



دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی صنایع و سیستم‌های مدیریت

تمرین یکم درس اصول شبیه‌سازی

اعضا:

حامد اعراب – ۹۹۲۵۰۰۳

شهریار خلوتی – ۹۹۲۵۰۱۵

استاد: دکتر عباس احمدی

تدریس‌یار: مهدی محمدی

بهار ۱۴۰۳

فهرست

۱	مسائل
۱	مسئله یکم
۱	مسئله دوم
۲	مقدمه
۲	چارچوبی برای شبیه‌سازی رویدادهای گستته
۳	کنترلر
۵	رویداد
۷	مدل‌سازی و حل مسئله‌ها.....
۷	حالت کلی: چند خدمت‌دهنده
۷	موجودیت‌های مختص به این سیستم
۸	کنترلر
۹	رویدادها.....
۱۱.....	حالات خاص: دو خدمت‌دهنده
۱۲.....	نتایج.....
۱۲.....	مسئله یکم
۱۳.....	مسئله دوم
۱۵.....	نتیجه‌گیری
۱۵.....	جدول گزارش عملکرد اعضای گروه

مسائل

مسائل

مسئله یکم

آریا و بردیا دو خدمتدهنده در یک رستوران خودرویی هستند. فاصله میان ورود هر دو خودرو یکنواخت و در بازه ۳ تا ۶ دقیقه است. زمان خدمت آریا و بردیا به هر خودرو، یکنواخت و در بازه ۹ تا ۱۲ دقیقه است.

- (۱) نمودارهای جریان را برای کنترلر و رویدادهای شبیه‌سازی بکشید.
- (۲) سیستم را برای ۹ ساعت شبیه‌سازی کنید. در حالت اول، هر بار نخستین خدمتدهنده در دسترس را برای خدمتدهی، و در حالت دوم، یک خدمتدهنده در دسترس را تصادفی انتخاب کنید.
- (۳) میزان به کارگیری (درصد زمان اشتغال) آریا و بردیا را در هر حالت تخمین بزنید.
- (۴) میانگین زمان انتظار هر مشتری را در هر حالت به دست آورید.
- (۵) در هر حالت چند نفر وارد صف انتظار می‌شوند؟

مسئله دوم

در یک سیستم خدماتی با چندین ایستگاه موازی (I خدمتدهنده)، زمان میان ورود هر دو مشتری یکنواخت و در بازه ۰ تا ۴ دقیقه، و زمان خدمت هر خدمتدهنده به هر مشتری، یکنواخت و در بازه ۶ تا ۱۰ دقیقه است.

- (۱) نمودارهای جریان را برای کنترلر و رویدادهای شبیه‌سازی بکشید.
- (۲) سیستم را برای ۸ ساعت شبیه‌سازی کنید.
- (۳) تعداد بهینه خدمتدهنده‌ها را به دست آورید، به طوری که میانگین زمان انتظار کمینه و درصد زمان اشتغال هر خدمتدهنده بیشینه شود.

مقدمه

مسئله‌هایی که پیش‌تر آمده‌اند، در بنیان خود، شبیه‌سازی رویدادهای گستته^۱ را بیان می‌کنند. در مسئله نخست، سیستمی داریم با یک صفت^۲ و دو خدمت‌دهنده^۳ که در کنار هم به مشتریان^۴ خدمت می‌دهند. رویدادهای^۵ سیستم نیز به ورود^۶ و خروج^۷ مشتریان محدود می‌شوند. در مسئله دوم، حالت کلی مسئله پیشین را داریم: یک صفت، چندین خدمت‌دهنده موازی^۸، 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^۹، مانند درصد زمان اشتغال^{۱۰} خدمت‌دهنده‌ها، روش برتر را بیابیم. اما در مسئله دوم، صرفاً باید تعداد بهینه‌ی خدمت‌دهنده‌ها را با توجه به مقادیر داده شده برای زمان‌های میان‌هر رویداد بیابیم، به طوری که میانگین زمان انتظار کمینه^{۱۱} و درصد زمان اشتغال هر خدمت‌دهنده بیشینه^{۱۲} شود.

چارچوبی برای شبیه‌سازی رویدادهای گستته

حال، با توجه به این که (۱) هر دو مسئله، شبیه‌سازی رویدادهای گستته را در خود گنجانده‌اند و (۲) احتمالاً در تمرین‌های آینده نیز با این نوع مسائل مواجه خواهیم شد، تصمیم گرفتیم که یک چارچوب^{۱۳} و بنیان واحد برای حل آن‌ها طراحی و پیاده کنیم. این چارچوب باید ویژگی‌های زیر را داشته باشد:

¹ Discrete-Event Simulation

² Queue

³ Server

⁴ Customers

⁵ Events

⁶ Arrival

⁷ Departure

⁸ Parallel

⁹ Performance Metrics

¹⁰ Busy Time Percentage (BTP)

¹¹ Minimized

¹² Maximized

¹³ Framework

چارچوبی برای شبیه‌سازی رویدادهای گستته

(۱) گسترش‌پذیری^{۱۴}: در مسائل گوناگون با سیستم‌های سروکار داریم که متغیرها^{۱۵}، وضعیت^{۱۶}، منطق^{۱۷}، و رفتار^{۱۸}

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.

(۲) مقیاس‌پذیری^{۱۹}: مسائل شبیه‌سازی در پیچیدگی^{۲۰} و اندازه با هم تفاوت دارند. مثلاً ممکن است تعداد صفحه‌ها، انواع و

مراحل خدمت‌دهی، و تعداد خدمت‌دهنده‌ها در هر مسئله متفاوت باشند. بنابراین، چارچوب مورد استفاده باید قابلیت

به کارگیری را در مقیاس‌های مختلف داشته باشد.

(۳) بهره‌وری^{۲۱}: 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^{۲۲} 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^{۲۳} می‌شوند. کنترلر آن‌ها را به «فهرست رویدادهای آینده»^{۲۴} می‌افزاید و این فهرست را بر اساس «زمان مقرر»^{۲۵} رویدادها

مرتب می‌کند. پس از پایان شلیک^{۲۶} رویداد پیشین، رویداد پیش رو^{۲۷} از FEL خارج و پس از جلو بردن ساعت شبیه‌سازی^{۲۸}

¹⁴ Extensibility

¹⁵ Variables

¹⁶ State

¹⁷ Logic

¹⁸ Behavior

¹⁹ Scalability

²⁰ Complexity

²¹ Efficiency

²² Controller

²³ Dispatch

²⁴ Future Events List (FEL)

²⁵ Due Time

²⁶ Triggering

²⁷ Upcoming Event

²⁸ Simulation Clock

چارچوبی برای شبیه‌سازی رویدادهای گستته

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^{۲۹} و زمان پایان شبیه‌سازی^{۳۰} را دریافت و صحت‌سننجی^{۳۱} می‌کند. سنجش صحت پارامترها عبارتند از: (۱) مشبّت بودن زمان پایان شبیه‌سازی و (۲) 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 به کار می‌روند. خلاصه‌ای از پارامترها و متغیرهای مربوط به کنترلر را می‌توانید مشاهده کنید:

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

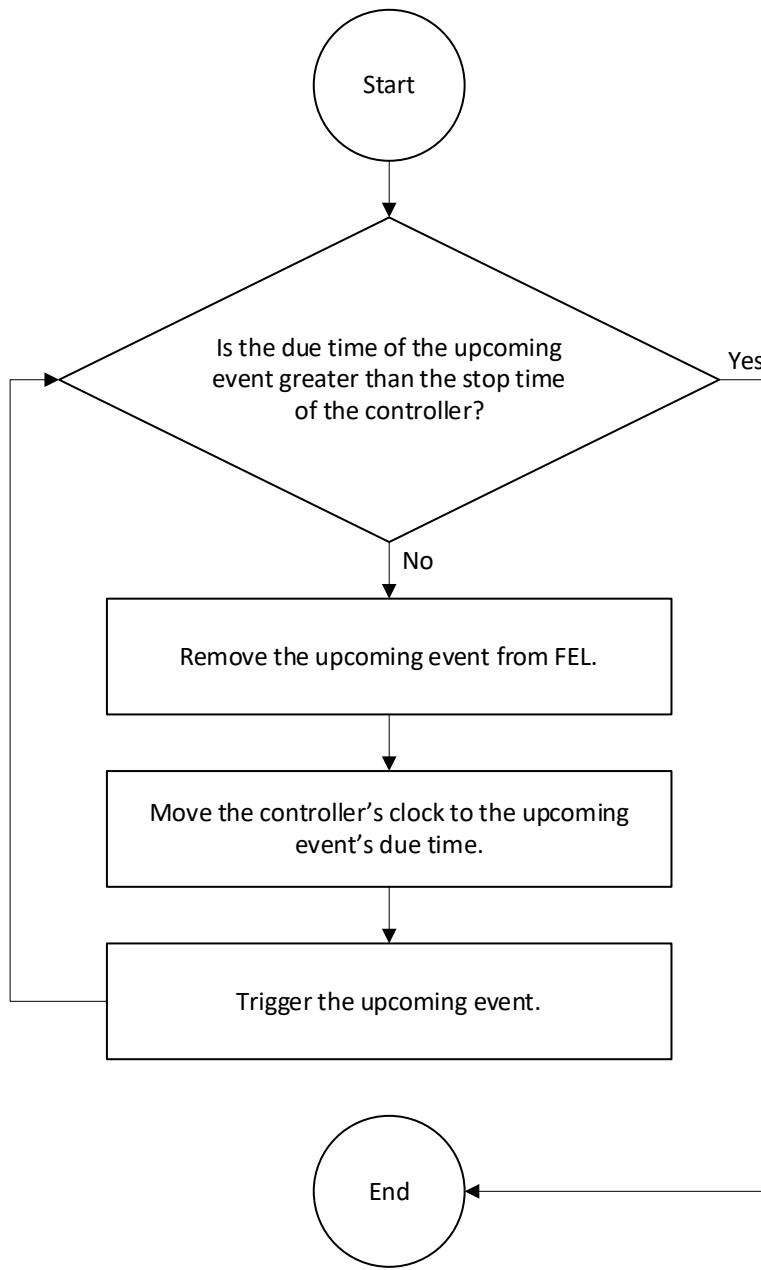
حال، کنترلر برای شروع فرآیند شبیه‌سازی آمده است. با شروع فرآیند، رویداد نخستین که به FEL افزوده شده بود را از این فهرست خارج و سپس شلیک می‌کنیم. در بی‌آن، رویدادهای بعدی دریافت شده و به FEL افزوده می‌گردند و این حلقه^{۳۲} تکرار می‌گردد، تا این که به زمان پایان شبیه‌سازی برسیم. در شکل بعدی، نمودار جریان را برای فرآیند شبیه‌سازی کنترلر می‌توانید مشاهده کنید. این نمودار همان توضیحات ساده‌ای که پیش‌تر داده شدند را با جزئیات بیش‌تر ارائه می‌دهد.

²⁹ Initial Event

³⁰ Simulation Stop Time

³¹ Validate

³² Loop



رویداد

این موجودیت بستر را برای ایجاد رویدادهای مختلف در شبیه‌سازی یک سیستم فراهم می‌کند. تنها پارامتری که دریافت می‌کند فاصله^{۳۳} زمان شلیک آن با ساعت شبیه‌سازی است و پس از آن که به کنترل 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.

^{۳۳} Interval

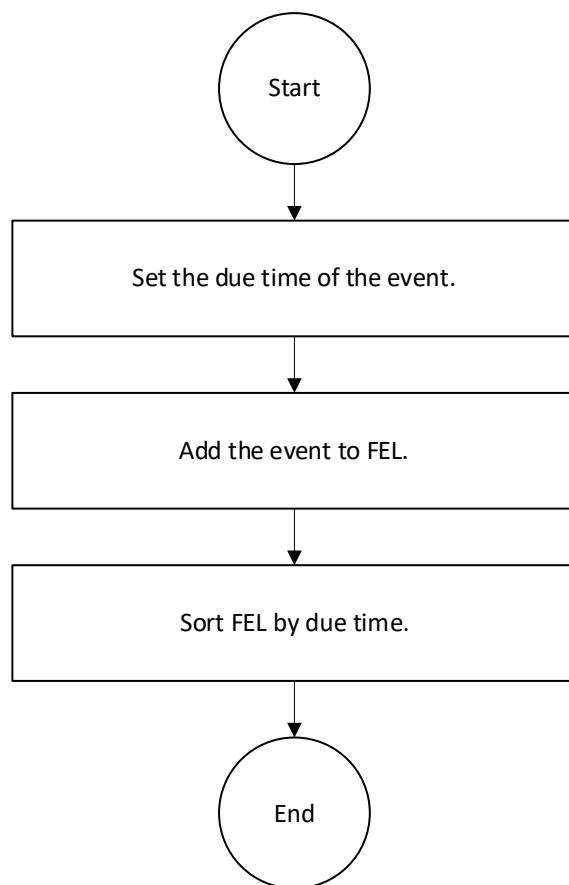
چارچوبی برای شبیه‌سازی رویدادهای گستته

$$dueTime_{event} = clock + interval_{event}$$

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

متغیرها		پارامترها	
توضیحات	نام	توضیحات	نام
زمان مقرر	dueTime	فاصله زمانی	interval

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 هر 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^{۳۴} نگاه کرده و در کنار کنترلر، رویدادهای شبیه‌سازی را نیز با ارث‌بری^{۳۵} از رویداد بنیادین پیاده می‌کنیم.

از آنجایی که سیستم نخست حالتی خاص از سیستم دوم است، می‌تواند رفتارها و منطق مربوط به شبیه‌سازی این سیستم را با اندکی تغییر به ارث ببرد. بنابراین ابتدا به حالت کلی پرداخته و سپس به سراغ حالت خاص می‌رویم.

حالت کلی: چند خدمت‌دهنده

موجودیت‌های مختص به این سیستم

این سیستم دو موجودیت مخصوص به خود را دارد. خدمت‌دهنده و مشتری. خدمت‌دهنده دارای دو متغیر است: مجموع زمان اشتغال و وضعیت که می‌تواند یکی از مقادیر «مشغول» یا «دردسترس» را بگیرد. از متغیر اول در ادامه برای محاسبه معیارهای کارایی سیستم و از دومین متغیر برای یافتن خدمت‌دهنده‌های دردسترس برای خدمت‌دهی به مشتریان استفاده می‌کنیم.

هر مشتری نیز دارای پارامترهای زمان ورود، مدت زمان دریافت خدمت، و زمان خروج است. با استفاده از این سه، همچنین می‌توان دو متغیر مربوط به زمانی که مشتری در سیستم سپری کرده و زمان انتظار وی را محاسبه کرد.

$$totalTime_{customer} = departureTime_{customer} - arrivalTime_{customer}$$

$$waitingTime_{customer} = totalTime_{customer} - serviceTime_{customer}$$

³⁴ Object-Oriented

³⁵ Inheritance

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 خدمت‌دهنده می‌تواند «اولین دردسترس»^{۳۶} یا «دردسترس تصادفی»^{۳۷} باشد. همچنین خروجی‌ها با استفاده از «فهرست مشتریان تاریخی»^{۳۸} و فهرست خدمت‌دهنده‌ها محاسبه می‌شوند:

$$ABTP = \frac{1}{|servers|} \sum_{server \in servers} \frac{totalBusyTime_{server}}{clock}$$

$$AWTP = \frac{1}{|historicalCustomers|} \sum_{customer \in historicalCustomers} \frac{waitingTime_{customer}}{totalTime_{customer}}$$

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

خرجی‌ها		متغیرها		پارامترها	
توضیحات	نام	توضیحات	نام	توضیحات	نام
میانگین درصد زمان اشتغال خدمت‌دهنده‌ها ^{۳۹}	ABTP	زمان مقرر	dueTime	تعداد خدمت‌دهنده‌ها	numberOfServers
میانگین درصد زمان انتظار مشتریان تاریخی ^{۴۱}	AWTP	فهرست مشتریان تاریخی	historicalCustomers	رویکرد انتخاب خدمت‌دهنده ^{۴۰}	serverSelectionStrategy

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

ببینید:

^{۳۶} First Available

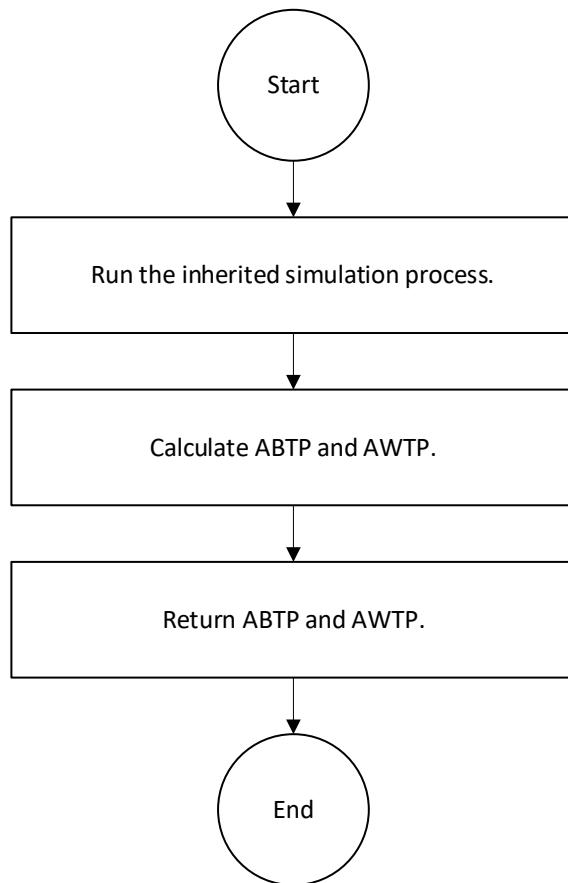
^{۳۷} Random Available

^{۳۸} Historical Customers List (HCL)

^{۳۹} Average Busy Time Percentage (ABTP)

^{۴۰} Server Selection Strategy

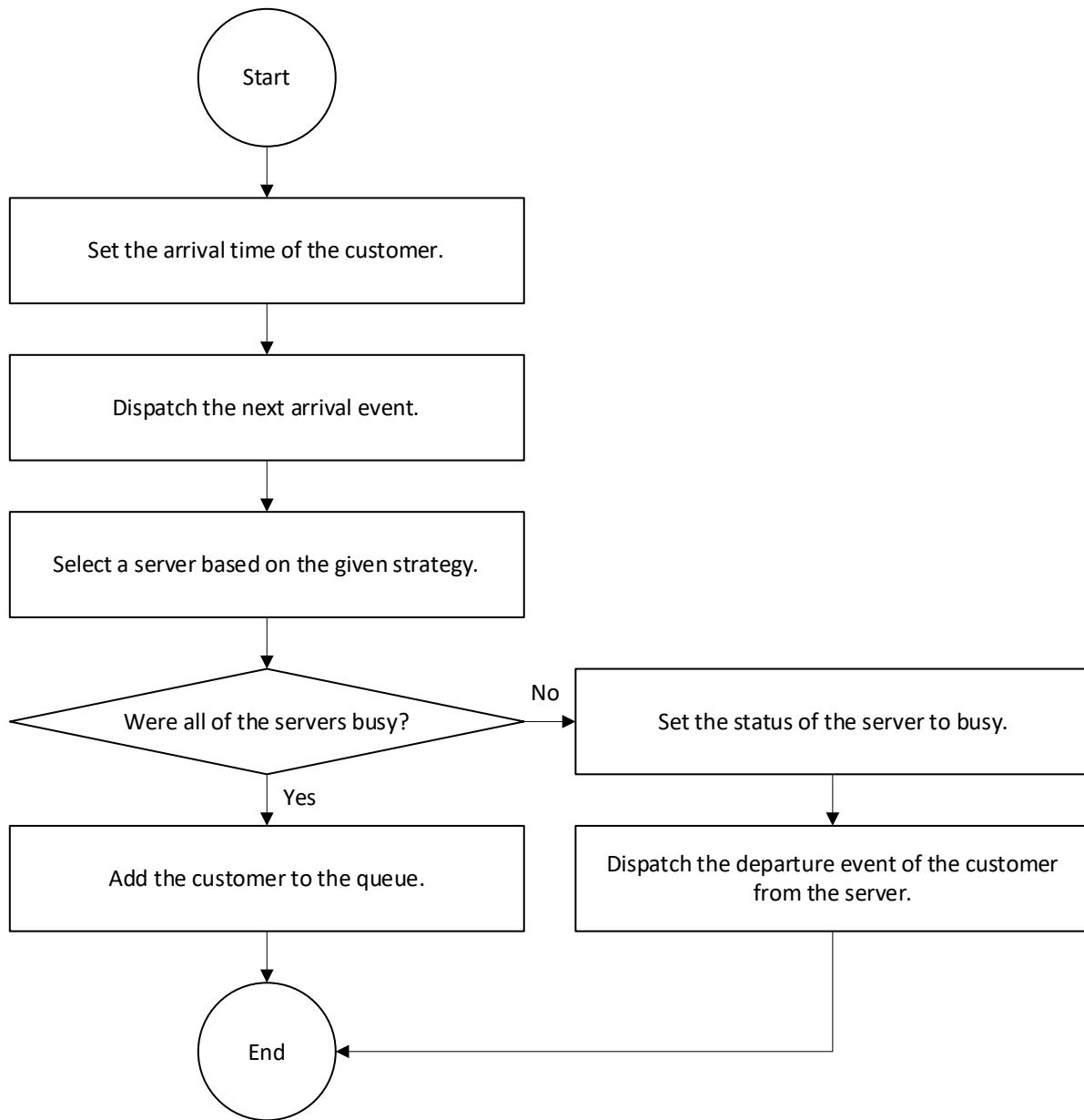
^{۴۱} Average Waiting Time Percentage (AWTP)



رویدادها

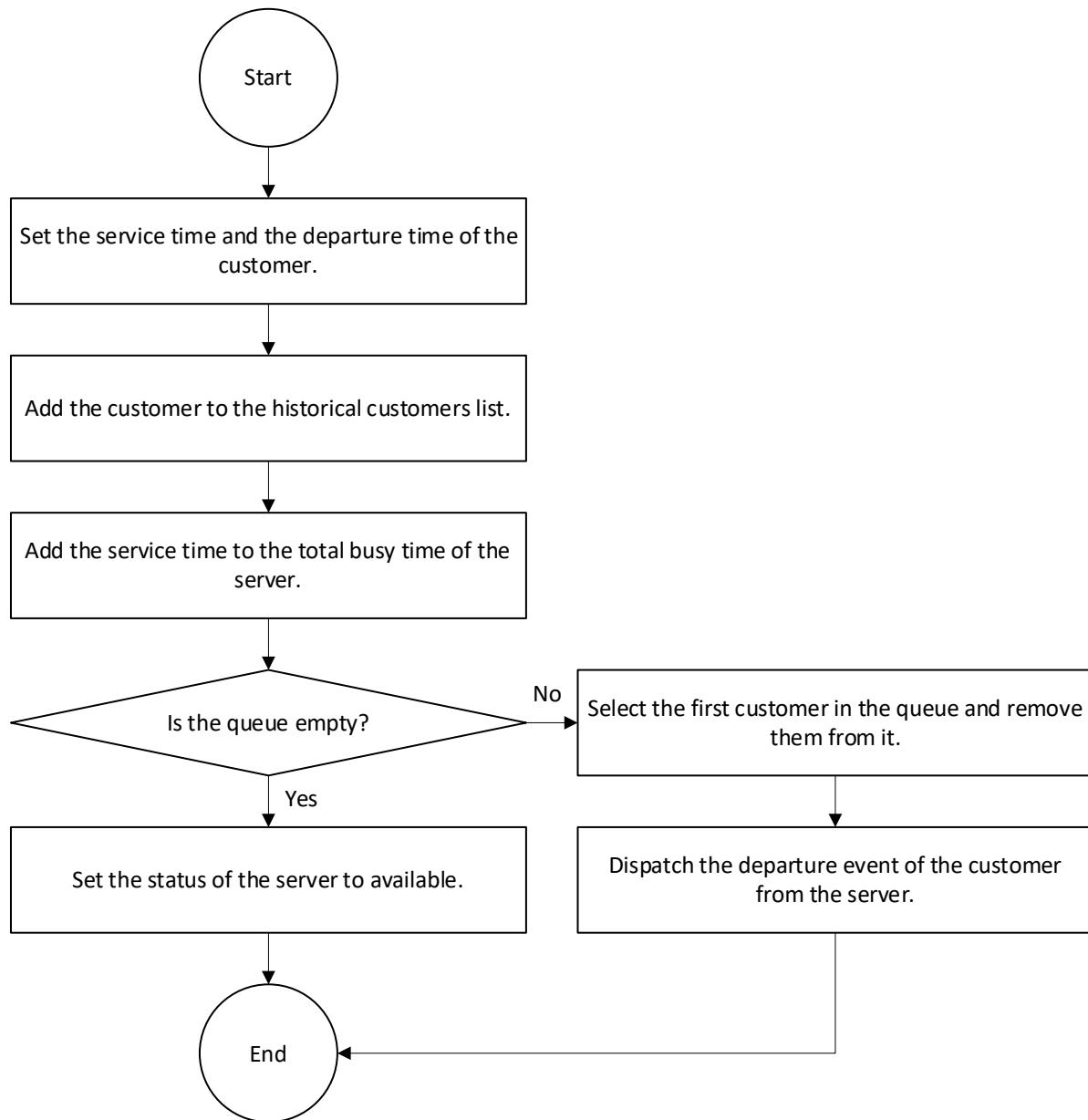
ورود

با شلیک رویداد ورود، زمان ورود مشتری مشخص و رویداد ورود بعدی نیز به کنترلر ارسال می‌شود. با استفاده از رویکرد مشخص شده 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.



حالت خاص: دو خدمت‌دهنده

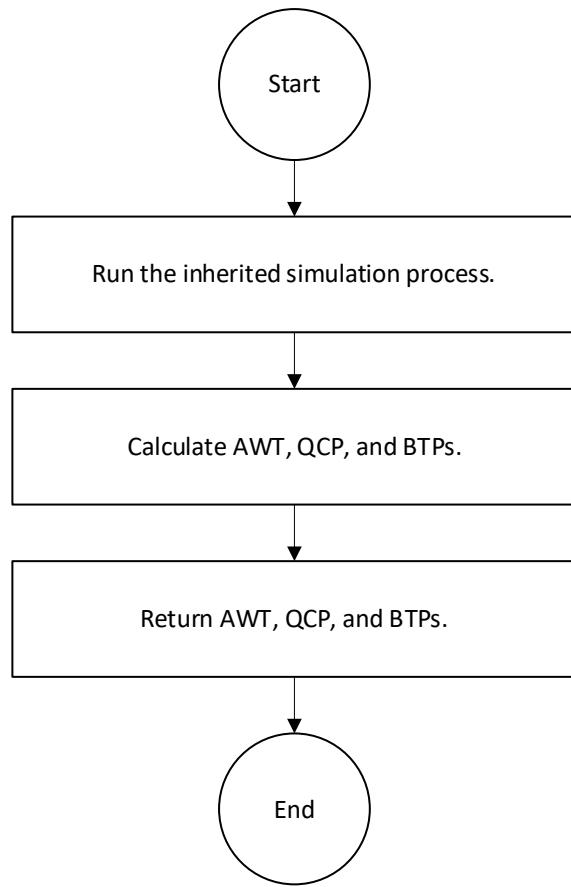
در این حالت، از کنترلر و دو رویداد حالت قبلی استفاده می‌کنیم. تنها تفاوت این است که پارامتر تعداد خدمت‌دهنده‌ها را برابر ۲ قرار داده و به جای معیارهای کارایی پیشین، از معیارهای میانگین زمان انتظار^{۴۲}، درصد مشتریانی که وارد صف انتظار شدند^{۴۳}، و

⁴² Average Waiting Time (AWT)

⁴³ Queued Customers Percentage (QCP)

نتایج

در صد زمان اشتغال هر خدمتدهنده استفاده می‌کنیم. برای محاسبه معیار دوم، تعداد مشتریان تاریخی که زمان انتظاری بزرگ‌تر از صفر دارند را بر تعداد کل مشتریان تاریخی تقسیم می‌کنیم.



نتایج

مسئله یکم

همانطور که در جدول زیر می‌توانید مشاهده کنید، با استفاده از دو رویکرد متفاوت در انتخاب خدمتدهنده برای خدمتدهی، نتایجی با شباهت بسیار بالا و اختلاف ناچیز به دست آورده‌یم.

serverSelectionStrategy	AWT	QCP	BTP _{Able}	BTP _{Baker}
firstAvailable	37.235234	0.977239	0.993042	0.984505
randomAvailable	37.625851	0.977429	0.988809	0.988666

مسئله دوم

برای بهدست آوردن تعداد بهینه خدمتدهنده‌ها، یک معیار امتیاز محاسبه کردیم که برابر با میانگین وزنی دو شاخص ABTP و AWTP می‌باشد.

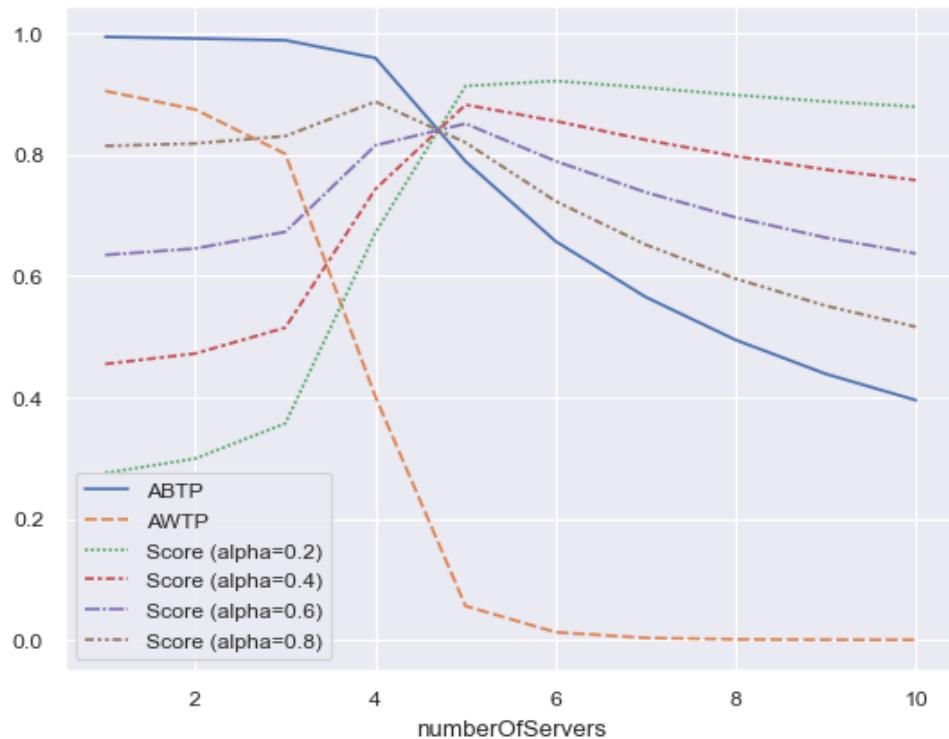
$$Score = \alpha ABTP + (1 - \alpha)(1 - AWTP)$$

با محاسبه‌ی این امتیاز می‌توانیم تعدادی از خدمتدهنده‌ها را بیابیم که میزان زمان انتظار مشتریان را کمینه و درصد به کارگیری خدمتدهنده‌ها را بیشینه می‌کند. همچنین مقادیر بیشتر آلفا، به بیشینه‌سازی درصد به کارگیری، و مقادیر کمتر آن به کمینه‌سازی زمان انتظار اولویت می‌دهند.

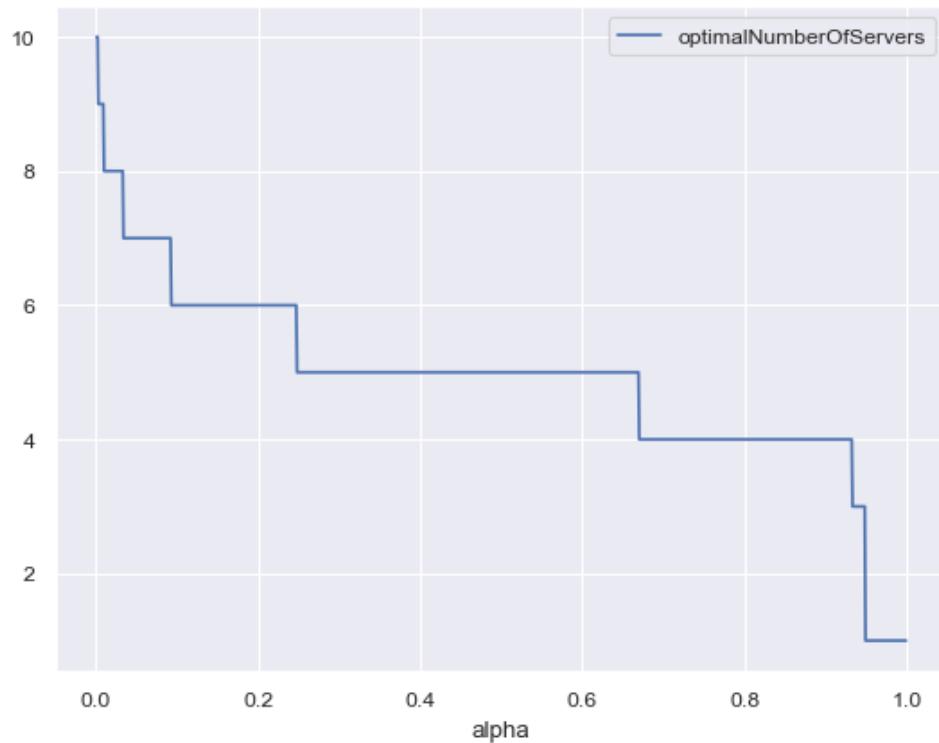
numberOfServers	ABTP	AWTP	Score (alpha=0.2)	Score (alpha=0.4)	Score (alpha=0.6)	Score (alpha=0.8)
1	0.994148	0.904932	0.274884	0.454700	0.634516	0.814332
2	0.991460	0.874347	0.298814	0.471976	0.645137	0.818299
3	0.988517	0.801101	0.356822	0.514746	0.672670	0.830593
4	0.959321	0.400762	0.671254	0.743271	0.815288	0.887305
5	0.788848	0.055736	0.913181	0.882098	0.851014	0.819931
6	0.657091	0.012480	0.921434	0.855348	0.789262	0.723177
7	0.565161	0.003114	0.910541	0.824196	0.737851	0.651506
8	0.494349	0.000686	0.898321	0.797328	0.696335	0.595342
9	0.438406	0.000137	0.887572	0.775280	0.662989	0.550697
10	0.395044	0.000018	0.878994	0.758007	0.637019	0.516032

همچنین، نمودار مربوط به این جدول را در شکل پایین می‌توانید مشاهده کنید:

نتایج



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



نتیجه‌گیری

از مسئله اول نتیجه می‌گیریم که به دلیل یکسان بودن تابع توزیع زمان خدمتدهی آریا و بردهیا، 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 به کارگیری آریا کمی بیشتر و درصد به کارگیری بردهیا کمی کمتر است. با این حال، میانگین درصد به کارگیری آریا و بردهیا 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 ۵ خدمتدهنده بیشترین بازه مقادیر آلفا را در بهینگی را به خود اختصاص داده است. بنابراین، در حالت کلی، ما ۵ خدمتدهنده را پیشنهاد می‌کنیم.

جدول گزارش عملکرد اعضای گروه

وظایف							شماره دانشجویی	نام
ویرایش	نگارش	طراحی	پیاده‌سازی	پیاده‌سازی کنترلر	پیاده‌سازی کنترلر	طراحی و پیاده‌سازی چارچوب		
گزارش	گزارش	نمودارها	رویدادها	مسئله ۲	مسئله ۱		۹۹۲۵۰۰۳	حامد اعرب
		●	●	●		●	۹۹۲۵۰۱۵	شهریار خلوتی