Optimality of first-come-first-served: a unified approach

N. Uuganbaatar

Department of Economics, Stockholm School of Economics, Box 6501, 113 83 Stockholm, Sweden, uugnaa.ninjbat@gmail.com.

Abstract

This paper provides a unified approach that can directly verify the following results related to First-Come-First-Served (FCFS): **(a)** in the case of a single server system, FCFS is optimal for *max* of C (completion time) and *max* of F (flow time), **(b)** in the case of a multi server system with identical servers, when customers have the equal processing time, any optimal discipline for the *total (sum)* of C, F and W (waiting time) has the same service starting times as FCFS, and **(c)** in the later case, FCFS is optimal for *max* of C, *max* of F and *max* of W.

Keywords: Optimality of FCFS, Optimal non-preemptive queue disciplines, Parallel machine scheduling with equal processing time.

1 Introduction

In queueing models FCFS is often assumed to be the queue discipline. Moreover, FCFS is very common in real life situations such as the grocery stores. This paper aims to (re)investigate certain optimality properties of this commonly used queue discipline. There are several results on the optimality of FCFS in queueing literature (see [6], [4], [3], [2], [12], [9], [8], [5]). Generality of these results necessarily depends on their setting: properties of the queueing system, class of disciplines among which the comparison is made and the optimality criterion that is considered. For instance, Gittins [6] shows that in $(GI/GI/m)$ queues, if the processing times are $i.i.d$ across customers, then the *expected waiting time* for a typical customer in steady state is minimum under FCFS among all non-preemptive queue disciplines.[1] More recent works provide rather general results: in $(G/GI/m)$ queues,

[1] In queueing theory, so called Kendall's notation, $A/B/m$, is often used to describe queueing systems. A describes the arrival time distribution, B describes the service times distribution and

if the processing times are *i.i.d* across customers, the expected value of any Schur convex function of *customer waiting times* and *total workload after arrival of each customer*, and of any symmetric and convex function of *customer flow times* are minimum under FCFS, among all non-preemptive queue disciplines (see [4], [2], [8], [11], [5]).

In this paper we treat the discipline design problem as a scheduling problem. Accordingly, we use performance measures used in scheduling theory to evaluate different queue disciplines: *total* and *max* of *completion time*, *flow time* and *waiting time* (C, F, W), and all of our results can be interpreted in the context of (n/m) parallel machine scheduling problem. Our main findings are as follows. First, in Theorem 3.1 in Sect. 3 we show that, in the case of a single server system, FCFS minimizes *max* of C and *max* of F. We then consider systems with identical customers, i.e. with equal processing times. However, all performance measures that we consider are convex and symmetric (hence, Schur convex) and the equal processing time case is a special case of that being *i.i.d.* But in Theorem 3.2 in Sect. 3 we show that, when all customers have the same processing time, not only FCFS is optimal for the the *sum* of C, F, W, but any optimal schedule has the same service starting times as FCFS. Then, in Theorem 3.4 in Sect. 3 we also show that, in that case FCFS is optimal for *max* of W. Finally, in Theorem 4.1 and 4.2 in Sect. 4, we extend results in Theorem 3.2 and 3.4 in Sect. 3 to the case of a multi server system with identical servers.

Most of our results are known in scheduling literature. For instance, results in Theorem 3.1 are well known in scheduling theory (see [7], [1]). Results in Theorem 3.2 **(a)** and 4.1 follow from "*optimality of greedy schedules*," whereas results in Theorem 3.4 and 4.2 follow from a more general result in [10]. However, our proof technique is based on a recursive reasoning and it is different than the other popular techniques: interchange argument, forward and backward induction, majorization argument and linear and dynamic programming. It is based on a simple observation that "*in order to show optimality of FCFS for max (bottleneck)-problems, it suffices to compare feasible schedules over the last busy period before system reaches its peak under FCFS*," and provides a direct, self-contained and unified approach, used thoroughly in proving Theorem 3.1, 3.4, 4.2 in Sect. 3 and 4.

In the next section we introduce our notation and main definitions. In Section 3, we consider single server systems and in Section 4 we consider multi server systems with identical servers and the last section concludes.

2 The set up

Let there be $n \in \mathbb{N}$ customers $J = \{1, 2, ..., n\}$ and a single server. Each customer $i \in J$ has a processing time $p_i \geq 0$ and an arrival time $t_i \geq 0$ and let $p = (p_1, ..., p_n) \in \mathbb{R}^n_+$ and $t = (t_1, ..., t_n) \in \mathbb{R}^n_+$ be the corresponding vectors. Without loss of generality we may assume that $0 \leq t_1 \leq \cdots \leq t_n$. A **schedule** $s = (s_1, \ldots, s_n) \in \mathbb{R}^n_+$ for a given $(p, t) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+$ assigns to each customer $i \in J$ a starting time $s_i \geq 0$ when the server begins to process it. It is **feasible** if processing of a customer does not start before his arrival, $s_i \geq t_i, i \in J$, and the server does not process more than one customer simultaneously, $\forall i, j \in J$ with $i \neq j$, $[s_i, s_i + p_i) \cap [s_j, s_j + p_j) = \emptyset$. **First-come-first-served-schedule** (FCFSS) is a schedule s^* that processes all jobs in the order of their arrival and does so as soon as possible: $s^*_1 = t_1$

the last entry, m describes the number of service channels. G stands for general and GI stands for general independent distributions.

and $s_i^* = \max\{s_{i-1}^* + p_{i-1}, t_i\}$ for $i = 2, \ldots, n$. Note that, according to the above definition we only consider permutation schedules without preemption, but we allow the server to stay idle when there are customers available for the service. From now on we only consider feasible schedules and we first verify that the FCFSS is feasible.

Proposition 2.1. *s^* is feasible for any $(p,t) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$.*

Proof. We can express the feasibility condition above as follows: $s_i \geq t_i, i \in J$, and $\forall i, j \in J$ such that $i \neq j$, if $s_i < s_j$, then $s_i + p_i \leq s_j$. Then, by definition s^* satisfies both of these conditions. □

A queue discipline is defined as a complete contingent plan of schedules. More formally, a **queue discipline** is a mapping $q : \mathbb{R}_+^n \times \mathbb{R}_+^n \to \mathbb{R}_+^n$ which assigns to each $(p,t) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$ a feasible schedule s. The **FCFS** is a queue discipline q^* such that $\forall (p,t) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$, $q^*(p,t) = s^*$. The following performance measures are commonly used in both queueing and scheduling theory. The **flow time**, the **completion time** and the **waiting time** for customer $i \in J$ under schedule s are the time that he spends in the queueing system, the time that is needed before he gets the service completed and the time that he waits in the queue until he gets the service, respectively. The corresponding formulas are: $F_i(s) = s_i + p_i - t_i$, $C_i(s) = s_i + p_i$ and $W_i(s) = s_i - t_i$. The **sum (total)** and the **maximum** of these measures are defined as usual: $F(s) = \sum_1^n F_i(s)$, $F_{\max}(s) = \max\{F_1(s), ..., F_n(s)\}$; $C(s) = \sum_1^n C_i(s)$, $C_{\max}(s) = \max\{C_1(s), ..., C_n(s)\}$; $W(s) = \sum_1^n W_i(s)$, $W_{\max}(s) = \max\{W_1(s), ..., W_n(s)\}$.

A schedule s is an **optimal schedule** for the performance measure M if there is no other schedule s' such that $M(s') < M(s)$ and a queue **discipline** q **is optimal** for M if $\forall (p,t) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$, $q(p,t)$ is optimal for M. Finally, the following permutation defined for any set of finitely many real numbers is very useful in our proofs. Let $\alpha = (\alpha_{k+1}, ..., \alpha_{k+m}) \in \mathbb{R}_+^m$ be a set of m nonnegative real numbers and let $\alpha^\pi = (\alpha_{\pi_1}, ..., \alpha_{\pi_m})$ a permutation of α. Then we call α^π as **ranking** of α if $\alpha_{\pi_i} \leq \alpha_{\pi_{i+1}}$ for $i = 1, ..., m-1$.

3 Single server systems

Theorem 3.1. *Consider a single server system and let $(p,t) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$ be arbitrary. Then q^* is optimal for $C_{\max}$ and $F_{\max}$.*

Proof. For the first claim, we need to prove that, given $(p,t) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$, for any schedule s, $C_{\max}(s) \geq C_{\max}(s^*)$. By definition, $s_i^* = \max\{s_{i-1}^* + p_{i-1}, t_i\}$ for $i = 2, ..., n$, which implies that $s_i^* + p_i \geq s_{i-1}^* + p_{i-1}$, for $i = 2, ..., n$ and hence, $C_{\max}(s^*) = s_n^* + p_n$.

Let us define $j = \max\{i : 1 \leq i \leq n, s_i^* = t_i\}$. So, j is the the last customer in J who gets the service at his arrival under s^*. Note that j is well defined since $s_1^* = t_1$. For an arbitrary schedule s, consider $\alpha = (s_j, ..., s_n)$ and its ranking $\alpha^\pi = (s_{\pi_1}, ..., s_{\pi_{n-j+1}})$. By feasibility, $s_i \geq t_i \geq t_j$ for $i = j, ..., n$, and $s_{\pi_{n-j+1}} + p_{\pi_{n-j+1}} \geq s_{\pi_{n-j}} + p_{\pi_{n-j}} + p_{\pi_{n-j+1}} \geq ... \geq t_j + \sum_{i=j}^n p_i$. By definition, $C_{\max}(s) \geq s_{\pi_{n-j+1}} + p_{\pi_{n-j+1}}$. But since $s_i^* = s_{i-1}^* + p_{i-1}$ for $i = j+1, ..., n$ and $s_j^* = t_j$, $C_{\max}(s^*) = t_j + \sum_{i=j}^n p_i$. Hence, $C_{\max}(s) \geq C_{\max}(s^*)$. This proves the first claim.

For the second claim, we need to prove that, given $(p,t) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+$, for any schedule s, $F_{\max}(s) \geq F_{\max}(s^*)$. By definition, $F_1(s^*) = p_1$ and for $i = 2, ..., n$, $F_i(s^*) = s^*_i + p_i - t_i = \max\{s^*_{i-1} + p_{i-1}, t_i\} - t_i + p_i = \max\{s^*_{i-1} + p_{i-1} - t_i, 0\} + p_i$. Let $k \in J$ be such that $F_{\max}(s^*) = F_k(s^*)$. Let us define $j = \max\{i : 1 \leq i \leq k, s^*_i = t_i\}$. So, j is the the last customer in $\{1, ..., k\}$ who gets the service at his arrival under s^*. Note that j is well defined since $s^*_1 = t_1$. Then by definition, $F_k(s^*) = s^*_k + p_k - t_k = t_j + \sum_{i=j}^{k} p_i - t_k$.

For an arbitrary schedule $s = (s_1, ..., s_n)$, consider $\alpha = (s_j, ..., s_k)$ and its ranking $\alpha^\pi = (s_{\pi_1}, ..., s_{\pi_{k-j+1}})$. Note that $t_j \leq t_{\pi_1} \leq s_{\pi_1}$ and $t_{\pi_{k-j+1}} \leq t_k$ and by feasibility we conclude that $F_{\pi_{k-j+1}}(s) = s_{\pi_{k-j+1}} + p_{\pi_{k-j+1}} - t_{\pi_{k-j+1}} \geq s_{\pi_{k-j}} + p_{\pi_{k-j}} + p_{\pi_{k-j+1}} - t_{\pi_{k-j+1}} \geq ... \geq s_{\pi_1} + \sum_{i=j}^{k} p_i - t_{\pi_{k-j+1}} \geq t_j + \sum_{i=j}^{k} p_i - t_k = F_k(s^*)$. Since $F_{\max}(s) \geq F_{\pi_{k-j+1}}(s)$, this completes the proof. $\square$

Results in Theorem 3.1 show that s^* is always optimal for $C_{\max}$ and $F_{\max}$. The following example shows that for the other performance measures, such a general result does not hold.

Let $n = 2$ and $t = (0, 1)$ and $p = (10, 1)$. Consider schedule $s = (2, 1)$. Then $F(s) = 13 < F(s^*) = 20$; $C(s) = 14 < C(s^*) = 21$; $W(s) = W_{\max}(s) = 2 < 9 = W(s^*) = W_{\max}(s^*)$.

However, if all customers have the equal processing time, s^* is optimal for all of these performance measures.

Theorem 3.2. *Let all customers have the equal processing time, $p_i = p_0 \in \mathbb{R}_+$, $i \in J$. Then,*

(a) schedule s is optimal for $F(s)$,$W(s)$and $C(s)$ if and only if $s_{\pi_i} = s^*_i$ for all $i \in J$, where $s^\pi = (s_{\pi_1}, ..., s_{\pi_n})$ is the ranking of s, and

*(b) s^* is optimal for the performance measures in **(a)**. Moreover, s^* is the unique optimal schedule if and only if $t \in \mathbb{R}^n_+$ is such that $t_i + p_0 < t_{i+2}$, for $i = 1, ..., n-2$.*

Proof. **(a)** Note that the objectives differ by a constant, hence the optimal schedules coincide. Each objective take its minimum value whenever $\sum s_i$ is at its minimum. Let s be an arbitrary schedule.

For the if part, it suffices to show that $\sum s^*_i \leq \sum s_i$. Consider the ranking $s^\pi = (s_{\pi_1}, ..., s_{\pi_n})$ of s. Since s^π is a permutation of s, $\sum s_{\pi_i} = \sum s_i$. Note that by feasibility, $s_{\pi_j} \geq t_j$ for all $j = 1, ..., n$ since there must be at least j customers have arrived in order π_j to be the $j'th$ customer to be served. In particular, $s_{\pi_1} \geq t_1 = s^*_1$. For $2 \leq i \leq n$, if $s_{\pi_{i-1}} \geq s^*_{i-1}$, then it is also true that $s_{\pi_i} \geq s^*_i$ since $s_{\pi_i} \geq s_{\pi_{i-1}} + p_0 \geq s^*_{i-1} + p_0$ and $s_{\pi_i} \geq t_i$. But since $s_{\pi_1} \geq s^*_1$, we conclude that $s_{\pi_i} \geq s^*_i$, $i \in J$. Hence, $\sum s_{\pi_i} \geq \sum s^*_i$.

For the only if part, suppose s is optimal. Then it must be the case that $\sum s_{\pi_i} = \sum s^*_i$ and since it is also true that $s_{\pi_i} \geq s^*_i$, $i \in J$, the equality of the two sums is possible only if each term in the sum is equal. This completes the proof.

(b) The optimality of s^* is trivial from **(a)**. Note that, if s is an optimal schedule, then $s_{\pi_1} = s_1^* = t_1$, which implies that $\pi_1 = 1$ since it is possible to assign t_1 only to the customer 1. For the if part, suppose $t_1 + p_0 < t_3$. Since $t_2 < t_3$, we conclude that $t_3 > s_2^*$. But for an optimal schedule s, the only customer that can be scheduled at $s_{\pi_2} = s_2^*$ is the customer 2 since no other customer is available at s_2^*. Hence, for any optimal schedule s, $\pi_2 = 2$. Similarly, we conclude that $\pi_i = i$ for $i = 3, ..., n$. Hence, if s is optimal, then $s = s^*$.

For the only if part, let s^* be the only optimal schedule and let there be $j \in \{1, ..., n-2\}$ such that $t_j + p_0 > t_{j+2}$. Consider schedule s such that it agrees with s^* in all positions but $(j+1)'th$ and $(j+2)'th$: $s_{\pi_i} = s_i^*$ for $i \in J$ and $\pi_i = i$ for $i \in J \backslash \{j+1, j+2\}$ and $\pi_{j+1} = j+2$ and $\pi_{j+2} = j+1$. Then by construction s is optimal and $s \neq s^*$, which contradicts to the uniqueness of s^*. □

The theorem above can be interpreted as: *when all customers have the same processing time, the optimal schedule is characterized by the starting times of* s^*. Moreover, from this exact characterization, for any $t \in \mathbb{R}_+^n$, one can fully describe the set of corresponding optimal schedules:

Corollary 3.3. *Let all customers have the equal processing time,* $p_i = p_0 \in \mathbb{R}_+$, $i \in J$, *and let* $t \in \mathbb{R}_+^n$ *be the vector of arrival times. Let us define* $L_1, ..., L_n$ *and* $i_1, ..., i_n$ *as follows:* $L_k = \{i \in \mathbb{N} : 1 \leq i \leq n, s_k^* \geq t_i\} = \{1, 2, ..., i_k\}$. *Then,*

(a) $L_1 = 1, L_n = \{1, ..., n\}$ *and* $k \leq i_k \leq i_{k+1} \leq n$, *for* $k = 1, ..., n$, *and*

(b) *There are* $\varphi = \prod_{k=1}^{n} (i_k - k + 1)$ *many distinct optimal schedules and every such schedule can be generated by the following procedure:*

Step 1: *Assign for the first position of the schedule* $\pi_1 = 1$ *and update* L_k *into* L_k^1 *by deleting all the first entries of* L_k, *for* $k = 2, ..., n$,

Step j for $2 \leq j \leq n$: *Assign for the* $j'th$ *position of the schedule any* $\pi_j \in L_j^{j-1}$ *and update* L_k^{j-1} *into* L_k^j *by deleting all the first entries of* L_k^{j-1} *and replacing all* π_j *in* L_k^{j-1} *by the deleted first entry, for* $k = j+1, ..., n$.

Proof. **(a)** By definition $L_1 = 1, L_n = \{1, ..., n\}$ and $k \leq i_k \leq n$, for $k = 1, ..., n$. Note that if $i_k \in L_k$, then $i_k \in L_{k+1}$ for $k = 1, ..., n$ since $s_{k+1}^* > s_k^* \geq t_{i_k}$. Hence, $i_k \leq i_{k+1}$ for $k = 1, ..., n$.

(b) Note that every optimal schedule uniquely corresponds to an assignment

$$\{\pi_1, ..., \pi_n : \pi_k \in L_k, \pi_k \neq \pi_j \text{ for } k \neq j\}$$

and by construction the procedure above gives all such assignments for any given $t \in \mathbb{R}_+^n$. Consider *Step j*: any customer in L_j^{j-1} can be assigned to the $j'th$ position and those are the only possible choices for that position. Note that, by construction there are $(i_j - j + 1)$ elements in L_j^{j-1}. Hence, there are φ distinct optimal schedules. □

Let us demonstrate procedure in Corollary 3.3 with an example:

Let $n = 6$ and $t \in \mathbb{R}^6_+$ and $p_0 \in \mathbb{R}_+$ be such that $L_1 = \{1\}, L_2 = \{1,2\}, L_i = \{1,2,3,4,5\}$, for $3 \le i \le 5$, and $L_6 = \{1,2,...,6\}$. Let us construct a matrix M with L_i in its $i'th$ row: $M = \begin{bmatrix} 1 & & & & & \\ 1 & 2 & & & & \\ 1 & 2 & 3 & 4 & 5 & \\ 1 & 2 & 3 & 4 & 5 & \\ 1 & 2 & 3 & 4 & 5 & \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$.

In Step 1 we assign $\pi_1 = 1$ and obtain an updated matrix: $M_1 = \begin{bmatrix} \mathbf{1} & & & & & \\ . & 2 & & & & \\ . & 2 & 3 & 4 & 5 & \\ . & 2 & 3 & 4 & 5 & \\ . & 2 & 3 & 4 & 5 & \\ . & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$.

In Step 2 we assign $\pi_2 = 2$ since that is the only feasible assignment and update M_1: $M_2 = \begin{bmatrix} \mathbf{1} & & & & & \\ . & \mathbf{2} & & & & \\ . & . & 3 & 4 & 5 & \\ . & . & 3 & 4 & 5 & \\ . & . & 3 & 4 & 5 & \\ . & . & 3 & 4 & 5 & 6 \end{bmatrix}$.

In Step 3 we may assign any of $\{3,4,5\}$ for π_3 and let $\pi_3 = 4$, we update M_2: $M_3 = \begin{bmatrix} \mathbf{1} & & & & & \\ . & \mathbf{2} & & & & \\ . & . & 3 & \mathbf{4} & 5 & \\ . & . & . & 3 & 5 & \\ . & . & . & 3 & 5 & \\ . & . & . & 3 & 5 & 6 \end{bmatrix}$.

In Step 4 we can assign any of $\{3,5\}$ for π_4 and let $\pi_4 = 5$, we update M_3: $M_4 = \begin{bmatrix} \mathbf{1} & & & & & \\ . & \mathbf{2} & & & & \\ . & . & 3 & \mathbf{4} & 5 & \\ . & . & . & . & \mathbf{5} & \\ . & . & . & . & 3 & \\ . & . & . & . & 3 & 6 \end{bmatrix}$.

Then, for π_5 the only feasible assignment is 3 and for π_6 its 6: $M_6 = \begin{bmatrix} \mathbf{1} & & & & & \\ . & \mathbf{2} & & & & \\ . & . & 3 & \mathbf{4} & 5 & \\ . & . & . & . & \mathbf{5} & \\ . & . & . & . & \mathbf{3} & \\ . & . & . & . & . & \mathbf{6} \end{bmatrix}$.

Theorem 3.4. *When all customers have the equal processing time, $p_i = p_0 \in \mathbb{R}_+, i \in J$, s^* is optimal for $W_{\max}$.*

Proof. By definition, $W_1(s^*) = 0$ and for $i = 2, ..., n$,

$$W_i(s^*) = s_i^* - t_i = \max\{s_{i-1}^* + p_0, t_i\} - t_i = \max\{s_{i-1}^* + p_0 - t_i, 0\}.$$

Let $k \in J$ be such that $W_{\max}(s^*) = W_k(s^*)$. Let us define $j = \max\{i : 1 \leq i \leq k, s_i^* = t_i\}$. So, j is the the last customer in $\{1, ..., k\}$ who gets the service at his arrival under s^*. Note that j is well defined since $s_1^* = t_1$. Then by definition, $W_k(s^*) = s_k^* - t_k = t_j + (k-j) \cdot p_0 - t_k$.

For an arbitrary schedule s, consider $\alpha = (s_j, ..., s_k)$ and its ranking $\alpha^\pi = (s_{\pi_1}, ..., s_{\pi_{k-j+1}})$. Note that $t_j \leq t_{\pi_1} \leq s_{\pi_1}$ and $t_{\pi_{k-j+1}} \leq t_k$ and by feasibility we conclude that $W_{\pi_{k-j+1}}(s) = s_{\pi_{k-j+1}} - t_{\pi_{k-j+1}} \geq s_{\pi_1} + (k-j) \cdot p_0 - t_{\pi_{k-j+1}} \geq t_j + (k-j) \cdot p_0 - t_k = W_k(s^*)$. Since $W_{\max}(s) \geq W_{\pi_{k-j+1}}(s)$, this completes the proof. $\square$

4 Multi server systems with equal processing time

In this section we extend the main results in Sect. 3 to a multi server system. Before we state the main results, we shall introduce some more notations and extend some of the main definitions to the new setting. Let there be $k \in \mathbb{N}$ identical servers, $M = \{m_0, m_1, ..., m_{k-1}\}$. For $i \in J$, let $z(i), y(i) \in \mathbb{Z}_+$ be such that $i = z(i) \cdot k + y(i)$ with $y(i) < k$, i.e. $i \equiv y \mod k$. Each customer has a processing time $p \geq 0$ and let $t = (t_1, ..., t_n) \in \mathbb{R}_+^n$ be the vector of customer arrival times. As before we may assume that $0 \leq t_1 \leq \cdots \leq t_n$. We redefine the notions of **schedule**, **feasibility** and **FCFSS**, and the other notions are defined same as in Sect. 2. A **schedule** $[s] = [(s_1, 1_M), \ldots, (s_n, n_M)]$ for a given $t \in \mathbb{R}_+^n$ assigns to each customer $i \in J$ a pair of (s_i, i_M) where $s_i \geq 0$ is the starting time for i gets the service and $i_M \in M$ is the corresponding server. It is **feasible** if processing of a customer does not start before his arrival: $s_i \geq t_i, \forall i \in J$ and none of the servers processes more than one job simultaneously: $\forall m \in M$, $\forall i, j \in J$ with $i \neq j$, if $i_M = j_M = m$, then $[s_i, s_i + p) \cap [s_j, s_j + p) = \emptyset$. **First-come-first-served-schedule** (FCFSS) is a schedule $[s^*] = [(s_1^*, 1_M^*), \ldots, (s_n^*, n_M^*)]$ that processes all jobs in the order of their arrival and does so as soon as possible: for $1 \leq i < k$, $(s_i^*, i_M^*) = (t_i, m_i)$; for $i = k$, $(s_k^*, k_M^*) = (t_k, m_0)$; and for $k < i \leq n$, $(s_i^*, i_M^*) = (\max\{s_{(z(i)-1) \cdot k + y(i)}^* + p, t_i\}, m_{y(i)})$.

Note that FCFSS can be defined up to an arbitrary assignment of the initial k arrivals to the servers. Here we have chosen a particular one, $i'th$ arrival is assigned to $i'th$ server. Since our enumeration of the servers was arbitrary, none of the results that follow depend on this particular choice. Before we state the main results of this section we prove the following lemma.

Lemma 3. *Let $[s]$ be any schedule and $s = (s_1, ..., s_n)$ be the vector of the starting times of $[s]$. Consider $s^\pi = (s_{\pi_1}, ..., s_{\pi_n})$, the ranking of s. For any $(k+1)$ sequence $(s_{\pi_l}, s_{\pi_{l+1}}, ..., s_{\pi_{l+k}})$, $\exists j \in \{0, ..., k-1\}$ such that $s_{\pi_{l+k}} \geq s_{\pi_{l+j}} + p$.*

Proof. Since there are k servers, there are at least two customers π_q, π_i with $q < i$ among $(\pi_l, ..., \pi_{l+k})$ who assigned to the same server, by the pigeonhole principle. Then the result follows by feasibility: $s_{\pi_{l+k}} \geq s_{\pi_i} \geq s_{\pi_q} + p$. $\square$

The following results are extensions of Theorem 3.2 and 3.4, subsequently.

Theorem 4.1. *Schedule $[s]$ is optimal for $F(s)$, $W(s)$ and $C(s)$ if and only if $s_{\pi_i} = s_i^*$ for all $i \in J$ where $s^\pi = (s_{\pi_1}, ..., s_{\pi_n})$ is the ranking of $s = (s_1, ..., s_n)$, the vector of starting times of $[s]$.*

Proof. Note that all of the measures in the theorem take their minimum value whenever $\sum s_i$ is at its minimum. Let $[s]$ be an arbitrary schedule and s be the vector of starting time of $[s]$.

For the if part, it suffices to show that $\sum s_i^* \leq \sum s_i$. Consider the ranking $s^\pi = (s_{\pi_1}, ..., s_{\pi_n})$ of s. Since s^π is a permutation of s, $\sum s_{\pi_i} = \sum s_i$. Note that by feasibility, $s_{\pi_j} \geq t_j$ for all $j \in J$ since there must be at least j customers have arrived in order π_j be the $j'th$ customer to be served. In particular, for $1 \leq i \leq k, s_{\pi_i} \geq s_i^* = t_i$. For $k < i \leq n$, if $s_{\pi_{(z(i)-1)\cdot k+y(i)}} \geq s^*_{(z(i)-1)\cdot k+y(i)}$, then it is also true that $s_{\pi_i} \geq s_i^*$ since $s_{\pi_i} \geq t_i$ and by Lemma 3, $s_{\pi_i} \geq s_{\pi_{(z(i)-1)\cdot k+y(i)}} + p \geq s^*_{(z(i)-1)\cdot k+y(i)} + p$. But since $s_{\pi_i} \geq s_i^* = t_i$ for $1 \leq i \leq k$, we conclude that $s_{\pi_j} \geq s_j^*$ for all $j \in J$. Hence, $\sum s_{\pi_i} \geq \sum s_i^*$.

For the only if part, suppose $[s]$ is optimal. Then it must be the case that $\sum s_{\pi_i} = \sum s_i^*$ and since it is also true that $s_{\pi_i} \geq s_i^*$ for all $i \in J$, the equality of the two sums is possible only if each term in the sum is equal. This completes our proof. $\square$

Theorem 4.2. *$[s^*]$ is optimal for $W_{\max}, C_{\max}$ and $F_{\max}$.*

Proof. We prove the result for $W_{\max}$ and essentially the same procedure works for $C_{\max}$ and $F_{\max}$. By definition, for $1 \leq i \leq k, W_i(s^*) = 0$ and for $k < i \leq n$, $W_i(s^*) = s_i^* - t_i = \max\{s^*_{(z(i)-1)\cdot k+y(i)} + p, t_i\} - t_i = \max\{s^*_{(z(i)-1)\cdot k+y(i)} + p - t_i, 0\}$. Let $r \in \{1, ..., n\}$ be such that $W_{\max}(s^*) = W_r(s^*)$.

Let us define $j = \max\{i : 1 \leq i \leq r, i = z(i) \cdot k + y(r), s_i^* = t_i\}$. So, j is the the last customer in $\{y(r), k + y(r), 2 \cdot k + y(r), ..., r\}$ (here we identify $0'th$ customer with $k'th$) who gets the service at his arrival under $[s^*]$. Note that j is well defined since $s^*_{y(r)} = t_{y(r)}$.

Then by definition, $W_r(s^*) = s_r^* - t_r = t_j + (z(r) - z(j)) \cdot p - t_r$. For an arbitrary schedule $[s]$ with the vector of starting times s, consider $\alpha = (s_j, s_{j+1}, ..., s_r)$ and its ranking $\alpha^\pi = (s_{\pi_1}, ..., s_{\pi_{r-j+1}})$. Note that $t_j \leq t_{\pi_1} \leq s_{\pi_1}$ and $t_{\pi_{r-j+1}} \leq t_r$ and by Lemma 3 we conclude that $W_{\pi_{r-j+1}}(s) = s_{\pi_{r-j+1}} - t_{\pi_{r-j+1}} \geq s_{\pi_1} + (z(r) - z(j)) \cdot p - t_{\pi_{r-j+1}} \geq t_j + (z(r) - z(j)) \cdot p - t_r = W_r(s^*)$. Since $W_{\max}(s) \geq W_{\pi_{r-j+1}}(s)$, this completes our proof. $\square$

5 Conclusion

Since FCFS is commonly used both in theory and in applications, its optimality properties receive considerable attention. This paper provides a technique that can be used to investigate optimality properties related to FCFS in a single and (identical) multi server settings.

The underlying idea of our technique is to *compare schedules (queue disciplines) over the last busy period before the system reaches to its peak under FCFS*. Then, *max* -objectives can be expressed as recursive sums over that period and optimality proofs pin down to simple comparisons of sums of real numbers. Hence, our approach provides simple, unified and self-contained proofs for optimality results related to FCFS.

References

[1] K.R.Baker, D.Trietsch, Principles of Sequencing and Scheduling. Wiley, New York, 2009.

[2] D.J.Daley, Certain optimality properties of the first-come first-served discipline for $G/G/s$ queues, Stochastic Processes and their Applications 25 (1987) 301-308

[3] B.T.Doshi, E.H.Lipper, Comparisons of service disciplines in a queueing system with delay dependent customer behavior, in: R.Disney, T.Ott (Eds) Applied Probability–Computer Science: the Interface, vol. II. Birkhauser, Cambridge, MA, 1982, pp. 269-301.

[4] S.G.Foss, Comparison of servicing strategies in multi-channel queueing systems, Siberian Mathematical Journal 22 (1981) 190-197

[5] S.G.Foss, N.I.Chernova, On the optimality of the FCFS discipline in multiserver queueing systems and networks, Siberian Mathematical Journal 42 (2001) 372-385

[6] J.C.Gittins, A comparison of service disciplines for GI/G/m queues, Math. Operationsforsch. Statist., Ser. Optimization 9 (1978) 255-260

[7] E.L.Lawler, Optimal sequencing of a single machine subject to precedence constraints, Management Science 19 (1973) 544-546

[8] Z.Liu, D.Towsley, Effects of service disciplines in $G/GI/s$ queueing systems, Annals of Oper. Res. 48 (1994) 401-429

[9] R.Righter, G.J.Shanthikumar, Extremal properties of the FIFO discipline in queueing networks, J. of Applied Probability 29 (1992) 967-978

[10] B.Simons, Multi processor scheduling of unit-time jobs with arbitrary release times and deadlines, SIAM J. Comp. 12 (1983) 294-299

[11] D.Towsley, Application of Majorization to Control Problems in Queueing Systems, in P.Chretienne et al. (Eds) Scheduling Theory and its Applications, Wiley, New York, 1995

[12] R.W.Wolf, Upper bounds on works in system for multichannel queues, J. of Applied Probability 24 (1987) 547-551