Optimal Control for Parallel Queues with a Single Batch Server

Hongyu Chen[a], Zizhuo Wang[b,*]

[a] *School of Mathematical Sciences, Peking University, Beijing, 100871, China*
[b] *School of Data Science, The Chinese University of Hong Kong, Shenzhen, 518172, China*

Abstract

We consider a queueing system with multiple Poisson arrival queues and a single batch server that has infinite capacity and a fixed service time. The problem is to allocate the server at each moment to minimize the long-run average waiting cost. We propose a Cost-Arrival Weighted (CAW) policy for this problem based on the structure of the optimal policy of a corresponding fluid model. We show that this simple policy enjoys a superior performance by numerical experiments.

Keywords: queueing system, batch service system, fluid model

1. Introduction

In this paper, we consider a model of multiple parallel queues and a single batch server that has the ability to serve all the customers in a certain queue within a fixed amount of time. The decision at each time is to choose which queue to serve and the goal is to minimize the long-run average waiting cost.

The model can serve as an adequate description for a wide range of applications. For example, consider a shuttle bus at an airport serving various nearby destinations (hotels, car rental locations, etc). It has a relatively large capacity and can serve all the passengers aiming at a given location at once within a certain amount of time. The decision it has to make is which destination to go to in each run in order to minimize the total waiting time of passengers. Another example would be certain online computing service applications, where in some cases, the actual computing time may be negligible compared to the fixed setup time (software initialization, warm up, etc). Under this scenario, the platform must decide which type of job it should process at each time with the objective of minimizing the total delays of jobs.

We first formulate the problem as a Markov Decision Process (MDP). However, the state space of the MDP is very large, making it computationally intractable. To propose a computationally efficient policy, we consider a simplified fluid model, which is a deterministic counterpart of the stochastic model. We show that the action sequence is cyclic for any optimal stationary policy, and present a characterization for the optimal policy. In particular, in the optimal policy, each time we calculate for each queue the product of the waiting cost coefficient, the current queue length, and a "future" waiting time, and we serve the queue with the maximum of the product. Based on the above characterization, we propose a heuristic policy by assuming

*Corresponding author
Email addresses: hongyuchen@pku.edu.cn (Hongyu Chen), wangzizhuo@cuhk.edu.cn (Zizhuo Wang)

that each queue is served at equally spanned intervals. By some calculation, we deduce that the optimal interval is proportional to the square root of the ratio between the waiting cost coefficient and the arrival rate of each queue. Then by using this interval as a substitute for the "future" waiting time, we propose a policy called the Cost-Arrival Weighted policy, or the CAW policy, which can give service decisions in both fluid and stochastic models. In particular, at each time period, the CAW policy calculates the following value for each queue i:

$$Q_i\sqrt{\frac{c_i}{\lambda_i}}$$

where Q_i is the current queue length, c_i is the waiting cost coefficient and λ_i is the arrival rate. The CAW policy chooses the queue with the largest value to serve. We then propose a generalization of the CAW policy to the case when the server's capacity is finite. Finally, we show by numerical experiments that the CAW policy and its generalization perform well under both the fluid and the stochastic settings.

The remainder of this paper is organized as follows. In Section 2, we review the related literature. In Section 3, we present our model, including both the stochastic and the fluid models, and present our main analytical results. We also propose the CAW policy and its generalization in Section 3. In Section 4, we conduct numerical experiments to validate the performance of the CAW policy. Section 5 concludes the paper.

2. Literature Review

Our work is related to the literature of batch service queueing systems where the server can serve multiple customers at the same time. We refer the interested readers to the monograph by Chaudhry and Templeton [1] for a comprehensive review of this topic. For more recent advances, see Armero and Conesa [2], Chang and Takine [3], Chen et al. [4] and the reference therein.

Another line of research focuses on the optimal control of multi-class queueing systems, where the classical $c\mu$-rule is proposed as the optimal policy under a variety of input assumptions, see Baras et al. [5], Buyukkoc et al. [6] and Shanthikumar and Yao [7]. Briefly speaking, when the system consists of a single server and multiple parallel queues, in order to minimize the expected total cost, the server should serve the non-idle queue with the maximal product of the queue-dependent waiting cost coefficient c and the service rate μ. While the $c\mu$-rule is of both theoretical and practical significance, it cannot be applied to our model because of the batch-service feature. In fact, we will demonstrate in Section 4 that a myopic policy based on the $c\mu$-rule performs relatively poorly in our model.

Liu and Wang [8] consider a similar model to ours with only two queues and analyze the optimal state-independent policy. They prove that the optimal state-independent policy would be serving the slow-arriving queue once followed by serving the fast-arriving queue multiple times (k). They also give an explicit formula for the optimal k. While we adapt this model to the case of multiple queues, our focus is on deriving an efficient state-dependent policy, which sets our two works apart.

There have also been vast literature on modeling complex systems with batch queues. For example, Ignall and Kolesar [9] and Weiss [10] consider an infinite capacity airport shuttle bus transporting passengers between two terminals. Deb [11] focuses on the case when the capacity is finite. Khazaei et al. [12] [13] and Santhi and Saravanan [14] use batch queues to model cloud computing centers and analyze their performance under different setups. While the background may be similar, our work differs from those in terms of the queue structure and the cost function.

3. Model and Analysis

We consider the scenario where there are N infinite capacity queues with a single batch server. In the base case, we assume that the batch server has infinite capacity, which means that it can serve all the customers in a single queue at once. We assume the service time of each queue is fixed and equals 1. The arriving process for queue i is assumed to be a Poisson process with parameter λ_i. Let $a(t) \in \{1, 2, \ldots, N\}$ denote the index of the queue served at time t. We let $Q_i(t)$ denote the length of the i-th queue at time t. Then $Q_i(t)$ inherits the following dynamics:

$$Q_i(t+1) = (1 - \mathbf{1}_{\{a(t)=i\}})Q_i(t) + Z_i, \quad t = 0, 1, 2, \ldots$$

where Z_i is a random variable following Poisson distribution with parameter λ_i and $\mathbf{1}$ is the indicator function. We use vector $\boldsymbol{q}(t) = (Q_1(t), Q_2(t), \ldots, Q_N(t))$ to denote the state of the system which records the length of each queue. We consider all non-preemptive stationary policies $\pi : \mathbb{Z}^N \to \{1, 2, \ldots, N\}$ that map a current state to the index of the queue which the server will serve. We define the cost-to-go function $J^\pi(\boldsymbol{q}, T)$ starting from state $\boldsymbol{q}$ following policy π for T time periods as

$$J^\pi(\boldsymbol{q}, T) := \mathbb{E}^\pi \left(\sum_{t=1}^{T} \sum_{i=1}^{N} c_i Q_i(t) \Big| \boldsymbol{q}(0) = \boldsymbol{q} \right),$$

where c_i is the waiting cost coefficient for a single customer in queue i in each period. We are interested in minimizing the long-run average cost, which is defined by:

$$\min_\pi J^\pi(\boldsymbol{q}) := \limsup_{T \to \infty} \frac{1}{T} J^\pi(\boldsymbol{q}, T). \tag{1}$$

Note that under our formulation, the states are communicative under every stationary policy. Hence the optimal value for (1) would be the same for every initial state $\boldsymbol{q}$ by Bertsekas et al. [15]. And the average cost optimality equation can be written as

$$\gamma^* + h^*(\boldsymbol{q}) = \boldsymbol{c}^T \boldsymbol{q} + \min_{a \in \{1,2,\ldots,N\}} \mathbb{E}\left(h^*(\boldsymbol{q} - q_a \boldsymbol{e}_a + \boldsymbol{Z})\right). \tag{2}$$

Here $\boldsymbol{Z}$ is a N-dimensional random vector with the i-th element following Poisson distribution with parameter λ_i, $\boldsymbol{e}_a$ is the unit vector with the a-th element equalling 1 and q_a is the a-th element of vector $\boldsymbol{q}$. By Bertsekas et al. [15], equation (2) can be interpreted as follows. If there exists solution (γ^*, h^*) for equation (2), then

γ^* is the optimal value of the average cost problem (1) and h^* is called the relative value function. Without loss of generality, we can set $h^*(\mathbf{0}) = 0$. Then $h^*(\boldsymbol{q})$ would be the minimum difference between the expected cost to reach $\mathbf{0}$ from $\boldsymbol{q}$ and the cost that would be incurred if the cost per stage is γ^*. Unfortunately, the state space of the above problem is very large, which makes solving (2) precisely computationally difficult. Hence, in the following, we focus on proposing an easy-to-compute heuristic policy.

The idea of our heuristic policy is based on a deterministic counterpart of the original model, in which the arrival of customers is deterministic in each time period. We also call the model the fluid model. Specifically, the fluid model has the following dynamics:

$$Q_i(t+1) = (1 - \mathbf{1}_{\{a(t)=i\}})Q_i(t) + \lambda_i. \quad t = 0, 1, 2, \dots$$

By Sennott [16], there exists an optimal stationary policy for the average cost problem (1) under the fluid model. In what follows, we wish to characterize the optimal stationary policy. To start with, we prove that the action sequence generated by the optimal stationary policy is cyclic. Because under the fluid model, the action sequence $a(t)$ is uniquely determined by the policy π and the initial state $\boldsymbol{q}_0$, we refer to $a(t)$ and π interchangeably in the following analysis.

Proposition 1. *For any initial state $\boldsymbol{q}(0)$, suppose π^* is an optimal stationary policy for the fluid model. Then the corresponding action sequence $a^*(t)$ would be cyclic after a certain time threshold, i.e., there exists $T_0 > 0$ and $t_0 > 0$ such that $a^*(t + t_0) = a^*(t)$ for every $t > T_0$.*

Proof. We first prove that under policy π^* there exists two time steps $T_0 < T_1$ such that the state $\boldsymbol{q}^*(T_0) = \boldsymbol{q}^*(T_1)$. We prove this by contradiction. Suppose for any $T_0 < T_1$, we have $\boldsymbol{q}^*(T_0) \neq \boldsymbol{q}^*(T_1)$. Note that the length of each queue i at time t has the following structure: $Q_i^*(t) = Q_i^*(0) + k\lambda_i$ or $Q_i^*(t) = k\lambda_i$ for some $k \in \mathbb{N}$. Therefore, for any $M > 0$, there are at most $\Pi_{i=1}^N (2\lceil \frac{M}{\lambda_i} \rceil + 1)$ states $\boldsymbol{q}^*(t)$ satisfying $\max_i\{Q_i^*(t)\} < M$. This implies that there exists a time threshold $T^* > 0$ such that $\max_i\{Q_i^*(t)\} \geq M$ for any $t > T^*$. As a result, the waiting cost induced at time $t > T^*$ would be $\sum_{i=1}^N c_i Q_i^*(t) \geq M c_{\min}$, where $c_{\min} = \min_i\{c_i\}$. And the long-run average cost of π^* is bounded below by

$$J^{\pi^*}(\boldsymbol{q}_0) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} c_i Q_i^*(t) \geq \limsup_{T\to\infty} \frac{T - T^*}{T} M c_{\min} = M c_{\min}. \tag{3}$$

Note that inequality (3) holds for arbitrary $M > 0$, which indicates $J^{\pi^*}(\boldsymbol{q}_0) = \infty$. We next show that we are able to find a policy π' (which may be non-stationary) that achieves finite average cost. In fact, by taking action $a'(i + kN) = i + 1$ for $0 \leq i \leq N - 1$ and $k \in \mathbb{N}$, the length of the i-th queue is bounded by $Q_i'(t) \leq Q_i'(0) + N\lambda_i$ for every $t > 0$. As a result, the average cost of π' has an upper bound:

$$J^{\pi'}(\boldsymbol{q}_0) \leq \sum_{i=1}^{N} c_i (Q_i'(0) + N\lambda_i).$$

This means $J^{\pi^*}(\boldsymbol{q}_0) > J^{\pi'}(\boldsymbol{q}_0)$, which is a contradiction since π^* is an optimal policy. As a result, there exists $T_0 < T_1$ such that $\boldsymbol{q}^*(T_0) = \boldsymbol{q}^*(T_1)$. Recall that the policy π^* is stationary, which means it will take

the same action facing the same state, so the action sequence $a^*(t)$ would be cyclic after T_0 and the cyclic period t_0 can be chosen as $T_1 - T_0$. □

Proposition 1 shows that we only need to consider a cyclic policy for the long-run average problem. We next give a necessary condition for the cyclic policy to be optimal. For any action sequence $a(t)$, we define the following quantity as the next time the server serves certain queue i:

$$w_i^a(t) = \min\left(\{s : a(s) = i, s > t\} \cup \{\infty\}\right).$$

Then for a cyclic optimal policy a^*, it must satisfy the following property.

Proposition 2. *Suppose an optimal action sequence a^* is cyclic after time T, then for any $t > T$, we have*

$$a^*(t) \in \arg\max_i \left(c_i Q_i(t)(w_i^{a^*}(t) - t)\right). \tag{4}$$

Proof. We prove by contradiction. Suppose this is not true for certain $t_1 > T$. We denote $k = a^*(t_1 + lt_0)$ as the action taken at time $t_1 + lt_0$ for all $l \in \mathbb{N}$ where t_0 is the cyclic period of a^*. We choose queue j such that $j \in \arg\max_i \left(c_i Q_i(t_1)(w_i^{a^*}(t_1) - t_1)\right)$. Then we form a new action sequence a' defined as

$$a'(t) = \begin{cases} j, & t = t_1 + 2lt_0,\ l \in \mathbb{N} \\ a^*(t) & \text{otherwise.} \end{cases}$$

This policy a' is a cyclic policy starting from T with period $2t_0$. We now focus on the single cycle starting from time t_1 to time $t_1 + 2t_0$. Because we only change the action at time t_1 during $[t_1, t_1 + 2t_0)$ and $[t_1 + t_0, t_1 + 2t_0)$ is still a full cycle as in the optimal policy a^*, both queue k and j must be served at least once between $t_1 + t_0$ and $t_1 + 2t_0$. As the result, we have $w_j^{a^*}(t_1 + 2lt_0) = w_j^{a'}(t_1 + 2lt_0), l = 1, 2, \ldots$ and $w_k^{a^*}(t_1 + 2lt_0) = w_k^{a'}(t_1 + 2lt_0), l = 1, 2, \ldots$. This means that by cancelling the service at time t_1 for queue k, the $Q_k(t_1)$ customers of queue k at time t_1 would have to wait $w_k^{a^*}(t_1) - t_1$ more time periods, inducing more cost of this single cycle by $c_k Q_k(t_1)(w_k^{a^*}(t_1) - t_1)$. In the same way, by adding a service to queue j at time t_1, we could reduce the cost of this single cycle by $c_j Q_j(t_1)(w_j^{a^*}(t_1) - t_1)$. Thus, we can calculate the difference between the average cost of a^* and a' as:

$$J^{a^*}(\boldsymbol{q}_0) - J^{a'}(\boldsymbol{q}_0) = \frac{c_j Q_j(t_1)(w_j^{a^*}(t_1) - t_1) - c_k Q_k(t_1)(w_k^{a^*}(t_1) - t_1)}{2t_0} > 0.$$

Therefore, we can conclude that the cost of a' is less than the cost of a^*, which is a contradiction since a^* is an optimal policy. □

Proposition 2 provides a rule for choosing the optimal action at each time step t. The rule takes the current queue length $Q_i(t)$ as well as the "future" waiting time $w_i^{a^*}(t) - t$ into consideration. The intuition behind this is that if we are going to serve some queue in the near future, because we can serve the customers all at once, then we would incline to serve other queues at the current time.

However, Proposition 2 does not directly give us a practical policy because it contains the future information $w_i^{a^*}(t)$. To overcome such difficulty, we propose an approximation method based on the above idea by assuming that each queue i is served at an equally spanned time interval h_i. Given a certain time step t, we use this h_i as a proxy for the "future" waiting time of queue i at time t, which is $w_i^{a^*}(t) - t$. Under such assumption, a heuristic policy at time t can be written as (we break ties arbitrarily):

$$a^H(t) = \arg\max_i \left(c_i Q_i(t) h_i\right).$$

Now we study how to choose h_i. First, if queue i has a homogeneous serving interval h_i, then its average waiting cost can be calculated as

$$R_i(h_i) = \frac{c_i(1+2+\cdots+h_i)\lambda_i}{h_i} = \frac{(1+h_i)c_i\lambda_i}{2}.$$

Moreover, since we only have a single server, the frequency of serving each queue needs to be added up to 1, i.e., $\sum_{i=1}^N 1/h_i = 1$. As a result, we obtain the following optimization problem (5) for solving the optimal serving interval h_i. Here we ignore whether it is feasible to find a sequence with the specified serving frequency for each queue.

$$\begin{aligned} \min_{h_1,\ldots,h_N} \ & \sum_{i=1}^N R_i(h_i), \\ s.t. \ & \sum_{i=1}^N \frac{1}{h_i} = 1, \\ & h_i \geq 0, \quad i = 1,2,\ldots,N. \end{aligned} \tag{5}$$

To solve (5), we note that it is equivalent to solving $\min\{\sum_{i=1}^N h_i c_i \lambda_i | \sum_{i=1}^N \frac{1}{h_i} = 1; h_i \geq 0, i = 1,2,\ldots,N\}$. By Cauchy-Schwartz inequality, we have

$$\sum_{i=1}^N h_i c_i \lambda_i = \left(\sum_{i=1}^N \frac{1}{h_i}\right)\left(\sum_{i=1}^N h_i c_i \lambda_i\right) \geq \left(\sum_{i=1}^N \sqrt{c_i \lambda_i}\right)^2,$$

and the equality is achieved when $h_i \propto \frac{1}{\sqrt{c_i\lambda_i}}$. Combining this with the constraint $\sum_{i=1}^N \frac{1}{h_i} = 1$, we have the solution to (5) to be $h_i^* = \frac{\sum_{k=1}^N \sqrt{c_k\lambda_k}}{\sqrt{c_i\lambda_i}}$.

Based on the above analysis, by taking $h_i = h_i^*$, the action at each time step would be $a^H(t) \in \arg\max_i\{c_i Q_i(t) \frac{\sum_{k=1}^N \sqrt{c_k\lambda_k}}{\sqrt{c_i\lambda_i}}\} = \arg\max_i\{Q_i(t)\sqrt{\frac{c_i}{\lambda_i}}\}$. That is, we calculate a weighted length of each queue weighted by $\sqrt{c_i/\lambda_i}$ and choose the queue with the maximal weighted length. We call this policy the Cost-Arrival Weighted (CAW) policy and we formally define it in Algorithm 1.

In the CAW policy, instead of following the myopic policy to serve the queue with the maximal product $c_i Q_i(t)$, we weigh each queue with a coefficient $\sqrt{c_i/\lambda_i}$, which takes the arriving speed of each queue into consideration and attenuates the effect of the waiting cost coefficient c_i. Specifically, this coefficient is smaller for larger λ_i, which aligns with our observations that we should sometimes wait for the queues with faster-arriving speed and serve the slow-arriving queues. This is because by delaying the service for the fast-arriving

Algorithm 1: Cost-Arrival Weighted (CAW) Policy

input : The arrival rate λ_i and the waiting cost coefficient c_i.

for $t = 1, 2, \ldots$ **do**

  Obtain the current length of each queue $Q_i(t)$.

  Take action $a^{CAW}(t) = \arg\max_i\{Q_i(t)\sqrt{\frac{c_i}{\lambda_i}}\}$. Break ties arbitrarily.

end for

queues, we could potentially serve more customers and lower down the overall cost. For the same reason, the cost coefficient c_i also enters the form as $\sqrt{c_i}$ instead of c_i as in the classical $c\mu$-rule.

An advantage of the CAW policy is that it can also be used in the stochastic setting in which the arrival is random. That is, Algorithm 1 can be used as a policy for the original problem (1). Note that the CAW policy is very easy to implement in practice. In the next section, we show that it also enjoys superior performance.

Before we close this section, we generalize the CAW policy to the case when the server only has a finite capacity K, i.e., each time it can only process up to K customers. Such a situation is also quite common in practice. The generalization is based on a similar idea as discussed in this section. In particular, we also propose a heuristic policy based on the fluid model that each queue i is served at an equally spanned time interval h_i. Here, because of the capacity constraint, we need to further ensure that $\lambda_i h_i \leq K$. Otherwise, queue i will accumulate infinite customers in the long run. To formalize this idea, instead of solving (5), we now solve the following optimization problem (6).

$$\begin{aligned} \min_{h_1,\ldots,h_N} \ & \sum_{i=1}^{N} R_i(h_i), \\ s.t. \ & \sum_{i=1}^{N} \frac{1}{h_i} = 1, \\ & 0 \leq \lambda_i h_i \leq K, i = 1, 2, \ldots, N. \end{aligned} \tag{6}$$

The following proposition shows that one can solve (6) through a single-direction search.

Proposition 3. *Suppose (6) has a feasible solution. Then there exists an optimal solution h_i^o to (6) satisfying*

$$h_i^o = \begin{cases} \frac{\theta K}{\sqrt{c_i \lambda_i}} & \sqrt{\frac{c_i}{\lambda_i}} \geq \theta, \\ \frac{K}{\lambda_i} & \sqrt{\frac{c_i}{\lambda_i}} < \theta, \end{cases} \tag{7}$$

where θ is the unique solution to

$$\theta = \frac{\sum_{\{i:\sqrt{c_i/\lambda_i} \geq \theta\}} \sqrt{c_i \lambda_i}}{K - \sum_{\{i:\sqrt{c_i/\lambda_i} < \theta\}} \lambda_i}.$$

With Proposition 3, we propose a capacity-constrained CAW policy (C-CAW). The main part of the policy follows the same idea as in the CAW policy, that is, to serve the queue with the highest value of $c_i Q_i(t) h_i^o$ among all queues. However, in the capacitated case, when the length of a queue exceeds K, the

cost will be higher than that is captured in the objective function of (6). We note that in the fluid model, the queue length will not exceed K due to the last constraint in (6). However, in the stochastic setting, the queue length may exceed K at some time period. To address this issue, we further prioritize the queue which will reach length K at the next service time. Particularly, at each time, we calculate $Q_i(t) + (h_i^o - 1)\lambda_i$ for each queue i. Here, $Q_i(t) + (h_i^o - 1)\lambda_i$ represents the expected queue length at the next service. If some of $Q_i(t) + (h_i^o - 1)\lambda_i$ exceed K, then we serve the queue with the maximum value of $c_i(Q_i(t) + (h_i^o - 1)\lambda_i)$. Otherwise, we choose the queue with the largest $c_i Q_i(t) h_i^o$ to serve. The detailed algorithm is given in Algorithm 2.

Algorithm 2: Capacity-Constrained Cost-Arrival Weighted (C-CAW) Policy

input : The arrival rate λ_i, the waiting cost coefficient c_i, and the capacity K.

Let $h_i^o, i = 1, 2, \ldots, N$ be the optimal solution to (7).

for $t = 1, 2, \ldots$ **do**

  Obtain the current length of each queue $Q_i(t)$.

  if $\max_i\{Q_i(t) + (h_i^o - 1)\lambda_i\} \geq K$ **then**

    Take action $a^{C-CAW}(t) = \arg\max_i\{c_i(Q_i(t) + (h_i^o - 1)\lambda_i)\}$. Break ties arbitrarily.

  else

    Take action $a^{C-CAW}(t) = \arg\max_i\{c_i Q_i(t) h_i^o\}$. Break ties arbitrarily.

  end if

end for

4. Numerical Experiments

In this section, we validate the performance of the CAW policy. In particular, we compare the performance of the CAW policy with other policies under both the fluid model and the stochastic model. We show that the CAW policy performs well in the test cases.

We first consider the deterministic case where we know the customer arrival of each time step. For this case, we can actually calculate the optimal policy by solving a mixed-integer linear programming (MILP) problem. Specifically, suppose the number of customers arriving at queue i at time t is $R_{i,t}$. We use $a_{i,t} \in \{0, 1\}$ to represent the action at time t for queue i with $a_{i,t} = 1$ meaning to serve the i-th queue at time t. We further use $b_{i,t}$ to capture the waiting time for the $R_{i,t}$ customers arriving at queue i at time t.

Then we can write the problem of minimizing the cost of T time periods starting from $\boldsymbol{q}_0 = \mathbf{0}$ as follows:

$$\begin{aligned} \min_{a,b} \quad & \frac{1}{T}\sum_{i=1}^{N}\sum_{t=0}^{T-1} c_i b_{i,t} R_{i,t}, \\ s.t. \quad & b_{i,t} \geq 1 + t' - t - \sum_{k=t+1}^{t'} (t'-k+1)a_{i,k}, \quad 0 \leq t \leq t' \leq T-1, 1 \leq i \leq N, \\ & \sum_{i=1}^{N} a_{i,t} = 1, \quad 0 \leq t \leq T-1, \\ & a_{i,t} \in \{0,1\}. \end{aligned} \tag{8}$$

Here we take by convention that $\sum_{i=a}^{b} x_i = 0$ for any $a > b$. In (8), the objective is calculated as the summation of the total waiting time for each customer arrival, which is the product of the waiting time periods $b_{i,t}$ and the number of customers $R_{i,t}$. The first constraint is to characterize the time elapsed from t to the next serving time. In particular, when $a_{i,k} = 0$ for $k = t+1, \ldots, t_0 - 1$ and $a_{i,t_0} = 1$, we have for any $t \leq t' \leq t_0 - 1$, the right hand side of the first constraint is $f(t') := 1 + t' - t - \sum_{k=t+1}^{t'} (t'-k+1)a_{i,k} = 1 + t' - t \leq t_0 - t$. Moreover, we have $f(t_0) = t_0 - t$ and for any $t' > t_0$, the expression $f(t') \leq 1 + t' - t - (t' - t_0 + 1)a_{i,t_0} = t_0 - t'$. Hence, the first constraint would reduce to $b_{i,t} \geq t_0 - t$, which is exactly the number of waiting time periods for customer $R_{i,t}$. The last two constraints in (8) are because of the definition of $a_{i,t}$. Since we only have a single server, we must require $a_{i,t}$ to be added up to 1 at each time step.

We first compare the performance of the CAW policy with the optimal solution in the fluid model. We consider the case with $N = 3$ queues, each has cost coefficient $c_1 = c_2 = c_3 = 1$, and the arrival rates are $\lambda_1 = 1, \lambda_2 = w, \lambda_3 = w \cdot v$, where $w, v = 2, 4, 8$ respectively. The initial state is $\boldsymbol{q}_0 = (0, 0, 0)$. By taking $R_{i,t} = \lambda_i$ and $T = 100$, we solve the MILP problem (8) for the first 100 time periods and consider this result to be the long-run average cost. In fact, the minimal average cost solved from the optimization problem (8) converges quickly as the time horizon T increases in these test problems. (We show three cases in Figure 1 to illustrate this. Other cases all have a similar phenomenon.) The numerical results about the comparison between the CAW policy and the minimal average cost are presented in Table 1. We can see that our CAW policy can achieve almost optimal results in the deterministic case with the performance gap smaller than 2% under most cases. Specifically, the performance is better when the difference of arrival rates between each queue is larger, in which case waiting for a fast-arriving queue would be more beneficial.

	$v = 2$			$v = 4$			$v = 8$		
	CAW	Optimal	Gap	CAW	Optimal	Gap	CAW	Optimal	Gap
$w = 2$	13.86	13.44	3.13%	19.34	19.34	0.00%	31.29	31.19	0.32%
$w = 4$	24.40	24.11	1.20%	36.18	35.84	0.95%	58.55	57.90	1.12%
$w = 8$	44.79	44.06	1.66%	68.26	67.30	1.43%	111.90	110.94	0.87%

Table 1: Long-run average cost of the CAW policy and the optimal policy in the fluid model when the arrival rates are $\lambda_1 = 1, \lambda_2 = w, \lambda_3 = w \cdot v$ and the cost coefficient $c_1 = c_2 = c_3 = 1$.

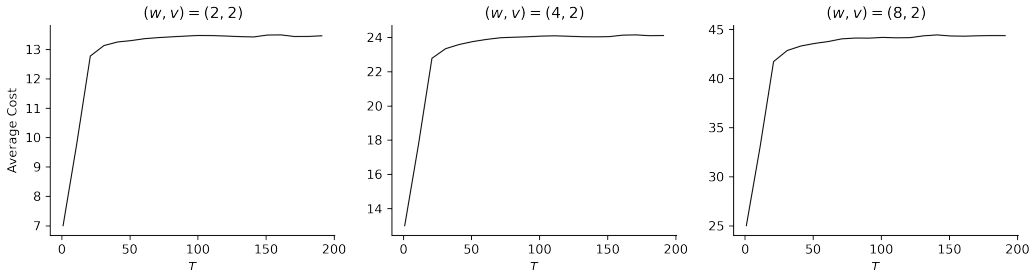

Figure 1: Optimal average cost of the first T time periods in the fluid model when $(w, v) = (2, 2), (4, 2)$ and $(8, 2)$.

Next we study the stochastic setting. In this setting, we consider the following four policies.

- Myopic Policy: $a^M(t) \in \arg\max_i\{c_i Q_i(t)\}$, which minimizes the cost at each time step.
- Fixed Scheduling Policy: the cyclic optimal policy of the corresponding fluid model.
- CAW Policy: $a^{CAW}(t) \in \arg\max_i\{Q_i(t)\sqrt{\frac{c_i}{\lambda_i}}\}$.
- Hindsight Policy: the ex post optimal policy, which is obtained by solving the corresponding MILP problem (8) after observing all the customer arrivals.

We first use the same problem setting as in the fluid model. The optimal cyclic policy is obtained by observing the optimal action sequence of the first 100 steps for the deterministic counterpart of each case. In fact, the optimal action sequence of the first 100 steps already has a strong cyclic characteristic. For example, we can observe that the optimal cyclic policies are $(1, 3, 2, 3)$ and $(1, 3, 2, 3, 2, 3)$ for the case when $(w, v) = (2, 2)$ and $(2, 4)$ respectively. For all the experiments, we set the time horizon $T = 100$ and repeat 50 times to calculate the average cost and performance gap. The performance of the four policies is presented in Table 2. Because the hindsight policy serves as a lower bound for our problem, we calculate the gap of the first three policies with respect to the hindsight policy, and denote the gaps as Gap-M, Gap-F, Gap-C respectively. As we can see in Table 2, the CAW policy performs the best among the three policies in each scenario with a maximal gap of 5.38%, while the gap may achieve 25.85% and 11.43% for the myopic and the fixed scheduling policies respectively.

w	v	Myopic	Fix	CAW	Hindsight	Gap-M	Gap-F	Gap-C
2	2	12.55	13.17	12.45	11.81	6.24%	11.43%	5.38%
2	4	19.63	19.13	18.53	17.79	10.32%	7.54%	4.14%
2	8	33.89	31.16	29.85	28.83	17.54%	8.07%	3.54%
4	2	23.85	23.82	22.85	22.04	8.22%	8.10%	3.69%
4	4	38.07	36.07	34.68	33.64	13.18%	7.23%	3.08%
4	8	67.24	57.93	56.83	55.22	21.78%	4.91%	2.91%
8	2	45.50	43.91	42.97	41.72	9.07%	5.26%	2.99%
8	4	74.75	67.53	66.46	64.71	15.53%	4.37%	2.72%
8	8	135.96	112.15	110.41	108.03	25.85%	3.81%	2.20%

Table 2: Comparison of the expected long-run average cost of the myopic policy, the fixed scheduling policy, the CAW policy and the hindsight policy in the stochastic setting when the arrival rates are $\lambda_1 = 1, \lambda_2 = w, \lambda_3 = w \cdot v$ and the cost coefficients are $c_1 = c_2 = c_3 = 1$. The time horizon T is set to be 100. We run 50 instances for each case and report the average results.

Now we compare the CAW policy to the state-independent optimal policy proposed by Liu and Wang [8]. Specifically, Liu and Wang [8] studied the optimal state-independent policy in the same setting when there are only two queues. They proved that the optimal policy holds the structure of serving the slow-arriving queue once followed by serving the fast-arriving queue multiple times. For the experiment, we assume the arrival rates to be $(\lambda_1, \lambda_2) = (1, r), r = 2, 4, 8, 16$ for the two queues. We repeat 50 times for each case and report the average cost of the state-independent policy (S-I), the CAW policy, and the hindsight policy in Table 4. Again, we calculate the gap between the first two policies and the hindsight policy, namely Gap-S and Gap-C. We can still observe that the CAW policy significantly outperforms the optimal state-independent policy, as it makes use of the information in current states.

r	S-I	CAW	Hindsight	Gap-S	Gap-C
2	4.58	3.95	3.84	19.32%	3.08%
4	7.36	6.68	6.48	13.46%	3.02%
8	12.36	11.63	11.32	9.16%	2.76%
16	22.24	21.33	20.82	6.83%	2.49%

Table 3: Comparison of the expected long-run average cost of the optimal state-independent policy, the CAW policy, and the hindsight policy in the stochastic setting when the arrival rates are $\lambda_1 = 1, \lambda_2 = r$ and the cost coefficients are $c_1 = c_2 = 1$. The time horizon T is set to be 100. We run 50 instances for each case and report the average results.

Next, we conduct experiments on problems with a larger scale. In particular, we consider the cases with $N = 10, 20, 30$ queues, each has cost coefficient $c_i = 1$, and the arrival rate λ_i follows truncated normal

distribution of mean 20 and standard deviation $\sigma = 5, 10, 15$, i.e., $\lambda_i = \max(0, z_i)$ where $z_i \sim \mathcal{N}(20, \sigma^2)$. We take the time horizon $T = 4N$ for each experiment. Under these scenarios, we compare the performance of the CAW policy and the myopic policy with respect to the hindsight policy. In Table 4, we report the average gap of 50 independent experiments. Specifically, the CAW policy performs better than the myopic policy in all the instances with a maximal gap of 3.12%, while the gap of the myopic policy may achieve 8.44% when $\sigma = 15$ and $N = 30$. Also, we notice that the CAW policy enjoys a larger advantage when the number of queues gets larger and the arrival rates have larger variation, in which cases the myopic policy often has poorer performance.

	$N = 10$		$N = 20$		$N = 30$	
	Gap-M	Gap-C	Gap-M	Gap-C	Gap-M	Gap-C
$\sigma = 5$	3.49%	2.37%	3.84%	2.43%	3.83%	2.14%
$\sigma = 10$	6.20%	3.12%	6.13%	2.65%	6.73%	2.60%
$\sigma = 15$	6.76%	2.93%	6.67%	2.74%	8.44%	2.61%

Table 4: Average efficiency gap of the CAW policy and the myopic policy compared to the hindsight policy. The number of queues is set to be $N = 10, 20, 30$ and the arrival rates are $\lambda_i = \max(0, z_i), z_i \sim \mathcal{N}(20, \sigma^2)$. The cost coefficients are $c_i = 1, i = 1, 2, \ldots, N$. The time horizon T is taken to be $4N$. We run 50 instances for each case and report the average results.

Finally, we conduct experiments for the finite capacity case. We use the same setting in the second experiment (Table 2) and we let the server's capacity $K = \alpha \sum_{i=1}^{3} \lambda_i$, where $\alpha = 1.3$ or 1.6. Specifically, we consider the myopic policy, the C-CAW policy, and the hindsight policy in this experiment. As the result in Table 4 shows, the C-CAW policy still outperforms the myopic policy in most cases. Specifically, the average gap of the C-CAW policy is 5.46%, which is notably smaller compared to the average gap of 12.07% of the myopic policy. We can also see that the C-CAW policy performs better when the arrival rates have larger variation and when the capacity is large, which is consistent with the findings from previous experiments.

		$\alpha = 1.3$					$\alpha = 1.6$				
w	v	Myopic	C-CAW	Hind.	Gap-M	Gap-C	Myopic	C-CAW	Hind.	Gap-M	Gap-C
2	2	13.46	13.62	12.62	6.65%	7.94%	12.85	12.76	11.99	7.16%	6.44%
2	4	21.09	21.03	19.65	7.36%	7.04%	20.33	19.47	18.54	9.65%	5.01%
2	8	35.89	33.83	32.08	11.86%	5.46%	34.12	31.33	29.77	14.60%	5.22%
4	2	25.75	26.20	24.15	6.64%	8.49%	24.37	23.61	22.53	8.17%	4.80%
4	4	40.05	38.71	36.35	10.17%	6.49%	38.70	35.92	34.49	12.18%	4.12%
4	8	69.84	62.92	60.32	15.78%	4.31%	67.98	59.54	56.80	19.69%	4.81%
8	2	47.93	47.03	43.86	9.27%	7.23%	46.11	43.40	41.92	10.01%	3.55%
8	4	78.32	72.86	69.58	12.57%	4.72%	76.04	68.26	65.79	15.59%	3.75%
8	8	138.69	122.73	118.09	17.45%	3.93%	135.26	115.96	110.42	22.49%	5.02%

Table 5: Comparison of the expected long-run average cost of the myopic policy, the CAW policy and the hindsight policy in the stochastic setting when the arrival rates are $\lambda_1 = 1, \lambda_2 = w, \lambda_3 = w \cdot v$ and the capacity $K = \alpha \sum_{i=1}^{3} \lambda_i$. The cost coefficients are $c_1 = c_2 = c_3 = 1$. The time horizon T is set to be 100. We run 50 instances for each case and report the average results.

5. Conclusion

In this paper, we consider the problem of allocating a single batch server to multiple queues in order to minimize the long-run average waiting cost. We first formulate the problem as a Markov Decision Process (MDP) problem. However, it is computationally intractable due to its large state space. To propose a computationally efficient policy, we analyze the structure of the optimal policy of a corresponding fluid model. We find out that there exists a cyclic optimal policy that chooses the queue with the maximal product of the queue length, the waiting cost coefficient, and a "future" service time. Based on this observation, by further assuming the service interval to be homogeneous, we propose a Cost-Arrival Weighted (CAW) policy, which weighs the i-th queue length by $\sqrt{c_i/\lambda_i}$ and chooses the longest weighted queue to serve. We also generalize the CAW policy to the case when the server only has a finite capacity. The efficiency of the CAW policy and its generalization is tested by extensive numerical experiments. The numerical results show that the CAW policy enjoys superior performance in both deterministic and stochastic cases. Specifically, the CAW policy has larger advantage when there are more queues and the arrival rates of each queue have larger variation.

References

[1] M. Chaudhry, J. Templeton, A First Course in Bulk Queues, A Wiley-Interscience Publication, Wiley, 1983. URL: `https://books.google.com.hk/books?id=zBtOAQAAIAAJ`.

[2] C. Armero, D. Conesa, Prediction in Markovian bulk arrival queues, Queueing Systems 34 (2000) 327–350.

[3] S. H. Chang, T. Takine, Factorization and stochastic decomposition properties in bulk queues with generalized vacations, Queueing Systems 50 (2005) 165–183.

[4] A. Chen, P. Pollett, J. Li, H. Zhang, Markovian bulk-arrival and bulk-service queues with state-dependent control, Queueing Systems 64 (2010) 267–304.

[5] J. Baras, D.-J. Ma, A. Makowski, K competing queues with geometric service requirements and linear costs: The μc-rule is always optimal, Systems & Control Letters 6 (1985) 173–180.

[6] C. Buyukkoc, P. Varaiya, J. Walrand, The cμ rule revisited, Advances in Applied Probability 17 (1985) 237–238.

[7] J. G. Shanthikumar, D. D. Yao, Multiclass queueing systems: Polymatroidal structure and optimal scheduling control, Operations Research 40 (1992) 293–299.

[8] Y. Liu, Z. Wang, A simple policy for multiple queues with size-independent service times, Operations Research Letters 41 (2013) 535–539.

[9] E. Ignall, P. Kolesar, Optimal dispatching of an infinite-capacity shuttle: Control at a single terminal, Operations Research 22 (1974) 1008–1024.

[10] H. J. Weiss, Further results on an infinite capacity shuttle with control at a single terminal, Operations Research 29 (1981) 1212–1217.

[11] R. K. Deb, Optimal dispatching of a finite capacity shuttle, Management Science 24 (1978) 1362–1372.

[12] H. Khazaei, J. Misic, V. B. Misic, Performance analysis of cloud computing centers using M/G/m/m+r queuing systems, IEEE Transactions on Parallel and Distributed Systems 23 (2011) 936–943.

[13] H. Khazaei, J. Misic, V. B. Misic, A fine-grained performance model of cloud computing centers, IEEE Transactions on Parallel and Distributed Systems 24 (2012) 2138–2147.

[14] K. Santhi, R. Saravanan, Performance analysis of cloud computing using batch queueing models in healthcare systems, Research Journal of Pharmacy and Technology 10 (2017) 3331–3336.

[15] D. P. Bertsekas, et al., Dynamic Programming and Optimal Control: Vol. 2, Athena Scientific, 2000.

[16] L. I. Sennott, Average cost optimal stationary policies in infinite state markov decision processes with unbounded costs, Operations Research 37 (1989) 626–633.

Appendix

Appendix A. Proof for Proposition 3

We can write the Lagrangian of the optimization problem (6) as

$$L(h_i, \mu_i, \tau) = \sum_{i=1}^{N} c_i h_i \lambda_i + \sum_{i=1}^{N} \mu_i (\lambda_i h_i - K) + \tau \left(\sum_{i=1}^{N} \frac{1}{h_i} - 1 \right).$$

Here, we first ignore the constraint $h_i \geq 0, i = 1, 2, \ldots, N$. As we will show in the following, the solution obtained will satisfy $h_i \geq 0$ automatically. Thus, our derivation is valid. Suppose problem (6) has a feasible solution, we next calculate its optimal solution by investigating its KKT conditions. For the first order condition, we have

$$\frac{\partial L}{\partial h_i} = c_i \lambda_i - \frac{\tau}{h_i^2} + \mu_i \lambda_i = 0, \quad i = 1, 2, \ldots, N. \tag{A.1}$$

For complementary slackness, we have

$$\mu_i (\lambda_i h_i - K) = 0, \quad i = 1, 2, \ldots, N.$$

Hence, for each queue i, either $\mu_i = 0$ or $\lambda_i h_i - K = 0$. For those queue i that satisfies $\mu_i = 0$, we get $h_i = \sqrt{\frac{\tau}{c_i \lambda_i}}$ by (A.1). It should also satisfy the constraint $\lambda_i h_i \leq K$, which renders us $\sqrt{\tau} \leq K\sqrt{c_i/\lambda_i}$. On the other hand, for those queue i that satisfies $\lambda_i h_i - K = 0$, i.e., $h_i = K/\lambda_i$, we have $\mu_i \lambda_i = \tau/h_i^2 - c_i \lambda_i$ by equation (A.1). Because the KKT condition also requires $\mu_i \geq 0$, we can get $\sqrt{\tau} \geq K\sqrt{c_i/\lambda_i}$. Thus, for a given τ, the optimal solution h_i has the following structure:

$$h_i = \begin{cases} \sqrt{\frac{\tau}{c_i \lambda_i}} & \sqrt{\tau} \leq K\sqrt{\frac{c_i}{\lambda_i}}, \\ \frac{K}{\lambda_i} & \sqrt{\tau} > K\sqrt{\frac{c_i}{\lambda_i}}. \end{cases} \tag{A.2}$$

Finally, by plugging the structure (A.2) into the constraint $\sum_{i=1}^{N} \frac{1}{h_i} = 1$, we get

$$\sqrt{\tau} = \frac{\sum_{\{i:\sqrt{\tau} \leq K\sqrt{c_i/\lambda_i}\}} \sqrt{c_i \lambda_i}}{K - \sum_{\{i:\sqrt{\tau} > K\sqrt{c_i/\lambda_i}\}} \lambda_i} K.$$

Take $\theta = \sqrt{\tau}/K$ and we would have the desired result in Proposition (3).