Online Supplement for "Maximizing throughput in zero-buffer tandem lines with dedicated and flexible servers" by Mohammad H. Yarmand and Douglas G. Down

Appendix A. Lemma 1 - the remaining cases

In this appendix, we provide the proof of the remaining cases of Lemma 1.

First scenario:

The details of Cases 1 and 2 of Lemma 1 for its first scenario is as follows.

Case 1: Assume $d^2_{p,t_0} > f^1_{k+1,t_0}$. Let $t_1 = f^1_{k+1,t_0}$.

Case 1.1: Assume $f^{\omega,2}_{k,t_0} - t_1 < d^2_{p,t_0+t_1} < f^{\omega',2}_{k,t_0+t_1}$.

Case 1.2: Assume $d^2_{p,t_0+t_1} < f^{\omega,2}_{k,t_0} - t_1$. Let $t_2 = d^2_{p,t_0+t_1}$.

Case 1.2.1: Assume $d^1_{k+2,t_0+t_1+t_2} < \min\{d^2_{k+1,t_0+t_1+t_2}, f^{\omega,2}_{k,t_0} - t_1 - t_2\}$. This case is discussed in Section **??**.

Case 1.2.2: Assume $d^1_{k+2,t_0+t_1+t_2} > \min\{d^2_{k+1,t_0+t_1+t_2}, f^{\omega,2}_{k,t_0} - t_1 - t_2\}$. We divide this case further.

Case 1.2.2.1: Further assume $f^{\omega,2}_{k,t_0} - t_1 - t_2 < d^2_{k+1,t_0+t_1+t_2}$. Let $t_3 = f^{\omega,2}_{k,t_0} - t_1 - t_2$. Consider the system at $t_0 + t_1 + t_2 + t_3$. Under ω we have $d^1_{k+2,t_0+t_1+t_2+t_3}$, $f^{\omega,1}_{k+3,t_0+t_1+t_2+t_3}$, and $d^2_{k+1,t_0+t_1+t_2+t_3}$ as residual service times for jobs $k+2$, $k+3$, and $k+1$, respectively. Under ω' we have $d^1_{k+2,t_0+t_1+t_2+t_3}$, $f^{\omega',2}_{k,t_0+t_1+t_2+t_3}$, and $d^2_{k+1,t_0+t_1+t_2+t_3}$ as residual service times for jobs $k+2$, $k+3$, and $k+1$, respectively. Figure A.1 illustrates this case. We divide this case further.

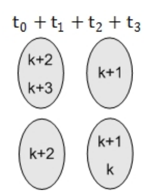

Figure A.1: Case 1.2.2.1

Case 1.2.2.1.1: Define $t_4 = f^{\omega',2}_{k,t_0+t_1+t_2+t_3}$ and assume that job k has the smallest residual service time. At $t_0 + t_1 + t_2 + t_3 + t_4$, σ^1_d is serving job $k+2$ and σ^2_d is serving job $k+1$ under both policies. Under ω the residual service time of job $k+3$ is $f^{\omega,2}_{k+3,t_0+t_1+t_2+t_3} - t_4$ and under ω' it is $f^{\omega',2}_{k+3,t_0+t_1+t_2+t_3+t_4}$. This is similar to Case 3.2, discussed below. Figure A.2 illustrates this case.

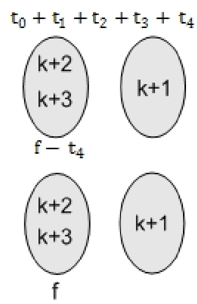

Figure A.2: Case 1.2.2.1.1

Case 1.2.2.1.2: Assume job $k+3$ has the smallest residual service time. Then $D^{\omega}(n+2) < D^{\omega'}(n+2)$.

Case 1.2.2.1.3: Assume job $k+1$ has the smallest residual service time. This is similar to Case 1.2.2.1.1.

Case 1.2.2.1.4: Assume job $k+2$ has the smallest residual service time. The policy ω performs a hand-off as job $k+2$ is served and therefore $D^{\omega}(n+2) < D^{\omega'}(n+2)$.

Case 1.2.2.1.5: Assume the residual service times of jobs $k+2$ and $k+3$ are equal to or less than the residual service time for job $k+1$. This is similar to Case 1.2.2.1.4.

Case 1.2.2.1.6: Assume the residual service times of jobs $k+1$ and $k+2$ are equal to or less than the residual service time for job $k+3$. This is similar to Case 1.2.2.1.1.

Case 1.2.2.2: Further assume $d^2_{k+1,t_0+t_1+t_2} < f^{\omega,2}_{k,t_0} - t_1 - t_2$. Let $t_3 = d^2_{k+1,t_0+t_1+t_2}$. Consider the system at time $t_0 + t_1 + t_2 + t_3$. Jobs $k+2$ and $k+3$ are being served at the first station under both policies. Under ω the residual service time of job k is $d^{\omega,2}_{k,t_0} - t_1 - t_2 - t_3$. Under ω' the residual service time of job k is $d^{\omega',2}_{k,t_0+t_1} - t_2 - t_3$.

Figure A.3 illustrates this case. We have $D^{\omega}(n+2) = D^{\omega'}(n+2)$.

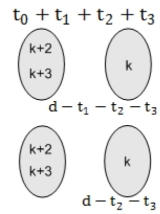

Figure A.3: Case 1.2.2.2

Case 1.2.2.3: Further assume $d^2_{k+1,t_0+t_1+t_2} = f^{\omega,2}_{k,t_0} - t_1 - t_2$. This is similar to Case 1.2.2.2.

Case 1.2.3: Assume $d^1_{k+2,t_0+t_1+t_2} = d^2_{k+1,t_0+t_1+t_2} = f^{\omega,2}_{k,t_0} - t_1 - t_2$. This is similar to Case 1.2.2.1.

Case 1.3: Assume $f^{\omega',2}_{k,t_0+t_1} \leq d^2_{p,t_0+t_1}$. Then $D^{\omega}(n+1) = t_0 + t_1 + f^{\omega,2}_{k,t_0} - t_1$ and $D^{\omega'}(n+1) = t_0 + t_1 + f^{\omega',2}_{k,t_0+t_1}$. Therefore $D^{\omega}(n+1) < D^{\omega'}(n+1)$.

Case 1.4: Assume $d^2_{p,t_0+t_1} = f^{\omega,2}_{k,t_0} - t_1$. This is similar to Case 1.2.2.1.

Case 2: Assume $d^2_{p,t_0} < f^1_{k+1,t_0}$, meaning that the dedicated server completes its service before the flexible server. Let $t_1 = d^2_{p,t_0}$, assume σ^1_d is holding the k^{th} job, and $n-1$ departures from the first station have occurred. Under ω', t_1 time units elapse before σ^2_d becomes available. So at t_0+t_1, the blocked job goes to the second station with residual service time $d^{\omega',2}_{k,t_0+t_1}$. The policy ω performs a hand-off at t_0 and sends the flexible server to the second station with a service time of $f^{\omega,2}_{k,t_0}$. At t_0+t_1, σ^2_d becomes available, and the flexible server hands off its job to σ^2_d with residual service time $d^{\omega,2}_{k,t_0} - t_1$. Therefore at t_0+t_1, $D^{\omega}(n) < D^{\omega'}(n)$. As the servers are indistinguishable, without loss of generality, we can assume that under ω' at time t_0+t_1 the two servers at the first station swap their jobs. Figure A.4 illustrates this case.

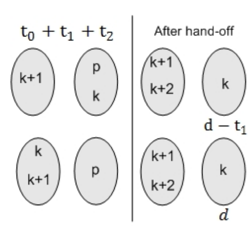

Figure A.4: Case 2

Consider the system at time t_0+t_1 and let $t_2 = \min\{d^1_{k+1,t_0+t_1}, f^1_{k+2,t_0+t_1}\}$.

Case 2.1: Assume $d^{\omega,2}_{k,t_0} - t_1 < t_2 < d^{\omega',2}_{k,t_0+t_1}$. At time $t_0 + t_1 + t_2$, ω sends σ^1_d's job (job $k+1$) to the second station and admits the $k+3^{rd}$ job. At time $t_0 + t_1 + t_2$, ω' performs a hand-off between σ^1_d and σ^1_f (jobs $k+1$ and $k+2$) and the flexible server is sent to the second station. Therefore $D^{\omega}(n+1) = D^{\omega'}(n+1) = t_0 + t_1 + t_2$. Let $t_3 = \min\{d^{\omega',2}_{k,t_0+t_1} - t_2, f^2_{k+1,t_0+t_1+t_2}\}$. Figure A.5 illustrates this case. We divide this case further.

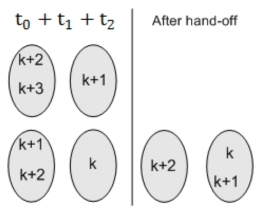

Figure A.5: Case 2.1

Case 2.1.1: Assume $\min\{f^{\omega,1}_{k+3,t_0+t_1+t_2}, d^1_{k+2,t_0+t_1+t_2}\} < t_3$. This means that under ω a service completion occurs in the first station earlier than a service completion in the second station under ω'. Therefore $D^{\omega}(n+2) = t_0 + t_1 + t_2 + t_3 < D^{\omega'}(n+2)$. Figure A.6 illustrates this case.

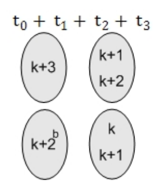

Figure A.6: Case 2.1.1

Case 2.1.2: Assume $\min\{f^{\omega,1}_{k+3,t_0+t_1+t_2}, d^1_{k+2,t_0+t_1+t_2}\} > t_3$. This case is similar to Case 1.2.1.1. Figure A.7 illustrates this case.

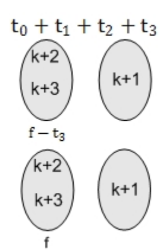

Figure A.7: Case 2.1.2

Case 2.2: Assume $t_2 < d^{\omega,2}_{k,t_0} - t_1$. This is similar to Case 1.2. Figure A.8 illustrates this case.

Case 2.3: Assume $d^{\omega',2}_{k,t_0+t_1} < t_2$. Then at $t_0 + t_1 + d^{\omega',2}_{k,t_0+t_1}$ the system is the same under both

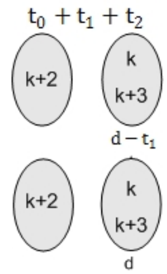

Figure A.8: Case 2.2

policies. We have $D^{\omega}(n+1) = D^{\omega'}(n+1) = t_0 + t_1 + t_2$. Figure A.9 illustrates this case.

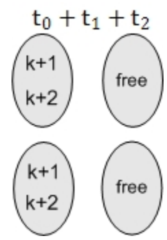

Figure A.9: Case 2.3

The *ahead* property is required when we make cross references between cases. For example, it is stated that Case 1.2.2.1.1 is similar to Case 3.2. In order to be able to adapt Case 3.2, we need to ensure all the assumptions made in Cases 3.2 and 3 hold. In other words, if Case 3.2 is applied to a system in which the *ahead* property does not hold, it is possible to have earlier departures under ω'. For example if Case 3.2 is applied to a system similar to Figure A.2, with the exception that job $k+3$ has smaller residual service time under ω', the *ahead* property would be violated and departures would happen earlier under ω'.

Second scenario:

For the second scenario, we enumerate sub-cases of Cases 3 and 4 of Lemma 1 without presenting the details.

The proof is by induction.

Basis step:

If time t_0 is not reached, both systems remain the same and there is nothing to prove in the basis step. Otherwise, start the system from the empty state and let the first departure from the first station happen ($D^{\omega}(1) = D^{\omega'}(1)$). If the flexible server serves the departed task in the second station, the system follows scenario 2. The server σ_d^1 is serving job 2, σ_f^2 is serving job 1, and σ_d^2 is free. Up until time t_0, the two policies are the same. When time t_0 is reached, the following cases are possible.

Case 3: Assume $d^1_{2,t_0} < f^2_{1,t_0}$, meaning that the dedicated server completes its service before the flexible server. Let $t_1 = d^1_{2,t_0}$. Under policy ω', σ^1_d completes its service after t_1 time units. So at $t_0 + t_1$, the completed job is sent to the second station with corresponding residual service time d^2_{2,t_0+t_1}. The policy ω performs a hand-off at time t_0 and sends the flexible server to the first station with residual service time $f^{\omega,1}_{3,t_0}$. At time $t_0 + t_1$, σ^1_d completes its service, and the flexible server hands off its job to σ^1_d with residual service time $d^{\omega,1}_{3,t_0} - t_1$. The flexible server goes to the second station with residual service time f^2_{2,t_0+t_1}. Hence $D^\omega(2) = D^{\omega'}(2) = t_0 + t_1$.

Case 4: Now assume $d^1_{2,t_0} > f^2_{1,t_0}$, meaning that the flexible server completes its service before the dedicated server. Let $t_1 = f^2_{1,t_0}$. Under ω', σ^2_f becomes available after t_1 time units. So at $t_0 + t_1$, the flexible server is sent to the first station with residual service time $f^{\omega',1}_{3,t_0+t_1}$ and σ^2_d is free. The policy ω performs a hand-off at t_0 and sends the flexible server to the first station with residual service time $f^{\omega,1}_{3,t_0}$. Now if $f^{\omega,1}_{3,t_0} - t_1 < d^1_{2,t_0+t_1}$, $D^\omega(2) < D^{\omega'}(2)$. Otherwise $D^\omega(2) = D^{\omega'}(2)$.

Inductive step:

If t_0 has occurred in the basis step, Lemma 2 shows that for each case considered in the inductive step, the system respects the *ahead* property. Hence, under ω', departures will not occur sooner than under ω. If time t_0 is not reached, the two systems remain the same and there will be nothing to prove at this stage.

If time t_0 occurs after the $n-1^{st}$ departure but before the n^{th} departure, consider the following configuration: a busy dedicated server at the first station (job k) and a free dedicated server and a busy flexible server (job p) at the second station. Up until time t_0, the two policies are the same. When time t_0 is reached, the following cases are possible.

Case 3: Assume $d^1_{k,t_0} < f^2_{p,t_0}$, meaning that the dedicated server completes its service before the flexible server. Let $t_1 = d^1_{k,t_0}$. Under policy ω', σ^1_d completes its service after t_1 time units. So at $t_0 + t_1$, the completed job is sent to the second station with residual service time d^2_{k,t_0+t_1}. There is also an admission at the first station with residual service time $d^{\omega',1}_{k+1,t_0+t_1}$. The policy ω performs a hand-off at t_0 and sends the flexible server to the first station with residual service time $f^{\omega,1}_{k+1,t_0}$. At $t_0 + t_1$, σ^1_d completes its service, and the flexible server hands off its job to σ^1_d with residual service time $d^{\omega,1}_{k+1,t_0} - t_1$. The flexible server goes to the second station with residual service time f^2_{k,t_0+t_1}.

Let $t_2 = \min\{d^2_{k,t_0+t_1}, f^2_{p,t_0+t_1}\}$.

Case 3.1: Assume $d^{\omega',1}_{k+1,t_0+t_1} < t_2$.

Case 3.2: Assume $d^{\omega,1}_{k+1,t_0} - t_1 > t_2$.

Case 3.2.1: Assume $\min\{d^{\omega,1}_{k+1,t_0} - t_1 - t_2, f^1_{k+2,t_0+t_1+t_2}\} < d^2_{k,t_0+t_1}$.

Case 3.2.1.1: Further assume $d^{\omega,1}_{k+1,t_0} - t_1 - t_2 < f^1_{k+2,t_0+t_1+t_2} < d^{\omega',1}_{k+1,t_0+t_1} - t_2$.

Case 3.2.1.2: Assume $f^1_{k+2,t_0+t_1+t_2} < d^{\omega,1}_{k+1,t_0} - t_1 - t_2$.

Case 3.2.1.3: Assume $d^{\omega',1}_{k+1,t_0+t_1} - t_2 < f^1_{k+2,t_0+t_1+t_2}$.

Case 3.2.2: Assume $\min\{d^{\omega,1}_{k+1,t_0} - t_1 - t_2, f^1_{k+2,t_0+t_1+t_2}\} > d^2_{k,t_0+t_1}$.

Case 3.3: Assume $d^{\omega,1}_{k+1,t_0} - t_1 < t_2 < d^{\omega',1}_{k+1,t_0+t_1}$.

Case 4: Now assume $d^1_{k,t_0} > f^2_{p,t_0}$, meaning that the flexible server completes its service before the dedicated server. Let $t_1 = f^2_{p,t_0}$. Under policy ω', server σ^2_f becomes available after t_1 time units. So at $t_0 + t_1$, the flexible server is sent to the first station with residual service time $f^{\omega',1}_{k+1,t_0+t_1}$ and σ^2_d is free. The policy ω performs a hand-off at t_0 and sends the flexible server to the first station with residual service time $f^{\omega,1}_{k+1,t_0}$.

Let $t_2 = d^1_{k,t_0+t_1}$.

Case 4.1: Assume $f^{\omega,1}_{k+1,t_0} - t_1 < t_2$.

Case 4.2: Assume $f^{\omega,1}_{k+1,t_0} - t_1 > t_2$.

(Lemma 1) $\square$

Appendix B. Proof of Lemma 2

In this appendix, the proof of Lemma 2 is explained. The following lemma shows that after each departure, the system state corresponds to one of the cases described in Lemma 1.

Lemma 2: $\forall m > 2. ahead(m) \Rightarrow ahead(m+1)$.

Proof: To show this we should connect the output of each case in Lemma 1 (the configuration after $n+1$ departures) to the assumed state for some other case. However instead of performing an exhaustive evaluation, we find that we can group cases together in a generic manner. We identify these generic cases and describe how they connect.

Consider the following generic cases: *g1*: different jobs in the first station and no blocking in the future (until the next departure under ω); *g2*: different jobs in the first station with blocking in the future (until the next departure under ω); *g3*: identical jobs with identical residual service times in the first station and identical jobs in the second station; *g4*: identical jobs with identical residual service times in the first station and different jobs in the second station; *g5*: identical jobs with different residual service times in the first station and identical jobs in the second station; *g6*: identical jobs with identical residual service times in the first station and identical jobs with different residual service times in the second station.

Note that in all of the generic cases, $ahead(m)$ holds. Starting from a generic case, when a departure occurs from the first station under ω, the system either navigates to another generic case or it becomes the same under both policies (shown below). In either case, $ahead(m+1)$ holds, as both the generic cases and the situation when systems are the same respect the *ahead* property.
Assume ω' is serving a job in the first station that is already served by ω (e.g. Case 1.1 or 1.2.1.1 in Lemma 1). The worst case scenario (for keeping the system ahead under ω) is when ω' serves its job in the first station and admits the job ω is currently serving in the first station (label this job k). Now the residual service time of this job is t time units less under ω compared to ω'. *g1*: if there is no blocking, job k departs sooner under ω than ω' and the system stays in *g1*. *g2*: otherwise if job k is blocked under ω, the same would happen under ω' up to t time units later, as the same jobs (or jobs previously served at the second station by ω) will be served at the second station under ω' (Cases 1.3 and 1.2.2.1.4 in Lemma 1 are examples). From *g2*, the system can go to either of the following generic cases (i.e. *g3*-*g6*).
For *g3*-*g6*, assume that a specific job is being served in the first station under both policies. Let the residual service times of the job at the first station be identical. *g3*: if the states of the jobs are identical (i.e. the same jobs with identical residual service times) in the second station under both policies, then the policies behave the same (e.g. Case 2.3 in Lemma 1). *g4*: if ω has admitted new jobs in the second station, ω is already ahead in the number of departures and ω' needs to go through the same steps (e.g. Case 1.2.2.1.2 in Lemma 1). The system stays in *g4*. *g6*: if there are identical jobs at the second station with smaller residual service time under ω, jobs depart earlier from the second station under ω (e.g. Cases 1.2.1.2 and 1.2.2.2 in Lemma 1). The system either stays in *g6* or becomes the same under both policies. Now assume the residual service time under ω is less in the first station. *g5*: if jobs are in the same state (i.e. the same jobs with identical residual service times) in the second station under both policies, ω can lead to earlier departures

from the first station (e.g. Cases 1.2.2.1.1 and 3.2 in Lemma 1). The system can go to *g1* or *g2*.

(Lemma 2) □

Appendix C. Corollaries 1 and 2 proofs

In this appendix, we present the proofs of Corollaries 1 and 2.

Corollary 1: *Theorems 1 and 2 hold for i) systems where jobs are waiting at the first station, ii) systems with arrivals, and iii) clearing systems.*

Proof:

In the following we show the proof for Theorem 1. To construct a proof for Theorem 2, replace ω and ω', with π and π', respectively.

(i) The proofs of Theorems 1 and 2 assume that jobs are always waiting at the first station.

(ii) Whenever decisions are made, if there are jobs waiting at the first station, the proof follows (i). If there are no jobs waiting at the first station, we show that ω' does not gain any benefit when there are no arrivals and $ahead(m)$ is not violated. The following cases could occur:

   (a) If there are no free servers at the first station under both policies, the proof follows (i) as long as servers at the first station remain busy.

   (b) If there are free servers at the first station under ω and no free servers at the first station under ω', ω' is serving a job at the first station that ω has already served. Let r be the time when the next arrival occurs and u be the remaining time until the systems become identical under both policies (i.e. the systems are serving the same jobs with the same residual service times). If $r < u$, the proof continues as (i). If $r > u$, the systems are identical under both policies.

   (c) If there are free servers at the first station under ω' and no free servers at the first station under ω, ω' is waiting for a job that ω is serving or has already served. Let r be the time when the next arrival occurs and u be the remaining time until a server becomes idle at the first station under ω. If $r < u$, when the next arrival happens, under ω there have been more departures and/or jobs have less residual service time ($ahead(m)$). If $r > u$, both policies should wait for the next departure. However, the system under ω has serviced more jobs and/or jobs have less residual service time ($ahead(m)$).

(d) If there are free servers at the first station under both policies and if both policies are waiting for the same job, the systems are identical under both policies. Otherwise ω' is waiting for a job that ω has already served. Upon the next arrival, the systems continue working with ω having a higher number of departures from the first station ($ahead(m)$).

In Theorem 2, let r be the time of the next arrival after time t_0. If $r \geq t$, the systems are the same under π and π'. If $r < t$, t is replaced with $t - r$ throughout the proof.

(iii) For a clearing system, the proof is the same as (i) up until the point where there are no more jobs to admit under ω (assuming time t_0 has passed). At this point either:

(a) the systems are identical; from here the policies behave the same and clear the system at the same time.

(b) the same jobs are being served under both policies with smaller residual service times under ω; there are only three jobs left in the system and from Theorems 1 and 2 we know that under ω, the next departure does not happen later than under ω'. Therefore the system reaches a time where there are two jobs in the system under ω and either two or three jobs under ω'. In the best case for ω', assume two jobs are in the system under both policies with identical residual service times. From here the policies clear the system at the same time.

(c) different jobs are being served under the two policies (i.e. ω has completed more jobs than ω'); ω can be left with two jobs in the system sooner than ω'. Therefore it can never be the case that ω' clears the system sooner than ω.

If time t_0 is reached after there are no more jobs to admit, the systems are identical at this point. Employing Lemma 1, we know that the *ahead* property holds and therefore ω' will not complete jobs sooner than ω.

(Corollary 1) $\square$

Corollary 2: *Theorems 1 and 2 and Corollary 1 hold for arbitrary numbers of dedicated and flexible servers.*

Proof: The required extensions are as follows:

- Number of dedicated servers ($N = 2, F = 1$): the case with many dedicated servers is a generalization of the case with one dedicated server at each station. The generalization is as follows: a station including blocked servers is considered a blocked station; a station including only busy servers is considered a busy station; and a station including free servers is considered a free station. For example if the first station is blocked and the flexible server is at the first station, hand-off is possible.

When a station has more than one dedicated server, extending the above proofs becomes more complicated. With only one dedicated server at each station, we had to compare at most three different residual service times. With more dedicated servers, more comparisons are required. Moreover, the order in which the dedicated servers in a station should be considered (for comparison), becomes an issue. In other words, when the system is compared under two different policies, we need to determine the jobs that should be compared.

However, these additional complexities become conceptually easier to manage with the following considerations. Note that jobs are indistinguishable and service times depend only on the station. For example, if more than one job is blocked in the first station, it does not matter which one is chosen for departure (they are indistinguishable). However in order to be able to adapt the proofs, without loss of generality, we assume that both policies choose blocked servers in the same order, as the proofs try to compare the departure times or residual service times of the same jobs (comparing different jobs in the proofs will lead to the same results with the drawback that it would make it far harder for a reader to follow the proof).

In all of the above proofs if the first station is not blocked, consider the two dedicated servers in each of the stations with the earliest completion times. If the first station is blocked, one of the blocked servers and the dedicated server with the earliest completion time in the second stations are considered. Note that the dedicated server with the earliest completion time in the second station is a free dedicated server, if the second station is free.

- Number of flexible servers ($N = 2, F > 1$): introducing more than one flexible server complicates the proofs. For example, if a station contains more than one flexible server, in which order should they be picked for comparison? Or as another example, is it required to compare the residual service times of flexible servers? Also, does the assignment of the flexible servers affect the proofs?

  It turns out that in all of the above proofs, we need to consider only one flexible server at a time. Note that flexible servers do not affect each other in the proof scenarios, i.e. each case of the proof involves only one flexible server and one or two dedicated servers. Assume a first flexible server is assigned to a job. When assigning a second flexible server, the only difference with $F = 1$ is that one of the busy servers is a flexible server. However when analysing the behavior of the second flexible server, we ignore the first flexible server and only consider dedicated servers to perform allocations.

The fact that flexible servers become idle once they complete a job at the second station or hand off to a free dedicated server, makes it possible to treat the flexible servers independently. Note that here we only consider the correctness of the stated properties and do not specify the optimal policy. An optimal policy's decision on allocation of a flexible server will depend on the position of other flexible servers.

(Corollary 2) $\square$