  
دانشگاه صنعتی امیرکبیر (پلی‌تکنیک تهران)  
دانشکده مهندسی صنایع و سیستم‌های مدیریت

پروژه درس اصول شبیه‌سازی

**اعضا:**  
حامد اعراب – 9925003  
شهریار خلوتی – 9925015

**استاد:** دکتر عباس احمدی  
**تدریس‌یار:** مهدی محمدی

بهار 1403

**فهرست**

[شرح مسئله 1](#_Toc170667600)

[مقدمه 2](#_Toc170667601)

[چارچوبی برای شبیه‌سازی رویدادهای گسسته 2](#_Toc170667602)

[کنترلر 3](#_Toc170667603)

[رویداد 5](#_Toc170667604)

[مدل‌سازی و حل مسئله‌ها 7](#_Toc170667605)

[موجودیت‌ها 7](#_Toc170667606)

[کنترلر 8](#_Toc170667607)

[رویدادها 9](#_Toc170667608)

[نتایج 11](#_Toc170667609)

[مسئله یکم 12](#_Toc170667610)

[مسئله دوم 12](#_Toc170667611)

[نتیجه‌گیری 15](#_Toc170667612)

[جدول گزارش عملکرد اعضای گروه 15](#_Toc170667613)

شرح مسئله

به اورژانس یک بیمارستان هر 4±20دقیقه یک بیمار وارد می‌شود. اورژانس دارای دو پزشک است. 15 درصد بیماران کسانی هستند که نیاز به رسیدگی فوری دارند اما بقیه می‌توانند صبر کنند. به بیماران دسته اول بالاترین درجه اولویت، یعنی اولویت 3 داده می‌شود تا در اسرع وقت، پزشکی را به مدت 5±30 دقیقه ببینند. ولی بعد اولویت آنها به 2کاهش می‌یابد و به انتظار می‌مانند تا دوباره یک پزشک آزاد شود تا این بار به مدت 13±28 دقیقه مداوا و سپس مرخص شوند. بیماران دسته دوم ابتدا اولویت 1می‌گیرند و هنگامی که نوبتشان برسد به مدت 10±15 دقیقه درمان می‌شوند، سپس اولویتشان به 2 افزایش می‌یاید. به انتظار می‌مانند تا یک پزشک آزاد شود و به مدت 7±10دقیقه، آنها را به طور نهایی درمان کند و سرانجام مرخص شوند.

الف) ابتدا نمودار کنترولر شبیه‌سازی و نمودارهای جریان رابرای پیشامدهای اصلی ترسیم نمایید. سپس مدل کامپیوتری شبیه‌سازی را برای 30 روز 24ساعته اجرا نمایید. به منظور ایجاد باری از بیماران در سیستم، پیش از اجرای 30روزه فوق، یک دوره راه‌اندازی 2روزه در نظر بگیرید. شرایط را برای دوره 2روز راه‌اندازی و دوره 30روزه به‌طور جداگانه گزارش کنید. در صورت امکان، از برآوردهای نقطه‌ای و فاصله‌ای استفاده نمایید.

ب) متوسط و ماکزیمم طول صف را برای بیماران (دسته اول و دوم) به تفکیک صف‌ها محاسبه کنید. چند درصد از بیماران مجبور به انتظار در صف نیستند؟ نمودار فراوانی مدت انتظار در صف را برای بیماران به تفکیک صف ترسیم کنید. چند درصد از بیماران برای دیدن اولین پزشک کمتر از 4 دقیقه صبر می‌کنند؟ مقادیر زمان انتظار برای بیماران را به تفکیک دو دسته به همراه نمودار تابع توزیع ارائه نمایید. درصد اشتغال پزشکان را محاسبه کنید.

ج) تعداد بهینه پزشک‌ها را با هدف حداقل کردن کل زمان انتظار و ماکزیمم کردن درصد اشتغال پزشکان بیابید.

# مقدمه

مسئله‌هایی که پیش‌تر آمده‌اند، در بنیان خود، شبیه‌سازی رویدادهای گسسته[[1]](#footnote-1) را بیان می‌کنند. در مسئله نخست، سیستمی داریم با یک صف[[2]](#footnote-2) و دو خدمت‌دهنده[[3]](#footnote-3) که در کنار هم به مشتریان[[4]](#footnote-4) خدمت می‌دهند. رویدادهای[[5]](#footnote-5) سیستم نیز به ورود[[6]](#footnote-6) و خروج[[7]](#footnote-7) مشتریان محدود می‌شوند. در مسئله دوم، حالت کلی مسئله پیشین را داریم: یک صف، چندین خدمت‌دهنده موازی[[8]](#footnote-8)، و همان رویدادها.

حال، باید به این نکته نیز بپردازیم که علی‌رغم شباهت سیستم‌های این دو مسئله، خواسته‌های هر مسئله متفاوت است. در مسئله نخست، باید سیستم را با دو روش متفاوت در انتخاب خدمت‌دهنده شبیه‌سازی کنیم. سپس با به‌دست آوردن و مقایسه چند معیار کارایی[[9]](#footnote-9)، مانند درصد زمان اشتغال[[10]](#footnote-10) خدمت‌دهنده‌ها، روش برتر را بیابیم. اما در مسئله دوم، صرفا باید تعداد بهینه‌ی خدمت‌دهنده‌ها را با توجه به مقادیر داده‌شده برای زمان‌های میان هر رویداد بیابیم، به طوری که میانگین زمان انتظار کمینه[[11]](#footnote-11) و درصد زمان اشتغال هر خدمت‌دهنده بیشینه[[12]](#footnote-12) شود.

چارچوبی برای شبیه‌سازی رویدادهای گسسته

از ابتدای این ترم، تصمیم گرفتیم که یک چارچوب[[13]](#footnote-13) و بنیان واحد برای حل مسائل شبیه‌سازی طراحی و پیاده کنیم. این چارچوب باید ویژگی‌های زیر را داشته باشد:

1. گسترش‌پذیری[[14]](#footnote-14): در مسائل گوناگون با سیستم‌هایی سروکار داریم که متغیرها[[15]](#footnote-15)، وضعیت[[16]](#footnote-16)، منطق[[17]](#footnote-17)، و رفتار[[18]](#footnote-18) مخصوص خود را دارند. بنابراین، به چارچوبی نیاز داریم که حداقل امکانات شبیه‌سازی را فراهم کرده و اجازه‌ی افزودن سایر اجزای مورد نیاز را بدهد.
2. مقیاس‌پذیری[[19]](#footnote-19): مسائل شبیه‌سازی در پیچیدگی[[20]](#footnote-20) و اندازه با هم تفاوت دارند. مثلا ممکن است تعداد صف‌ها، انواع و مراحل خدمت‌دهی، و تعداد خدمت‌دهنده‌ها در هر مسئله متفاوت باشند. بنابراین، چارچوب مورد استفاده باید قابلیت به‌کارگیری را در مقیاس‌های مختلف داشته باشد.
3. بهره‌وری[[21]](#footnote-21): چارچوب ما باید در اجرای فرآیند شبیه‌سازی سریع بوده و مصرف منابع کمی داشته باشد. این منابع شامل انرژی، حافظه، و فضای ذخیره‌سازی‌اند. با داشتن بهره‌وری مناسب می‌توان یک سیستم را به دفعات بیش‌تری شبیه‌سازی، و از صحت خروجی‌ها و درستی مدل شبیه‌سازی‌مان اطمینان حاصل کرد.

با توجه به این ویژگی‌ها، تصمیم بر آن شد که چارچوب ما دو موجودیت کنترلر[[22]](#footnote-22) و رویداد را در سطحی پایه‌ای و بنیادین برای استفاده فراهم کند. بدین ترتیب می‌توانیم این دو موجودیت را متناسب با شرایط هر مسئله گسترش داده و سیستم را مدل‌سازی کنیم. در ادامه، به جزئیات آن‌ها بیش‌تر می‌پردازیم.

کنترلر

کنترلر وظیفه‌ی اجرا و مدیریت فرآیند شبیه‌سازی را بر عهده دارد. به زبان ساده، با شروع فرآیند شبیه‌سازی، رویدادها به کنترلر ارسال[[23]](#footnote-23) می‌شوند. کنترلر آن‌ها را به «فهرست رویدادهای آینده[[24]](#footnote-24)» می‌افزاید و این فهرست را بر اساس «زمان مقرر[[25]](#footnote-25)» رویدادها مرتب می‌کند. پس از پایان شلیک[[26]](#footnote-26) رویداد پیشین، رویداد پیش رو[[27]](#footnote-27) از FEL خارج و پس از جلو بردن ساعت شبیه‌سازی[[28]](#footnote-28) شلیک می‌گردد. در این میان، اگر ساعت شبیه‌سازی به زمان مشخص شده برای پایان برسد یا از آن فراتر رود، شبیه‌سازی متوقف می‌گردد.

کنترلر دو پارامتر رویداد نخستین[[29]](#footnote-29) و زمان پایان شبیه‌سازی[[30]](#footnote-30) را دریافت و صحت‌سنجی[[31]](#footnote-31) می‌کند. سنجش صحت پارامترها عبارتند از: (1) مثبت بودن زمان پایان شبیه‌سازی و (2) صفر بودن فاصله زمانی رویداد نخستین. پس از سنجش صحت پارامترها، رویداد نخستین به FEL افزوده شده و ساعت شبیه‌سازی برابر با دقیقه صفر قرار داده می‌شود. لازم به ذکر است که در این چارچوب، تمامی زمان‌ها با واحد دقیقه به‌کار می‌روند. خلاصه‌ای از پارامترها و متغیرهای مربوط به کنترلر را می‌توانید مشاهده کنید:

|  |  |  |  |
| --- | --- | --- | --- |
| **پارامترها** | | **متغیرها** | |
| **نام** | **توضیحات** | **نام** | **توضیحات** |
| initialEvent | رویداد نخستین | futureEvents | فهرست رویدادهای آینده |
| stopTime | زمان پایان شبیه‌سازی | clock | ساعت شبیه‌سازی |

حال، کنترلر برای شروع فرآیند شبیه‌سازی آماده است. با شروع فرآیند، رویداد نخستین که به FEL افزوده شده بود را از این فهرست خارج و سپس شلیک می‌کنیم. در پی آن، رویدادهای بعدی دریافت شده و به FEL افزوده می‌گردند و این حلقه[[32]](#footnote-32) تکرار می‌گردد، تا این که به زمان پایان شبیه‌سازی برسیم. در شکل بعدی، نمودار جریان را برای فرآیند شبیه‌سازی کنترلر می‌توانید مشاهده کنید. این نمودار همان توضیحات ساده‌ای که پیش‌تر داده شدند را با جزئیات بیش‌تر ارائه می‌دهد.



رویداد

این موجودیت بستر را برای ایجاد رویدادهای مختلف در شبیه‌سازی یک سیستم فراهم می‌کند. تنها پارامتری که دریافت می‌کند فاصله[[33]](#footnote-33) زمان شلیک آن با ساعت شبیه‌سازی است و پس از آن که به کنترلر ارسال شد، «زمان مقرر» آن برای شلیک مشخص می‌گردد. در نهایت، کنترلر رویداد را به FEL افزوده و فهرست بر اساس زمان مقرر مرتب می‌کند.

خلاصه‌ای از پارامترها و متغیرهای مربوط به رویداد را می‌توانید مشاهده کنید:

|  |  |  |  |
| --- | --- | --- | --- |
| **پارامترها** | | **متغیرها** | |
| **نام** | **توضیحات** | **نام** | **توضیحات** |
| interval | فاصله زمانی | dueTime | زمان مقرر |

در ادامه، نمودار جریان را برای ارسال یک رویداد به کنترلر شبیه‌سازی مشاهده می‌کنید.



در ادامه به نحوه مدل‌سازی و رسیدن به جواب هر مسئله با استفاده از این چارچوب می‌پردازیم.

مدل‌سازی و حل مسئله‌ها

برای حل مسئله با استفاده از چارچوب پیشنهادی، کنترلری طراحی می‌شود که رفتارهای بنیادین و پایه‌ای را از کنترلر چارچوب به‌ارث می‌برد. به نوعی، ما به این مسائل با دیدی شیء‌گرا[[34]](#footnote-34) نگاه کرده و در کنار کنترلر، رویدادهای شبیه‌سازی را نیز با ارث‌بری[[35]](#footnote-35) از رویداد بنیادین پیاده‌ می‌کنیم.

موجودیت‌ها

**بیمار**

در این سیستم، بیمار همان مشتری است که خدمت دریافت می‌کند؛ از این رو، پارامترهای زیر به به تفکیک مرحله (درمان/معاینه) برایش تعریف می‌شود:

* زمان ورود
* زمان خدمت
* زمان خروج
* پزشک ❓
* زمان انتظار
* اولویت (کم/زیاد)

پارامتر زمان انتظار در هر مرحله از رابطه‌ی زیر به دست می‌آید:

و مقدار مجموع زمان انتظار از جمع زمان‌های انتظار هر مرحله:

هر بیمار که وارد بیمارستان می‌شود، یا وضعیت وخیم و اورژانسی دارد یا وضعی قابل کنترل. اگر وضعیتش اورژانسی باشد، برای مرحله اول (معاینه) اولویت 3 که بیشترین اولویت است را می‌گیرد و سپس به اولویت 2 کاهش پیدا می‌کند. اگر وضعیتش غیراورژانسی باشد، برای مرحله اول اولویت 1 می‌گیرد و سپس به اولویت 2 افزایش پیدا می‌کند. به همین دلیل، پارامتری با نام priority نیز تعریف می‌شود که نوع بیمار را مشخص کند. از آنجا که در فاز نتیجه‌گیری، می‌خواهیم آماره‌های متفاوتی برای هر دو نوع اورژانسی و غیراورژانسی داشته باشیم، موجودیت بیمار به دو موجودیت بیمار اورژانسی یا High Priority و غیر اورژانسی Low Priority تقسیم می‌شود که البته تمام پارامترهای موجودیت بیمار را به ارث می‌برند.

پزشک

همانطور که بیمار به مثابه‌ی مشتری سیستم است، پزشک نیز سرور یا خدمت‌دهنده است و پارامترهای زیر برایش تعریف می‌شود:

* وضعیت (آزاد/مشغول)
* زمان اشتغال
* بیماران درمان شده
* مرحله (معاینه/درمان)

مقدار زمان اشتغال با محاسبه‌ی مجموع زمان‌های درمانِ بیماران درمان‌شده توسط همان پزشک در همان مرحله به دست می‌آید.

با توجه به اینکه که در این سیستم دو مرحله‌ی معاینه و درمان وجود دارد و آماره‌های متفاوتی برای هر بخش می‌خواهیم، دو موجودیت پزشک معاینه‌گر و پزشک درمانگر تعریف می‌شود که تمامی پارامترهای موجودیت پزشک را به ارث می‌برند و تنها تفاوتشان در نوع کارشان است.

کنترلر

کنترلر این سیستم فرآیند شبیه‌سازی را از چارچوب به‌ارث می‌برد. اما نیاز است تا پارامترها، متغیرها، و خروجی‌های شبیه‌سازی مختص به این مسئله را به آن اضافه کنیم. رویکرد انتخاب سرور به صورت رندوم است و از بین پزشکان آزاد، یک نفر به صورت رندوم به عنوان خدمت‌دهنده به بیمار بعدی انتخاب می‌شود. پارامترهای دیگر کنترلر، تعداد دکتر‌های معاینه‌گر و تعداد دکترها‌ی درمانگر است. در مسئله‌ی اصلی، مقدار هر دو پارامتر برابر 1 است، اما هنگام محاسبه‌ی تعداد دکتر بهینه، این دو پارامتر مقادیر متفاوتی می‌گیرند.

سه صف سیستم، صف بیماران اورژانسی، صف بیماران عادی، و صف مرحله‌ی درمان است که مورد آخر بین هر دو نوع بیمار مشترک است.

موارد دیگری که در کنترلر تعریف می‌شوند:

- مدت زمان گرم کردن

- بیماران درمان‌شده

- پزشکان درمانگر

- پزشکان معاینه‌گر

خروجی‌هایی که از سیستم انتظار داریم و مقادیرشان محاسبه می‌شود:

خلاصه‌ای از این بخش را در جدول زیر می‌توانید مشاهده کنید:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **پارامترها** | | **متغیرها** | | **خروجی‌ها** | |
| **نام** | **توضیحات** | **نام** | **توضیحات** | **نام** | **توضیحات** |
| numberOfServers | تعداد خدمت‌دهنده‌ها | dueTime | زمان مقرر | ABTP | میانگین درصد زمان اشتغال خدمت‌دهنده‌ها[[36]](#footnote-36) |
| serverSelectionStrategy | رویکرد انتخاب خدمت‌دهنده[[37]](#footnote-37) | historicalCustomers | فهرست مشتریان تاریخی | AWTP | میانگین درصد زمان انتظار مشتریان[[38]](#footnote-38) |

در نهایت، در شکل زیر، نمودار جریان فرآیند شبیه‌سازی کنترلر شبیه‌سازی سیستم چندخدمت‌دهی موازی دارای یک صف را ببینید:



### رویدادها

#### ورود

با شلیک رویداد ورود، زمان ورود مشتری مشخص و رویداد ورود بعدی نیز به کنترلر ارسال می‌شود. با استفاده از رویکرد مشخص‌شده برای انتخاب خدمت‌دهنده‌، به دنبال خدمت‌دهنده‌ای برای خدمت‌دهی برای مشتری می‌گردیم. اگر خدمت‌دهنده‌ای دردسترس نبود، مشتری را به صف انتظار اضافه می‌کنیم. در غیر این صورت، وضعیت خدمت‌دهنده را به «مشغول» تغییر داده و رویداد خروج مشتری از این ایستگاه خدمت‌دهی را به کنترلر ارسال می‌کنیم. نمودار جریان این رویداد را در شکل بعد می‌توانید مشاهده کنید.



#### خروج

با شلیک رویداد خروج، مدت دریافت خدمت و زمان خروج مشتری را مشخص می‌کنیم. سپس، مشتری را به فهرست مشتریان تاریخی می‌افزاییم. همچنین، مدت خدمت‌دهی را به مجموع زمان اشتغال خدمت‌دهنده اضافه می‌کنیم. حال، اگر صف انتظار خالی از مشتری باشد، وضعیت خدمت‌دهنده را به «دردسترس» تغییر می‌دهیم. در غیر این صورت، اولین مشتری حاضر در صف را از صف خارج کرده و رویداد خروج وی از این ایستگاه خدمت‌دهی را به کنترلر ارسال می‌کنیم. در نمودار پایین، جزئیات جریان این رویداد را می‌توانید ببینید.



نتایج

شبیه سازی را برای 30 روز 24 ساعته اجرا می‌کنیم. قبل از اجرای اصلی، یک دوره راه‌اندازی 2 روزه خواهیم داشت.

راه‌اندازی

نتایج به‌دست آمده در دوره راه‌اندازی به صورت زیر است:

|  |  |
| --- | --- |
| آماره | مقدار |
| میانگین درصد اشتغال پزشکان معاینه‌گر | 0.816000 |
| میانگین درصد اشتغال پزشکان درمان‌گر | 0.548436 |
| تعداد کل بیماران | 142.900000 |
| میانگین درصد انتظار | 0.273416 |
| بیشترین طول صف بیماران اورژانسی مرحله معاینه | 1.980000 |
| میانگین طول صف بیماران اورژانسی مرحله معاینه | 0.006251 |
| درصد بیماران اورژانسی بدون معطلی مرحله معاینه | 0.401133 |
| درصد بیماران اورژانسی کم معطلی (زیر 4 دقیقه) مرحله معاینه | 0.185962 |
| بیشترین طول صف بیماران غیراورژانسی مرحله معاینه | 4.860000 |
| میانگین طول صف بیماران غیراورژانسی مرحله معاینه | 0.317147 |
| درصد بیماران غیراورژانسی بدون معطلی مرحله معاینه | 0.423251 |
| درصد بیماران غیراورژانسی کم معطلی (زیر 4 دقیقه) مرحله معاینه | 0.124247 |
| بیشترین طول صف مرحله درمان | 3.220000 |
| میانگین طول صف مرحله درمان | 0.051021 |
| درصد بیماران بدون معطلی مرحله درمان | 0.699558 |
| درصد بیماران کم معطلی (زیر 4 دقیقه) مرحله درمان | 0.115073 |

اجرای اصلی

نتایج به‌دست آمده در شبیه‌سازی اصلی به صورت زیر است:

|  |  |
| --- | --- |
| آماره | مقدار |
| میانگین درصد اشتغال پزشکان معاینه‌گر | 0.819477 |
| میانگین درصد اشتغال پزشکان درمان‌گر | 0.554320 |
| تعداد کل بیماران | 2160.060000 |
| میانگین درصد انتظار | 0.287117 |
| بیشترین طول صف بیماران اورژانسی مرحله معاینه | 1.980000 |
| میانگین طول صف بیماران اورژانسی مرحله معاینه | 0.006251 |
| درصد بیماران اورژانسی بدون معطلی مرحله معاینه | 0.430730 |
| درصد بیماران اورژانسی کم معطلی (زیر 4 دقیقه) مرحله معاینه | 0.165075 |
| بیشترین طول صف بیماران غیراورژانسی مرحله معاینه | 4.860000 |
| میانگین طول صف بیماران غیراورژانسی مرحله معاینه | 0.317147 |
| درصد بیماران غیراورژانسی بدون معطلی مرحله معاینه | 0.428526 |
| درصد بیماران غیراورژانسی کم معطلی (زیر 4 دقیقه) مرحله معاینه | 0.112624 |
| بیشترین طول صف مرحله درمان | 3.220000 |
| میانگین طول صف مرحله درمان | 0.051021 |
| درصد بیماران بدون معطلی مرحله درمان | 0.696804 |
| درصد بیماران کم معطلی (زیر 4 دقیقه) مرحله درمان | 0.105609 |

تعداد بهینه پزشکان

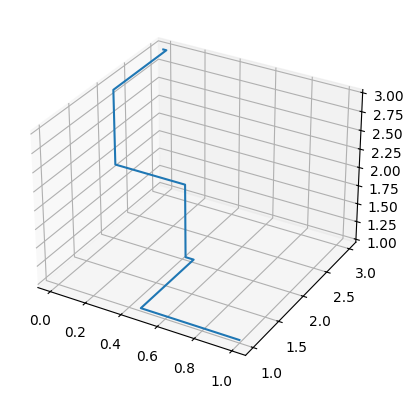
برای به دست آوردن تعداد بهینه‌ی پزشکان، یک معیار امتیاز با رابطه‌ی زیر تعریف می‌کنیم:

مقادیر بیشتر آلفا، به بیشینه‌سازی درصد به‌کارگیری، و مقادیر کمتر آن به کمینه‌سازی زمان انتظار اولویت می‌دهند.

شبیه‌سازی را به تعداد 50 بار برای هر زوج (تعداد پزشک درمانگر، تعداد پزشک معاینه‌گر) اجرا کرده، و درصدهای خدمت‌دهی و انتظار را به دست می‌آوریم. تعداد هر نوع پزشک بین 1 تا 3 متغیر است و در نتیجه 9 ترکیب مختلف داریم و امتیاز هر حالت را به ازای آلفاهای متفاوت محاسبه می‌کنیم:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| تعداد پزشکان معاینه‌گر | تعداد پزشکان درمانگر | میانگین درصد اشتغال | میانگین درصد انتظار | امتیاز (آلفا=0.2) | امتیاز (آلفا=0.4) | امتیاز (آلفا=0.6) | امتیاز (آلفا=0.8) |
| 1 | 1 | 0.686930 | 0.283851 | 0.710305 | 0.704461 | 0.698618 | 0.692774 |
| 2 | 0.4528201 | 0.234258 | 0.74234 | 0.642725 | 0.581217 | 0.519709 |
| 3 | 0.342691 | 0.237104 | 0.678855 | 0.594814 | 0.510773 | 0.426732 |
| 2 | 1 | 0.458467 | 0.084746 | 0.823897 | 0.732539 | 0.641182 | 0.549824 |
| 2 | 0.343806 | 0.002386 | 0.866852 | 0.736091 | 0.605329 | 0.474567 |
| 3 | 0.274620 | 0.000915 | 0.854192 | 0.709299 | 0.564406 | 0.419513 |
| 3 | 1 | 0.343409 | 0.083822 | 0.801624 | 0.687070 | 0.572517 | 0.457963 |
| 2 | 0.274319 | 0.001573 | 0.853606 | 0.708784 | 0.563962 | 0.419141 |
| 3 | 0.228843 | 0.000003 | 0.845766 | 0.691535 | 0.537304 | 0.383074 |

حال، اگر امتیاز را با مقادیر مختلف آلفا (با گام‌هایی به اندازه 0.001) محاسبه کنیم، خواهیم داشت:



با این حساب، می‌توان گفت تعداد 3 دکتر درمانگر و 3 دکتر معاینه‌گر بهینه است.

# نتیجه‌گیری

از مسئله اول نتیجه می‌گیریم که به دلیل یکسان بودن تابع توزیع زمان خدمت‌دهی آریا و بردیا، تفاوت رویکردها در انتخاب خدمت‌دهنده برای خدمت‌دهی به هر مشتری، تغییر چشمگیری در کارایی سیستم و نتایج حاصل نمی‌کند. در انتخاب ترتیبی نسبت به انتخاب تصادفی، درصد به‌کارگیری آریا کمی بیشتر و درصد به‌کارگیری بردیا کمی کمتر است. با این حال، میانگین درصد به‌کارگیری آریا و بردیا در هر دو حالت برابر است.

اما از مسئله‌ی دوم، با استفاده از به معیار امتیاز محاسبه شده، نتیجه می‌گیریم که با توجه به اولیت‌های مختلف ذینفعان، تعداد بهینه‌ی خدمت‌دهنده‌ها می‌تواند متفاوت باشد. اما همانطور که در شکل آخر مشاهده می‌کنید، تعداد 5 خدمت‌دهنده بیشترین بازه‌ مقادیر آلفا را در بهینگی را به خود اختصاص داده است. بنابراین، در حالت کلی، ما 5 خدمت‌دهنده را پیشنهاد می‌کنیم.

# جدول گزارش عملکرد اعضای گروه

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **نام** | **شماره دانشجویی** | **وظایف** | | | | | | |
| طراحی و پیاده‌سازی چارچوب | پیاده‌سازی کنترلر مسئله 1 | پیاده‌سازی کنترلر مسئله 2 | پیاده‌سازی رویدادها | طراحی نمودارها | نگارش گزارش | ویرایش گزارش |
| حامد اعراب | 9925003 |  |  |  |  |  |  |  |
| شهریار خلوتی | 9925015 |  |  |  |  |  |  |  |

1. Discrete-Event Simulation [↑](#footnote-ref-1)
2. Queue [↑](#footnote-ref-2)
3. Server [↑](#footnote-ref-3)
4. Customers [↑](#footnote-ref-4)
5. Events [↑](#footnote-ref-5)
6. Arrival [↑](#footnote-ref-6)
7. Departure [↑](#footnote-ref-7)
8. Parallel [↑](#footnote-ref-8)
9. Performance Metrics [↑](#footnote-ref-9)
10. Busy Time Percentage (BTP) [↑](#footnote-ref-10)
11. Minimized [↑](#footnote-ref-11)
12. Maximized [↑](#footnote-ref-12)
13. Framework [↑](#footnote-ref-13)
14. Extensibility [↑](#footnote-ref-14)
15. Variables [↑](#footnote-ref-15)
16. State [↑](#footnote-ref-16)
17. Logic [↑](#footnote-ref-17)
18. Behavior [↑](#footnote-ref-18)
19. Scalability [↑](#footnote-ref-19)
20. Complexity [↑](#footnote-ref-20)
21. Efficiency [↑](#footnote-ref-21)
22. Controller [↑](#footnote-ref-22)
23. Dispatch [↑](#footnote-ref-23)
24. Future Events List (FEL) [↑](#footnote-ref-24)
25. Due Time [↑](#footnote-ref-25)
26. Triggering [↑](#footnote-ref-26)
27. Upcoming Event [↑](#footnote-ref-27)
28. Simulation Clock [↑](#footnote-ref-28)
29. Initial Event [↑](#footnote-ref-29)
30. Simulation Stop Time [↑](#footnote-ref-30)
31. Validate [↑](#footnote-ref-31)
32. Loop [↑](#footnote-ref-32)
33. Interval [↑](#footnote-ref-33)
34. Object-Oriented [↑](#footnote-ref-34)
35. Inheritance [↑](#footnote-ref-35)
36. Average Busy Time Percentage (ABTP) [↑](#footnote-ref-36)
37. Server Selection Strategy [↑](#footnote-ref-37)
38. Average Waiting Time Percentage (AWTP) [↑](#footnote-ref-38)