American Journal of Operations Management and Information Systems
2021; 6(3): 29-34
http://www.sciencepublishinggroup.com/j/ajomis
doi: 10.11648/j.ajomis.20210603.11
ISSN: 2578-8302 (Print); ISSN: 2578-8310 (Online)

A Note on Principal-Agent Problem in a Stochastic System

Jagan Jacob

Business Analytics and Information Systems, Xavier University, Cincinnati, USA

Email address:
jacobj6@xavier.edu

To cite this article:
Jagan Jacob. A Note on Principal-Agent Problem in a Stochastic System. *American Journal of Operations Management and Information Systems*. Vol. 6, No. 3, 2021, pp. 29-34. doi: 10.11648/j.ajomis.20210603.11

Received: July 27, 2021; **Accepted**: August 5, 2021; **Published**: August 11, 2021

Abstract: We consider a principal (e.g., a ridesharing platform such as Uber or Lyft) who receives two types of jobs (e.g., passengers requesting solo or shared rides) according to a Poisson process. The principal first decides which jobs to admit and then assigns an agent (e.g., driver) to perform them. The agent who is assigned the job has preference between the two types of jobs. The agent can independently decide whether to accept or reject a job which is assigned to them. The principal and the agent receive different rewards from each job thus resulting in incentives misalignment. The research questions are: (1) which job(s) should the principal admit? (2) How much should the principal pay the agent? To answer these questions, we model the agent as an *M / M / 1* loss system. Using a Markov decision process and dynamic programming, we find the optimal wage the principal should pay the agent and a threshold admission policy (also known as trunk-reservation or switching-curve policy). Prior literature did not consider two players (agent and principal) with misaligned objectives and *each* making dynamic decisions. We contribute to the literature by adding another layer of decision making and by introducing server (agent) independence wherein the servers have preferences regarding the type of job they wish to perform.

Keywords: Revenue Management, Queueing, Principal-Agent, Stochastic

1. Introduction

Consider a *principal* that assigns jobs to an *agent*. The principal first decide whether to admit a job and then decide how much to pay the agent to do the job. There are two types of jobs and the agent has preference between them. The agent first observes the type of job and the pay offered by the principal and then decides whether to do it or not. The research problem is to maximize the principal's profit when there is misalignment between the players' incentives. Both the principal and the agent are strategic, risk-neutral, and independent. This means both aim to maximize their profit/rewards and will participate if and only if their expected reward is at least as high as the utility from external options. The specific research questions are as follows: (1) which job(s) should the principal admit? (2) How much to should the principal pay the agent?

To motivate the problem setting, consider the case of a ride-sharing platform such as Uber or Lyft (see Figure 1). The ride-share platform (principal) gets requests from passengers (jobs) and then match them with drivers (agents). Drivers are independent contractors who use their private cars to transport passengers on the ride-share platform's behalf. Passengers can ride alone (solo) or share (pool) the ride with other passengers (e.g., UberPOOL, Lyft Line). It has been reported that drivers generally prefer solo rides over pooled rides. According to Uber's head of driver product, "We heard from drivers that Pool feels like extra work without additional pay. Multiple pickups in particularly made Pool trips more challenging." (Source: *Los Angeles Times*).

Drivers experience a higher inconvenience from a pooled ride because it is more stressful and a hassle to pick-up and drop-off two different passengers in a single trip. While a pooled ride is cheaper than a solo ride, the expected travel time for the former is higher than that of the latter. This is due to the additional pick-ups/drop-offs and possible detours during a pooled ride. Moreover, pooled passengers may be strangers and sharing the space with each other may cause discomfort between them, creating an unpleasant service experience. The driver's rating could be negatively affected if one of the customers disliked the service due to co-passenger's bad behavior.

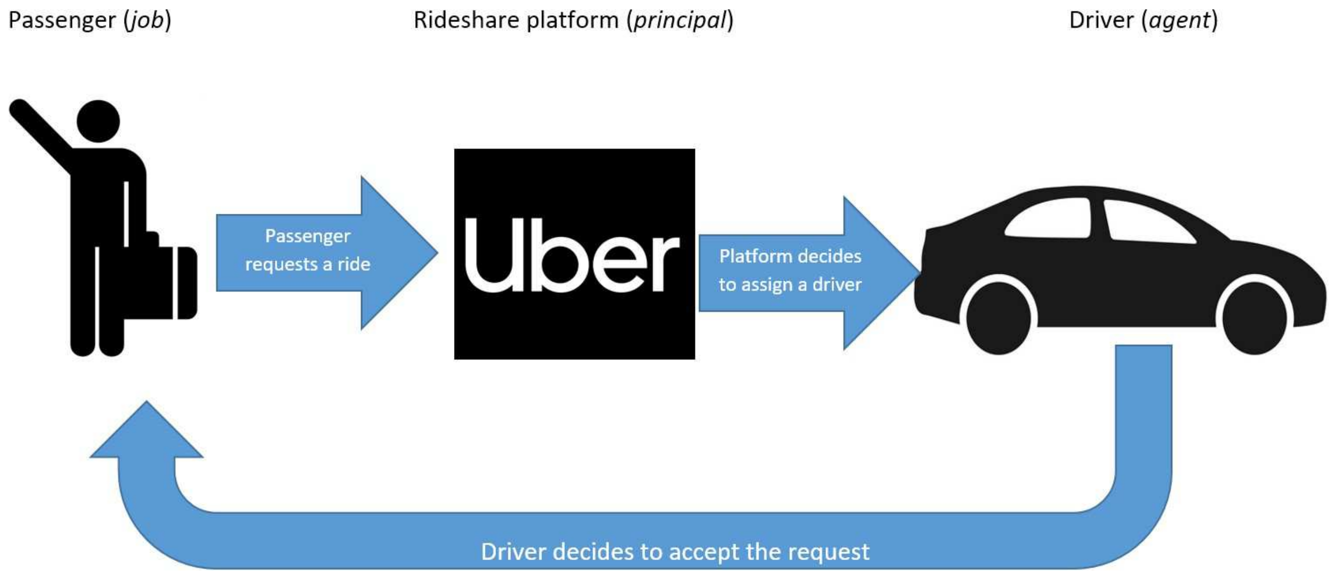

Figure 1. *An example of principal-agent setting.*

Therefore, a driver may have an incentive to not accept a pooled ride request and instead wait for a solo request. In doing so, the driver forgoes the certain reward from the pooled ride she just rejected in hopes of getting the more preferred choice (solo ride) soon. Meanwhile, the ride-share platform does not experience any inconvenience cost that the drivers incur when serving pooled rides. It may so happen that the ride-share platform may even prefer a pooled ride over a solo ride depending on the net profit (=total fare−driver's wage) per unit time. This presents misalignment in the incentives of the driver and the ride-share platform. In this context, the problem is to maximize the ride-share platform's profit given the drivers' preference between solo and pooled rides.

Another example of principal-agent misalignment when the agent is independent is the case of a journal editor assigning a manuscript to a referee/reviewer for peer-review. The editor (principal) first decide whether to desk-reject a paper (job). The editor then assigns a referee (agent) to review the paper based on the referee's area of expertise. The referee has preference over the type of papers to review (e.g., some referees prefer theoretical papers over empirical ones) and can choose to accept/reject an invitation to review. The referee's personal preference and decision to accept the invitation to review is independent of the editor. The editor is concerned with managing the cycle-time and throughput.

2. Literature Review

Many papers have tackled admission control problems under different settings [3-10, 15-19]. Our paper in particular relates to Altman et al. [1], Ormeci et al. [11] and Savin et al. [13]. Altman et al. [1] study admission control of calls made by multiple classes of customers with no waiting room. They build a discrete model and use dynamic programming and fluid approximation to develop structural properties of optimal policies. They introduced the so-called "threshold policy" which states that in a system with two customer classes (phone calls to a call-center), a call of one type should be admitted only if the number of calls of the other type is below a threshold. Ormeci et al. [11] conducted a similar study. They consider a Markovian loss queueing system with two classes of customers. Each class has a different service rate and reward. Similar to Altman et al. [1], they also show the existence of a threshold policy. Furthermore, they show the conditions under which there exists a preferred class such that whenever a server is free, any customer belonging to the preferred class is accepted by the system. Compared to the above-mentioned works, a more application-based study was done by Savin et al. [13]. They study the capacity allocation of a car rental business with two types of customers. Using fluid approximation, they develop a computationally efficient heuristic to allocate capacity on the optimal fleet size.

We contribute to the literature by adding another layer of decision making and by introducing server (agent) preference wherein the servers have preferences regarding the type of job they perform. Unlike our work, prior literature did not have two players (agent and principal) with misaligned objectives and *each* making dynamic decisions. We relax the assumption present in most optimal admission control problems that each server performs the task assigned to it.

3. The Agent's Problem

Now we will discuss the model specifics, starting with the agent's problem. Assume there are two types of jobs, indexed by $i = 1, 2$. The agent receives type-i jobs according to a Poisson process with parameter $\lambda_i > 0$. The service-time for a type-i job is distributed exponentially with mean $1/\mu_i > 0$. When the agent is busy doing a job, she receives no additional requests. In other words, there is no waiting room for requests if the agent is busy and therefore, we model the agent as an $M/M/1$ loss system. When doing a job, the agent receives wages at a rate of w per unit time. Without the loss of generality, we normalize the agent's operating cost to zero.

Suppose the agent have preference between the two jobs. For example, suppose the agent dislikes doing one type of job (say, type-2). To capture this, assume the agent incurs an inconvenience cost d per unit time when doing type-2 jobs. Hence, the effective wages of the agent for doing a type-1 job is $w_1 = w$ and a type-2 job is $w_2 = w - d$. Whenever the agent receives a job-request, she must either accept it or reject it right away. Intuitively, she will decline a request if the expected earnings from it is too low or if she finds it better to forego the immediate earnings from it in hopes of getting a more lucrative job soon. Assume earnings are continuously discounted at a rate $0 < \beta < 1$. The agent's problem is to find the optimal action (accept or reject) whenever a new request arrives, such that it maximizes the total expected discounted rewards over an infinite horizon.

At any given time, the agent is in state $j \in \{0, 1, 2\}$, where $j = 0$ indicates when the agent is idle, and $j = 1$ and $j = 2$ indicate when the agent is serving job types- 1 and 2 respectively. When the agent is idle ($j = 0$) and a type-i job arrives, she can either accept or reject it. Denote the action "accept" as $a = 1$, and "reject" as $a = 0$. Note that even though the state evolves over continuous time, the actions are taken only at arrival epochs, and therefore, we can maximize the total expected discounted rewards using a discrete-time Markov Decision Process [12].

To find the equivalent discrete-time system, we use uniformization to re-scale the transition-rates so that each transition happens at regular intervals [6, 14]. Specifically, we define $\gamma = \lambda_1 + \lambda_2 + \mu_1 + \mu_2$ and divide the actual transition rates λ_i and μ_i by γ. Then the transition rates out of state $j = 0$ is $(\lambda_1 + \lambda_2)/\gamma$, out of state $j = 1$ and $j = 2$ are μ_1/γ and μ_2/γ, respectively. Next, we add a "fictitious" transition with rate $1 - (\lambda_1 + \lambda_2)/\gamma$ from state $j = 0$ to itself, and "fictitious" self-transitions with rates $1 - \mu_1/\gamma$ and $1 - \mu_2/\gamma$, respectively, to states $j = 1$ and $j = 2$ (Markov-Chain before and after uniformization are given in Figure 2). Continuous-time discounting at a rate β is equivalent to the system terminating with probability β in the next transition [12]. This implies, at every stage, with probability β, all future rewards are set to zero. So, the maximum transition rate is $\gamma + \beta$.

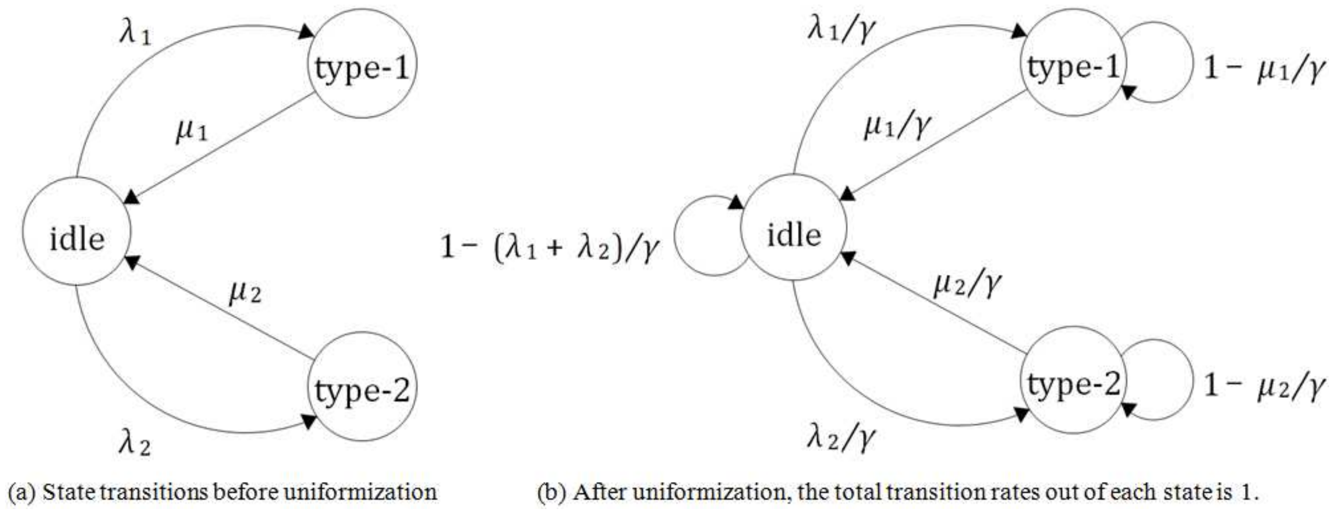

Figure 2. *Uniformization and fictitious transitions.*

The agent makes decisions $a \in \{0, 1\}$ only at arrival epochs. When the agent is busy, she cannot accept any request, meaning, actual decision making occurs only when the agent's state is $j = 0$. Furthermore, at service completions, no actions are necessary. Let v_0 be the optimal total expected discounted earnings starting at a non-arrival event (include actual and fictitious service-completions). Let $v_i, i = 1, 2,$ be the corresponding value starting when a type-i job arrives and the agent is not busy. The optimality equations are as follows:

$$v_0 = \frac{1}{\beta+\gamma}[\lambda_1 v_1 + \lambda_2 v_2 + (\mu_1 + \mu_2)v_0], \tag{1}$$

$$v_i = \frac{1}{\beta+\gamma}\max_{a=0,1} v_i(a), i = 1, 2, \tag{2}$$

where,

$$v_i(a) = \begin{cases} \frac{1}{\beta+\gamma}((w_i + v_0)\mu_i + (\gamma - \mu_i)v_i), & \text{when } a = 1, \\ \frac{\gamma}{\beta+\gamma} v_0, & \text{when } a = 0. \end{cases} \tag{3}$$

Proposition 1 states the optimal policy for the agent, which in turn helps the principal decide the wages to pay so that the agent is incentivized to make the choice desired by the principal.

PROPOSITION 1 [How much to pay the agent?] It is optimal for the agent to (1) accept both job-types if $w_{j'}\mu_{j'} = w_j\mu_j$, or if $w_j\mu_j < w_{j'}\mu_{j'}$ and $w_{j'}\mu_{j'}\lambda_{j'} \le w_j\mu_j(\lambda_{j'} + \beta + \mu_{j'})$, and (2) accept only job-type j' if $w_j\mu_j(\lambda_{j'} + \beta + \mu_{j'}) \le w_{j'}\mu_{j'}\lambda_{j'}$, where $j, j' \in \{1, 2\}$ and $j \neq j'$.

Proof of Proposition 1. For a given policy $\mathcal{P}$, denote the expected discounted infinite-horizon earnings starting in state j as $v_j^{\mathcal{P}}$. If the agent accepts both jobs, denote the policy as $\mathcal{P} = (1,1)$, and if she accepts either only type-1 job or only type-2 job, denote the policies as $\mathcal{P} = (1,0)$ and $\mathcal{P} = (0,1)$ respectively.

If the agent accepts both jobs,

$$v_0^{(1,1)} = \frac{\lambda_2\mu_2w_2(\beta+\mu_1)+\lambda_1\mu_1w_1(\beta+\mu_2)}{\beta(\lambda_1(\beta+\mu_2)+(\beta+\mu_1)(\beta+\lambda_2+\mu_2))}, \tag{4}$$

$$v_1^{(1,1)} = \frac{\mu_1(w_1(\lambda_1(\beta+\mu_2)+\beta(\beta+\lambda_2+\mu_2))+\lambda_2\mu_2w_2)}{\beta(\lambda_1(\beta+\mu_2)+(\beta+\mu_1)(\beta+\lambda_2+\mu_2))}, \tag{5}$$

$$v_2^{(1,1)} = \frac{\mu_2(w_2((\beta+\lambda_2)(\beta+\mu_1)+\beta\lambda_1)+\lambda_1\mu_1w_1)}{\beta(\lambda_1(\beta+\mu_2)+(\beta+\mu_1)(\beta+\lambda_2+\mu_2))}. \tag{6}$$

If the agent accepts only job 1,

$$v_0^{(1,0)} = \frac{\lambda_1\mu_1w_1}{\beta(\beta+\lambda_1+\mu_1)}, \tag{7}$$

$$v_1^{(1,0)} = \frac{\mu_1w_1(\beta+\lambda_1)}{\beta(\beta+\lambda_1+\mu_1)}, \tag{8}$$

$$v_2^{(1,0)} = \frac{\lambda_1\mu_1w_1}{\beta(\beta+\lambda_1+\mu_1)}. \tag{9}$$

If the agent accepts only job 2,

$$v_0^{(0,1)} = \frac{\lambda_2\mu_2w_2}{\beta(\beta+\lambda_2+\mu_2)}, \tag{10}$$

$$v_1^{(0,1)} = \frac{\lambda_2\mu_2w_2}{\beta(\beta+\lambda_2+\mu_2)}, \tag{11}$$

$$v_2^{(0,1)} = \frac{\mu_2w_2(\beta+\lambda_2)}{\beta(\beta+\lambda_2+\mu_2)}. \tag{12}$$

Note that $\lambda_j > 0, \mu_j > 0, w_j \ge 0, 0 \le \beta \le 1$ where $j = 1, 2$. Policy $\mathcal{P} = (1,1)$ is optimal (accept both job-types) if $v_k^{(1,1)} \ge v_k^{(1,0)}, v_k^{(0,1)}$ for every $k = 0,1,2$. On simplification, we get the required conditions:

$$\left(w_2 = \frac{\mu_1w_1}{\mu_2}\right) \text{or} \left(w_2 < \frac{\mu_1w_1}{\mu_2} \text{and} \lambda_1 \le \frac{\mu_2w_2(\beta+\mu_1)}{\mu_1w_1-\mu_2w_2}\right) \text{or} \left(\lambda_2 \le \frac{\mu_1w_1(\beta+\mu_2)}{\mu_2w_2-\mu_1w_1} \text{and} w_2 > \frac{\mu_1w_1}{\mu_2}\right). \tag{13}$$

Similarly, policy $\mathcal{P} = (1,0)$ is optimal (accept only type-1 job) if $v_k^{(1,0)} \ge v_k^{(1,1)}, v_k^{(0,1)}$ for $k = 0,1,2$, which holds if

$$\mu_2 \le \frac{\mu_1w_1\lambda_1}{w_2(\beta+\lambda_1+\mu_1)} \tag{14}$$

and policy $\mathcal{P} = (0,1)$ is optimal if

$$\mu_1 < \frac{\lambda_2w_2}{w_1} \text{and} \mu_2 \ge \frac{\mu_1w_1(\beta+\lambda_2)}{\lambda_2w_2-\mu_1w_1}. \tag{15}$$

4. The Principal's Problem

Using Proposition 1, the principal learns how the agent makes her choices regarding accepting/rejecting a job and therefore, can set wages w_1, w_2 so that the agent accept the types of jobs the principal assigns her. Now the question is, how does the principal decide which jobs to assign? The principal has to take into account agent preference and the total number of agents (capacity). To find out, let the system state, defined as the number of jobs of each type being served, be denoted as (x_1, x_2). Let $V_n(x_1, x_2)$ be the expected discounted reward of the principal when there are n more periods to go. Denote p_i as the exogenous price paid by customer type- i to the principal. Define uniformization constant $\Gamma = \lambda_1 + \lambda_2 + c(\mu_1 + \mu_2) + \beta$. Without the loss of generality, assume $\Gamma = 1$. Let $\mathcal{S} = \{1, 2, \dots, c\}$ where $c \ge 1$ is the number of agents. Let T be Bellman operator such that $\forall x_1 + x_2 <= c$:

$$V_{n+1}(x_1,x_2) \equiv TV_n(x_1,x_2) = \begin{cases} \lambda_1 \max\{p_1 - w_1 + V_n(x_1+1,x_2), V_n(x_1,x_2)\} + \\ \lambda_2 \max\{p_2 - w_2 + V_n(x_1,x_2+1), V_n(x_1,x_2)\} + \\ x_1\mu_1 V_n(x_1-1,x_2) + x_2\mu_2 V_n(x_1,x_2-1) + \\ ((c-x_1)\mu_1 + (c-x_2)\mu_2)V_n(x_1,x_2) \end{cases} \tag{16}$$

Also, define $V_0(x_1,x_2) = 0$ for all $x_1, x_2 \in \mathcal{S}$ and $x_1 + x_2 \leq c$.

PROPOSITION 2 [Which job(s) should the principal admit?] For given x_i, x_j where $i,j \in \{1,2\}, i \neq j$, there exist a threshold $\overline{x}_i$ which depends on x_i, such that a fresh arrival of type-i job is accepted if and only if $x_j < \overline{x}_i$, where $\overline{x}_i = c - x_i$ if $V(x_i+1, c-x_i-1) > V(x_i, c-x_i-1)$ and $\overline{x}_i = min(x_j: 0 \leq x_j \leq c - x_i - 1, V(x_i+1,x_j) \leq V(x_i,x_j))$ otherwise.

Proof of Proposition 2. We need the following two lemmas to prove Proposition 2.

LEMMA 1: $V_n(\cdot)$ is a (weakly) decreasing function.

Proof of Lemma 1: We use mathematical induction to prove $V_n(x_1,x_2) \geq V_n(x_1+1,x_2)$ and $V_n(x_1,x_2) \geq V_n(x_1,x_2+1)$ for all $x_1, x_2 \in \mathcal{S}, x_1 + x_2 + 1 \leq c$ and $n \in \mathbb{Z}^+$. First, we will prove

$$V_n(x_1,x_2) \geq V_n(x_1+1,x_2). \tag{17}$$

For $n = 0$, $V_0(x_1,x_2) = V_0(x_1+1,x_2) = 0$. So (17) is true for $n = 0$.

Next, suppose $V_n(x_1,x_2) \geq V_n(x_1+1,x_2)$ is true for some $n > 0$ (induction hypothesis).

We have $V_{n+1}(x_1+1,x_2) = \lambda_1 \max\{p_1 - w_1 + V_n(x_1+2,x_2), V_n(x_1+1,x_2)\} + \lambda_2 \max\{p_2 - w_2 + V_n(x_1+1,x_2+1), V_n(x_1+1,x_2)\} + (x_1+1)\mu_1 V_n(x_1,x_2) + x_2\mu_2 V_n(x_1+1,x_2-1) + ((c-x_1-1)\mu_1 + (c-x_2)\mu_2)V_n(x_1+1,x_2)$.

We also have $V_{n+1}(x_1,x_2) = \lambda_1 \max\{p_1 - w_1 + V_n(x_1+1,x_2), V_n(x_1,x_2)\} + \lambda_2 \max\{p_2 - w_2 + V_n(x_1,x_2+1), V_n(x_1,x_2)\} + x_1\mu_1 V_n(x_1-1,x_2) + x_2\mu_2 V_n(x_1,x_2-1) + ((c-x_1)\mu_1 + (c-x_2)\mu_2)V_n(x_1,x_2)$.

It is easy to see that if our induction hypothesis is true, then $V_n(x_1+1,x_2) \geq V_n(x_1+2,x_2), V_n(x_1,x_2) \geq V_n(x_1+1,x_2), V_n(x_1,x_2-1) \geq V_n(x_1+1,x_2-1), V_{n+1}(x_1,x_2) \geq V_{n+1}(x_1+1,x_2)$.

Hence, $V_n(x_1,x_2) \geq V_n(x_1+1,x_2)$ is true for $n = 0$ and, if it is true for some $n > 0$, then it holds true for $n+1$ as well.

The case for type-2 jobs (i.e., $V_n(x_1,x_2) \geq V_n(x_1,x_2+1)$) can be shown analogously.

LEMMA 2: $V_n(\cdot)$ is a sub-modular function.

Proof of Lemma 2: We want to prove that

$$V_n(x_1,x_2) - V_n(x_1,x_2+1) \leq V_n(x_1+1,x_2) - V_n(x_1+1,x_2+1), \tag{18}$$

for $x_1 + x_2 + 2 \leq c, x_1, x_2 \in \mathcal{S}$ and $n \in \mathbb{Z}^+$.

Just like in the proof of Lemma 4, we use mathematical induction to prove (18).

For $n = 0$, clearly (18) holds true because $V_0(\cdot) = 0$. Next, suppose (18) holds true for some $n > 0$ (induction hypothesis).

Consider the following strategy: It is optimal to accept job type-1 if it arrives during state (x_1,x_2) and to reject if it arrives during state (x_1+1,x_2+1). We have,

$$\begin{aligned} &\max\{p_1 - w_1 + V_n(x_1+2,x_2), V_n(x_1+1,x_2)\} + \max\{p_1 - w_1 + V_n(x_1+1,x_2+1), V_n(x_1,x_2+1)\} \geq \\ &V_n(x_1+1,x_2) + p_1 - w_1 + V_n(x_1+1,x_2+1) = \\ &\max\{p_1 - w_1 + V_n(x_1+1,x_2), V_n(x_1,x_2)\} + \max\{p_1 - w_1 + V_n(x_1+1,x_2+1), V_n(x_1,x_2+1)\} \end{aligned} \tag{19}$$

The last two terms in (19) represent the one-stage optimal decision under consideration (that is, accept job type-1 if it arrives during state (x_1,x_2) and reject if it arrives during state (x_1+1,x_2+1)). Also, note that the last three terms in (16) represent non-arrival epochs where no decisions are made. Therefore, using the induction hypothesis and (19), if the optimal actions are as stated above, whenever (18) holds for $n > 0$, it holds for $n+1$ as well.[1]

Next, consider the following strategy: It is optimal to reject job type-1 if it arrives during state (x_1,x_2) and to accept if it arrives during state (x_1+1,x_2+1). Note that

$$\begin{aligned} &\max\{p_1 - w_1 + V_n(x_1+2,x_2), V_n(x_1+1,x_2)\} + \max\{p_1 - w_1 + V_n(x_1+1,x_2+1), V_n(x_1,x_2+1)\} \geq \\ &p_1 - w_1 + V_n(x_1+2,x_2) + V_n(x_1,x_2+1). \end{aligned} \tag{20}$$

Using (18) twice, we get

$$\begin{aligned} &p_1 - w_1 + V_n(x_1+2,x_2) + V_n(x_1,x_2+1) \geq V_n(x_1,x_2) + p_1 - w_1 + V_n(x_1+2,x_1+1) = \\ &\max\{p_1 - w_1 + V_n(x_1+1,x_2), V_n(x_1,x_2)\} + \max\{p_1 - w_1 + V_n(x_1+1,x_2+1), V_n(x_1,x_2+1)\}. \end{aligned} \tag{21}$$

[1] This step is inspired by [1] and [2] that study the submodular value functions in dynamic programming.

Therefore, using the induction hypothesis, equations (20 and (21), if the optimal actions are to reject job type-1 if it arrives during state (x_1, x_2) and to accept if it arrives during state $(x_1 + 1, x_2 + 1)$, whenever (18) holds for $n > 0$, it holds for $n + 1$ as well.

Lastly, it is easy to see that if it is optimal to take the same decision when job type-1 arrives during state (x_1, x_2) and state $(x_1 + 1, x_2 + 1)$, then whenever (18) holds for $n > 0$, it holds for $n + 1$ as well.

Note that (16) is a contraction mapping and the Bellman operator T preserves monotonicity and sub-modularity properties of $V_n(\cdot)$ when applied repeatedly [12]. Also, both state space $(x_1, x_2) \in S^2$ and action space (accept/reject/do nothing) are finite, which gives $\lim_{n\to\infty} V_n = V$. From Lemma 2, $V(x_1, x_2)$ is a submodular function. As a consequence, we have an optimal threshold policy [2, 12] (also known as trunk-reservation or switching-curve policy [13]). A direct application of Theorem 2 in Savin et al. [13] leads us to Proposition 2.

5. Conclusion

We consider a principal that receives two types of jobs according to a Poisson process. The principal assigns jobs to an agent (server) who may or may not accept them. The agent prefers one type of job over the other. The principal has two decisions to make: which job to admit and how much to pay the agent so that they accept the job assigned to them. We derived propositions 1 and 2 to help make the latter and the former decisions respectively. There are a number of ways to extend this paper. In the paper, we consider a single representative agent because we assume all agents have similar preferences. But what happens when the agents have heterogeneous preferences? What is the optimal number of agents c to hire? What happens when there are more than two types of jobs? What should be the optimal prices p_1 and p_2 paid by the customers to the principal? These are some of the possible research questions to explore in future works.

References

[1] Eitan Altman, Tania Jim´enez, and Ger Koole. On optimal call admission control in resource-sharing system. *IEEE Transactions on Communications*, 49 (9): 1659–1668, 2001.

[2] Eitan Altman and Ger Koole. On submodular value functions and complex dynamic programming. *Stochastic Models*, 14 (5): 1051–1072, 1998.

[3] Christian Borgs, Jennifer T Chayes, Sherwin Doroudi, Mor Harchol-Balter, and Kuang Xu. The optimal admission threshold in observable queues with state dependent pricing. *Probability in the Engineering and Informational Sciences*, 28 (1): 101–119, 2014.

[4] Noah Gans and Sergei Savin. Pricing and capacity rationing for rentals with uncertain durations. *Management Science*, 53 (3): 390–407, 2007.

[5] Mark E Lewis, Hayriye Ayhan, and Robert D Foley. Bias optimal admission control policies for a multiclass nonstationary queueing system. *Journal of Applied Probability*, 39 (1): 20–37, 2002.

[6] Steven A Lippman. Applying a new device in the optimization of exponential queuing systems. *Operations Research*, 23 (4): 687–710, 1975.

[7] Steven A Lippman and Sheldon M Ross. The streetwalkers' dilemma: A job shop model. *SIAM Journal on Applied Mathematics*, 20 (3): 336–342, 1971.

[8] Bruce L Miller. A queueing reward system with several customer classes. *Management Science*, 16 (3): 234–245, 1969.

[9] Srimathy Mohan, Ferdous M Alam, John W Fowler, Mohan Gopalakrishnan, and Antonios Printezis. Capacity planning and allocation for web-based applications. *Decision Sciences*, 45 (3): 535–567, 2014.

[10] E Lerzan Ormeci and Jan van der Wal. Admission policies for a two class loss system with general interarrival times. *Stochastic Models*, 22 (1): 37–53, 2006.

[11] EL Ormeci, A Burnetas, and J van der Wal. Admission policies for a two class loss system. *Stochastic Models*, 17 (4): 513–539, 2001.

[12] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[13] Sergei V Savin, Morris A Cohen, Noah Gans, and Ziv Katalan. Capacity management in rental businesses with two customer bases. *Operations Research*, 53 (4): 617–631, 2005.

[14] Richard F Serfozo. An equivalence between continuous and discrete time Markov decision processes. *Operations Research*, 27 (3): 616–620, 1979.

[15] Shaler Stidham. Optimal control of admission to a queueing system. *IEEE Transactions on Automatic Control*, 30 (8): 705–713, 1985.

[16] Julie Leanne Vile, Jonathan William Gillard, Paul R Harper, and Vincent A Knight. A queueing theoretic approach to set staffing levels in time-dependent dual-class service systems. *Decision Sciences*, 48 (4): 766–794, 2017.

[17] Utku Yildirim and John J Hasenbein. Admission control and pricing in a queue with batch arrivals. *Operations Research Letters*, 38 (5): 427–431, 2010.

[18] Seunghwan Yoon and Mark E Lewis. Optimal pricing and admission control in a queueing system with periodically varying parameters. *Queueing Systems*, 47 (3): 177–199, 2004.

[19] Gabriel Zayas-Cab´an and Mark E Lewis. Admission control in a two-class loss system with periodically varying parameters and abandonments. *Queueing Systems*, 94 (1-2): 175–210, 2020.