



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

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

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 به‌کارگیری (درصد زمان اشتغال) آریا و بردیا را در هر حالت تخمین بزنید.
- (۴) میانگین زمان انتظار هر مشتری را در هر حالت به‌دست 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 به‌دست 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 گسسته^۱ را بیان می‌کنند. در مسئله نخست، سیستمی داریم با یک صف^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 صف، 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 داده‌شده برای زمان‌های میان هر رویداد بیابیم، به طوری که میانگین زمان انتظار کمینه^{۱۱} و درصد زمان اشتغال هر خدمت‌دهنده بیشینه^{۱۲} شود.

چارچوبی برای شبیه‌سازی رویدادهای گسسته

حال، با توجه به این که (۱) هر دو مسئله، شبیه‌سازی رویدادهای گسسته 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 گسسته

(۱) گسترش‌پذیری^{۱۴}: در مسائل گوناگون با سیستم‌هایی 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 صف‌ها، انواع و مراحل خدمت‌دهی، و تعداد خدمت‌دهنده‌ها در هر مسئله کم و یا زیاد باشند. بنابراین، چارچوب مورد استفاده باید قابلیت به‌کارگیری 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 کنیم و از صحت خروجی‌ها و درستی مدل شبیه‌سازی‌مان اطمینان حاصل کنیم.

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 به اقتضای 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، و 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 گسسته

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 صحت‌سنجی^{۳۱} می‌کند. سنجش صحت پارامترها عبارتند از مثبت بودن زمان پایان شبیه‌سازی و صفر بودن زمان مقرر 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:

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

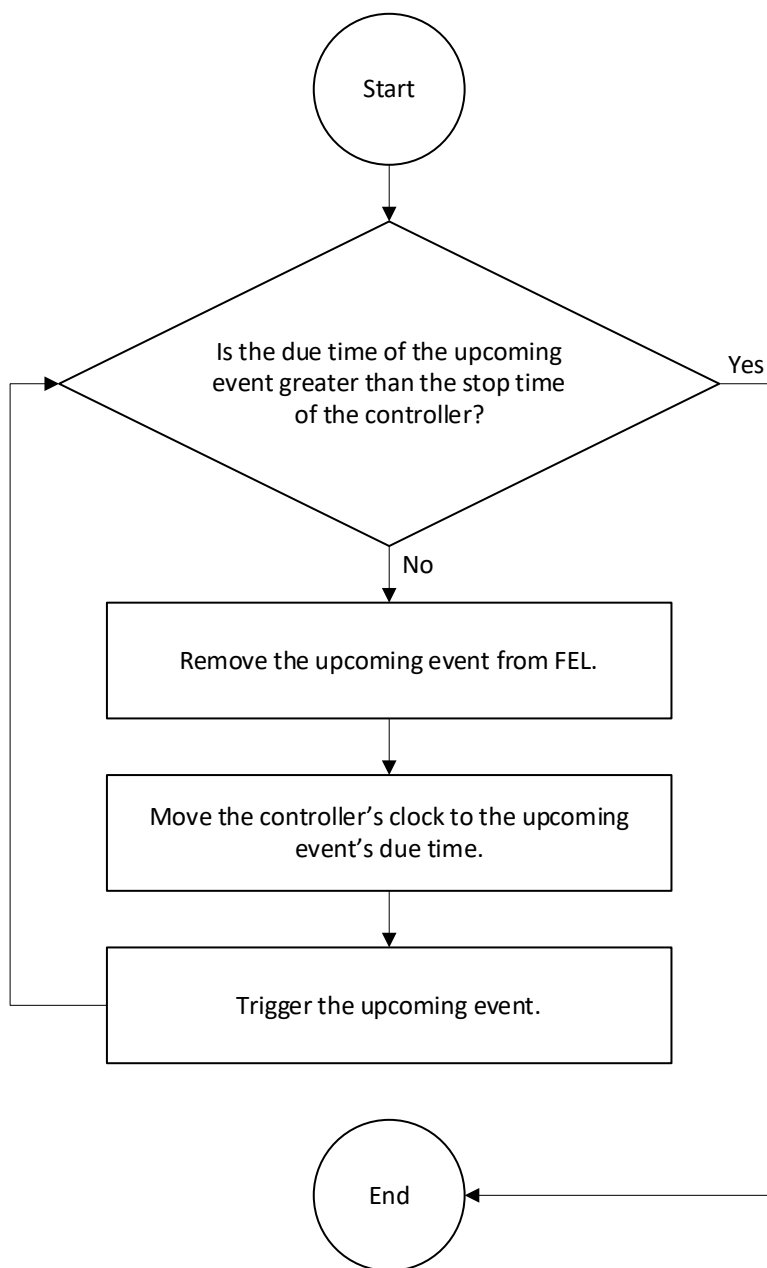
حال، کنترلر برای شروع فرآیند شبیه‌سازی آماده است. با شروع فرآیند، رویداد نخستین که به FEL افزوده شده بود را از این فهرست خارج و سپس شلیک می‌کنیم. در پی آن، رویدادهای بعدی دریافت شده و به FEL افزوده می‌گردند و این حلقه^{۳۲} تکرار می‌گردد، تا این که به زمان پایان شبیه‌سازی برسیم. در شکل بعدی، نمودار جریان 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 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.

³³ Interval

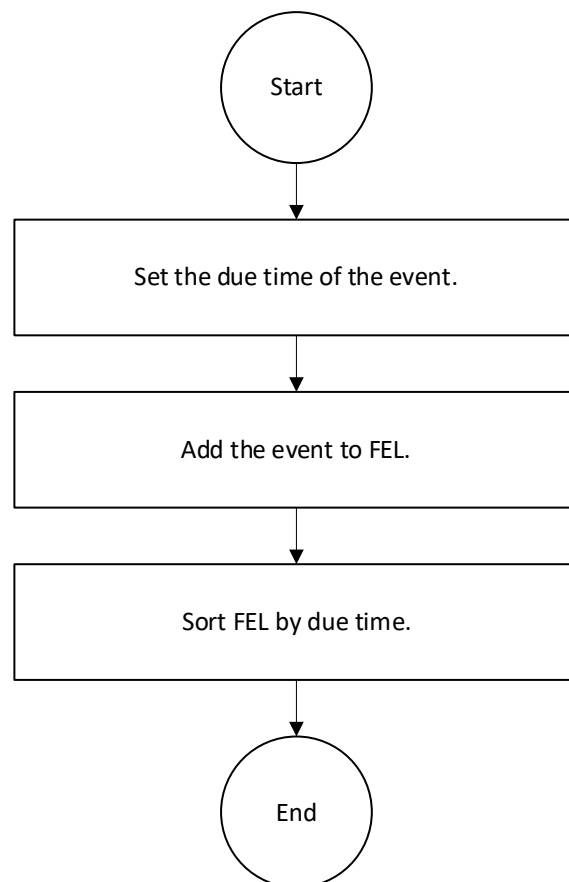
چارچوبی برای شبیه‌سازی رویدادهای گسسته

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

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

متغیرها		پارامترها	
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.

مدل سازی و حل مسئله‌ها

برای حل هر مسئله با استفاده از چارچوب پیشنهادی، کنترلری طراحی می‌شود که رفتارهای بنیادین و پایه‌ای را از کنترلر چارچوب به ارث 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 شبیه سازی را از چارچوب به ارث می برد. اما نیاز است تا پارامترها، متغیرها، و خروجی های شبیه سازی مختص به این مسئله را به آن اضافه کنیم. رویکرد انتخاب خدمت دهنده می تواند «اولین در دسترس^{۳۶}» یا «در دسترس تصادفی^{۳۷}» باشد. همچنین خروجی ها با استفاده از «فهرست مشتریان تاریخی^{۳۸}» و فهرست خدمت دهنده ها محاسبه می شوند:

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

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

خلاصه ای از این بخش را در جدول زیر می توانید مشاهده کنید:

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

در نهایت، در شکل زیر، نمودار جریان فرآیند شبیه سازی کنترلر شبیه سازی سیستم چند خدمت دهی موازی دارای یک صف را ببینید:

³⁶ First Available

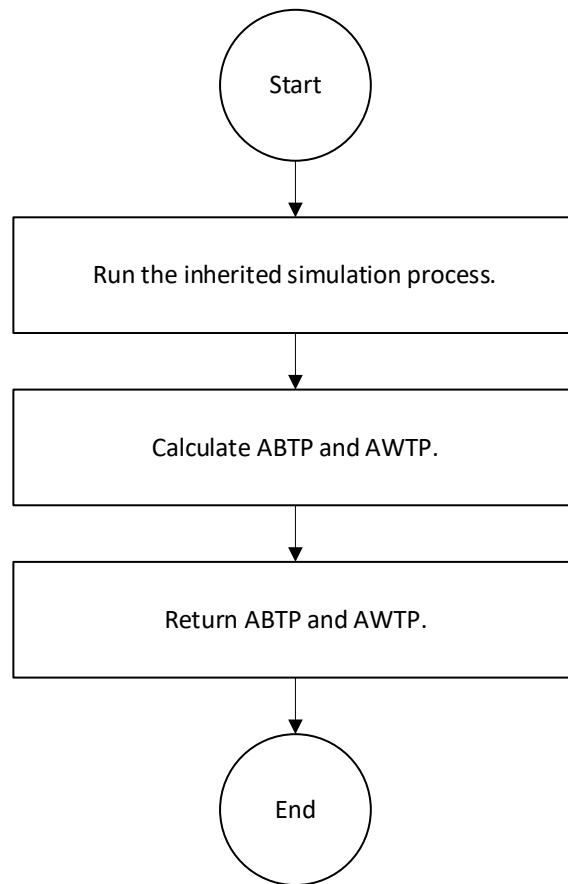
³⁷ Random Available

³⁸ Historical Customers List (HCL)

³⁹ Average Busy Time Percentage (ABTP)

⁴⁰ Server Selection Strategy

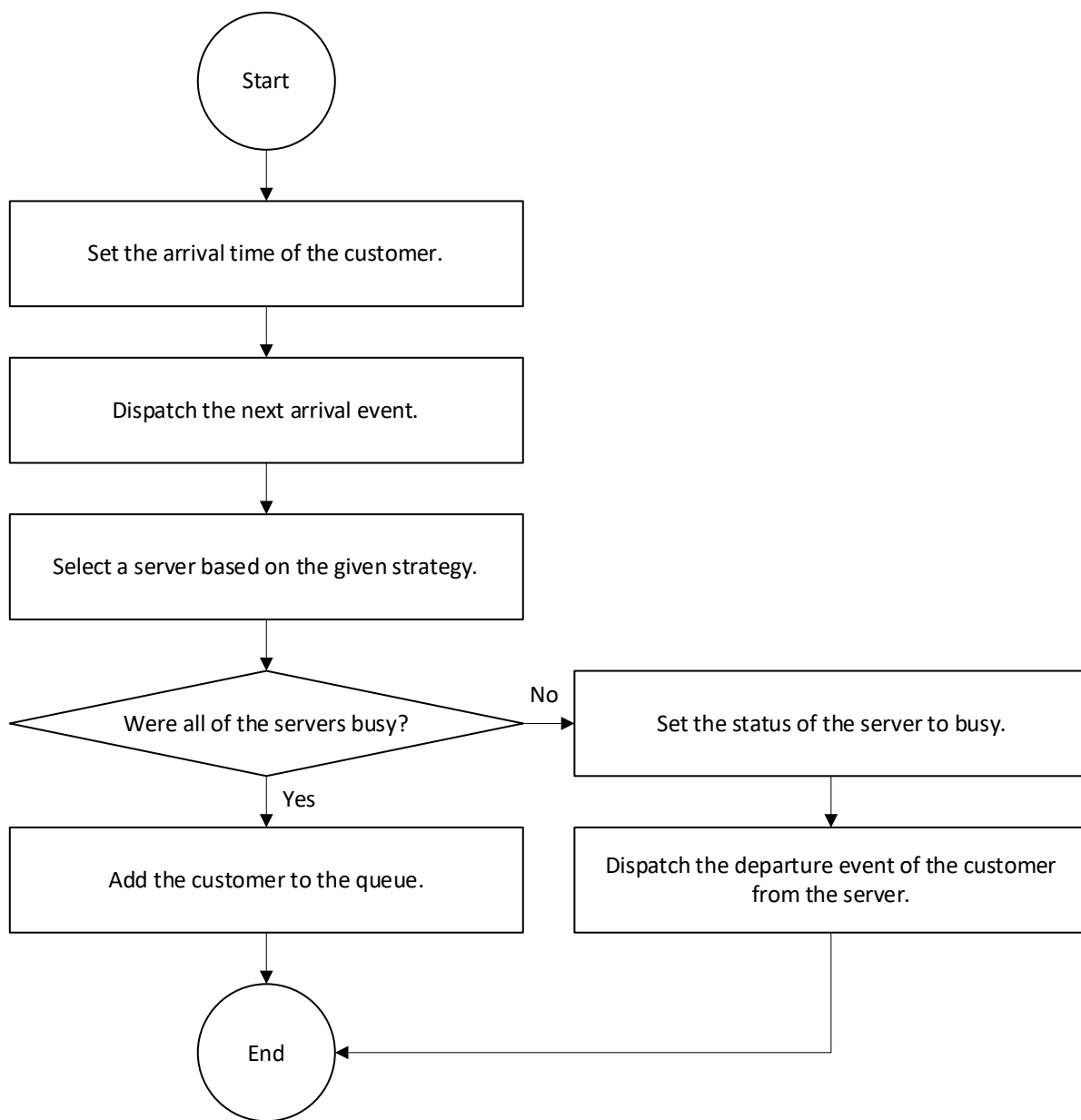
⁴¹ Average Waiting Time Percentage (AWTP)



رویدادها

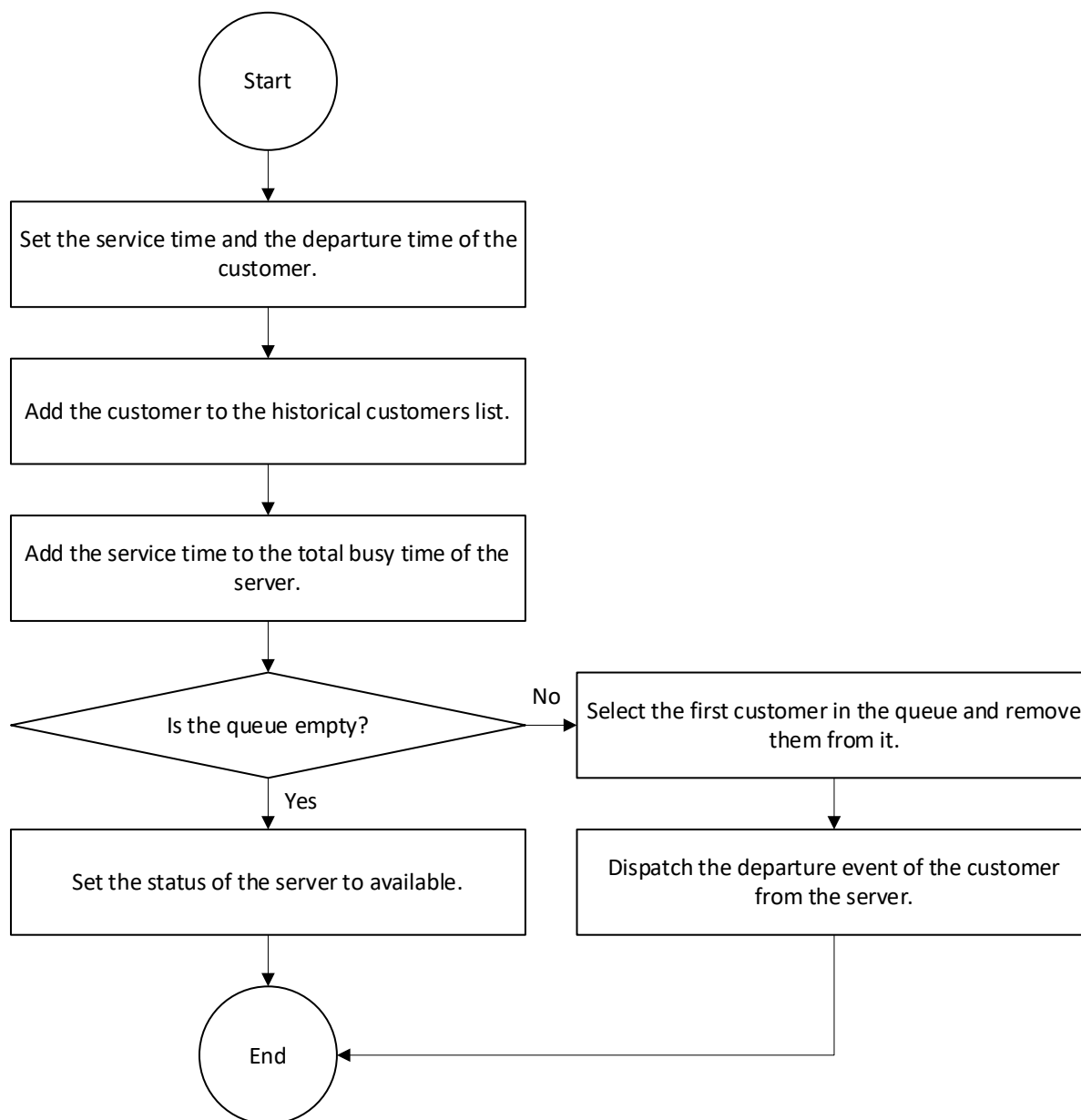
ورود

با شلیک رویداد ورود، زمان ورود مشتری مشخص می‌گردد. 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 در دسترس نبود، مشتری را به صف اضافه می‌کنیم. در غیر این صورت، وضعیت خدمت‌دهنده را به «مشغول» تغییر داده و رویداد خروج مشتری 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 را به کنترلر ارسال می‌کنیم. در نمودار پایین، جزئیات جریان این رویداد را می‌توانید ببینید.



حالت خاص: دو خدمت‌دهنده

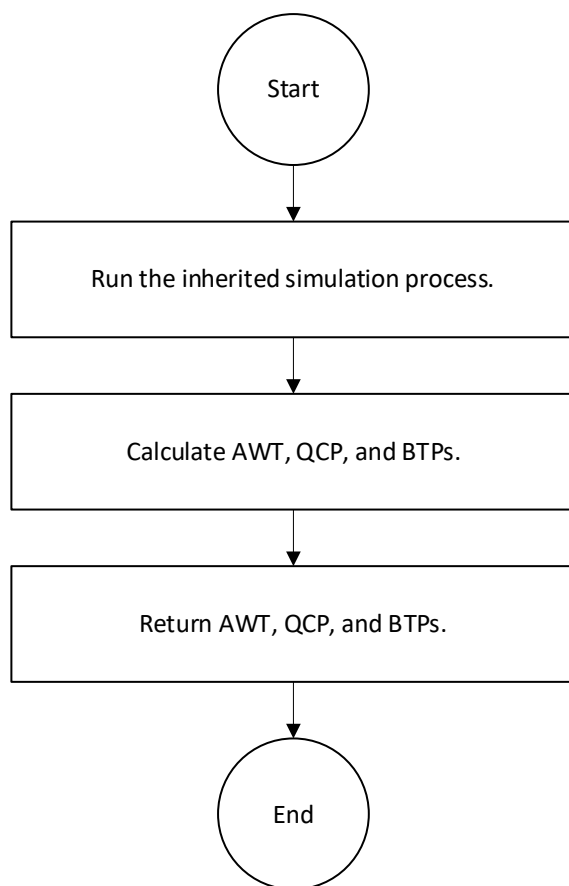
در این حالت، کنترلر و دو رویداد حالت قبلی را استفاده می‌کنیم. تنها تفاوت این است که پارامتر تعداد خدمت‌دهنده‌ها را برابر ۲ قرار داده، و به جای معیارهای کارایی پیشین، از معیارهای میانگین زمان انتظار^{۴۲}، درصد مشتریانی که وارد صف انتظار شدند^{۴۳}، و درصد زمان اشتغال هر خدمت‌دهنده استفاده می‌کنیم. برای محاسبه معیار اول، از زمان‌های انتظار بزرگ‌تر از صفر میانگین گرفته و

^{۴۲} Average Waiting Time (AWT)

^{۴۳} Queued Customers Percentage (QCP)

نتایج

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 به‌دست آوردیم.

serverSelectionStrategy	AWT	QCP	BTP _{Able}	BTP _{Baker}
firstAvailable	38.289973	0.977452	0.992514	0.984589
randomAvailable	38.260221	0.976934	0.988687	0.988736

مسئله دوم

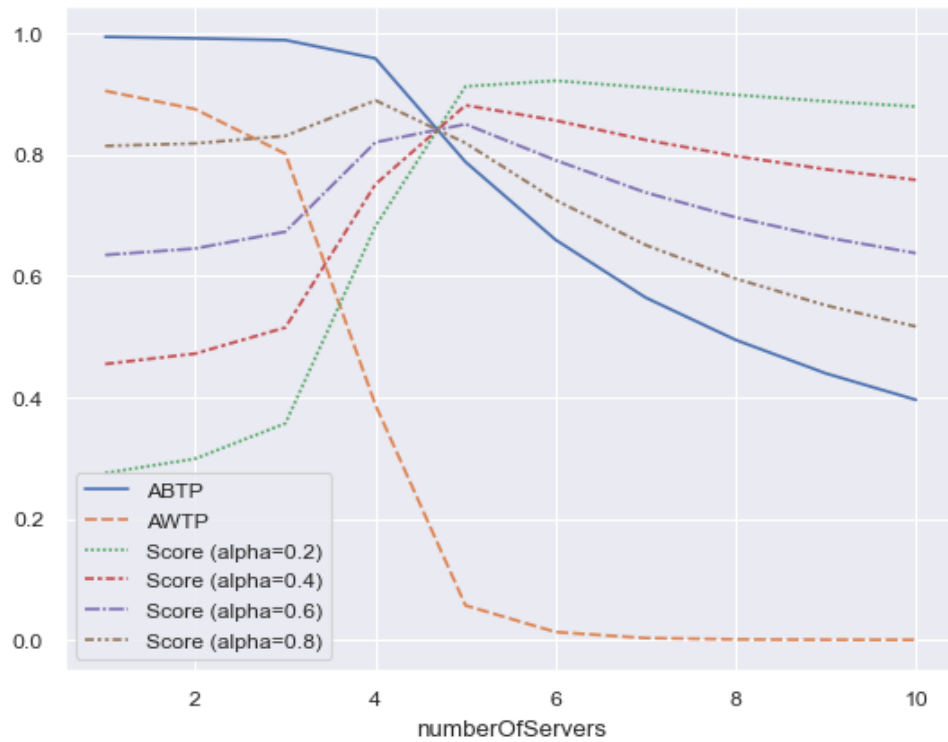
برای به دست آوردن تعداد بهینه‌ی خدمت‌دهنده‌ها، یک معیار امتیاز محاسبه کردیم که برابر با میانگین وزنی دو شاخص ABTP و AWTP - 1 می‌باشد.

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

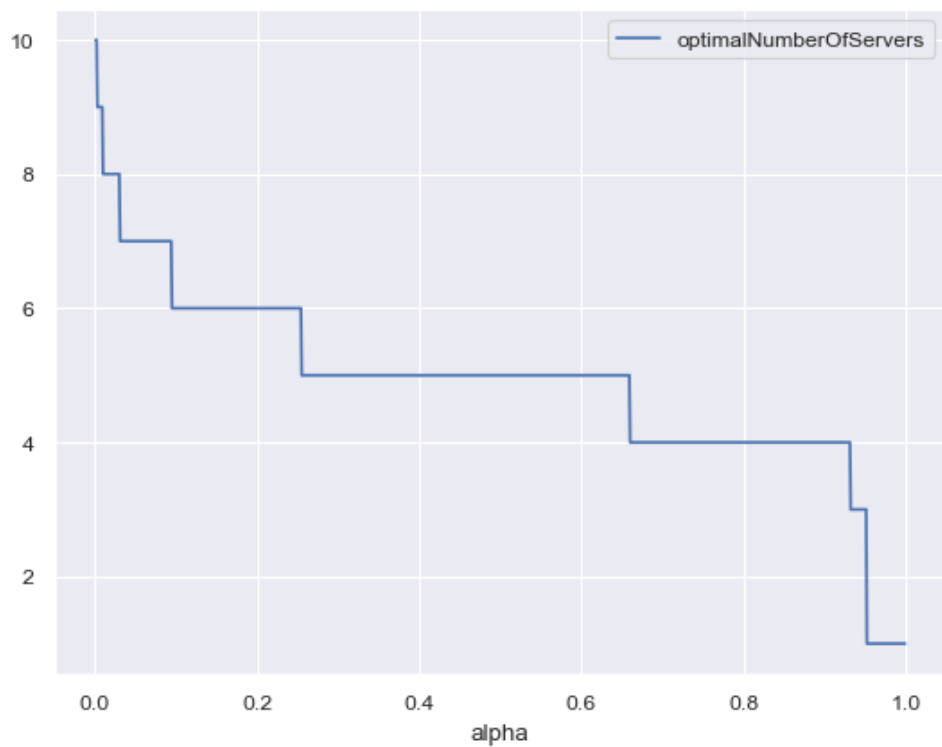
با محاسبه‌ی این امتیاز می‌توانیم تعدادی از خدمت‌دهنده‌ها را بیاییم که میزان زمان انتظار مشتریان را کمینه و درصد به کارگیری خدمت‌دهنده‌ها را بیشینه می‌کند. همچنین مقادیر بیشتر آلفا، به بیشینه‌سازی درصد به کارگیری، و مقادیر کمتر آن به کمینه‌سازی زمان انتظار اولویت می‌دهند.

numberOfServers	ABTP	AWTP	Score (alpha=0.2)	Score (alpha=0.4)	Score (alpha=0.6)	Score (alpha=0.8)
1	0.993950	0.904884	0.274883	0.454649	0.634416	0.814183
2	0.991524	0.874705	0.298541	0.471786	0.645032	0.818278
3	0.988641	0.801172	0.356790	0.514753	0.672716	0.830678
4	0.958405	0.387283	0.681855	0.750992	0.820130	0.889267
5	0.787742	0.056613	0.912258	0.881129	0.850000	0.818871
6	0.659381	0.012741	0.921684	0.856108	0.790532	0.724957
7	0.564128	0.002841	0.910553	0.823947	0.737341	0.650734
8	0.494319	0.000661	0.898335	0.797331	0.696327	0.595323
9	0.439193	0.000138	0.887728	0.775595	0.663461	0.551327
10	0.395576	0.000022	0.879097	0.758217	0.637337	0.516456

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



حال، اگر امتیاز را با مقادیر مختلف آلفا محاسبه کنیم، خواهیم داشت:



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 مسئله 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. بنابراین، در حالت کلی، ما ۵ خدمت‌دهنده را پیشنهاد می‌کنیم.

جدول گزارش عملکرد اعضای گروه