



به نام خدا
دانشگاه صنعتی شریف
دانشکده مهندسی صنایع

«پروژه شبیه‌سازی گسسته پیشامد کارگاه تولیدی»

استاد: دکتر نفیسه صدقی

نگارندگان:

بهزاد یعقوبی ۹۶۱۰۴۷۲۷

مهرداد مرادی ۹۶۱۰۴۵۶۸

زمستان ۱۳۹۹

۱. مقدمه و شرح سیستم..... ۱

۱.۱. مقدمه..... ۱

۱.۲. شرح سیستم (شرح کارگاه مورد مطالعه)..... ۱

۲. توصیف ایستا..... ۳

۲.۱. متغیرهای حالت..... ۳

۲.۲. موجودیت‌ها..... ۴

۲.۳. ویژگی‌ها..... ۴

۲.۴. پیشامدها..... ۴

۲.۵. فعالیت‌ها..... ۴

۲.۶. تأخیرها..... ۵

۲.۷. آماره‌های تجمعی..... ۵

۲.۸. لیست پیشامدهای آتی..... ۷

۲.۸.۱. توضیح ساختار لیست پیشامدهای آتی..... ۷

۲.۸.۲. معرفی اعلان پیشامدهای موجود در این لیست در زمان صفر (لیست اولیه)..... ۸

۳. بیان فرضیات به کار رفته در شبیه سازی..... ۹

۳.۱. زمان جابجایی بین صفوف..... ۹

۳.۲. نظام صف ۹

۳.۳. وضعیت اولیه ۹

۳.۴. ماشین‌های خدمت دهنده A و B ۹

۳.۵. صفوف خدمت‌دهنده C ۹

۳.۶. زمان شبیه‌سازی ۹

۳.۷. محاسبه خروجی‌ها ۱۰

۳.۸. نحوه‌ی تولید اعداد تصادفی ۱۰

۴. معیارهای ارزیابی عملکرد ۱۱

۴.۱. معیارهای ارزیابی خواسته شده مدیریت ۱۱

۴.۱.۱. تعداد سفارشات راه اندازی شده ۱۱

۴.۱.۲. متوسط ضریب اشتغال هر یک از سه ماشین به طور جداگانه (بهره وری ماشین‌ها) ۱۱

۴.۱.۳. میانگین مدت زمان انتظار در صف هر سه ایستگاه خدمت دهی ۱۲

۴.۱.۴. میانگین طول صف برای همه صف‌ها (میانگین زمانی تعداد سفارشات منتظر در صف) ۱۲

۴.۲. معیارهای ارزیابی پیشنهادی ۱۳

۴.۲.۱. متوسط مدت زمان حضور در سیستم ۱۳

۴.۲.۲. نسبت زمانی اشتغال ماشین‌ها در زمان سالم بودن ۱۳

۴.۲.۳. متوسط مدت زمان انتظار برای تعمیر ماشین به ازای هر سفارش قطع خدمت دهی شده ۱۴

۴.۲.۴. نسبت سفارشات قطع خدمت دهی شده ۱۵

۵. توصیف پویا ۱۶

۵.۱. پیشامد ورود سفارشات نوع ۱ به ماشین A ۱۶

۱۶	۵.۲. پیشامد ورود سفارشات نوع ۲ به ماشین B
۱۸	۵.۳. پیشامد اتمام خدمت دهی ماشین A
۱۹	۵.۴. پیشامد اتمام خدمت دهی ماشین B
۲۰	۵.۵. پیشامد خرابی ماشین A
۲۱	۵.۶. پیشامد خرابی ماشین B
۲۲	۵.۷. پیشامد اتمام خرابی ماشین A
۲۳	۵.۸. پیشامد اتمام خرابی ماشین B
۲۴	۵.۹. پیشامد اتمام خدمت دهی ماشین C
۲۵	۶. خروجی و محاسبات
۲۵	۶.۱. معیارهای ارزیابی عملکرد
۲۷	۶.۲. صحت سنجی
۳۲	۶.۳. اعتبار سنجی
۳۶	۶.۴. بینش و تحلیل معیارها
۳۷	۶.۵. تحلیل حساسیت برای بهینه‌سازی و تصمیم‌گیری
۳۷	۶.۵.۱. تحلیل حساسیت زمان بین ورود سفارشات نوع ۲
۴۰	۶.۵.۲. تحلیل حساسیت زمان کارکردن ماشین A
۴۳	۶.۵.۳. تحلیل حساسیت زمان تعمیر ماشین A
۴۶	۶.۵.۴. تحلیل حساسیت زمان خدمت دهی ماشین C

۶.۵.۵. نتایج تحلیل حساسیت برای بهینه سازی و تصمیم گیری ۴۹

۷. سیاست‌های جدید ۵۰

۷.۱. معرفی یک سیاست جدید ۵۰

۷.۲. مقایسه آماری سیاست جدید و سیاست فعلی از منظر تعداد سفارش تکمیل شده ۵۳

۷.۲.۱. روش نمونه گیری مستقل: ۵۳

۷.۲.۲. روش اعداد تصادفی مشترک ۵۵

۷.۲.۳. بررسی میانگین مدت زمان انتظار در صف ماشین A برای سیاست فعلی ۵۸

۸. ارائه‌ی چند سیاست پیشنهادی برای بهبود سیستم ۵۹

۸.۱. شهود نسبت به سیستم ۵۹

۸.۲. سیاست‌های پیشنهادی ۶۰

۹. جداول و نمودارها ۶۱

۹.۱. جداول ۶۱

۹.۲. نمودارها ۶۲

۱۰. پیوست ۶۵

۱۰.۱. شرح کد پایتون ۶۵

۱۰.۲. کد پایتون ۶۵

۱۰.۳. جدول ردیابی ۱۰۱

۱۱. منابع ۱۰۳

۱. مقدمه و شرح سیستم

۱.۱. مقدمه

ریاضیات، زبان طبیعت است، در علم احتمالات و آمار ابزارهایی وجود دارد که با آن‌ها به تحلیل جهان پیرامون می‌پردازیم، با ابزارهای ریاضیاتی که در تحلیل ریاضی صف در علم تئوری صف آموخته می‌شود، می‌توان درکی از سیستم‌های صف پیدا کرد، با ابزارهای مدل‌سازی ریاضی به درک بهتر سیستم‌های پیرامون می‌پردازیم، اما زمانی که سیستم پیچیده می‌شود استفاده از این ابزارها به صرف، ممکن است دیگر کارا نباشد.

شبیه‌سازی، علمی است که با استفاده از قابلیت محاسبه فوق‌العاده کامپیوترها و بر پایه علوم مختلف احتمالات و آمار، تئوری صف، برنامه‌ریزی و مدل‌سازی و ابزارهای دیگر ریاضیاتی به تحلیل یک سیستم پیچیده می‌پردازد، نقطه قوت شبیه‌سازی دادن امکان آزمایش به مهندس است، در حالی که وی در واقع چیزی را در دنیای پیرامون تغییر نمی‌دهد، اما با شبیه‌سازی آن می‌تواند شناخت بسیار ارزشمندی راجع به سیستم‌های اطرافش به دست آورد، حتی در جاهایی که مدل‌های ریاضیاتی نمی‌تواند معادلات را حل کند و مقادیر را محاسبه کند، شبیه‌سازی با کمک قابلیت کامپیوتر، فرصت تجربه کردن و به دست آوردن بینش می‌دهد.

شبیه‌سازی گسسته پیشامد^۱، یکی از شاخه‌های شبیه‌سازی برای سیستم‌های با پیشامدهای گسسته است، این علم یکی از علوم بسیار مهم در تصمیم‌گیری برای مدیران، کارخانه‌ها و هر جا که سیستمی انسانی یا تولیدی برقرار باشد به کار می‌رود.

در این مقاله کارگاهی تولیدی شرح داده می‌شود و سپس شبیه‌سازی شده و در نهایت با استفاده از زبان برنامه‌نویسی پایتون، شبیه‌سازی اجرا شده و تحلیل و بررسی این سیستم انجام شده است، در انتهای مقاله در قسمت پیوست کد پایتون و جدول ردیابی، آمده است.

۱.۲. شرح سیستم (شرح کارگاه مورد مطالعه)

دو نوع سفارش پیوسته به کارگاهی می‌رسد تا به وسیله سه ماشین راه‌اندازی شوند. فواصل زمانی بین ورود دو سفارش نوع ۱ طبق توزیع نمایی با میانگین ۷۰ دقیقه تعریف می‌شود. همگی این سفارش‌ها ابتدا به ماشین A می‌روند. سفارش‌های نوع ۲ در بدو ورود به ماشین B می‌روند و فاصله زمانی بین دو ورود آن‌ها نمایی با میانگین ۱۵ دقیقه است. مدت خدمت‌گیری سفارش نوع ۱ از ماشین A، توزیع یکنواخت در بازه ۳۵ تا ۶۵ دقیقه را دارد. ماشین B به سفارش‌های نوع ۲ طبق توزیع یکنواخت در بازه ۳ تا ۱۳ دقیقه خدمت می‌دهد. سفارش‌ها بعد از گرفتن اولین مرحله خدمت از ماشین A و B همگی به ماشین C می‌روند تا طبق توزیع مثلی با پارامترهای ۶، ۸ و ۱۰ دقیقه قسمت آخر خدمت خود را دریافت کنند و از سیستم خارج شوند. در

¹ Discrete-Event Simulation

قسمت دوم خدمت که در ماشین C صورت می‌گیرد، سفارش‌های نوع ۱ همواره نسبت به سفارش‌های نوع ۲ اولویت دارند و پس از اتمام فعالیت در ماشین A در ابتدای صف ماشین C قرار می‌گیرند.

از سه ماشین تنها دو ماشین A و B در معرض خرابی هستند. مدت کارکرد ماشین A تا لحظه از کار ماندن به صورت $100x+50$ (به دقیقه) است که در آن x از توزیع ویبول با پارامترهای شکل ۲ و مقیاس ۱ پیروی می‌کند. این مدت برای ماشین B به صورت یکنواخت در بازه ۵۰ تا ۳۵۰ دقیقه تعریف می‌شود. با رویداد خرابی ماشین A و B به ترتیب طبق توزیع یکنواخت ۱ تا ۲۹ دقیقه و ۲ تا ۱۸ دقیقه از چرخه فعالیت بیرون می‌مانند.

هر گاه ماشینی خراب شود و سفارشی را در دست انجام داشته باشد، این سفارش با حفظ اولویت در انتظار تعمیر ماشین خواهد ماند تا ادامه خدمت خود را دریافت کند.

۲. توصیف ایستا^۲

مدل سازی مسئله در قالب شبیه سازی به همراه معرفی پیشامدها، فعالیت ها و متغیرهای حالت و غیره (توصیف ایستا).

۲.۱. متغیرهای حالت^۳

۱. طول صف ماشین A

۲. طول صف ماشین B

۳. طول صف سفارشات نوع ۱ ماشین C

۴. طول صف سفارشات نوع ۲ ماشین C

۵. وضعیت ماشین A: این متغیر می تواند مقادیر ۰ و ۱ و ۲ را داشته باشد. اگر مقدار آن ۰ باشد، به معنای خراب بودن ماشین A است؛ اگر مقدار آن ۱ باشد، یعنی ماشین A سالم است اما بیکار (آزاد) است و در صورتی که مقدار آن ۲ باشد، به این معناست که ماشین A هم سالم است و هم مشغول و در حال خدمت دهی می باشد.

۶. وضعیت ماشین B: این متغیر می تواند مقادیر ۰ و ۱ و ۲ را داشته باشد. اگر مقدار آن ۰ باشد، به معنای خراب بودن ماشین B است؛ اگر مقدار آن ۱ باشد، یعنی ماشین B سالم است اما بیکار (آزاد) است و در صورتی که مقدار آن ۲ باشد، به این معناست که ماشین B هم سالم است و هم مشغول و در حال خدمت دهی می باشد.

۷. وضعیت مشغول بودن ماشین C: این متغیر از نوع ۰ و ۱ است و مقدار ۱ به معنای مشغول بودن ماشین C است.

همچنین ۴ متغیر کمکی تعریف کرده ایم که البته، در جدول ردیابی نیز آن ها را برای فهم بهتر، می نویسیم.

۱. مدت زمان باقی مانده از زمان خدمت سفارش قطع خدمت شده نوع ۱ در ماشین A: در صورت خرابی ماشین A و قطع خدمت دهی یک سفارش، مدت زمان باقیمانده از خدمت آن را در یک متغیر می ریزیم، همچنین مقدار پیشفرض این متغیر برابر منفی یک است و در صورتی که سفارش قطع خدمت دهی شده نوع ۱ نداشته باشیم، همان مقدار باقی خواهد ماند.

۲. مدت زمان باقی مانده از زمان خدمت سفارش قطع خدمت شده نوع ۲ در ماشین B: در صورت خرابی ماشین B و قطع خدمت دهی یک سفارش، مدت زمان باقیمانده از خدمت آن را در یک متغیر می ریزیم، همچنین مقدار پیشفرض این متغیر نیز برابر منفی یک است و در صورتی که سفارش قطع خدمت دهی شده نوع ۲ نداشته باشیم، همان مقدار باقی خواهد ماند.

2 Static Description
3 State Variables

۳. سفارش قطع خدمت دهی شده نوع ۱ در ماشین A که در حالت پیشفرض مقدار خالی میگیرد که نشان بدهد سفارش قطع خدمت دهی شده وجود ندارد.

۴. سفارش قطع خدمت دهی شده نوع ۲ در ماشین B که در حالت پیشفرض مقدار خالی میگیرد که نشان بدهد سفارش قطع خدمت دهی وجود ندارد.

۲.۲. موجودیت‌ها^۴

ماشین A – ماشین B – ماشین C – سفارش نوع ۱ – سفارش نوع ۲

۲.۳. ویژگی‌ها^۵

ترتیب ورود در سفارش نوع ۱ یا ۲، مثلاً اولین سفارش ورودی نوع ۱، دومین سفارش نوع ۱ و ...

۲.۴. پیشامدها^۶

۱. ورود سفارشات نوع ۱ به ماشین A (arrival1) ۲. ورود سفارشات نوع ۲ به ماشین B (arrival2)

۳. اتمام خدمت دهی ماشین A (end of service A) ۴. اتمام خدمت دهی ماشین B (end of service B)

(service B)

۵. خرابی ماشین A (out of service A) ۶. خرابی ماشین B (out of service B)

۷. اتمام خرابی ماشین A (being repaired A) ۸. اتمام خرابی ماشین B (being repaired B)

۹. اتمام خدمت دهی ماشین C (end of service C)

۲.۵. فعالیت‌ها^۷

در سیستم تحت بررسی در این پروژه، فعالیتهای زیر وجود دارد که توزیع احتمال مدت زمان انجام هر کدام در صورت پروژه (فایل توضیحات پروژه) به ما داده شده است.

a1 : فواصل زمانی بین ورود دو سفارش نوع ۱

a2 : فواصل زمانی بین ورود دو سفارش نوع ۲

s1 = مدت خدمت گیری سفارش نوع ۱ از ماشین A

s2 = مدت خدمت گیری سفارش نوع ۲ از ماشین B

w1 = مدت کارکرد ماشین A تا لحظه از کار ماندن

w2 = مدت کارکرد ماشین B تا لحظه از کار ماندن

r1 = مدت زمان بیرون ماندن از چرخه فعالیت برای ماشین A (تعمیر)

4 Entities

5 Attributes

6 Events

7 Activities

مدت زمان بیرون ماندن از چرخه فعالیت برای ماشین B (تعمیر) = $r2$

مدت خدمت گیری از ماشین C = $s3$

۲.۶. تأخیرها^۸

در این سیستم، سه نوع تأخیر از جنس در صف ماندن داریم (مربوط به صف های ماشین های A و B و C). همچنین دو نوع تأخیر از جنس منتظر ماندن برای خدمت دهی مجدد داریم (مربوط به سفارشات قطع خدمت دهی شده ماشین های A و B)

۲.۷. آماره های تجمعی^۹

تعداد سفارشات تکمیل شده:

سفارشاتى که توسط ماشین C خدمت رسانی شان به اتمام می رسد را می شمارد.

خدمت رسانی شده توسط ماشین A :

تعداد سفارشات نوع یکی که توسط ماشین A خدمت رسانی شده اند را می شمارد.

خدمت رسانی شده توسط ماشین B :

تعداد سفارشات نوع یکی که توسط ماشین B خدمت رسانی شده اند را می شمارد.

سفارشات نوع یکی که توسط ماشین C خدمت رسانی شده اند

سفارشات نوع دویی که توسط ماشین C خدمت رسانی شده اند

مدت زمان اشتغال سرورها (A و B و C)

مجموع زمان انتظار در صف ماشین A برای سفارشات نوع ۱

برای هر سفارش، زمان خروج از صف منهای زمان ورود به صف را به این آماره اضافه می کند.

مجموع زمان انتظار در صف ماشین B برای سفارشات نوع ۲

برای هر سفارش، زمان خروج از صف منهای زمان ورود به صف را به این آماره اضافه می کند.

مجموع زمان انتظار در صف ماشین C برای سفارشات نوع ۱

برای هر سفارش، زمان خروج از صف منهای زمان ورود به صف را به این آماره اضافه می کند.

مجموع زمان انتظار در صف ماشین C برای سفارشات نوع ۲

برای هر سفارش، زمان خروج از صف منهای زمان ورود به صف را به این آماره اضافه می کند.

مجموع وزنی زمانی طول صف ماشین A برای سفارشات نوع ۱

در هر بار جلوبری زمان، طول صف در بازه ی زمانی ضرب شده و به این آماره افزوده می گردد.

8 Delays

9 Cumulative Statistics

مجموع وزنی زمانی طول صف ماشین B برای سفارشات نوع ۲
در هر بار جلوبری زمان، طول صف در بازه‌ی زمانی ضرب شده و به این آماره افزوده می‌گردد.

مجموع وزنی زمانی طول صف ماشین C برای سفارشات نوع ۱
در هر بار جلوبری زمان، طول صف در بازه‌ی زمانی ضرب شده و به این آماره افزوده می‌گردد.

مجموع وزنی زمانی طول صف ماشین C برای سفارشات نوع ۲
در هر بار جلوبری زمان، طول صف در بازه‌ی زمانی ضرب شده و به این آماره افزوده می‌گردد.

مجموع زمان حضور در سیستم برای سفارشات نوع ۱
برای هر سفارش نوع ۱، زمان خروج از سیستم را منهای زمان ورود به سیستم کرده و در این آماره می‌ریزد

مجموع زمان حضور در سیستم برای سفارشات نوع ۲
برای هر سفارش نوع ۲، زمان خروج از سیستم را منهای زمان ورود به سیستم کرده و در این آماره می‌ریزد

۲.۸. لیست پیشامدهای آتی^{۱۰}

۲.۸.۱. توضیح ساختار لیست پیشامدهای آتی

در اعلان پیشامدها^{۱۱}، ما پیشامدها و زمان های وقوع آنها را می نویسیم.

ورود سفارش i ام نوع ۱ به ماشین A در زمان t : $(arrival1, t, ord1i)$

ورود سفارش i ام نوع ۲ به ماشین B در زمان t : $(arrival2, t, ord2i)$

اتمام خدمت دهی سفارش i ام نوع ۱ توسط ماشین A در زمان t : $(end\ of\ service\ A, t, ord1i)$

اتمام خدمت دهی سفارش i ام نوع ۲ توسط ماشین B در زمان t : $(end\ of\ service\ B, t, ord2i)$

خرابی ماشین A در زمان t : $(out\ of\ service\ A, t)$

خرابی ماشین B در زمان t : $(out\ of\ service\ B, t)$

اتمام خرابی ماشین A در زمان t : $(being\ repaired\ A, t)$

اتمام خرابی ماشین B در زمان t : $(being\ repaired\ B, t)$

اتمام خدمت دهی سفارش i ام نوع ۱ یا ۲ توسط ماشین C در زمان t :

$(end\ of\ service\ C, t, ord2i)$ or $(end\ of\ service\ C, t, ord1i)$

¹⁰ Future Event List (FEL)

¹¹ Event notices

۲.۸.۲. معرفی اعلان پیشامدهای موجود در این لیست در زمان صفر (لیست اولیه)

در ثانیه صفر، با شروع شبیه سازی ۴ پیشامد ساخته می شود؛ اول اینکه با تولید اعداد تصادفی مناسب، ورود اولین سفارش نوع ۱ و اولین سفارش نوع ۲ ساخته می شود، همچنین دو پیشامد خرابی ماشین A و B نیز ساخته می شود، یعنی در ثانیه صفر ماشین های A و B نیز تازه شروع به کار می کنند و در شروع شبیه سازی سالم هستند.

۳. بیان فرضیات به کار رفته در شبیه سازی

۳.۱. زمان جابجایی بین صفوف

زمان جابجا شدن بین صف‌ها قابل صرف نظر کردن است.

۳.۲. نظام صف

رفتار و نظام صف به صورت FIFO^{۱۲} فرض شده است. (نکته : برای سفارشات قطع خدمت دهی شده نیز این نکته صادق است و در ابتدای صف قرار میگیرند.) در این نظام، سفارشی که زودتر وارد شده است، زودتر خدمت دریافت می‌کند.

۳.۳. وضعیت اولیه

ما در این شبیه سازی در مورد وضعیت اولیه سیستم در زمان صفر فرض کرده‌ایم، در ثانیه صفر، با شروع شبیه سازی ۴ پیشامد ساخته می‌شود؛ با تولید اعداد تصادفی مناسب، ورود اولین سفارش نوع ۱ و اولین سفارش نوع ۲ ساخته می‌شود، همچنین دو پیشامد خرابی ماشین A و B نیز ساخته میشود، یعنی در ثانیه صفر ماشین‌های A و B نیز تازه شروع به کار می‌کنند.

۳.۴. ماشین‌های خدمت دهنده A و B

ماشین‌های A و B از زمان صفر شبیه سازی یا خراب هستند یا سالم اند که در حالت سالم بودن یا مشغول اند و یا بیکار. منظور از توزیع‌های احتمال مطرح شده در صورت سوال برای مدت کارکرد ماشین‌های A و B شامل هر دو زمان مشغول بودن و بیکاری می‌شود و منظور از آن همان سالم بودن است.

۳.۵. صفوف خدمت‌دهنده C

با توجه به اولویت سفارشات نوع ۱ نسبت به نوع ۲ در صف مربوط به ماشین C، برای C دو صف موازی در نظر می‌گیریم، یکی برای سفارشات نوع ۱ و یکی برای سفارشات نوع ۲.

۳.۶. زمان شبیه‌سازی

ما در این پروژه، مدت زمان شبیه سازی را برابر ۱ روز (۲۴ ساعت) در نظر می‌گیریم.

¹² First-In-First-Out

۳.۷. محاسبه خروجی‌ها

در این پروژه، همه آماره‌های تجمعی تا زمان آخرین پیشامد قبل از پایان شبیه سازی جمع آوری می‌شوند، برای مثال اگر آخرین پیشامد در ۲۳.۹ ساعت رخ داده باشد، ۰.۱ ساعت پایانی زمان در آماره‌های تجمعی تاثیری نخواهد داشت، دلیل این امر آن است که رفتار خروجی‌ها در صورتی که تا آخرین پیشامد محاسبه شود به واقعیت سیستم نزدیک‌تر است، چرا که برای ما ۲۴ ساعت مستقل از جلوبری زمان شبیه سازی، معنایی ندارد، درحالی که چیزی که به شبیه سازی و خروجی‌ها معنا می‌دهد پیشامدها هستند، یعنی با وارد نکردن ۰.۱ ساعت پایانی، خطر تاثیر عاملی خارج از پیشامدها و شبیه سازی (که در واقع همان انتخاب ۲۴ ساعت است) بر خروجی‌ها را از بین برده‌ایم.

۳.۸. نحوه‌ی تولید اعداد تصادفی

در این شبیه سازی، ما به متغیرهای تصادفی^{۱۳} از توزیع‌های نمایی^{۱۴}، ویبول^{۱۵}، یکنواخت^{۱۶} و مثلثی^{۱۷} احتیاج داشتیم که برای تولید این اعداد، از روش تبدیل تجمعی معکوس^{۱۸} استفاده نمودیم.

13 Random Variables
14 Exponential
15 Weibull
16 Uniform
17 Triangular
18 Inverse-Transform method

۴. معیارهای ارزیابی عملکرد^{۱۹}

۴.۱. معیارهای ارزیابی خواسته شده مدیریت

۴.۱.۱. تعداد سفارشات راه اندازی شده

یکی از مهم‌ترین معیارهای موجود در هر کارگاه تولیدی این معیار است، مدیریت مایل است بداند سیستم فعلی در زمان مشخص، چه تعداد محصول تولید می‌کند، این تعداد، برای برآورد سود و هزینه‌ها به مدیریت کمک میکند و اگر از نقطه سر به سر تولید^{۲۰} پایین‌تر باشد بدین معنی است که سیستم فعلی سودده نخواهد بود. همچنین مدیریت مایل است که تاثیر تغییر پارامترهای مختلف مانند میانگین زمان بین ورود سفارشات نوع ۱ و ۲ را در تعداد سفارشات راه اندازی شده ببیند که در قسمت تحلیل حساسیت این کار انجام گرفته است.

برای محاسبه این معیار، پس از اتمام شبیه سازی، مقدار آماره تجمعی را که تعداد سفارشات تکمیل شده را در خود ذخیره می‌کرد، به عنوان این خروجی گزارش می‌کنیم.

۴.۱.۲. متوسط ضریب اشتغال^{۲۱} هر یک از سه ماشین به طور جداگانه (بهره وری ماشین‌ها)

همانطور که در معادله حسابداری گفته می‌شود، دارایی‌ها می‌تواند از جنس غیر نقدی باشد، یکی از دارایی‌های مهم یک کارگاه تولیدی ماشین‌های آن کارگاه است، در این کارگاه سه ماشین A و B و C موجود است که اولاً برای مجموعه حکم دارایی را دارد، بنابراین لازم است کنترل و نگهداری لازم از آن به عمل آید، ثانیاً به کارانداختن این ماشین‌ها هزینه‌بر است و لازم است تا از میزان بهره‌وری آن‌ها مطلع بود.

این معیار به مدیریت مجموعه دیدگاهی راجع به پتانسیل استفاده نشده این ماشین‌ها بیان می‌کند، آیا وضعیت بهره‌وری در سیستم و به طبع آن، سود می‌تواند بهتر شود؟

همچنین نکته دیگر این است، هرچقدر بهره‌وری افزایش می‌یابد، میزان کارکردن و به طبع آن استهلاک ماشین‌آلات زیادتر می‌شود، بنابراین مهم است که مدیریت مجموعه دیدی دقیق راجع به ساعت کارکرد ماشین‌ها داشته باشد.

نکته مهم دیگر، این است که با دانستن بهره‌وری، می‌توان به جست و جوی راه حل، برای افزایش آن به مقدار لازم رفت، در صورتی که از بهره‌وری اطلاعی نداشته باشیم، حتی در صورت انجام اقدامات متعدد بهبود نمی‌توانیم بهبود را در سیستم اندازه گیری کنیم.

19 performance indicator

20 Break Even Point

21 Utilization

۴.۱.۳. میانگین مدت زمان انتظار در صف هر سه ایستگاه خدمت دهی^{۲۲}

از یک جهت این معیار، به معیار تعداد سفارش تکمیل شده و معیار بهره‌وری ماشین‌ها مربوط است، به این معنی که هرچقدر مدت زمان انتظار کمتر باشد احتمالاً بهره‌وری ماشین‌ها بیشتر شده است و در حال خدمت رسانی بیشتری هستند، یا اینکه ممکن است به این دلیل باشد که پتانسل سیستم ارضا نشده و امکان خدمت رسانی به تعداد بیشتری سفارش وجود دارد، بنابراین مدیریت می‌تواند با اقداماتی در صورت امکان، تعداد این سفارشات را افزایش دهد.

حالتی دیگر کمی متفاوت است، برای بعضی کارگاه‌ها سفارش مورد نظر، ممکن است در صورت گذر زمان بسیار خراب بشود یا غیرقابل کاربرد بشود، یا حتی یک کارگاه ریخته‌گری ممکن است فلزی که باید ذوب شده و در صورت گذشت زمان بسیار به حالت سخت درآید و دیگر نتواند ادامه خدمت خود را ببیند، بنابراین از این جهت نیز مهم است که بدانیم سفارش مورد نظر در بخش سیستم به طور متوسط چه مقدار در صف چه ماشینی منتظر میماند.

یک سناریوی دیگر این است که مدت انتظار در صف پایین باشد و بهره‌وری نیز پایین باشد، فرضیه محتملی که وجود دارد این است که دلیل بهره‌وری پایین، احتمالاً نبود سفارش است و به همین دلیل، صفی هم تشکیل نمی‌شود و سفارشات منتظر نمیمانند، بنابراین در این حالت می‌توان با تحلیل حساسیت نرخ ورود سفارشات (یا زمان‌های بین ورود) واکنش بهره‌وری و سفارشات تکمیل شده سیستم را به این پارامتر سنجید.

۴.۱.۴. میانگین طول صف برای همه صف‌ها (میانگین زمانی تعداد سفارشات منتظر در صف)^{۲۳}

این معیار برای هر یک از سه ماشین A و B و C به طور جداگانه تعریف می‌شود و از رابطه زیر محاسبه می‌شود. لازم به ذکر است که مقدار کمتر این شاخص عملکرد برای ما مطلوب تر است.

$$\hat{L}_Q = \frac{1}{T} \times \sum_{i=0}^{\infty} i \times T_i^Q$$

در رابطه فوق، T_i^Q برابر با مجموع مدت زمانی است که در آن بازه‌های زمانی دقیقاً i سفارش در داخل صف خدمت دهی آن ماشین قرار دارند و T برابر با مدت زمان شبیه سازی می‌باشد.

درک و تفسیر این شاخص ارتباط زیادی با شاخص قبل دارد، هرچه میانگین طول صف بیشتر باشد میانگین مدت زمان انتظار صف نیز بیشتر است، زیاد بودن این عدد می‌تواند نشان دهنده‌ی کم بودن بهره‌وری ماشین‌ها باشد یا می‌تواند نشان دهنده کم بودن تعداد خدمت دهندگان باشد.

یکی از سناریوهای ممکن، این است که میانگین مدت انتظار زیاد باشد اما طول صف به نسبت کم باشد، در این حالت به نظر می‌رسد سفارشات با نرخ آرامی به سیستم می‌رسند و زمان خدمت دهی نیز زیاد است.

22 Average queue Waiting Time

23 Time-Weighted-Average number of orders waiting in queue

در حالتی دیگر، ممکن است مدت انتظار کم باشد اما طول صف به نسبت زیاد باشد، در اینجا به نظر می‌رسد ممکن است نرخ ورود سفارشات به سیستم بالا باشد و زمان خدمت دهی نیز کم باشد.

۴.۲. معیارهای ارزیابی پیشنهادی

یکی از کارکردهای مهم معیارهای ارزیابی عملکرد، دادن بینش نسبت به سیستم است، لازم است که با محاسبه این شاخص‌ها راجع به رفتار سیستم‌ها بینش و شناخت پیدا کنیم، سیستم‌هایی که در حدی پیچیدگی دارند که با نگاهی بدون استفاده از ابزار شبیه سازی نمی‌توان درکی از آن‌ها داشت، همچنین این شاخص‌ها به تصمیم‌گیران کمک میکند تا بر محور شاخص‌های مشخص تصمیم‌گیری کنند و این تصمیم‌گیری‌ها را در بوته شبیه سازی به اجرا بگذارند.

۴.۲.۱. متوسط مدت زمان حضور در سیستم^{۲۴}

یکی از مهم‌ترین معیارها در زبان ساده این است، سفارشات نوع ۱ و ۲، از لحظه ورود تا زمانی که خدمت نهایی‌شان را دریافت می‌کنند و از سیستم خارج می‌شوند، به طور متوسط چه زمانی در سیستم حضور دارند، برای مدیریت مهم است که بدانند در نهایت، سیستم پیچیده‌ای که وجود دارد در چه زمانی می‌تواند سفارش نوع ۱ و در چه زمانی سفارش نوع ۲ ورودی را خدمت رسانی کند، این مهم، برای کارگاه‌های با تولید بالا بسیار مهم است، به ویژه اگر سفارشات ورودی ارزش بالایی داشته باشند.

همچنین یک کارگاه تولیدی را در نظر بگیرید، این کارگاه در مورد کیفیت خود چه چیزهایی می‌تواند به شما بگوید تا شما ترغیب شوید که سفارشات را به آن‌ها بدهید؟ اگر در نهایت سفارشاتتان به طور متوسط زودتر آماده شوند، قاعدتاً تمایل شما برای تحویل سفارشاتتان به آن‌ها بیشتر خواهد بود.

همچنین در قسمت تحلیل حساسیت، مدیریت می‌تواند با استفاده از ابزار شبیه سازی با تغییر پارامترهای مختلف، تاثیر هر یک را بر این معیار شناسایی کرده و در صورت لزوم تغییراتی را حاصل دهد.

۴.۲.۲. نسبت زمانی اشتغال ماشین‌ها در زمان سالم بودن^{۲۵}

این معیار برای هر یک از سه ماشین A و B و C به طور جداگانه تعریف می‌شود و از رابطه زیر محاسبه می‌شود. لازم به ذکر است که مقدار کمتر این شاخص عملکرد برای ما مطلوب تر است.

²⁴ Average time spent in system

²⁵ busy time ratio

$$\text{working time ratio} = \frac{\text{مجموع مدت زمان اشتغال ماشین مورد نظر}}{\text{مجموع مدت زمان سالم بودن آن ماشین}}$$

این دسته از شاخص ها به صورت نسبت بین ۰ و ۱ تعریف میشوند و مشخص میکنند که هر یک از ماشین ها، در چند درصد از زمان هایی که سالم می باشد، در حال کار بوده است. فرق این معیار با متوسط ضریب اشتغال که در معیارهای خواسته شده مدیریت بود این است که در اینجا ما حساب میکنیم در مواقعی که سالم است چقدر کار میکند، در آنجا میگوییم در کل زمان چقدر شاغل است، چون ماشین C زمان خرابی ندارد، بنابراین این معیار تنها برای ماشین های A و B محاسبه می شود. راجع به ماشین های A و B، این شاخص ها اطلاعات دقیق تری از ضریب اشتغال به ما می دهند، یکی از سناریوهای جالب که ممکن است رخ دهد این است که اگر این شاخص بالا باشد و متوسط ضریب اشتغال پایین باشد یعنی آن ماشین کهنه و فرسوده است و زود به زود خراب می شود.

۴.۲.۳. متوسط مدت زمان انتظار برای تعمیر ماشین به ازای هر سفارش قطع خدمت دهی شده^{۲۶}

این معیار برای ماشین های A و B به طور جداگانه تعریف می شود و از رابطه زیر محاسبه می شود.

$$\text{Average waiting time for repairing machine} = \frac{\text{مجموع مدت زمان های انتظار برای تعمیر ماشین نظر}}{\text{تعداد کل سفارشات قطع خدمت دهی شده}}$$

این دسته از شاخص ها نشان مشخص میکند که به طور متوسط هر یک از سفارشات که به دلیل خرابی ماشین مربوطه، قطع خدمت دهی شده اند، چه مدت زمانی را برای تعمیر آن ماشین منتظر می مانند. طبیعتاً مقدار کمتر برای این شاخص مطلوبتر است. این شاخص به مدیریت می گوید، سفارشات که قطع خدمت دهی شده اند به طور متوسط چه مدت برای تعمیر ماشین مورد نظر صبر کرده اند، هر چه این عدد زیادتر باشد، یعنی بهره وری ماشین مورد نظر پایین تر است، احتمالاً ابزارهای تعمیراتی ما ضعیف هستند یا ماشین مورد نظر کهنه است و به سختی تعمیر می شود.

مدیریت مجموعه از این شاخص می تواند برای سنجش بخش تعمیرات، که ممکن است یک نفر یا یک دپارتمان باشد استفاده کند، همچنین زمان انتظار برای تعمیر بینش خوبی در ارتباط به سایر شاخص ها می دهد؛ برای مثال ممکن است در صورتی که بهره وری پایین بوده باشد این شاخص دلیل آن را به ما بگوید، یا اگر میانگین انتظار سفارشات در صف زیاد باشد یکی از دلایلیش احتمالاً زیاد بودن مدت تعمیر ماشین است.

²⁶ Average waiting time for repairing machine

همچنین در صورت داشتن اطلاعات دقیق تر می توان تحلیل های هزینه ای نیز انجام داد، مسائلی از این قبیل که بهبود مدت زمان کارکرد نسبت سود به هزینه اش با سیستم فعلی چه خواهد بود؟ آیا میتوان دو شاخص را تغییر داد و نسبت سود به هزینه بهتری گرفت؟ چه شاخص هایی باید تغییر کند تا نسبت بهتری گرفته شود؟

۴.۲.۴. نسبت سفارشات قطع خدمت دهی شده^{۲۷}

این معیار برای ماشین های A و B به طور جداگانه تعریف میشود و از رابطه زیر محاسبه میشود.

$$\text{Stop of being serviced ratio} = \frac{\text{تعداد سفارشات قطع خدمت دهی شده}}{\text{تعداد کل سفارشات}}$$

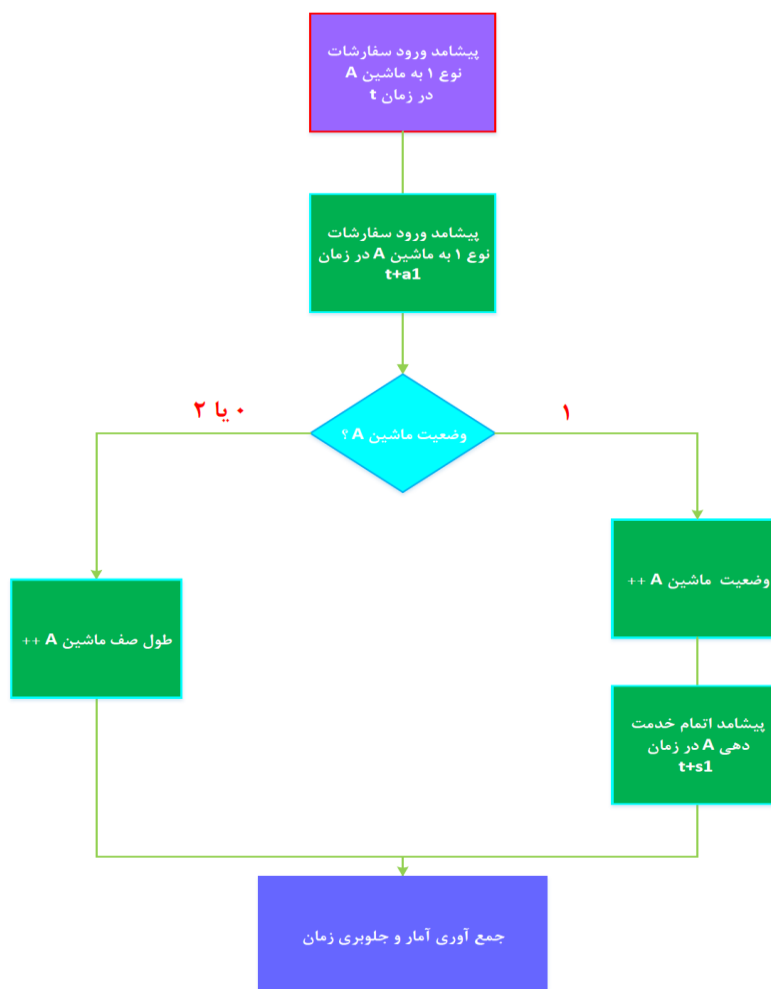
این دسته از شاخص ها تعیین میکند که چه نسبتی از کل سفارشات دچار قطع خدمت دهی شده اند. واضح است که مقدار کوچکتر برای این شاخص مطلوب تر است. این شاخص دیدی می دهد که چه درصدی از سفارشات قطع خدمت دهی می شوند، اگر این درصد بزرگ تر باشد باید به فکر تعویض ماشین یا تعمیر اساسی آن بیفتیم، در صورتی که عدد، مقدار معقولی باشد می توانیم با ماشین های کنونی ادامه بدهیم. همچنین به طور جزئی تر با تحلیل این شاخص در کنار سایر شاخص ها به درک عمیق تری از رفتار سیستم خواهیم رسید.

²⁷ Stop of being serviced ratio

۵. توصیف پویا^{۲۸}

همان طور که در بخش توصیف ایستا (خواسته اول پروژه) توضیح دادیم، در این سیستم ۹ پیشامد وجود دارد که قبلاً آنها را معرفی کردیم. حال در این قسمت، به ازای هر یک از این ۹ پیشامد، نمودار جریان^{۲۹} مربوطه را رسم میکنیم.

۵.۱. پیشامد ورود سفارشات نوع ۱ به ماشین A

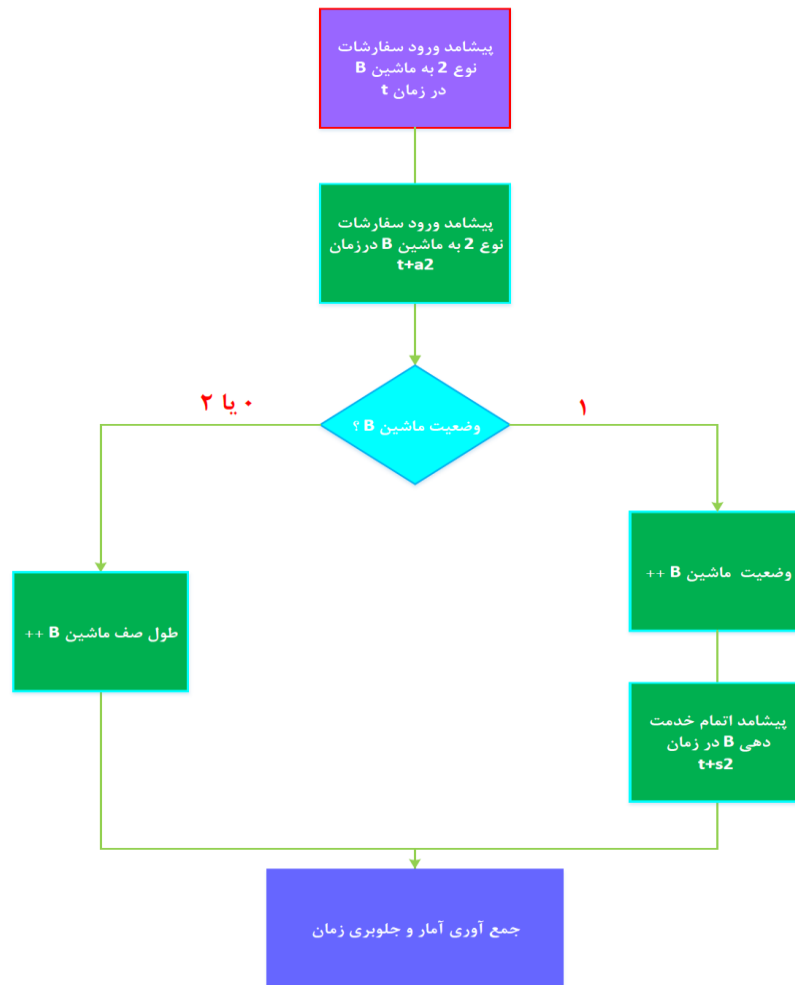


نمودار ۱، پیشامد ورود سفارشات نوع ۱ به ماشین A

۵.۲. پیشامد ورود سفارشات نوع ۲ به ماشین B

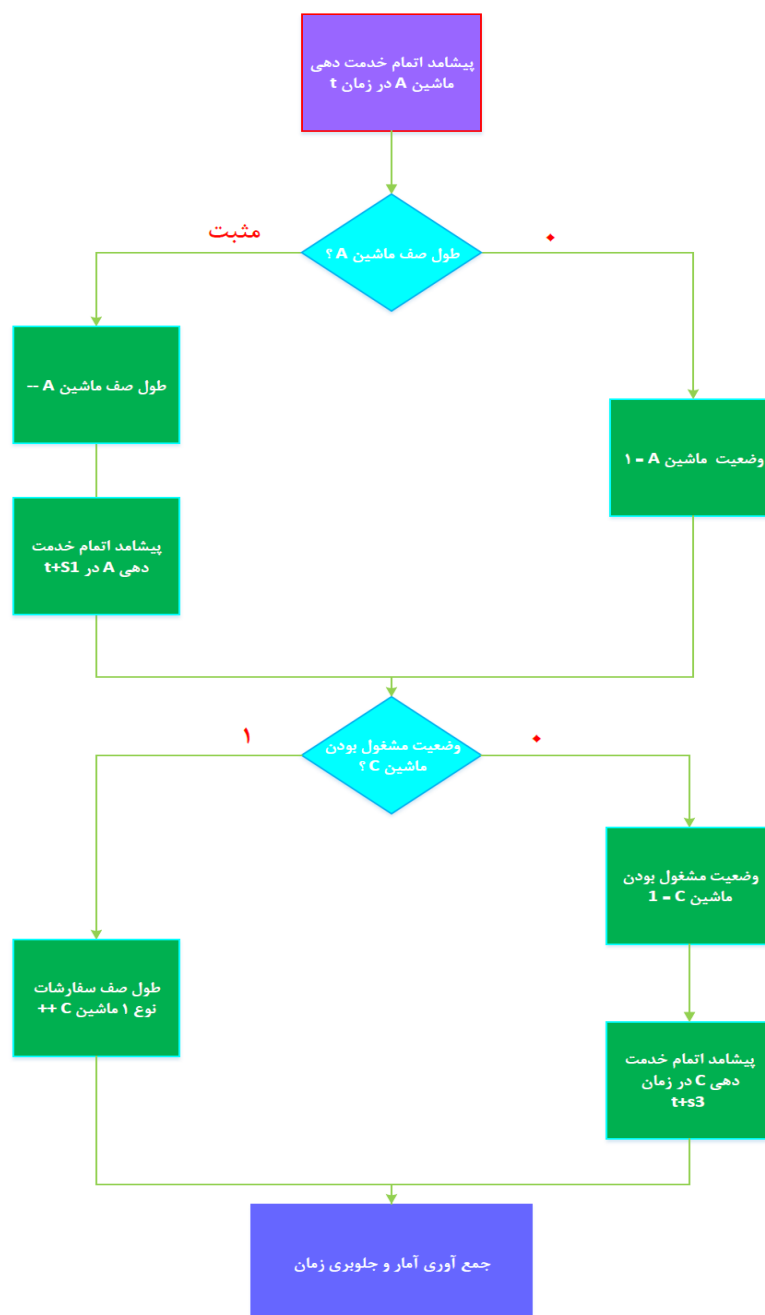
²⁸ Dynamic description

²⁹ Flow chart diagram



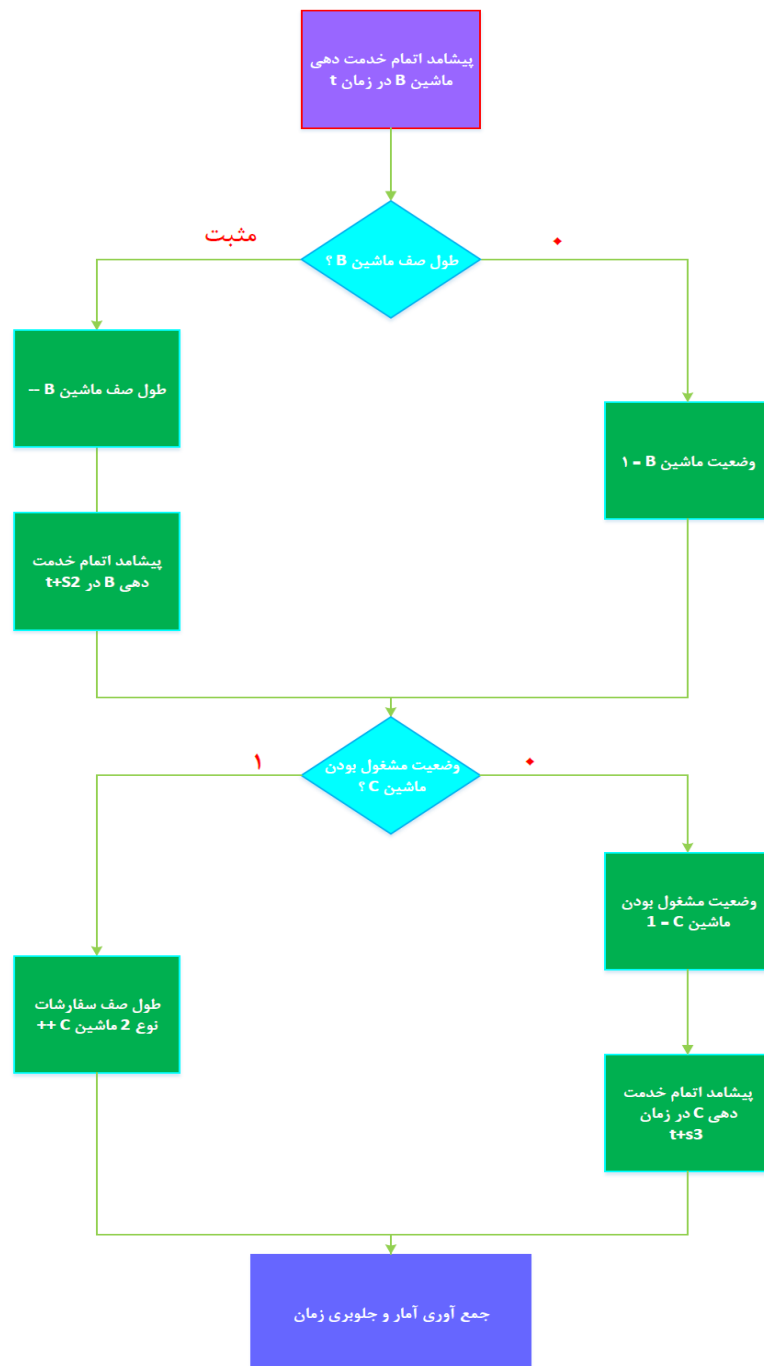
نمودار ۲، ورود سفارشات نوع ۲ به ماشین B

۵.۳. پیشامد اتمام خدمت دهی ماشین A



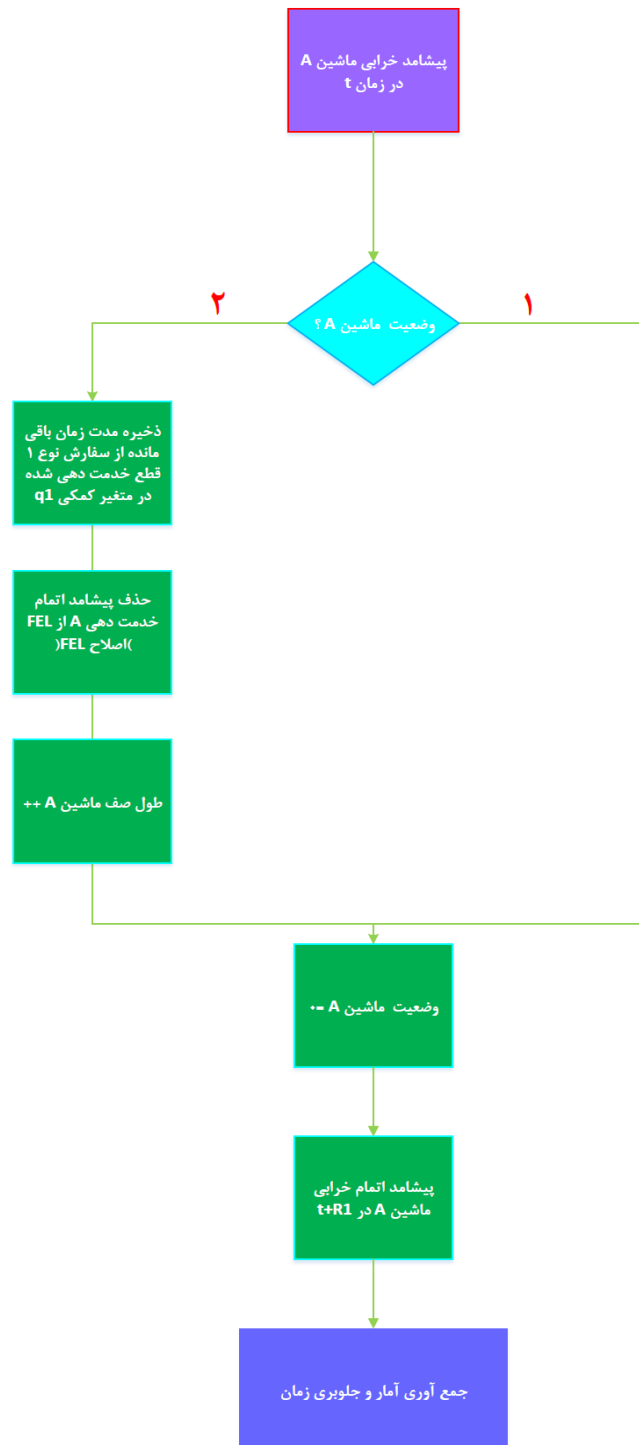
نمودار ۳، پیشامد اتمام خدمت دهی ماشین A

۵.۴. پیشامد اتمام خدمت دهی ماشین B



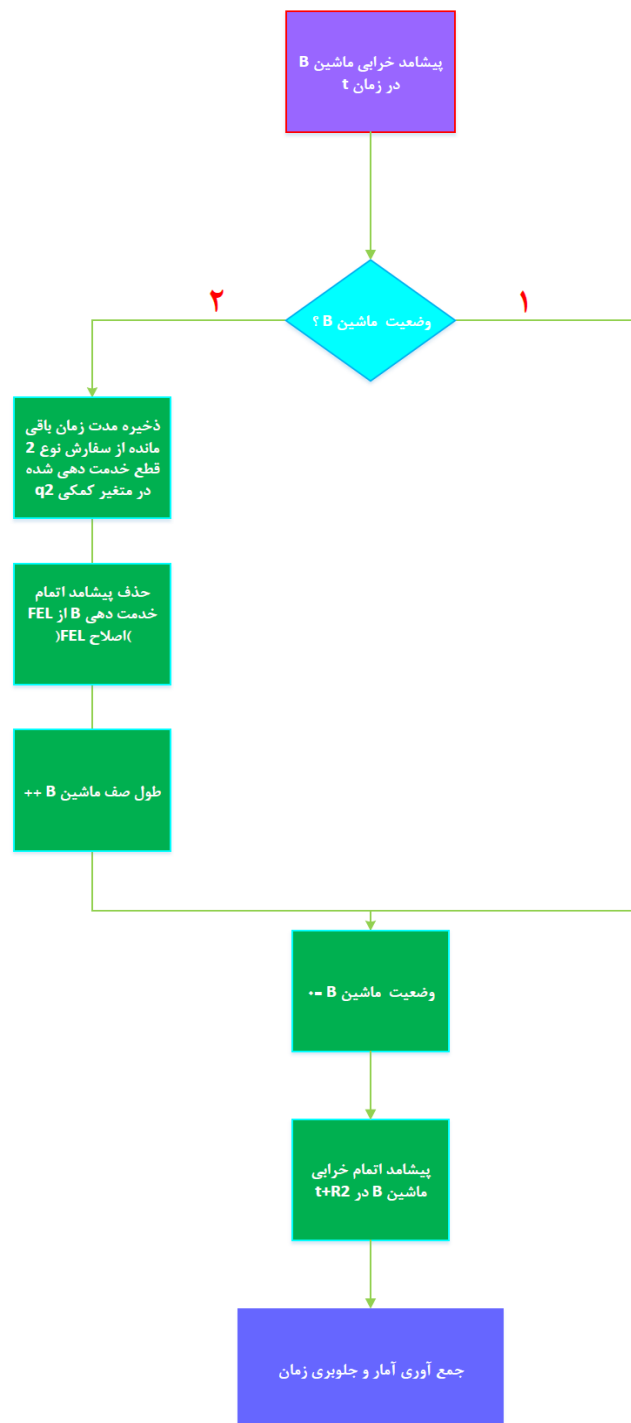
نمودار ۴، پیشامد اتمام خدمت دهی ماشین B

۵.۵. پیشامد خرابی ماشین A



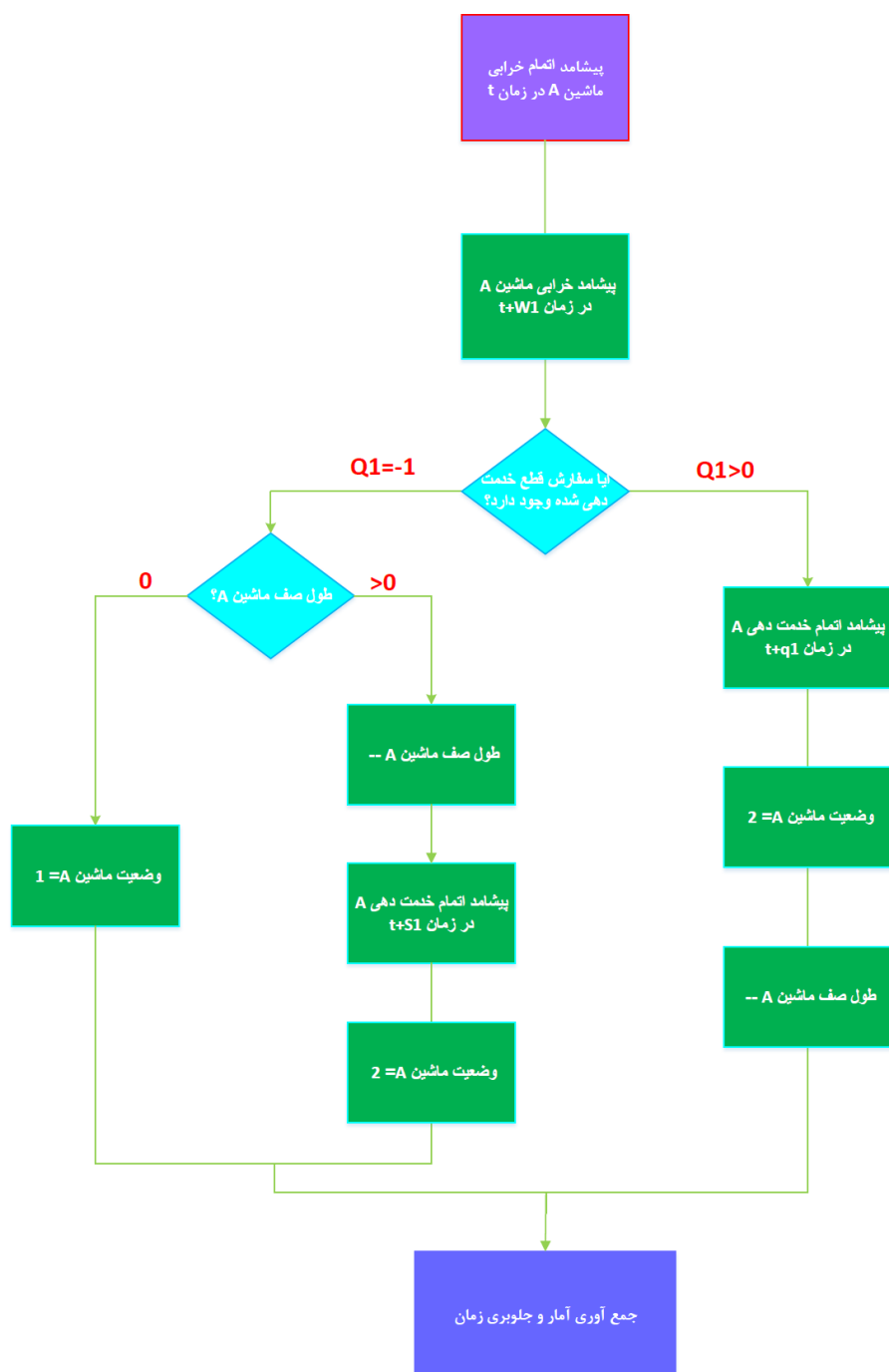
نمودار ۵، پیشامد خرابی ماشین A

۵.۶. پیشامد خرابی ماشین B



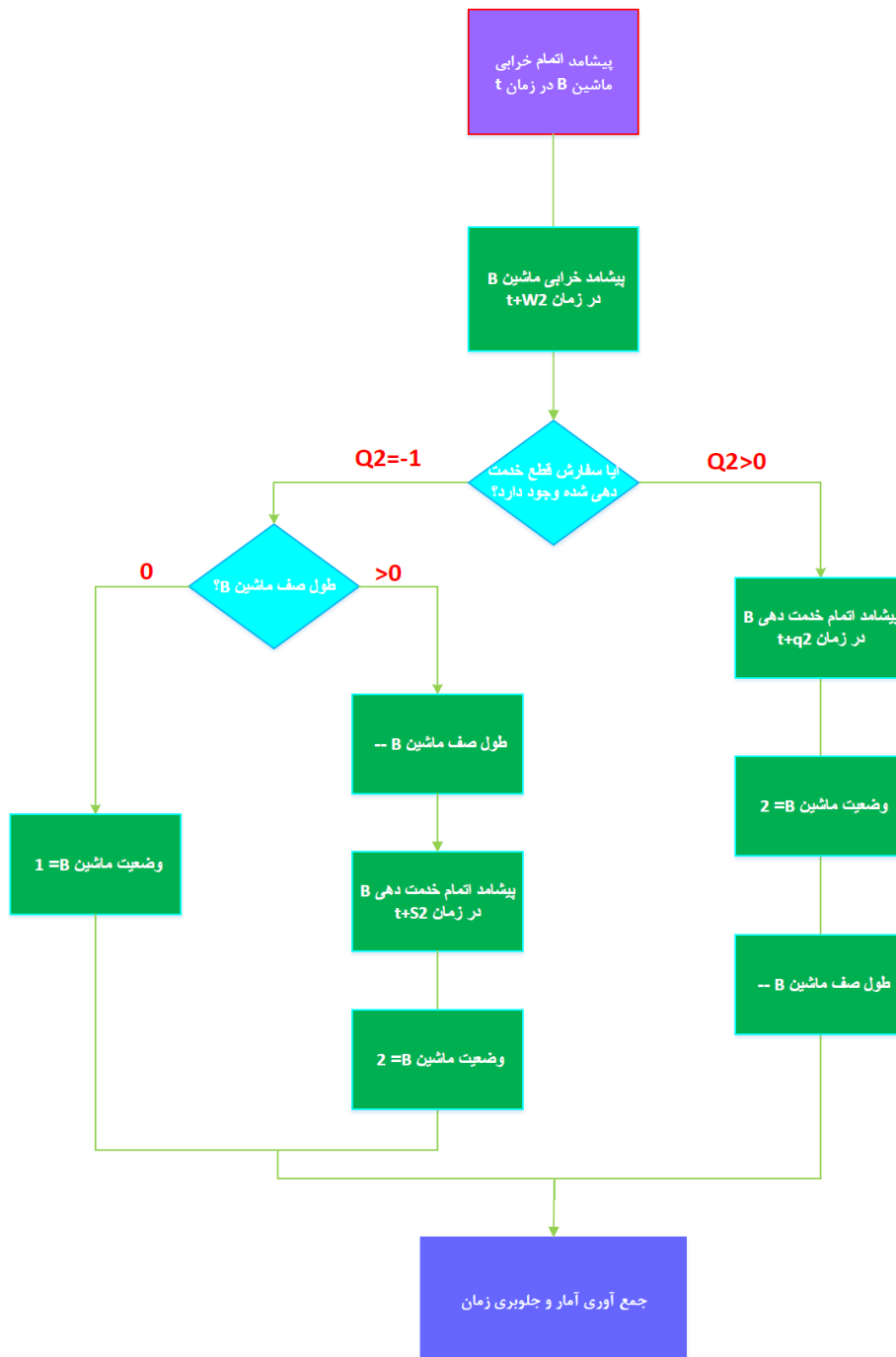
نمودار ۶، پیشامد خرابی ماشین B

۵.۷. پیشامد اتمام خرابی ماشین A



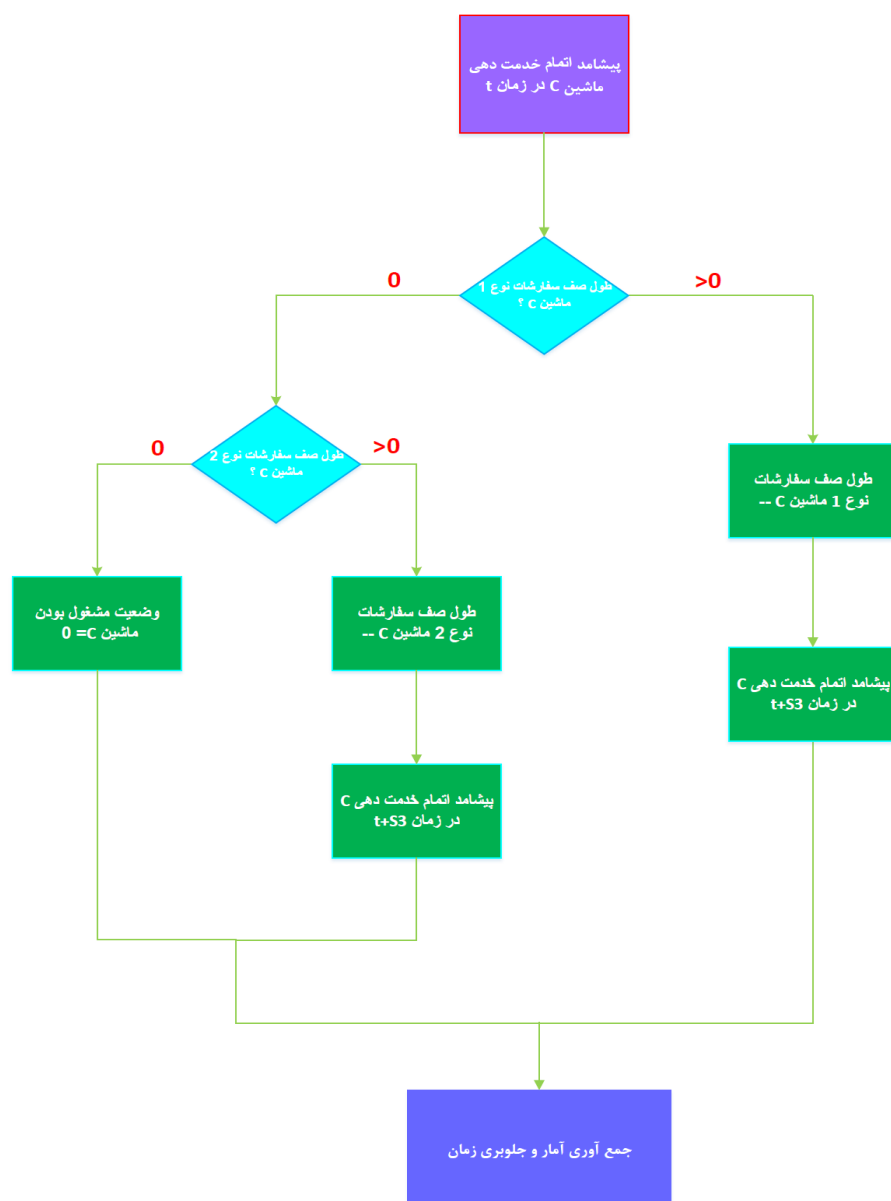
نمودار ۷، پیشامد اتمام خرابی ماشین A

۵.۸. پیشامد اتمام خرابی ماشین B



نمودار ۸، پیشامد اتمام خرابی ماشین B

۵.۹. پیشامد اتمام خدمت دهی ماشین C



نمودار ۹، پیشامد اتمام خدمت دهی ماشین C

۶. خروجی^{۳۰} و محاسبات

۶.۱. معیارهای ارزیابی عملکرد

با استفاده از کد پایتون که در پیوست ۲ موجود است، ۱۰۰ بار شبیه سازی را اجرا کرده و از همه خروجی‌های مورد نیاز میانگین گرفتیم که در جدول زیر نتایج حاصل را مشاهده می‌کنید، قابل ذکر است که واحد زمانی ما در تمام این مقاله ساعت می‌باشد.

جدول ۱، خروجی‌های بدست‌آمده از صد بار تکرار شبیه‌سازی در ۲۴ ساعت با آلفای ۰.۰۵

ردیف	نام خروجی	نام خروجی به انگلیسی	برآوردگر نقطه‌ای	کران پایین بازه اطمینان	کران بالای بازه اطمینان
۱	سفارشات انجام شده	completed orders	۱۱۱.۹۸۰	۱۰۹.۹۶۸	۱۱۳.۹۹۱
۲	بهره وری ماشین A	utility A	۰.۶۳۷	۰.۶۱۰	۰.۶۶۴
۳	بهره وری ماشین B	utility B	۰.۵۲۶	۰.۵۱۵	۰.۵۳۷
۴	بهره وری ماشین C	utility C	۰.۶۲۳	۰.۶۱۱	۰.۶۳۴
۵	میانگین وزنی زمانی طول صف A	mean lq A	۱.۱۳۹	۰.۹۰۵	۱.۳۷۴
۶	میانگین وزنی زمانی طول صف B	mean lq B	۰.۴۳۹	۰.۳۹۴	۰.۴۸۵
۷	میانگین وزنی زمانی طول صف C	mean lq C	۰.۲۶۲	۰.۲۳۸	۰.۲۸۶

				C	
۰.۰۲۸	۰.۰۲۴	۰.۰۲۶	mean lq Cord1	میانگین وزنی زمانی طول صف سفارشات نوع ۱	۸
				C	
۰.۲۵۹	۰.۲۱۲	۰.۲۳۵	mean lq Cord2	میانگین وزنی زمانی طول صف سفارشات نوع ۲	۹
				C	
۱.۳۵۹	۰.۹۸۸	۱.۱۷۳	mean wq A	میانگین مدت انتظار صف	۱۰
				A	
۰.۱۱۶	۰.۰۹۸	۰.۱۰۷	mean wq B	میانگین مدت انتظار صف	۱۱
				B	
۰.۰۵۸	۰.۰۵۰	۰.۰۵۴	mean wq C	میانگین مدت انتظار صف	۱۲
				C	
۰.۰۳۶	۰.۰۳۲	۰.۰۳۴	mean wq Cord1	میانگین مدت انتظار صف سفارشات نوع ۱ C	۱۳
۰.۰۶۳	۰.۰۵۳	۰.۰۵۸	mean wq Cord2	میانگین مدت انتظار صف سفارشات نوع ۲ C	۱۴
۲.۰۵۵	۱.۷۶۵	۱.۹۱۰	mean being in system for order1	میانگین مدت ماندن در سیستم سفارشات نوع ۱	۱۵
۰.۴۳۷	۰.۴۱۲	۰.۴۲۴	mean being in system for order2	میانگین مدت ماندن در سیستم سفارشات نوع ۲	۱۶

۱۷	میانگین مدت ماندن در سیستم همه‌ی سفارشات	mean being in system for all orders	۰.۷۰۰	۰.۶۶۲	۰.۷۳۸
----	---	---	-------	-------	-------

۶.۲. صحت سنجی^{۳۱}

در هر کار شبیه سازی، بعد از ساختن مدل مفهومی^{۳۲} و مدل عملیاتی، باید به سراغ صحت سنجی برویم، به این معنی که مدل عملیاتی که در اینجا یک برنامه‌ی پایتون است، باید دقیقاً مطابق با مدل مفهومی باشد و در واقع باید مطمئن شویم که مدل مفهومی به درستی در مدل عملیاتی پیاده سازی شده است. علاوه بر رفع اشکالات کد در زمان کدزنی، روش ما برای صحت سنجی این است که سه مقدار از پارامترهای ورودی را در نظر گرفته و آن‌ها را تغییر می‌دهیم، اگر تغییرات خروجی مطابق با انتظار بود پس می‌توانیم بگوییم مدل عملیاتی به خوبی، مدل مفهومی را نشان می‌دهد.

خروجی‌های نوشته شده در جدول، متوسط ۱۰۰ بار تکرار شبیه سازی می‌باشد و زمان نیز به واحد ساعت اندازه گیری شده است. با بررسی اعداد داخل جدول تغییرات منطقی در پاسخ به تغییرات پارامترها رخ می‌دهد، بنابراین نتیجه می‌گیریم که برنامه عملیاتی نوشته شده، به خوبی مدل مفهومی را ارائه می‌کند.

ب عنوان یک مثال، اگر میانگین زمان بین ورود سفارشات نوع ۲، افزایش یابد، طبق انتظار، بهره‌وری ماشین B کاهش می‌یابد.

در جدول زیر، برای هر پارامتر، خروجی‌هایی که مستقیماً تحت تاثیر هستند و انتظار تغییرات در آن‌ها داریم قرمز شده‌اند.

دقت کنید که تغییر میانگین زمان کارکرد A تاثیر زیادی نداشته است (بر خلاف انتظار)، دلیل این است که در کل تعداد سفارشات نوع ۱ بسیار کم بوده و همچنین پیشامد خرابی نیز در ۲۴ ساعت به تعداد کمی رخ می‌دهد، این تحلیل به ما بینشی نیز راجع به سیستم می‌دهد که در تحلیل‌های آینده برایمان بسیار مفید است.

همچنین نکته‌ی دیگر این است که با تغییر پارامتری مربوط به سفارشات نوع ۱، این سفارش در صف ماشین C از آنجا که بر نوع ۲ اولویت دارند، در صف نوع ۱ و نوع ۲ ماشین C تاثیر خود را خواهد گذاشت. به طور عکس نیز، تغییر پارامتر مربوط به سفارشات نوع ۲، خروجی‌های مربوط به سفارشات نوع ۱ در نقطه برخوردشان یعنی صف C را تغییر خواهد داد.

³¹ Verification

³² Conceptual Model

جدول ۲، صحت سنجی خروجی‌های بدست‌آمده از صد بار تکرار شبیه‌سازی در ۲۴ ساعت

ردیف	نام خروجی	نام خروجی به انگلیسی	میانگین زمان ورود بین سفارشات نوع ۲			میانگین زمان کارکرد ماشین A			میانگین زمان خدمت دهی ماشین C		
۱			۵۰ درصد کمتر	میانگین فعلی	۵۰ درصد بیشتر	۵۰ درصد کمتر	میانگین فعلی	۵۰ درصد بیشتر	۵۰ درصد کمتر	میانگین فعلی	۵۰ درصد بیشتر
۲	سفارشات انجام شده	completed orders	۱۷۲.۵۸	۱۱۱.۲۴	۸۳.۱۱	۱۱۲.۹۸	۱۱۱.۶۶	۱۱۳.۲۶	۱۱۳.۰۳	۱۱۲.۵۰	۱۰۷.۴۹
۳	بهره وری ماشین A	utility A	۰.۶۵۴	۰.۶۳۶	۰.۶۵۳	۰.۶۲۹	۰.۶۲۷	۰.۶۴۲	۰.۶۴۲	۰.۶۵۸	۰.۶۳۷
۴	بهره وری ماشین B	utility B	۰.۹۱۹	۰.۵۲۳	۰.۳۶۱	۰.۵۳۲	۰.۵۲۵	۰.۵۳۰	۰.۵۲۷	۰.۵۲۸	۰.۵۲۵
۵	بهره وری ماشین C	utility C	۰.۹۶۱	۰.۶۱۹	۰.۴۶۳	۰.۶۲۹	۰.۶۲۱	۰.۶۳۰	۰.۳۱۴	۰.۶۲۷	۰.۸۹۸
۶	میانگین وزنی زمانی	mean lq A	۱.۱۴۷	۰.۹۷۰	۱.۰۶۹	۱.۴۸۱	۰.۹۵۹	۰.۹۰۲	۱.۰۳۲	۱.۰۳۶	۰.۹۸۴

										طول صف A	
۰.۴۶۲	۰.۴۱۹	۰.۴۵۰	۰.۴۴۰	۰.۴۳۶	۰.۴۳۵	۰.۱۵۵	۰.۴۳۹	۱۲.۵۸۹	mean lq B	میانگین وزنی زمانی طول صف B	۷
۳.۳۷۵	۰.۲۷۱	۰.۰۱۹	۰.۲۸۲	۰.۲۷۶	۰.۲۹۶	۰.۱۰۵	۰.۲۶۸	۵.۲۱۶	mean lq C	میانگین وزنی زمانی طول صف C	۸
۰.۰۶۷	۰.۰۲۹	۰.۰۰۸	۰.۰۲۸	۰.۰۲۷	۰.۰۲۸	۰.۰۱۹	۰.۰۲۸	۰.۰۵۱	mean lq Cord1	میانگین وزنی زمانی طول صف سفارشات نوع ۱ C	۹
۳.۳۰۸	۰.۲۴۲	۰.۰۱۲	۰.۲۵۴	۰.۲۴۸	۰.۲۶۸	۰.۰۸۵	۰.۲۳۹	۵.۱۶۵	mean lq Cord2	میانگین وزنی زمانی طول صف	۱۰

										سفارشات نوع ۲ C	
۱.۰۵۹	۱.۰۹۶	۱.۰۸۶	۰.۹۴۳	۱.۰۲۶	۱.۵۸۵	۱.۰۸۹	۱.۰۳۸	۱.۱۶۵	mean wq A	میانگین مدت انتظار صف A	۱۱
۰.۱۱۳	۰.۱۰۳	۰.۱۰۹	۰.۱۰۸	۰.۱۰۶	۰.۱۰۵	۰.۰۵۶	۰.۱۰۹	۱.۵۵۹	mean wq B	میانگین مدت انتظار صف B	۱۲
۰.۶۸۲	۰.۰۵۶	۰.۰۰۴	۰.۰۵۸	۰.۰۵۷	۰.۰۶۱	۰.۰۲۹	۰.۰۵۵	۰.۶۶۷	mean wq C	میانگین مدت انتظار صف C	۱۳
۰.۰۸۷	۰.۰۳۷	۰.۰۰۹	۰.۰۳۶	۰.۰۳۷	۰.۰۳۸	۰.۰۲۵	۰.۰۳۶	۰.۰۶۴	mean wq Cord1	میانگین مدت انتظار صف سفارشات نوع ۱ C	۱۴
۰.۷۹۸	۰.۰۶۱	۰.۰۰۳	۰.۰۶۳	۰.۰۶۱	۰.۰۶۶	۰.۰۳۰	۰.۰۵۹	۰.۷۳۸	mean wq Cord2	میانگین مدت انتظار صف	۱۵

										سفارشات نوع ۲ C	
۱.۹۱۵	۱.۸۵۷	۱.۷۲۸	۱.۶۹۵	۱.۸۰۲	۲.۲۳۹	۱.۸۲۲	۱.۸۳۵	۱.۹۳۷	mean being in system for order1	میانگین مدت ماندن در سیستم سفارشات نوع ۱	۱۶
۱.۲۰۸	۰.۴۲۳	۰.۳۰۸	۰.۴۳۱	۰.۴۲۶	۰.۴۳۰	۰.۳۴۹	۰.۴۲۷	۲.۱۹۵	mean being in system for order2	میانگین مدت ماندن در سیستم سفارشات نوع ۲	۱۷
۱.۳۳۹	۰.۶۸۸	۰.۵۷۱	۰.۶۶۳	۰.۶۷۸	۰.۷۶۲	۰.۷۲۲	۰.۶۸۴	۲.۱۷۷	mean being in system for all orders	میانگین مدت ماندن در سیستم همه سفارشات	۱۸

۶.۳. اعتبار سنجی^{۳۳}

همچنین در هر کار شبیه سازی، لازم است که فرآیند اعتبار سنجی به درستی و دقت زیاد انجام شود؛ اعتبارسنجی، یعنی بررسی تطابق مدل مفهومی و مدل عملیاتی با سیستم واقعی^{۳۴}. در اعتبارسنجی، می‌خواهیم مطمئن شویم که مدل‌های ساخته شده توسط ما، آیا نمایانگر و تصویر درستی از دنیای واقعی هستند یا خیر. اگر اعتبار سنجی به درستی انجام نشود، به هیچ عنوان نمی‌توان از نتایج شبیه سازی برای تحلیل سیستم واقعی استفاده کرد، نمی‌توان به بینش به دست آمده از آن اتکا کرد و نمی‌توان سیاست بهبود پیشنهاد داد.

در این نوشته، از آنجا که کارگاه تولیدی نه به صورت واقعی، بلکه به صورت فرضی و توصیفی مفهومی داده شده بود، پس ما نمی‌توانیم فرآیند اعتبارسنجی را به طور کامل و واقعی انجام دهیم، اما در اینجا توضیح مبسوطی برای این فرآیند ارائه خواهیم کرد تا خواننده‌ی گرامی بعدها در صورت لزوم از آن استفاده کند:

• ساخت مدل با اعتبار صوری^{۳۵} بالا

یعنی مدل طوری ساخته شود که به صورت ظاهری و منطقی درست باشد؛ یکی از کارهایی که می‌توان در این مرحله انجام داد این است که بعضی از پارامترها را تغییر دهیم و ببینیم آیا خروجی‌ها به طور منطقی تغییر می‌کنند یا خیر. در اعتبار صوری مدل، نظر همه‌ی خبرگان و همچنین استفاده‌کنندگان مختلف مدل شبیه سازی مهم است (باید دید و نظر آنان را در نظر بگیریم)؛ همچنین اعتبار صوری بالا، باید در همه‌ی مراحل ساخت مدل مفهومی و مدل عملیاتی (در اینجا کد پایتون) مد نظر قرار گیرد. مثلاً، در این کارگاه، اگر میانگین زمان بین ورود سفارشات نوع ۱ را پایین ببریم، به عنوان یک اصل کلی انتظار داریم هرچه نرخ ورود سفارشات نوع ۱ بالا رود، بهره‌وری ماشین ۱ افزایش یابد، طول صف و میانگین انتظار سفارشات نوع ۱ نیز - اگر از حدی بیشتر ورود کند - بالا رود، همچنین انتظار داریم این اتفاق برای ماشین C و صف آن نیز رخ دهد.

مثال‌هایی دیگر: تغییر نرخ ورود سفارش نوع ۱، نوع ۲، تغییر نرخ خدمت دهی ماشین‌های A, B و C

• اعتبار سنجی فرضیات مدل^{۳۶}

فرضیات در درون مدل به دو دسته تقسیم می‌شوند:

• ساختاری^{۳۷} و مربوط به منطق مدل:

³³ Validation

³⁴ Real World System

³⁵ face Validation

³⁶ model assumptions validation

³⁷ Structural assumptions

مربوط به ساختار است؛ مثلاً در مدل فرض کردیم که در صف ماشین C سفارشات نوع ۱ همواره ارجحیت دارند یا اینکه با رفتن یک سفارش در صورت آزاد بودن ماشین، بلافاصله خدمت بعدی شروع می‌شود.

• داده‌ها^{۳۸} و پارامترهای ورودی:

مثلاً اینکه توزیع ورود سفارشات نوع ۱ و ۲ چه توزیع با چه پارامترهایی است و ... در مورد فرض‌های داده‌ها، تست‌های آماری^{۳۹} بسیار خوبی وجود دارد، دو آزمون آماری مهم، یکی آزمون کولموگروف-اسمیرنوف^{۴۰} و یکی هم مربع کای^{۴۱} است. همچنین توجه نمایید که کل فرایند اعتبارسنجی از طریق کالیبراسیون^{۴۲} رخ می‌دهد، به این معنی که در یک فرایند تکرار شونده^{۴۳}، در هر مرحله با مقایسه سیستم واقعی و مدل، از تفاوت‌های این دو درس گرفته و مدل را اصلاح می‌کنیم، سپس، دوباره به مقایسه سیستم واقعی و مدل می‌پردازیم و دوباره اصلاحات لازم را انجام می‌دهیم و این فرایند را تا جایی انجام می‌دهیم که به خطای قابل قبولی برسیم. در هر مرحله، اصلاح مدل، می‌تواند شامل بررسی و تطابق و آزمون فرض‌های ساختاری و منطقی یا فرض‌های داده‌های ورودی باشد.

مثلاً، در این پروژه:

- جمع آوری داده‌ها از سیستم واقعی در مورد ورود سفارشات نوع ۱
- استفاده از ابزارهای غیر رسمی مانند نمودار Q-Q plot
- استفاده از نظرات خبرگان برای پی بردن به توزیع داده‌ها
- استفاده از آزمون‌های رسمی آماری کولموگروف-اسمیرنوف و مربع کای
- همچنین اگر در حین فرایند اعتبارسنجی، تفاوت معناداری بین مدل و سیستم واقعی مشاهده کردیم، مشکوک می‌شویم که ممکن است ناشی از فرض‌های ساختاری یا فرض‌های داده‌های ورودی و فرض‌های ورودی باشد، در اینجا به سیستم واقعی رفته و با مشاهده سیستم، می‌بینیم که آیا فرضی را اشتباه در نظر گرفته‌ایم یا اینکه فرضی هست که اضافه در نظر گرفته باشیم یا فرضی هست که در نظر نگرفته باشیم.
- همچنین، ممکن است در کارگاه مورد نظر، در ساعاتی توزیع ورودها متفاوت و نرخ ورودها بالا برود (ساعات شلوغی)، به همین دلیل دوباره با جمع آوری داده‌ها و مشاهده سیستم، حدس‌هایی در مورد توزیع داده‌های ورودی می‌زنیم، سپس با

³⁸ Data assumptions

³⁹ Statistical tests

⁴⁰ Kolmogorov Smirnov test

⁴¹ Chi square test

⁴² Calibration

⁴³ Iterative Process

آزمون‌های آماری آن‌ها را به دقت بررسی می‌کنیم) ممکن است توزیع عوض نشود، صرفاً پارامترهای توزیع عوض شود).

- مثال‌های دیگر: نوشتن تمام فرضیات و مشاهده‌ی سیستم؛ مثلاً اولویت سفارش نوع ۱ به ۲، خرابی ماشین‌ها و بازگشت سفارش قطع خدمت‌دهی شده به ابتدای صف، مدت زمان تعمیر ماشین‌ها (توزیع داده‌ها)

• تبدیل ورودی-خروجی مدل⁴⁴

برای این روش، ابتدا تا جای ممکن باید یکسان بودن فرضیات ساختاری و منطقی مدل و فرضیات در مورد داده‌ها و پارامترهای ورودی مدل با سیستم واقعی بررسی شده‌باشند، بعد از این کار، دو مجموعه داده از ورودی‌ها و خروجی‌های سیستم واقعی جمع‌آوری می‌کنیم (ورودی‌ها را برای روش بعدی که معرفی می‌شود باید جمع‌آوری کنیم)، سپس، داده‌های متناظر را با آزمایش شبیه‌سازی تولید می‌کنیم (دقت کنید، تناظر بین داده‌ها بسیار مهم است)، سپس، با یک آزمون آماری (برای مثال توزیع دانشجو) و یک سطح معنی‌داری⁴⁵، بررسی می‌کنیم که خروجی سیستم واقعی و آزمایش شبیه‌سازی آیا تفاوت معنی‌داری دارند یا خیر، دقت کنید برای اینکه متوجه باشیم که چه تعداد داده جمع‌آوری کنیم می‌توانیم از نمودارهای منحنی عملکرد در کنار خطای نوع ۲ مجاز، استفاده کنیم، همچنین هزینه‌های ناشی از جمع‌آوری داده و تولید آن نیز باید مد نظر قرار بگیرد.

مثال در پروژه‌ی کارگاه تولیدی:

- فرض کنید ساعت کاری کارگاه مورد نظر از ۸ صبح شروع شده و تا ۲۴ ساعت بعد از آن ادامه پیدا می‌کند، شرایط اولیه‌ی ساعت ۸ صبح همواره یکسان است (کل کارگاه خالی است و سفارشی وجود ندارد)، با این تنظیمات به سراغ شبیه‌سازی رفته و برای ۲۴ ساعت با شرایط اولیه‌ی معین، داده تولید می‌کنیم (دقت کنید که این شبیه‌سازی، یک شبیه‌سازی منقطع⁴⁶ و نه پایا⁴⁷ است)، سپس با آزمون آماری، خروجی‌ها را با داده‌های واقعی مقایسه می‌کنیم.
- اگر تفاوت معناداری مشاهده کردیم، به سراغ بررسی فرضیات متفاوت می‌رویم، آن‌ها را به دقت تغییر می‌دهیم و اصلاح می‌کنیم، سپس به سراغ تولید دوباره داده از آزمایش شبیه‌سازی و آزمون آماری می‌پردازیم و این فرآیند را تا جایی انجام می‌دهیم که دیگر با توجه به سطح معنا‌داری‌مان، تفاوت معناداری مشاهده نکنیم.

⁴⁴ Input Output Transformations

⁴⁵ Significance level

⁴⁶ Terminating Simulations

⁴⁷ Steady State Simulation

در نهایت، مدل شبیه سازی را دوباره با داده‌های جدید (مجموعه داده‌ی دوم که از سیستم جمع‌آوری نمودیم و در فرآیند اعتبار سنجی استفاده نکردیم)، مقایسه و اعتبار سنجی می‌کنیم. این کار برای جلوگیری از بیش برآزش^{۴۸} است.

- به عنوان کارهای بیشتر در این روش، از مدیریت میزان دقت مورد نیاز برای تصمیم‌گیری را جویا می‌شویم، تعداد داده‌های مورد نیاز و قدرت آزمون را به دست می‌آوریم (با استفاده از نمودارهای منحنی عملکرد)، با تولید آن تعداد و جمع آوری آن تعداد از سیستم (به صورت متناظر)، همه‌ی خروجی‌ها (تعداد سفارشات تکمیل شده، میانگین طول صف‌ها، میانگین مدت زمان انتظارها) را به صورت آماری با هم مقایسه می‌کنیم و فرآیند را ادامه می‌دهیم.

• استفاده از داده‌های تاریخی^{۴۹} در اعتبار سنجی

در این روش، به جای تولید عدد تصادفی برای آزمایش شبیه سازی، از داده‌های جمع‌آوری شده‌ی سیستم استفاده می‌کنیم (که در توضیح روش قبل گفته شد)، سپس خروجی‌های شبیه‌سازی را با خروجی‌های جمع‌آوری شده‌ی سیستم قبل به صورت آماری و با آزمون‌های آماری مقایسه می‌کنیم. در واقع این روش مانند تبدیل ورودی_خروجی مدل است، با این تفاوت که داده‌های ورودی آزمایش شبیه سازی از سیستم واقعی جمع‌آوری شده است؛ همچنین دقت کنید که هر داده‌ای که از سیستم جمع می‌شود در مدل، باید به همان منظور استفاده شود، مثلاً زمان خدمت‌دهی ماشین C که از سیستم واقعی جمع شده، در آزمایش شبیه سازی باید به عنوان ورودی برای خدمت‌دهی ماشین C استفاده شود.

⁴⁸ Overfitting

⁴⁹ Historical Input Data

۶.۴. بینش و تحلیل معیارها

در ابتدا لازم است ماشین‌های A و B را مقایسه کنیم؛ ورود سفارشات به ماشین A از توزیع نمایی با میانگین ۷۰ دقیقه و ورود سفارشات به ماشین B دارای توزیع نمایی با میانگین ۱۵ دقیقه است، بنابراین انتظار داریم سفارشات بیشتری به ماشین B برسد، همچنین زمان خدمت دهی ماشین A یک توزیع یکنواخت با میانگین ۵۰ دقیقه است، در حالی که ماشین B در توزیعی یکنواخت با میانگین ۸ دقیقه خدمت رسانی می‌کند، بنابراین در اینجا هم ماشین B سریع‌تر و بهتر عمل می‌کند، همچنین مدت زمان کارکرد ماشین A دارای میانگین حدوداً ۱۳۸ دقیقه‌ای است، در حالی که مدت زمان کارکرد ماشین B توزیعی یکنواخت با میانگین ۲۰۰ دقیقه دارد، در نهایت مدت زمان تعمیر ماشین A توزیعی یکنواخت با میانگین ۱۵ دقیقه‌ای دارد، در حالی که مدت زمان تعمیر ماشین B دارای توزیع یکنواخت با میانگین ۱۰ دقیقه است؛ همانطور که شرح داده شد، در همه‌ی این زمینه‌ها ماشین B عملکرد بهتری دارد، اما با نگاهی به بهره‌وری ماشین B در جدول قبل می‌بینیم در مقایسه با ماشین A بهره‌وری کمتری دارد، دلیل این امر چیست؟ و چرا بهره‌وری ماشین A بیشتر است؟

با توجه به میانگین طول صف و میانگین مدت انتظار در صف این دو ماشین در جدول قبل، می‌بینیم که در نگاهی کلی، صف کمتری برای ماشین B تشکیل می‌شود، همه‌ی این شرح‌ها می‌تواند ما را به این فرضیه برساند که ممکن است ماشین B پتانسیل بالقوه‌ی زیادی داشته باشد که به دلیل نداشتن سفارشات نوع ۲ کافی نمی‌تواند آن‌ها را عملی کند، بنابراین تحلیل حساسیت نخست را بر روی میانگین زمان‌های بین ورود سفارشات نوع ۲ انجام می‌دهیم، به این صورت که وضعیت فعلی میانگین (۱۵ دقیقه) را از ۱۰ درصد مقدار فعلی تا ۲ برابر مقدار فعلی تغییر می‌دهیم و تاثیر آن را بر روی ۱۷ خروجی جدول قبل مشاهده می‌کنیم، انتظار داریم با سریع‌تر شدن زمان بین ورود سفارشات نوع ۲، بهره‌وری ماشین B افزایش یافته و همچنین طول صف و مدت انتظار آن نیز افزایش یابد، از طرفی به دلیل افزایش سفارشات نوع ۲ انتظار داریم با افزایش این پارامتر، بهره‌وری ماشین C، طول صف مربوط به سفارش نوع ۲ آن و همچنین مدت بودن در سیستم نیز افزایش یابد.

قابل ذکر است که همه خروجی‌های محاسبه شده در همه مراحل تحلیل حساسیت با ۱۰۰ بار تکرار و در نهایت، میانگین گیری به دست آمده‌اند.

مورد دیگری که موجود است، این است که در ردیف ۱۵ و ۱۶ جدول خروجی‌ها، مشاهده می‌شود مدت بودن در سیستم برای سفارشات نوع ۱ به طرز فاحشی بالاست، از آنجا که در فرضیات مسئله اولویت خدمت دهی ماشین C برای سفارشات نوع ۱ بود، به نظر می‌رسد سفارشات نوع ۱ دارای اهمیت ویژه‌ای هستند، بنابراین برای به دست آوردن بینش راجع به راه حل کم کردن مدت بودن سفارشات نوع ۱ در سیستم، به تحلیل حساسیت مواردی که می‌تواند در این زمان تاثیر بگذارد می‌پردازیم، از جمله‌ی این موارد مدت زمان کارکرد ماشین A تا خرابی، مدت زمان تعمیر ماشین A و مدت زمان خدمت دهی ماشین C می‌باشند که در ادامه همه این موارد با فرض ثابت بودن سایر موارد تحلیل حساسیت شده‌اند، لازم به ذکر است که در همه تحلیل حساسیت‌ها، پارامترها از ۱۰ درصد مقدار فعلی تا ۲۰۰ درصد مقدار فعلی تغییر کرده‌اند.

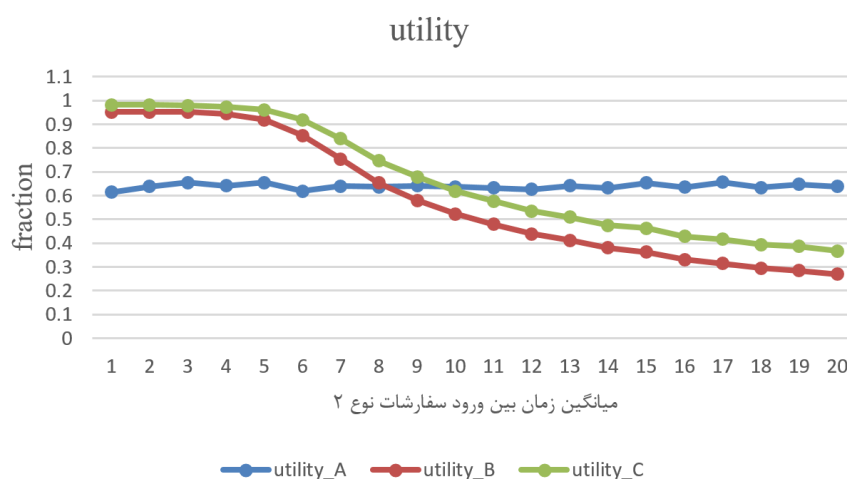
۶.۵. تحلیل حساسیت برای بهینه‌سازی و تصمیم‌گیری^{۵۰}

۶.۵.۱. تحلیل حساسیت زمان بین ورود سفارشات نوع ۲

در نمودار زیر مشاهده میشود که با زیاد کردن میانگین زمان بین ورودهای سفارش‌های نوع ۲، از نصف میانگین کنونی به بعد، تغییر فاحشی در بهره‌وری ماشین‌های B و C رخ میدهد، مدیریت کارگاه با توجه به هزینه‌های مربوط به استهلاک این ماشین‌ها می‌تواند نقطه‌ای را برای میانگین زمان بین ورود سفارش‌های نوع ۲ در نظر بگیرد و بر همان اساس، ورود سفارش‌های نوع ۲ را تنظیم نماید.

البته دقت کنید، همواره ممکن است مسئله پیچیده شود و مسائل دیگری در کار بیاید، در نهایت این نمودارها به نوعی اطلاعات خام^{۵۱} محسوب شده و با توجه به هدف و هزینه‌ها و محدودیت‌ها و امکانات موجود باید تصمیم‌گیری نمود.

یکی دیگر از نکات جالب این نمودار، همبستگی^{۵۲} بهره‌وری ماشین‌های B و C است و قابل انتظار است، چرا که هرچه ماشین B سفارش بیشتری ترخیص کند، ماشین C نیز سفارش بیشتری ترخیص خواهد کرد، همچنین به دلیل اینکه ماشین A تعداد سفارش کمی ترخیص میکند این همبستگی بسیار مشهود است.



نمودار ۱۰، متوسط بهره‌وری ماشین‌ها با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی

⁵⁰ Sensitivity Analysis

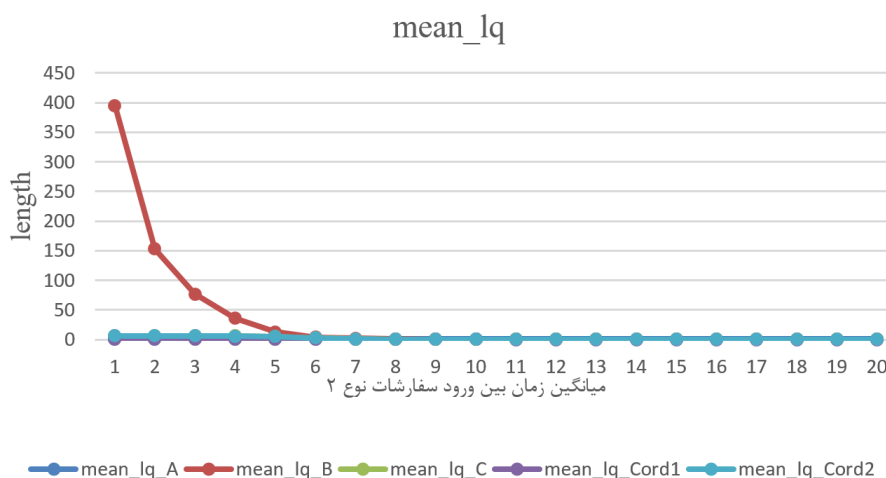
⁵¹ Raw data

⁵² Correlation

با توجه به نمودار زیر به نظر می‌رسد، هرچه قدر میانگین زمان بین ورود از نصف حالت فعلی کمتر شود، تعداد سفارش تکمیل شده افزایش نمی‌یابد و بنابر نمودار قبل، بهره‌وری نیز به حداکثر خود می‌رسد و دیگر تغییری نمی‌کند (به نزدیک ۱ می‌رسد) بنابراین قاعدتا طول صف و میانگین انتظار در صف B و C افزایش خواهد یافت، بنابراین به نظر می‌رسد مقدار بهینه پارامتر زمان بین ورود سفارشات نوع ۲ نصف مقدار حالت فعلی است.

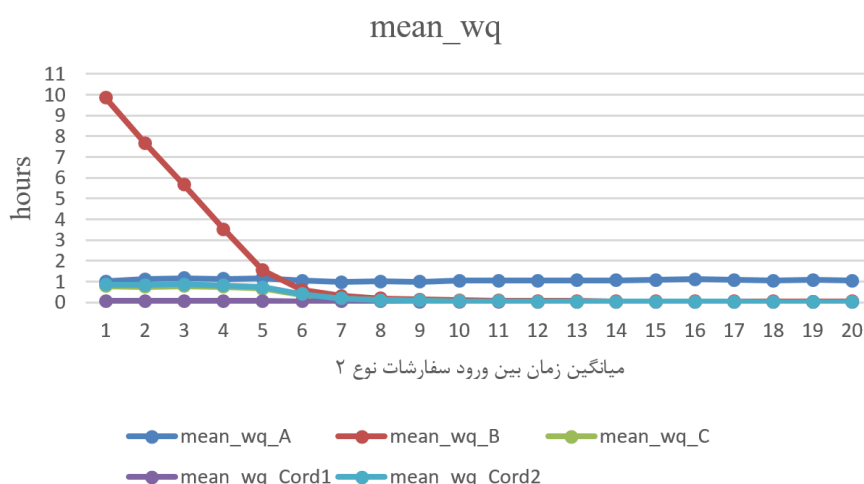


نمودار ۱۱، متوسط سفارشات تکمیل شده با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی



نمودار ۱۲، متوسط طول صف ماشین‌ها با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی

همانطور که میبینید فرضیه مطرح شده تایید شد و به نظر می‌رسد از نصف مقدار فعلی این پارامتر به پایین‌تر میانگین طول صف B به شدت افزایش یافته بنابراین باز هم نقطه ۰.۵ بهینه انتخاب می‌شود، اما چیزی که جالب است این است که صف C تقریباً همیشه ثابت و نزدیک به صفر است، پس به نظر می‌رسد حتی اگر ماشین B با حداکثر توان کار کند باعث ایجاد صف در C نمیشود، البته بنابر نمودار مربوط به بهره‌وری، از نصف مقدار فعلی پارامتر به پایین، بهره‌وری ماشین C نیز تقریباً ۱ است و گنجایش ترخیص بیشتر سفارش را ندارد.

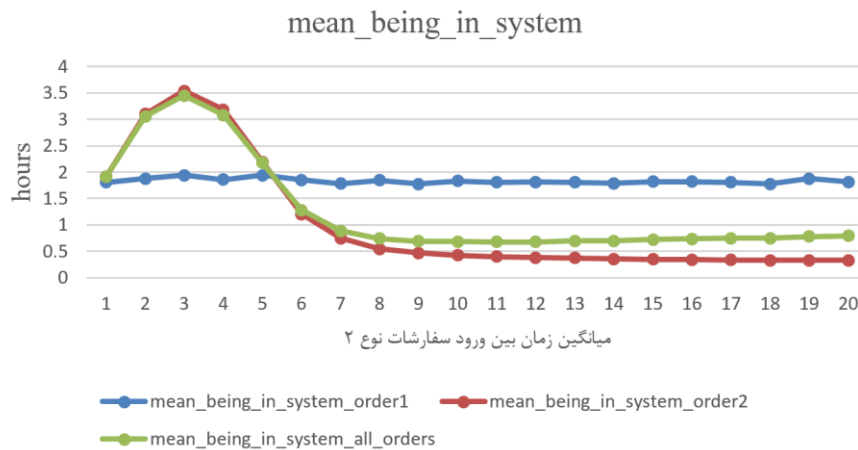


نمودار ۱۳، متوسط زمان انتظار در صفوف ماشین‌ها با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی

در نمودار بالا رفتار جالب را می‌بینیم، از ۰.۶ مقدار فعلی پارامتر به پایین مدت زمان انتظار صف B به شدت زیاد شده است، بنابراین به نظر می‌رسد در حول و حوش نصف حالت فعلی، باز هم شرایط بسیار خوبی خواهیم داشت.

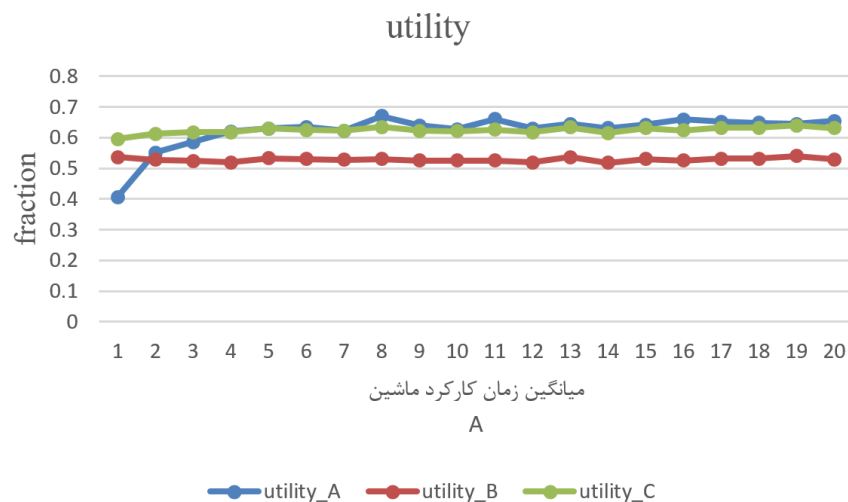
در نمودار بعد نیز همین رفتار را در مورد مدت حضور در سیستم می‌بینید. نکته جالب نمودار زیر استقلال مدت زمان حضور در سیستم برای سفارشات ۱ از همه تغییرات انجام شده است. همچنین دو نکته‌ی جالب دیگر در مورد این نمودار وجود دارد، از ۰.۳ حالت فعلی به پایین‌تر آن‌قدر تعداد سفارشات نوع ۲ زیاد شده که با وجود مجموعاً حضور بیشتر سفارشات نوع ۲ در سیستم، میانگین آن‌ها کم شده است.

همچنین این نمودار نشان می‌دهد روند صعودی مدت حضور در سیستم برای سفارشات نوع ۲، تقریباً از ۰.۸ حالت فعلی به پایین آغاز شده است، باز هم این نمودار نقطه ۰.۵ را تایید می‌کند.



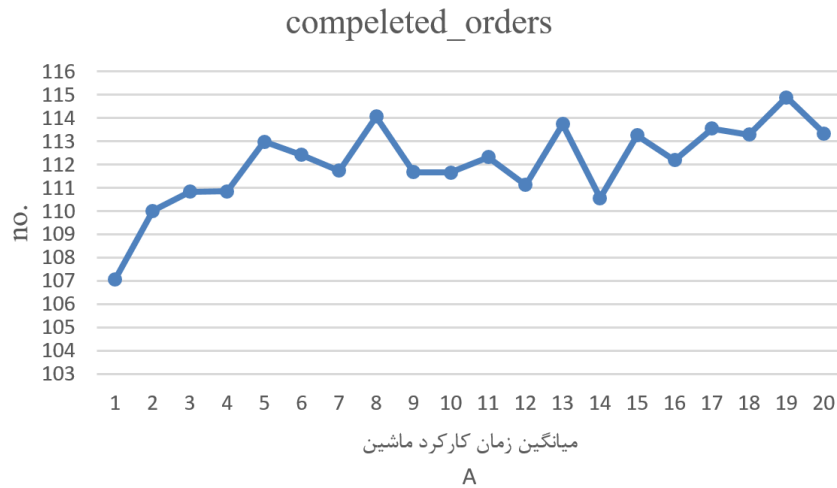
نمودار ۱۴، متوسط حضور در سیستم برای سفارشات با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی

۶.۵.۲. تحلیل حساسیت زمان کارکردن ماشین A



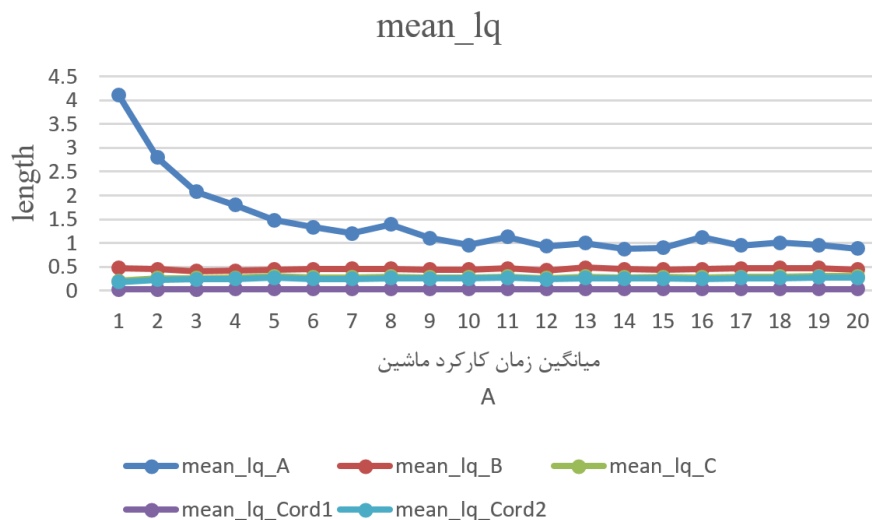
نمودار ۱۵، متوسط بهره‌وری ماشین‌ها با تغییر میانگین زمان کارکرد ماشین A پس از صد بار تکرار شبیه‌سازی

با توجه به نمودار بالا به نظر می‌رسد حالت بهینه در نقطه ۰.۵ حالت فعلی رخ می‌دهد و حداقل تا ۲ برابر میانگین کارکرد فعلی بهبود چندانی در بهره‌وری ماشین A رخ نخواهد داد، به همین دلیل می‌توان گفت نقطه فعلی نقطه بسیار خوبی است و نقطه‌ی بهینه به نظر ۰.۵ می‌باشد.



نمودار ۱۶، متوسط سفارشات تکمیل شده با تغییر میانگین زمان کارکرد ماشین نخست پس از صد بار تکرار شبیه سازی

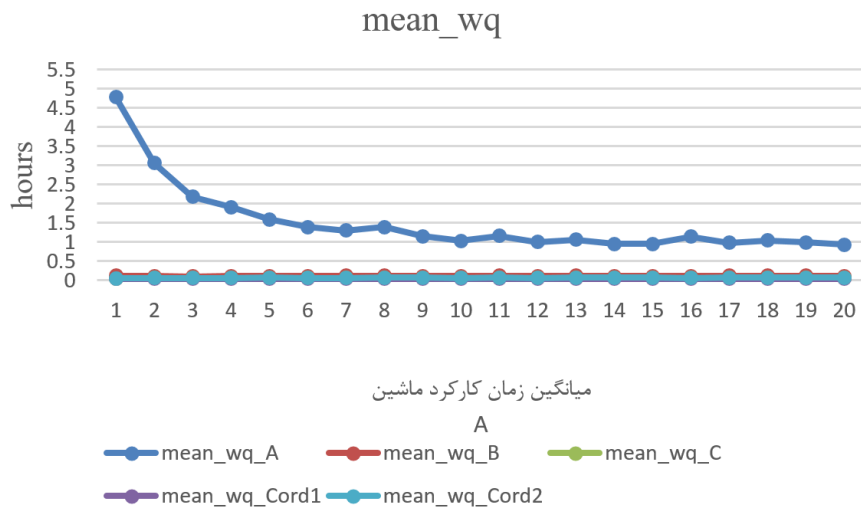
با توجه به نمودار فوق به نظر می رسد از مقدار ۰.۵ مقدار فعلی پارامتر به بعد، تقریباً تغییر مشهودی در تعداد سفارشات تکمیل شده رخ نداده است، عملاً تغییر میانگین مدت کارکرد A تاثیر خاصی در تعداد سفارشات تکمیل شده ندارد.



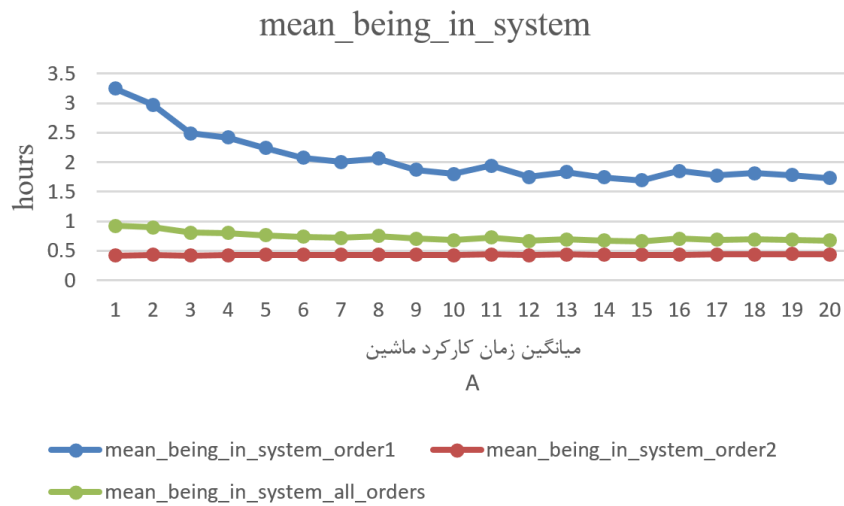
نمودار ۱۷، متوسط طول صف ماشین ها با تغییر میانگین زمان کارکرد ماشین نخست پس از صد بار تکرار شبیه سازی

این نمودار نشان می دهد که تغییر میانگین کارکرد ماشین A تاثیر شدیدی در میانگین طول صف دارد، همچنین از ۰.۹ مقدار فعلی پارامتر به بعد به نظر می رسد تغییر خاصی در میانگین طول صف رخ نداده و برای بهبود در آن نقاط باید پارامترهای دیگر را تغییر داد.

به دلیل همبستگی طول صف و مدت زمان انتظار در صف، نمودار بعد نیز تقریباً مورد انتظار است و نتایج مشابهی در مورد آن صدق می کند.



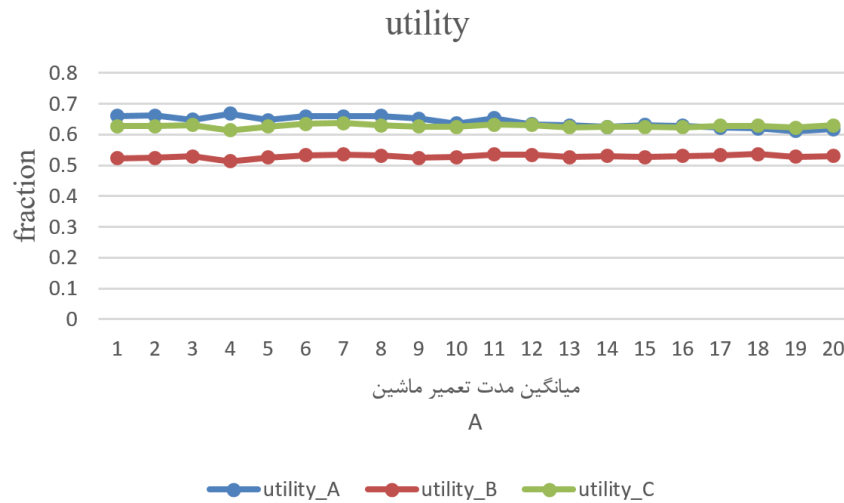
نمودار ۱۸، متوسط زمان انتظار در صفوف ماشین‌ها با تغییر میانگین زمان کارکرد ماشین نخست پس از صد بار تکرار شبیه‌سازی



نمودار ۱۹، متوسط حضور در سیستم برای سفارشات با تغییر میانگین زمان کارکرد ماشین نخست پس از صد بار تکرار شبیه‌سازی

در ابتدا، دلیلی که به این تحلیل حساسیت رو آوردیم اختلاف فاحش مدت زمان حضور سفارشات نوع ۱ نسبت به نوع ۲ بود، با توجه به نمودار بالا به نظر می‌رسد از ۰.۷ مقدار فعلی پارامتر به بعد تغییر معناداری در میانگین مدت حضور در سیستم برای سفارشات نوع ۱ رخ نداده‌است، به همین دلیل به نظر می‌رسد برای بهبود این مشکل، از تغییر این پارامتر نمی‌توانیم کمک چندانی بگیریم و باید به سراغ تحلیل حساسیت‌های بعد برویم.

۶.۵.۳. تحلیل حساسیت زمان تعمیر ماشین A



نمودار ۲۰، متوسط بهره‌وری ماشین‌ها با تغییر میانگین مدت تعمیر ماشین A پس از صد بار تکرار شبیه‌سازی

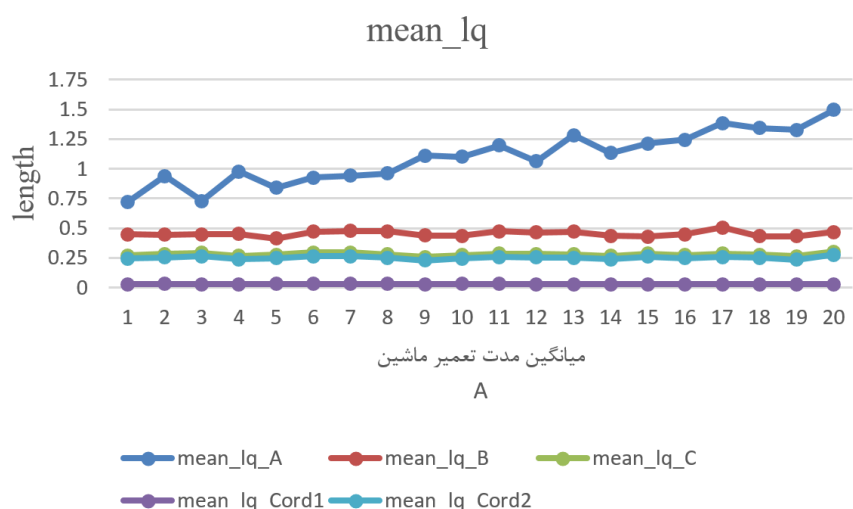
به نظر می‌رسد تغییر میانگین مدت تعمیر ماشین، تاثیر خاصی در بهره‌وری ندارد، به نظر می‌رسد به این دلیل حتی تاثیری در بهره‌وری A ندارد، زیرا بهره‌وری پایین A ناشی از مدت طولانی تعمیر نیست، بلکه ناشی از زمان بین ورود سفارشات نوع ۱ که ۷۰ دقیقه است می‌باشد، به نوعی می‌توان گفت A سفارش کافی در اختیار ندارد تا بهره‌وری‌اش را افزایش دهد.



نمودار ۲۱، متوسط سفارشات تکمیل‌شده با تغییر میانگین مدت تعمیر ماشین نخست پس از صد بار تکرار شبیه‌سازی

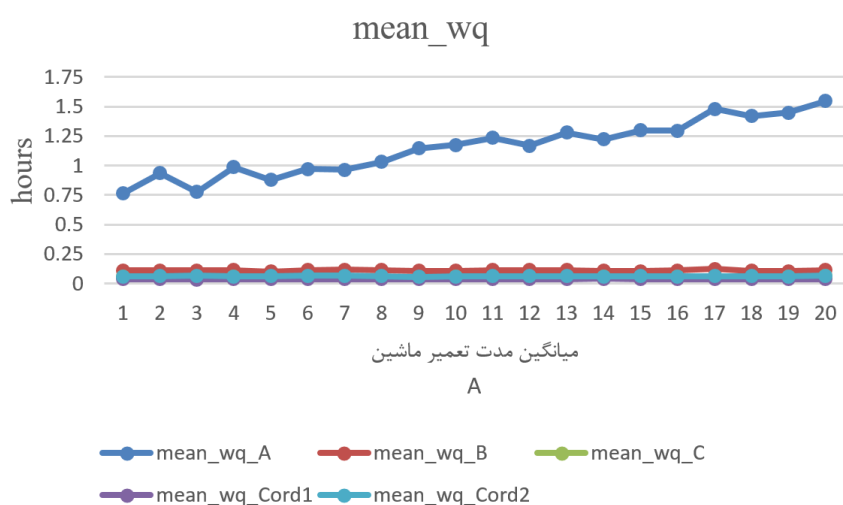
می‌توان گفت رفتار تعداد سفارشات تکمیل شده مستقل از میانگین مدت زمان تعمیر ماشین A است، همانطور که می‌بینید این اعداد از ۱۱۱ تا ۱۱۵ هستند و تغییری در آن‌ها رخ نمی‌دهد، این نتیجه از نمودار قبل قابل برداشت

بود، چرا که نتیجه گرفتیم دلیل کمبود بهره‌وری A نبود سفارش کافی برای خدمت رسانی است، نه میانگین مدت زمان تعمیر.

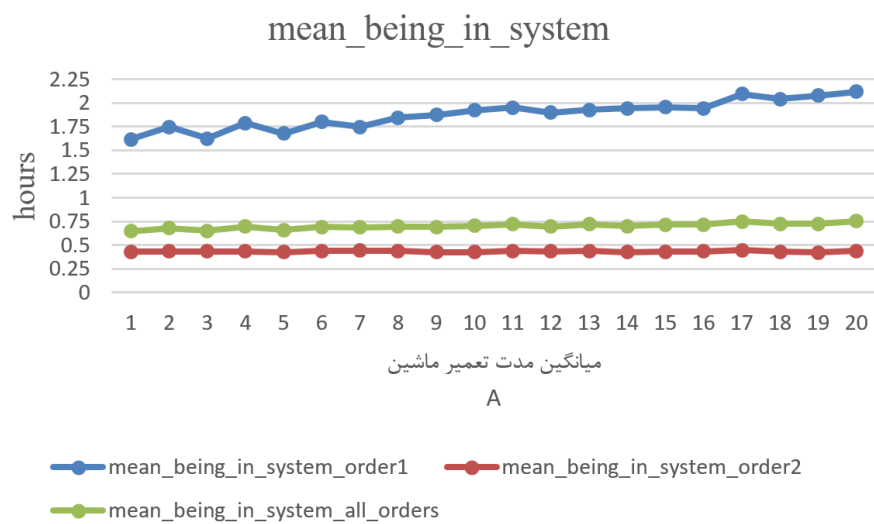


نمودار ۲۲، متوسط طول صف ماشین‌ها با تغییر میانگین مدت تعمیر ماشین نخست پس از صد بار تکرار شبیه‌سازی

به نظر می‌رسد میانگین مدت زمان تعمیر A تاثیر مشهودی در میانگین طول صف A دارد، به نظر می‌رسد در ۰.۱ میانگین مدت تعمیر فعلی، به میزان حدوداً ۳۵ درصد میانگین طول صف کاهش یافته است، همچنین در نمودار بعد تاثیر آن را در میانگین مدت زمان انتظار صف A می‌بینیم که در ۰.۱ حالت فعلی، میانگین مدت انتظار به میزان تقریباً ۳۵ درصد کاهش یافته است؛ بنابراین برای مدیریت به نظر می‌رسد تغییر میانگین تعمیر ماشین A تاثیر زیادی روی میانگین طول صف، میانگین مدت زمان انتظار و طبعاً، میانگین حضور در سیستم سفارشات نوع ۱ خواهد داشت، که در دو نمودار بعد می‌بینید در ۰.۱ مدت زمان تعمیر فعلی، میانگین حضور در سیستم سفارشات نوع ۱، ۲۵ درصد کاهش یافته است.

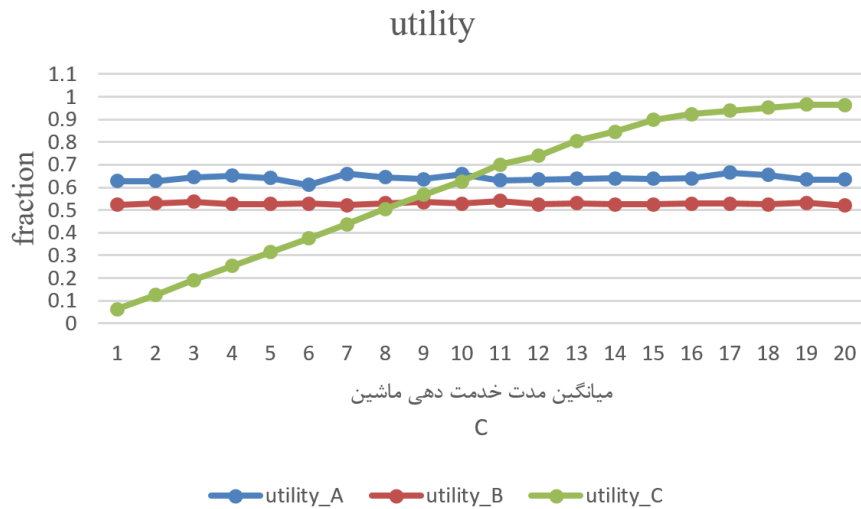


نمودار ۲۳، متوسط زمان انتظار در صفوف ماشین‌ها با تغییر میانگین مدت تعمیر ماشین نخست پس از صد بار تکرار شبیه‌سازی



نمودار ۲۴، متوسط حضور در سیستم برای سفارشات با تغییر میانگین مدت تعمیر ماشین نخست پس از صد بار تکرار شبیه‌سازی

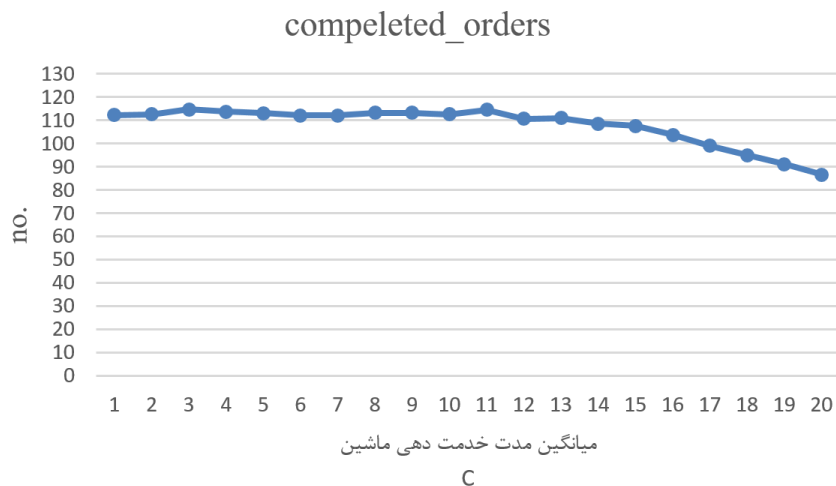
۶.۵.۴. تحلیل حساسیت زمان خدمت دهی ماشین C



نمودار ۲۵، متوسط بهره‌وری ماشین‌ها با تغییر میانگین مدت خدمت دهی ماشین C پس از صد بار تکرار شبیه‌سازی

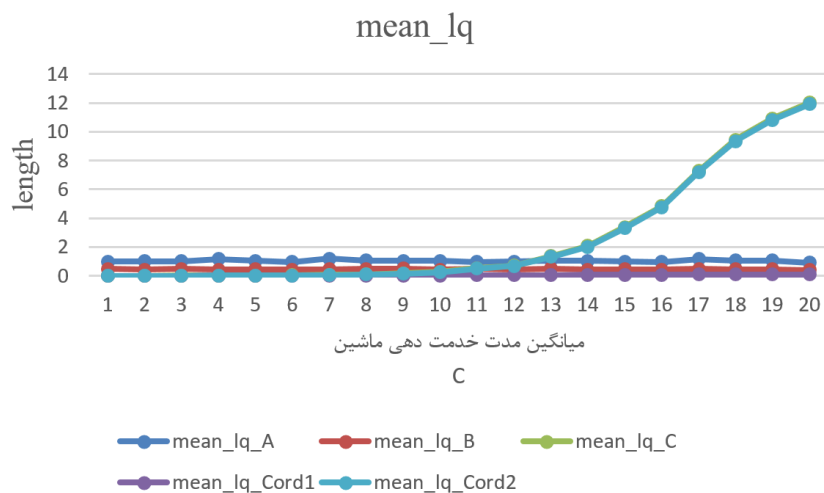
همانطور که مشاهده می‌شود با افزایش میانگین مدت زمان خدمت دهی ماشین C بهره‌وری آن نیز افزایش می‌ابد، این موضوع مهر تاییدی بر درک ساخته شده‌مان از رفتار سیستم دارد: C در حالت فعلی آنقدر پتانسیل دارد که با بهره‌وری پایین نزدیک به نیم، باز هم، همه سفارشات را پاسخ می‌دهد، بنابراین از این درک نتیجه می‌شود که می‌توانیم نرخ ورود سفارشات نوع ۱ و ۲ را افزایش دهیم و از پتانسیل استفاده نشده C استفاده کنیم، یا اینکه در حالت فعلی میانگین مدت خدمت دهی C را افزایش دهیم که می‌تواند در عمل استفاده از یک ماشین C قدیمی‌تر، کندتر و ارزان‌تر باشد.

البته در نظر داشته باشید که این راه حل فقط با توجه به این نمودار گفته شده، در حالت کلی همه نمودارها و تحلیل حساسیت‌ها و همچنین بقیه موارد محدودیت و هزینه‌ها نیز باید در نظر گرفته شوند.



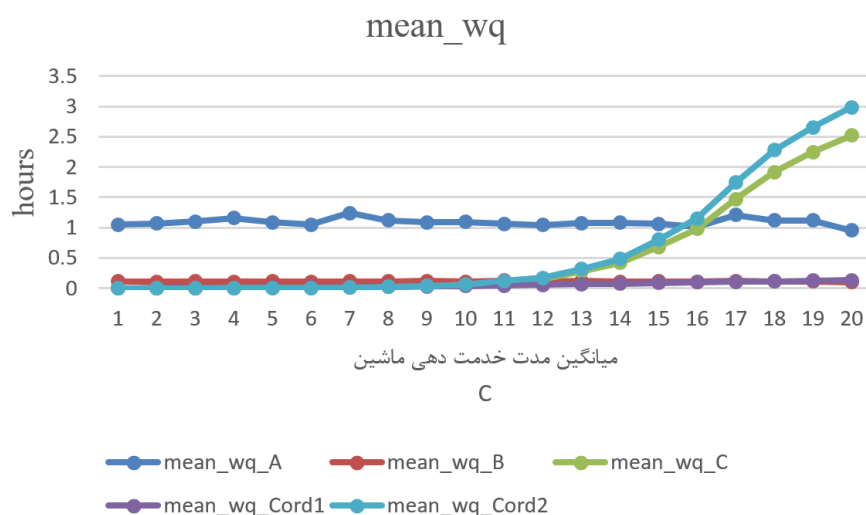
نمودار ۲۶، متوسط سفارشات تکمیل شده با تغییر میانگین مدت خدمت دهی ماشین سوم پس از صد بار تکرار شبیه سازی

همانطور که مشاهده میکنید، افزایش میانگین مدت خدمت دهی C از ۰.۱ مقدار فعلی تا ۱.۴ مقدار فعلی تقریباً تاثیری در تعداد سفارش های تکمیل شده ندارد، به نظر می رسد بازهم افزایش این پارامتر تا ۱.۴ مقدار فعلی گزینه مناسبی باشد.

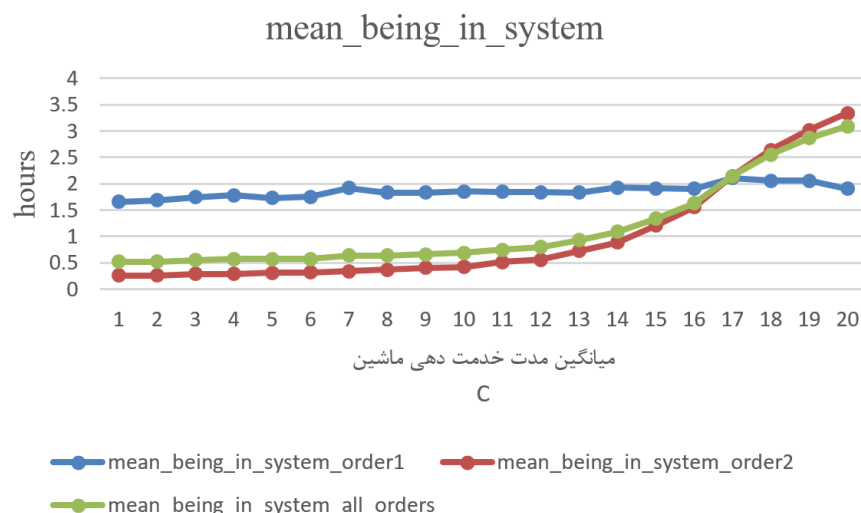


نمودار ۲۷، متوسط طول صف ماشین ها با تغییر میانگین مدت خدمت دهی ماشین سوم پس از صد بار تکرار شبیه سازی

همانطور که مشاهده می شود، ماشین B آنقدر سفارش ترخیص میکند که در صورتی که میانگین مدت خدمت دهی ماشین C از ۱.۳ مقدار حالت فعلی بالاتر برود میانگین طول صف سفارشات نوع ۲ در صف C افزایش خواهد یافت، بنابراین برای کاهش هزینه ها بازهم افزایش این پارامتر تا ۱.۳ مقدار حالت فعلی معقولانه است.



نمودار ۲۸، متوسط زمان انتظار در صفوف ماشین ها با تغییر میانگین مدت خدمت دهی ماشین سوم پس از صد بار تکرار شبیه سازی



نمودار ۲۹، متوسط حضور در سیستم برای سفارشات با تغییر میانگین مدت خدمت دهی ماشین سوم پس از صد بار تکرار شبیه‌سازی

در این دو نمودار نیز، رفتار مشابهی دیده می‌شود، با افزایش این پارامتر، به دلیل انبوه سفارشات نوع ۲ ترخیص شده از B، میانگین زمان انتظار در صف نوع ۲ ماشین C و میانگین حضور در سیستم برای سفارشات نوع ۲ از ۱.۲ مقدار حالت فعلی به بعد، مقادیر مربوطه شروع به تغییر زیاد میکنند. در نتیجه همه تحلیل‌ها، حالت فعلی نسبتاً حالت معقولانه‌ای است اما به نظر می‌رسد ماشین C تا ۱.۲ مقدار فعلی یا ۱.۳ مقدار حالت فعلی می‌تواند در خدمت دهی کندتر بشود بدون اینکه به هیچ یک از خروجی‌های مورد نظر آسیب بزند.

قابل توجه است که به دلیل کمبود سفارشات ترخیص شده از A، تغییر این پارامتر در مدت زمان حضور سفارش‌های نوع ۱ در سیستم تاثیری ندارد.

نتیجه اول:

دیدیم که با وجود همه نکات خوبی که ماشین B نسبت به A داشت بهره‌وری پایین‌تری نسبت به A داشت و طبق نمودارها متوجه شدیم که کاهش میانگین زمان بین ورود سفارشات نوع ۲، بهره‌وری را افزایش خواهد داد، تا اینکه نقطه ۰.۵ برابر حالت فعلی برای این پارامتر به بهره‌وری تقریباً ۱ می‌رسیم، این تغییر موجب افزایش بهره‌وری B و C خواهد بود؛ همچنین متوجه شدیم که کاهش کمتر از ۰.۵ برابر مقدار فعلی پارامتر نیز غیر منطقی است، زیرا علاوه بر اینکه بهره‌وری را تغییر چندانی دیگر نمی‌دهد، باعث افزایش میانگین طول صف، افزایش میانگین انتظار در صف و افزایش میانگین حضور در سیستم خواهد شد.

نتیجه دوم:

برای کاهش متوسط حضور در سیستم سفارشات نوع ۱، موثرترین کار کاهش میانگین مدت تعمیر ماشین A است که حدوداً ۲۵ درصد زمان حضور در سیستم سفارشات نوع ۱ را کاهش خواهد داد، به طور کاربردی مثلاً تعمیرکاران بهتری استخدام کنیم، دستگاه‌های تعمیر پیشرفته‌تری بخریم یا واحد نگهداری و تعمیرات کارگاه را بهبود ببخشیم.

۷. سیاست‌های جدید^{۵۴}

۷.۱. معرفی یک سیاست جدید

فرض کنید که مدیریت این کارگاه می‌خواهد عملکرد سیستم را در طول ۳ روز با دو تیم جدید تعمیرکار نسبت به سیستم قبل مقایسه کند. یکی از تیم‌های جدید، کندتر و دیگری سریعتر از تیم فعلی هستند. توزیع‌های زمان تعمیرات برای تیم‌ها به صورت زیر است:

جدول ۳، مشخصات تیم‌های تعمیرات

ردیف	تیم	تعمیر ماشین A	تعمیر ماشین B
۱	فعلی	یکنواخت بین ۱ دقیقه تا ۲۹ دقیقه	یکنواخت بین ۲ دقیقه تا ۱۸ دقیقه
۲	کندتر	یکنواخت بین ۳ دقیقه تا ۳۱ دقیقه	یکنواخت بین ۶ دقیقه تا ۲۰ دقیقه
۳	سریع‌تر	یکنواخت بین ۱ دقیقه تا ۱۵ دقیقه	یکنواخت بین ۲ دقیقه تا ۱۴ دقیقه

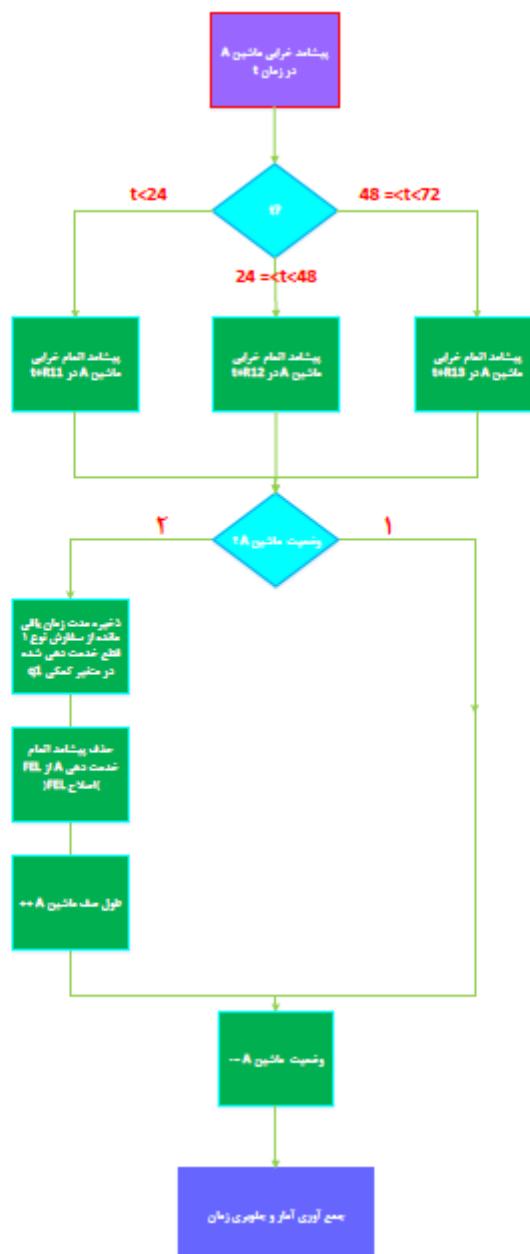
به دلیل هزینه‌ی بالای تیم تندتر، در سیاست جدید مدیریت می‌خواهد روز اول و سوم را به تیم کندتر و روز دوم را به تیم تندتر اختصاص دهد. در این مقایسه به دلیل هزینه‌هایی که سیاست جدید در پی خواهد داشت، اگر این سیاست نتواند بهبودی بیش از ۱۵ عدد در تعداد کل سفارشات آماده شده در طی سه روز به وجود آورد، مدیریت تمایلی به اجرای آن نخواهد داشت.

فرض می‌کنیم اگر زمان تعویض تیم تعمیرکار فرا برسد (ساعت ۲۴ یا ساعت ۴۸) ولی این تیم در حال تعمیر یک ماشین باشد باید کار را تمام کند سپس جایگزین شود ولی اگر در همین مدت ماشین دیگری نیز خراب شود تیم جدید به آن رسیدگی خواهد کرد.

با این توضیحات، در سیاست جدید، مدل مفهومی کمی تغییر می‌کند، به این صورت که توصیف پویا مربوط به پیشامدهای مربوط به خرابی ماشین A و B به صورت زیر تغییر می‌کند:

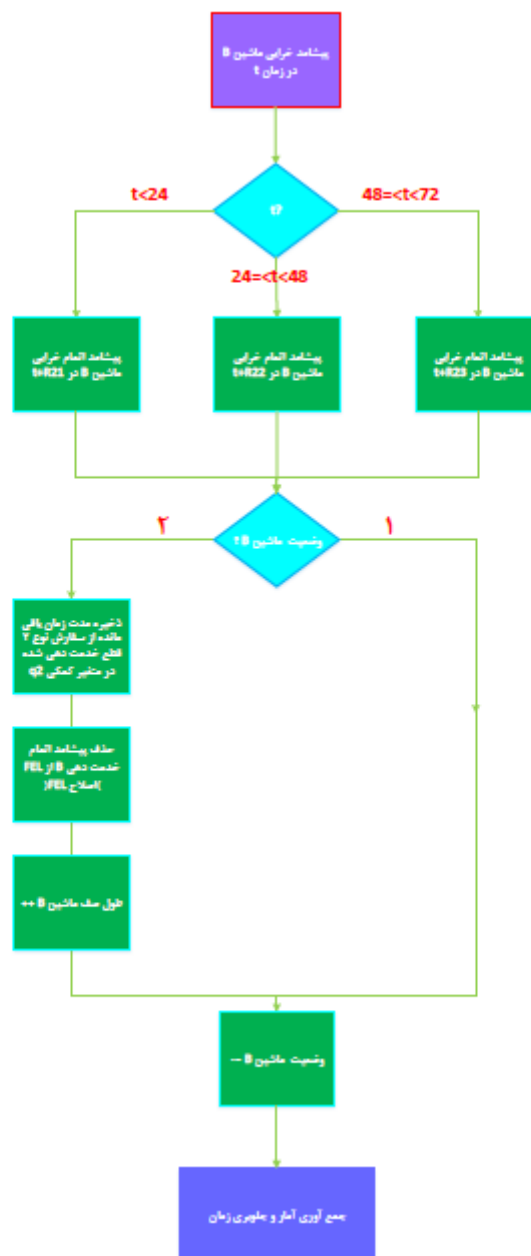
پیشامد خرابی ماشین A در زمان t:

⁵⁴ New Policies



نمودار ۳۰، پیشامد خرابی A در سیاست پیشنهادی

پیشامد خرابی ماشین B در زمان t:



نمودار ۳۱، پیشامد خرابی B در سیاست پیشنهادی

نکته:

جدول ۴، متغیرهای تصادفی مربوط به تعمیر ماشین، سیاست جدید

	روز اول	روز دوم	روز سوم
ماشین A	R11	R12	R13
ماشین B	R21	R22	R23

۷.۲. مقایسه آماری سیاست جدید و سیاست فعلی از منظر تعداد سفارش تکمیل شده

برای مقایسه این دو سیاست دو روش داریم:

۷.۲.۱. روش نمونه گیری مستقل^{۵۵}:

در این روش، آزمایش شبیه سازی را به مدت ۷۲ ساعت، برای سیاست فعلی و سیاست جدید، به تعداد ۱۰۰ بار به صورت مستقل اجرا می‌کنیم؛ برای اطمینان از استقلال باید اعداد تصادفی تولید شده مستقل باشند، در شبیه سازی‌های بزرگ باید هسته تولیدکننده‌ی عدد تصادفی را متفاوت بدهیم تا ورودی‌های متفاوتی تولید شود و خروجی‌های به دست آمده از آزمایش شبیه سازی، مستقل شود. در اینجا چون تعداد اعداد تولیدی زیاد نیست، پس می‌توانیم اطمینان کنیم که تابع تولید عدد تصادفی پایتون، اعداد تصادفی مستقلی برایمان تولید خواهد کرد (دوره مولد عدد تصادفی پایتون به قدری زیاد است که در این مسئله به چالش نمی‌خوریم و شرط استقلال برایمان، فراهم خواهد شد).

⁵⁵ Independent Sampling

خروجی‌های به دست آمده از آزمایش شبیه سازی:

جدول ۵، نتایج آزمایش شبیه سازی در ۷۲ ساعت و ۱۰۰ بار تکرار، برای سیاست فعلی و جدید

انحراف معیار ۱۰۰ خروجی سفارشات تکمیل شده	برآوردگر نقطه‌ای سفارشات تکمیل شده ^{۵۶}	
۱۹.۹۴۱۰	۳۴۶.۵۱	سیاست فعلی
۱۵.۳۶۶۶	۳۴۱.۹۲	سیاست جدید

برای آزمودن این دو سیاست، خطای نوع ۱ را ۰.۰۵ در نظر می‌گیریم و باید حداقل ۶ بار نمونه‌گیری مستقل در آزمایش شبیه سازی انجام شود. فرضیات آزمون به شرح زیر است:

فرض صفر: این دو سیاست در تعداد سفارشات تکمیل شده، تفاوت معناداری ندارند.

فرض یک: این دو سیاست در تعداد سفارشات تکمیل شده، تفاوت معناداری دارند.

همچنین آزمونی که انجام خواهیم داد، با استفاده از توزیع تی برای مقایسه دو سیستم، در حالتی که واریانس‌ها مجهول هستند و در مورد برابری آن‌ها اطلاعی نداریم.

بازه اطمینان^{۵۷}:

$$\bar{Y}_1 - \bar{Y}_2 \pm t_{\alpha/2, v} * s.e. (\bar{Y}_1 - \bar{Y}_2)$$

$$v = \frac{\left(\frac{S_1^2}{R_1} + \frac{S_2^2}{R_2}\right)^2}{\frac{\left(\frac{S_1^2}{R_1}\right)^2}{R_1 - 1} + \frac{\left(\frac{S_2^2}{R_2}\right)^2}{R_2 - 1}}$$

$$s.e. (\bar{Y}_1 - \bar{Y}_2) = \sqrt{\frac{S_1^2}{R_1} + \frac{S_2^2}{R_2}}$$

نکته: عدد یک مربوط به سیاست فعلی و عدد دو مربوط به سیاست جدید است.

^{۵۶} Point Estimators

^{۵۷} Confidence Interval

نتیجه‌ی محاسبات را در جدول زیر مشاهده می‌کنید:

جدول ۶، محاسبات مربوط به مقایسه سیاست فعلی و جدید از منظر سفارشات تکمیل شده

R_1	R_2	S_1^2	S_2^2	v	$t_{\frac{\alpha}{2}, v}$	$s.e. (\bar{Y}_1 - \bar{Y}_2)$	$\bar{Y}_1 - \bar{Y}_2$	بازه‌ی اطمینان	نیم بازه‌ی اطمینان	
۱۰۰		۳۹۷.۶۴۳		۱۸۶	۱.۹۷۳	۲.۵۱۷	۴.۵۹	-۰.۳۷۷، +۹.۵۵۷	۴.۹۶۷	سیاست فعلی
	۱۰۰		۲۳۶.۱۳۲							سیاست جدید

همانطور که مشاهده می‌شود، نیم بازه اطمینان کم‌تر از سطح معنی داری عملیاتی یعنی ۱۵ است، بنابراین از نظر عملیاتی و کاربردی^{۵۸} برای تصمیم‌گیران، این دو سیاست، تفاوت معناداری از نظر کاربرد برای مدیریت، ندارند. از نظر آماری، بازه‌ی اطمینان صفر را در بر می‌گیرد، بنابراین نمی‌توانیم فرض صفر را رد کنیم و دلیل آماری محکمی^{۵۹} برای تفاوت دو سیاست نداریم؛ یعنی از نظر آماری، اختلافی معنادار بین دو سیاست وجود ندارد.

۷.۲.۲. روش اعداد تصادفی مشترک^{۶۰}

معرفی کلی روش:

روش پیشنهادی ما، استفاده از اعداد تصادفی مشترک می‌باشد، این روش اگر به درستی استفاده شود، دو فایده خواهد داشت:

- بازه اطمینان کوچک‌تر و پراکندگی کم‌تری را ارائه می‌دهد که دقت را افزایش داده و هزینه انجام آزمایش‌های شبیه سازی برای رسیدن به دقت مورد نظر را کاهش می‌دهد.
- به دلیل استفاده از اعداد تصادفی مشترک، احتمال زیاد، به تعداد کمتری عدد تصادفی تولیدی نیاز دارد.

⁵⁸ practically significant

⁵⁹ Strong Statistical Evidence

⁶⁰ Common Random Numbers (CRN)

در روش اعداد تصادفی مشترک (CRN)، باید دقیقاً رشته اعداد تصادفی ای که به منظور خاصی در سیستم اول به کار رفته است را به همان منظور در سیستم دوم استفاده کنیم، همچنین در این روش باید تعداد تکرار شبیه سازی برای هر سیستم، برابر با دیگری باشد.

همگام سازی^{۶۱}:

باید برای هر منظور در سیستم اول، یک رشته از اعداد تصادفی مستقل تولید کنیم، برای اینکه این رشته‌ها هم‌پوشانی نداشته باشند، یا از مولدهای اعداد تصادفی متفاوت استفاده کنیم یا اینکه از هسته‌هایی با طول بازه‌ی بسیار بلند استفاده کنیم، همچنین از این اعداد تصادفی، با روش تبدیل تجمعی معکوس^{۶۲}، متغیرهای تصادفی لازم در حین شبیه سازی تولید می‌شود تا همگام سازی تضمین شود. در این پروژه، رشته‌های تولیدی باید برای اهداف زیر استفاده شود:

- زمان بین ورود سفارشات نوع ۱
- زمان بین ورود سفارشات نوع ۲
- مدت زمان خدمت دهی ماشین A
- مدت زمان خدمت دهی ماشین B
- مدت زمان خدمت دهی ماشین C
- مدت زمان کارکرد ماشین A
- مدت زمان کارکرد ماشین B
- مدت زمان تعمیر ماشین A
- مدت زمان تعمیر ماشین B

روش اعداد تصادفی مشترک چه زمانی کار میکند؟

زمانی که همبستگی و کورلیشن مثبت میان مقادیر خروجی سیستم اول و مقادیر خروجی سیستم دوم در هر بار دوباره سازی^{۶۳} وجود داشته باشد، روش CRN در این شرایط باعث کاهش واریانس می‌شود و می‌توانیم از آن استفاده کنیم، اما در صورتی که همبستگی منفی باشد، این روش باعث افزایش واریانس و در نتیجه افزایش طول بازه اطمینان می‌شود و بهتر است استفاده نشود.

در نتیجه در این روش، وقتی که آزمایش شبیه سازی را اجرا کردیم و خروجی‌های مورد نظر را به دست آوردیم، حتماً باید همبستگی بین خروجی‌ها را بررسی کنیم، اگر همبستگی مثبت شد می‌توانیم از این روش استفاده کنیم، در صورتی که همبستگی منفی شد از این روش استفاده نمی‌کنیم و از همان روش دوباره سازی‌های مستقل بهره می‌بریم.

⁶¹ Synchronization

⁶² Inverse-Transform Method

⁶³ Replication

در روش اعداد تصادفی مشترک، برای مقایسه خروجی ها و عملکرد دو سیستم، ابتدا اختلاف مقادیر خروجی دو سیستم در هر بار دوباره سازی را طبق رابطه زیر محاسبه میکنیم :

$$D_r = Y_{r1} - Y_{r2}$$

سپس مقادیر برآورد گر نقطه ای، واریانس نمونه^{۶۴}، خطای استاندارد^{۶۵}، درجه آزادی^{۶۶} و بازه اطمینان را طبق روابط زیر محاسبه کرده و از آزمون t زوجی استفاده میکنیم :

$$\bar{D} = \frac{1}{R} \sum_{r=1}^R D_r$$

$$S_D^2 = \frac{1}{R-1} \sum_{r=1}^R (D_r - \bar{D})^2$$

$$s.e.(\bar{D}) = \frac{S_D}{\sqrt{R}}$$

$$v = R - 1$$

$$\bar{D} \pm s.e.(\bar{D}) \times t_{\alpha/2, v}$$

نحوه‌ی اجرا و تغییرات لازم در کد

می‌دانیم تعداد سفارش تکمیل شده در سیاست فعلی، کمتر از ۴۰۰ بود، همچنین در سیاست جدید نیز دانستیم کمتر از ۴۰۰ سفارش تکمیلی در ۷۲ ساعت به دست می‌آمد، بنابراین از آنجا که طول بازه‌ی مولد عدد تصادفی^{۶۷} پایتون؛ یعنی تابع رندوم زیاد است، برای هر منظور از منظورهای گفته شده با استفاده از یک هسته‌ی متفاوت، رشته‌ای از اعداد تصادفی به طول ۱۰۰۰ تولید میکنیم و این‌ها را در آرایه‌هایی ذخیره می‌کنیم، دقت کنید که ما در هر بار آزمایش، ۹ رشته از اعداد تصادفی مستقل می‌خواهیم و چون ۱۰۰ بار آزمایش‌ها را اجرا می‌کنیم، پس ۹۰۰ رشته از اعداد تصادفی با هسته‌های مختلف لازم داریم. سپس مانند گذشته، مقادیر تصادفی مورد نظر را با استفاد از روش تبدیل تجمعی معکوس و بنابر پارامترهای هر توزیع، تولید می‌کنیم.

حال، در تابع fel_maker هرگاه که می‌خواهیم یک پیشامد جدید در آینده تولید کنیم، به جای تولید لحظه‌ای از رشته‌هایی که در آرایه‌ها ذخیره کرده بودیم استفاده می‌نماییم، پس از استفاده از یک عدد در هر رشته نیز، آن عدد را از آن رشته پاک می‌کنیم تا دیگر از آن استفاده نشود.

خروجی‌های مربوطه در طول هر شبیه سازی را؛ مثلاً مدت زمان انتظار در صف A، در آرایه‌های متفاوتی ذخیره می‌کنیم (در کد، ۱۷ خروجی متفاوت داشتیم، بنابراین در هر تکرار شبیه سازی برای هر سیستم، ۱۷ آرایه متفاوت تشکیل می‌شود)، سپس، در هر بار تکرار، پس از محاسبه همبستگی میان خروجی‌ها و اطمینان از مثبت بودن آن، با استفاده از آماره معرفی شده و در یک سطح معناداری مشخص به محاسبه برآوردگر نقطه‌ای و فاصله‌ای می‌پردازیم.

⁶⁴ Sample Variance

⁶⁵ Standard Error

⁶⁶ Degrees of freedom

⁶⁷ Random Numbers Generator (RNG)

۷.۲.۳. بررسی میانگین مدت زمان انتظار در صف ماشین A برای سیاست فعلی

۷.۲.۳.۱. برآوردگر نقطه‌ای و فاصله‌ای

جدول ۷، محاسبات در ۱۰۰ بار دوباره سازی با آلفای ۰.۰۵

خروجی	برآوردگر نقطه‌ای	نیم بازه اطمینان	فاصله‌ی اطمینان	انحراف استاندارد	انحراف معیار	$t_{\frac{\alpha}{2}, 99}$
میانگین مدت انتظار در صف ماشین A	۱.۴۴۰	۰.۱۴۴۶	۱.۲۹۵۴، ۱.۵۸۴۶	۰.۰۷۲۹	۰.۷۲۹۰	۱.۹۸۴۲

۷.۲.۳.۲. تعداد دوباره سازی لازم برای نصف کردن نیم بازه

روابط و فرمول‌های مربوط

حداقل اندازه نیم‌بازه‌ی مطلوب:

$$\varepsilon = \frac{0.1446}{2} = 0.0723$$

انحراف معیار ناشی از ۱۰۰ تکرار فعلی:

$$S_0 = 0.729$$

Z نیز نشان دهنده‌ی توزیع نرمال استاندارد است.

خطای نوع یک یا آلفا نیز، ۰.۰۵ در نظر می‌گیریم.

در نهایت، نیم بازه را نیز با H نمایش می‌دهیم.

$$R \geq \left(\frac{S_0 * Z_{\frac{\alpha}{2}}}{\varepsilon} \right)^2 = 390.56$$

بنابراین حداقل تعداد دوباره سازی تا به نیم بازه‌ی مطلوب برسیم ۳۹۱ است؛ از آنجا که تا کنون ۱۰۰ دوباره سازی انجام داده‌ایم، ۲۹۱ دوباره سازی دیگر نیاز خواهیم داشت.

۸. ارائه‌ی چند سیاست پیشنهادی برای بهبود سیستم

۸.۱. شهود نسبت به سیستم

با توجه به قسمت تحلیل حساسیت برای بهینه سازی و تصمیم گیری، به چند نکته در مورد رفتار سیستم رسیدیم:

- افزایش میانگین مدت زمان خدمت دهی ماشین C تا ۱.۳ حالت فعلی (ارزان تر کردن استفاده از ماشین C) تاثیری در خروجی‌ها نمی‌گذارد؛ یعنی ماشین C فعلی پتانسیلی بیشتر از حالت فعلی سیستم دارد و افزایش مدت زمان خدمت دهی آن تا ۱.۳، صرفه جویی خوبی به حساب می‌آید.
- کاهش میانگین مدت کارکرد ماشین A تا نصف حالت فعلی، تغییر مشهودی در خروجی‌ها ندارد، بنابراین کاهش آن می‌تواند یکی از گزینه‌های صرفه جویی باشد.
- تغییر مدت کارکرد ماشین A و مدت تعمیر ماشین A تغییر مشهودی در تعداد سفارشات تکمیل شده نخواهد داشت، دلیل این امر این است که مشکل اساسی سفارشات نوع ۱، کمبود آنهاست، بنابراین در حالت فعلی کاهش زمان بین ورود سفارشات نوع ۱، به نظر، تاثیر به سزایی در سفارشات تکمیل شده خواهد گذاشت.
- مهر تایید دیگری که بر کم بودن تعداد سفارشات ورودی نوع ۱ داریم، بهره‌وری نسبتاً پایین ماشین A می‌باشد.

- برخلاف سفارشات نوع ۱، سفارشات نوع ۲ با سرعتی زیاد وارد سیستم می‌شوند و با سرعت نیز توسط ماشین B خدمت دهی می‌شوند؛ با این حال ماشین B بهره‌وری نسبتاً پایینی دارد، بنابراین به نظر می‌رسد با کاهش زمان بین ورود سفارشات نوع ۲ تا نصف مقدار فعلی، تاثیر به سزایی در تعداد سفارشات تکمیل شده خواهیم داشت، برای مثال با توجه به نمودارهای تحلیل حساسیت برای بهینه سازی و تصمیم گیری، در ۲۴ ساعت حدوداً ۱۸۰، که به نظر می‌رسید مقدار بهینه می‌باشد.
- همچنین در ارائه پیشنهادات بهبود، باید هزینه را نیز مد نظر قرار داد، بهبود هر یک از بخش‌های سیستم، یا افزایش نرخ ورود سفارشات در یک کارخانه‌ی واقعی نیازمند صرف هزینه‌های ثابت و متغیر می‌باشد.

۸.۲. سیاست‌های پیشنهادی^{۶۸}

سیاست‌های پیشنهادی (اگر برای مدیریت تعداد سفارشات تکمیل شده آن هم با تغییر ۱۵ عدد مهم باشد)

- میانگین مدت زمان بین ورود سفارشات نوع ۱ و ۲ را تا ۵۰ درصد حالت فعلی کاهش دهیم و نرخ ورود این سفارشات را افزایش دهیم.
- میانگین مدت زمان بین ورود سفارش نوع ۲ را تا ۵۰ درصد کاهش دهیم، حال برای کاهش آن از ۵۰ درصد به پایین‌تر برای مثال ۱۰ درصد، به طور همزمان، میانگین مدت زمان خدمت دهی ماشین B و C را کاهش داده تا بتوانند به انبوه سفارشات دریافتی برسند.

سیاست‌های پیشنهادی دیگر

- افزایش میانگین مدت زمان خدمت دهی ماشین C تا ۱.۳ حالت فعلی برای صرفه جویی
- کاهش میانگین مدت تعمیر ماشین A تا نصف حالت فعلی برای صرفه جویی
- کاهش مدت تعمیر ماشین A تا ۰.۱ حالت فعلی برای حدوداً ۳۰ درصد کاهش حضور در سیستم سفارشات نوع ۱، البته به هزینه‌های ناشی نیز باید توجه ویژه‌ای بشود.

⁶⁸ Recommended Policies

۹. جداول و نمودارها

۹.۱. جداول

شماره صفحه	نام
۲۵	جدول ۱، خروجی‌های بدست‌آمده از صد بار تکرار شبیه‌سازی در ۲۴ ساعت با آلفای ۰.۰۵
۲۸	جدول ۲، صحت سنجی خروجی‌های بدست‌آمده از صد بار تکرار شبیه‌سازی در ۲۴ ساعت
۵۰	جدول ۳، مشخصات تیم‌های تعمیرات
۵۳	جدول ۴، متغیرهای تصادفی مربوط به تعمیر ماشین، سیاست جدید
۵۴	جدول ۵، نتایج آزمایش شبیه‌سازی در ۷۲ ساعت و ۱۰۰ بار تکرار، برای سیاست فعلی و جدید
۵۵	جدول ۶، محاسبات مربوط به مقایسه سیاست فعلی و جدید از منظر سفارشات تکمیل شده
۵۸	جدول ۷، محاسبات در ۱۰۰ بار دوباره‌سازی با آلفای ۰.۰۵

۹.۲. نمودارها

شماره صفحه	نام
۱۶	نمودار ۱، پیشامد ورود سفارشات نوع ۱ به ماشین A
۱۷	نمودار ۲، ورود سفارشات نوع ۲ به ماشین B
۱۸	نمودار ۳، پیشامد اتمام خدمت دهی ماشین A
۱۹	نمودار ۴، پیشامد اتمام خدمت دهی ماشین B
۲۰	نمودار ۵، پیشامد خرابی ماشین A
۲۱	نمودار ۶، پیشامد خرابی ماشین B
۲۲	نمودار ۷، پیشامد اتمام خرابی ماشین A
۲۳	نمودار ۸، پیشامد اتمام خرابی ماشین B
۲۴	نمودار ۹، پیشامد اتمام خدمت دهی ماشین C
۳۸	نمودار ۱۰، متوسط بهره‌وری ماشین‌ها با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی
۳۸	نمودار ۱۱، متوسط سفارشات تکمیل‌شده با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی
۳۹	نمودار ۱۲، متوسط طول صف ماشین‌ها با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی

- نمودار ۱۳، متوسط زمان انتظار در صفوف ماشین‌ها با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی
- ۳۹
- نمودار ۱۴، متوسط حضور در سیستم برای سفارشات با تغییر میانگین زمان بین ورود سفارشات نوع ۲ پس از صد بار تکرار شبیه‌سازی
- ۴۰
- نمودار ۱۵، متوسط بهره‌وری ماشین‌ها با تغییر میانگین زمان کارکرد ماشین A پس از صد بار تکرار شبیه‌سازی
- ۴۰
- نمودار ۱۶، متوسط سفارشات تکمیل‌شده با تغییر میانگین زمان کارکرد ماشین نخست پس از صد بار تکرار شبیه‌سازی
- ۴۱
- نمودار ۱۷، متوسط طول صف ماشین‌ها با تغییر میانگین زمان کارکرد ماشین نخست پس از صد بار تکرار شبیه‌سازی
- ۴۱
- نمودار ۱۸، متوسط زمان انتظار در صفوف ماشین‌ها با تغییر میانگین زمان کارکرد ماشین نخست پس از صد بار تکرار شبیه‌سازی
- ۴۲
- نمودار ۱۹، متوسط حضور در سیستم برای سفارشات با تغییر میانگین زمان کارکرد ماشین نخست پس از صد بار تکرار شبیه‌سازی
- ۴۲
- نمودار ۲۰، متوسط بهره‌وری ماشین‌ها با تغییر میانگین مدت تعمیر ماشین A پس از صد بار تکرار شبیه‌سازی
- ۴۳
- نمودار ۲۱، متوسط سفارشات تکمیل‌شده با تغییر میانگین مدت تعمیر ماشین نخست پس از صد بار تکرار شبیه‌سازی
- ۴۴
- نمودار ۲۲، متوسط طول صف ماشین‌ها با تغییر میانگین مدت تعمیر ماشین نخست پس از صد بار تکرار شبیه‌سازی
- ۴۴

- نمودار ۲۳، متوسط زمان انتظار در صفوف ماشین‌ها با تغییر میانگین مدت تعمیر ماشین
 ۴۵ نخست پس از صد بار تکرار شبیه‌سازی
- نمودار ۲۴، متوسط حضور در سیستم برای سفارشات با تغییر میانگین مدت تعمیر
 ۴۵ ماشین نخست پس از صد بار تکرار شبیه‌سازی
- نمودار ۲۵، متوسط بهره‌وری ماشین‌ها با تغییر میانگین مدت خدمت دهی ماشین C
 ۴۶ پس از صد بار تکرار شبیه‌سازی
- نمودار ۲۶، متوسط سفارشات تکمیل‌شده با تغییر میانگین مدت خدمت دهی ماشین
 ۴۷ سوم پس از صد بار تکرار شبیه‌سازی
- نمودار ۲۷، متوسط طول صف ماشین‌ها با تغییر میانگین مدت خدمت دهی ماشین سوم
 ۴۷ پس از صد بار تکرار شبیه‌سازی
- نمودار ۲۸، متوسط زمان انتظار در صفوف ماشین‌ها با تغییر میانگین مدت خدمت دهی
 ۴۸ ماشین سوم پس از صد بار تکرار شبیه‌سازی
- نمودار ۲۹، متوسط حضور در سیستم برای سفارشات با تغییر میانگین مدت خدمت
 ۴۸ دهی ماشین سوم پس از صد بار تکرار شبیه‌سازی
- نمودار ۳۰، پیشامد خرابی A در سیاست پیشنهادی
 ۵۲
- نمودار ۳۱، پیشامد خرابی B در سیاست پیشنهادی
 ۵۳

۱۰. پیوست

۱۰.۱. شرح کد پایتون

در این کد که با زبان پایتون نوشته شده است، سعی شده که از علائم نوشتاری موجود و به سادگی در گزارش استفاده شود، در جای جای کد، با کامنت کردن، سعی شده است توضیحات لازم نوشته شود. برای هر عمل، تابعی در نظر گرفته شده تا کد، مرتبتر و قابل استفاده تر گردد، همچنین در صورت نیاز به تمام کد، این کد به صورت (.py) و (.ipynb) در لینک زیر برای دانلود قرار داده شده که علاقمندان می توانند از آن استفاده نمایند.

[codes](#)

خروجی هایی که در این کد محاسبه شده اند، هر چهار خروجی خواسته شده مدیریت و یکی از خروجی های پیشنهادی ما، یعنی مدت حضور در سیستم بوده است.

۱۰.۲. کد پایتون

''''''''

simulation of a workshop

in this workshop we have 2 kind of orders, order 1 and order 2

interarrival time of order 1 distribution ~ exponential with mean of 7/6 hours

interarrival time of order 2 distribution ~ exponential with mean of 1/4 hours

service time of A machine distribution ~ uniform between 35/60 to 65/60 hours

service time of B machine distribution ~ uniform between 3/60 to 13/60 hours

service time of C machine distribution ~ Triangular of 6/60,8/60,10/60 hours

working time of A machine distribution ~ $100x/60 + 50/60$ that x has weibul distribution with shape of 2 and scale of 1

working time of B machine distribution ~ uniform between 50/60 to 350/60 hours

repairing time of A distribution ~ uniform between 1/60 to 29/60 hours

repairing time of B distribution ~ uniform between 2/60 to 18/60 hours

No limit on Queue's length

People get service in a FIFO system

outputs: 1: completed orders

2: servers(A, B & C)efficiency

3: Queue waiting time mean for A,B & C

4: Time weighted length mean for A, B & C

5: being in the system time mean

starting state: system is empty

authors: Mehrdad Moradi, Behzad Yaghubi

date: 2020-12-3

#modules needed

import numpy as num

import random

import math

from math import sqrt

import xlswriter as xw # Used to create Excel Directly

from numpy import mean

import statsmodels.stats.api as sms

#functions

#starting_state function

def starting_state():

state=dict() #state: defining variable to put state variables

state['server condition']={'A': 1,'B': 1,'C': 0} #state variables for servers

state['queues']={'A': [],

'B': [],


```

'Cord1': [],
'Cord2': []}    # we save the queues as lists of orders

state['helping']={'q1':-1,
                 'q2':-1,
                 'stockorder1':'null',
                 'stockorder2':'null'} #we use these for when A or B get out of service


#data collecting: used to collect datas we need to collect, used in calculating cumulative statistics
data_collecting=dict()

data_collecting['clock']=0 #we use this clock to calculate time weighted length for queues

data_collecting['order1s']=dict()
data_collecting['order2s']=dict() #we save order1s or order2s as dict, keys are the name of order
like order11 and the

#value is the list of times, # this list saves these times in this
#order as a list: entering to system, beginning of service A,beginning of
service B
#begining of service C,end of service A, end of service B, end of service
C,
#last stop of service time,last back to service time


# Cumulative statistics
#cumulative statistics needed
cumulative_stat = {'compeleted orders':0 ,
                  'served by':{'A':0,'B':0,'Cord1':0,'Cord2':0},
                  'server busy time':{'A':0,'B':0,'C':0 } ,

```

```

    'waiting in queue': {'A':0,'B':0,'Cord1':0,'Cord2':0},
    'time weighted length': {'A':0,'B':0,'Cord1':0,'Cord2':0},
    'total being in system': {'order1':0,'order2':0},
    'stock_num': {'A':0,'B':0}
}

```

```

#future-event-list

```

```

future_event_list = list()

```

```

fel_maker(future_event_list, 'arrival1', 0, 'order11')    # Difference: make an entrance of specific
order1 (order11)

```

```

fel_maker(future_event_list, 'arrival2', 0, 'order21')    # Difference: make an entrance of specific
order2 (order21)

```

```

fel_maker(future_event_list, 'out of service A',0,'A')    #at t=0: A start working and we should
generate an out of service event fo A

```

```

fel_maker(future_event_list, 'out of service B',0,'B')    #at t=0: B start working and we should
generate an out of service event fo B

```

```

return state, data_collecting, future_event_list, cumulative_stat

```

#fel_maker: when we need to make a new event, we use this function to make new event(based on activity dist.) and update

```

#future event list

```

```

def fel_maker(future_event_list,event_type,clock,order):

```

#next 4 global variables will be used for sensitivity analysis, at first we should let them be 1, for sensitivity

```

#analysis we change them to change the parameters

```

```

global arrB

```

```
global work_A
```

```
global rep_A
```

```
global serv_C
```

```
rand_num = random.random() #making random number, we use inverse transform method to  
generate random variates
```

```
event_time=0 #defining event_time, we change it later
```

```
if event_type=='arrival1':
```

```
    event_time=clock+round((-7/6)*math.log(rand_num),3)
```

```
if event_type=='arrival2':
```

```
    event_time=clock+round(arr_B*((-1/4)*math.log(rand_num)),3)
```

```
if event_type=='end of service A':
```

```
    event_time=clock+round((35/60)+(30/60)*rand_num,3)
```

```
if event_type=='end of service B':
```

```
    event_time=clock+round((3/60)+(10/60)*rand_num,3)
```

```
if event_type=='end of service C':
```

```
    if rand_num<=(0.5):
```

```
        event_time=clock+round(serv_C*((sqrt(2*rand_num)+3)/30),3)
```

```
    else:
```

```
        event_time=clock+round(serv_C*((5-sqrt(2*(1-rand_num)))/30),3)
```

```
if event_type=='out of service A':
```

```
    event_time=clock+round(work_A*(50/60+(100/60)*math.sqrt(-math.log(rand_num))),3)
```

```
if event_type=='out of service B':
```

```
    event_time=clock+round(50/60+(300/60)*rand_num,3)
```

```
if event_type=='being repaired A':
```

```
    event_time=clock+round(rep_A*(1/60+(28/60)*rand_num),3)
```

```
if event_type=='being repaired B':
```

```
    event_time=clock+round(2/60+(16/60)*rand_num,3)
```

```
new_event = {'event type': event_type,
```

```
            'event time': event_time,
```

```
            'order or server': order} # additional element in event notices (order No.)
```

```
future_event_list.append(new_event)
```

```
#event functions: when we call these functions,its like that special event has happend
    #and every thing that need to change(state,data_collecting and cumulative_stat),
    #will change.its like we translate our dynamic diagram into python codes
```

```
def arrival1(future_event_list, state, data_collecting, clock, order1, cumulative_stat):
```

```
    #first of all, we update time weighted length
```

```
    cumulative_stat['time weighted length']['A']+=(len(state['queues']['A']))*(clock-
data_collecting['clock'])
```

```
    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B']))*(clock-
data_collecting['clock'])
```

```
    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1']))*(clock-
data_collecting['clock'])
```

```
    cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2']))*(clock-
data_collecting['clock'])
```

```
    data_collecting['order1s'][order1]=[0]*9 #dafining the list to save times
```

```
    data_collecting['order1s'][order1][0]=clock#saves the entering to system time for this order
```

```
    #figure next number of order1
```

```
    order1_num = int(order1[6:])
```

```
    order1_num+= 1
```

```
    fel_maker(future_event_list,'arrival1',clock,'order1'+str(order1_num)) #make new arrival for order
```

```
1
```

```
#based on dynamic diagram(dynamic description)
```

```

if state['server condition']['A']==1:
    state['server condition']['A']+=1
    fel_maker(future_event_list,'end of service A',clock,order1)
    data_collecting['order1s'][order1][1]=clock #saving the begining of service A time for this order

```

```

else:
    state['queues']['A'].append(order1)

    #other changes in data_collecting and cumulative_stat

```

```

data_collecting['clock']=clock#updating time

```

```

def arrival2(future_event_list, state, data_collecting, clock, order2, cumulative_stat):
    #first of all, we update time weighted length
    cumulative_stat['time weighted length']['A']+=(len(state['queues']['A'])*(clock-
data_collecting['clock']))
    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B'])*(clock-
data_collecting['clock']))
    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1'])*(clock-
data_collecting['clock']))

```

```
cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2']))*(clock-  
data_collecting['clock'])
```

```
data_collecting['order2s']['order2]=[0]*9 #dafining the list to save times
```

```
data_collecting['order2s']['order2'][0]=clock#saves the entering to system time for this order
```

```
#figure out how to find next number of order2
```

```
order2_num = int(order2[6:])
```

```
order2_num+= 1
```

```
fel_maker(future_event_list,'arrival2',clock,'order2'+str(order2_num)) #make new arrival for  
order 2
```

```
#based on dynamic diagram(dynamic description)
```

```
if state['server condition']['B']==1:
```

```
    state['server condition']['B']+=1
```

```
    fel_maker(future_event_list,'end of service B',clock,order2)
```

```
    data_collecting['order2s']['order2'][2]=clock#saving the begining of service A time for this order
```

```
else:
```

```
    state['queues']['B'].append(order2)
```

```
    #other changes in data_collecting and cumulative_stat
```

```
data_collecting['clock']=clock#updating time
```

```
def end_of_service_A(future_event_list, state, data_collecting, clock, order1, cumulative_stat):
```

```

#first of all, we update time weighted length

cumulative_stat['time weighted length']['A']+=(len(state['queues']['A'])*(clock-
data_collecting['clock']))

cumulative_stat['time weighted length']['B']+=(len(state['queues']['B'])*(clock-
data_collecting['clock']))

cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1'])*(clock-
data_collecting['clock']))

cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2'])*(clock-
data_collecting['clock']))

```

#based on diagram:

#backward look

if len(state['queues']['A'])==0:

#nessesary changes to state, data_collecting,cumulative_stat:

state['server condition']['A']=1

data_collecting['order1s'][order1][4]=clock

cumulative_stat['server busy time']['A']+=data_collecting['order1s'][order1][4]-
data_collecting['order1s'][order1][1]

cumulative_stat['served by']['A']+=1

else:

#nessesary changes to state, data_collecting,cumulative_stat:

#new one enters from the queue to server A

order=state['queues']['A'][0]

state['queues']['A'].pop(0)

fel_maker(future_event_list,'end of service A',clock,order)

data_collecting['order1s'][order][1]=clock


```

    cumulative_stat['waiting in queue']['A']+=clock-data_collecting['order1s'][order][0]
    data_collecting['order1s'][order1][4]=clock
    cumulative_stat['server busy time']['A']+=data_collecting['order1s'][order1][4]-
data_collecting['order1s'][order1][1]
    cumulative_stat['served by']['A']+=1

```

#forward look

```

if state['server condition']['C']==0:

```

```

    #nessesary changes to state, data_collecting,cumulative_stat:

```

```

    #state

```

```

    state['server condition']['C']=1

```

```

    fel_maker(future_event_list,'end of service C',clock,order1)

```

```

    #data_collecting&cumulative_stat

```

```

    data_collecting['order1s'][order1][3]=clock

```

```

    #nothing happen to cumulative_stat

```

```

else:

```

```

    #state

```

```

    state['queues']['Cord1'].append(order1)

```

```

data_collecting['clock']=clock#updating clock of data_collecting to current clock

```

```

def end_of_service_B(future_event_list, state, data_collecting, clock, order2, cumulative_stat):

    #first of all, we update time weighted length

    cumulative_stat['time weighted length']['A']+=(len(state['queues']['A'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2'])*(clock-
data_collecting['clock']))

```

#based on diagram:

#backward look

if len(state['queues']['B'])==0:

#nessesary changes to state, data_collecting,cumulative_stat:

state['server condition']['B']=1

data_collecting['order2s'][order2][5]=clock

cumulative_stat['server busy time']['B']+=data_collecting['order2s'][order2][5]-
data_collecting['order2s'][order2][2]

cumulative_stat['served by']['B']+=1

else:

#nessesary changes to state, data_collecting,cumulative_stat:

#new one enters from the queue to server A

```

order=state['queues']['B'][0]
state['queues']['B'].pop(0)
fel_maker(future_event_list,'end of service B',clock,order)
data_collecting['order2s'][order][2]=clock

cumulative_stat['waiting in queue']['B']+=clock-data_collecting['order2s'][order][0]
data_collecting['order2s'][order2][5]=clock
cumulative_stat['server busy time']['B']+=data_collecting['order2s'][order2][5]-
data_collecting['order2s'][order2][2]
cumulative_stat['served by']['B']+=1

```

#forward look

```
if state['server condition']['C']==0:
```

```
    #nessesary changes to state, data_collecting,cumulative_stat:
```

```
    #state
```

```
    state['server condition']['C']=1
```

```
    fel_maker(future_event_list,'end of service C',clock,order2)
```

```
    #data_collecting&cumulative_stat
```

```
    data_collecting['order2s'][order2][3]=clock
```

```
    #nothing happen to cumulative_stat
```

```
else:
```

```
    #state
```

```
    state['queues']['Cord2'].append(order2)
```

```
data_collecting['clock']=clock#updating clock of data_collecting to current clock
```

```
def end_of_service_C(future_event_list, state, data_collecting, clock, order, cumulative_stat):
```

```
    #first of all, we update time weighted length
```

```
    cumulative_stat['time weighted length']['A']+=(len(state['queues']['A']))*(clock-  
data_collecting['clock'])
```

```
    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B']))*(clock-  
data_collecting['clock'])
```

```
    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1']))*(clock-  
data_collecting['clock'])
```

```
    cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2']))*(clock-  
data_collecting['clock'])
```

```
if len(state['queues']['Cord1'])==0:
```

```
    if len(state['queues']['Cord2'])==0:
```

```
        state['server condition']['C']=0
```

```
else:
```

```
    orderr=state['queues']['Cord2'][0]#note that this is another order entering to C
```

```
    state['queues']['Cord2'].pop(0)
```

```
    fel_maker(future_event_list,'end of service C',clock,orderr)
```

```
    #data_collecting & cumulative_stat
```

```
    data_collecting['order2s'][orderr][3]=clock
```

```

        cumulative_stat['waiting in queue']['Cord2']+=data_collecting['order2s'][orderr][3]-
data_collecting['order2s'][orderr][5]

```

```

else:

```

```

    orderr=state['queues']['Cord1'][0]#note that this is another order entering to C
    state['queues']['Cord1'].pop(0)

```

```

    fel_maker(future_event_list,'end of service C',clock,orderr)

```

```

#data_collecting & cumulative_stat

```

```

data_collecting['order1s'][orderr][3]=clock

```

```

    cumulative_stat['waiting in queue']['Cord1']+=data_collecting['order1s'][orderr][3]-
data_collecting['order1s'][orderr][4]

```

```

#data_collecting & cumulative_stat

```

```

if order[5]=='1':

```

```

    data_collecting['order1s'][order][6]=clock

```

```

    cumulative_stat['total being in system']['order1']+=data_collecting['order1s'][order][6]-
data_collecting['order1s'][order][0]

```

```

    cumulative_stat['server busy time']['C']+=data_collecting['order1s'][order][6]-
data_collecting['order1s'][order][3]

```

```

    cumulative_stat['served by']['Cord1']+=1

```

```

else:

```

```

    data_collecting['order2s'][order][6]=clock

```

```

    cumulative_stat['total being in system']['order2']+=data_collecting['order2s'][order][6]-
data_collecting['order2s'][order][0]

```

```

    cumulative_stat['server busy time']['C']+=data_collecting['order2s'][order][6]-
data_collecting['order2s'][order][3]

```

```

    cumulative_stat['served by']['Cord2']+=1

```

```
data_collecting['clock']=clock#updating clock of data_collecting to current clock
```

```
cumulative_stat['compeleted orders']+=1
```

```
def out_of_service_A(future_event_list, state, data_collecting, clock, cumulative_stat):
```

```
    #first of all, we update time weighted length
```

```
    cumulative_stat['time weighted length']['A']+=(len(state['queues']['A'])*(clock-  
data_collecting['clock']))
```

```
    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B'])*(clock-  
data_collecting['clock']))
```

```
    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1'])*(clock-  
data_collecting['clock']))
```

```
    cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2'])*(clock-  
data_collecting['clock']))
```

```
#based on diagram
```

```
if state['server condition']['A']==2:
```

```
    #now we want to find end of service A element for the order A was working on
```

```
    index=next((i for i, item in enumerate(future_event_list) if item['event type']=='end of service  
A'), None)
```

```
    #witch order was that A was working on?
```

```
    state['helping']['stockorder1']=future_event_list[index]['order or server']
```

```
    state['helping']['q1']=future_event_list[index]['event time']-clock
```

```
    #removing that event from FEL
```

```

future_event_list.pop(index)
#adding new order in queue
state['queues']['A'].insert(0,state['helping']['stockorder1'])
#data_collecting
data_collecting['order1s'][state['helping']['stockorder1']][7]=clock
#maybe need some cumulative_stat, check it later!
cumulative_stat['stock_num']['A']+=1

```

```

state['server condition']['A']=0
fel_maker(future_event_list,'being repaired A',clock,'A')
    #other changes in data_collecting and cumulative_stat

```

```

data_collecting['clock']=clock#updating time

```

```

def out_of_service_B(future_event_list, state, data_collecting, clock, cumulative_stat):
    #first of all, we update time weighted length
    cumulative_stat['time weighted length']['A']+=(len(state['queues']['A'])*(clock-
data_collecting['clock']))
    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B'])*(clock-
data_collecting['clock']))
    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1'])*(clock-
data_collecting['clock']))
    cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2'])*(clock-
data_collecting['clock']))

```

#based on diagram

if state['server condition']['B']==2:

 #now we want to find end of service A element for the order B was working on

 index=next((i for i, item in enumerate(future_event_list) if item['event type']=='end of service B'), None)

 #which order was that A was working on?

 state['helping']['stockorder2']=future_event_list[index]['order or server']

 state['helping']['q2']=future_event_list[index]['event time']-clock

 #removing that event from FEL

 future_event_list.pop(index)

 #adding new order in queue

 state['queues']['B'].insert(0,state['helping']['stockorder2'])

 #data_collecting

 data_collecting['order2s'][state['helping']['stockorder2']][7]=clock

 #maybe need some sumulative_stat, check it later!

 cumulative_stat['stock_num']['B']+=1

state['server condition']['B']=0

fel_maker(future_event_list,'being repaired B',clock,'B')

 #other changes in data_collecting and cumulative_stat

data_collecting['clock']=clock#updating time

def being_repaired_A(future_event_list, state, data_collecting, clock, cumulative_stat):

 #first of all, we update time weighted length

 cumulative_stat['time weighted length']['A']+=(len(state['queues']['A'])*(clock-data_collecting['clock']))


```

    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B']))*(clock-
data_collecting['clock'])

    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1']))*(clock-
data_collecting['clock'])

    cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2']))*(clock-
data_collecting['clock'])

```

```

    #making next out of service event

```

```

    fel_maker(future_event_list,'out of service A',clock,'A')

```

```

if state['helping']['q1']==-1:

```

```

    if len(state['queues']['A'])==0:

```

```

        state['server condition']['A']=1

```

```

    else:

```

```

        state['server condition']['A']=2

```

```

        order=state['queues']['A'][0]

```

```

        state['queues']['A'].pop(0)

```

```

        fel_maker(future_event_list,'end of service A',clock,order)

```

```

        #data_collecting & cumulative_stat

```

```

        data_collecting['order1s'][order][1]=clock

```

```

        cumulative_stat['waiting in queue']['A']+=data_collecting['order1s'][order][1]-
data_collecting['order1s'][order][0]

```

```

    else:

```

```

future_event_list.append({'event type': 'end of service A','event time':
clock+state['helping']['q1'],'order or server': state['helping']['stockorder1']})

state['server condition']['A']=2

#remove that stock order from A queue

state['queues']['A'].pop(0)

#data_collecting & cumulative_stat

data_collecting['order1s'][state['helping']['stockorder1']][8]=clock

cumulative_stat['server busy time']['A']-
=(data_collecting['order1s'][state['helping']['stockorder1']][8]-
data_collecting['order1s'][state['helping']['stockorder1']][7])

cumulative_stat['waiting in
queue']['A']+=(data_collecting['order1s'][state['helping']['stockorder1']][8]-
data_collecting['order1s'][state['helping']['stockorder1']][7])

data_collecting['clock']=clock#updating clock of data_collecting to current clock

#we don't have any atock order 1 anymore!

state['helping']['stockorder1']='null'

state['helping']['q1']=-1

```

```

def being_repaired_B(future_event_list, state, data_collecting, clock, cumulative_stat):

    #first of all, we update time weighted length

    cumulative_stat['time weighted length']['A']+=(len(state['queues']['A'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2'])*(clock-
data_collecting['clock']))

```

```

    #making next out of service event
    fel_maker(future_event_list,'out of service B',clock,'B')

if state['helping']['q2']==-1:
    if len(state['queues']['B'])==0:
        state['server condition']['B']=1

    else:
        state['server condition']['B']=2
        order=state['queues']['B'][0]
        state['queues']['B'].pop(0)
        fel_maker(future_event_list,'end of service B',clock,order)
        #data_collecting & cumulative_stat
        data_collecting['order2s'][order][2]=clock
        cumulative_stat['waiting in queue']['B']+=data_collecting['order2s'][order][2]-
data_collecting['order2s'][order][0]

else:

    future_event_list.append({'event type': 'end of service B',
    'event time': clock+state['helping']['q2'],'order or server': state['helping']['stockorder2']})
    state['server condition']['B']=2
    #remove that stock order from A queue

```

```

state['queues']['B'].pop(0)

#data_collecting & cumulative_stat

data_collecting['order2s'][state['helping']['stockorder2']][8]=clock

cumulative_stat['server busy time']['B']-
=data_collecting['order2s'][state['helping']['stockorder2']][8]-
data_collecting['order2s'][state['helping']['stockorder2']][7]

cumulative_stat['waiting in
queue']['B']+=data_collecting['order2s'][state['helping']['stockorder2']][8]-
data_collecting['order2s'][state['helping']['stockorder2']][7]


data_collecting['clock']=clock #updating clock of data_collecting to current clock
#we don't have any atock order 1 anymore!
state['helping']['stockorder2']='null'
state['helping']['q2']=-1

```

#next tow functions will be used to write in trace excel

```

def end_of_simulation(state, data_collecting, clock, cumulative_stat):

    #first of all, we update time weighted length

    cumulative_stat['time weighted length']['A']+=(len(state['queues']['A'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['B']+=(len(state['queues']['B'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['Cord1']+=(len(state['queues']['Cord1'])*(clock-
data_collecting['clock']))

    cumulative_stat['time weighted length']['Cord2']+=(len(state['queues']['Cord2'])*(clock-
data_collecting['clock']))

```

```

#server busy time in cumulative stat
if(state['server condition']['A']==2):#A
    cumulative_stat['server busy time']['A']+=clock-data_collecting['clock']
if(state['server condition']['B']==2):#B
    cumulative_stat['server busy time']['B']+=clock-data_collecting['clock']
if(state['server condition']['C']==1):#C
    cumulative_stat['server busy time']['C']+=clock-data_collecting['clock']

#waiting in queue in cumulative stat
#A
if(len(state['queues']['A'])!=0):
    for i in range(len(state['queues']['A'])):
        cumulative_stat['waiting in queue']['A']+=data_collecting['clock']-
data_collecting['order1s'][state['queues']['A'][i]][0]
#B
if(len(state['queues']['B'])!=0):
    for j in range(len(state['queues']['B'])):
        cumulative_stat['waiting in queue']['B']+=data_collecting['clock']-
data_collecting['order2s'][state['queues']['B'][j]][0]

#Cord1
#in addition, we want to calculate total being in system too

if(len(state['queues']['Cord1'])!=0):
    for i in range(len(state['queues']['Cord1'])):
        cumulative_stat['waiting in queue']['Cord1']+=data_collecting['clock']-
data_collecting['order1s'][state['queues']['Cord1'][i]][4]
        cumulative_stat['total being in system']['order1']+=data_collecting['clock']-
data_collecting['order1s'][state['queues']['Cord1'][i]][0]

#Cord2

```

#in addition, we want to calculate total being in system too

```
if(len(state['queues']['Cord2'])!=0):
```

```
    for i in range(len(state['queues']['Cord2'])):
```

```
        cumulative_stat['waiting in queue']['Cord2']+=data_collecting['clock']-  
data_collecting['order2s'][state['queues']['Cord2'][i]][5]
```

```
        cumulative_stat['total being in system']['order2']+=data_collecting['clock']-  
data_collecting['order2s'][state['queues']['Cord2'][i]][0]
```

```
#update clock
```

```
data_collecting['clock']=clock
```

```
def output_excel(worksheet, future_event_list, state,cumulative_stat,row_num):
```

```
    global max_fel
```

```
    global header_list    # we update the header list in this function
```

```
    future_event_list1 = sorted(future_event_list, key=lambda x: x['event time'])
```

```
    new_row = [row_num, future_event_list1[0]['event type'], future_event_list1[0]['event time'],
```

```
               state['server condition']['A'],state['server condition']['B'],state['server condition']['C'],
```

```
               len(state['queues']['A']), len(state['queues']['B']), len(state['queues']['Cord1']),
```

```
               len(state['queues']['Cord2']),state['helping']['stockorder1'],
```

```
               state['helping']['stockorder2'],state['helping']['q1'],state['helping']['q2'],
```

```
               cumulative_stat['compeleted orders'],cumulative_stat['served by']['A'],
```

```
               cumulative_stat['served by']['B'],cumulative_stat['served by']['Cord1'],
```

```
               cumulative_stat['served by']['Cord2'],cumulative_stat['server busy time']['A'],
```

```
               cumulative_stat['server busy time']['B'],cumulative_stat['server busy time']['C'],
```

```
               cumulative_stat['waiting in queue']['A'],cumulative_stat['waiting in queue']['B'],
```

```
               cumulative_stat['waiting in queue']['Cord1'],cumulative_stat['waiting in queue']['Cord2'],
```

```
               cumulative_stat['time weighted length']['A'],cumulative_stat['time weighted length']['B'],
```

```

        cumulative_stat['time weighted length']['Cord1'],cumulative_stat['time weighted
length']['Cord2'],

        cumulative_stat['total being in system']['order1'],cumulative_stat['total being in
system']['order2'],

        cumulative_stat['stock_num']['A'],cumulative_stat['stock_num']['B']]      # Creating
new row

```

```

# Update the header list and max_fel
if len(future_event_list1)-1 > max_fel:
    for fel_counter in range(max_fel, len(future_event_list1) - 1):
        header_list.extend(("Future Event Type "+str(fel_counter+1), "Future Event Time
"+str(fel_counter+1)))
        max_fel = len(future_event_list1) - 1

else:
    for add_number in range(max_fel - (len(future_event_list1) - 1)):
        future_event_list1.append({"event type": "", "event time": ""})

for evno in future_event_list1[1:]:
    new_row.extend((evno['event type'], evno['event time']))

for col in range(len(header_list)):
    worksheet.write(0, col, header_list[col])
    worksheet.write(row_num, col, new_row[col])

```

```

#setting the format of excel as we want

```

```

def excel_formatting(workbook, worksheet, row_num):

    cell_format_header = workbook.add_format()
    cell_format_header.set_align('center')
    cell_format_header.set_align('vcenter')
    cell_format_header.set_font('Times New Roman')
    cell_format_header.set_bg_color('#e2ac8d') # Find the color in Excel use RGB to HEX online
tool
    cell_format_header.set_bold(True)

    worksheet.set_row(0, None, cell_format_header)

    worksheet.set_column(0, 0, 20)
    worksheet.set_column(1, 1, 13)
    worksheet.set_column(2, 2, 9)
    worksheet.set_column(3, 5, 8)
    worksheet.set_column(6, 9, 6)
    worksheet.set_column(10, 9 + 2*max_fel, 19)#set the size of cells
    cell_format = workbook.add_format()
    cell_format.set_align('center')
    cell_format.set_font('Times New Roman')
    for row in range(row_num):
        worksheet.set_row(row+1, None, cell_format)

```

#simulation

```

def simulation(simulation_time):

```



```

#next 4 global variables will be used in sensitivity analysis

global arrB
global work_A
global rep_A
global serv_C

global sen_analysis#this global list will save the outputs for each parameter value in sensitivity
analysis

utility_A=[0]*100
utility_B=[0]*100
utility_C=[0]*100
mean_lq_A=[0]*100
mean_lq_B=[0]*100
mean_lq_Cord1=[0]*100
mean_lq_Cord2=[0]*100      #we run the simulation for 100 times and save the outputs in
these lists

mean_lq_C=[0]*100
mean_wq_A=[0]*100
mean_wq_B=[0]*100
mean_wq_C=[0]*100
mean_wq_Cord1=[0]*100
mean_wq_Cord2=[0]*100
mean_being_in_system_order1=[0]*100
mean_being_in_system_order2=[0]*100
mean_being_in_system_all_orders=[0]*100
compeleted_orders=[0]*100

for i in range(100):

```

#note when we want to do sensitivity analysis we comment ** codes, because we dont want to rewrite trace excel

```
workbook = xw.Workbook("C:/Users/System-Pc/Desktop/handy/sut/٧ پروژہ/فاز /simulation/ ٧/Output_XW.xlsx") ***
```

Open a New Workbook (Simultaneously Created an Excel) #note that if you want run this in your computer you should

#change the address

```
# worksheet = workbook.add_worksheet("production workshop simulation") # Add a Sheet  
***
```

```
# row_num = 1 # Define row number ***
```

```
state, data_collecting, future_event_list, cumulative_stat = starting_state() # Starting state  
clock = 0
```

Add specific events

```
future_event_list.append({'event type': 'end of simulation', 'event time': simulation_time, 'order  
or server': 'null'})
```

Continue till the simulation time ends

```
while clock <= simulation_time:
```

```
sorted_fel = sorted(future_event_list, key=lambda x: x['event time']) # Sort the FEL based  
on event times
```

```
current_event = sorted_fel[0] # The first element is what happening now
```

```
clock = current_event['event time'] # Move the time forward
```

```
if clock <= simulation_time:
```

```
current_order=current_event['order or server']
```

```
if current_event['event type'] == 'arrival1':
```

```

        arrival1(future_event_list, state, data_collecting, clock, current_order, cumulative_stat)
    if current_event['event type'] == 'arrival2':
        arrival2(future_event_list, state, data_collecting, clock, current_order, cumulative_stat)
    if current_event['event type'] == 'end of service A':
        end_of_service_A(future_event_list, state, data_collecting, clock, current_order,
cumulative_stat)
    if current_event['event type'] == 'end of service B':
        end_of_service_B(future_event_list, state, data_collecting, clock, current_order,
cumulative_stat)
    if current_event['event type'] == 'end of service C':
        end_of_service_C(future_event_list, state, data_collecting, clock, current_order,
cumulative_stat)
    if current_event['event type'] == 'out of service A':
        out_of_service_A(future_event_list, state, data_collecting, clock, cumulative_stat)
    if current_event['event type'] == 'out of service B':
        out_of_service_B(future_event_list, state, data_collecting, clock, cumulative_stat)
    if current_event['event type'] == 'being repaired A':
        being_repaired_A(future_event_list, state, data_collecting, clock, cumulative_stat)
    if current_event['event type'] == 'being repaired B':
        being_repaired_B(future_event_list, state, data_collecting, clock, cumulative_stat)
    if current_event['event type'] == 'end of simulation':
        end_of_simulation(state, data_collecting, clock, cumulative_stat)

#        output_excel(worksheet, future_event_list, state,cumulative_stat,row_num)  # Use the
xlsxwriter Module ***
#        row_num += 1        # Preparing row number for the next time ***

    #adjust future_event_list
    future_event_list.remove(current_event)
else:
    future_event_list.clear()#clearing future event list after simulation ends

```

```
# excel_formatting(workbook, worksheet, row_num)      # Format the Output_XW Excel #**
# workbook.close()          # Close the workbook #**
```

```
# Calculating the outputs
```

```
#desired by management
```

```
compeleted_orders[i]=cumulative_stat['compeleted orders']
```

```
utility_A[i]=cumulative_stat['server busy time']['A']/simulation_time
```

```
utility_B[i]=cumulative_stat['server busy time']['B']/simulation_time
```

```
utility_C[i]=cumulative_stat['server busy time']['C']/simulation_time
```

```
mean_lq_A[i]=cumulative_stat['time weighted length']['A']/simulation_time
```

```
mean_lq_B[i]=cumulative_stat['time weighted length']['B']/simulation_time
```

```
mean_lq_Cord1[i]=cumulative_stat['time weighted length']['Cord1']/simulation_time
```

```
mean_lq_Cord2[i]=cumulative_stat['time weighted length']['Cord2']/simulation_time
```

```
mean_lq_C[i]=mean_lq_Cord1[i]+mean_lq_Cord2[i]
```

```
mean_wq_A[i]=cumulative_stat['waiting in queue']['A']/len(data_collecting['order1s'])
```

```
mean_wq_B[i]=cumulative_stat['waiting in queue']['B']/len(data_collecting['order2s'])
```

```
mean_wq_C[i]=(cumulative_stat['waiting in queue']['Cord1']+cumulative_stat['waiting in
queue']['Cord2'])/(cumulative_stat['served by']['A']+
```

```
cumulative_stat['served by']['B'])
```

```
mean_wq_Cord1[i]=cumulative_stat['waiting in queue']['Cord1']/(cumulative_stat['served
by']['A'])
```

```
mean_wq_Cord2[i]=cumulative_stat['waiting in queue']['Cord2']/(cumulative_stat['served
by']['B'])
```

```

#suggested by our team

mean_being_in_system_order1[i]=cumulative_stat['total being in
system']['order1']/len(data_collecting['order1s'])

mean_being_in_system_order2[i]=cumulative_stat['total being in
system']['order2']/len(data_collecting['order2s'])

mean_being_in_system_all_orders[i]=((cumulative_stat['total being in
system']['order1']+cumulative_stat['total being in
system']['order2'])/(len(data_collecting['order1s'])+len(data_collecting['order2s'])))

# print all outputs

# print('outputs desired by management:')

# print('compeleted orders  =' +str(mean(compeleted_orders))+'\nutility A
=' +str(mean(utility_A))

#  +'\nutility B  =' +str(mean(utility_B))+'\nutility C  =' +str(mean(utility_C))

#  +'\nmean lq A  =' +str(mean(mean_lq_A))+'\nmean lq B  =' +str(mean(mean_lq_B))+'\nmean lq
C  =' +str(mean(mean_lq_C))

#  +'\nmean lq Cord1  =' +str(mean(mean_lq_Cord1))+'\nmean lq Cord2
=' +str(mean(mean_lq_Cord2))

#  +'\nmean wq A  =' +str(mean(mean_wq_A))+'\nmean wq B
=' +str(mean(mean_wq_B))+'\nmean wq C  =' +str(mean(mean_wq_C))

#  +'\nmean wq Cord1  =' +str(mean(mean_wq_Cord1))+'\nmean wq Cord2
=' +str(mean(mean_wq_Cord2)))

# print('suggested by our team:')

# print('mean being in system for order1  =' +str(mean(mean_being_in_system_order1)))

# print('mean being in system for order2  =' +str( mean(mean_being_in_system_order2)))

# print('mean being in system for all orders  =' +str( mean(mean_being_in_system_all_orders)))

# print('end!')

# print('CI:')

```

```

# print('compeleted orders '+str(sms.DescrStatsW(compeleted_orders).tconfint_mean()))
# print('utility_A'+str(sms.DescrStatsW(utility_A).tconfint_mean()))
# print('utility_B'+str(sms.DescrStatsW(utility_B).tconfint_mean()))
# print('utility_C'+str(sms.DescrStatsW(utility_C).tconfint_mean()))
# print('mean lq A'+str(sms.DescrStatsW(mean_lq_A).tconfint_mean()))
# print('mean lq B'+str(sms.DescrStatsW(mean_lq_B).tconfint_mean()))
# print('mean lq C'+str(sms.DescrStatsW(mean_lq_C).tconfint_mean()))
# print('mean lq Cord1'+str(sms.DescrStatsW(mean_lq_Cord1).tconfint_mean()))
# print('mean lq Cord2'+str(sms.DescrStatsW(mean_lq_Cord2).tconfint_mean()))
# print('mean wq A'+str(sms.DescrStatsW(mean_wq_A).tconfint_mean()))
# print('mean wq B'+str(sms.DescrStatsW(mean_wq_B).tconfint_mean()))
# print('mean wq C'+str(sms.DescrStatsW(mean_wq_C).tconfint_mean()))
# print('mean wq Cord1'+str(sms.DescrStatsW(mean_wq_Cord1).tconfint_mean()))
# print('mean wq Cord2'+str(sms.DescrStatsW(mean_wq_Cord2).tconfint_mean()))
# print('mean being in system for order1
'+str(sms.DescrStatsW(mean_being_in_system_order1).tconfint_mean()))
# print('mean being in system for order2
'+str(sms.DescrStatsW(mean_being_in_system_order2).tconfint_mean()))
# print('mean being in system for all
orders'+str(sms.DescrStatsW(mean_being_in_system_all_orders).tconfint_mean()))

```

#sensitivity analysis

```

sen_analysis[0]=['compeleted_orders',mean(compeleted_orders)]
sen_analysis[1]=['utility_A',mean(utility_A)]
sen_analysis[2]=['utility_B',mean(utility_B)]
sen_analysis[3]=['utility_C',mean(utility_C)]
sen_analysis[4]=['mean_lq_A',mean(mean_lq_A)]
sen_analysis[5]=['mean_lq_B',mean(mean_lq_B)]

```

```

sen_analysis[6]='mean_lq_C',mean(mean_lq_C)]
sen_analysis[7]='mean_lq_Cord1',mean(mean_lq_Cord1)]
sen_analysis[8]='mean_lq_Cord2',mean(mean_lq_Cord2)]
sen_analysis[9]='mean_wq_A',mean(mean_wq_A)]
sen_analysis[10]='mean_wq_B',mean(mean_wq_B)]
sen_analysis[11]='mean_wq_C',mean(mean_wq_C)]
sen_analysis[12]='mean_wq_Cord1',mean(mean_wq_Cord1)]
sen_analysis[13]='mean_wq_Cord2',mean(mean_wq_Cord2)]
sen_analysis[14]='mean_being_in_system_order1',mean(mean_being_in_system_order1)]
sen_analysis[15]='mean_being_in_system_order2',mean(mean_being_in_system_order2)]
sen_analysis[16]='mean_being_in_system_all_orders',mean(mean_being_in_system_all_orders)]

```

```

#main_code

```

```

max_fel = 0      # Maximum length that FEL get (the current event does not count in)

header_list = ['Step', 'Current Event', 'Clock', 'Server A','Server B','Server C', 'Queue A','Queue
B','Queue Cord1','Queue Cord2','Stock Order1','Stock Order2','q1','q2',

               'Compeleted orders','Served by A','Served by B','Served by Cord1','Served by Cord2','Server
busy time A','Server busy time B',

               'Server busy time C','waiting in queue A','waiting in queue B','waiting in queue
Cord1','waiting in queue Cord2',

               'time weighted length A','time weighted length B','time weighted length Cord1','time
weighted length Cord2',

               'total being in system order1','total being in system order2', 'stock_num A', 'stock_num B']
# Headers at First

```

```

arr_B=1
work_A=1
rep_A=1
serv_C=1 #global variables for sensitivity analysis

```

```
simulation(int(input("enter the simulation time:"))))
```

Sensitivity analysis

#sensitivity analysis on interarrival time for order2

#note that before sensitivity analysis we should comment all this section

```
arr_B=0.1#seting the target parameter to 0.1 and increasing that in loop
```

```
work_A=1
```

```
rep_A=1
```

```
serv_C=1
```

```
sen_analysis=[[0,0]]*17 #defining 2D list
```

```
workbook = xw.Workbook("C:/Users/System-Pc/Desktop/handy/sut/۷ پروژہ/فاز /simulation/ترم ۷/Sensitivity analysis_XW.xlsx")
```

Open a New Workbook (Simultaneously Created an Excel) #note that if you want run this in your computer you should

#change the address

```
worksheet = workbook.add_worksheet(" arr_B analysis") # Add a Sheet
```

```
for i in range(20):
```

```
    arr_B=(i*0.1)+0.1
```

```
    simulation(24)
```

```
    for j in range(17):
```

```
        worksheet.write(0,0,'step')
```

```
        worksheet.write(i+1,0,i+1)
```

```
        worksheet.write(0,j+1,sen_analysis[j][0])
```

```
        worksheet.write(i+1,j+1,sen_analysis[j][1]) #writing the values
```

```
    excel_formatting(workbook, worksheet,i+1) # Format the Sensitivity analysis Excel sheet
```

arr_B

#sensitivity analysis on working time of A


```

arr_B=1
work_A=0.1 #seting the target parameter to 0.1 and increasing that in loop
rep_A=1
serv_C=1
sen_analysis=[[0,0]]*17 #defining 2D list

worksheet = workbook.add_worksheet("work_A analysis") # Add a Sheet

for i in range(20):
    work_A=(i*0.1)+0.1
    simulation(24)
    for j in range(17):
        worksheet.write(0,0,'step')
        worksheet.write(i+1,0,i+1)
        worksheet.write(0,j+1,sen_analysis[j][0])
        worksheet.write(i+1,j+1,sen_analysis[j][1]) #writing the values
        excel_formatting(workbook, worksheet,i+1) # Format the Sensitivity analysis Excel sheet
work_A
#sensitivity analysis on repairing time of A
arr_B=1
work_A=1
rep_A=0.1 #seting the target parameter to 0.1 and increasing that in loop
serv_C=1
sen_analysis=[[0,0]]*17 #defining 2D list

worksheet = workbook.add_worksheet("rep_A analysis") # Add a Sheet

for i in range(20):
    rep_A=(i*0.1)+0.1
    simulation(24)
    for j in range(17):

```

```

worksheet.write(0,0,'step')
worksheet.write(i+1,0,i+1)
worksheet.write(0,j+1,sen_analysis[j][0])
worksheet.write(i+1,j+1,sen_analysis[j][1])
    excel_formatting(workbook, worksheet,i+1)    # Format the Sensitivity analysis Excel sheet
rep_A
#sensitivity analysis on service time of C
arr_B=1
work_A=1
rep_A=1
serv_C=0.1 #seting the target parameter to 0.1 and increasing that in loop
sen_analysis=[[0,0]]*17 #defining 2D list

worksheet = workbook.add_worksheet("serv_C analysis") # Add a Sheet

for i in range(20):
    serv_C=(i*0.1)+0.1
    simulation(24)
    for j in range(17):
        worksheet.write(0,0,'step')
        worksheet.write(i+1,0,i+1)
        worksheet.write(0,j+1,sen_analysis[j][0])
        worksheet.write(i+1,j+1,sen_analysis[j][1])
        excel_formatting(workbook, worksheet,i+1)    # Format the Sensitivity analysis Excel sheet
serv_C
workbook.close()    # Close the workbook

```

Step	Current Event	Clock	Server A	Server B	Server C	Queue A	Queue B	Queue Cord1	Queue Cord2	Stock Order1	Stock Order2	q1	q2
1	arrival2	0.115	1	2	0	0	0	0	0	null	null	-1	-1
2	end of service B	0.255	1	1	1	0	0	0	0	null	null	-1	-1
3	arrival2	0.26	1	2	1	0	0	0	0	null	null	-1	-1
4	end of service C	0.414	1	2	0	0	0	0	0	null	null	-1	-1
5	end of service B	0.467	1	1	1	0	0	0	0	null	null	-1	-1
6	end of service C	0.608	1	1	0	0	0	0	0	null	null	-1	-1
7	arrival2	0.744	1	2	0	0	0	0	0	null	null	-1	-1
8	end of service B	0.816	1	1	1	0	0	0	0	null	null	-1	-1
9	arrival2	0.904	1	2	1	0	0	0	0	null	null	-1	-1
10	arrival2	0.907	1	2	1	0	1	0	0	null	null	-1	-1
11	end of service C	0.953	1	2	0	0	1	0	0	null	null	-1	-1
12	end of service B	0.969	1	2	1	0	0	0	0	null	null	-1	-1
13	end of service C	1.118	1	2	0	0	0	0	0	null	null	-1	-1
14	end of service B	1.178	1	1	1	0	0	0	0	null	null	-1	-1
15	end of service C	1.307	1	1	0	0	0	0	0	null	null	-1	-1
16	arrival1	1.378	2	1	0	0	0	0	0	null	null	-1	-1
17	arrival2	1.601	2	2	0	0	0	0	0	null	null	-1	-1
18	end of service B	1.657	2	1	1	0	0	0	0	null	null	-1	-1
19	arrival2	1.717	2	2	1	0	0	0	0	null	null	-1	-1
20	end of service C	1.794	2	2	0	0	0	0	0	null	null	-1	-1
21	end of service B	1.878	2	1	1	0	0	0	0	null	null	-1	-1
22	end of service A	1.973	1	1	1	0	0	1	0	null	null	-1	-1
23	end of service C	2.007	1	1	1	0	0	0	0	null	null	-1	-1
24	end of service C	2.139	1	1	0	0	0	0	0	null	null	-1	-1
25	arrival2	2.188	1	2	0	0	0	0	0	null	null	-1	-1

Step	Completed orders	Served by A	Served by B	Served by Cord1	Served by Cord2	Server busy time A	Server busy time B	Server busy time C
1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0.14	0
3	0	0	1	0	0	0	0.14	0
4	1	0	1	0	1	0	0.14	0.159
5	1	0	2	0	1	0	0.347	0.159
6	2	0	2	0	2	0	0.347	0.3
7	2	0	2	0	2	0	0.347	0.3
8	2	0	3	0	2	0	0.419	0.3
9	2	0	3	0	2	0	0.419	0.3
10	2	0	3	0	2	0	0.419	0.3
11	3	0	3	0	3	0	0.419	0.437
12	3	0	4	0	3	0	0.484	0.437
13	4	0	4	0	4	0	0.484	0.586
14	4	0	5	0	4	0	0.693	0.586
15	5	0	5	0	5	0	0.693	0.715
16	5	0	5	0	5	0	0.693	0.715
17	5	0	5	0	5	0	0.693	0.715
18	5	0	6	0	5	0	0.749	0.715
19	5	0	6	0	5	0	0.749	0.715
20	6	0	6	0	6	0	0.749	0.852
21	6	0	7	0	6	0	0.91	0.852
22	6	1	7	0	6	0.595	0.91	0.852
23	7	1	7	0	7	0.595	0.91	0.981
24	8	1	7	1	7	0.595	0.91	1.113
25	8	1	7	1	7	0.595	0.91	1.113

Step	waiting in queue A	waiting in queue B	waiting in queue Cord1	waiting in queue Cord2	time weighted length A	time weighted length B	time weighted length Cord1	time weighted length Cord2
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0.046	0	0
12	0	0.062	0	0	0	0.062	0	0
13	0	0.062	0	0	0	0.062	0	0
14	0	0.062	0	0	0	0.062	0	0
15	0	0.062	0	0	0	0.062	0	0
16	0	0.062	0	0	0	0.062	0	0
17	0	0.062	0	0	0	0.062	0	0
18	0	0.062	0	0	0	0.062	0	0
19	0	0.062	0	0	0	0.062	0	0
20	0	0.062	0	0	0	0.062	0	0
21	0	0.062	0	0	0	0.062	0	0
22	0	0.062	0	0	0	0.062	0	0
23	0	0.062	0.034	0	0	0.062	0.034	0
24	0	0.062	0.034	0	0	0.062	0.034	0
25	0	0.062	0.034	0	0	0.062	0.034	0

Step	total being in system order1	total being in system order2	stock_num A	stock_num B	Future Event Type 1	Future Event Time 1	Future Event Type 2	Future Event Time 2
1	0	0	0	0	end of service B	0.255	arrival2	0.26
2	0	0	0	0	arrival2	0.26	end of service C	0.414
3	0	0	0	0	end of service C	0.414	end of service B	0.467
4	0	0.299	0	0	end of service B	0.467	arrival2	0.744
5	0	0.299	0	0	end of service C	0.608	arrival2	0.744
6	0	0.647	0	0	arrival2	0.744	arrival1	1.378
7	0	0.647	0	0	end of service B	0.816	arrival2	0.904
8	0	0.647	0	0	arrival2	0.904	end of service C	0.953
9	0	0.647	0	0	arrival2	0.907	end of service C	0.953
10	0	0.647	0	0	end of service C	0.953	end of service B	0.969
11	0	0.856	0	0	end of service B	0.969	arrival1	1.378
12	0	0.856	0	0	end of service C	1.118	end of service B	1.178
13	0	1.07	0	0	end of service B	1.178	arrival1	1.378
14	0	1.07	0	0	end of service C	1.307	arrival1	1.378
15	0	1.47	0	0	arrival1	1.378	arrival2	1.601
16	0	1.47	0	0	arrival2	1.601	end of service A	1.973
17	0	1.47	0	0	end of service B	1.657	arrival2	1.717
18	0	1.47	0	0	arrival2	1.717	end of service C	1.794
19	0	1.47	0	0	end of service C	1.794	end of service B	1.878
20	0	1.663	0	0	end of service B	1.878	end of service A	1.973
21	0	1.663	0	0	end of service A	1.973	end of service C	2.007
22	0	1.663	0	0	end of service C	2.007	arrival2	2.188
23	0	1.953	0	0	end of service C	2.139	arrival2	2.188
24	0.761	1.953	0	0	arrival2	2.188	out of service A	2.418
25	0.761	1.953	0	0	end of service B	2.362	out of service A	2.418

Step	Future Event Type 3	Future Event Time 3	Future Event Type 4	Future Event Time 4	Future Event Type 5	Future Event Time 5	Future Event Type 6	Future Event Time 6
1	arrival1	1.378	out of service A	2.418	out of service B	5.211	end of simulation	24
2	arrival1	1.378	out of service A	2.418	out of service B	5.211	end of simulation	24
3	arrival2	0.744	arrival1	1.378	out of service A	2.418	out of service B	5.211
4	arrival1	1.378	out of service A	2.418	out of service B	5.211	end of simulation	24
5	arrival1	1.378	out of service A	2.418	out of service B	5.211	end of simulation	24
6	out of service A	2.418	out of service B	5.211	end of simulation	24		
7	arrival1	1.378	out of service A	2.418	out of service B	5.211	end of simulation	24
8	arrival1	1.378	out of service A	2.418	out of service B	5.211	end of simulation	24
9	end of service B	0.969	arrival1	1.378	out of service A	2.418	out of service B	5.211
10	arrival1	1.378	arrival2	1.601	out of service A	2.418	out of service B	5.211
11	arrival2	1.601	out of service A	2.418	out of service B	5.211	end of simulation	24
12	arrival1	1.378	arrival2	1.601	out of service A	2.418	out of service B	5.211
13	arrival2	1.601	out of service A	2.418	out of service B	5.211	end of simulation	24
14	arrival2	1.601	out of service A	2.418	out of service B	5.211	end of simulation	24
15	out of service A	2.418	out of service B	5.211	end of simulation	24		
16	out of service A	2.418	arrival1	3.297	out of service B	5.211	end of simulation	24
17	end of service A	1.973	out of service A	2.418	arrival1	3.297	out of service B	5.211
18	end of service A	1.973	out of service A	2.418	arrival1	3.297	out of service B	5.211
19	end of service A	1.973	arrival2	2.188	out of service A	2.418	arrival1	3.297
20	arrival2	2.188	out of service A	2.418	arrival1	3.297	out of service B	5.211
21	arrival2	2.188	out of service A	2.418	arrival1	3.297	out of service B	5.211
22	out of service A	2.418	arrival1	3.297	out of service B	5.211	end of simulation	24
23	out of service A	2.418	arrival1	3.297	out of service B	5.211	end of simulation	24
24	arrival1	3.297	out of service B	5.211	end of simulation	24		
25	arrival2	2.534	arrival1	3.297	out of service B	5.211	end of simulation	24

Step	Future Event Time 4	Future Event Type 5	Future Event Time 5	Future Event Type 6	Future Event Time 6	Future Event Type 7	Future Event Time 7	Future Event Type 8	Future Event Time 8
1	2.418	out of service B	5.211	end of simulation	24				
2	2.418	out of service B	5.211	end of simulation	24				
3	1.378	out of service A	2.418	out of service B	5.211	end of simulation	24		
4	2.418	out of service B	5.211	end of simulation	24				
5	2.418	out of service B	5.211	end of simulation	24				
6	5.211	end of simulation	24						
7	2.418	out of service B	5.211	end of simulation	24				
8	2.418	out of service B	5.211	end of simulation	24				
9	1.378	out of service A	2.418	out of service B	5.211	end of simulation	24		
10	1.601	out of service A	2.418	out of service B	5.211	end of simulation	24		
11	2.418	out of service B	5.211	end of simulation	24				
12	1.601	out of service A	2.418	out of service B	5.211	end of simulation	24		
13	2.418	out of service B	5.211	end of simulation	24				
14	2.418	out of service B	5.211	end of simulation	24				
15	5.211	end of simulation	24						
16	3.297	out of service B	5.211	end of simulation	24				
17	2.418	arrival1	3.297	out of service B	5.211	end of simulation	24		
18	2.418	arrival1	3.297	out of service B	5.211	end of simulation	24		
19	2.188	out of service A	2.418	arrival1	3.297	out of service B	5.211	end of simulation	24
20	2.418	arrival1	3.297	out of service B	5.211	end of simulation	24		
21	2.418	arrival1	3.297	out of service B	5.211	end of simulation	24		
22	3.297	out of service B	5.211	end of simulation	24				
23	3.297	out of service B	5.211	end of simulation	24				
24	5.211	end of simulation	24						
25	3.297	out of service B	5.211	end of simulation	24				

[١]

Banks, Jerry, John S. CARSON II, and L. Barry. "Discrete-event system simulation fifth edition."
(2014).