy, Markov decision process (MDP), dynamic programming.

DOI: 10.23919/JSEE.2022.000015

1. Introduction

Queueing models are now widely used to study the manufacturing systems, public service systems, distributed computer systems, data communication networks, traffic flow systems, healthcare operations management, etc. A queueing system typically consists of three components: jobs, queues, and servers. Here, a job could be a part, a telephone call, a data file, a computer program, a patient, or a plane. Correspondingly, a server could be a workstation in a plant, a telecommunication transmission channel, a data transmission channel, a CPU, a clinic, or a runway.

Controls are often applied to queueing systems to improve system performance. Queueing controls usually take the form of static (open-loop) control and dynamic (close-loop) control. For dynamic controls, we can dynamically change some parameters of queueing systems, such as arrival rate and service rate, or we may implement the rules for routing jobs to parallel servers upon job arrivals.

Queueing control problems have been extensively studied in literature. Here we mention a few significant works among the earliest ones. For the admission control models, readers may refer to [1−4] for detailed illustration. The research works [5−8] considered routing control problems while [9] and [10] studied the service rate control problems. For the joint admission and routing controls, [11] and [12] were among the earliest to study this issue. Comprehensive surveys on controlling queueing systems can be found in [13].

In this paper, we study the optimal joint control of admission, routing, service, and jockeying in a queueing system of two parallel servers. Jobs arrive according to a Poisson process. Upon each arrival, a system controller will decide which job is admitted into the system and to which server an admitted job is sent. Each server is associated with a queue with no capacity limit. Two exponential servers with distinct service rates are controlled in the following manner: once a service is completed, a server may stop service, or serve the job from its own queue, or serve a jockeying job from another queue.

The system performance is measured in terms of the revenues from the accepted jobs, the holding costs for jobs in queue, the service costs for processing jobs, and the jockeying costs associated with transferring jobs from one queue to another. To characterize the optimal control policy, we formulate a Markov decision process (MDP) model with an objective to maximize the total expected discounted return in infinite horizon.

Job scheduling and logistics and supply chain coordination are challenging as shown in [14−17]. Our research is motivated by the example of managing global supply

Manuscript received October 19, 2020.
*Corresponding author.
This work was supported by the National Social Science Fund of China (19BGL100).

chains with multiple production bases since many transnational corporations maintain two production bases to fulfill their global operations strategies. For instance, Zara and H&M, global leaders in apparel industry, have one production base in Europe and the other in Asia. The well-known sportswear companies Nike and Adidas keep two production bases: one in China and the other in Southeast Asian countries. Haier, a leading manufacturer in home appliances, operates one production base in China serving the whole global market and the other in the US focusing on the American market. Even for local supply chains in China, our research can find potential applications. Suppose a manufacturing firm has two separate plants in the Pearl River Delta and the Yangtze River Delta, the two most developed areas in China. When customer orders arrive, the firm dispatches the orders to two plants. To fully exploit the production and logistics resources of two plants, the firm further allows order jockeying before their final entry into the production process. For instance, an order originally assigned to the plant in the Pearl River Delta can be conveniently reassigned to the plant in the Yangtze River Delta via information systems. Along with the above applications in operations management, our model can be applied to telecommunication networks, computer systems, and vehicular traffic flow. One specific example given in [18] is that of a multibeam satellite system that serves the earth-based stations organized into disjoint zones. For such a system, an effective routing and jockeying control rule is necessary to achieve efficient packet transmission.

The research on queueing systems with jockeying dates to the 1950s. However, most of the studies are classified into the descriptive models which focus on performance evaluation under specific jockeying rules. In this stream of research, [19–34] are typical works. The shortest queue problem with two parallel queues and threshold jockeying was studied in [19]. Adan et al. [20] and Zhao et al. [21] further studied the case of multiple parallel queues. The special case of the shortest queue problem with instantaneous jockeying, i.e., jockeying occurs whenever one queue is shorter than others, was examined in [22–27]. For more recent research on the shortest queue problem with jockeying, please refer to [28,29]. For the applications, Jeganathan et al. [30] studied the jockeying in inventory management while Stagje [31] considered the jockeying of cost-conscious customer in service systems. Tarabia [32] examined jockeying in parallel queues in the case of restricted capacities. Baykasoglu et al. [33] modeled the jockeying problem associated with manufacturing systems and Chaleshtori et al. [34] analyzed the location-allocation problem with jockeying.

In contrast with numerous studies on controlling queueing systems in terms of admission, routing, and service, there are only a few on jockeying. Xu et al. [18] considered optimal control of routing and jockeying in a two-station queueing system which had a Poisson arrival of jobs and exponential service time at two stations in parallel. They formulated the queueing control problem as an MDP and used dynamic programming to characterize the optimal policy as a switching-curve policy for both discounted and long-run average cost criteria. Down et al. [35] studied a system of multiple parallel queues with each queue having a dedicated arrival stream. They first discussed the condition under which the policies yield a stable Markov chain. For the two-server case with Poisson arrivals and exponential service time, they formulated an MDP model to characterize the optimal policy which was used to further develop a heuristic policy for the general case. Rosberg et al. [36] investigated a problem of energy efficiency for stochastically assigning jobs in a server farm with multiple processor-sharing servers and finite buffer sizes. The cases of jockeying and no jockeying were considered. For the case with jockeying, the authors formulated the problem as a semi-MDP to derive the optimal assignment policy and two heuristic policies. Dehghanian et al. [37] examined the optimal joining and jockeying policy for a queueing system of two stations in parallel. Customers arrived in a Poisson stream and chose to join one of the two stations with one chance of jockeying to the other queue. The optimal individual policies for joining and jockeying were characterized as the monotone threshold policy with the problem formulated as an MDP to minimize total holding and jockeying costs.

Our research is closely related to [18]. Both [18] and this paper use MDPs to characterize the optimal control policy for a system of two parallel queues with infinite buffer sizes. However, a significant difference exists between the two papers. Regarding models, only the routing and jockeying controls were studied in [18] which was motivated by a multibeam satellite system serving earth-based stations that were organized into disjoint zones. Since our research is oriented towards applications in operations management which are more concerned with closely matching supply with customer orders, we study a more complex model which includes joint controls of admission, routing, service, and jockeying. Consequently, the optimal policy in their paper was characterized as monotonically nondecreasing switching curves while the optimal policy in this paper was a hedging point policy characterized by one nonincreasing switching curve and one nondecreasing switching curve.

Moreover, we consider unit jockeying, i.e., transferring only one job each time from one queue to another, while they studied the case of batch jockeying under which multiple jobs were transferred each time. Obviously, unit jockeying is more responsive than the batch jockeying. Therefore, unit jockeying is more likely to be implemented in the operations management systems which are often intended for quick response to customer orders.

To the best of our knowledge, the problem in this paper has never been studied before. Our main contribution is the incorporation of jockeying controls into queueing systems, the formulation of a general model for various applications as mentioned above, and the subsequent characterization of the optimal control policy.

The rest of the paper is organized as follows. In Section 2, we present the model formulation. The optimal policy is characterized in Section 3 and system analysis is conducted in Section 4. Numerical studies are given to illustrate our results in Section 5. Conclusions are drawn in Section 6.

2. Model formulation

The problem discussed in this paper is illustrated in Fig. 1.

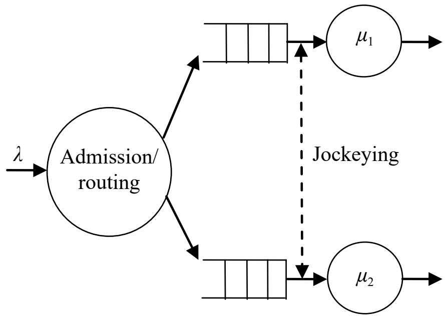

Fig. 1 Queueing system with two servers in parallel

The system state is $\boldsymbol{x} = (x_1, x_2)$, where x_1 and x_2 denote the number of jobs in two queues. Hence, x_1 and x_2 are nonnegative integers, i.e., $\boldsymbol{x} = (x_1, x_2) \in \boldsymbol{X} \equiv \mathbf{Z}_+^2$, where the state space $\boldsymbol{X}$ is a two-dimensional nonnegative integer set $\mathbf{Z}_+^2$. Jobs arrive according to a Poisson process with rate λ. Upon arrival, each job is subject to admission control. The rejected jobs are lost, and an accepted job is routed to one of the two servers. The server-dependent revenues generated from the accepted jobs are r_1 and r_2, respectively. The service times follow the exponential distributions with rates μ_1 and μ_2, respectively. Suppose the job holding cost function is $h(\boldsymbol{x}) = h_1x_1 + h_2x_2$, where h_1 and h_2 are the unit holding cost per unit time at two queues. The unit service costs at two servers are c_1 and c_2, respectively. The unit job jockeying cost c_{12} or c_{21} is associated with transferring a job from Queue 1 to Queue 2 or from Queue 2 to Queue 1. Further, we assume $\lambda < \mu_1 + \mu_2$ which assures the stability of the system in the long run.

For the above-described queueing system, the admission/routing control actions are made only at the decision epochs when each job arrives while the service and jockeying control actions are made at the decision epochs when the service of a job is completed. Thanks to the memoryless property of the exponential inter-arrival time and service time distributions, the system evolution is only influenced by the control actions made at the decision epochs. Consequently, the system evolution forms a two-dimensional continuous-time Markov chain, and all decision epochs are the Markov renewal points of the process. Hence, we can restrict our attention to Markov policies since a Markov policy is the optimal.

Denote the control action by $\boldsymbol{a} = (a_0, a_1, a_2)$, where $a_0 =$ 0, 1, 2, indicating the control action of rejecting a job, routing an accepted job to Server 1, and routing an accepted job to Server 2, respectively; $a_i = 0, 1, 2$ $(i = 1, 2)$, indicating the control action of stopping service, serving its own job, and serving a jockeying job, respectively. Thus, the control action $\boldsymbol{a}$ takes integer values within the finite set $[0,1,2]\times[0,1,2]\times[0,1,2]$. An admissible policy u consists of a sequence of functions $u = \{u_0, u_1, u_2, \cdots\} \in U$, where U denotes the set of all admissible policies. At each decision epoch k $(k = 0, 1, 2, \cdots)$, the function u_k maps the state $\boldsymbol{x} = (x_1, x_2)$ into a control action, i.e., $u_k(\boldsymbol{x}) = \boldsymbol{a}$ for all $\boldsymbol{x} \in \boldsymbol{X}$.

Let α $(\alpha > 0)$ denote the continuous-time discount rate. For the infinite horizon model of the problem considered in this paper, given the initial state $\boldsymbol{x} = (x_1, x_2)$, the total expected discounted return associated with a policy u, which is denoted by $J_u(\boldsymbol{x})$, can be written as

$$J_u(\boldsymbol{x}) = \mathrm{E}_x^u\left[\sum_{i=1}^{2}\int_0^\infty \mathrm{e}^{-\alpha t} r_i \mathrm{d}M_i(t) - \sum_{i=1}^{2}\int_0^\infty \mathrm{e}^{-\alpha t} c_i \mathrm{d}N_i(t) - \int_0^\infty \mathrm{e}^{-\alpha t} h(\boldsymbol{x})\mathrm{d}t - \int_0^\infty \mathrm{e}^{-\alpha t} c_{12}\mathrm{d}N_{12}(t) - \int_0^\infty \mathrm{e}^{-\alpha t} c_{21}\mathrm{d}N_{21}(t)\right]$$

where $M_i(t)$, $N_i(t)$ $(i = 1, 2)$, $N_{12}(t)$, and $N_{21}(t)$ represent the number of jobs accepted to two servers, the number of service completed at two servers, the number of jobs transferred from Queue 1 to Queue 2, and the number of jobs transferred from Queue 2 to Queue 1 up to time t, respectively. Next, we seek a stationary policy $u = \{u_\infty, u_\infty, u_\infty, \cdots\} \in U$ to maximize $J_u(\boldsymbol{x})$. Thus, the optimal value function V can be written as follows:

$$V(\boldsymbol{x}) = \max_{u \in U} J_u(\boldsymbol{x}).$$

The above V is then shown to satisfy the Bellman equation based on the dynamic programming optimality principle. Following the technique in [38], we uniformize the transition rate as $\Lambda = \lambda + \mu_1 + \mu_2$. After the uniformization of transitions, the original continuous-time Markov chain is transformed into a probabilistically equivalent system observed at evenly spaced points in time. In other words, a random event occurs in the system with a rate Λ and the event happens to be a job arrival with the probability λ/Λ, a service completion at Server 1 with the probability μ_1/Λ, a service completion at Server 2 with the probability μ_2/Λ. Next, we describe in sequel the system transitions considering the control actions.

Denote $\boldsymbol{e}_1 = (1, 0)^{\mathrm{T}}$ and $\boldsymbol{e}_2 = (0, 1)^{\mathrm{T}}$. Given a state (x_1, x_2) and for each control action a, the next state after transition is denoted by $\boldsymbol{y}$. Then we can define the transition probability function $p(\boldsymbol{y} \mid \boldsymbol{x}, u)$ as

$$\begin{aligned}
p(\boldsymbol{y}|\boldsymbol{x},u) = {} & \frac{\lambda}{\Lambda} I\{\boldsymbol{x}|\boldsymbol{x}, a_0 = 0\} + \frac{\lambda}{\Lambda} I\{\boldsymbol{x}+\boldsymbol{e}_1|\boldsymbol{x}, a_0 = 1\} + \frac{\lambda}{\Lambda} I\{\boldsymbol{x}+\boldsymbol{e}_2|\boldsymbol{x}, a_0 = 2\} + \\
& \frac{\mu_1}{\Lambda} I\{\boldsymbol{x}|\boldsymbol{x}, a_1 = 0\} + \frac{\mu_1}{\Lambda} I\{\boldsymbol{x}-\boldsymbol{e}_1|\boldsymbol{x}, a_1 = 1\} + \frac{\mu_1}{\Lambda} I\{\boldsymbol{x}-\boldsymbol{e}_2|\boldsymbol{x}, a_1 = 2\} + \\
& \frac{\mu_2}{\Lambda} I\{\boldsymbol{x}|\boldsymbol{x}, a_2 = 0\} + \frac{\mu_2}{\Lambda} I\{\boldsymbol{x}-\boldsymbol{e}_2|\boldsymbol{x}, a_2 = 1\} + \frac{\mu_2}{\Lambda} I\{\boldsymbol{x}-\boldsymbol{e}_1|\boldsymbol{x}, a_2 = 2\}
\end{aligned}$$

where $I\{\cdot\}$ is the indicator function. For instance, the term $\frac{\mu_1}{\Lambda} I\{\boldsymbol{x}-\boldsymbol{e}_2|\boldsymbol{x}, a_1 = 2\}$ implies that an event of service completion happens at Server 1 with the probability $\frac{\mu_1}{\Lambda}$ and a job is transferred from Queue 2 to Queue 1 if the action $a_1 = 2$ is selected, leading to a transition from the current state $\boldsymbol{x}$ to the next $(\boldsymbol{x} - \boldsymbol{e}_2)$.

In the following analysis, for the convenience of notations and analysis, we define the operators T_0, T_1, T_2 as follows:

$$(T_0 V)(\boldsymbol{x}) = \max\{r_1 + V(\boldsymbol{x}+\boldsymbol{e}_1), r_2 + V(\boldsymbol{x}+\boldsymbol{e}_2), V(\boldsymbol{x})\},$$

$$(T_1 V)(\boldsymbol{x}) = \begin{cases} \max\{V(\boldsymbol{x}-\boldsymbol{e}_1) - c_1, V(\boldsymbol{x}-\boldsymbol{e}_2) - c_{21}, V(\boldsymbol{x})\}, & x_1 > 0;\ x_2 > 0 \\ \max\{V(\boldsymbol{x}-\boldsymbol{e}_1) - c_1, V(\boldsymbol{x})\}, & x_1 > 0;\ x_2 = 0 \\ \max\{V(\boldsymbol{x}-\boldsymbol{e}_2) - c_{21}, V(\boldsymbol{x})\}, & x_1 = 0;\ x_2 > 0 \\ V(\boldsymbol{x}), & x_1 = 0;\ x_2 = 0 \end{cases},$$

$$(T_2 V)(\boldsymbol{x}) = \begin{cases} \max\{V(\boldsymbol{x}-\boldsymbol{e}_1) - c_{12}, V(\boldsymbol{x}-\boldsymbol{e}_2) - c_2, V(\boldsymbol{x})\}, & x_1 > 0;\ x_2 > 0 \\ \max\{V(\boldsymbol{x}-\boldsymbol{e}_1) - c_{12}, V(\boldsymbol{x})\}, & x_1 > 0;\ x_2 = 0 \\ \max\{V(\boldsymbol{x}-\boldsymbol{e}_2) - c_2, V(\boldsymbol{x})\}, & x_1 = 0;\ x_2 > 0 \\ V(\boldsymbol{x}), & x_1 = 0;\ x_2 = 0 \end{cases},$$

where T_0, T_1, and T_2 are the operators for admission/routing controls, service and jockeying controls at Server 1, and service and jockeying controls at Server 2, respectively.

Then, from the principle of optimality of dynamic programming, the optimal value function V can be shown to satisfy the following Bellman equation:

$$V = TV \tag{1}$$

where the dynamic programming operator (or the Bellman operator) T is as follows:

$$TV = \frac{\Lambda}{\alpha + \Lambda}\left\{\frac{-h}{\Lambda} + \frac{\lambda}{\Lambda} T_0 V + \frac{\mu_1}{\Lambda} T_1 V + \frac{\mu_2}{\Lambda} T_2 V\right\}. \tag{2}$$

Furthermore, without loss of generality for analysis and computation, we can rescale the time to achieve $\alpha + \Lambda = 1$. Hence, TV can be simplified as

$$TV = -h + \lambda T_0 V + \mu_1 T_1 V + \mu_2 T_2 V. \tag{3}$$

3. Optimal control policy

In this section, we explore the structure of the optimal policy. Define V^s as the set of functions on $\mathbf{Z}_+^2$ such that if $V \in V^s$, then the following properties exist:

(i) $V(\boldsymbol{x} + \boldsymbol{e}_1) - V(\boldsymbol{x}) \downarrow x_1 \downarrow x_2$;

(ii) $V(\boldsymbol{x} + \boldsymbol{e}_2) - V(\boldsymbol{x}) \downarrow x_1 \downarrow x_2$;

(iii) $V(\boldsymbol{x} + \boldsymbol{e}_1) - V(\boldsymbol{x} + \boldsymbol{e}_2) \downarrow x_1 \uparrow x_2$.

$\uparrow$ and $\downarrow$ indicate non-decreasing and non-increasing, respectively. The $V(\boldsymbol{x} + \boldsymbol{e}_1) - V(\boldsymbol{x}) \downarrow x_1$ in (i) and the $V(\boldsymbol{x} + \boldsymbol{e}_2) - V(\boldsymbol{x}) \downarrow x_2$ in (ii) refer to the discrete concavity in x_1 and x_2, respectively. The $V(\boldsymbol{x} + \boldsymbol{e}_1) - V(\boldsymbol{x}) \downarrow x_2$ in (i) and the $V(\boldsymbol{x} + \boldsymbol{e}_2) - V(\boldsymbol{x}) \downarrow x_1$ in (ii) are identical and referred to as the submodularity of $V(\boldsymbol{x})$. In some papers,

(iii) refers to the subconcavity of $V(\boldsymbol{x})$, where $V(\boldsymbol{x}+\boldsymbol{e}_1)-V(\boldsymbol{x}+\boldsymbol{e}_2)\downarrow x_1$ implies $V(\boldsymbol{x}+\boldsymbol{e}_1)-V(\boldsymbol{x}+\boldsymbol{e}_2)\geqslant V(\boldsymbol{x}+2\boldsymbol{e}_1)-V(\boldsymbol{x}+\boldsymbol{e}_1+\boldsymbol{e}_2)$ and $V(\boldsymbol{x}+\boldsymbol{e}_1)-V(\boldsymbol{x}+\boldsymbol{e}_2)\uparrow x_2$ suggests $V(\boldsymbol{x}+\boldsymbol{e}_1)-V(\boldsymbol{x}+\boldsymbol{e}_2)\leqslant V(\boldsymbol{x}+\boldsymbol{e}_1+\boldsymbol{e}_2)-V(\boldsymbol{x}+2\boldsymbol{e}_2)$. For brevity, we will use increasing (decreasing) and non-decreasing (non-increasing) interchangeably. Next result shows the existence of an optimal policy.

Proposition 1 There exists an optimal deterministic stationary policy for (1).

Proof Since the state space is discrete and the control action set is finite for each $\boldsymbol{x}\in\boldsymbol{X}$ in our model, we can find an admissible policy u to attain the maximum of the right-hand side of (1). According to Theorem 6.2.10 in [39], u is a stationary optimal policy. □

Then we show that all the operators defined above propagate the structural properties (i)−(iii).

Lemma 1 T_0V, T_1V, and T_2V and $TV\in V^s$ if $V\in V^s$.

Proof Readers may refer to [40] for the detailed proofs of such typical operations. □

Next lemma shows that the optimal value function retains the properties (i)−(iii).

Lemma 2 The optimal value function $V\in V^s$.

Proof The result is proved by value iteration. Let $V_0=0$ which is in V^s. Based on Lemma 1, we apply T repeatedly to V_0, leading to $T^nV_0\in V^s$ for all n. As n approaches infinity, $(T^nV_0)(\boldsymbol{x})$ takes the point-wise convergence to the optimal value function $V(\boldsymbol{x})$ for all $\boldsymbol{x}$. Hence, V retains all the structural properties (i)−(iii) based on the knowledge of mathematical analysis. Thus, $V\in V^s$. □

Then we define the switching functions which are necessary to characterize the structure of the optimal policy. Thus, we have

$$S_1(x_1)=\min\{x_2|r_1+V(\boldsymbol{x}+\boldsymbol{e}_1)-V(\boldsymbol{x})\leqslant 0,\ r_2+V(\boldsymbol{x}+\boldsymbol{e}_2)-V(\boldsymbol{x})\leqslant 0,\ \text{given } x_1\},$$

$$S_2(x_1)=\min\{x_2|V(\boldsymbol{x}+\boldsymbol{e}_1)-V(\boldsymbol{x}+\boldsymbol{e}_2)+r_1-r_2\geqslant 0,\ \text{given } x_1\},$$

$$L_1(x_1)=\min\{x_2|V(\boldsymbol{x})-V(\boldsymbol{x}-\boldsymbol{e}_1)+c_1\leqslant 0 \text{ or } V(\boldsymbol{x})-V(\boldsymbol{x}-\boldsymbol{e}_2)+c_{21}\leqslant 0,\ \text{given } x_1, \text{and } x_1>0,\ x_2>0\},$$

$$L_2(x_1)=\min\{x_2|V(\boldsymbol{x}-\boldsymbol{e}_2)-V(\boldsymbol{x}-\boldsymbol{e}_1)-c_{21}+c_1\geqslant 0, \text{given } x_1,\ \text{and } x_1>0,\ x_2>0\},$$

$$L_3(x_2=0)=\min\{x_1|V(\boldsymbol{x})-V(\boldsymbol{x}-\boldsymbol{e}_1)+c_1\leqslant 0,\ \text{given } x_2=0 \text{ and } x_1>0\},$$

$$L_4(x_1=0)=\min\{x_2|V(\boldsymbol{x})-V(\boldsymbol{x}-\boldsymbol{e}_2)+c_{21}\leqslant 0,\ \text{given } x_1=0 \text{ and } x_2>0\},$$

$$G_1(x_1)=\min\{x_2|V(\boldsymbol{x})-V(\boldsymbol{x}-\boldsymbol{e}_1)+c_{12}\leqslant 0 \text{ or } V(\boldsymbol{x})-V(\boldsymbol{x}-\boldsymbol{e}_2)+c_2\leqslant 0,\ \text{given } x_1, \text{and } x_1>0,\ x_2>0\},$$

$$G_2(x_1)=\min\{x_2|V(\boldsymbol{x}-\boldsymbol{e}_2)-V(\boldsymbol{x}-\boldsymbol{e}_1)-c_2+c_{12}\geqslant 0, \text{given } x_1,\ \text{and } x_1>0,\ x_2>0\},$$

$$G_3(x_2=0)=\min\{x_1|V(\boldsymbol{x})-V(\boldsymbol{x}-\boldsymbol{e}_1)+c_{12}\leqslant 0,\ \text{given } x_2=0 \text{ and } x_1>0\},$$

$$G_4(x_1=0)=\min\{x_2|V(\boldsymbol{x})-V(\boldsymbol{x}-\boldsymbol{e}_2)+c_2\leqslant 0,\ \text{given } x_1=0 \text{ and } x_2>0\}.$$

From the above definitions, the switching functions S_1 and S_2 are associated with the admission and routing decisions. The switching functions L_i ($i=1,2,3,4$) are associated with the service and jockeying controls at Server 1. And the switching functions G_i ($i=1,2,3,4$) are associated with the service and jockeying controls at Server 2. In the following discussion, we demonstrate that the switching functions are monotone with respect to the state variables.

Lemma 3 $S_1(x_1)$ is decreasing in x_1 and $S_2(x_1)$ is increasing in x_1; $L_1(x_1)$ is decreasing in x_1 and $L_2(x_1)$ is increasing in x_1; $G_1(x_1)$ is decreasing in x_1 and $G_2(x_1)$ is increasing in x_1.

Proof We only prove the result of $S_1(x_1)$ and the rest can be proved analogously.

For $S_1(x_1)$, it can be regarded as a function of two parts made up from the following two respective functions:

$$S'_1(x_1)=\min\{x_2|r_1+V(\boldsymbol{x}+\boldsymbol{e}_1)-V(\boldsymbol{x})\leqslant 0,\ \text{given } x_1\}$$

$$S''_1(x_1)=\min\{x_2|r_2+V(\boldsymbol{x}+\boldsymbol{e}_2)-V(\boldsymbol{x})\leqslant 0,\ \text{given } x_1\}.$$

Regarding $S'_1(x_1)$, for a given x_1, $S'_1(x_1)$ is the least value of x_2 to satisfy

$$r_1+V(\boldsymbol{x}+\boldsymbol{e}_1)-V(\boldsymbol{x})\leqslant 0,$$

that is,

$$r_1+V(x_1+1,S'_1(x_1))-V(x_1,S'_1(x_1))\leqslant 0.$$

Further, the concavity of V implies that

$$r_1+V(x_1+2,S'_1(x_1))-V(x_1+1,S'_1(x_1))\leqslant 0.$$

By definition, $S'_1(x_1+1)$ is the least value to satisfy

$$r_1+V(x_1+2,S'_1(x_1+1))-V(x_1+1,S'_1(x_1+1))\leqslant 0.$$

After comparing the above two inequalities, we have $S'_1(x_1)\geqslant S'_1(x_1+1)$. Hence, $S'_1(x_1)$ is decreasing in x_1. Analogously, we can show that $S''_1(x_1)$ is decreasing in x_1.

$S'_1(x_1)$ and $S''_1(x_1)$ have exactly one intersection point for the reason that the slope of $S'_1(x_1)$ is less than or equal to −1 while the slope of $S''_1(x_1)$ is greater than or equal to −1. Readers may refer to [40] for detailed argument on this result.

Based on the above result, as in Fig. 2, the left part of $S'_1(x_1)$(the solid curve on the left side of the intersection point) and the right part of $S''_1(x_1)$(the solid curve on the right side of the intersection point) are combined to form $S_1(x_1)$ which is also decreasing in x_1.

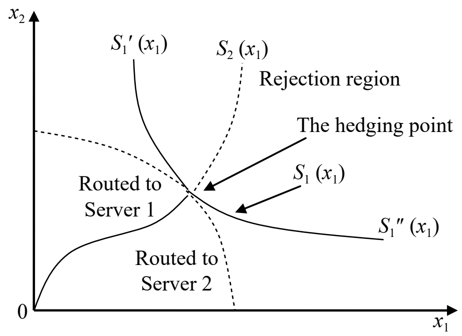

Fig. 2 Hedging point policy for admission and routing controls

Other results are proved analogously. □

The above lemma states that S_i, L_i, and G_i (i = 1, 2) are monotone in the state variables. From now on, we can call them switching curves because each of them partitions the state space into two distinct decision regions. L_i and G_i (i = 3, 4) are the degenerate switching functions characterized by a hedging point, i.e., the so called critical point or threshold in other papers, which segment one axis (horizontal or vertical) into two decision parts. In Fig. 2, the solid curve $S_1(x_1)$ and the solid part of $S_2(x_1)$ partition the whole state space into three decision regions and the intersection point of $S_1(x_1)$ and $S_2(x_1)$ is called the hedging point. The decision regions in Fig. 3 are interpreted in the same manner. Next, we demonstrate that the hedging point policy as illustrated in Fig. 2 and Fig. 3 is optimal.

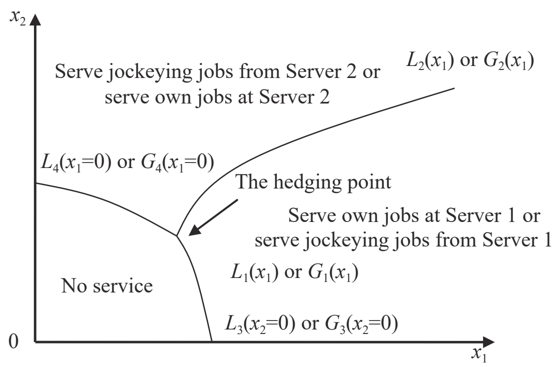

Fig. 3 Hedging point policy for service and jockeying controls

Theorem 1 The optimal policy for admission and routing controls is a hedging point policy characterized by the monotone switching curves $S_1(x_1)$ and $S_2(x_1)$ (solid curves in Fig. 2); the optimal policy for service and jockeying controls at Server 1 is a hedging point policy characterized by the monotone switching curves $L_1(x_1)$ and $L_2(x_1)$ (solid curves in Fig. 3) while $L_3(x_2 = 0)$ and $L_4(x_1 = 0)$ are for the degenerating cases when the states lie on the axes; the optimal policy for service and jockeying controls at Server 2 is also a hedging point policy characterized by the monotone switching curves $G_1(x_1)$ and $G_2(x_1)$ (solid curves in Fig. 3) while $G_3(x_2 = 0)$ and $G_4(x_1 = 0)$ are for the degenerating cases when the states lie on the axes. Given a state (x_1, x_2), the optimal control actions are prescribed by the optimal policy as follows:

(i) Regarding the admission and routing controls, an incoming job will be accepted if and only if $x_2 < S_1(x_1)$; otherwise reject it. An accepted job is routed to Server 1 if $x_2 \geqslant S_2(x_1)$; otherwise, it is routed to Server 2.

(ii) For the service and jockeying controls at Server 1, there are four cases.

First, when $x_1 > 0$ and $x_2 > 0$, the server remains active if $x_2 \geqslant L_1(x_1)$; otherwise, stays idle. The active server processes the job from its own queue if $x_2 < L_2(x_1)$; otherwise, processes a jockeying job from another queue. Second, when $x_1 > 0$ and $x_2 = 0$, the server keeps serving its own jobs if $x_1 \geqslant L_3(x_2 = 0)$; otherwise, stays idle. Third, when $x_1 = 0$ and $x_2 > 0$, the server keeps serving the jockeying jobs if $x_2 \geqslant L_4(x_1 = 0)$; otherwise, stays idle. Fourth, when $x_1 = 0$ and $x_2 = 0$, the server stays idle.

(iii) For the service and jockeying controls at Server 2, there are four cases.

First, when $x_1 > 0$ and $x_2 > 0$, the server remains active if $x_2 \geqslant G_1(x_1)$; otherwise, stays idle. The active server processes the job from its own queue if $x_2 \geqslant G_2(x_1)$; otherwise, processes a jockeying job from another queue. Second, when $x_1 > 0$ and $x_2 = 0$, the server keeps serving the jockeying jobs if $x_1 \geqslant G_3(x_2 = 0)$; otherwise, stays idle. Third, when $x_1 = 0$ and $x_2 > 0$, the server keeps serving its own jobs if $x_2 \geqslant G_4(x_1 = 0)$; otherwise, stays idle. Fourth, when $x_1 = 0$ and $x_2 = 0$, the server stays idle.

Proof For the admission and routing controls in (i), from Lemma 3, $S_1(x_1)$ is decreasing in x_1. According to the definition of $S_1(x_1)$, the submodularity and concavity of the optimal value function ensure $r_1 + V(\boldsymbol{x} + \boldsymbol{e}_1) - V(\boldsymbol{x}) \leqslant 0$ and $r_2 + V(\boldsymbol{x} + \boldsymbol{e}_2) - V(\boldsymbol{x}) \leqslant 0$ whenever $x_2 \geqslant S_1(x_1)$. Hence, it is optimal to reject the job; otherwise accept some job whenever $x_2 < S_1(x_1)$; moreover, when $x_2 < S_1(x_1)$ and $x_2 \geqslant S_2(x_1)$, we have $V(\boldsymbol{x} + \boldsymbol{e}_1) - V(\boldsymbol{x} + \boldsymbol{e}_2) + r_1 - r_2 \geqslant 0$, suggesting that it is optimal to route the accepted job to Server 1; when $x_2 < S_1(x_1)$ and $x_2 < S_2(x_1)$, the inequality $V(\boldsymbol{x} + \boldsymbol{e}_1) - V(\boldsymbol{x} + \boldsymbol{e}_2) + r_1 - r_2 < 0$ holds, implying that it is optimal to route the accepted job to Server 2. (ii) and (iii) on the service and jockeying controls are proved analogously. □

4. System analysis

In this section, we demonstrate that the system with jockeying control performs no worse than that without jockeying. Here, the model for the system without jockeying has the same parameters, i.e., arrival rate, service rates, revenues, holding costs and service costs, as those in Section 2.

Define U as the optimal value function for the model without jockeying in which the operators T_1 and T_2 should be removed of the jockeying control terms. Hence, the modified operators denoted by $\bar{T}_1$, $\bar{T}_2$, and $\bar{T}$ are

$$(\bar{T}_1 U)(\boldsymbol{x}) = \begin{cases} \max\{U(\boldsymbol{x}-\boldsymbol{e}_1)-c_1, U(\boldsymbol{x})\}, & x_1 > 0 \\ U(\boldsymbol{x}), & x_1 = 0 \end{cases},$$

$$(\bar{T}_2 U)(\boldsymbol{x}) = \begin{cases} \max\{U(\boldsymbol{x}-\boldsymbol{e}_2)-c_2, U(\boldsymbol{x})\}, & x_2 > 0 \\ U(\boldsymbol{x}), & x_2 = 0 \end{cases},$$

and

$$\bar{T}U = -h + \lambda T_0 V + \mu_1 \bar{T}_1 U + \mu_2 \bar{T}_2 U.$$

Here, $\bar{T}$ takes the form after the time is rescaled to achieve $\alpha + \Lambda = 1$. Notice that Λ herein is the same one as that in Section 2. This is critical for comparison. Thus, the Bellman equation is written as

$$U = \bar{T}U. \tag{4}$$

Next, we show the optimal value function V associated with jockeying control is no less than U. First, we have the following lemma:

Lemma 4 $TV \geqslant \bar{T}U$ if $V(\boldsymbol{x}) \geqslant U(\boldsymbol{x})$ for all $\boldsymbol{x} \in \boldsymbol{X}$.

Proof Suppose $V(\boldsymbol{x}) \geqslant U(\boldsymbol{x})$ for all $\boldsymbol{x} \in \boldsymbol{X}$. And we will show $T_1 V \geqslant \bar{T}_1 U$, $T_2 V \geqslant \bar{T}_2 U$, and $T_0 V \geqslant T_0 U$.

To verify $T_1 V \geqslant \bar{T}_1 U$, we prove it in four cases. First, for $x_1 > 0$ and $x_2 > 0$, since $V(\boldsymbol{x}) \geqslant U(\boldsymbol{x})$ and $V(\boldsymbol{x} - \boldsymbol{e}_1) \geqslant U(\boldsymbol{x} - \boldsymbol{e}_1)$, we have

$$\max\{V(\boldsymbol{x}-\boldsymbol{e}_1)-c_1, V(\boldsymbol{x}-\boldsymbol{e}_2)-c_{21}, V(\boldsymbol{x})\} \geqslant \max\{U(\boldsymbol{x}-\boldsymbol{e}_1)-c_1, U(\boldsymbol{x})\}.$$

For $x_1 > 0$ and $x_2 = 0$, obviously, we have

$$\max\{V(\boldsymbol{x}-\boldsymbol{e}_1)-c_1, V(\boldsymbol{x})\} \geqslant \max\{U(\boldsymbol{x}-\boldsymbol{e}_1)-c_1, U(\boldsymbol{x})\}.$$

For $x_1 = 0$ and $x_2 > 0$, it is evident that

$$\max\{V(\boldsymbol{x}-\boldsymbol{e}_2)-c_{21}, V(\boldsymbol{x})\} \geqslant U(\boldsymbol{x}).$$

For $x_1 = 0$ and $x_2 = 0$, simply, $V(\boldsymbol{x}) \geqslant U(\boldsymbol{x})$.

$T_2 V \geqslant \bar{T}_2 U$ can be proved analogously.

For $T_0 V \geqslant T_0 U$, since $V(x) \geqslant U(x)$, $V(\boldsymbol{x} + \boldsymbol{e}_1) \geqslant U(\boldsymbol{x} + \boldsymbol{e}_1)$, and $V(\boldsymbol{x} + \boldsymbol{e}_2) \geqslant U(\boldsymbol{x} + \boldsymbol{e}_2)$, we have

$$\max\{r_1 + V(\boldsymbol{x}+\boldsymbol{e}_1), r_2 + V(\boldsymbol{x}+\boldsymbol{e}_2), V(\boldsymbol{x})\} \geqslant \max\{r_1 + U(\boldsymbol{x}+\boldsymbol{e}_1), r_2 + U(\boldsymbol{x}+\boldsymbol{e}_2), U(\boldsymbol{x})\}.$$

Finally, because both T and $\bar{T}$ are positive linear combinations of T_0, T_1, T_2, $\bar{T}_1$, and $\bar{T}_2$, the result applies. □

The above lemma states that the inequality is preserved after the operators T and $\bar{T}$ act on V and U, respectively. Then we can prove our main result.

Proposition 2 $V(\boldsymbol{x}) \geqslant U(\boldsymbol{x})$ for all $\boldsymbol{x} \in \boldsymbol{X}$.

Proof We arbitrarily choose $V_0(\boldsymbol{x})$ and $U_0(\boldsymbol{x})$, for instance, $V_0(\boldsymbol{x}) = U_0(\boldsymbol{x}) = 0$, to make $V_0(\boldsymbol{x}) \geqslant U_0(\boldsymbol{x})$ for all $\boldsymbol{x} \in \boldsymbol{X}$. From Lemma 4, $TV_0 \geqslant \bar{T}U_0$. Further, repeatedly applying T and $\bar{T}$ to V_0 and U_0, respectively, yields $T^n V_0 \geqslant \bar{T}^n U_0$ for all n. By the standard result, $T^n V_0 \to V$ and $\bar{T}^n U_0 \to U$ in a manner of point-wise convergence as $n \to \infty$. Hence, $V(\boldsymbol{x}) \geqslant U(\boldsymbol{x})$ for all $\boldsymbol{x} \in \boldsymbol{X}$. □

The above result demonstrates the value of jockeying by showing that the system with jockeying control performs at least no worse than that without jockeying. Next, we show that the system performance is monotone with respect to some of its parameters.

Proposition 3 $V \uparrow \lambda \uparrow \mu_1 \uparrow \mu_2 \downarrow c_{12} \downarrow c_{21}$

Proof First, we show $V \uparrow \lambda$, i.e., the optimal values are nondecreasing in the arrival rate. Let the system update arrival rate $\lambda' \geqslant \lambda$ and other parameters remain unchanged from (1) in Section 2. Now, for model formulations, both systems use the same uniformized transition rate $\Lambda' = \lambda' + \mu_1 + \mu_2$. Without loss of generality, rescale time to achieve $\alpha + \Lambda' = 1$. Then the Bellman equation for the original system is modified as follows:

$$V = TV \tag{5}$$

where

$$TV = -h + \lambda T_0 V + \mu_1 T_1 V + \mu_2 T_2 V + (\lambda' - \lambda)V.$$

The Bellman equation for the system with a new arrival rate is written as

$$V' = T'V' \tag{6}$$

where

$$T'V' = -h + \lambda' T_0 V' + \mu_1 T_1 V' + \mu_2 T_2 V'.$$

In (5) and (6), T_0, T_1, and T_2 remain the same forms as those in Section 2.

Next, to show $V' \geqslant V$, we first prove that $T'V' \geqslant TV$ if $V' \geqslant V$. The result is readily proved by showing

$$T'V' - TV = \lambda(T_0 V' - T_0 V) + \mu_1(T_1 V' - T_1 V) + \mu_2(T_2 V' - T_2 V) + (\lambda' - \lambda)(T_0 V' - V) \geqslant 0.$$

Then let $V_0'(\boldsymbol{x}) = V_0(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \boldsymbol{X}$. Thus, $T'V_0' \geqslant TV_0$. Apply T' and T repeatedly, leading to $T'^n V_0' \geqslant T^n V_0$ for all n. As $n \to \infty$, we have $V' \geqslant V$, that is, $V \uparrow \lambda$.

The results that V is nondecreasing in service rates and nonincreasing in job jockeying costs can be proved in the same manner as the above argument. □

5. Numerical studies

The following example is used to illustrate the hedging point policy. Jobs arrive according to a Poisson process with the rate $\lambda = 2$. The service rates of two servers are set to $\mu_1 = 2$ and $\mu_2 = 2$, which assures $\lambda < \mu_1 + \mu_2$. The revenues from the accepted jobs are $r_1 = 7$ for Server 1 and $r_2 = 7$ for Server 2. The unit holding cost rates are assumed to be $h_1 = 1$ and $h_2 = 1$. The service costs at two respective servers are $c_1 = 2$ and $c_2 = 2$. The jockeying costs are $c_{12} = 3$ and $c_{21} = 3$, respectively.

For Bellman equations, typically, we can apply the standard methods of value iteration, policy iteration and linear programming. However, the value iteration algorithm is in general the best computational method for solving large-scale Markov decision problems [41]. Here, to solve (1), we employ the value iteration algorithm, i.e., the dynamic programming approach, which reads

$$V(\boldsymbol{x}) = \lim_{k\to\infty}(T^k V_0)(\boldsymbol{x}), \quad \boldsymbol{x} \in \boldsymbol{X}.$$

That is, if we apply the operator T in (2) to any bounded function V_0 repeatedly, we can finally attain the optimal value function V. For our example, for simplicity, let $V_0(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \boldsymbol{X}$. The continuous-time discount rate α is usually a very small value. However, to achieve a fast convergence of the value iteration algorithm for our model, we let $\alpha = 0.1$.

For the large state space of $\mathbf{Z}_+^2$, we need to truncate the state space in computation. If the target optimal decisions associated with the discrete states $[\delta_1,\delta_2] \times [\delta_3,\delta_4]$, where the integers $\delta_2 > \delta_1 \geqslant 0$ and $\delta_4 > \delta_3 \geqslant 0$, are desired, we can apply n number of value iterations from an initial truncated state space $[(\delta_1 - n)^+, \delta_2 + n] \times [(\delta_3 - n)^+, \delta_4 + n]$ to achieve our goal since each iteration causes a state transition which could be a service completion at Server 1, i.e., $(\delta_1 - 1)$, a service completion at Server 2, i.e., $(\delta_3 - 1)$, a job routed to Server 1, i.e., $(\delta_2 + 1)$, and a job routed to Server 2, i.e., $(\delta_4 + 1)$.

The uniformization rate is $\Lambda = \lambda + \mu_1 + \mu_2 = 6$ and hence $\alpha + \Lambda = 6.1$. From (2),

$$TV = \frac{-h}{\alpha+\Lambda} + \frac{\lambda}{\alpha+\Lambda}T_0V + \frac{\mu_1}{\alpha+\Lambda}T_1V + \frac{\mu_2}{\alpha+\Lambda}T_2V =$$
$$-\frac{x_1+x_2}{6.1} + \frac{2}{6.1}T_0V + \frac{2}{6.1}T_1V + \frac{2}{6.1}T_2V.$$

Applying the above operator T to $V_0(\boldsymbol{x}) = 0$ repeatedly with 500 times, we attain $(T^{500}V_0)(\boldsymbol{x})$ for $\boldsymbol{x} \in [0,15] \times [0,15]$, which is our target state. Then compare two optimal values at the origin point, i.e., $(T^{500}V_0)(0,0) = 22.6987$ and $(T^{499}V_0)(0,0) = 22.6984$ and find that $((T^{500}V_0)(0,0) - (T^{499}V_0)(0,0))/(T^{499}V_0)(0,0) = 0.0015\%$ which is sufficiently small. Hence, $(T^{500}V_0)(\boldsymbol{x})$ is a suitable approximation to $V(\boldsymbol{x})$. Correspondingly, the derived optimal decisions are computed and listed in Fig. 4, Fig. 5, and Fig. 6, respectively.

x_2 \ x_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
14	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
11	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
10	1	1	1	1	1	1	1	1	1	1	1	2	0	0	0	0
9	1	1	1	1	1	1	1	1	1	1	2	2	2	0	0	0
8	1	1	1	1	1	1	1	1	1	2	2	2	2	2	0	0
7	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	0
6	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
5	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
4	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2
2	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Fig. 4 Decisions for admission and routing controls

x_2 \ x_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 5 Decisions for service and jockeying controls at Server 1

x_2 \ x_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2

Fig. 6 Decisions for service and jockeying controls at Server 2

In Fig. 4, “0”, “1”, and “2” denote the decisions of rejecting the job, routing a job to Server 1, and routing a job to Server 2, respectively. Whenever a job arrives, if the state $\boldsymbol{x}$ is found to be, for instance, at (3, 5), i.e., three jobs in Queue 1 and five jobs in Queue 2, the corresponding decision in Fig. 4 is “1”, which suggests that we should accept the job and route it to Server 1.

The decisions of service and jockeying controls are listed in Fig. 5 and Fig. 6 where “0”, “1”, and “2” indicate stopping service, serving own jobs, and serving a jockeying job, respectively.

The structural properties (i) and (iii) of $V(x)$ are illustrated in Fig. 7 and Fig. 8, respectively. For instance, in Table 1, the difference $V(\boldsymbol{x}+\boldsymbol{e}_1) - V(\boldsymbol{x})$ is monotone non-increasing in x_1 and x_2, respectively. The similar interpretation applies to Fig. 8. The diagram of property (ii) is analogous to that of property (i) and hence it is not shown here.

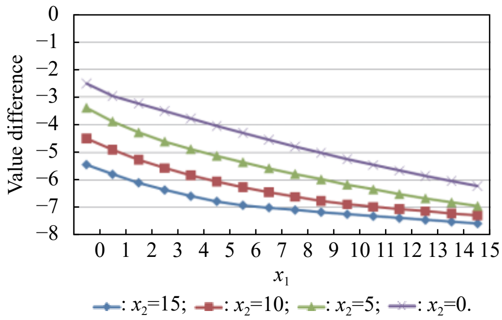

Fig. 7 Monotone property of the value difference $V(\boldsymbol{x}+\boldsymbol{e}_1) - V(\boldsymbol{x})$

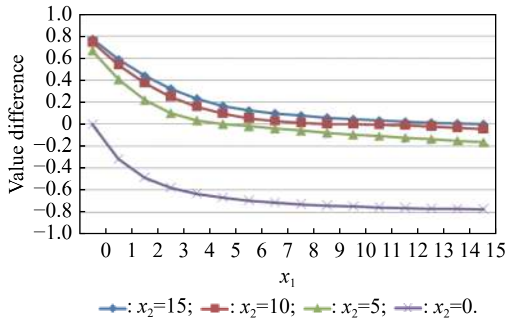

Fig. 8 Mmonotone property of the value difference $V(\boldsymbol{x}+\boldsymbol{e}_1) - V(\boldsymbol{x}+\boldsymbol{e}_2)$

Furthermore, we investigate the system performance associated with various system parameters. Comparisons of the optimal values at the origin are made under different scenarios and the corresponding results are presented in Table 1.

Table 1 System performance with different paremeters

Case	Paremeters of the system											$V(0,0)$	Variation/%
	r_1	r_2	c_1	c_2	c_{12}	c_{21}	h_1	h_1	λ	μ_1	μ_2		
Base case	7	7	2	2	3	3	1	1	2	2	2	22.699	0
No jockeying	7	7	2	2	0	0	1	1	2	2	2	16.372	−27.87
1	7	7	2	2	2.1	2.1	1	1	2	2	2	26.395	16.28
2	7	7	2	2	3.5	3.5	1	1	2	2	2	21.47	−5.42
3	7	7	2	2	4	4	1	1	2	2	2	20.332	−10.43
4	7	7	2	2	4.5	4.5	1	1	2	2	2	19.275	−15.09
5	7	7	2	2	5	5	1	1	2	2	2	18.322	−19.28
6	7	7	2	2	3	3	1	1	1	2	2	−14.333	−163.15
7	7	7	2	2	3	3	1	1	1.5	2	2	5.205	−77.07
8	7	7	2	2	3	3	1	1	2.5	2	2	37.187	63.83
9	7	7	2	2	3	3	1	1	3	2	2	47.86	110.84
10	7	7	2	2	3	3	1	1	3.5	2	2	55.071	142.62
11	7	7	2	2	3	3	1	1	2	1.5	1.5	1.228	−94.59
12	7	7	2	2	3	3	1	1	2	1.5	2	13.327	−41.29
13	7	7	2	2	3	3	1	1	2	2.5	2	29.691	30.8
14	7	7	2	2	3	3	1	1	2	3	2	34.861	53.58
15	7	7	2	2	3	3	1	1	2	3	3	39.907	75.81

From Table 1, we can identify that the system performance of no jockeying worsens by 27.87% in contrast with the case of jockeying. For Cases 1−5, we only change the jockeying costs while other parameters are fixed. It is obvious that the optimal values are decreasing with respect to the jockeying costs. For Cases 6−10, the arrival rates are different while other parameters remain unchanged. The results show that the optimal value is increasing in the arrival rate. This is intuitive because more incoming jobs could bring in more revenues. For Cases 11−15, we examine the system performance with varied service rates. Clearly, the performance improves as the service rate increases.

6. Conclusions

In this paper, we study the optimal control of queueing systems with a Poisson arrival of jobs and two parallel exponential servers. With the structure properties of the optimal value function, we characterize the optimal control policy as a hedging point policy of which two monotone switching curves intersected at the hedging point and typically partitioned the state space into three decision zones. The simple structure of the optimal policy is convenient for implementing the control actions in practice.

One important contribution of this paper is the incorporation of jockeying controls into queueing systems in addition to the commonly used job admission, routing, and service controls. This queueing control system has not been addressed by previous researchers. Furthermore, our numerical studies confirm the structural properties of the optimal value function and the structure of the hedging point policy and demonstrate the value of jockeying control for our queueing system. Besides, we compare system performances associated with varied system parameters and verify some monotonicity results which might offer insights into the design and operation of such queueing systems.

Although we give an analysis on the queueing system of two parallel servers, it would be interesting to study jockeying control problems with more than two servers.

References

[1] NAOR P. On the regulation of queue size by levying tolls. Econometrica, 1969, 37(1): 15–24.

[2] LIPPMAN S, STIDHAM S. Individual versus social optimization in exponential congestion systems. Operations Research, 1977, 25(2): 233–247.

[3] STIDHAM S. Socially and individually optimal control of arrivals to a GI/M/1 queue. Management Science, 1978, 24(15): 1598–1610.

[4] VAN NUNEN J A E E, PUTERMAN M L. Computing optimal control limits for GI/M/s queueing systems with controlled arrivals. Management Science, 1983, 29(6): 725–734.

[5] EPHREMIDES A, VARAIYA P, WALRAND J. A simple dynamic routing problem. IEEE Trans. on Automatic Control, 1980, 25(4): 690–693.

[6] HAJEK B. Optimal control of two interacting service stations. IEEE Trans. on Automatic Control, 1984, 29(6): 491–498.

[7] WEBER R R. On the optimal assignment of customers to parallel servers. Journal of Applied Probability, 1978, 15(2): 406–413.

[8] KRISHNAN K R. Joining the right queue: a Markov decision rule. Proc. of the 26th IEEE Conference on Decision and Control, 1987: 1863–1868.

[9] WEBER R R, STIDHAM S. Optimal control of service rates in networks of queues. Advances in Applied Probability, 1987, 19(1): 202–218.

[10] YAO D D, SCHECHNER Z. Decentralized control of service rates in a closed Jackson network. IEEE Trans. on Automatic Control, 1989, 34(2): 236–240.

[11] DAVIS E L. Optimal control of arrivals to a two-server queueing system with separate queues. North Carolina, USA: North Carolina State University, 1977.

[12] ABDEL-GAWAD E F. Optimal control of arrivals and routing in a network of queues. North Carolina, USA: North Carolina State University, 1984.

[13] LI Q L, MA J Y, FAN R N, et al. An overview for Markov decision processes in queues and networks. LI Q , WANG J, YU H B, ed. Stochastic models in reliability, network security and system safety. Singapore: Springer, 2009.

[14] CAI J H, WANG L P, ZHOU G G. Supply chain coordination mechanisms under flexible contracts. Journal of Systems Engineering and Electronics, 2010, 21(3): 440–448.

[15] LI H Y, YOU T H, LUO X Y. Collaborative supply chain planning under dynamic lot sizing costs with capacity decision. Journal of Systems Engineering and Electronics, 2011, 22(2): 247–256.

[16] WEN M L, LU B H, LI S Y, et al. Location and allocation problem for spare parts depots on integrated logistics support. Journal of Systems Engineering and Electronics, 2019, 30(6): 1252–1259.

[17] YUE F, SONG S J, JIA P, et al. Robust single machine scheduling problem with uncertain job due dates for industrial mass production. Journal of Systems Engineering and Electronics, 2020, 31(2): 350–358.

[18] XU S H, ZHAO Y Q. Dynamic routing and jockeying controls in a two-station queueing system. Advances in Applied Probability, 1996, 28(4): 1201–1226.

[19] ADAN I J B F, WESSELS J, ZIJM W H M. Analysis of the asymmetric shortest queue problem with threshold jockeying. Communications in Statistics. Stochastic Models, 1991, 7(4): 615–627.

[20] ADAN I J B F, WESSELS J, ZIJM W H M. Matrix-geometric analysis of the shortest queue problem with threshold jockeying. Operations Research Letters, 1993, 13(2): 107–112.

[21] ZHAO Y Q, GRASSMANN W K. Queueing analysis of a jockeying model. Operations Research, 1995, 43(3): 520–529.

[22] ZHAO Y Q, GRASSMANN W K. A solution of the shortest queue model with jockeying-in terms of traffic intensity ρ. Naval Research Logistics, 1990, 37(5): 773–787.

[23] DISNEY R L, MITCHELL W E. A solution for queues with instantaneous jockeying and other customer selection rules. Naval Research Logistics Quarterly, 1970, 17(3): 315–325.

[24] ELSAYED E A, BASTANI A. General solutions of jockeying problem. European Journal Operational Research, 1985, 22(3): 387–396.

[25] HAIGHT F A. Two queues in parallel. Biometrika, 1958, 45(3/4): 401–410.

[26] KAO E P C, LIN C. A matrix-geometric solution of the jockeying problem. European Journal of Operational Research, 1990, 44(1): 67–74.

[27] KOENIGSBERG E. On jockeying in queues. Management Science, 1966, 12(5): 412–436.

[28] SAKUMA Y. Asymptotic behavior for MAP/PH/c queue with shortest queue discipline and jockeying. Operations Research Letters, 2010, 38(1): 7–10.

[29] RAVID R. A new look on the shortest queue system with jockeying. Probability in the Engineering and Informational Sciences, 2021, 35(3): 557–564.

[30] JEGANATHAN K, SUMATHI J, MAHALAKSHMI G. Markovian inventory model with two parallel queues, jockeying and impatient customers. Yugoslav Journal of Operations Research, 2016, 26(4): 467–506.

[31] STAGJE W. A queueing system with two parallel lines, cost-conscious customers, and jockeying. Communications in Statistics-Theory and Methods, 2009, 38(16/17): 3158–3169.

[32] TARABIA A M K. Analysis of two queues in parallel with jockeying and restricted capacities. Applied Mathematical Modelling, 2008, 32(5): 802–810.

[33] BAYKASOGLU A, DURMUSOGLU Z D. Flow time analyses of a simulated flexible job shop by considering jockeying. International Journal of Advanced Manufacturing Technology, 2012, 58(5/8): 693–707.

[34] CHALESHTORI A E, JAHANI H, AGHAIE A. Bi-objective optimization approach to a multi-layer location-alloca-

tion problem with jockeying. Computers & Industrial Engineering, 2020, 149: 106740.
[35] DOWN D G, LEWIS M E. Dynamic load balancing in parallel queueing systems: stability and optimal control. European Journal of Operational Research, 2006, 168(2): 509–519.
[36] ROSBERG Z, PENG Y, FU J, et al. Insensitive job assignment with throughput and energy criteria for processor-sharing server farms. IEEE/ACM Trans. on Networking, 2014, 22(4): 1257–1270.
[37] DEHGHANIAN A, KHAROUFEH J P, MODARRES M. Strategic dynamic jockeying between two parallel queues. Probability in the Engineering and Informational Sciences, 2016, 30(1): 41–60.
[38] LIPPMAN S. Applying a new device in the optimization of exponential queueing systems. Operations Research, 1975, 23(4): 687–710.
[39] PUTERMAN M L. Markov decision processes: discrete stochastic dynamic programming. New Jersey: John Wiley & Sons, 1994.
[40] LIN B. Dynamic revenue management of airline and production systems. Singapore: Nanyang Technological University, 2008.
[41] TIJMS H C. A first course in stochastic models. West Sussex, England: John Wiley & Sons, 2003.

Biographies

LIN Bing was born in 1969. He received his B.S. degree from Fudan University in 1992, MBA from Tsinghua University in 2003, and Ph.D. degree in operations management from Nanyang Technological University, Singapore in 2008. Currently, he is a lecturer in Jiangsu Normal University. His research interests are modeling and optimization of queueing systems, production/inventory systems, and supply chain systems.
E-mail: linbing_jsnu@163.com

LIN Yuchen was born in 2000. She is currently an undergraduate student in electrical engineering at the Wenzheng College of Soochow University. She has publications on logistics technology and logistics science and technology. Her research interests are wireless charging technology, smart logistics system, and machine learning.
E-mail: yclin_ee@163.com

BHATNAGAR Rohit was born in 1960. He received his B.Tech. degree from Indian Institute of Technology (BHU), India in 1983, MBA from the India Institute of Management, Bangalore, India in 1985, and Ph.D. degree in operations management from McGill University, Canada in 1994. Currently, he is an associate professor in Nanyang Business School, Nanyang Technological University, Singapore. His research interests are design and planning of manufacturing and supply chain systems.
E-mail: arbhatnagar@ntu.edu.sg