مقدمه

شبیه سازی در حال حاضر یکی از پرکاربردترین ابزارهای موجود تحقیق در عملیات است و برای اقشار مختلفی همچون مهندسان، مدیران، اساتید یا متخصصین علوم مختلف کاربرد دارد.

با استفاده از شبیه سازی میشود پیش از بوجود آمدن سیستم واقعی، عملکرد اجرایی آن را ارزیابی نمود و گزینههای اجرایی متعددی را بدون ایجاد خلل در کارکرد سیستم واقعی آزمایش کرد. همچنین به دلیل در دسترس بودن زبان های کامپیوتری ویژه شبیه سازی، به راحتی می توانیم از آن استفاده کنیم. اما باید توجه کرد که مانند سایر ابزارها، استفاده ی نادرست از شبیه سازی و اعتماد بیش از حد به نتایج به دست آمده از آن می تواند باعث ایجاد خسارات جبران ناپذیری گردد.

شبیه سازی کامپیوتر خصوصیاتی دارد که آن را برای تحلیل گران به صورت ابزار جالبی درآورده است.

با شبیه سازی کامپیوتری می توان زمان را فشرده کنیم به شکلی که فعالیت های چند سال ظرف چند دقیقه و گاهی چند ثانیه شبیه سازی شود. با استفاده از این امتیاز تحلیلگر می تواند طرح های متنوعی با صرف زمان ناچیزی اجرا کند و ارزیابی هایی از آنها بدست آورد. با استفاده از شبیه سازی کامپیوتری می توان زمان را بسط هم بدهیم. در حقیقت با استفاده از این امکان تحلیلگر می تواند با توجه به داده هایی که تولید و مشاهده می شوند، شناخت قابل توجهی از تغییرات ساختاری سیستم بدست بیاورد که دست یافتن به چنین شناختی براساس زمان واقعی میسر نیست.

در هنگام انجام آزمایش های واقعی معمولاً خطاهای مختلفی اجتناب ناپذیر است، زیرا هیچ ابزار سنجشی کامل و بدون خطا در دنیا واقعی وجود ندارد. اما در شبیه سازی کامپیوتری چنین خطاهایی وجود نداشته، زیرا مدل شبیه سازی یعنی برنامه کامپیوتری اعدادی را تولید می کند که بدون خطا هستند. امتیاز دیگر شبیه سازی کامپیوتری قابلیت اجرای مدل به طور مکرر و تحت شرایط شروع یکسان است. به این صورت که می توان در اجراهای مختلفی از یک مدل شبیه سازی، پارامترهای ثابت، داده های متفاوتی را به عنوان ورودی آماده کرد و نتایج هر اجرا را بررسی کرد. علاوه بر این می توان با ثابت نگهداشتن مجموعه داده های ورودی پارامترهای مدل را تغییر داده و شبیه سازی را تکرار کنیم.

شاید مهمترین مزیت شبیه سازی این باشد که می توان آن را در هر لحظه متوقف کرده و بدون تاثیر بر ادامه آزمایش نتایج به دست آمده را بررسی و آزمایش را ادامه دهیم. این امکان تقریباً در هیچ آزمایش دنیای واقعی امکان پذیر نمی باشد.

درس شبیه سازی در رشته های مختلفی مثل مهندسی صنایع، مدیریت، ریاضیات کاربردی و کامپیوتر در مقطع کارشناسی تدریس می شود. برای گذراندن این درس دانشجو نیاز به آشنایی مختصری با حساب دیفرانسیل و انتگرال، نظریه احتمال و آمار مقدماتی دارد. به این دلیل که این درس دارای پایه های ریاضی و آماری است نیاز زیادی به ارائه مثال های متعدد وجود دارد و همچنین برای دانشجویان رشته ی کامپیوتر پیاده سازی مثال ها با یکی از زبان های برنامه نویسی شدیداً توصیه می شود. کتابی که برای این درس توسط وزارت علوم به عنوان مرجع اصلی درس انتخاب شده کتاب Discrete- Event System Simulation تالیف بنکس و کارسون است که توسط آقای هاشم محلوجی به نام "شبیه سازی سیستم های گسسته- پیشامد" ترجمه شده و انتشارات علمی دانشگاه صنعتی شریف آن را به چاپ رسانده است. این کتاب شامل چهار قسمت است.

قسمت اول: مقدمه ای بر شبیه سازی سیستم های گسسته پیشامد

۱)

۱- مقدمه ای بر شبیه سازی، ۲- مزایا و معایب شبیه سازی

۳- زمینه های کاربرد، ۴- سیستم های گسسته و پیوسته

۲)

مثال هایی از شبیه سازی

۱- شبیه سازی سیستم های صف، ۲- شبیه سازی سیستم های موجودی

۳)

شبیه سازی گسسته پیشامد

۱- اصول کلی، ۲- مفاهیم مربوط به شبیه سازی

۳- زبان های برنامه نویسی برای شبیه سازی

قسمت دوم: مدل های ریاضی و آماری

۴)

مدل های آماری در شبیه سازی

۱- توزیع های گسسته، ۲- توزیع های پیوسته

۵)

مدل های صف

۱- ویژگی های سیستم های صف، ۲- نمادگذاری سیستم های صف

۶)

سیستم های موجودی

۱- سیستم های قطعی، ۲- سیستم های احتمالی

قسمت سوم: اعداد تصادفی

۷) تولید اعداد تصادفی

۱- خواص اعداد تصادفی ۲- تولید اعداد شبه تصادفی

۸)تولید مقادیر تصادفی

قسمت چهارم: تحلیل داده های شبیه سازی

۹)

تجزیه تحلیل داده های ورودی به مدل

۱- گردآوری داده ها ۲- برآورد پارامترها
۱۰)
آزمایش مدلهای شبیه سازی و تعیین اعتبار آنها
۱۱)
تجزیه تحلیل نتایج بدست امده از یک مدل شبیه سازی
۱۲)
مقایسه و ارزیابی طرحهای مختلف از سیستم.

فصل اول

می توان شبیه سازی را تقلیدی از عملکرد فرآیند یا سیستم واقعی با گذشت زمان تعریف نمود. هدف از شبیه سازی ایجاد ساختگی تاریخچه ی سیستم و بررسی آن به منظور دستیابی به نتیجه گیری هایی در مورد ویژگی های عملکرد سیستم واقعی است.

معمولاً برای بررسی رفتار یک سیستم، مدلی برای آن تهیه می شود. این مدل معمولاً به شکل مجموعه ای از فرضهای مربوط به عملکرد سیستم است. این فرضها در چارچوب روابط ریاضی، منطقی و نمادین بین موجودیت ها یا اهداف مورد نظر طراح سیستم بیان می شوند. اکثراً بعد از ایجاد چنین مدلی آن را معتبرسازی کرده و برای بررسی پرسش های مختلفی از نوع "اگر تغییر X انجام شود چه نتیجه ای دارد؟" بکار می برند.

علاوه بر این از شبیه سازی برای بررسی سیستم های در دست طراحی نیز استفاده می شود. پس می توانیم بگوئیم که شبیه سازی، هم ابزاری برای پیش بینی تاثیر تغییرات در سیستم های موجود و هم ابزاری برای طراحی سیستم های جدید است. در ادامه راجع به مزایا و معایب شبیه سازی، کاربردهای آن و موارد استفاده از این ابزار صحبت خواهیم کرد.

در این قسمت هفت مورد از مواردی که شبیه سازی در آن ها کاربرد دارد را معرفی می کنیم. اولین کاربرد شبیه سازی بررسی و آزمایش روابط متقابل هر سیستم یا زیر سیستم پیچیده است. کاربرد دوم شبیه سازی مشاهده تاثیر تغییرات اطلاعاتی، سازمانی و محیطی بر رفتار مدل های موجود است.

با توجه به اینکه شبیه سازی و ارائه مدلی از سیستم باعث ایجاد شناخت از آن سیستم می شود و سومین کاربرد برای شبیه سازی ارائه پیشنهادهای مفید برای انجام اصلاحات در سیستم مورد مطالعه می باشد.

یکی از مزایای ایجاد مدل های شبیه سازی قابلیت ایجاد تغییر در ورودی های شبیه سازی و بررسی خروجی های بدست آمده از آنهاست. این قابلیت منجر به چهارمین کاربرد شبیه سازی یعنی شناخت متغیرها و چگونگی رابطه متقابل آنها شده است.

پنجمین کاربرد شبیه سازی را می توانیم قابلیت استفاده از آن به عنوان ابزاری آموزشی برای بهبود روش های تحلیلی پاسخ یابی بدانیم.

به عنوان کاربرد دیگر می تواند شبیه سازی را برای آزمایش طرح ها یا خط مشی های جدید، پیش از اجرای آنها استفاده نمود و به این وسیله آمادگی لازم را برای روبرو شدن با پیشامدهای ممکن به دست آورد. آخرین کاربرد شبیه سازی را می توان تحقیق درباره پاسخ های تحلیلی سیستم دانست.

هر چند شبیه سازی ابزار مناسبی برای تحلیل در بسیاری از موارد است اما پیش از بکارگیری این ابزار، تحلیلگر باید مزایا و معایب آن را هم در نظر داشته باشد.

مزایای اصلی شبیه سازی که در منابع مختلف آورده شده است شامل این موارد می باشد:

۱ - مدیر سیستم می تواند به منظور تحلیل طرح ها یا خطی مشی های پیشنهادی، پس از ساختن هر مدلی بارها و بارها شبیه سازی را بکار گیرد.

۲ - در مواردی که داده های ورودی، تقریبی یا ناقص باشند، باز هم می توان از شبیه سازی برای تحلیل سیستم استفاده نمود.

۳ - دسترسی به داده های شبیه سازی معمولاً بسیارکم هزینه تر از فراهم آوردن داده های مربوط به سیستم واقعی است.

۴ - به کار بردن روش های شبیه سازی در بیشتر موارد بسیار ساده تر از روش های تحلیلی است. و همین سادگی یکی از دلایل استفاده ی بیشتر شبیه سازی نسبت به روش های تحلیلی است.

۵ - معمولاً محدودیت هایی که در روش های تحلیلی وجود دارند در شبیه سازی کمتر به چشم می‌خورد. بنابراین در شبیه سازی نیاز کمتری به فرض های ساده کننده داریم.

۶ - علاوه بر تمامی موارد گفته شده، شبیه سازی در برخی از مسائل تنها وسیله یافتن راه حل مساله است بنابراین ناچاریم از شبیه سازی استفاده کنیم.

به این باید توجه کنیم که شبیه سازی معایبی هم دارد. در ادمه به برخی عیب های شبیه سازی که باید پیش از بکارگیری آن بررسی شود، اشاره می شود.

۱ - به دلیل پر هزینه بودن و نیاز به زمان، استفاده از مدل های شبیه سازی کامپیوتری، ممکن است کارایی بالایی نداشته باشد.

۲ - نیاز به اجراهای فراوان برای هر مدل شبیه سازی منجر به افزایش هزینه استفاده از مدل شبیه سازی می شود.

۳ - کاربرانی که تازه با شبیه سازی آشنا می شوند، ممکن است برای کاربردهایی که روش های تحلیلی و ریاضی کافی به نظر می رسد، از شبیه سازی استفاده بکنند و هزینه زیادی برای آن بپردازند.

اما با توجه به پیشرفت تکنولوژی و بوجود آمدن سخت افزار و نرم افزارهای سریع، همچنین وجود زبان های مخصوص شبیه سازی دو مورد اول امروزه کمتر به چشم می‌خورد.

زمینه های کاربرد

برای شبیه سازی در زمینه های مختلفی کاربرد وجود دارد که هر کدام بسته به مورد استفاده می تواند مقادیر زیادی در هزینه ها صرفه جویی کند. از جمله مهمترین این کاربردها می توان به موارد زیر اشاره کرد:

۱ - شبیه سازی مراحل مختلفی که در فرودگاه های بزرگ اتفاق می افتد. مثلاً کنترل ترافیک، ظرفیت و نگهداری و تعمیر، امکانات حمل و نقل مسافر و ...

۲ - شبیه سازی حمل و نقل در سطح شهرها برای کنترل بهینه چراغ های راهنمایی و زمانبندی آنها.

۳ - شبیه سازی عملیات نگهداری و تعمیر برای دستیابی به مجموعهی بهینه ای از افراد در گروه های تعمیراتی.

۴ - شبیه سازی مراحل مختلف موجود در صنایع برای ارزیابی تغییرات در طرز انجام عملیات و جابجایی مراحل مختلف آن.

۵ - شبیه سازی سیستم های اقتصادی در سطح کلان، به منظور پیش بینی تأثیر تصمیم های مختلف در جامعه.

۶ - شبیه سازی جنگ های نظامی

البته کاربردهای متعدد دیگری هم وجود دارد که به عنوان مثال می توان به مواردی مثل کاربرد شبیه سازی در سیستم های ارتباطی، خطوط تولید، بنگاه های تجاری، سیستم های برقی و ... اشاره کرد.

سیستم و اجزای آن

پیش از مدلسازی باید درک درستی از مفهوم سیستم و مرز سیستم داشته باشیم. به عنوان یک تعریف می توانیم سیستم را "گروهی از اشیاء که برای رسیدن به هدف خاصی در یک رابطه یا با وابستگی متقابل منظم به هم پیوسته باشند" تعریف کنیم. مثلاً سیستم تولیدی خودرو را در نظر بگیرید. در این سیستم، ماشین ها، قطعات و کارگران با هم و در امتداد خط مونتاژ برای تولید یک وسیله کار می کنند.

محیط بیرونی یک سیستم را پیرامون آن سیستم می گویند و تغییراتی که در اطراف سیستم اتفاق می افتد را تغییرات در پیرامون سیستم می نامند. در مدلسازی سیستم باید بتوانیم مرز بین سیستم و پیرامون سیستم را مشخص کنیم. در حقیقت این مهارتی است که با تجربه ی فرد مدلساز بدست می آید.

به عنوان مثال در مورد سیستم کارخانه می توان عوامل کنترل کننده ی ورود سفارشها را خارج از اختیار کارخانه بدانیم. بنابراین با این دیدگاه این روابط و عوامل در پیرامون سیستم جای خواهند داشت، اما اگر بخواهیم تاثیر عرضه بر تقاضا یعنی رابطه ی بین ورود اجناس تولیدی کارخانه بر نیازهای آتی افراد به همان اجناس را هم در نظر بگیریم، آن وقت این عوامل جزء خود سیستم باید تلقی بشوند.

به عنوان مثالی دیگر سیستم بیمه را در نظر بگیریم. معمولاً در بیمه نرخ محاسبه ی یبمه برای هر مورد مشخص است، اگر این حد را فقط در بررسی یک شرکت بیمه در نظر بگیریم مسلماً یک عامل پیرامونی خواهد بود اما در صورتی که بخواهیم تاثیر چنین قوانینی را بر صنعت بیمه بسنجیم، مسلماً این عامل یکی از عوامل موجود در خود سیستم خواهد بود.

برای درک بهتر سیستم باید تعاریفی برای موارد زیر عنوان کنیم:

- نهاد: عنصری مورد توجه در سیستم
- خصیصه: ویژگی هر نهاد
- فعالیت: دوره ی زمانی با طول مشخص
- حالت: مجموعه متغیرهای لازم برای تشریح سیستم در هر زمان
- در هر سیستم پیشامد را به عنوان رویدادی لحظه ای که بتواند حالت سیستم را تغییر دهد، تعریف می کنیم.

- به فعالیت ها و پیشامدهایی که درون سیستم اتفاق می افتد درون زا و به پیشامدهایی که پیرامون سیستم اتفاق می افتد، برون زا می گوییم.

به عنوان مثال در مورد سیستم بانک، می توان مشتریان را نهاد در نظر گرفت، موجودی حساب های آنها را خصیصه و سپرده گذاری را فعالیت در نظر گرفت. متغیرهای ممکن حالت عبارتند از تعداد کارمندان در حال کار یا تعداد مشتریان منتظر در صف. ورود هر مشتری پیشامدی برون زا و تکمیل کارمشتری پیشامدی درون زا می باشد.

به عنان مثالی دیگر می توان قطار را به عنوان یک سیستم در نظر گرفت، که مسافران نهاد، مبدأ و مقصد سفر آنها خصیصه، سفر به عنوان فعالیت تلقی می شود. ورود به ایستگاه و رسیدن به مقصد پیشامدهای سیستم و تعداد مسافران منتظر در هر ایستگاه متغیرهای حالت سیستم را تشکیل می دهند.

سیستم دیگری که می توان مورد بررسی قرارداد سیستم انتقال پیام هاست. در این سیستم پیامها نهادهای موجود در سیستم، طول و مقصد آنها خصیصه ها و انتقال پیام فعالیت ها را تشکیل می دهند. رسیدن یک پیام به مقصد یک پیشامد است و تعداد پیام های منتظر در صف، متغیر حالت سیستم است.

سیستم های گسسته و پیوسته

سیستم ها را می توان به دو دستهی گسسته و پیوسته تقسیم بندی کرد. سیستم گسسته سیستمی است که متغیر یا متغیرهای حالت در آن تنها در مجموعه ای از نقاط گسسته زمان تغییر کند. مثلاً بانک، سیستم گسسته است، زیرا متغیر حالت که تعداد مشتری حاضر در بانک است وقتی تغییر می کند که مشتری وارد یا از بانک خارج می شود.

سیستم پیوسته سیستمی است که متغیرهای حالت در آن به صورت پیوسته در طی زمان تغییر می کند. به عنوان مثال، میزان آب جمع شده در یک سد سیستمی پیوسته است زیرا به صورت پیوسته ای پس از هر بارندگی آب به آن اضافه شده و با باز شدن دریچه سد یا تبخیر از آن کاسته می شود.

مدل سیستم

برای پی بردن به روابط بین اجزای سیستم باید بتوانیم آن سیستم را بررسی کنیم. اما همیشه این امکان وجود ندارد. مثلاً ممکن است هنوز سیستم در مرحله طراحی می باشد و بوجود نیامده باشد. یا حتی اگر سیستم موجود باشد ممکن است ایجاد تغییر در آن ممکن نباشد. مثلاً در مورد سیستم بانک امکان نصف کردن تعداد کارمندان برای بررسی اثر آن بر طول صف انتظار وجود ندارد. در چنین مواردی معمولاً به جای خود سیستم، مدل سیستم را بررسی می کنند. در حقیقت مدل معرف سیستم است و چون در بیشتر بررسی ها در نظرگرفتن همه جزئیات سیستم لازم نیست، مدل را بدون آن جزئیات می سازند. بنابراین علاوه براینکه مدل معرف سیستم است یک ساده سازی از آن هم هست عمل ساده کردن مدل معمولاً به یکی از راه های زیر انجام می شود:

۱- تبدیل متغیرها به مقادیر ثابت

۲- حذف متغیرها یا ادغام آنها در یکدیگر

۳- خطی فرض کردن روابط

۴- افزودن محدودیت های بیشتر

انواع مدل ها

مدلها را معمولاً به دو دسته ی مدل های ریاضی یا فیزیکی دسته بندی می کنند. در مدل ریاضی ازنمادها و معادله های ریاضی استفاده می شود و مدل شبیه سازی هم نوعی خاص از مدل ریاضی سیستم است. مدل های شبیه سازی را می توان در مدل های ایستا یا پویا، قطعی یا تصادفی و گسسته یا پیوسته رده بندی کرد.

مدل ایستا که به آن شبیه سازی مونت کارلو هم می گویند معرف سیستم در لحظه ای خاص از زمان است. درنقطه مقابل مدل های پویا، سیستم را با توجه به تغییر زمان معرفی می کنند، مثلاً شبیه سازی بانک از ۸ صبح تا ۲ بعدازظهر. مدل های شبیه سازی که در آنها متغیر تصادفی وجود ندارد را مدل قطعی می گویند. در این مدل ها در صورت حضور ورودی ها به صورت قطعی خروجی سیستم مشخص می شود. مثل سیستم یک مطب دندانپزشکی که در آن بیماران در زمان های از پیش تعیین شده وارد می شوند. اما در مدل هایی که یک یا چند متغیر تصادفی به منزله ورودی وجود دارند، خروجی نیز ماهیت تصادفی دارد. مثل سیستم بانک که در آن مدت زمان انجام کار هر مشتری و تعداد مشتری های وارد شده به صورت تصادفی است. در این گونه مدل ها معیارهای خروجی سیستم برآوردهایی آماری از ویژگی های واقعی سیستم است.

به مدل هایی که متغیر حالت آنها تنها در مجموعه ای از مقاطع گسسته زمان تغییر می کنند، مدل گسسته و به مدل هایی که متغیر حالت آنها پیوسته است، مدل پیوسته می گویند. مدل های گسسته را بیشتر با استفاده از روش های عددی حل می کنند و مدل های پیوسته را با استفاده از روش های تحلیلی. به این دلیل که بیشتر بحثمان در ادامه کلاس راجع به مدل سازی گسسته است، جا دارد که قبل از ادامه بحث مثالی از شبیه سازی پیوسته را بررسی کنیم.

مدل صید و صیاد

مدلهای صید و صیاد (یا میهمان و میزبان) در زیست شناسی توسط نویسندگان زیادی مورد بررسی قرار گرفته است. محیطی را در نظر بگیرید که از جمعیت صیاد و جمعیت صید تشکیل می شود و این دو جمعیت با یکدیگر ارتباط دارد. طبیعت ارتباط مورد بحث چنین است که جمعیت صید منبع غذایی جمعیت صیاد شمرده می شود. برای مثال جمعیت صیاد ممکن است از کوسه ها تشکیل شود و ماهیان کوچک محیط نیز جمعیت صید را به وجود آوردند. چنین فرض کنید که تعداد جمعیت صیاد و تعداد جمعیت صید در لحظهt ، به ترتیب با $x(t), y(t)$ نمادگذاری شود. علاوه براین، فرض کنید که جمیعت صید از منبع غذایی کافی برخوردار است و در صورت عدم وجود جمعیت صیاد می تواند با آهنگ رشد $rx(t)$ توسعه یابد.

ثابت r $(r > 0)$ را می توان به صورت مابه التفاوت دو آهنگ زاد و میر طبیعی تعبیر کرد. چون دو جمعیت صیاد و صید با هم در ارتباط هستند، منطقی است اگر فرض کنیم که آهنگ مرگ و میر ناشی از وجود چنین ارتباطی برای جمعیت صید با حاصلضرب اندازه دو جمعیت، یعنی $x(t)y(t)$ نسبت مستقیم دارد. بدین ترتیب، اگر a ضریب ثابت و مثبتی باشد، آهنگ کلی تغییر در جمعیت صید (dx/dt) به طریق زیر تعریف می شود:

$$dx/dt = rx(t) - ax(t)y(t)$$

چون جمعیت صیاد برای بقای خود به جمعیت صید متکی است، در صورت عدم وجود جمعیت صید آهنگ تغییر جمعیت صیاد $-sy(t)$ می شود $(s>0)$. به علاوه، ارتباط موجود بین دو جمعیت باعث می شود که آهنگ افزایش جمعیت صید نیز با $x(t)y(t)$ نسبت مستقیم داشته باشد. بنابراین، آهنگ کلی تغییر در جمعیت صیاد نیز به ازای $b>0$ به شرح زیر تعریف می شود:

$$dy/dt = -sy(t) + bx(t)y(t)$$

اگر شرایط شروع به صورت $y(0)>0, x(0)>0$ تعریف شود، نتیجه حل مدل شامل دو معادله بالا ناظر به صدق روابط $y(t)>0, x(t)>0$ به ازای همه مقادیر t خواهد بود. به این ترتیب، جمعیت صید هیچ گاه توسط جمعیت صیاد منقرض نمی شود. نتیجه که در قالب مجموعه $\{x(t), y(t)\}$ مشخص می شود، تابعی متناوب از زمان است. به بیان دیگر، مثبتی مانند T وجود دارد به طوری که به ازای مقادیر $1,2,...,n$ ، روابط $x(t+nT)=x(t)$ و $y(t+nT)=y(t)$ برقرار است. حصول چنین نتیجه ای نامنتظره نیست. هر گاه جمعیت صیاد رو به افزایش گذارد، جمعیت صید رو به کاهش می گذارد. کاهش یافتن جمعیت صید باعث کند شدن آهنگ افزایش جمعیت صیاد می شود و این به نوبه خود جمعیت صیاد را کاهش می دهد و جمعیت صید را بالا می برد .

حل عددی دو معادله بالا به ازای ورودی های $r=$ ۰/۰۰۱، $a=2\times10^{-6}$، $s=$۰/۰۱، $y(0)=600, x(0)=12000, b=10^{-6}$ در شکل ۱-۳ نشان داده شده است.

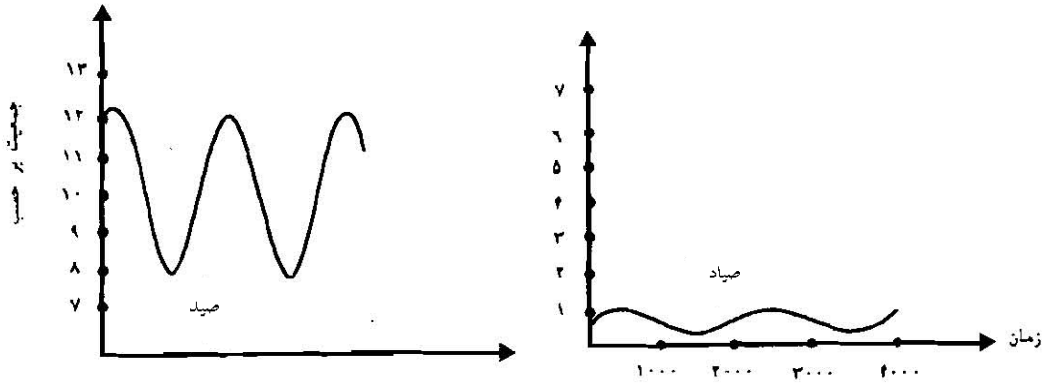
گام های اساسی در شبیه سازی

فرآیند ساخت مدل شبیه سازی را می توان به چهار مرحله تقسیم کرد. مرحله اول شامل صورت بندی مساله و تعیین اهداف و طرح کلی پروژه می باشد. صورت اولیه مساله معمولاً مبهم بوده و اهداف اولیه معمولاً باید دوره تعیین شود. همچنین طرح اولیه پروژه نیز باید مجدداً تنظیم شود. این کارها را می توان در همین مرحله یا مراحل دیگر انجام داد.

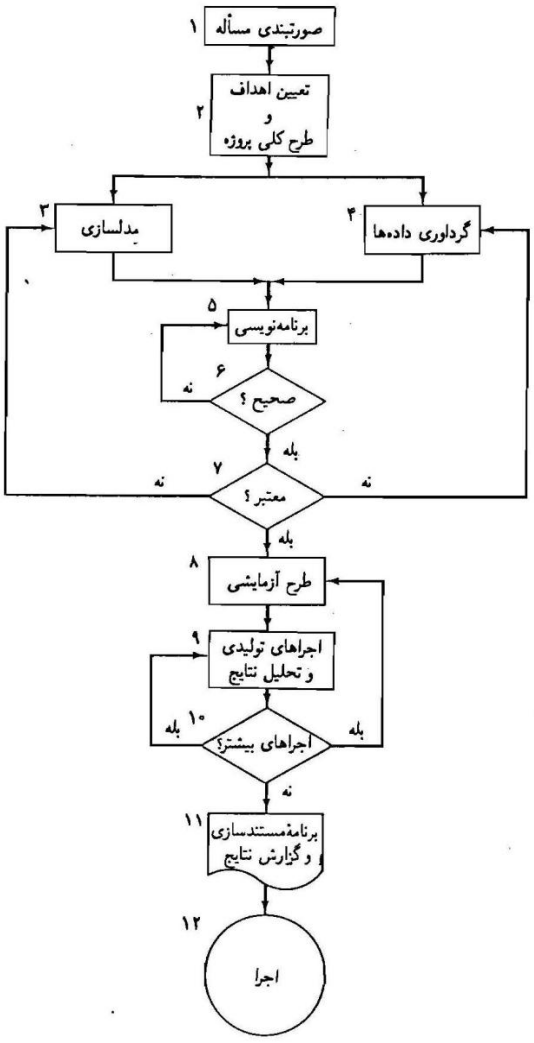
مرحله دوم شامل گام های گردآوری داده ها، مدل سازی، برنامه نویسی، وارسی برنامه و معتبرسازی آن می باشد. در این مرحله ارتباط بین کاربر و مدل ساز باید همیشه وجود داشته باشد.

مرحله سوم که به اجرای مدل مربوط است، شامل ایجاد طرح آزمایشی، اجرای مدل و تحلیل نتایج و اجراهای متعدد آن می باشد.

مرحله چهارم شامل مستند سازی برنامه و گزارش نتایج و بکارگیری مدل می باشد.



شکل ۱-۳ حل عددی مدل صید و صیاد



شکل ۱-۴ گامهای اساسی در بررسی مبتنی بر شبیه‌سازی.

فصل دوم

مثال هایی از شبیه سازی

هدف این فصل ارائه مثالهای متعددی از شبیه سازی است که مستقیماً، یعنی بدون استفاده از کامپیوتر قابل اجرا باشد. مثال هایی از این قبیل شناخت مناسبی از روش شبیه سازی سیستم های گسسته و تجزیه و تحلیل آن در اختیار دانشجو قرار می دهد . با ارائه مثال های شبیه سازی در این مرحله از شروع کتاب، دانشجو ارزش بسیاری از نکات ظریفی را که در فصل های بعد ارائه می شوند خواهد دانست. شبیه سازیهای این فصل با برداشتن سه گام زیر انجام می شود:

۱. ویژگیهای هر یک از ورودیهای شبیه سازی را تعیین کنید. در اکثر موارد، این گونه ویژگی ها را می توان در قالب توزیع های پیوسته یا گسسته احتمال مدلسازی کرد.

۲. یک جدول شبیه سازی ایجاد کنید. هر مساله شبیه سازی جدول شبیه سازی خاص خود را دارد، زیرا هر جدول برای هدف خاصی ایجاد می شود. جدول ۲-۱ مثالی از جدول شبیه سازی است . در این مثال تعداد ورودی های x_{ij} ، مساویp است، یعنی $j = 1,2,...,p$ و به ازای هر تکرار، $i = 1,2,...,n$ یک پاسخ، y_i وجود دارد.

۳. در نوبت i ام تکرار، مقداری برای هر یک از p ورودی تولید و تابع محاسبه کننده مقدار پاسخ y_i را ارزیابی کنید. این گام با نمونه گیری از توزیع های تعیین شده در گام ۱ اجرا می شود.

شبیه سازی ابزاری نیرومند است که می توان آن را به منظور تحلیل بسیاری از مسائل پیچیده به کار برد. اما، پیش از آنکه شبیه سازی به عنوان راه حل برگزیده شود، باید هر کوشش ممکن برای حل ریاضی مساله احیاناً با مدلهای ریاضی به عمل آید. ساختن مدل شبیه سازی ممکن است عملی وقتگیر باشد و اگر راه حل دیگری موجود باشد، ممکن است بسیار کم هزینه تر از شبیه سازی باشد.

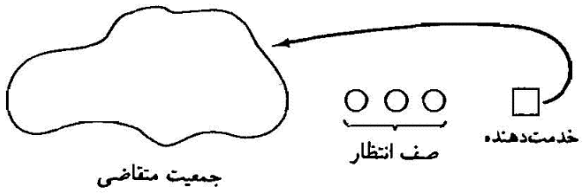
دفعات تکرار	ورودیها					پاسخ (y_i)
	x_{i1}	x_{i2}	$\cdots$ x_{ij} $\cdots$		x_{ip}	
۱						
۲						
۳						
⋮						
n						

جدول۲-۱ جدول شبیه سازی

در این فصل، مثال های بسیار درباره شبیه سازی ارائه می کنیم.

۲-۱ شبیه سازی سیستم های صف

سیستم صف با جمعیت متقاضی، چگونگی ورود وخدمتدهی، ظرفیت سیستم و نظام صف مشخص می شود، این ویژگیهای سیستم صف را به تفصیل در فصل های بعدی شرح داده‌ایم. یک سیستم ساده صف در شکل ۲-۱ نمایش داده شده است.



شکل ۲-۱ سیستم صف

در این سیستم، جمعیت متقاضی نامحدود است، یعنی، اگر یک نفر، جمعیت متقاضی را ترک کند و به صف انتظار ملحق شود یا به محل دریافت خدمت برود، هیچ گونه تغییری در آهنگ ورود سایر متقاضیان نیازمند خدمت روی نخواهد داد. به علاوه، در این سیستم، ورودها هر بار یکی و آن نیز به صورت تصادفی رخ می دهد و اگر واردشدگان به صف انتظار ملحق شوند، سرانجام خدمت دریافت خواهند کرد. در ضمن، مدتهای خدمتدهی تصادفی است و در قالب توزیع احتمالی تعیین می شوند که با گذشت زمان بدون تغییر می ماند . ظرفیت سیستم نیز نامحدود است (سیستم، واحد در حال دریافت خدمت و آنهایی که در صف انتظارند را در بر می گیرد) سرانجام، متقاضیان به ترتیب ورود (اغلب مشهور به FIFO) از یک خدمت دهنده یا مجرا خدمت می گیرند .

ورودها و خدمت دهیها با توزیع های مدت بین دو ورود و مدتهای خدمتدهی مشخص می شوند. به طور کلی، آهنگ موثر ورود باید از ماکسیمم آهنگ خدمتدهی کمتر باشد و گرنه طول صف انتظار به طور نامحدود افزایش می یابد. هر گاه صفها به طور نامحدود رشد کنند آنها را انفجار آمیز یا ناپایدار می نامند

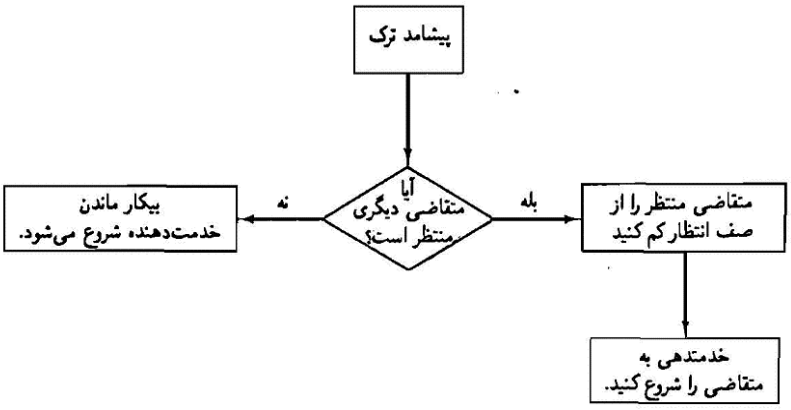
پیش از معرفی چند شبیه سازی از سیستم های صف، درک مفاهیم حالت سیستم، پیشامدها و ساعت شبیه سازی لازم است. حالت سیستم، تعداد حاضران در سیستم و وضعیت خدمت دهنده از لحاظ مشغول بودن یا بیکار بودن است. پیشامد، مجموعه شرایطی است که موجب تغییری لحظه ای در حالت سیستم می شود. در مساله تک مجرایی صف، تنها دو پیشامند ممکن است حالت سیستم را تغییر دهد.

این دو پیشامد، ورود یک واحد (پیشامد ورود) و پیشامند تکمیل خدمتدهی به یک واحد(پیشامد ترک) است. سیستم صف در برگیرنده خدمت دهنده، واحد در حال خدمتگیری (اگر واحدی در حال خدمتگیری باشد) و آحاد حاظر در صف (اگر در صف واحدی باشد) است.

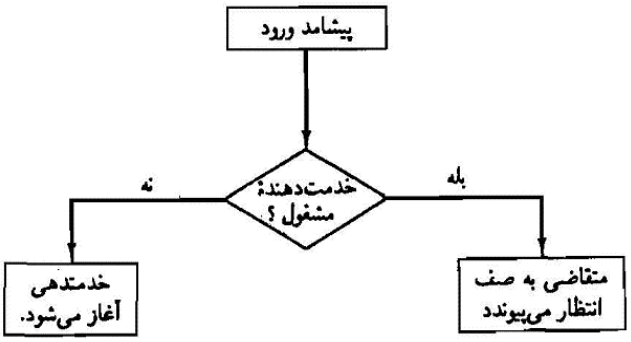
اگر خدمتدهی تازه کامل شده باشد، شبیه سازی مطابق دیاگرام جریان که در شکل ۲- ۲ نشان داده شد است ادامه می یابد. توجه کنید که خدمت دهنده در شکل ۲-۲ تنها دو وضعیت دارد: یا مشغول یا بیکار است.

پیشامد دوم هنگامی روی می دهد که یک متقاضی به سیستم واردشود. دیاگرام چنین موردی در شکل ۲-۳ نشان داده شد است. متقاضی وارد شده ممکن است خدمت دهنده را بیکار یا مشغول بیاید. بنابراین، یا بر خدمت دهنده وارد می شود یا بدین منظور به صف ملحق می شود. اقدام مقتضی در مورد متقاضی مورد بحث به شرح شکل ۲-۴ اعمال می شود. اگر خدمت دهنده مشغول باشد، متقاضی به صف وارد می شود. اگر خدمت دهنده بیکار و صف خالی باشد، متقاضی به خدمت دهنده وارد می شود. این امر غیر ممکن است که خدمت دهنده بیکار و صف غیرخالی باشد.

با کامل کردن خدمتدهی ممکن است خدمت دهنده بیکار شود یا با خدمتدهی به متقاضی بعدی همچنان مشغول بماند. شکل۲-۵ رابطه این دو نتیجه با وضعیت صف را نشان می دهد.



شکل ۲-۲ دیاگرام جریان مربوط به خدمتدهی تازه تکمیل شده



شکل۲-۳ دیاگرام جریان ورود به سیستم .

		وضعیت صف	
		غیرخالی	خالی
وضعیت خدمتدهنده	مشغول	ورود به صف	ورود به صف
	بیکار	غیرممکن	شروع خدمتدهی

شکل ۲-۴ عملیات متصور به هنگام ورود یک متقاضی

		وضعیت صف	
		غیرخالی	خالی
وضعیت خدمتدهنده	مشغول		ناممکن
	بیکار	ناممکن	

شکل ۲-۵ وضعیت خدمت دهنده پس از تکمیل خدمتدهی

اگر صف خالی نباشد، متقاضی دیگری به خدمت دهنده می رسد و خدمت دهنده مشغول می ماند. اگر صف خالی باشد، پس از کامل کردن خدمتدهی ، خدمت دهنده بیکار خواهد شد. این دو امکان با بخشهای سایه خورده شکل ۲-۵ نشان داده شد است. با کامل شدن هر خدمتدهی، اگر صف خالی باشد، امکان ندارد که خدمت دهنده مشغول بماند. همچنین پس از کامل شدن خدمتدهی، اگر صف خالی نباشد، امکان ندارد که خدمت دهنده بیکار بماند.

اینک باید دید که پیشامدهای گفته شده چگونه با گذشت زمان شبیه سازی رخ می دهد . به طور کلی، شبیه سازی سیستم های صف ناظر به نگهداری فهرستی از پیشامدهاست تا آنچه را که در زمانهای بعد رخ می دهد، تعیین کند. این فهرست معرف زمانهای رخ دادن انواع پیشامدهای گوناگون مربوط به هر یک از افراد حاظر در سیستم است. زمانگیری با « ساعتی» که مشخص کننده رخ دادن پیشامدها با گذشت زمان است انجام می شود . معمولاً در شبیه سازی، پیشامدها به طور تصادفی روی می دهند . تصادفی بودن تقلیدی از زندگی واقعی است که عدم اطمینان را نشان می دهد . مثلاً به طور قطع معلوم نیست که چه موقع مشتری بعدی برای ترک فروشگاه مواد غذایی ، به صندوق فروشگاه مراجعه می‌کند. یا قطعاً معلوم نیست که چقدر طول می کشد تا کارمند باجه بانک، ثبت یک نقل و انتقال مالی را به اتمام برساند .

معرفی عامل تصادف مورد نیاز برای تقلید زندگی واقعی، با استفاده از « اعداد تصادفی» میسر است. اعداد تصادفی به طور یکنواخت و مستقل در فاصله(۰،۱) توزیع می شود. ارقام تصادفی نیز به طور یکنواخت روی مجموعه {۹، ...، ۱، ۰} توزیع می شود. در تولید تعداد تصادفی می توان با کنار هم قرار دادن ارقام تصادفی به تعداد مناسب و نوشتن ممیز در سمت چپ عدد به دست آمده به مقصود رسید. تعداد مناسب ارقام را دقتی تعیین می کند که داده های مصرفی به عنوان ورودی باید از آن برخوردار باشد.

اعداد تصادفی را تولید هم می توان کرد. هر گاه اعداد با استفاده از شیوه‌ای از قبل تعیین شده تولید شوند، به آنها اعداد شبه تصادفی می گویند. چون روش تولید معلوم است، همواره می توان پیش از شبیه سازی دانست که دنباله این اعداد کدام است. روش های مخلف تولید اعداد تصادفی را در فصل ۷ بررسی کرده ایم.

در مساله تک مجرایی صف، مدتهای بین دو ورود و مدتهای خدمتدهی بر اساس توزیع های این متغیرهای تصادفی تعیین(تولید) می شوند . مثال های زیر نشان می دهد که این مدتها چگونه تولید می شوند.

برای رعایت سادگی فرض کنید که مدت های بین ورودها باپنج بار ریختن یک تاس و ثبت عددی که بر وجه بالایی تاس نمایان شده است، تولید شود . جدول ۲-۲ مجموعه پنج مدت بین ورود تولید شده به این ترتیب را نشان می دهد. ازاین پنج مدت بین دو ورود برای محاسبه زمان های ورود شش مشتری به سیستم صف استفاده شده است. فرض بر این است که اولین مشتری در زمان صفر وارد می شود. با این رخداد، ساعت به کار می افتد. مشتری دوم، دو واحد زمانی بعد از آن، یعنی در زمان ۲ وارد می شود. مشتری سوم، چهار واحد زمانی بعد، در زمان ۶ وارد می شود، و...

مدت مورد نیاز دیگر، مدت خدمتدهی است. جدل ۲-۳ مدت های خدمتدهی را در بر می گیرد که از توزیع تصادفی مدت های خدمتدهی تولید شده است. تنها مقادیر ممکن خدمتدهی ،یک ،دو، سه و چهار واحد زمانی است.

مشتری	مدت بین دو ورود	زمان ورود برحسب ساعت شبیه‌سازی
۱	–	۰
۲	۲	۲
۳	۴	۶
۴	۱	۷
۵	۲	۹
۶	۶	۱۵

جدل ۲-۲ مدتهای بین دو ورود و زمانهای ورود

مشتری	مدت خدمتدهی
۱	۲
۲	۱
۳	۳
۴	۲
۵	۱
۶	۴

جدول ۲-۳ مدتهای خدمتدهی

با پذیرش این فرض که این مقادیر چهارگانه دارای احتمال رخداد یکسان‌اند، این مقادیر بدین صورت قابل تولیدند که اعداد یک تا چهار را بر مهره هایی می نویسیم و با جانشینی آنها را از سبدی بیرون می آوریم و ثبت می کنیم. حال، برای شبیه سازی سیستم تک مجرایی صف باید مدتهای بین دو ورود و مدتهای خدمتدهی را به هم مرتبط کرد. همچنانکه جدول ۲-۴ نشان می دهد، اولین مشتری در زمان صفر وارد و خدمتدهی به او که نیازمند دو دقیقه وقت است، بلافاصله شروع می شود. خدمتدهی در زمان ۲ کامل می شود. مشتری دوم در زمان ۲ وارد و کار او در زمان ۳ تمام می شود. توجه کنید که مشتری چهارم در زمان ۷ وارد شده است ولی خدمتدهی را تا زمان ۹ نمی توان شروع کرد. زیرا خدمتدهی به مشتری ۳ تا زمان ۹ تمام نشده است.

جدول ۲-۴ مشخصاً برای مساله تک مجرایی صف طراحی شده است که به مشتریان بر اساس ترتیب ورود به سیستم خدمت می دهد و در این جدول بر اساس ساعت شبیه سازی ، حساب زمان رخداد هر پیشامد ثبت شده است. در ستون دوم جدول۲-۴ زمان هر پیشامد ورود ثبت شده است در حالی که در

ستون آخر، زمان هر پیشامند ترک ثبت شده است. رخ دادن این دو پیشامد با رعایت ترتیب زمانی در جدول ۲-۵ و شکل ۲-۶ نشان داده شده است.

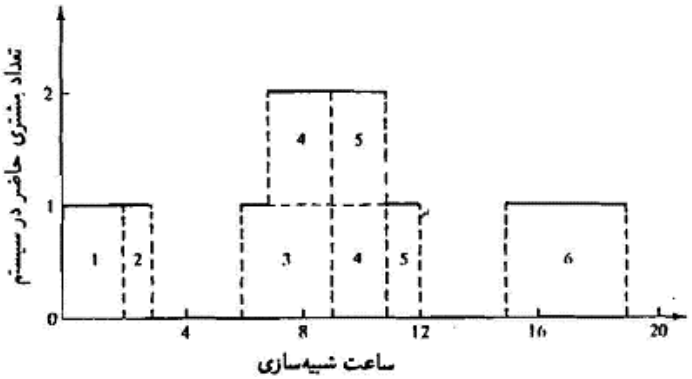
باید توجه داشت که جدول ۲-۵ بر اساس ساعت شبیه سازی تنظیم شده است و ممکن است پیشامدها در آن لزوماً بر حسب شماره مشتری مرتب نشده باشد. مرتب کردن پیشامدها بر حسب زمان که آن را درفصل بعدی تشریح کرده ایم اساس شبیه سازی پیشامدهای گسسته را تشکیل می دهد. شکل ۲-۶ تعداد مشتری حاضر در سیستم را در زمانهای مختلف شبیه سازی نشان می دهد. در واقع، این شکل نمایش تصویری فهرست مندرج در جدول ۲-۵ است. مشتری ۱ از زمان صفر تا ۲ در سیستم حاضر است. مشتری ۲ در زمان ۲ به سیستم وارد و در زمان ۳ از سیستم خارج می شود. از زمان ۳ تا ۶ مشتریانی در سیستم نیستند و در برخی دوره ها دو مشتری در سیستم حاضرند، مانند زمان ۸ که مشتریان ۳ و ۴ در سیستم حاضرند. زمانهایی نیز وجود دارد که پیشامدها با هم رخ می دهد، مثل زمان ۹ که مشتری ۳، سیستم را ترک می کند و مشتری ۵ به آن وارد می شود.

مشتری	زمان ورود	زمان شروع خدمت	مدت خدمتدهی	زمان پایان خدمتدهی
۱	۰	۰	۲	۲
۲	۲	۲	۱	۳
۳	۶	۶	۳	۹
۴	۷	۹	۲	۱۱
۵	۹	۱۱	۱	۱۲
۶	۱۵	۱۵	۴	۱۹

جدول ۲-۴ جدول شبیه سازی با تاکید بر اینکه زمانها بر اساس ساعت شبیه سازی باشد.

نوع پیشامد	مشتری	ساعت شبیه‌سازی
ورود	۱	۰
ترک	۱	۲
ورود	۲	۲
ترک	۲	۳
ورود	۳	۶
ورود	۴	۷
ترک	۳	۹
ورود	۵	۹
ترک	۴	۱۱
ترک	۵	۱۲
ورود	۶	۱۵
ترک	۶	۱۹

جدول ۲-۵ ترتیب زمانی پیشامد



شکل ۲-۶ تعداد مشتری حاضر در سیستم

مثال ۲-۱ صف تک مجرایی

یک فروشگاه مواد غذایی تنها یک باجه صندوق دارد. مشتری ها به طور تصادفی با فواصل زمانی یک تا ۸ دقیقه به صندوق مراجعه می کنند . همان طور که جدول ۲-۶ نشان می دهد، هر مقدار ممکن برای مدت ورود، احتمالی یکسان برای رخ دادن دارد. مدت های خدمتدهی از یک تا شش دقیقه و طبق احتمالات نشان داده شده در جدول۲-۷ تغییر می‌کند. دو ستون آخر جداول ۲-۶ و ۲-۷ را پس از این تشریح خواهیم کرد. این مساله تحلیل سیستم از طریق شبیه سازی ورود ۲۰ مشتری و خدمتدهی به آنهاست.

مدتهای بین ورود (دقیقه)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۱	۰٫۱۲۵	۰٫۱۲۵	۰۰۱-۱۲۵
۲	۰٫۱۲۵	۰٫۲۵۰	۱۲۶-۲۵۰
۳	۰٫۱۲۵	۰٫۳۷۵	۲۵۱-۳۷۵
۴	۰٫۱۲۵	۰٫۵۰۰	۳۷۶-۵۰۰
۵	۰٫۱۲۵	۰٫۶۲۵	۵۰۱-۶۲۵
۶	۰٫۱۲۵	۰٫۷۵۰	۶۲۶-۷۵۰
۷	۰٫۱۲۵	۰٫۸۷۵	۷۵۱-۸۷۵
۸	۰٫۱۲۵	۱٫۰۰۰	۸۷۶-۰۰۰

جدول ۲-۶ توزیع مدتهای بین دو ورود .

مدت خدمتدهی (دقیقه)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۱	۰٫۱۰	۰٫۱۰	۰۱-۱۰
۲	۰٫۲۰	۰٫۳۰	۱۱-۳۰
۳	۰٫۳۰	۰٫۶۰	۳۱-۶۰
۴	۰٫۲۵	۰٫۸۵	۶۱-۸۵
۵	۰٫۱۰	۰٫۹۵	۸۶-۹۵
۶	۰٫۰۵	۱٫۰۰	۹۶-۰۰

جدول ۲-۷ توزیع مدتهای خدمتدهی

در عمل، اجرایی که ۲۰ مشتری را در برگیرد برای نتیجه گیری نهایی بسیار کوچک است. اما، هدف این تمرین، تشریح چگونگی اجرای شبیه سازی های دستی است و نه توصیه انجام تغییراتی در فروشگاه .

به منظور تولید ورودی ها به باجه صندوق، به مجموعه ای از اعداد تصادفی با توزیع یکنواخت نیاز داریم. اعداد تصادفی دارای خاصیت های زیرهستند:

۱ - مجموعه اعداد تصادفی به طور یکنواخت بین صفر و یک توزیع می شوند .

۲ - اعداد تصادفی متوالی مستقل‌اند .

قبلاً خاطر نشان کردیم که اعداد تصادفی با توزیع یکنواخت از راههای فراوانی قابل تولیدند که برخی از این راهها را در فصلهای بعدی شریح کرده ایم. به علاوه جدولهای ارقام تصادفی نیز در دسترس اند برای تولید اعداد تصادفی در این مثال، از این جدول ها استفاده کرده‌ایم.

با زدن ممیز در جای مناسب، ارقام تصادفی به اعداد تصادفی تبدیل می شوند .در این مثال اعداد تصادفی با سه رقم اعشار کافی اند. برای تولید مدت های بین دو ورودی تنها با ۱۹ عدد تصادفی نیازمندیم، زیرا اولین ورود طبق فرض در زمان صفر رخ می دهد، پس برای ۲۰ نفر شدن تعداد مشتری ها نیاز به تولید ۱۹ ورود دیگر داریم.

از دو ستون آخر جدول های ۲-۶ و ۲-۷ برای تولید ورودها و مدت های خدمتدهی تصادفی استفاده می شود. ستون ما قبل آخر هر جدول نمایشگر احتمال تجمعی مربوط به هر توزیع است. ستون آخر تخصیص ارقام تصادفی را نشان می دهد. حالا جدول ۲-۶ را در نظر بگیرید. اولین ارقام تصادفی تخصیص یافته ۰۰۱-۱۲۵ است. شمار اعداد سه رقمی ممکن ، ۱۰۰۰ است (۰۰۱ تا ۰۰۰). احتمال یک دقیقه شدن مدت بین ورود۰/۱۲۵ است و ۱۲۵ عدد سه رقمی از ۱۰۰۰ عدد ممکن به چنین رخدادی تخصیص می یابد. به اعداد سه رقمی نیاز داریم، زیرا توزیع احتمال با دقت سه رقم اعشار تعریف شده است . مثلاً احتمال ۴ دقیقه شدن مدت بین دو ورود ۰/۱۲۵ است. با تهیه فهرست ۱۹ عدد سه رقمی از جدولی که قبلاً گفته شد و مقایسه آنها با ارقام تصادفی تخصیص یافته در جدول ۲-۶ مدت های بین ورود برای ۱۹مشتری تولید می شود .

تعیین مدت بین دو ورود را در جدول۲-۸ نشان داده ایم. توجه کنیدکه اولین ارقام تصادفی ۹۱۳ است. برای یافتن مدت بین ورود مربوطه، به ستون چهارم جدول ۲-۶ وارد شوید و در ستون اول جدول ۸ دقیقه را بخوانید.

مشتری	ارقام تصادفی	مدت بین دو ورود (دقیقه)	مشتری	ارقام تصادفی	مدت بین دو ورود (دقیقه)
۱	-	-	۱۱	۱۰۹	۱
۲	۹۱۳	۸	۱۲	۰۹۳	۱
۳	۷۲۷	۶	۱۳	۶۰۷	۵
۴	۰۱۵	۱	۱۴	۷۳۸	۶
۵	۹۴۸	۸	۱۵	۳۵۹	۳
۶	۳۰۹	۳	۱۶	۸۸۸	۸
۷	۹۲۲	۸	۱۷	۱۰۶	۱
۸	۷۵۳	۷	۱۸	۲۱۲	۲
۹	۲۳۵	۲	۱۹	۴۹۳	۴
۱۰	۳۰۲	۳	۲۰	۵۳۵	۵

جدول۲-۸ تعیین مدتهای بین دو ورود

مشتری	ارقام تصادفی	مدت خدمتدهی (دقیقه)	مشتری	ارقام تصادفی	مدت خدمتدهی (دقیقه)
۱	۸۴	۴	۱۱	۳۲	۳
۲	۱۰	۱	۱۲	۹۳	۵
۳	۷۴	۴	۱۳	۷۹	۴
۴	۵۳	۳	۱۴	۰۵	۱
۵	۱۷	۲	۱۵	۷۹	۵
۶	۷۹	۴	۱۶	۸۳	۴
۷	۹۱	۵	۱۷	۵۲	۳
۸	۶۷	۴	۱۸	۵۵	۳
۹	۸۹	۵	۱۹	۳۰	۲
۱۰	۳۸	۳	۲۰	۵۰	۳

جدول ۲-۹ مدتهای تولید شده برای خدمتدهی

مدتهای خدمتدهی برای هر ۲۰ مشتری در جدول ۲-۹ نشان داده شده است. این مدت ها بر اساس روش تشریح شده در فوق و با استفاده از جدول۲-۷ تولید شده‌اند. مدت خدمتدهی به مشتری اول ۴ دقیقه است زیرا ارقام تصادفی ۸۴ در رده ۶۱-۸۵ قرار می گیرد .

جدول شبیه سازی خلاصه شبیه سازی دستی است. این گونه جدول ها برای وضعیت در دست بررسی طراحی و چنان ساخته می شوند که پرسشهای مطروحه را پاسخ گویند. جدول شبیه سازی این مساله را در جدول ۲-۱۰ نشان دادیم که متممی برای جدولی از نوع جدول۲-۴ است که قبلاًدیده‌ایم . فرض می کنیم که مشتری اول در زمان صفر وارد شود.خدمتدهی بلافاصله شروع و در زمان ۴ تمام می شود . مشتری به مدت ۴ دقیقه در سیستم بوده است. مشتری دوم در زمان ۸ وارد می شود. بدین ترتیب، خدمت دهنده (دریافت کننده پول) به مدت ۴ دقیقه بیکار بوده است.

این فرایند در مورد هر ۲۰ مشتری اجرا می شود. همان طور که دیده می شود، مجموعها برای مدت های خدمتدهی، مدتهای ماندن مشتریان در سیستم، مدت بیکاری خدمت دهنده و مدت انتظار مشتریان در صف تعیین می شود .

برخی از یافته های شبیه سازی مدت به شرح زیر است:

۱. متوسط مدت انتظار هر مشتری ۲/۸ دقیقه است. این نتیجه به طریق زیر تعیین می شود:

$$\text{متوسط مدت انتظار (دقیقه)} = \frac{\text{مجموع مدت انتظار مشتریان در صف(دقیقه)}}{\text{مجموع تعداد مشتریان}}$$

$$= \frac{56}{20} = \text{۲/۸ دقیقه}$$

۲. احتمال مجبور شدن هر مشتری به انتظار کشیدن در صف ۰/۶۵ است. این نتیجه به طریق زیر تعیین می شود:

$$\text{احتمال(انتظار)} = \frac{\text{تعداد مشتریانی که در انتظار می مانند}}{\text{مجموع تعداد مشتریان}}$$

$$= \frac{13}{20} = \text{۰/۶۵}$$

۳. نسبت مدت بیکاری خدمت دهنده ۰/۲۱ است. این نتیجه به طریق زیر تعیین می‌شود:

$$\text{احتمال بیکاری خدمت دهنده} = \frac{\text{مجمو مدت بیکاری خدمت دهنده(دقیقه)}}{\text{مجموع مدت اجرای شبیه سازی(دقیقه)}}$$

$$= \frac{18}{86(} = \text{۰/۲۱}$$

احتمال مشغول بودن خدمت دهنده، مکمل ۰/۲۱ یا۰/۷۹ است.

۴. متوسط مدت خدمتدهی۳/۴ دقیقه است. نتیجه به طریق زیر تعیین می‌شود:

متوسط مدت خدمتدهی = مجموع مدت خدمتدهی(دقیقه) / مجموع تعداد مشتریان

$$= \frac{68}{20} = 3/4 \text{ دقیقه}$$

می توان این نتیجه را با یافتن مانگین توزیع مدت خدمتدهی با به کارگیری معادله

$$E(S) = \sum_{s=0}^{\infty} sp(s)$$

با امید ریاضی مدت خدمتدهی مقایسه کرد. با به کارگیری معادله امید ریاضی در مورد توزیع مندرج در جدول۲-۷ به نتیجه زیر می رسیم.

$$\text{امید ریاضی مدت خدمتدهی} = 1(0/10) + 2(0/20) + 3(0/30) + 4(0/25) + (0/10) + 6(0/05)$$

$$= ۳/۲ \text{ دقیقه}$$

امید ریاضی مدت خدمتدهی اندکی کمتراز متوسط مدت خدمتدهی در شبیه سازی است. هر چه شبیه سازی طولانی تر باشد، این متوسط به $E(S)$ نزدیکتر می شود.

۵. متوسط مدت بین دو ورود۴/۳ دقیقه است. این نتیجه به طریق زیر تعیین می شود:

متوسط مدت بین دو ورود(دقیقه) = جمع تمام مدتها بین دو ورود(دقیقه) / تعداد ورودها منهای یک

$$= \frac{82}{19} = ۴/۳ \text{ دقیقه}$$

یک را از مخرج کم می کنیم، زیرا فرض بر این است که اولین ورود در زمان صفر روی می دهد . می توان این نتیجه را با یافتن میانگین توزیع یکنواخت گسسته‌ای که نقاط شروع و پایان آن $a = 1$ و$b = 8$ است، مقایسه کرد. میانگین از رابطه

$$E(A) = \frac{a+b}{2} = \frac{1+8}{2} = 4/5 \text{ دقیقه}$$

به دست می آید. امید ریاضی مدت بین ورودها کمی بیش از مقدار متوسط است. اما، در شبیه سازی های طولانی تر مقدار متوسط مدت بین ورودها باید به میانگین تئوریک$E(A)$ ، میل کند.

۶. متوسط مدت انتظار آنهایی که به انتظار می مانند، ۴/۳ دقیقه است. این به طریق زیر تعیین می شود:

متوسط مدت انتظار آنهایی که به انتظار می مانند(دقیقه) = مجموع مدتی که مشتریان در صف به انتظار می مانند(دقیقه) / مجموع تعداد مشتریانی که در صف به انتظار می مانند

$$= \frac{56}{13} = 4/3 \text{ دقیقه}$$

۷. متوسط مدتی که هر مشتری در سیستم می گذارند، ۶/۲ دقیقه است. این نتیجه را می توان از دو طریق به دست آورد. اول اینکه، محاسبه را می توان با استفاده از رابطه زیر انجام داد:

متوسط مدت ماندن مشتری در سیستم (دقیقه) = مجموع مدت ماندن مشتریان در سیستم(دقیقه) / مجموع تعداد مشتریان

$$= \frac{124}{20} = 6/2 \text{ دقیقه}$$

راه دیگر محاسبه همین نتیجه، تشخیص این مطلب است که رابطه زیر باید برقرار باشد:

متوسط مدتی که مشتری برای خدمتگیری صرف می کند(دقیقه) + متوسط مدتی که مشتری در صف به انتظار می ماند(دقیقه) = متوسط مدتی که مشتری در سیستم می ماند(دقیقه)

یافته های ۱ و ۴ فهرست بالا داده های لازم برای سمت راست این معادله را فراهم می آورند تا همین نتیجه به دست آید.

مثال۲-۲ مساله اتورستوران هابیل و خباز

هدف این مثال، ارائه شیوه شبیه سازی در موردی است که بیش از یک مجرا وجود داشته باشد. یک اتورستوران را در نظر بگیرید که آورندگان غذا سفارش ها را دریافت می کنند و غذا را به خودروها می آورند. خودروها به صورت مندرج درجدل ۲-۱۱ وارد می شوند. تعداد آورندگان غذا دو نفر است به نام های هابیل و خباز. هابیل برای انجام این کار تواناتر است و کمی سریعتر از خباز کار می کند. توزیع مدتهای خدمتدهی در جدولهای ۲-۱۲ و ۲-۱۳ نشان داده شده است.

شبیه سازی به طریقی همانند مثال ۲-۱ انجام می گیرد، با این تفاوت که این بار پیچیده تر است. قاعده ساده کننده این است که اگر هر دو آورنده غذا بیکار باشند، هابیل مشتری از راه رسیده را می گیرد. شاید هابیل با سابقه تر باشد (اگر تصمیم در این مورد که هر گاه هر دو بیکارند چه کسی به خودروی وارد شده خدمت دهد بر پایه تصادفی استوار می بود جواب متفاوتی به دست می آمد). مساله این است که روش فعلی تا چه حد خوب کار می‌کند . برای براورد معیارهای عملکرد سیستم، از شبیه سازی یک ساعته عملیات سیستم استفاده می کنیم. شبیه سازی طولانی تر بسیار مطمئنتر خواهد بود، ولی بنا به مقاصد توضیحی دوره یک ساعته را انتخاب کرده‌ایم.

مدتهای بین دو ورود(دقیقه)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۱	۰٫۲۵	۰٫۲۵	۰۱-۲۵
۲	۰٫۴۰	۰٫۶۵	۲۶-۶۵
۳	۰٫۲۰	۰٫۸۵	۶۶-۸۵
۴	۰٫۱۵	۱٫۰۰	۸۶-۰۰

جدول ۲-۱۱ توزیع مدتهای بین ورود خوردها

مدت خدمتدهی (دقیقه)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۲	۰٫۳۰	۰٫۳۰	۰۱-۳۰
۳	۰٫۲۸	۰٫۵۸	۳۱-۵۸
۴	۰٫۲۵	۰٫۸۳	۵۹-۸۳
۵	۰٫۱۷	۱٫۰۰	۸۴-۰۰

جدول ۲-۱۲ توزیع خدمتدهی هابیل

مدت خدمتدهی (دقیقه)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۳	۰٫۳۵	۰٫۳۵	۰۱-۳۵
۴	۰٫۲۵	۰٫۶۰	۳۶-۶۰
۵	۰٫۲۰	۰٫۸۰	۶۱-۸۰
۶	۰٫۲۰	۱٫۰۰	۸۱-۰۰

جدول ۲-۱۳ توزیع خدمتدهی خباز

شبیه سازی بیشتر به شیوه مثال ۲-۱ انجام می شود ، در این مثال شبیه سازی پیشامدهای بیشتری وجود دارد، زیرا خدمت دهنده دومی نیز داریم. پیشامدهای ممکن به شرح زیر است: مشتریان وارد می شوند، خدمتدهی به مشتریان توسط هابیل شروع می شود. خدمتدهی به مشتریان توسط هابیل کامل می شود. خدمتدهی به مشتریان توسط خباز شروع می شود. خدمتدهی به مشتریان توسط خباز کامل می شود. جدول ۲-۱۴ جدول شبیه سازی را نشان می دهد .

مشتری	ارقام تصادفی ورود	دقیقه‌های بین دو ورود	زمانهای ورود برحسب دقیقه شبیه‌سازی	ارقام تصادفی خدمتدهی	هابیل: زمانهای شروع خدمت	هابیل: دقیقه‌های خدمتدهی	هابیل: زمانهای پایان خدمت	خباز: زمانهای شروع خدمت	خباز: دقیقه‌های خدمتدهی	خباز: زمانهای پایان خدمت	مدت انتظار در صف
۱	-	-	۰	۹۵	۰	۵	۵				۰
۲	۴۶	۲	۲	۲۱				۲	۳	۵	۰
۳	۹۸	۴	۶	۵۱	۶	۳	۹				۰
۴	۹۰	۲	۱۰	۹۲	۱۰	۵	۱۵				۰
۵	۴۶	۲	۱۲	۸۹				۱۲	۶	۱۸	۰
۶	۲۲	۲	۱۴	۳۸	۱۵	۳	۱۸				۱
۷	۷۳	۳	۱۷	۱۳	۱۸	۲	۲۰				۱
۸	۸۰	۳	۲۰	۶۱	۲۰	۲	۲۲				۰
۹	۶۸	۳	۲۳	۵۰				۲۳	۴	۲۷	۰
۱۰	۲۲	۱	۲۴	۲۹	۲۴	۳	۲۷				۰
۱۱	۲۸	۲	۲۶	۲۹	۲۷	۳	۳۰				۱
۱۲	۳۳	۲	۲۸	۵۳				۲۸	۴	۳۲	۰
۱۳	۲۵	۲	۳۰	۸۸	۳۰	۵	۳۵				۰
۱۴	۲۲	۱	۳۱	۰۱				۳۲	۳	۳۵	۱
۱۵	۲۳	۲	۳۳	۸۱	۳۵	۲	۳۶				۲
۱۶	۶۳	۲	۳۵	۵۳				۳۵	۴	۳۹	۰
۱۷	۲۸	۲	۳۷	۸۱	۳۹	۲	۴۱				۲
۱۸	۸۰	۳	۴۰	۶۳				۴۰	۵	۴۵	۰
۱۹	۲۲	۲	۴۲	۰۱	۴۳	۲	۴۵				۱
۲۰	۵۶	۲	۴۴	۶۷	۴۵	۲	۴۷				۱
۲۱	۸۹	۲	۴۸	۰۱				۴۸	۳	۵۱	۰
۲۲	۱۸	۱	۴۹	۳۳	۴۹	۳	۵۲				۰
۲۳	۵۱	۲	۵۱	۷۵				۵۱	۵	۵۶	۰
۲۴	۷۱	۳	۵۴	۵۲	۵۴	۳	۵۷				۰
۲۵	۱۶	۱	۵۵	۸۷				۵۶	۶	۶۲	۱
۲۶	۱۲	۲	۵۹	۲۳	۵۹	۳	۶۲				۰
						۵۶			۳۳		۱۱

جدول ۲-۱۴ جدول شبیه سازی برای مثال رستوران

تجزیه و تحلیل جدول ۲-۱۴ به یافته های زیر می انجامد:

۱ - در دوره ۶۲ دقیقه ای، هابیل در۹۰ درصد از وقت به خدمتدهی مشغول است.

۲ - خباز تنها ۶۹ درصد از وقت را به خدمتدهی مشغول بوده است، قاعده حق تقدم موجب شده است که خباز کمتر مشغول باشد.

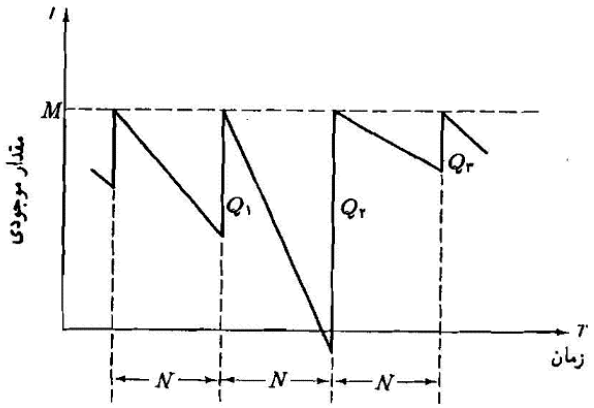
۳ - ۹ نفر از ۲۶ نفر افراد وارد شده یا در حدود ۳۵ درصد ازآنها ناچار از انتظار کشیدن بوده اند. متوسط مدت انتظار برای تمام مشتریان تنها ۰/۴۲ دققه یا۲۵ ثانیه بوده است که مقداری بسیار کم است.

۴ - ۹ موردی که مجبور بوده اند منتظر بمانند، به طور متوسط تنها ۱/۲۲دقیقه انتظار کشیده‌اند که مقداری بسیار کم است.

۵ - خلاصه، این سیستم متوازن به نظر می رسد. یک خدمت دهنده نمی توان از عهده خدمتدهی به همه متقاضیان غذا برآید و سه خدمت دهنده نیز احتمالاً بسیار زیاد است. افزودن یک خدمت دهنده دیگر، مطمئناً مدت انتظار را تقریباً به صفر می رساند . اما برای توجیه به کارگیری خدمت دهنده دیگر، هزینه مورد انتظار باید بسیار زیاد باشد.

شبیه سازی سیستم های موجودی

رده مهمی از مسائل شبیه سازی به سیستم های موجودی مربوط است. سیستم ساده موجودی در شکل ۲-۷ نشان داده شده است.



شکل۲-۷ سیستم موجودی احتمالی سطح سفارش

این سیستم موجودی، مروری دوره‌ای به طول N دارد که در آن، سطح موجودی وارسی می شود و سفارشی صادر می شود که موجودی را به سطح M بالا خواهد آورد. در پایان اولین دوره بررسی، سفارش به مقدار Q_1 صادر می شود. مدت تحویل(یعنی بین صدور و دریافت سفارش) دراین سیستم موجودی صفر است. چون مقادیر تقاضا معمولاً با اطمینان مشخص نیست، مقادیر سفارش احتمالی اند. در شکل ۲-۷ تقاضا به صورت یکنواخت در دوره زمانی نشان داده شده است. مقادیر تقاضا در عمل یکنواخت نیست و با گذشت زمان دستخوش نوسان می شود. یک امکان این است که همه تقاضا در شروع دوره برسد. امکان واقع بینانه دیگر نیز این است که مهلت تحویل غیر صفر و احتمالی باشد.

توجه کنید که مقدار موجودی در دور دوم به زیر صفر کاهش می یابد که این موضوع معرف کمبود است. در شکل ۲-۷ ، این واحد سفارش تحویل نشده را تشکیل می دهد. هر گاه سفارشی برسد، ابتدا به تقاضای مربوط به اقلام سفارش تحویل نشده پاسخ داده می شود. برای پرهیز از کمبود، نیاز به نگهداری یک ذخیره یا موجودی اطمینان وجود دارد.

برای نگهداری موجودی در انبار، هزینه ای وجود دارد که آن را به بهره پرداختی برای تهیه منابع مالی قرض شده جهت خرید اقلام نسبت می دهند. (این هزینه را می توان به عنوان زیان ناشی از نبود منابع مالی برای سایر مقاصد سرمایه گذاری نیز تعبیر کرد) سایر هزینه ها از قبیل اجاره فضای انبار، استخدام نگهبان، و... را می توان در ستون هزینه های نگهداری موجودی قرار داد.

راه دیگری که می توان به جای نگهداری موجودی فراوان در انبار از ان بهره گرفت، بررسی وضعیت انبار به دفعات بیشتر و در نتیجه، خریدها یا بازسازی های موجودی به دفعات بیشتر است. این شیوه نیز هزینه ای دارد به نام هزینه سفارشدهی. کم بودن موجودی نیز هزینه هایی دارد. ممکن است مشتریان ناخرسند شوند و شهرت تجاری بدین گونه از دست برود. داشتن موجودی های زیاد از امکان کمبود می کاهد . این هزینه ها باید چنان متوازن شوند که هزینه کل سیستم موجودی مینیمم شود.

هزینه کل (یا سود کل) سیستم موجودی، معیار سنجش عملکرد سیستم است که می تواند تحت تاثیر خط مشی های متفاوت قرار گیرد. مثلاً در شکل ۲-۷ مهلت تحویل صفر باشد، دو پیشامد آخر همزمان رخ می دهند.

یک مساله قدیمی و مهم موجودی، به خرید و فروش روزنامه مربوط است. روزنامه فروش، هر نسخه روزنام را به۱۳ واحد پول می خرد و به ۲۰ واحد پول می فروشد. روزنامه های فروش نرفته در انتهای روز به عنوان باطله و هر نسخه به ۲ واحد پول فروخته می شود. روزنامه در بسته های ده تایی قابل خریدن است و روزنامه فروش تنها می تواند ۵۰، ۶۰ ... روزنامه بخرد. از لحاظ چگونگی اخبار سه نوع روز «خوب»، «متوسط»، و «بد» با احتملات به ترتیب، ۰/۳۵، ۰/۴۵ و ۰/۲۰ وجود دارد. توزیع روزنامه مورد تقاضا در هر یک از این روزها در جدول ۲-۱۵ ارائه شده است. هدف مساله تعیین تعداد بهینه روزنامه هایی است که روزنامه فروش باید بخرد. با شبیه سازی تقاضا برای ۲۰ روز و ثبت سود ناشی از فروش روزانه، این خواسته را تامین می‌کنیم. با استفاده از نظریه موجودی در فصل ۶، این مساله به راحتی حل می شود.

سود طبق رابطه زیر تعیین می شود:

سود از دست رفته به خاطر فزونی تقاضا − هزینه خرید روزنامه − درآمد ناشی ازفروش = سود درآمد درآمد ناشی از فروش روزنامه های باطله+

به موجب صورت مساله، درآمد ناشی از فروش هر نسخه روزنامه ۲۰ واحد پول است. هزینه خرید روزنامه نیز ۱۳ واحد پول به ازای هر نسخه است. سود از دست رفته به خاطر فزونی تقاضا به ازای هر نسخه روزنامه مورد تقاضا و غیر موجود، ۷ واحد پول است. هزینه کمبود بدین گونه تا حدی قابل بحث است ولی مساله را بسیار جالب می کند. درآمد ناشی ازفروش روزنامه های باطله ۲ واحد پول به ازای هر نسخه است.

تقاضا	توزیع احتمال تقاضا		
	خوب	متوسط	بد
۴۰	۰/۰۳	۰/۱۰	۰/۴۴
۵۰	۰/۰۵	۰/۱۸	۰/۲۲
۶۰	۰/۱۵	۰/۴۰	۰/۱۶
۷۰	۰/۲۰	۰/۲۰	۰/۱۲
۸۰	۰/۳۵	۰/۰۸	۰/۰۶
۹۰	۰/۱۵	۰/۰۴	۰/۰۰
۱۰۰	۰/۰۷	۰/۰۰	۰/۰۰

جدول ۲-۱۵ توزیع روزنامه های مورد تقاضا

جدولهای ۲-۱۶ و ۲-۱۷ تخصیص ارقام تصادفی برای تعیین نوع روزها و تقاضای مربوط به آن روزها را ارائه می کنند . حل این مساله از راه شبیه سازی ، نیازمند تعیین خط مشی خرید تعداد مشخصی روزنامه در هر روز و سپس شبیه سازی تقاضا برای روزنامه طی یک دوره زمانی ۲۰ روزه به منظور تعیین سود است. خط مشی(تعداد روزنامه‌های خریداری شد) با انتخاب مقادیر دیگر تغییر داده می شود تا جایی که سود در سطوح پیش و پس از آن کاهش یابد. مقدار میانی، تعداد بهینه روزنامه هایی است که روزنامه فروش باید خریداری کند.

جدول شبیه سازی برای خرید ۷۰ روزنامه در جدول ۲-۱۸ نشان داده شده است. در روز ۱ تقاضا برای روزنامه ۶۰ و درآمد ناشی ازفروش ۶۰ روزنامه ۱۲۰۰ واحد پول است. در پایان روز، ۱۰ روزنامه باقی می ماند .درآمد ناشی از فروش به قیمت باطله ۲۰ واحد پول و از قرار هر نسخه ۲ واحد پول است. سود روز اول به شرح زیر تعیین می شود:

سود = ۱۲۰۰ -۹۱۰-۰ +۲۰ =۳۱۰

نوع روز	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
خوب	۰٫۳۵	۰٫۳۵	۰۱-۳۵
متوسط	۰٫۴۵	۰٫۸۰	۳۶-۸۰
بد	۰٫۲۰	۱٫۰۰	۸۱-۰۰

جدول ۲-۱۶ تخصیص ارقام تصافی برای تعیین نوع روز

تقاضا	توزیع تجمعی			تخصیص اعداد تصادفی		
	خوب	متوسط	بد	خوب	متوسط	بد
۴۰	۰٫۰۳	۰٫۱۰	۰٫۴۴	۰۱-۰۳	۰۱-۱۰	۰۱-۴۴
۵۰	۰٫۰۸	۰٫۲۸	۰٫۶۶	۰۴-۰۸	۱۱-۲۸	۴۵-۶۶
۶۰	۰٫۲۳	۰٫۶۸	۰٫۸۲	۰۹-۲۳	۲۹-۶۸	۶۷-۸۲
۷۰	۰٫۴۳	۰٫۸۸	۰٫۹۴	۲۴-۴۳	۶۹-۸۸	۸۳-۹۴
۸۰	۰٫۷۸	۰٫۹۶	۱٫۰۰	۴۴-۷۸	۸۹-۹۶	۹۵-۰۰
۹۰	۰٫۹۳	۱٫۰۰	۱٫۰۰	۷۹-۹۳	۹۷-۰۰	
۱۰۰	۱٫۰۰	۱٫۰۰	۱٫۰۰	۹۴-۰۰		

جدل ۲-۱۷ تحصیص ارقام تصادفی برای روزنامه های مورد تقاضا

روز	ارقام تصادفی برای تعیین نوع روز	نوع روز	ارقام تصادفی برای تقاضا	تقاضا	درامد حاصل از فروش	سود از دست رفته به‌خاطر فروش تقاضا	درامد ناشی از فروش به قیمت باطله	سود روزانه
۱	۹۴	بد	۸۰	۶۰	۱۲۰۰	-	۲۰	۳۱۰
۲	۷۷	متوسط	۲۰	۵۰	۱۰۰۰	-	۴۰	۱۳۰
۳	۴۹	متوسط	۱۵	۵۰	۱۰۰۰	-	۴۰	۱۳۰
۴	۴۵	متوسط	۸۸	۷۰	۱۴۰۰	-	-	۴۹۰
۵	۴۳	متوسط	۹۸	۹۰	۱۴۰۰	۱۴۰	-	۳۵۰
۶	۳۲	خوب	۶۵	۸۰	۱۴۰۰	۷۰	-	۴۲۰
۷	۴۹	متوسط	۸۶	۷۰	۱۴۰۰	-	-	۴۹۰
۸	۰۰	بد	۷۳	۶۰	۱۲۰۰	-	۲۰	۳۱۰
۹	۱۶	خوب	۲۴	۷۰	۱۴۰۰	-	-	۴۹۰
۱۰	۲۴	خوب	۶۰	۸۰	۱۴۰۰	۷۰	-	۴۲۰
۱۱	۳۱	خوب	۶۰	۸۰	۱۴۰۰	۷۰	-	۴۲۰
۱۲	۱۴	خوب	۲۹	۷۰	۱۴۰۰	-	-	۴۹۰
۱۳	۴۱	متوسط	۱۸	۵۰	۱۰۰۰	-	۴۰	۱۳۰
۱۴	۶۱	متوسط	۹۰	۸۰	۱۴۰۰	۷۰	-	۴۲۰
۱۵	۸۵	بد	۹۳	۷۰	۱۴۰۰	-	-	۴۹۰
۱۶	۰۸	خوب	۷۳	۸۰	۱۴۰۰	۷۰	-	۴۲۰
۱۷	۱۵	خوب	۲۱	۶۰	۱۲۰۰	-	۲۰	۳۱۰
۱۸	۹۷	بد	۴۵	۵۰	۱۰۰۰	-	۴۰	۱۳۰
۱۹	۵۲	متوسط	۷۶	۷۰	۱۴۰۰	-	-	۴۹۰
۲۰	۷۸	متوسط	۹۶	۸۰	۱۴۰۰	۷۰	-	۴۲۰
					۲۵۸۰۰	۵۶۰	۲۲۰	۷۲۶۰

جدول۲-۱۸ شبیه سازی برای خرید ۷۰ روزنامه

در روز پنجم تقاضا بیش از عرضه است. درآمد ناشی ازفروش ۱۴۰۰ واحد پول است، زیرا با خط مشی جاری تنها ۷۰ نسخه روزنامه وجود دارد. چون ۲۰ روزنامه دیگر هم می توانست فروخته شود، سود از دست رفته معادل ۱۴۰ واحد پول(۷×۲۰) ارزیابی می شود. سود روزانه به شرح زیر تعیین می شود.

سود =۱۴۰۰ -۹۱۰ -۱۴۰ +۰ =۳۵۰

سود دوره ۲۰ روزه برابر با جمع سودهای روزانه، یعنی ۷۲۶۰ واحد پول است. این مبلغ را به شرح زیر و بر اساس جمع های به دست آمده از ۲۰ روز شبیه سازی نیز می توان یافت:

۷۲۶۰= ۲۲۰+ ۵۶۰- ۱۸۲۰۰ -۲۵۸۰۰= سود کل

تعیین تعداد بهینه روزنامه هایی که باید خرید به عنوان تمرین به خواننده واگذار می شود.

مثال ۲-۴ شبیه سازی سیستم موجودی (M,N)

این مثال از الگوی سیستم موجودی احتمالی سطح سفارش پیروی می کند، که در شکل ۲-۷ نشان داده شده است. فرض کنید که بالاترین سطح موجودی، M، ۱۱ واحد و دوره بررسی، N، ۵ روز باشد. مساله در مورد برآورد متوسط واحدهای مانده در انبار در پایان روز و تعداد روزهایی که شرایط کمبود در آنها وجود داشته باشد، از طریق شبیه سازی میباشد. توزیع تعداد واحدهای مورد تقاضا در روز، در جدول ۲-۱۹ نشان داده شده است. در این مثال، مهلت تحویل، متغیری تصادفی است که در جدول ۲-۲۰ نشان داده شده است. فرض کنید که سفارش ها در پایان روز صادر و در چارچوب تعیین شده توسط مهلت تحویل، در ابتدای روز وارد می شوند.

برای برآورد میانگین واحدهای مانده در انبار در پایان روز، دوره های بسیار باید شبیه سازی شود. برای این مثال تنها پنج دوره نشان داده خواهد شد.

تخصیص ارقام تصادفی برای تقاضای روزانه و مهلت تحویل در آخرین ستون سمت چپ جدول های ۲-۱۹ و ۲-۲۰ و جدول شبیه سازی به دست آمده نیز در جدول ۲-۲۱ نشان داده شده است. شبیه سازی تحت شرایطی آغاز شده که سطح موجودی ۳ واحد بوده و ورود یک سفارش ۸ واحدی در مدت دو روز برنامه ریزی شده بوده است.

جدول ۲-۱۹ تخصیص ارقام تصادفی برای تقاضای روزانه

تقاضا	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۰	۰/۱۰	۰/۱۰	۰۱-۱۰
۱	۰/۲۵	۰/۳۵	۱۱-۳۵
۲	۰/۳۵	۰/۷۰	۳۶-۷۰
۳	۰/۲۱	۰/۹۱	۷۱-۹۱
۴	۰/۰۹	۱/۰۰	۹۲-۰۰

جدول ۲-۲۰ تحصیص ارقام تصافی برای مهلت تحویل

مهلت تحویل (روز)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۱	۰/۶	۰/۶	۱-۶
۲	۰/۳	۰/۹	۷-۹
۳	۰/۱	۱/۰	۰

جدول ۲-۲۱ جدول شبیه سازی برای سیستم موجودی (M,N

دور	روز	موجودی در ابتدای روز	ارقام تصادفی برای تقاضا	تقاضا	موجودی در انتهای روز	مقدار کمبود	مقدار سفارش	ارقام تصادفی برای مهلت تحویل	روزهای مانده تا ورود سفارش
۱	۱	۳	۲۲	۱	۲	۰	-	-	۱
	۲	۲	۳۵	۱	۱	۰	-	-	۰
	۳	۹	۶۵	۲	۷	۰	-	-	-
	۴	۷	۸۱	۳	۴	۰	-	-	-
	۵	۴	۵۴	۲	۲	۰	۹	۵	۱
۲	۱	۲	۰۳	۰	۲	۰	-	-	۰
	۲	۱۱	۸۷	۳	۸	۰	-	-	-
	۳	۸	۲۷	۱	۷	۰	-	-	-
	۴	۷	۷۳	۳	۴	۰	-	-	-
	۵	۴	۷۰	۲	۲	۰	۹	۰	۳
۳	۱	۲	۴۷	۲	۰	۰	-	-	۲
	۲	۰	۴۵	۲	۰	۲	-	-	۱
	۳	۰	۴۸	۲	۰	۴	-	-	۰
	۴	۹	۱۷	۱	۴	۰	-	-	-
	۵	۴	۰۹	۰	۴	۰	۷	۳	۱
۴	۱	۴	۴۲	۲	۲	۰	-	-	۰
	۲	۹	۸۷	۳	۶	۰	-	-	-
	۳	۶	۲۶	۱	۵	۰	-	-	-
	۴	۵	۳۶	۲	۳	۰	-	-	-
	۵	۳	۴۰	۲	۱	۰	۱۰	۴	۱
۵	۱	۱	۰۷	۰	۱	۰	-	-	۰
	۲	۱۱	۶۳	۲	۹	۰	-	-	-
	۳	۹	۱۹	۱	۸	۰	-	-	-
	۴	۸	۸۸	۳	۵	۰	-	-	-
	۵	۵	۹۴	۴	۱	۰	۱۰	۸	۲
					۸۷				

بررسی جدول شبیه سازی برای چند روز منتخب، نحوه عمل فرایند را نشان می دهد . سفارش مربوط به ۸ واحد در صبح روز سوم از دوره اول وارد می شود و سطح موجودی را از ۱ به ۹ افزایش می دهد . تقاضا در خلال بقیه دوره اول موجودی پایان روز را به ۲ واحد در روز پنجم کاهش می دهد. بنابراین، سفارشی برای ۹ واحد صادر می شود.

مهلت تحویل برای این سفارش یک روز است. سفارش ۹ واحدی در صبح روز ۲ از دور ۲ به موجودی افزوده می شود.

توجه کنید که موجودی در ابتدای روز دوم از دور سوم صفر است. تقاضایی به میزان ۲ واحد در این روز وجود دارد، بنابراین کمبود ایجاد شده که این واحد در این روز و روز بعد در شمار واحدهای سفارش تحویل نشده، قرار می گیرند .در صبح روز ۴ از دور ۳، میزان وجودی در آغاز روز ۹ واحد است. ۴ واحد سفارش تحویل نشده و یک واحد مورد تقاضا دراین روز، موجودی در پایان روز را به ۴ واحد کاهش می دهد .

بر اساس پنج دور شبیه سازی، متوسط موجودی در پایان روز، تقریباً ۳/۵، واحد ۲۵÷۸۷) است و در دو روز از ۲۵ روز شرایط کمبود وجود دارد.

دیگر مثالهای شبیه سازی

این بخش شامل مثالهایی از شبیه سازی در زمینه پایایی، یک ماموریت بمباران و تولید توزیع احتمال تقاضا در مهلت تحویل بر اساس توزیعهای تقاضا و مهلت تحویل است.

یک ماشین فرز بزرگ، سه قطعه مختلف دارد که در جریان کار دچار خرابی می شوند. تابع توزیع تجمعی عمر قطعات یکسان و به شرح جدول ۲-۲۲ است. هرگاه قطعه ای خراب شود، فرز از کار می افتد، تعمیر

کاری احضار می شود و قطعه تازه ای نصب می شود. مدت تاخیر تعمیر کار در ورود به محل ماشین فرز نیز متغیری تصادفی با توزیع ارائه شده در جدول ۲-۲۳ است. هزینه مدت از کار ماندگی فرز از قرار دقیقه‌ای ۵ واحد پول برآورد می شود . هزینه مستقیم دستمزد تعمیرکار، ساعتی ۱۲ واحد پول است . تعویض یک قطعه به مصرف ۲۰ دقیقه وقت، تعویض دو قطعه به مصرف ۳۰ دقیقه وقت و تعویض سه قطعه به مصرف ۴۰ دقیقه وقت نیاز دارد. هزینه هر قطعه نیز ۱۶ واحد پول است. پیشنهادی ارائه شده است مبنی بر اینکه هر گاه یک قطعه خراب شود، هر سه قطعه تعویض شوند. مدیریت، نیاز به ارزیابی این پیشنهاد دارد.

جدول ۲-۲۲ توزیع عمر قطعه

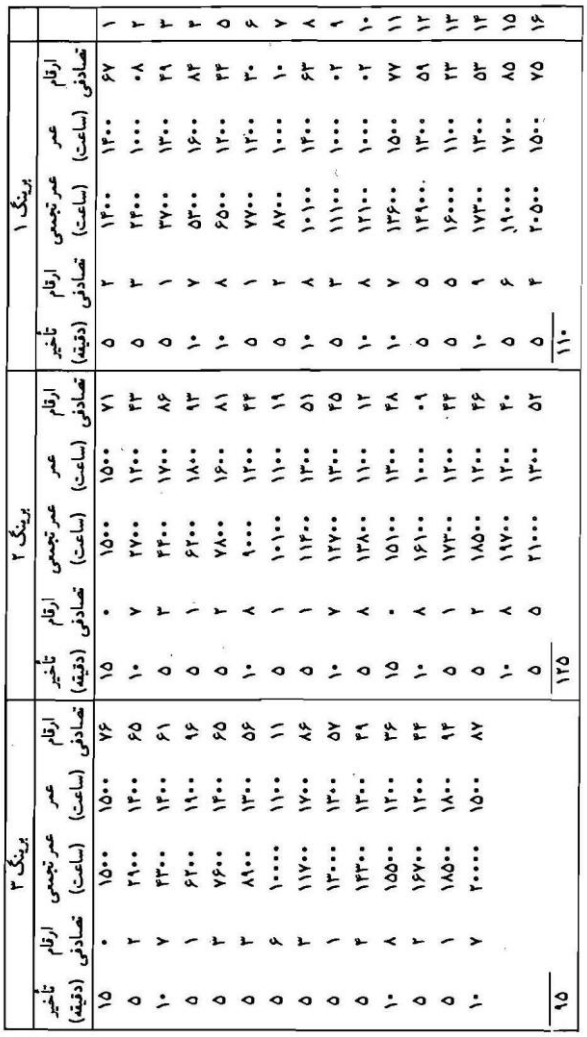
عمر قطعه (ساعت)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۱۰۰۰	۰/۱۰	۰/۱۰	۰۱-۱۰
۱۱۰۰	۰/۱۳	۰/۲۳	۱۱-۲۳
۱۲۰۰	۰/۲۵	۰/۴۸	۲۴-۴۸
۱۳۰۰	۰/۱۳	۰/۶۱	۴۹-۶۱
۱۴۰۰	۰/۰۹	۰/۷۰	۶۲-۷۰
۱۵۰۰	۰/۱۲	۰/۸۲	۷۱-۸۲
۱۶۰۰	۰/۰۲	۰/۸۴	۸۳-۸۴
۱۷۰۰	۰/۰۶	۰/۹۰	۸۵-۹۰
۱۸۰۰	۰/۰۵	۰/۹۵	۹۱-۹۵
۱۹۰۰	۰/۰۵	۱/۰۰	۹۶-۰۰

جدول ۲- ۲۳ توزیع مدت تاخیر

مدت تأخیر (دقیقه)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۵	۰/۶	۰/۶	۱-۶
۱۰	۰/۳	۰/۹	۷-۹
۱۵	۰/۱	۱/۰	۰

جدول ۲-۲۴ نتیجه شبیه سازی ۲۰۰۰۰ ساعت کار سیستم، طبق خط مشی جاری را ارائه می کند . توجه کنید که مواردی وجود دارد که در یک زمان بیش از یک قطعه خراب می شود . رخداد چنین حالتی در عمل نامحتمل است و در اینجا ناشی از فاصله های نسبتاً زیاد ۱۰۰ ساعته است. در این مثال فرض بر این است که مدتها هیچ گاه یکسان نیستند و به این ترتیب، در

جدول۲-۲۴ تعویض قطعه با استفاده از روش جاری



هر خرابی ، بیش از یک قطعه تعویض نمی شود. برای قطعه های ۱ و ۲، شانزده تعویض ولی برای قطعه ۳ تنها ۱۴ تعویض صورت گرفته است. هزینه سیستم فعلی به شرح زیر برآورد می شود:

۷۳۶= ۱۶ واحد پول برای هر قطعه × ۴۶ قطعه = هزینه قطعه ها

۱۶۵۰= ۵ واحد پول در دقیقه× (۹۵+۱۲۵+۱۱۰) دقیقه= هزینه مدت تاخیر

۴۶۰۰= ۵ واحد پول در دقیقه×۲۰ دقیقه برای هر قطعه ×۴۶ قطعه = هزینه مدت از کارماندگی حین تعمیر

۱۸۴= ۱۲ واحد پول در ۶۰ دقیقه × ۲۰ دقیقه برای هر قطعه × ۴۶ قطعه = هزینه تعمیر کار

۷۱۷۰= ۱۸۴+۴۶۰۰+ ۱۶۵۰ +۷۳۶= هزینه کل

جدول ۲-۲۵ شبیه سازی با استفاده از روش پیشنهادی است. توجه کنید که تا جایی که ممکن است عمر یکسانی برای هر سه قطعه ظاهر می‌شد. فرض بر این است که قطعات به ترتیب در قفسه‌ای قرار دارند و به دنبال یکدیگر برداشته و روی فرز نصب می شوند .

جدول۲-۲۵ تعویض قطعه با استفاده از روش پیشنهادی

	عمر برینگ ۱ (ساعت)	عمر برینگ ۲ (ساعت)	عمر برینگ ۳ (ساعت)	اولین خرابی (ساعت)	عمر تجمعی (ساعت)	ارقام تصادفی	تأخیر (دقیقه)
۱	۱۴۰۰	۱۵۰۰	۱۵۰۰	۱۴۰۰	۱۴۰۰	۳	۵
۲	۱۰۰۰	۱۲۰۰	۱۴۰۰	۱۰۰۰	۲۴۰۰	۷	۱۰
۳	۱۳۰۰	۱۷۰۰	۱۴۰۰	۱۳۰۰	۳۷۰۰	۵	۵
۴	۱۶۰۰	۱۸۰۰	۱۹۰۰	۱۶۰۰	۵۳۰۰	۱	۵
۵	۱۲۰۰	۱۶۰۰	۱۴۰۰	۱۲۰۰	۶۵۰۰	۴	۵
۶	۱۲۰۰	۱۲۰۰	۱۳۰۰	۱۲۰۰	۷۷۰۰	۳	۵
۷	۱۰۰۰	۱۱۰۰	۱۱۰۰	۱۰۰۰	۸۷۰۰	۷	۱۰
۸	۱۴۰۰	۱۳۰۰	۱۷۰۰	۱۳۰۰	۱۰۰۰۰	۸	۱۰
۹	۱۰۰۰	۱۳۰۰	۱۳۰۰	۱۰۰۰	۱۱۰۰۰	۸	۱۰
۱۰	۱۰۰۰	۱۱۰۰	۱۳۰۰	۱۰۰۰	۱۲۰۰۰	۳	۵
۱۱	۱۵۰۰	۱۳۰۰	۱۲۰۰	۱۲۰۰	۱۳۲۰۰	۲	۵
۱۲	۱۳۰۰	۱۰۰۰	۱۲۰۰	۱۰۰۰	۱۴۲۰۰	۴	۵
۱۳	۱۱۰۰	۱۲۰۰	۱۸۰۰	۱۱۰۰	۱۵۳۰۰	۱	۵
۱۴	۱۳۰۰	۱۲۰۰	۱۵۰۰	۱۲۰۰	۱۶۵۰۰	۶	۵
۱۵	۱۷۰۰	۱۲۰۰	۱۴۰۰ (۶۳)	۱۲۰۰	۱۷۷۰۰	۲	۵
۱۶	۱۵۰۰	۱۳۰۰	۱۱۰۰ (۲۱)	۱۱۰۰	۱۸۸۰۰	۷	۱۰
۱۷	۱۷۰۰ (۸۵)	۱۳۰۰ (۵۳)	۱۱۰۰ (۲۳)	۱۱۰۰	۱۹۹۰۰	۰	۱۵
۱۸	۱۰۰۰ (۰۵)	۱۲۰۰ (۲۹)	۱۳۰۰ (۵۱)	۱۰۰۰	۲۰۹۰۰	۵	۵
							۱۲۵

هر گاه از خط مشی تازه استفاده شود به ۱۸ دست قطعه نیاز بوده است. هزینه کل خط مشی تازه به شرح زیر محاسبه می شود:

۸۶۴=۱۶واحد پول به ازای هر قطعه × ۵۴ قطعه = هزینه قطعات

۶۲۵=۵ واحد پول در دقیقه ×۱۲۵ دقیقه= هزینه مدت تاخیر

۵ واحد پول در دقیقه× ۴۰ دقیقه برای هر دست ×۱۸ دست = هزینه مدت از کارماندگی حین تعمیر =۳۶۰۰

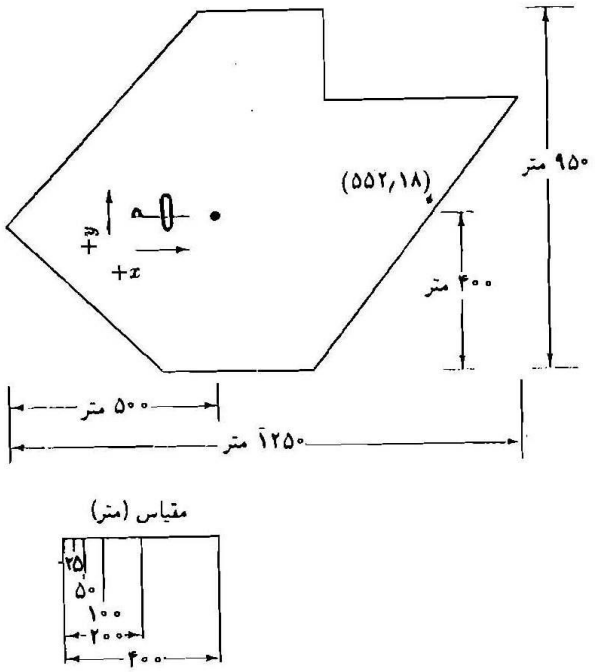
۱۴۴= ۱۲ واحد پول در ۶۰ دقیقه× ۴۰ دقیقه برای هر دست×۱۸ = هزینه تعمیر کار

۵۲۳۳= ۱۴۴+ ۳۶۰۰+ ۶۲۵+۸۶۴= هزینه کل

در شبیه سازی ۲۰۰۰۰ ساعته، با به کارگیری خط مشی تازه، معادل ۱۹۳۷ واحد پول صرفه جویی می شود. اگر ماشین به طور مداوم کار کند شبیه سازی تقریبا $2\frac{1}{4}$ سال را در بر می گیرد. بنابراین صرفه جویی سالانه حدود۸۶۰ واحد پول است.

اعداد تصادفی نرمال

یک مساله قدیمی شبیه سازی، مربوط به یک اسکادران بمب افکن است که سعی در نابود سازی یک زاعه مهمات دارد که به صورت نشان داده شده در شکل ۲-۸ است. اگر بمبی در هر نقطه زاغه فرود آید موفقیت ودر غیر این صورت شکست حاصل شده است. هواپیما در جهت افقی پرواز می کند. بمبهای پرتاب شده در جهت افقی، با انحراف معیار ۶۰۰ متر به زمین می افتند. در جهت عمودی، نیز با انحراف معیار ۳۰۰ متر می افتند. هر اسکادران ده بمب افکن دارد. نقطه ای که نشانه روی می شود، نقطه ای واقع در قلب زاغه مهمات است. هدف مساله، شبیه سازی عملیات و اظهار نظر در مورد تعداد بمبهای به هدف نشسته است.



شکل ۲-۸ زاغه مهمات

به یاد دارید که مقدار تصادفی نرمال استاندارد، Z به صورت

$$Z = \frac{X - \mu}{\sigma}$$

توزیع می شود که در آن X یک متغیر تصادفی نرمال، μ میانگین واقعی توزیع X،و σ انحراف معیار X است. بنابراین،

$$X = Z\sigma + \mu$$

نقطه هدف گیری شده می تواند دراین مثال نقطه(۰،۰) در نظر گرفته شود، یعنی مقدار μ در جهت افقی و مقدار μ در جهت عمودی صفر است. بنابراین،

$$X = z\sigma x$$
$$Y = Z\sigma Y$$

که(X,Y) مختصات شبیه سازی شده محل اصابت بمب است. حالا، $\sigma_X = 60$ و $\sigma_Y = 30$ پس

$$X = 60\mathbb{Z}_i$$
$$Y = 30\mathbb{Z}_j$$

اندیسهای j,i به این دلیل افزوده شده است که متفاوت بودن مقادیر Z را نشان دهد. این مقادیر Z چیستند و کجا می توان آنها را یافت؟ مقادیر Z اعداد تصادفی نرمال‌اند. می توان از اعداد تصادفی با توزیع یکنواخت این مقادیر را تولید کرد. راهی دیگر، به کارگیری جدولهای اعداد تصادفی نرمال تولید شده است که نمونه کوچکی از آن در جدول پ۲ ارائه شده است.

به منظور درک اینکه در این ماموریتهای بمباران چه می گذرد، یک شبیه سازی با شاید ۱۰ یا ۲۰ ماموریت می توان انجام داد، اما، محدودیت جا اجازه چنین شبیه سازی گسترده‌ای را نمی دهد. مثالی در مورد یک ماموریت نشان می‌دهد که چنین شبیه سازی هایی چگونه انجام می شود. جدول اعداد تصادفی

نرمال را همانند جدول اعداد تصادفی به کار می گیریم. یعنی کاملاً از جایی تصادفی در جدول شروع می کنیم و بدون استفاده دوباره از اعداد به طور منظم در مسیری جلو می رویم .

جدول ۲-۲۶ نتایج شبیه سازی یک ماموریت را نشان می دهد. علامت اختصاری RNN_x معرف « عدد تصادفی نرمال برای محاسبه مختصه x» و نظیر Z_i است. اولین عدد تصادفی نرمال مورد استفاده ۰/۸۴ − بود که مختصهx مساوی با ۵۰۴- = (۰/۸۴) ۶۰۰ تولید کرد. عدد تصادفی نرمال برای تولید مختصهy، ۰/۶۶ بود که به مختصهy برابر ۱۹۸ انجامید. در نظر گرفتن این دو عدد با هم، یعنی(۱۹۸، ۵۰۴-) معرف شکست است. زیرا نقطه‌ای بیرون از هدف را نشان می دهد.

جدول ۲-۲۶ شبیه سازی ماموریت بمباران

بمب‌افکن	RNN_x	مختصهٔ x ($۶۰۰ RNN_x$)	RNN_y	مختصهٔ y ($۳۰۰ RNN_y$)	نتیجه‌الف
۱	−۰/۸۴	−۵۰۴	۰/۶۶	۱۹۸	به خطا رفته
۲	۱/۰۳	۶۱۸	−۰/۱۳	−۳۹	به خطا رفته
۳	۰/۹۲	۵۵۲	۰/۰۶	۱۸	اصابت کرده
۴	−۱/۸۲	−۱۰۹۲	−۱/۴۰	−۴۲۰	به خطا رفته
۵	−۰/۱۶	− ۹۶	۰/۲۳	۶۹	اصابت کرده
۶	−۱/۷۸	−۱۰۶۸	۱/۳۳	۳۹۹	به خطا رفته
۷	۲/۰۴	۱۲۲۴	۰/۶۹	۲۰۷	به خطا رفته
۸	۱/۰۸	۶۴۸	−۱/۱۰	−۳۳۰	به خطا رفته
۹	−۱/۵۰	−۹۰۰	−۰/۷۲	−۲۱۶	به خطا رفته
۱۰	−۰/۴۲	−۲۵۲	−۰/۶۰	−۱۸۰	اصابت کرده

الف) ۳ بمب اصابت کرده، ۷ بمب به خطا رفته.

این نقطه و نقطه مربوط به بمب افکن سوم را روی شکل ۲-۸ مشخص کرده‌ایم. ده بمب افکن، سه موفقیت و هفت شکست داشته اند. به منظور ارزیابی توان نابود سازی زاغه به ماموریتهای اضافی را به صورت تمرینی برای خواننده در نظر گرفته‌ایم. این، مثالی از مونت کارلو، یا شبیه سازی ایستاست زیرا زمان عنصری مور در حل مساله نیست.

تقاضا در مهلت تحویل

تقاضا در مهلت تحویل ممکن است در سیستم موجودی‌ای مطرح شود که مهلت تحویل در آن غیر آنی باشد. مهلت تحویل مدتی است که از صدور یک سفارش تا دریافت آن به دراز می کشد. در وضعیتهای واقعی، مهلت تحویل متغیری تصادفی است. در مهلت تحویل، تقاضا نیز به صورت تصادفی رخ می نماید. بدین ترتیب، تقاضا در مهلت تحویل متغیری است تصادفی که به صورت جمع تقاضا در جریان مهلت تحویل یا $\sum_{i=0}^{T} D_i$ تعریف می شود، که i دوره زمانی، $D_i = i = 0,1,2,....$ تقاضا در i امین دوره زمانی، و T مهلت تحویل است. توزیع تقاضا در مهلت تحویل، با شبیه سازی دوره های فراوانی از مهلت تحویل و ایجاد هیستوگرمی از نتایج تعیین می شود .

شرکتی در کار فروش کاغذ روزنامه به صورت توپی است. تقاضای روزانه با توزیع احتمال زیر ارائه می شود:

تقاضای روزانه (توپ)	۳	۴	۵	۶
احتمال	۰/۲۰	۰/۳۵	۰/۳۰	۰/۱۵

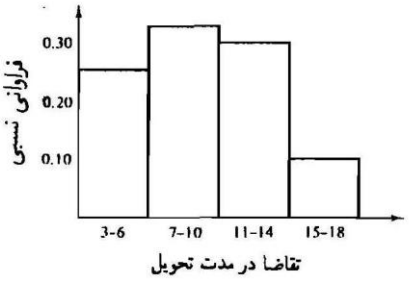
مهلت تحویل عبارت از تعداد روزهای از صدور سفارش تا دریافت آن از فروشنده توسط شرکت است. در این مورد، مهلت تحویل متغیری تصادفی است که با توزیع زیر ارائه می شود:

مدت تحویل (روز)	۱	۲	۳
احتمال	۰/۳۶	۰/۴۲	۰/۲۲

جدول۲-۲۷ تخصیص ارقام تصادفی برای تقاضا وجدول ۲-۲۸ نیز تخصیص ارقام تصادفی برای مهلت تحویل را نشان می دهد. جدول ناتمام شبیه سازی رادر جدول ۲-۲۹ نشان داده ایم ارقام تصادفی برای دوره اول۵۷ بودند که مهلت تحویل ۲ روزه را تولید می کنند . بدین ترتیب، برای تقاضای روزانه باید دو زوج ارقام تصادفی تولید کرد. زوج اول از این دو ۸۷ است و به تقاضای معادل ۶ می انجامد و یک تقاضای ۴ دنبال آن می آید . مهلت تحویل برای دوره اول ۱۰ است. پس از آنکه دوره های بسیاری شبیه سازی شوند، هیستوگرمی تعریف می شود که ممکن است همانند شکل ۲-۹ نمایان شود. این مثال نشان می دهد که چگونه شبیه سازی را می توان با تولید یک نمونه تصادفی از یک توزیع نامعلوم، در بررسی توزیع مورد استفاده قرار داد.

جدول ۲-۲۹ جدول شبیه‌سازی برای تقاضا در مهلت تحویل.

دور	ارقام تصادفی برای مهلت تحویل	مهلت تحویل (روز)	ارقام تصادفی برای تقاضا	تقاضا	تقاضا در مهلت تحویل
۱	۵۷	۲	۸۷	۶	
			۳۴	۴	۱۰
۲	۳۳	۱	۸۲	۵	۵
۳	۹۳	۳	۲۸	۴	
			۱۹	۳	
			۶۳	۵	۱۲
۴	۵۵	۲	۹۱	۶	
			۲۶	۴	۱۰
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.



شکل۲-۹ هیستوگرام برای تقاضا در مهلت تحویل.

جدول ۲-۲۷ تخصیص ارقام تصادفی برای تقاضا.

تقاضای روزانه	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۳	۰/۲۰	۰/۲۰	۰۱-۲۰
۴	۰/۳۵	۰/۵۵	۲۱-۵۵
۵	۰/۳۰	۰/۸۵	۵۶-۸۵
۶	۰/۱۵	۱/۰۰	۸۶-۰۰

جدول ۲-۲۸ تخصیص ارقام تصادفی برای مهلت تحویل.

مهلت تحویل (روز)	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۱	۰/۳۶	۰/۳۶	۰۱-۳۶
۲	۰/۴۲	۰/۷۸	۳۷-۷۸
۳	۰/۲۲	۱/۰۰	۷۹-۰۰

فصل سوم: اصول کلی و زبانهای شبیه سازی

مفاهیم مربوط به شبیه سازی گسسته پیشامد

مفهوم سیستم و مدلی از آن به اختصار در فصل ۱ مورد بحث قرار گرفت. این فصل منحصراً به سیستم های تصادفی پویایی (یعنی دارای عناصر تصادفی و مرتبط با زمان) می پردازد که به طریقی گسسته تغییر می یابند.این بخش چنین مفاهیمی را بسط می دهد چارچوبی برای ایجاد مدلی گسسته پیشامد از سیستم ارائه می کند. مفاهیم اصلی در آن به اختصار تعریف و سپس با مثال هایی تشریح می شوند.

- سیستم: مجموعه ای از نهادها (مثل آدم ها و ماشین ها) که طی زمان برهم تاثیر مقابل می گذارند تا به یک یا چند هدف نائل شوند.
- مدل: معرفی تجریدی سیستم که معمولا شامل روابط منطقی و یا ریاضی است که سیستم را بر حسب حالت، نهادها و ویژگی های آنها، مجموعه ها، پیشامدها، فعالیت ها، و تاخیرهایشان تشریح می کنند.
- حالت سیستم: جمعی از متغیرها که تمام اطلاعات لازم برای تشریح سیستم در هر لحظه را در برداشته باشد.
- نهاد: هر جزء از یک سیستم که معرفی صریح آن در مدل لازم باشد (مثلاً هر خدمت دهنده، هر مشتری، هر ماشین)
- ویژگی ها: خواص هر نهاد مفروض (مثلاً اولویت یک مشتری معین، ترتیب انجام سفارش معین در کارگاه)
- مجموعه: جمعی (دائمی یا موقتی) از نهادهای مرتبط که به طریقی منطقی آراسته شده باشد (مانند تمام مشتریانی که در حال حاضر در صف انتظارند و به ترتیب ورود یا بر حسب اولویت مرتب شده اند)
- پیشامد: رویداد لحظه ای که حالت سیستم را تغییر می دهد (مثل ورود هر مشتری جدید)
- فعالیت: فاصله ای زمانی با طول مشخص (مانند مدت خدمتدهی یا مدت بین دو ورود) که طول آن با شروعش معلوم می شود (هر چند که بتوان آن را بر حسب یک توزیع آماری تعریف کرد)
- تاخیر: فاصله ای زمانی با طول نامشخص که تا پایان نیافته است طول آن معلوم نمی شود (مانند مدت تاخیر مشتری معین در صف انتظار با نظام عکس ترتیب ورود که زمان پایان آن به ورودی های آینده بستگی خواهد داشت)

مجموعه گاهی فهرست، صف، یا زنجیره نیز نامیده می شود. هر فعالیت ممکن است قطعی(مانند یک مدت خدمتدهی که همواره ۵ دقیقه طول می کشد)، یا احتمالی (مثلاً،۳+۵ دقیقه با توزیع یکنواخت) یا هر نوع تابع ریاضی باشد. صرفنظر از اینکه کدام یک از اینها باشد، در مدل، مدت تداوم هر فعالیت به محض شروع آن قابل محاسبه است. در مقابل، هر تاخیر معمولاً وقتی پایان می یابد که یک شرط منطقی تحقق یابد چنین شرطی معمولاً ماحصل فعل و انفعال پیشامدهای بسیاری است. مدتی که یک مشتری در صف انتظار می ماند، مثالی بارز از تاخیر است. گاهی تاخیر انتظار مشروط نامیده می شود. اما فعالیت را انتظار نامشروط می نامند. توجه داشته باشید که پایان هر فعالیت پیشامدی است که اغلب پیشامد اساسی نامیده می شود.آغاز و پایان هر تاخیر، پیشامدی شرطی نامیده می شود. (آغاز هر فعالیت ممکن است پیشامدی اساسی یا شرطی باشد) واژه "پیشامد" در این کتاب به پیشامد اساسی اشاره دارد.

سیستم های مورد بررسی در اینجا پویاست، یعنی با گذشت زمان تغییر می کند. بنابراین، حالت سیستم، ویژگی ها و تعداد نهادهای فعال محتوای مجموعه ها و فعالیت ها و تاخیر در جریان همه توابعی از زمان و همواره با گذشت زمان در حال تغییر است. خود زمان با متغیری به نام CLOCK معرفی می شود.

مثال ۳-۱ (بررسی مجدد اتورستوران هابیل و خباز)

سیستم غذارسانی هابیل و خباز در مثال ۲-۲ را در نظر بگیرید. مدل گسسته دارای اجزا زیر است:

- حالت سیستم: LQ(t)، تعداد خودروهای در حال انتظار برای دریافت خدمت، در لحظه t، LA(t) معرف بیکار (صفر) یا مشغول (یک) بودن هابیل در لحظه t، LB(t) معرف بیکار (صفر) یا مشغول (یک) بودن خباز در لحظه t
- نهادها: نه مشتری ها (یعنی خودروها) و نه خدمت دهنده ها به جز در قالب متغیرهای حالت نیاز به معرفی صریح ندارد. مگر اینکه برخی متوسط های مربوط به مشتریها مدنظر باشد (مثالهای ۳-۲ و۳-۳ را مقایسه کنید)
- پیشامدها: پیشامد ورود، پیشامد تکمیل خدمتدهی توسط هابیل، پیشامد تکمیل خدمتدهی توسط خباز
- فعالیت ها: مدت بین دو ورود، به شرح جدول ۲-۱۱، مدت خدمتدهی توسط هابیل، به شرح جدول ۲-۱۲، مدت خدمتدهی توسط خباز به شرح جدول ۲-۱۳
- تاخیر: انتظار در صف تا هابیل یا خباز آزاد شود

تعریف اجزا مدل شرحی ایستا از مدل را فراهم می آورد. علاوه بر این، تشریح روابط پویا و فعل و انفعالات بین اجزا نیز مورد نیاز است. برخی از پرسشهای نیازمند پاسخ عبارت است از:

۱. چگونه هر پیشامد بر حالت سیستم، ویژگی های نهاد و محتوای مجموعه تاثیر می گذارد؟

۲. فعالیت ها چگونه تعریف می شود (یعنی، قطعی است، احتمالی است، یا نوعی دیگر از معادلات ریاضی درباره آن صدق می کند)؟ کدام پیشامد معرف شروع یا پایان هر فعالیت است؟ آیا فعالیت می تواند صرف نظر از حالت سیستم شروع شود، یا شروع شدن آن مشروط به بودن سیستم در حالت خاصی است؟ (مثلاً، فعالیت ماشینکاری را نمی توان شروع کرد مگر اینکه ماشین اولاً بیکار باشد، ثانیاً، شکسته یا در دست عملیات نگه داری و تعمیر نباشد)

۳. کدام پیشامدها آغاز و (پایان) هر نوع تاخیر را سبب می شود؟ هر تاخیر در کدام شرایط شروع یا تمام می شود؟

۴. حالت سیستم در زمان صفر چیست؟ در زمان صفر چه پیشامدهایی برای راه اندازی شبیه سازی (یعنی، برای شروع شبیه سازی) باید تولید شود؟

هر شبیه سازی گسسته پیشامد، مدلسازی طی زمان از سیستمی است که تمام تغییر حالت های آن در لحظه های گسسته زمان، یعنی در لحظه های وقوع پیشامدها رخ می دهد. شبیه سازی پیشامدهای گسسته (که از این پس شبیه سازی نامیده می شود) با ایجاد توالیی از تصاویر سیستم پیش می رود که معرف تکوین سیستم طی زمان است. هر تصویر در زمانی مفروض (CLOCK=t) نه تنها حالت سیستم در زمان t ، بلکه فهرستی (به نام فهرست پیشامدهای آتی) از تمام فعالیت های جاری در آن لحظه و زمان پایان هر فعالیت وضعیت تمام نهادها و اعضای فعلی تمام مجموعه ها و مقادیر فعلی آمارهای تجمعی و شمارشگرهایی را در بردارد، که در پایان شبیه سازی به منظور محاسبه ی آمار تلخیص شده به کار می رود.

نمونه ای از تصویر سیستم در شکل ۳-۱ نشان داده شده است. (هر مدل دارای همه عناصر نشان داده شده در شکل ۳-۱ نیست. تصاویر بیستری در مثال های موجود در این فصل ارائه خواهد شد)

ساعت	حالت سیستم	نهادها و ویژگیها	مجموعهٔ ۱	مجموعهٔ ۲	...	فهرست پیشامدهای آتی FEL	آمارهای تجمعی و شمارشگرها
t	$(x, y, z, \cdots)$					$(۳, t_۱)$– قرار است پیشامد نوع ۳ در زمان $t_۱$ رخ دهد $(۱, t_۲)$– قرار است پیشامد نوع ۱ در زمان $t_۲$ رخ دهد ⋮ ⋮	

شکل۳-۱ نمونهٔ تصویر سیستم در زمان شبیه‌سازی t.

شیوه جلو بردن زمان شبیه سازی و تضمین اینکه همه پیشامدها به ترتیب صحیح روی دهد ، بر اساس فهرست پیشامدهای آتی استوار است. این فهرست مجموعه ویژه ای است که تمام پیشامدهایی را که وقوع آنها در زمانی در آینده زمانبندی شده است در بردارد. زمانبندی هر پیشامد آتی بدین معنی است که در لحظه آغاز هر فعالیت مدت تداوم آن محاسبه (شاید به طریقی تصادفی تولید) و پیشامد انتهایی فعالیت همراه با زمان آن در فهرست پیشامدهای آتی قرار داده شود. در واقع، اکثر پیشامدهای آتی زمانبندی نمی شود بلکه صرفاً روی می دهد مانند خرابیهای تصادفی یا ورودی های تصادفی. این گونه پیشامد های تصادفی ،در مدل با پایان فعالیتی معین معرفی می شود که خود ان را با توزیعی آماری می توان معرفی کرد.

در زمان مفروض t، فهرست پیشامدهای آتی (FEL) در بردارنده تمام پیشامدهای از پیش برنامه ریزی شده برای آینده و زمان های مربوط به آنها (با نماد t1,t2...در شکل ۳-۱) است.

FEL بر حسب زمان وقوع پیشامد مرتب می شود. به عبارت دیگر، پیشامدها به ترتیب زمانی در آن جا می گیرد، یعنی زمان وقوع پیشامدها در رابطه ی

صدق می کند. زمان t مقدارCLOCK یعنی مقدار کنونی زمان شبیه سازی است.

پیشامد مربوط به زمان t1 را پیشامد قریب الوقوع می نامند، یعنی اولین پیشامدی است که رخ خواهد داد. پس از انکه تصویر سیستم در زمان شبیه سازی CLOCK=t کامل شود، CLOCK را به زمان شبیه سازی $CLOCK=t_1$ جلو می بریم و پیشامد قریب الوقوع را از FEL خارج واجرا می کنیم.

اجرای پیشامد قریب الوقوع بدین معنی است که بر اساس تصویر قبلی در زمان t و طبیعت پیشامد قریب الوقوع، تصویر تازهای از سیستم برای زمان t1 ایجاد کنیم. ممکن است در زمان t1 پیشامدهای آتی جدیدی تولید شود. اگر چنین باشد انها را با قرار دادن در موقعیت مناسب در FEL زمانبندی می کنیم. پس از انکه تصویر تازه ی سیستم در زمان t1 کامل شد ساعت را به زمان پیشامد قریب الوقوع جدید جلو می بریم و این پیشامد را راجرا می کنیم. این فرایند را انقدر تکرار می کنیم که شبیه سازی تمام شود. توالی اعمالی را که یک شبیه ساز (یا زبان شبیه سازی) برای جلو بردن ساعت و ایجاد تصویر تازه ای از سیستم انجام می دهد الگوریتم زمانبندی پیشامدها و جلوبری زمان می نامند که گام های آن در شکل ۳-۲ آورده شده است (و در زیر تشریح می شود).

با پیشرفت شبیه سازی، طول و محتوای FEL به طور مداوم در حال تغییر است و بدین ترتیب، اداره آن به طریقی کارا در یک شبیه سازی کامپیوتری، تاثیری عمده بر کارایی برنامه کامپیوتری معرف مدل دارد. عملیات اصلی پردازش فهرست که روی FEL صورت می گیرد عبارت است از خارج کردن پیشامد قریب الوقوع، افزودن یک پیشامد جدید به ان و گهگاه حذف پیشامدی از ان (مشهور به منتفی شدن یک پیشامد).

چون پیشامد قریب الوقوع معمولاً در راس فهرست قرار دارد، خارج کردن ان با بیشترین کارایی ممکن همراه است. افزودن پیشامدی تازه (و حذف پیشامدی قدیمی به سبب منتفی شدن آن) نیاز به جستجو در فهرست دارد. کارایی اینگونه جستجو به ترتیب منطقی فهرست و چگونگی انجام جستجو بستگی دارد. علاوه بر FEL تمام مجموعه ها در یک مدل به ترتیبی منطقی نگه داری می شود و عملیات افزودن و خارج کردن نهادها از مجموعه، معمولاً به شیوه های کارای پردازش فهرست نیازمند است.

خارج کردن پیشامدها از FEL افزودن پیشامدها به آن در شکل ۳-۲ نمایش داده شده است. مثلاً پیشامد ۳ با زمان وقوع t_1 معرف پیشامد تکمیل خدمتدهی از سوی خدمتدهنده ۳ است. چون این پیشامد قریب الوقوع در زمان t است در گام ۱ از الگوریتم زمانبندی از FEL خارج می شود. وقتی که پیشامد ۴ با زمان وقوع t^* در گام ۴ تولید شود، یک راه ممکن برای تعیین موقعیت آن در FEL انجام یک جستجوی بالا به پایین است:

- اگر $t^* < t_2$ است، پیشامد ۴ را در رأس FEL قرار دهید.
- اگر $t_2 \leq t^* < t_3$ است، پیشامد ۴ را در موقعیت دوم فهرست قرار دهید.

تصویر قبلی سیستم در زمان t

CLOCK	حالت سیستم	...	فهرست پیشامدهای آتی	...
t	(۵, ۱, ۶)		($۳, t_1$)- پیشامد نوع ۳ در زمان t_1 رخ می‌دهد ($۱, t_2$)- پیشامد نوع ۱ در زمان t_2 رخ می‌دهد ($۱, t_3$)- پیشامد نوع ۱ در زمان t_3 رخ می‌دهد ⋮ ⋮ ⋮ ($۲, t_n$)- پیشامد نوع ۲ در زمان t_n رخ می‌دهد	

الگوریتم زمانبندی پیشامدها و جلوبری زمان

گام ۱. پیشامد قریب‌الوقوع (پیشامد ۳، زمان t_1) را از FEL خارج کنید.
گام ۲. CLOCK را به زمان پیشامد قریب‌الوقوع جلو ببرید (یعنی CLOCKرا از t به t_1 جلو ببرید).
گام ۳. پیشامد قریب‌الوقوع را اجرا کنید: حالت سیستم را تازه کنید. ویژگیهای نهاد و اعضای مجموعه‌ها را بر حسب نیاز تغییر دهید.
گام ۴. (در صورت لزوم) پیشامدهای آتی تولید و در موقعیت صحیح در FEL قرار دهید. (مثال: پیشامد ۴ قرار است در زمان t^* رخ دهد، به‌طوری که $t_3 < t^* < t_2$ باشد).
گام ۵. آمارهای تجمعی و شمارشگرها را تازه کنید.

تصویر جدید سیستم در زمان t_1

CLOCK	حالت سیستم	...	فهرست پیشامدهای آتی	...
t_1	(۵, ۱, ۵)		($۱, t_2$)- پیشامد نوع ۱ در زمان t_2 رخ می‌دهد. ($۴, t^*$)-پیشامد نوع ۴ در زمان t^* رخ می‌دهد. ($۱, t_3$)- پیشامد نوع ۱ در زمان t_3 رخ می‌دهد. ⋮ ⋮ ⋮ ($۲, t_n$)-پیشامد نوع ۲ در زمان t_n رخ می‌دهد.	

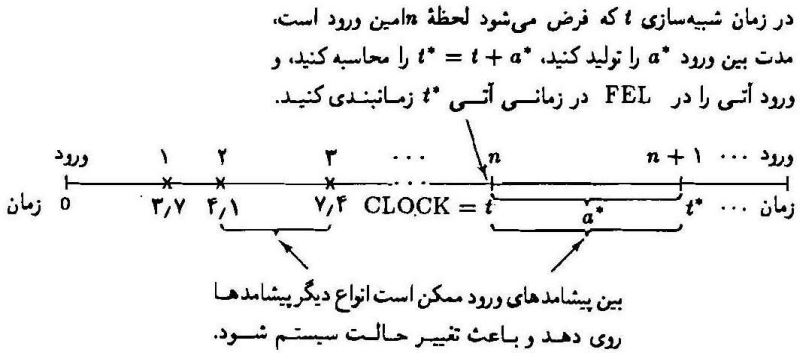
شکل ۳-۲ جلوبری زمان شبیه‌سازی و تازه کردن تصویر سیستم.

- اگر $t_3 \leq t^* < t_4$ است، پیشامد ۴ را در موقعیت سوم فهرست قرار دهید.
- ...
- اگر $t_n \leq t^*$ است، پیشامد ۴ را در آخر FEL قرار دهید.

راه دیگر انجام یک جستجوی از پایین به بالا ست. راهی برای نگهداری FEL که کمترین کارایی را دارد، رها کردن آن به صورت یک فهرست مرتب شده است. که در گام ۱ شکل ۳-۲ تعیین پیشامد قریب الوقوع پیش از هر مورد جلوبری ساعت، نیاز به جستجوی کامل در فهرست دارند.

تصویر سیستم در زمان صفر با مشخص کردن شرایط اولیه و تولید پیشامدهای به اصطلاح برونزا تعریف می شود. شرایط اولیه مشخص شده، حالت سیستم در زمان صفر را تعریف می کند. مثلاً اگر در شکل ۳-۲، $t = 0$ باشد، حالت (۵،۱،۶) را می توان معرف تعداد اولیه مشتری ها در سه نقطه مختلف در سیستم قرار دارد. پیشامد برونزا، رخدادی "خارج از سیستم" است که به سیستم نفوذ می کند.

یک مثال مهم، ورود به سیستم صف است. اولین پیشامد ورود در زمان صفر تولید و در FEL برنامه ریزی می شود. مدت بین دو ورود مثالی در مورد فعالیت است. سرانجام وقتی ساعت به زمان این اولین ورود جلو برده می شود، یک پیشامد دوم ورود تولید می شود. ابتدا، یک مدت بین دو ورود مانند a^* تولید و سپس به زمان کنونی، مثلاً $CLOCK = t$، افزوده می شود. زمان بدست آمده، $t + a^* = t^*$، برای پیشامد (آتی)، به منظور تعیین موقعیت پیشامد جدید ورود در FEL مورد استفاده قرار می گیرد. این روش تولید رشته ای از ورودهای خارجی روش خود راه انداز نامیده می شود و مثالی از چگونگی تولید پیشامدهای آتی در گام ۴ الگوریتم زمانبندی پیشامدها و جلوبری زمان را ارائه می کند. روش خود راه انداز در شکل ۳-۳ نمایش داده شده است. اولین سه مدت بین دو ورود تولید شده، ۳/۷، ۰/۴، ۳/۳ واحد زمان است. شروع و پایان هر فاصلهی بین دو ورود، مثال هایی از پیشامدهای اساسی اند.



شکل ۳-۳ تولید رشته‌ای از ورودهای خارجی با استفاده از روش خود راه‌انداز.

مثال دوم از چگونگی تولید پیشامدهای آتی (گام ۴ از شکل ۳-۲)، در قالب پیشامد تکمیل خدمتدهی در شبیه سازی یک سیستم صف معین ارائه می شود. هرگاه مشتری خدمتگیری را، مثلاً در زمان فعلی $CLOCK = t$ کامل کند در صورتی که مشتری بعدی حاضر باشد یک مدت جدید خدمتدهی مثل s^* برای او تولید خواهد شد. وقوع پیشامد بعدی تکمیل خدمتدهی در زمان $t^* = t + s^*$، با وارد کردن یک پیشامد تکمیل خدمتدهی با زمان رویداد t^* در FEL زمانبندی خواهد شد. به علاوه در زمان وقوع هر پیشامد ورود، یک پیشامد تکمیل خدمتدهی تولید و زمانبندی می شود مشروط بر اینکه به هنگام ورود، دست کم یک خدمتدهنده بیکار در گروه خدمت دهندگان وجود داشته باشد. مدت خدمتدهی نیز مثالی از فعالیت است. آغاز خدمتدهی پیشامدی شرطی است، زیرا صرفاً با این شرط شروع می شود که یک مشتری در سیستم حاضر و یک خدمتدهنده آزاد باشد. تکمیل خدمتدهی مثالی از پیشامد اساسی است. توجه کنید که هر پیشامد شرطی، از قبیل آغاز خدمتدهی، با وقوع یک پیشامد اساسی و وجود برخی شرایط حاکم در سیستم رخ می دهد.

سومین مثال مهم، تولید متناوب مدت های کار و مدتهای از کارماندگی در سراسر مدت شبیه سازی به صورت متناوب قرار می گیرد. مدت کار و مدت از کارماندگی مثال هایی از فعالیت و پایان مدت کار و پایان مدت از کارماندگی پیشامدهای اساسی است.

هر شبیه سازی باید دارای یک پیشامد پایان اجرا باشد که در اینجا با نماد E معرفی شده است. این پیشامد تعیین می کند که شبیه سازی به چه مدت اجرا می شود. به طور کلی برای به پایان بردن هر شبیه سازی دو راه وجود دارد:

۱. در زمان صفر، یک پیشامد پایان شبیه سازی را برای زمانی در آینده مانند T_E زمانبندی کنید. بنابراین، پیش از شبیه سازی معلوم است که شبیه سازی در فاصله زمانی $[0, T_E]$ اجرا خواهد شد. مثال: شبیه سازی یک کارگاه به مدت $T_E = 40$ ساعت.

۲. مدت اجرا، T_E، توسط خود شبیه سازی تعیین می شود. به طور کلی T_E زمان وقوع پیشامد معینی مانند E است. چند مثال: T_E زمان تکمیل صدمین خدمتدهی در یک مرکز است. T_E زمان خراب شدن یک سیستم پیچید هاست. T_E زمان پایان درگیری یا نابودی کامل در یک شیه سازی جنگی است.

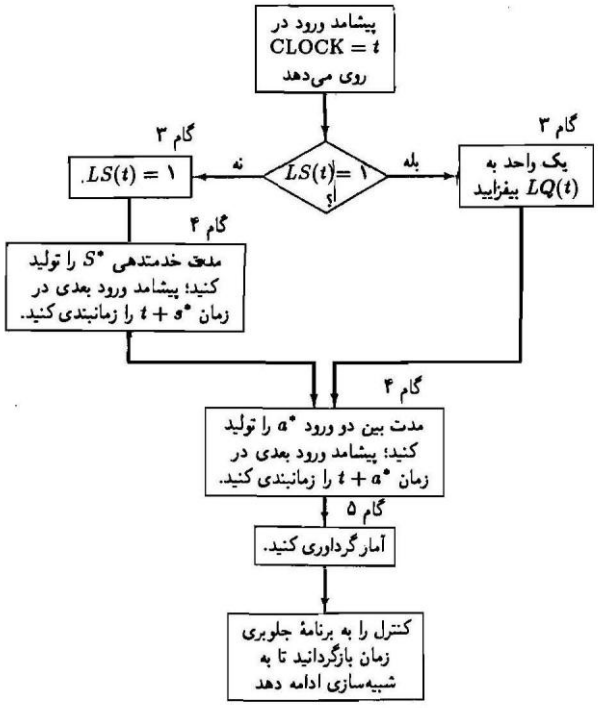
مثال ۳-۲ صف تک مجرایی

بار دیگر فروشگاه مواد غذایی با یک باجه صندوق را در نظر بگیرید که در مثال ۲-۱ با روشی موردی و مخصوص شبیه سازی شد. سیستم مشتریانی که در صف انتظارند به اضافه کسانی که در محل صندوق مشغول پرداخت وجه است، را در برمی گیرد.

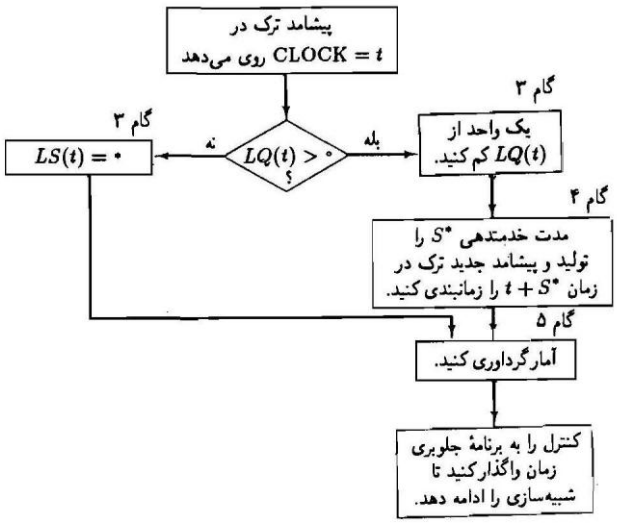
مدل دارای اجزاء زیر است:

- حالت سیستم. $(LQ(t),LS(t))$، که $LQ(t)$ تعداد مشتریان در صف انتظار و $LS(t)$ تعداد مشتری در حال خدمت گیری (۰ یا ۱) در زمان t است.
- نهادها. خدمت دهنده و مشتریان صریحاً مدلسازی نمی شوند، مگر در قالب متغیرهای حالت به شرح بالا.
- پیشامدها. ورود (A)، ترک (D)، پیشامد پایان اجرا (E) که وقوع آن برای زمان ۶۰ زمانبندی شده است.
- فعالیتها. مدت بین دو ورود، به شرح جدول ۲-۶، مدت خدمت دهی به شرح جدول ۲-۷.
- تأخیر. مدت ماندن مشتری در صف انتظار.

پیشامدها در FEL به صورت (زمان پیشامد، نوع پیشامد) نوشته می شوند. FEL در این مدل همواره ۲ یا ۳ پیشامد خواهد داشت. تأثیر پیشامدهای ورود و ترک، نخست در شکل های ۲-۲ و ۲-۳ نشان داده شد و با تفصیل بیشتری در شکلهای ۳-۵ و ۳-۶ نشان د اده می شود.



شکل ۳-۵ اجرای پیشامد ورود.



جدول ۳-۱ جدول شبیه‌سازی برای باجهٔ صندوق (مثال ۳-۲).

ساعت	حالت سیستم $LS(t)$	حالت سیستم $LQ(t)$	فهرست پیشامدهای آتی	توضیحات	آمار تجمعی B	آمار تجمعی MQ
۰	۱	۰	(D, ۴), (A, ۸), (E, ۶۰)	Aی اول رخ می‌دهد (۸ = a^*)Aی بعدی را زمانبندی کنید (۴ = s^*)Dی اول را زمانبندی کنید	۰	۰
۴	۰	۰	(A, ۸), (E, ۶۰)	Dی اول رخ می‌دهد: (D, ۴)	۴	۰
۸	۱	۰	(D, ۹), (A, ۱۴), (E, ۶۰)	Aی دوم رخ می‌دهد: (A, ۸) (۶ = a^*)Aی بعدی را زمانبندی کنید (۱ = s^*)Dی بعدی را زمانبندی کنید	۴	۰
۹	۰	۰	(A, ۱۴), (E, ۶۰)	Dی دوم رخ می‌دهد: (D, ۹)	۵	۰
۱۴	۱	۰	(A, ۱۵), (D, ۱۸), (E, ۶۰)	Aی سوم رخ می‌دهد: (A, ۱۴) (۴ = s^*)Dی بعدی را زمانبندی کنید	۵	۰
۱۵	۱	۱	(D, ۱۸), (A, ۲۳), (E, ۶۰)	Aی چهارم رخ می‌دهد: (A, ۱۵) (مشتری به انتظار می‌ماند)	۶	۱
۱۸	۱	۰	(D, ۲۱), (A, ۲۳), (E, ۶۰)	Dی سوم رخ می‌دهد: (D, ۱۸) (۳ = s^*)Dی بعدی را زمانبندی کنید	۹	۱
۲۱	۰	۰	(A, ۲۳), (E, ۶۰)	Dی چهارم رخ می‌دهد: (D, ۲۱)	۱۲	۱

جدول شبیه سازی باجه صندوق در جدول ۳-۱ ارائه شده است. دانشجو باید با شروع از تصویر اول تمام تصاویر سیستم به جز یکی را بررسی و سعی کند تصویر بعدی را از تصویر قبلی و منطق پیشامدهای شکل های ۳-۵ و ۳-۶ به وجود آورد. مدت های بین دو ورود و مدتهای خدمت دهی شبیه مدت های مورد استفاده در جدول ۲-۱۰ است، یعنی:

شرایط اولیه چنین است که اولین مشتری در زمان صفر وارد و خدمتدهی به او شروع می شود. این امر در جدول ۳-۱ و در قالب تصویر سیستم در زمان صفر (CLOCK = 0) به گونه ای منعکس است که LQ(0)=0، LS(0)=1 و یک پیشامد ترک و یک پیشامد ورود با هم در FEL است. شبیه سازی نیز به گونه ای زمانبندی شده است که در زمان ۶۰ متوقف شود.

تنها دو نوع آمار، یعنی بهره برداری از خدمت دهنده و بزرگترین طول صف گردآوری خواهد شد. بهره برداری از خدمت دهنده به صورت جمع مدت اشتغال خدمت دهنده (B) تقسیم بر مدت کل (T_E) تعریف می شود. همچنان که شبیه سازی پیشرفت می کند، جمع مدت اشتغال، B، و بزرگترین طول صف، MQ، تازه خواهد شد.

در این جدول a^* و s^* به ترتیب مدت های بین دو ورود و خدمت دهی است. به محض کامل شدن تصویر سیستم در زمان CLOCK = 0 شبیه سازی آغاز می شود. در زمان صفر پیشامد قریب الوقوع، (D,4) است. CLOCK به زمان ۴ جلو برده و (D,4) از FEL بیرون برده می شود. چون به ازای $0 < t < 4$، LS(t) = 1 است (یعنی خدمت دهنده به مدت ۴ دقیقه مشغول بوده است)، مدت اشتغال تجمعی از B=0 به B=4 افزایش می یابد، طبق منطق پیشامد در شکل ۳-۶ LS(4) را مساوی صفر قرار دهید (خدمت دهنده بیکار می شود). FEL تنها با دو پیشامد آتی، (A, 8) و (E, 60) باقی می ماند سپس CLOCK شبیه سازی به زمان ۸ جلو برده و یک پیشامد ورود اجرا می شود.

شبیه سازی در جدول ۳-۱ فاصله زمانی [۰.۲۱] را می پوشاند در زمان شبیه سازی شده ۲۱، سیستم خالی است ولی ورود بعدی در زمان آینده ی ۲۳ روی خواهد داد. از ۲۱ واحد زمان شبیه سازی شده، خدمت دهنده ۱۲ واحد زمان را مشغول بوده که برای کسب نتایج اعتماد پذیر بسیار کوتاه است.

وقتی یک الگوریتم زمانبندی پیشامدها کامپیوتری می شود، در حافظه کامپیوتر تنها یک تصویر (تصویر کنونی یا تصویری که بخش هایی از آن تازه شده است) نگهداری می شود. یک تصویر تازه تنها ممکن است با استفاده از تصویر قبلی، متغیرهای تصادفی تازه تولید شده و منطق پیشامدها (شکل های ۳-۵ و ۳-۶) ایجاد گردد. به هنگام جلو بردن ساعت تصاویر، گذشته را باید نادیده گرفت. تصویر کنونی نیز باید حاوی تمام اطلاعات لازم برای ادامه دادن شبیه سازی باشد.

مثال ۳-۳ (ادامه شبیه سازی باجهی صندوق)

فرض کنید که در شبیه سازی باجه صندوق در مثال ۳-۲، شبیه ساز مایل است میانگین مدت پاسخ و میانگین نسبت مشتریانی که ۴ دقیقه یا بیشتر در سیستم می مانند را براورد کند. یک مدت پاسخ، مدتی است که یک مشتری در سیستم می ماند. به منظور براورد کردن تعداد این مشتریان لازم است مدل مثال ۳-۲ را چنان گسترش دهیم که هر مشتری را صریحاً معرفی کند. به علاوه برای اینکه بتوان به هنگام ترک سیستم از سوی یک مشتری مدت پاسخ او را محاسبه کرد دانستن زمان ورود مشتری لازم خواهد بود.

بنابراین یک نهاد مشتری با ویژگی زمان ورود به فهرست اجزاء مدل در مثال ۳-۲ افزوده می شود. این گونه نهادهای مشتری را که در مجموعه ای به نام "زمان ورود" ذخیره می شود ... ,C1, C2, C3 می نامیم. سرانجام نمادهای مربوط به پیشامدها در FEL را برای نشان دادن اینکه کدام مشتری تحت تأثیر قرار می گیرد، گسترش می دهیم. مثلاً (D, 4, C1) بدین معنی است که مشتری C1 در زمان ۴ سیستم را ترک خواهد کرد. فهرست اجزاء اضافی مدل به شرح زیر است:

- نهادها. (Ci, t) معرف مشتری Ci است که در زمان t وارد شد.
- پیشامدها. (A, t.Ci) ورود مشتری Ci در زمان t و (D, t, Cj) ترک مشتری Cj در زمان t.
- مجموعه. "زمان ورود" مجموعه تمام مشتریانی که در حال حاضر در سیستم اند (در حال خدمت گیری یا به انتظار خدمت) به ترتیب زمان ورود.

سه قلم آمار تازه گردآوری می کنیم: S، مجموع مدتهای پاسخ برای تمام مشتریانی که تا زمان کنونی سیستم را ترک کرده اند. F جمع تعداد مشتریانی که ۴ دقیقه یا بیشتر در سیستم می مانند، و N_D جمع موارد ترک سیستم تا زمان کنونی شبیه سازی. هرگاه پیشامد ترک روی دهد، این سه قلم آمار تجمعی را تازه می کنیم. منطق مربوط به گردآوری این اقلام آماری را در گام ۵ پیشامد ترک در شکل ۳-۶ شرکت می دهیم.

جدول ۳-۲ جدول شبیه‌سازی برای مثال ۳-۲.

ساعت	حالت سیستم LS(t)	حالت سیستم LQ(t)	مجموعهٔ «زمان ورود»	فهرست پیشامدهای آتی	آمار تجمعی S	آمار تجمعی N_D	آمار تجمعی F
۰	۱	۰	(C۱, ۰)	(D, ۴, C۱), (A, ۸, C۲), (E, ۶۰)	۰	۰	۰
۴	۰	۰		(A, ۸, C۲), (E, ۶۰)	۴	۱	۱
۸	۱	۰	(C۲, ۸)	(D, ۹, C۲), (A, ۱۴, C۳), (E, ۶۰)	۴	۱	۱
۹	۰	۰		(A, ۱۴, C۳), (E, ۶۰)	۵	۲	۱
۱۴	۱	۰	(C۳, ۱۴)	(A, ۱۵, C۴), (D, ۱۸, C۳), (E, ۶۰)	۵	۲	۱
۱۵	۱	۱	(C۳, ۱۴), (C۴, ۱۵)	(D, ۱۸, C۳), (A, ۲۳, C۵), (E, ۶۰)	۵	۲	۱
۱۸	۱	۰	(C۴, ۱۵)	(D, ۲۱, C۴), (A, ۲۳, C۵), (E, ۶۰)	۹	۳	۲
۲۱	۰	۰		(A, ۲۳, C۵), (E, ۶۰)	۱۵	۴	۳

جدول شبیه سازی برای مثال ۳-۳ در جدول ۳-۲ نشان داده شده است. برای مدتهای بین دو ورود و خدمتدهی از همان داده ها مجدداً استفاده می کنیم. به طوری که جدول ۳-۲ اساساً به صورت تکرار جدول ۳-۱ در امده است. با این استثنا که اجزاء جدید نیز در آن گنجانیده شده است. این اجزاء جدید برای محاسبه S، F و N_D مورد نیاز است. مثلاً در زمان ۴ یک پیشامد ترک در مورد مشتری C1 رخ می دهد. نهاد مشتری C1 از مجموعه "زمان ورود" بیرون آورده می شود و همانطور که توجه دارید ویژگی "زمان ورود" صفر است. یعنی مدت پاسخ دهی برای مشتری ۴ دقیقه بوده است. بنابراین S به مقدار ۴ دقیقه و F و N_D هر یک به اندازه یک مشتری زیاد می شوند. همچنین به هنگام اجرای پیشامد ترک(D, 21, C4) در زمان ۲۱ مدت پاسخ مشتری C4 به صورت

مدت پاسخ – CLOCK = ویژگی "زمان ورود" = ۱۵ – ۲۱ = دقیقه ۶

محاسبه می شود. سپس، S به مقدار ۶ دقیقه و F و N_D به تعداد یک مشتری زیاد می شوند.

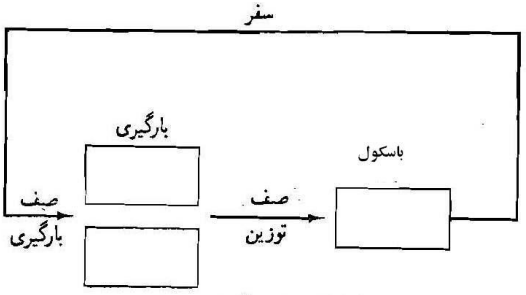
برای شبیه سازی با طول اجرای ۲۱ دقیقه، متوسط مدت پاسخ، $\frac{S}{N_D} = \frac{15}{4} = 3.75$ دقیقه و نسبت مشاهده شده مشتریانی که ۴ دقیقه یا بیشتر در سیستم ماندند، $\frac{F}{N_D} = 0.75$ بود.

مثال ۳-۴ (مسأله کامیونها)

شش کامیون برای حمل زغال از مدخل یک معدن کوچک به راه آهن مورد استفاده قرار دارند. شکل ۳-۷ شمایی از عملیات کامیون را ارائه می دهد. هر کامیون به وسیله یکی از دو دستگاه بارگیری با می گیرد و بلافاصله پس از بارگیری به سمت باسکول می رود تا در اسرع وقت توزین آن انجام شود. دستگاه های بارگیری و باسکول هر دو دارای صفهای انتظار به ترتیب ورود برای کامیونهاست. مدت سفر از دستگاه بارگیری به باسکول قابل اغماض می باشد. پس از تعیین وزن، هر کامیون سفری را شروع می کند (که در این مدت، کامیون بار خود را تخلیه می کند) و سپس به صف بارگیری باز می گردد. توزیع های مدت بارگیری، مدت توزین و مدت سفر، به ترتیب در جدولهای ۳-۳، ۳-۴ و ۳-۵ همراه با تخصیص ارقام تصادفی برای تولید این متغیرها با استفاده از ارقام تصادفی جدول پ ۱ ارائه شده است. مقصود از شبیه سازی، براورد کردن درصد مدت اشتغال هر دستگاه بارگیری و باسکول است.

مدل دارای اجزاء زیر است:

- حالت سیستم. [LQ(t),L(t),WQ(t),W(t)] به طوری که در زمان شبیه سازی t، LQ(t)= تعداد کامیون ها در صف بارگیری، L(t)= تعداد کامیون های در حال بارگیری (صفر، ۱ یا ۲)، W(t)= تعداد کامیون ها در صف توزین، W(t) = تعداد کامیون ها در حال توزین (صفر یا یک).
- پیشامدها. (ALQ ,t , DTi)، کامیون i در زمان t به صف بارگیری وارد می شود، (EL, t , DTi)، بارگیری کامیون i در زمان t به اتمام می رسد، (EW, t , DTi)، توزین کامیون i در زمان t به اتمام می رسد.
- نهادها. شش کامیون (DT1,DT2,…,DT6).
- مجموعه ها. صف بارگیری، تمام کامیون های منتظر برای شروع بارگیری که به ترتیب ورود مرتب شده است، صف توزین، تمام کامیون های منتظر برای توزین که به ترتیب ورود مرتب شده است.
- فعالیت ها. مدت بارگیری، مدت توزین، مدت سفر.
- تأخیرها. تأخیر در صف بارگیری و تأخیر در محل باسکول.



شکل ۳-۷ مسألهٔ کامیونها.

جدول ۳-۳ توزیع مدت بارگیری برای کامیونها.

مدت بارگیری	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۵	۰/۳۰	۰/۳۰	۱-۳
۱۰	۰/۵۰	۰/۸۰	۴-۸
۱۵	۰/۲۰	۱/۰۰	۹-۰

z

جدول ۳-۴ توزیع مدت توزین برای کامیونها.

مدت توزین	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۱۲	۰٫۷۰	۰٫۷۰	۱-۷
۱۶	۰٫۳۰	۱٫۰۰	۸-۰

جدول ۳-۵ توزیع مدت سفر برای کامیونها.

مدت سفر	احتمال	احتمال تجمعی	تخصیص ارقام تصادفی
۴۰	۰٫۴۰	۰٫۴۰	۱-۴
۶۰	۰٫۳۰	۰٫۷۰	۵-۷
۸۰	۰٫۲۰	۰٫۹۰	۸-۹
۱۰۰	۰٫۱۰	۱٫۰۰	۰

جدول ۳-۶ جدول شبیه‌سازی برای عملیات کامیونها (مثال ۳-۴).

ساعت	حالت سیستم				مجموعه‌ها		فهرست پیشامدهای آتی	آمار تجمعی	
	$W(t)$	$WQ(t)$	$L(t)$	$LQ(t)$	صف توزین	صف بارگیری		B_S	B_L
t									
۰	۱	۰	۲	۳	DT۴		(EL, ۵, DT۳)	۰	۰
					DT۵		(EL, ۱۰, DT۲)		
					DT۶		(EW, ۱۲, DT۱)		
۵	۱	۱	۲	۲	DT۵	DT۳	(EL, ۱۰, DT۲)	۱۰	۵
					DT۶		(EL, ۵ + ۵, DT۴)		
							(EW, ۱۲, DT۱)		
۱۰	۱	۲	۲	۱	DT۶	DT۳	(EL, ۱۰, DT۴)	۲۰	۱۰
						DT۲	(EW, ۱۲, DT۱)		
							(EL, ۱۰ + ۱۰, DT۵)		
۱۰	۱	۳	۲	۰		DT۳	(EW, ۱۲, DT۱)	۲۰	۱۰
						DT۲	(EL, ۲۰, DT۵)		
						DT۴	(EL, ۱۰ + ۱۵, DT۶)		
۱۲	۱	۲	۲	۰		DT۲	(EL, ۲۰, DT۵)	۲۴	۱۲
						DT۴	(EW, ۱۲ + ۱۲, DT۳)		
							(EL, ۲۵, DT۶)		
							(ALQ, ۱۲ + ۶۰, DT۱)		
۲۰	۱	۳	۱	۰		DT۲	(EW, ۲۴, DT۳)	۴۰	۲۰
						DT۴	(EL, ۲۵, DT۶)		
						DT۵	(ALQ, ۷۲, DT۱)		
۲۴	۱	۲	۱	۰		DT۴	(EL, ۲۵, DT۶)	۴۴	۲۴
						DT۵	(EW, ۲۴ + ۱۲, DT۲)		
							(ALQ, ۷۲, DT۱)		
							(ALQ, ۲۴ + ۱۰۰, DT۳)		
۲۵	۱	۳	۰	۰		DT۴	(EW, ۳۶, DT۲)	۴۵	۲۵
						DT۵	(ALQ, ۷۲, DT۱)		
						DT۶	(ALQ, ۱۲۴, DT۳)		
۳۶	۱	۲	۰	۰		DT۵	(EW, ۳۶ + ۱۶, DT۴)	۴۵	۳۶
						DT۶	(ALQ, ۷۲, DT۱)		
							(ALQ, ۳۶ + ۴۰, DT۲)		
							(ALQ, ۱۲۴, DT۳)		
۵۲	۱	۱	۰۲	۰		DT۶	(EW, ۵۲ + ۱۲, DT۵)	۴۵	۵۲
							(ALQ, ۷۲, DT۱)		
							(ALQ, ۷۶, DT۲)		
							(ALQ, ۵۲ + ۴۰, DT۴)		
							(ALQ, ۱۲۴, DT۳)		

جدول ۳-۶ (ادامه) جدول شبیه‌سازی برای عملیات کامونها (مثال ۳-۴).

ساعت	حالت سیستم				مجموعه‌ها			فهرست پیشامدهای آتی	آمار تجمعی	
	LQ(t)	L(t)	WQ(t)	W(t)	صف توزین	صف بارگیری			B_L	B_S
t										
۶۴	۰	۰	۰	۱				(ALQ, ۷۲, DT۱)	۴۵	۶۴
								(ALQ, ۷۶, DT۲)		
								(EW, ۶۴ + ۱۶, DT۶)		
								(ALQ, ۹۲, DT۴)		
								(ALQ, ۱۲۴, DT۳)		
								(ALQ, ۶۴ + ۸۰, DT۵)		
۷۲	۰	۱	۰	۱				(ALQ, ۷۶, DT۲)	۴۵	۷۲
								(EW, ۸۰, DT۶)		
								(EL, ۷۲ + ۱۰, DT۱)		
								(ALQ, ۹۲, DT۴)		
								(ALQ, ۱۲۴, DT۳)		
								(ALQ, ۱۴۴, DT۵)		
۷۶	۰	۲	۰	۱				(EW, ۸۰, DT۶)	۴۹	۷۶
								(EL, ۸۲, DT۱)		
								(EL, ۷۶ + ۱۰, DT۲)		
								(ALQ, ۹۲, DT۴)		
								(ALQ, ۱۲۴, DT۳)		
								(ALQ, ۱۴۴, DT۵)		

جدول شبیه سازی در جدول ۳-۶ ارائه شده است. فرض بر این است که در زمان صفر، پنج کامیون در قسمت بارگیری و یک کامیون در قسمت توزین است. مدت های فعالیت بسته به نیاز از فهرست زیر استخراج می شود:

مدت بارگیری	۱۰	۵	۵	۱۰	۱۵	۱۰	۱۰		
مدت توزین	۱۲	۱۲	۱۲	۱۶	۱۲	۱۶			
مدت سفر	۶۰	۱۰۰	۴۰	۴۰	۸۰				

هرگاه پیشامد اتمام بارگیری (EL)، مثلاً برای کامیون j در زمان t رخ می دهد، ممکن است این رویداد پیشامدهای دیگری را سبب شود. اگر باسکول بیکار باشد [W(t)=0] توزین کامیون j را آغاز و یک پیشامد اتمام توزین (EW) در FEL زمانبندی می کنیم، در غیر این صورت، کامیون j به صف توزین می پیوندد. اگر در این زمان کامیون دیگری به انتظار یک دستگاه بارگیری باشد، آن را از صف بارگیری خارج و با زمانبندی یک پیشامد اتمام بارگیری (EL) در FEL بارگیری آن را شروع می کنیم. چنین منطقی برای وقوع پیشامد اتمام بارگیری به اضافه منطق مناسب برای دو پیشامد دیگر باید در نمودارهای جریان مربوط به پیشامدها، همانند شکل های ۳-۵ و ۳-۶ شرکت داده شود.

برای کمک به خواننده هرگاه پیشامد جدیدی در جدول ۳-۶ زمانبندی شده، زمان پیشامد آن به صورت "مدت فعالیت+t" نوشته شده است. مثلاً، پیشامد قریب الوقوع در زمان صفر، یک پیشامد EL با زمان پیشامد ۵ است. ساعت به زمان t=5 جلو برده می شود. کامیون ۳ به صف توزین می پیوندد (زیرا باسکول مشغول است)، و کامیون ۴ شروع به بارگیری می کند.

بنابراین برای کامیون ۴ یک پیشامد EL در زمان آتی ۱۰ زمانبندی می شود که محاسبه این زمان به صورت ۱۰= ۵ + ۵ = (مدت بارگیری) + (زمان کنونی) است.

به منظور برآورد کردن بهره برداری از دستگاه های بارگیری و باسکول، دو آمار تجمعی را نگهداری می کنیم:

- B_L = مجموع مدت اشتغال هر دو دستگاه بارگیری از زمان صفر تا زمان t
- B_S = مجموع مدت اشتغال باسکول از زمان صفر تا زمان t

چون از زمان صفر تا زمان ۲۰ هر دو دستگاه بارگیری مشغول است، در زمان $t=20$، $B_L = 40$ است. اما از زمان ۲۰ تا ۲۴ تنها یک دستگاه بارگیری مشغول است. بنابراین در فاصله زمانی [۲۴ ، ۲۰]، B_L تنها به مقدار ۴ دقیقه افزایش می یابد. همچنین از زمان ۲۵ تا ۳۶ هر دو دستگاه بارگیری بیکار است ($L(25) = 0$)، و به این ترتیب B_L تغییر نمی کند. برای شبیه سازی نسبتاً کوتاه در جدول ۳-۶، معیارهای بهره برداری به شرح زیر براورد می شود:

این براوردها را نمی توان به عنوان براوردهای دقیق بهره برداری در "حالت پایا" از دستگاه های بارگیری و باسکول در بلند مدت محسوب داشت. به منظور کاستن از تأثیر شرایط مفروض در زمان صفر (حضور ۵ کامیون از ۶ کامیون در محل بارگیری) و بدست آوردن براوردهای دقیق، یک شبیه سازی بسیار طولانی مورد نیاز است. از سوی دیگر، اگر شبیه ساز به رفتار گذرای سیستم در مدتی کوتاه (مثل یک یا دو ساعت)، و به ازای شرایط مفروض اولیه علاقمند باشد، می تواند نتایج موجود در جدول ۳-۶ را معرف (یا نمونه ای از) این رفتار گذرا بداند. با انجام شبیه سازی های بیشتر، می توان نمونه های دیگری بدست آورد. به طوری که هر شبیه سازی دارای همان شرایط اولیه باشد ولی از رشتهی متفاوتی از ارقام تصادفی برای تولید مدتهای مربوط به فعالیت ها استفاده کند.

جدول ۳-۶ یعنی جدول شبیه سازی مربوط به عملیات کامیون ها را می شد با صریحاً مدلسازی نکردن کامیون ها به عنوان نهاد، قدری ساده تر کرد. به عبارت دیگر، پیشامدها ممکن بود به صورت (EL , t)، و ... نوشته شوند، و متغیرهای حالت را می شد فقط برای نگهداشتن حساب تعداد کامیون ها در هر بخش از سیستم به کار برد نه برای تعیین اینکه کدام کامیون ها رد این امور درگیرند. با اینگونه معرفی، همان آمار مربوط به بهره برداری را می توان گردآوری کرد. از سوی دیگر، اگر میانگین مدت پاسخ "سیستم" یا نسبت کامیونهایی را که بیش از ۳۰ دقیقه در "سیستم" می مانند براورد می کردیم، به طوری که "سیستم" صف بارگیری و دستگاه های بارگیری و صف توزین و باسکول را در برگیرد، نهادهای کامیون (DT_i) به اضافهی یک ویژگی، یعنی زمان ورود به صف بارگیری مورد نیاز خواهد بود. با ترک باسکول از سوی یک کامیون، مدت پاسخ این کامیون را ممکن بود به صورت زمان کنونی شبیه سازی (t) منهای ویژگی زمان ورود محاسبه کرد. این مدت پاسخ جدید برای تازه کردن آمار تجمعی مورد استفاده قرار می گیرد: S = مجموع مدت پاسخ تمام کامیون هایی که سراسر "سیستم" را پیموده اند و F = تعداد مدت های پاسخ کامیون ها که بیش از ۳۰ دقیقه است. مثال اخیر، مجدداً این نکته به نمایش می گذارد که میزان پیچیدگی مدل تا حدی به معیارهای عملکردی که براورد می شود، بستگی دارد.

زبانهای برنامه نویسی برای شبیه سازی

زبانهای کامپیوتری شبیه سازی، به طور قابل توجهی ایجاد و اجرای شبیه سازی سیستم های پیچیده واقعی را تسهیل می کند. به طور کلی، هر زبان نسبت به اوضاع واقعی یک جهت گیری یا نگرش کلی دارد. زبانهای تشریح شده در این فصل را می توان به زبانهایی رده بندی کرد که یا رهیافت زمانبندی پیشامدها یا رهیافت پردازش-تقابل را در مدلسازی در پیش می گیرند. رهیافت زمانبندی پیشامدها ایجاب می کند که تحلیلگر توجه خود را به پیشامدها و چگونگی تأثیر آنها بر حالت سیستم معطوف نماید. رهعیافت پردازش-تقابل به تحلیلگر اجازه می دهد تا توجه خود را به یک نهاد منفرد (مانند یک مشتری) و توالی پیشامدها و فعالیتهایی که او با گذر کردن از سیستم آنها را تجربه می کند، معطوف دارد. به هنگام استفاده از یک زبان همه منظوره از قبیل FORTRAN، ALGOL، BASIC یا پاسکال شبیه ساز به احتمال فراوان رهیافت زمانبندی پیشامدها را برمی گزینند.

زبانهایی ازقبیل GASP استفاده از رهیافت زمانبندی پیشامدها را تسهیل می کند. اما GPSS امکان استفاده از رهیافت پردازش-تقابل را به نوعی فراهم می آورد. برخی از زبانهای جدیدتر مانند SIMSCRIPT و SLAM به شبیه ساز اجازه می دهند یکی از دو طریق فوق یا ترکیبی از هردو را بر حسب متناسب بودن با مسأله مورد نظر به کار برد.

FORTRAN یک زبان برنامه نویسی علمی است و مشخصاً برای استفاده در شبیه سازی طراحی نشده است. به هنگام استفاده از FORTRAN، تحلیلگر احتمالا گرایش زمانبندی پیشامد ها را بر می گزینند. در سوی دیگر این طیف، GPSS قرار دارد که یک زبان خاص وبسیار منسجم شبیه سازی و زبانی نهاد گراست، که این مورد خاصی از نگرش پردازشگرا در حالت کلی آن است. GPSS برای شبیه سازی کردن نسبتاً ساده سیستمهای صف، از قبیل سیستمهای صف کارگاهی طراحی شد. FORTRAN و GPSS در سطح وسیعی برای طراحی مدل شبیه سازی گسسته پیشامد مورد استفاده قرار می گیرد.

SIMSCRIPT و SLAM زبانهای سطح بالای برنامه نویسی شبیه سازی است با ساختاری که به منظور تسهیل مدلسازی طراحی شده است. SIMSCRIPTII.5 و SLAM هر دو، انتخاب یکی از دو گرایش را در اختیار تحلیلگر قرار می دهد، یا توسط آنها می توان مدلی با استفاده توام از دو گرایش ساخت. برخلاف FORTRAN این دو زبان امکاناتی مانند مدیریت فهرست پیشامدهای آتی و دیگر مجموعه ها، مولدهای تعبیه شده مقادیر تصادفی و برنامه های تعبیه شده گردآوری آمار را فراهم می آورد. بر خلاف GPSS، محاسبات پیچیده توسط هر یک از این دو زبان به سادگی انجام می گیرد. SLAM و نسخه ای بسط داده شده از SIMSCRIPT (به نام C-SIMSCRIPT) توانایی انجام شبیه سازیهای پیوسته را نیز فراهم می آورد، یعنی مدلسازی از سیستم هایی که متغیرهای وضعیت با تغییرات پیوسته دارند. SLAM مبتنی بر FORTRAN است و به عنوان زیر مجموعه ای از خود، زبانی GASP گونه دارد. GASP که در بخش های بعدی شرح داده می شود، مجموعه ای از زیر برنامه های FORTRAN برای تسهیل شبیه سازی های دارای نگرش پیشامدگراست که به زبان FORTRAN نوشته می شود. جزء مربوط به نگرش پردازشگرا در SALM

در بخش بعدی توضیح داده می شود. از سوی دیگر SIMSCRIPT به عنوان زیر مجموعهی خود, یک زبان کامل برنامه نویسی علمی و قابل مقایسه با FORTRAN، PL/1، ALGOL است.

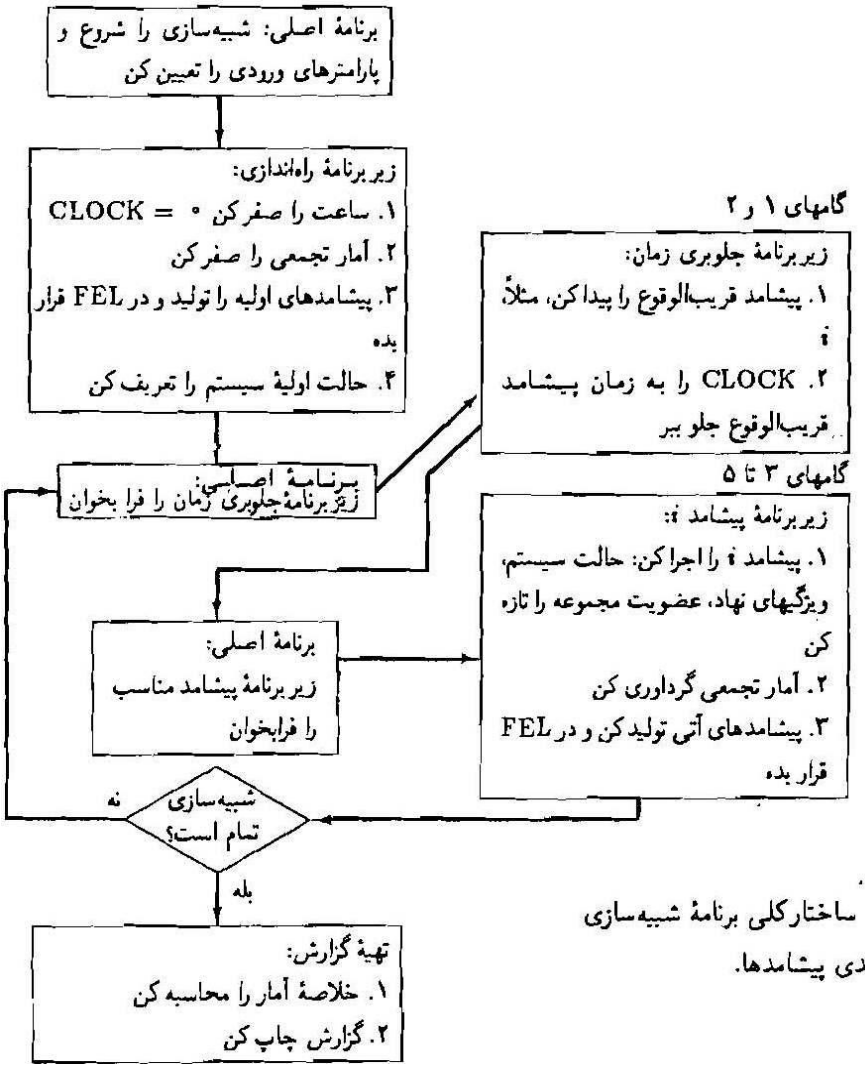
FORTRAN یک زبان برنامه نوسی است که در سطح وسیعی شناخته شده و در دسترس است، اما هیچ تسهیلاتی که مستقیماً هدف کمک رسانی به شبیه ساز داشته باشد را ارائه نمی دهد. شبیه ساز ناگریز است الگوریتم زمانبندی پیشامد ها و جلو بری زمان، توانایی گردآوری آمار، تولید نمونه ها از توزیعهای مشخص احتمال و مولد گزارش را، خود برنامه نویسی کند.

استفاده از زبان FORTRAN برای مدلهای بزرگ ممکن است کاملاً دشوار باشد: به علاوه، ممکن است این امر به مدلهایی بیانجامد که غلطگیری آن مشکل و اجرای آن کند باشد مگر اینکه برنامه نویس نحوه برخوردی به دقت سازمان یافته در پیش گیرد و از روشهای کارای پردازش فهرستها شناخت داشته باشد.
برای مدلهای کوچک، می توان شبیه سازی با زبان FORTRAN (یا هر زبان همه منظوره دیگر) را به عنوان ابزاری فراگیری مفاهیم الگوریتم زمانبندی پیشامدها و جلوبری زمان به کار گرفت.
زبانهای ویژه شبیه سازی عموماً ریزه کاریهای زمانبندی پیشامدها را از نظر پنهان می دارد. هر مدل شبیه سازی گسسته پیشامد نوشته شده به زبان FORTRAN در بر دارنده اجزا مورد بحث در بخش ۳-۱ یعنی حالت سیستم، نهادها و ویژگیها، مجموعهها، پیشامدها، فعالیتها و تاخیرها، به اضافه اجزا مندرج در فهرست زیر است. به منظور تسهیل ایجاد مدل FORTRAN و غلطگیری از آن، بهترین راه این است که مدل با به کار گیری زیر برنامه ها، به گونه ای نا متمرکز سازماندهی شود. اجزا زیر در تقریباً تمام مدلهای نوشته به زبان FORTRAN مشترک است:

- CLOCK. متغیری که زمان شبیه سازی شده را تعریف می کند.
- زیر برنامه راه اندازی. برنامه ای که برای تعریف حالت سیستم در زمان صفر به کار می رود.
- زیر برنامه جلوبری زمان. برنامه ای که برای یافتن پیشامد بعدی (به نام پیشامدقریب الوقوع و معروف به (IMEVT) فهرست پیشامدهای آتی (FEL) را جستجو می کند و ساعت را به زمان وقوع پیشامد قریب الوقوع جلو می برد.
- زیر برنامه زمانبندی. برنامه ای که پیشامدهای آتی تولید شده را در FEL قرار می دهد .
- زیر برنامه پیشامدها. زیر برنامه های مربوط به هر نوع پیشامد برای تازه کردن حالت سیستم (و آمار تجمعی) هرگاه آن پیشامد رخ دهد.
- مولد های مقادیر تصادفی. برنامه هایی برای تولید نمونه از توزیعهای احتمال مورد نظر .
- برنامه اصلی. کنترل کلی بر الگوریتم زمانبندی پیشامدها را فراهم می آورد.

- تهیه گزارش. برنامه ای که از آمار تجمعی، آمار خلاصه را محاسبه و در پایان شبیه سازی گزارشی چاپ می کند.

ساختار کلی برنامه شبیه سازی FORTRAN در شکل ۳-۸ نشان داده شده است. این نمودار جریان بسط یافته الگوریتم زمانبندی پیشامدها و جلوبری زمان است که به اختصار در شکل ۳-۲ ارائه شد . شبیه سازی با تنظیم ساعت (CLOCK) شبیه



شکل ۳-۸ ساختار کلی برنامهٔ شبیه‌سازی با دید زمانبندی پیشامدها.

سازی روی صفر، قرار دادن صفر در محلهای گردآوری اطلاعات تجمعی، تولید پیشامد ها ی اولیه (همواره دست کم، یک پیشامد از این گونه وجود دارد) و قرار دادن آنها در FEL، وتعریف حالت سیستم در لحظه صفر شروع می شود. پس از این، برنامه شبیه سازی آن قدر بین زیر برنامه جلوبری زمان و زیر برنامه های مربوط به پیشامد ها می گردد تا شبیه سازی به پایان برسد. زیر برنامه جلوبری زمان به منظور یافتن پیشامد قریب الوقوع که مثلاً پیشامدی از نوع I است، به جستجوی FEL می پردازد. سپس CLOCK شبیه سازی به زمان رویداد پیشامد قریب الوقوع جلو برده می شود. (به یاد دارید که حالت سیستم و ویژگیهای نهاد از لحاظ مقدار در دوره زمانی بین رویداد دو پیشامد متوالی دستخوش تغییر نمی شوند. در واقع، این تعریف شبیه سازی گسسته پیشامد است: حالت سیستم تنها هرگاه پیشامدی روی دهد تغییر می کند.) پس از این، زیر برنامه پیشامد I فرا خوانده می شود تا پیشامد قریب الوقوع را اجرا، آمار تجمعی را تازه و پیشامد های آتی را تولید کند (تا در FEL قرار داده شوند). اجرای پیشامد قریب الوقوع به این معناست که حالت سیستم،

ویژگیهای نهاد و اعضای مجموعه ها به منظور انعکاس این واقعیت که پیشامد I رخ داده است تغییر می پذیرد.

توجه داشته باشید که تمام فعل و انفعالات موجود در زیر برنامه هر پیشامد در یک لحظه زمان شبیه سازی شده، روی می دهد. مقدار متغیر CLOCK در زیر برنامه مربوط به پیشامدها تغییر نمی کند. اگر شبیه سازی پایان نیافته باشد مجدداً کنترل به زیر برنامه جلوبری زمان، سپس به زیر برنامه پیشامد مورد نظر، و... سپرده می شود. هرگاه شبیه سازی پایان یابد کنترل به زیر برنامه تهیه گزارش سپرده می شود، تا بر اساس آمار تجمعی گردآوری شده خلاصه آمار مورد نظر را محاسبه کند و گزارشی نیز چاپ کند.
کارایی هر مدل شبیه سازی بر حسب مدت اجرای کامپیوتری تا اندازه زیادی به وسیله تکنیکهای مورد استفاده در اداره FEL و سایر مجموعه ها تعیین می شود. چنانکه قبلاً شرح دادیم، بیرون آوردن پیشامد قریب الوقوع و افزودن پیشامد جدید دو عمل اصلی است که روی FEL انجام می شود.

شبیه سازی با GASP

GASPمجموعه ای است از زیر برنامه های FORTRAN که به منظور تسهیل شبیه سازی مبتنی بر زمانبندی پیشامد ها به زبان FORTRAN طراحی شده است . این مجموعه شامل بیش از ۳۰ زیر برنامه و تابع است که امکانات مورد نیاز فراوانی از جمله یک برنامه جلوبری زمان (به نام GASP) برنامه های اداره فهرست پیشامد های آتی، برنامه های افزودن و کاستن نهاد ها از مجموعه ها برنامه های گردآوری آمار، برنامه های تولید مقدار تصادفی و یک برنامه استاندارد تهیه گزارش را فراهم می آورد. برنامه نویس باید یک برنامه اصلی، یک برنامه راه اندازی مدل، برنامه های مربوط به پیشامد ها و در صورت تمایل، یک برنامه تهیه گزارش، به اضافه یک زیر برنامه به نام EVNTS را خود فراهم کند. برنامه اصلی باید دارای جمله CALL GASP باشد تا شبیه سازی را شروع کند. زیر برنامه GASP پیشامد قریب الوقوع را تعیین و از طریق مقدار شاخص خود زیر برنامه ای نوشته کاربر، یعنی EVNTS، را فرا می خواند. این شاخص (یعنی NEXT) تعیین می کند پیشامد قریب الوقوع کدام است تا به وسیله EVNTS فرا خوانده شود.

شبیه سازی با SIMSCRIPT

SIMSCRIPT یک زبان سطح بالای برنامه نویسی با امکاناتی است که مشخصاً برای ایجاد مدل شبیه سازی گسسته پیشامد طراحی شده است. به عنوان یک زبان شبیه سازی، نقطه مورد نظر زمانبندی پیشامد ها و پردازش - تقابل را مجاز می دارد. به عنوان یک زبان علمی، دست کم به توانمندی ALGOL ;PL / 1; FORTRAN یا پاسکال است. بسیاری از کارهای برنامه نویسی به زبان SIMSCRIPT با کارایی بیشتری نسبت به FORTRAN انجام پذیر است. هر برنامه SIMSCRIPT به جملات شبیه زبان انگلیسی و با شیوه قالب بندی آزاد قابل نوشتن است، یک چنین برنامه تقریباً خود مستند ساخته است و به راحتی برای افرادی غیر از برنامه نویس قابل تشریح است. بر خلاف، FORTRAN، SIMSCRIPT امکان نگهداری خودکار فهرست پیشامد های آتی و الگوریتم جلو بری زمان و زمانبندی پیشامد ها، نگهداری خودکار مجموعه ها، شامل عملیات افزودن و حذف نهاد ها از مجموعه ها، گردآوری خودکار آمارهای مورد نیاز و مولد های متعدد مقادیر تصادفی انواع وسیعی از توزیعهای احتمال را فراهم می کند.

هر برنامه SIMSCRIPT از یک دیباچه، یک برنامه اصلی، برنامه های پیشامدها، و زیر برنامه های متعارف تشکیل می شود. همانطور که اشاره شد، برنامه جلو بری زمان، برنامه های تولید مقادیر تصادفی و برنامه های گردآوری آمار، به طور خودکار موجود است. دیباچه شرحی ایستا از سیستم را از طریق تعریف تمام نهادها، ویژگی های آنها و مجموعه هایی که نهادها احتمالاً به آنها متعلق است ارائه می دهد. دیباچه همچنین متغیر های سراسری را تعریف آماری را که در مورد برخی متغیرها لازم است گردآوری شود، مشخص می کند، مجموعه وسیعی از متغیر ها به طور خودکار نگهداری می شود.

برنامه اصلی، مقدار پارامتر های ورودی را می خواند (یا مشخص می کند)، حالت سیستم در شروع کار را مشخص می کند، و اولین پیشامدها را تولید می کند. برنامه های پیشامد ها با زیر برنامه جلوبری زمان به طور خودکار فراخوانده می شود که این زیر برنامه به نوبه خود توسط جمله "START SIMULATION" در برنامه اصلی فعال می شود. زیر برنامه های متعارف را می توان از هر زیر برنامه پیشامد یا از برنامه اصلی فراخواند.

شبیه سازی با GPSS

GPSS یک زبان شدیداً ساختاربندی شده و ویژهی شبیه سازی است که رهیافت پردازش-تقابل را به کار می برد، نسبت به شبیه سازی مسائل صف گرایش دارد و دیاگرام بلوکی برای شرح سیستم فراهم می کند.

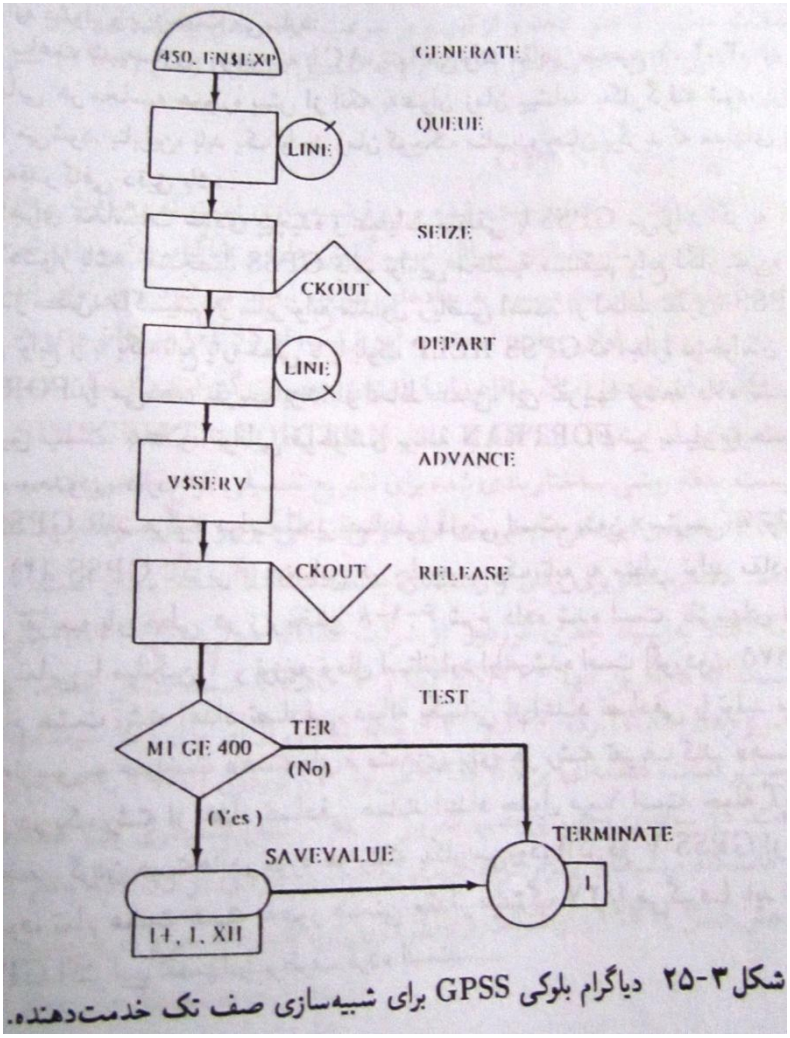
نهادهایی موقت موسوم به نهادهای گذرنده ایجاد می شود و می توان آنها را چنین تصویر کرد که در دیاگرام بلوکی جاری است. به این ترتیب، از GPSS می توان برای هر وضعیتی استفاده کرد که در آن نهادها (مثلاً متقاضیان) را بتوان به صورت گذر کننده از میان سیستم مجسم کرد (مثلاً شبکه ای از صف ها). GPSS به مانند FORTRAN یا SIMSCRIPT یک زبان دستوری نیست. بلکه روشی ساختاربندی شده برای تشریح تفصیلی انواع خاصی از سیستم هاست. پردازشگر GPSS سپس این شرح (یعنی دیاگرام بلوکی) را می گیرد و به طور خودکار یک شبیه سازی انجام می دهد. مکانیزم جلوبری زمان و زمانبندی پیشامدها کاملاً از دید پنهان است.

دو مفهوم اصلی در GPSS، عبارت از نهادهای گذرنده و بلوکهاست. هر بلوک میتواند به صورت نمادی تصویری یا جمله ای به زبان GPSS معرفی شود. در GPSS بیش از ۴۰ بلوک استاندارد وجود دارد. هر بلوک معرف فعالیت یا پیشامد مشخصی است که در هر سیستم عادی ممکن است، رخ دهد. این بلو ک ها در یک دیاگرام بلوکی که معرف پردازش یک "متقاضی" است، مرتب می شود. شکل ۳-۲۵ توصیف GPSS یک پردازش "متقاضی" را برای صف تک خدمتدهنده چنین پردازشی نشان داده شده است. نهادهای گذرنده که معرف نهادهای پویا و فعال است را می توان چنین تصویر کرد که در سراسر دیاگرام بلوکی جاری است. هر مسیری را که یک نهاد گذرنده بتواند در سیستم در پیش گیرد باید در دیاگرام بلوکی، که می تواند شاخه هایی داشته باشد نشان داده شود. منابع محدود هر سیستم با نهادها، امکانات و انبارهای از قبل تعریف شده-ی GPSS معرفی می شود. هر مورد از امکانات، اساساً یک خدمتدهنده است. هر انبار متشکل از گروهی از خدمتدهنده های موازی است. آمار به طور خودکار در مورد ضریب بهره برداری از امکانات و انبارها گردآوری می شود. به علاوه، نهادهای صفی و جدولی به منظور گردآوری آمار در مورد صفهای انتظار یا مدتهای انتقال دسترس پذیر است.

شبیه سازی با SLAM

SLAM یک زبان سطح بالا و مبتنی بر زبان FORTRAN در شبیه سازی است که گرایش زمانبندی پیشامدها یا پردازش-تقابل، یا ترکیبی از هر دو را میسر می شازد. بخش زمانبندی پیشامدها در SLAM کاملاً شبیه GASP است. بخش پردازش-تقابل SLAM از بسیاری جنبه ها شبیه GPSS است.

به منظور استفاده از رهیافت پردازش-تقابل با SLAM، شبیه ساز باید شبکه ای متشکل از گره ها و شاخه ها ایجاد کند که به صورت تصویری پردازش ها را در سیستم معرفی می کند. عناصر جاری در سیستم نهاد نامیده می شود.



شکل ۳-۲۵ دیاگرام بلوکی GPSS برای شبیه‌سازی صف تک خدمت‌دهنده.

هر مدل شبکه ای کامل SLAM از یک سیستم معرف تمام مسیرهایی است که هر نهاد به هنگام پیمودن سیستم ممکن است در پیش گیرد. به منظور اجرای مدل شبیه ساز مدل شبکه ای را مستقیماً به جملات کامپیوتری ترجمه می کند که به پردازنده SLAM وارد می شود.

SLAM به طور خودکار الگوریتم زمانبندی پیشامدها و جلوبری زمان، عملیات مجموعه ها از قبیل افزودن یا کاستن نهادها و ... را اجرا می کند. هر شبکه SLAM از شاخهها و گرهها تشکیل میشود. هر شاخه معرف گذر زمان است. یعنی یک فعالیت را نشان می دهد. به علاوه، هر شاخه ممکن است معرف تعداد محدودی خدمت دهنده باشد. هر شاخه به صورت یک جمله ACTIVITY رمزگذاری می شود. از گرهها برای معرفی پیشامد ورود، تأخیرها یا انتظارهای مشروط، پیشامد ترک یا سایر اعمال متداول سیستم استفاده می شود.

در این فصل مقدمه ای بر زبان های شبیه سازی ارائه شده است و در جدول ۳-۸ مقایسه ای بر این زبان ها آورده شده است.

جدول ۳-۸ مقایسهٔ زبانها برای شبیه‌سازی گسسته پیشامد.

ضابطه	زبان				
	FORTRAN	GASP	SIMSCRIPT II.5	GPSS V	SLAM
سهولت فراگیری	خوب	خوب	خوب	عالی	عالی
سهولت درک مسأله	ضعیف	متوسط	خوب	عالی(الف)	عالی(الف)
سیستم مورد گرایش رهیافت مدلسازی	هیچ(ب)	همه	همه	صف	همه
زمانبندی پیشامدها	نه(ب)	بله	بله	نه	بله
پردازش-تقابل	نه(ب)	نه	بله	بله	بله
پیوسته	نه(ب)	بله	بله	نه	بله
امکانات					
نمونه‌گیری تصادفی درونی	نه(ج)	بله	بله	نه(د)	بله
توان گردآوری آمار	ضعیف	عالی	عالی	خوب(د)	عالی
توان فهرست‌پردازی	ضعیف	خوب	عالی	متوسط	خوب
سهولت دریافت گزارش استاندارد	ضعیف	عالی	متوسط	عالی	عالی
سهولت طراحی گزارش ویژه	متوسط	خوب	عالی	ضعیف(د)	خوب
غلط‌یابی	متوسط	خوب	عالی	متوسط(د)	خوب
مدت اجرا روی کامپیوتر	عالی(ه)	خوب	خوب	ضعیف(د)	خوب
مستندسازی از لحاظ فراگیری زبان و از دید مراجعه	بسیار خوب	بسیار خوب	متوسط	بسیارخوب	بسیار خوب
خود مستند ساخته بودن رمز	ضعیف	خوب	خوب	عالی	خوب
هزینه	پایین(و)	پایین(و)	بالا	پایین (بالا GPSS/H)	متوسط

تمرین فصل اول:

۱. در مورد سیستم های زیر، چند نهاد، خصیصه، فعالیت، پیشامد، و متغیر حالت را نام ببرید:

أ. تعمیرگاه وسایل برقی خانگی

ب. کافه تریا

ج. فروشگاه مواد غذایی

د. لباسشویی عمومی

ه. غذا خوری سرپایی

و. اتاق اورژانس بیمارستان

ز. شرکت تاکسیرانی با ۱۰ دستگاه خودرو

ح. خط مونتاژ خودرو

سیستم	نهاد	خصیصه	فعالیت	پیشامد	متغیرهای حالت
تعمیرگاه وسایل برقی خانگی	مشتری (وسایل خانگی خراب)	نوع وسیله- نوع خرابی- (رفع عیب- تعویض قطعات- تأمین قطعات یدکی)	عیب یابی و تعمیر	ورود کالا برای تعمیر- خروج کالای تعمیر شده	تعداد وسایل معیوب در صف- تعداد وسایل تعمیر نشده- مدت زمان میان عیب یابی- مدت زمان میان تعمیر- تعداد تعمیرکاران موجود
کافه تریا	مشتری ها	میزان غذای درخواستی – نوع غذای درخواستی	انتخاب غذا – پرداخت برای غذا	رسیدن به صف تحویل – خروج از صف تحویل	تعداد مشتری ها در صف – تعداد خدمت دهنده ها

فروشگاه مواد غذایی	فروشنده ها	تعداد سفارش در هر لیست	مراجعه برای تسویه حساب	رسیدن به صندوق – خروج از صف صندوق	تعداد مشتری ها در صف – تعداد صندوق ها
لباسشویی عمومی	ماشین لباسشویی	نرخ خرابی دستگاه	تعمیر یک ماشین	رخ دادن خرابی دستگاه – اتمام کار دستگاه	تعداد ماشین های در حال کار – تعداد دستگاه های در حال تعمیر – تعداد دستگاه های منتظر تعمیر
غذا خوری سرپایی	مشتری ها	میزان سفارش	ارائه سفارش – پرداخت برای سفارش	رسیدن به صندوق – تکمیل خرید	تعداد مشتری های منتظر – تعداد سرویس دهنده ها
اتاق اورژانس بیمارستان	اتاق- بیمار- پرستار- پزشک	نوع بیماری- وخامت بیماری- ظرفیت بیمار- انواع آسیب دیدگی های قابل پذیرش	درمان	پذیرش- بستری- مداوا- ترخیص	تعداد بیماران مراجعه کننده- تعداد بیماران در انتظار بستری- مدت زمان هر مداوا- تعداد پرسنل متخصص- تعداد تخت و ...
شرکت تاکسیرانی	خودرو- راننده- مسافر	نوع خودرو- تعداد خودرو-تعداد راننده- تعداد سرویس قابل انجام توسط هر خودرو- حمل ونقل مسافر- مسافت- تعداد- وجود با یا عدم وجود بار	حمل و نقل مسافران از مبدأ به مقصد	ورود تقاضا- انجام تقاضا	تعداد خودرو در حال کار- تعداد خودرو غیر قابل استفاده (به هر دلیلی)- تعداد تقاضا- مدت زمان میان هر تقاضا- مدت زمان میان خرابی هر اتومبیل- مدت زمان رفت و برگشت هر خودرو
خط مونتاژ خودرو	کارگر- ماشین آلات- محصول	تخصص- تعداد تولید- مدت زمان میان هر تعمیر- عمر مفید- ارزش محصول	ورود و خروج کارگران– مونتاژ محصول	ورود و خروج کارگر- انواع مرخصی- ورود تقاضا- ورود مواد اولیه- خروج محصول مونتاژ شده	مدت زمان مونتاژ محصول – مدت زمان کارکرد ماشین آلات در روز – مدت زمان میان هر خرابی دستگاه ها – تعداد دستگاه های مونتاژ شده در واحد زمان-

۳. شبیه سازی ترافیک در تقاطعی مهم قرار است با این هدف انجام گیرد که جریان فعلی ترافیک را اصلاح کند. گامهای ۱ و ۲ فرایند شبیه سازی شکل ۱-۴ را سه بار تکرار کنید، به طوری که هر بار تکرار از بار قبلی پیچیده تر باشد.

مرحله	فرمول بندی مسأله	طرح پیشنهادی
۱	خودرو هایی که به چهارراه وارد می	نحوه زمانبندی چراغ باید تعیین گردد! معیارهای بررسی بازده :مدت زمان

	شوند توسط چراغ راهنمایی کنترل می گردند. مسیرهای ممکن : مستقیم، چرخش به چپ و راست می باشد .درو زدن نیز ممنوع در نظر گرفته شده است	تاخیر برای خودرو منابع مورد نیاز برای شبیه سازی: حداقل دو نفر نیروی انسانی برای ثبت نتایج و رخدادها به مدت ۵ روز – یک نفر برای آنالیز و بررسی اطلاعات گرد آوری شده - ۱نفر برای تولید مدل ۳ روز - ۱ نفر برای اجرای مدل ها - یک نفر برای ۳ روز کار برای پیاده سازی مدل
۲	کلیه موارد فوق + گردش براست بعد از توقف کامل در چهارراه مجاز می باشد .عابرین پیاده نیز از چراغ تبعیت می نمایند .حریم چهارراه رعایت شده است.	نحوه زمانبندی چراغ باید تعیین گردد! معیارهای بررسی بازده :مدت زمان تاخیر برای خودروها منابع مورد نیاز برای شبیه سازی: حداقل دو نفر نیروی انسانی برای ثبت نتایج و رخدادها به مدت ۸ روز – یک نفر برای آنالیز و بررسی اطلاعات گرد آوری شده به مدت ۵ روز- ۱ نفر برای تولید مدل ۴ روز - ۱نفر برای اجرای مدل ها به مدت ۴ روز -یک نفر برای ۳ روز کار برای پیاده سازی مدل
۳	کلیه موارد فوق + خودرو سنگین به چهاراه وارد می شود -یک اتومبیل خراب شده و یک لاین را مسدود می نماید - بر اثر وقوع تصادف برای یک بازه نامشخص ترافیک تولید می گردد.	نحوه زمانبندی چراغ باید تعیین گردد !آیا لازم است تا مسیر خیابان عریض گردد- معیارهای بررسی بازده :مدت زمان تاخیر برای کل وسایط نقلیه منابع مورد نیاز برای شبیه سازی: حداقل دو نفر نیروی انسانی برای ثبت نتایج و رخدادها به مدت ۱۰ روز – یک نفر برای آنالیز و بررسی اطلاعات گرد آوری شده به مدت ۵ روز- ۱ نفر برای تولید مدل ۵ روز- ۱ نفر برای اجرای مدل ها به مدت ۵ روز -یک نفر برای ۳ روز کار برای پیاده سازی مدل

۴. از کدام راه ها و با چه گام هایی می توان برای اجرای فرایند شبیه سازی شکل ۱-۴ از کامپیوتر شخصی استفاده کرد.

گام ۴- جمع آوری داده ها: ذخیره سازی اطلاعات در فایل باعث بالا رفتن سرعت دسترسی و حجم ذخیره سازی و کاهش هزینه می شود.

گام ۵- ترجمه مدل: تعداد زیادی زبان برنامه نویسی برای این کار موجود است.

گام ۷- تأیید مدل: تقریباً یک گام آماری است. برای این مرحله بسته های آماری مختلفی موجود است.

گام ۸- طراحی آزمایشی: مثل گام ۷.

گام ۹- اجراهای تولیدی: مثل گام ۵.

گام ۱۱- مستند سازی و تهیه گزارش: نرم افزارهای زیادی برای این کار وجود دارد.

۶. قرار است پخت اسپاگتی برای شام شبیه سازی و تعیین شود که برای حاضر بودن شام رأس ساعت ۷ شب بر روی میز، چه موقع باید کار را شروع کرد. به منظور اجرای شبیه سازی به نحوی که مدل همه مراحل تهیه غذا را در بر داشته باشد، آنچه را فکر می کنید در قسمت گردآوری داده های فرایند شبیه سازی شکل ۱-۴ مورد نیاز است به بهترین وجه ممکن پیگیری کنید.

داده های لازم:

تعداد مهمان ها
زمان لازم برای جوشیدن آب
زمان لازم برای پختن غذا
زمان لازم برای آماده سازی میز و ظروف
...

رخدادها:
شروع پخت
اتمام پخت
رسیدن مهمانها
شروع غذا خوردن

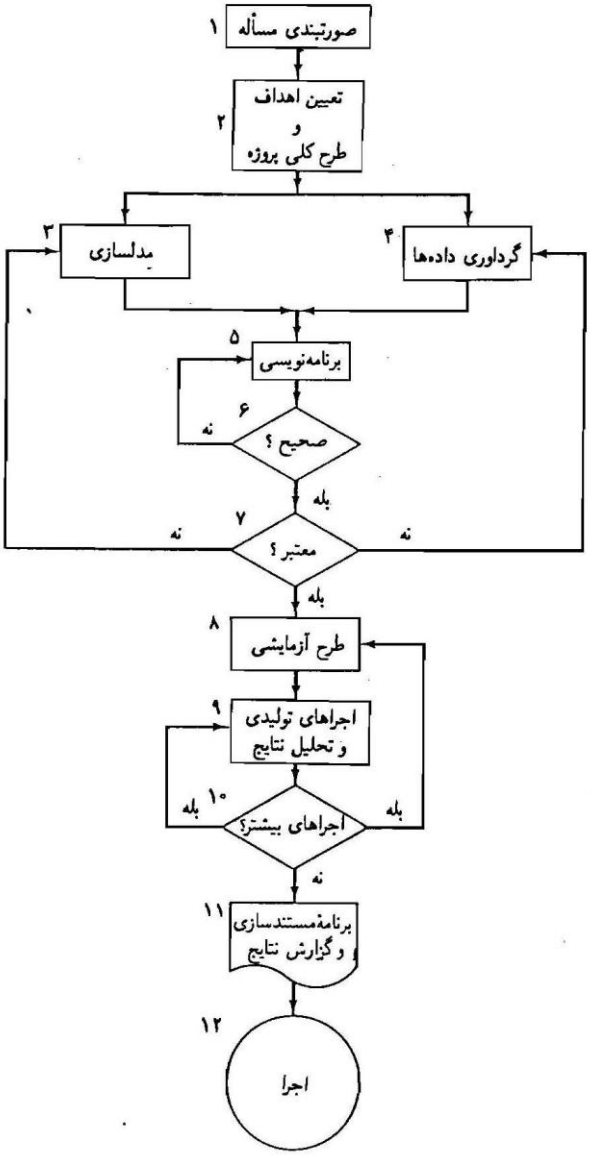
فعالیتها:
به جوش آوردن آب
طبخ غذا
خدمتدهی به مشتریان

متغیرهای حالت:
تعداد مهمان ها
حالت آب
حالت سس
حالت غذا

۷. پیشامدها و فعالیت های مربوط به عملیات دفترچه خساب جاری شما کدامند؟

رخداد ها:
واریز
برداشت

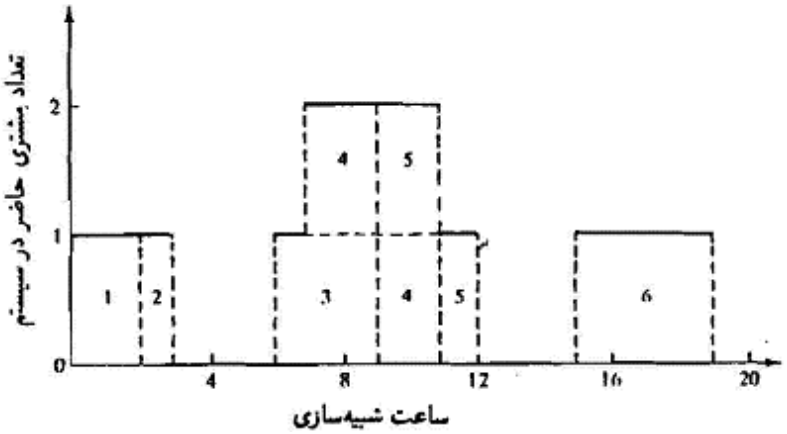
فعالیتها:
نوشتن چک
نقد کردن چک
واریز به حساب
تأیید موجودی



شکل ۱-۴ گامهای اساسی در بررسی مبتنی بر شبیه‌سازی.

تمرین های فصل دوم:

۴. در مثال ۲-۱ میانگین وزندار زمانی تعداد مشتری در سیستم و میانگین وزندار زمانی تعداد مشتری در صف انتظار را تعیین کنید. (راهنمایی: شکل ۲-۶ را بکار ببرید.)



شکل ۲-۶ تعداد مشتری حاضر در سیستم

$\hat{L} = \sum_{i=0}^{\infty} iT_i/T$

که $\hat{L}$ = میانگین وزندار زمانی تعداد مشتری در سیستم

T_i= کل زمان در بازه [0,T] که در آن دقیقاً i مشتری در سیستم بوده

$\hat{L} = \sum_{i=0}^{4} iT_i/$

$\hat{L}_Q = \sum_{i=0}^{\infty} iT_i$

که $\hat{L}_Q$= میانگین وزندار زمانی تعداد مشتری در صف انتظار

T_i^Q =کل زمان در بازه [0,T] که در آن دقیقاً i مشتری در صف انتظار بوده

$\hat{L} = \frac{[0(50) +}{}$

۶. مجدداً در مورد مثال ۲-۲، فرض کنید زانوی هابیل مجروح است و به سرعت قبل نمی تواند حرکت کند. در نتیجه، دو تغییر رخ می دهد. توزیع خدمتدهی هابیل تغییر می کند و اگر هر دو آورنده غذا بیکار باشند، خباز در گرفتن مشتری حق تقدم دارد. توزیع جدید خدمتدهی هابیل به شرح زیر است:

مدت خدمتدهی (دقیقه)	۳	۴	۵	۶
احتمال	۰/۳۰	۰/۳۰	۰/۲۵	۰/۱۵

أ. شبیه سازی و تحلیل پس از آن را برای ۳۰ خدمتدهی کامل شده انجام دهید. تأثیر آسیب دیدگی هابیل و قاعده جدید چیست؟

مشتری	ارقام تصادفی ورود	زمان بین دو ورود	زمان ورود بر حسب ساعت	عدد تصادفی خدمتدهی	هابیل			خباز			مدت انتظار در صف
					زمان شروع خدمتدهی	مدت خدمتدهی	زمان پایان خدمتدهی	زمان شروع خدمتدهی	مدت خدمتدهی	زمان پایان خدمتدهی	
1	-	-	0	95				0	6	6	0
2	26	2	2	25	2	3	5				0
3	98	4	6	51				6	4	10	0
4	90	4	10	92				10	6	16	0
5	26	2	12	89	12	6	18				0
6	42	2	14	38				16	4	20	2
7	74	3	17	13	18	3	21				1
8	80	3	20	61				20	5	25	0
.											
.											
.											
25	16	1	55	87	6	63					2

نتایج بدست آمده از شبیه سازی:

- سرویس دهی توسط هابیل: ۱۲ ماشین
- میانگین زمان انتظار در صف: ۱/۵ دقیقه

ب. تأثیر افزودن کارمند تازهای که به سرعت خباز کار می کند، چیست؟ کارمند جدید تمام کار زیاد آمده بعد از خباز و هابیل را انجام میدهد.

مدت خدمتدهی	احتمال	احتمال تجمعی	
۳	۰/۳۰	۰/۳۰	۰۱-۳۰
۴	۰/۳۰	۰/۶۰	۳۱-۶۰
۵	۰/۲۵	۰/۸۵	۶۱-۸۵
۶	۰/۱۵	۱/۰۰	۸۶-۰۰

مشتری	ارقام تصادفی ورود	زمان بین دو ورود	زمان ورود بر حسب ساعت	عدد تصادفی خدمتدهی	هابیل			خباز			علی			مدت انتظار در صف
					زمان شروع خدمتدهی	مدت خدمتدهی	زمان پایان خدمتدهی	زمان شروع خدمتدهی	مدت خدمتدهی	زمان پایان خدمتدهی	زمان شروع خدمتدهی	مدت خدمتدهی	زمان پایان خدمتدهی	
1	-	-	0	95				0	6	6			0	
2	26	2	2	25	2	3	5						0	0
3	98	4	6	51				6	4	10				0
4	90	4	10	92				10	6	16				0
5	26	2	12	89	12	6	18				14	4	18	0
6	42	2	14	38						20				2
7	74	3	17	13				17	3					0
8	80	3	20	61				20	5	25				0
.														
.														
.														
25	16	1	25	55							55	6	61	0
26	74	4	59	47				59	4	63				

نتایج بدست آمده از شبیه سازی:

- سرویس دهی توسط خباز: ۱۲ ماشین
- سرویس دهی توسط هابیل: ۸ ماشین
- سرویس دهی توسط علی: ۶ ماشین
- میانگین زمان انتظار در صف: ۰ دقیقه

۹. نانوایی سعی دارد تعیین کند که هر روز باید چند دوجین از نان خاصی بپزد. توزیع احتمال تعداد مشتریان این نوع نان به شرح زیر است:

تعداد مشتری در روز	۸	۱۰	۱۲	۱۴
احتمال	۰/۳۵	۰/۳۰	۰/۲۵	۰/۱۰

مشتری ها به موجب توزیع احتمال زیر، ۱، ۲، ۳، یا ۴ دوجین از این نان را سفارش میدهند.

تعداد دوجین نان سفارش داده شده توسط هر مشتری	۱	۲	۳	۴
احتمال	۰/۴	۰/۳	۰/۲	۰/۱

قیمت فروش هر دو جین از این نان ۵/۴ واحد پول و هزینه پختن هر دو جین ۳/۸۰ واحد پول است. هر مقدار از این نان که در پایان روز فروش نرفته باشد، به نصف قیمت فروخته می شود. بر

اساس ۵ روز شبیه سازی چند دوجین (به نزدیکترین مضرب ۱۰ گرد کنید) از این نان باید پخته شود؟

تعداد مشتری	احتمال	احتمال تجمعی	عدد تصادفی
۸	۰/۳۵	۰/۳۵	۰۱-۳۵
۱۰	۰/۳۰	۰/۶۵	۳۶-۶۵
۱۲	۰/۲۵	۰/۹۰	۶۶-۹۰
۱۴	۰/۱۰	۱/۰۰	۹۰-۰۰

جدول توزیع تعداد مشتری

دو جین سفارش	احتمال	احتمال تجمعی	عدد تصادفی
۱	۰/۴	۰/۴	۱-۴
۲	۰/۳	۰/۷	۵-۷
۳	۰/۲	۰/۹	۸-۹
۴	۰/۱	۱/۰	۰

جدول توزیع تعداد سفارش

ابتدا محاسبهی سود حاصل از فروش

سود = درآمد حاصل از فروش - هزینه تهیه نان + درآمد حاصل از فروش مازاد - سود از دست رفته به دلیل کمبود نان

: تعداد دوجین نان پخته شده در روزQ

$S = \sum_i O_i$ که در آن O_i میزان سفارش بر حسب دوجین به وسیله مشتری iام.

Q – S میزان فروش به صورت مازاد بر نیاز

S – Q میزان کمبود

$5.4\ \text{min} =$ سود

مشتری تعداد$\big) E$

$$
\begin{aligned}
E(\text{دوجین سفارش}) &= .4(1) + .3(2) + .2(3) + .1(4) = 2 \\
E(\text{دوجین فروش}) &= \bar{S} = (10.20)(2) = 20.4 \\
E(\text{سود}) &= 5.40\text{Min}(\bar{S}, Q) - 3.80Q + 2.70(Q - \bar{S}) - 1.60(\bar{S} - Q) \\
&= 5.40\text{Min}(20.4, Q) - 3.80Q + 2.70(Q - 20.4) \\
&\quad - 0.67(20.4 - Q) \\
E(\text{سود} \mid Q = 0) &= 0 - 0 + 1.60(20.4) = -32.64 \\
E(\text{سود} \mid Q = 10) &= 5.40(10) - 3.80(10) + 0 - 1.60(20.4 - 10) \\
&= -0.64 \\
E(\text{سود} \mid Q = 20) &= 5.40(20) - 3.80(20) + 0 - 1.60(20.4 - 20) \\
&= 15.36 \\
E(\text{سود} \mid Q = 30) &= 5.40(20.4) - 3.80(30) + 2.70(30 - 20.4) - 0 \\
&= 22.08 \\
E(\text{سود} \mid Q = 40) &= 5.40(20.4) - 3.80(40) + 2.70(40 - 20.4) - 0 \\
&= 11.08
\end{aligned}
$$

با توجه به نتایج تنها به ازای $Q=20,30,40$ باید شبیه سازی کرد.

بر اساس سود بیشتر یعنی $Q=30$ داریم:

روز	عدد تصادفی مشتری	تعداد مشتری	عدد تصادفی تقاضا	میزان سفارش	درآمد مازاد	سود از دست رفته
1	44	10	8	3	16.20	0
			2	1	5.40	0
			4	1	5.40	0
			8	3	16.20	0
			1	1	5.40	0
			6	2	10.80	0
			3	1	5.40	0
			0	4	21.60	0
			2	1	5.40	0
			0	4	21.60	0
				21	113.40	0

۱۰. تقاضا برای نوعی ابزار از توزیع احتمال نشان داده شده در در زیر پیروی می کند:

تقاضای روزانه	۰	۱	۲	۳	۴
احتمال	۰/۳۳	۰/۲۵	۰/۲۰	۰/۱۲	۱۰

موجودی انبار هر ۷ روز یک بار بررسی می شود و اگر موجودی به ۶ واحد یا رسیده باشد، ۱۰ عدد از این ابزار سفارش داده می شود. مهلت تحویل احتمالی است و طبق توزیع زیر تعریف می شود:

مهلت تحویل	۱	۲	۳
احتمال	۰/۳	۰/۵	۰/۲

شبیه سازی وقتی شروع می شود که آغاز هفته است، ۲۰۰ عدد از ابزار فوق موجود است وهیچ سفارشی با تأخیر روبرو نیست. ۶ هفته از سیستم فوق را شبیه سازی و سیستم را تجزیه تحلیل نمایید. به منظور تأثیر افزایش یا کاهش ۱) دوره بررسی، ۲) مقدار سفارش مجدد و ۳) نقطه زمان سفارش مجدد بر کمبودها، شبیه سازی بیشتری انجام دهید.

تقاضای روزانه	احتمال	احتمال تجمعی	عدد تصادفی
0	.33	.33	01-33
1	.25	.58	34-58
2	.20	.78	59-78
3	.12	.90	79-90
4	.10	1.00	91-00

مهلت تحویل	احتمال	احتمال تجمعی	عدد تصادفی
1	.3	.3	1-3
2	.5	.8	4-8
3	.2	1.0	9-0

روز تا رسیدن سفارش
1 0
2
1 0
1

احتمال کمبود: ۰/۲۵

متوسط پایان موجودی: ۳/۵

۱۱. یک شرکت تأمین کننده وسایل لوله کشی به تعیین توزیع تقاضا در مدت تحویل آبگیرهای صنعتی علاقمند است. تابع فراوانی تقاضای روزانه معلوم و به شرح زیر است:

تقاضای روزانه	۰	۱	۲	۳	۴
احتمال	۰/۱۸	۰/۳۹	۰/۲۹	۰/۰۹	۰/۰۵

توزیع مهلت تحویل بر اساس سوابق موجود ایجاد شده و به شرح زیر است:

مهلت تحویل	۰	۱	۲	۳	۴	۵
احتمال	۰/۱۳۵	۰/۲۲۳	۰/۲۸۸	۰/۲۱۳	۰/۱۱۸	۰/۰۲۳

بر اساس ۲۰ دوره مهلت تحویل، توزیع تقاضا در مهلتهای تحویل را ایجاد کنید. با استفاده از فواصل ۲-۰، ۵-۳، ... هیستوگرام این توزیع را تهیه کنید و سپس با استفاده از فواصل ۱-۰، ۳-۲، ۴-۳، ... هیستوگرام دیگری ایجاد کنید. آیا تغییر عرض فاصله تأثیر عمده ای بر شکل هیستوگرام توزیع تقاضا در مهلت تحویل می گذارد؟

تقاضای روزانه	احتمال	احتمال تجمعی	عدد تصادفی
0	.18	.18	01-18
1	.39	.57	19-57
2	.29	.86	58-86
3	.09	.95	87-95
4	.05	1.00	96-00

مهلت تحویل	احتمال	احتمال تجمعی	عدد تصادفی
0	.135	.135	001-135
1	.223	.358	136-358
2	.288	.646	359-646
3	.213	.859	647-859
4	.118	.977	860-977
5	.023	1.000	978-000

دوره	عدد تصادفی مهلت تحویل	مهلت تحویل	عدد تصادفی	تقاضا	مهلت تقاضا
1	024	0	-	-	0
2	330	1	14	0	0
3	288	1	53	1	1
4	073	0	-	-	0
5	197	1	24	1	1
6	924	4	53	1	
			81	2	
			70	2	
			18	0	5

"توضیح": به نظر می رسد که هیستوگرام با عرض کمتر فاصله زمانی (۱ واحد زمانی) از فاصله زمانی با عرض بیشتر دارای قابلیت توصیف پذیری بیشتر و

۱۴. یک شرکت تاکسیرانی بین ساعت ۹ صبح تا ۵ بعد از ظهر با یکی خودرو فعالیت می کند. در حال حاضر، افزودن خودرو دومی به این اتومبیل در دست بررسی است. تقاضا برای تاکسی از توزیع نشان داده شده در زیر پیروی می کند:

مدت بین تقاضای تلفنی	۱۵	۲۰	۲۵	۳۰	۳۵
احتمال	۰/۱۴	۰/۲۲	۰/۴۳	۰/۱۷	۰/۰۴

توزیع مدت کامل کردن هر خدمتدهی به شرح زیر است:

مدت خدمتدهی	۵	۱۵	۲۵	۳۵	۴۵
احتمال	۰/۱۲	۰/۳۵	۰/۴۳	۰/۱۷	۰/۰۴

پنج روز کار سیستم فعلی و سیستمی با یک تاکسی اضافه را شبیه سازی کنید. دو سیستم را برحسب مدت های انتظار مشتریان و هر معیار روشنگر دیگری مقایسه کنید.

زمان بین تقاضای تلفنی	احتمال	احتمال تجمعی	عدد تصادفی
15	.14	.14	01-14
20	.22	.36	15-36
25	.43	.79	37-79
30	.17	.96	80-96
35	.04	1.00	97-00

مدت خدمتدهی	احتمال	احتمال تجمعی	عدد تصادفی
5	.12	.12	01-12
15	.35	.47	13-47
25	.43	.90	48-90
35	.06	.96	91-96
45	.04	1.00	97-00

ابتدا شبیه سازی برای یک تاکسی در ۵ روزو سپس برای ۲ تاکسی در ۵ روز

مقایسه:

یک تاکسی

روز	تقاضای تلفنی	عدد تصادفی زمان بین تقاضای تلفنی	زمان بین تقاضای تلفنی	زمان تماس	عدد تصادفی زمان خدمتدهی	زمان خدمتدهی	شروع	انتظار	پایان	زمان مشتری در سیستم	زمان بیکاری
1	1	15	-	0	01	5	0	0	5	5	0
	2	01	20	20	53	25	20	0	55	25	0
	3	14	15	35	62	25	55	20	80	45	0
	4	65	25	60	55	25	80	20	105	45	0
	5	73	25	85	95	35	105	20	140	55	0
	6	48	25	110	22	15	140	30	155	45	0
	. . . 20	77	25	444	63	25	470	25	495	50	0
2 . .											

- **مجموع زمان بیکاری = ۲۶۵ دقیقه = ۴/۴ ساعت**
- **متوسط زمان بیکاری به ازای هر تماس تلفنی = ۲/۷ دقیقه**
- **نسبت زمان بیکاری = ۰/۱۱**
- **مجموع زمان انتظار مشتری = ۱۲۳۰ دقیقه**
- **متوسط زمان انتظار به ازای هر مشتری = ۱۱/۹ دقیقه**
- **تعداد مشتری های منتظر = ۶۱ (از ۱۰۳ مشتری)**
- **احتمال اجبار مشتری برای انتظار = ۰/۵۹**
- **متوسط زمان انتظار مشتری = ۲۰/۲ دقیقه**

					تاکسی 1			تاکسی 2						
روز	تقاضای تلفنی	زمان بین تقاضای تلفنی	زمان تماس	زمان خدمتدهی	شروع	زمان خدمت	پایان	شروع	زمان خدمت	پایان	زمان انتظار مشتری	زمان مشتری در سیستم	زمان بیکاری تاکسی 1	زمان بیکاری تاکسی 2
1	1	-	0	5	0	5	5				0	5		
	2	20	20	25	20	25	45				0	25		
	3	15	35	25				35	25	60	0	25		35
	4	25	60	25	60	25	85				0	25	15	
	5	25	85	35	80	35	120				0	35		
	6	25	110	15				110	15	125	0	15		50
	. . . 20	20	480	25	480	25	505				0	25	10	
2 . . .														

- زمان بیکاری تاکسی۱ = ۶۸۵ دقیقه
- زمان بیکاری تاکسی۲ = ۱۹۱۵ دقیقه
- مجموع زمان بیکاری = ۲۶۰۰ دقیقه = ۴۳ ساعت
- متوسط زمان بیکاری به ازای هر تماس تلفنی = ۲۵/۷ دقیقه
- نسبت زمان بیکاری = ۰/۵۴

- مجموع زمان انتظار مشتری = ۰ دقیقه
- تعداد مشتری های منتظر = ۰ مشتری

۲۰- بالابری در یک کارخانه تولیدی موادی دقیقاً به وزن ۴۰۰ کیلوگرم را حمل می کند. این مواد از سه نوعند و برای انتقال توسط بالابر به جعبه هایی وارد می شوند. اطن مواد و توزیع های مدت بطن ورودشان به شرح زیر است.

ماده	وزن (کیلوگرم)	مدت بین ورود (دقیقه)
A	۲۰۰	۵ ± ۲ (یکنواخت)
B	۱۰۰	۶ (ثابت)
C	۵۰	P(2) = 0.33
		P(3) = 0.67

بالابر در بالا رفتن به طبقه دوم یک دقیقه در تخلیه بار دو دقیقه و در بازگشت به طبقه اول یک دقیقه وقت صرف می کند. یک ساعت از عملکرد سیستم را شبیه سازی نمایید.
متوسط مدت انتقال یک جعبه مواد A چقدر است؟
متوسط مدت انتظار یک جعبه مواد B چقدر است؟
چند جعبه مواد C طی یک ساعت منتقل شده است؟

مواد A (۲۰۰ کیلوگرم):

مدت بین ورود	احتمال	احتمال تجمعی	عدد تصادفی
3	.2	.2	1-2
4	.2	.4	3-4
5	.2	.6	5-6
6	.2	.8	7-8
7	.2	1.0	9-0

جعبه	عدد تصادفی مدت بین دو ورود	مدت بین ورود	ساعت
1	1	3	3
2	4	4	7
3	8	6	13

جعبه	1	2	3	...	10
ساعت	6	12	18	...	60

مواد B (۱۰۰ کیلوگرم):

مواد C (۵۰ کیلوگرم):

عدد تصادفی	احتمال تجمعی	احتمال	مدت بین ورود
01-33	.33	.33	2
34-00	1.00	.67	3

ساعت	مدت بین ورود	عدد تصادفی مدت بین دو ورود	جعبه
3	5	58	1
6	3	92	2
9	3	87	3
11	2	31	4
⋮	⋮	⋮	⋮
60	3	62	22

مجموع مواد:

ساعت	ورود A	ورود B	ورود C
3	1		1
6		1	2
7	2		
9			3
11			4
12		2	
⋮			

میانگین زمان انتقال برای مواد A:

$$\bar{t}_A = \frac{\mathbf{A}}{}$$

میانگین زمان انتظار B:

تعداد نهایی جعبه های C که منتقل شده:

ساعت	تعداد A در صف	تعداد B در صف	تعداد C در صف	وزن صف	زمان شروع خدمت دهی	زمان پایان خدمت دهی	زمان A در صف	زمان B در صف	شمارنده A	شمارنده B	شمارنده C
3	1	0	1	250							
6	0	0	0	0	6	10	3	0	1	1	2
7	1	0	0	200							
9	1	0	1	250							
11	1	0	2	300							
12	0	0	0	350	12	16	5	0	2	2	4
.											
.											
.											

۲۲ جعبه.

تمرین فصل سوم:

۱. (الف) با استفاده از رهیافت زمانبندی پیشامدها شبیه سازی دستی باجه صندوق را که در مثال ۳-۲ جدول ۳-۱ شروع شد، ادامه دهید. از همان مدتهای بین ورود و خدمتدهی که قبلاً تولید و در مثال ۲-۱ بکار گرفته شد، استفاده نمایید. با استفاده از آخرین مدت بین دو ورود، شبیه سازی را ادامه دهید تا سیستم خالی شود. نتایج بدست آمده از این قسمت را با پاسخ های مثال ۲-۱ مقایسه کنید.

مانند مثال ادامه دهید.

(ب) تمرین ۳-۱ (الف) را با افزودن اجزاء لازم به منظور برآورد میانگین مدت پاسخ و در صد متقاضیانی که ۴ دقیقه یا بیشتر در سیستم می مانند بار دیگر انجام دهید.

اجزایی که باید اضافه شوند:

S: مجموع مدتهای پاسخ برای تمام مشتریانی که تا زمان حال سیستم را ترک کرده اند.

F: جمع تعداد مشتریانی که ۴ دقیقه یا بیشتر در سیستم می مانند.

N_D: جمع موارد ترک سیستم تا زمان حال.

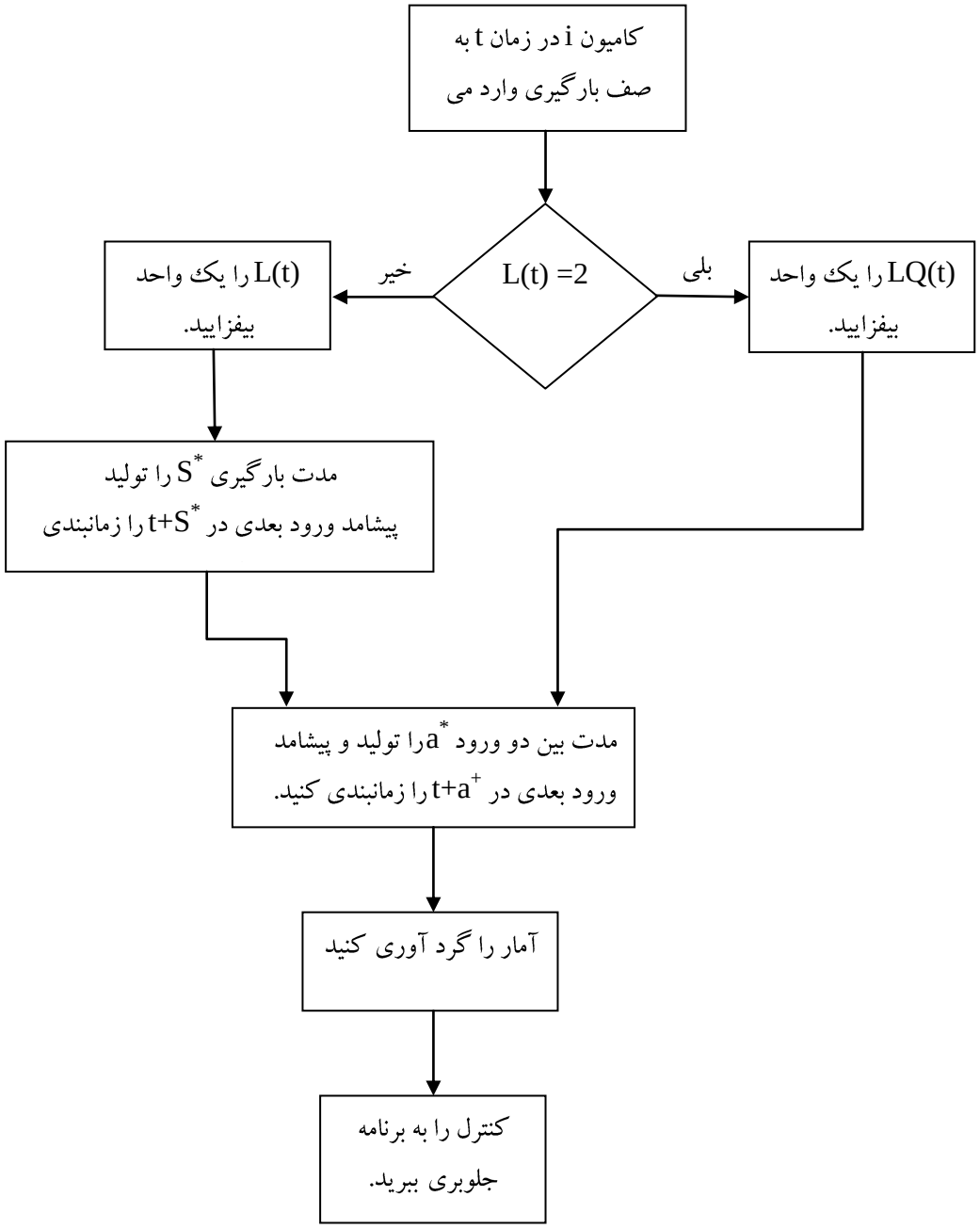
(ج) در زمینه مزایای نسبی شبیه سازی دستی و کامپیوترب در مقایسه با یکدیگر اظهار نظر کنید.

شبیه سازی دستی این امکان را می دهد که به خوبی فرآیند شبیه سازی مشاهده و مرحله به مرحله اجرا شود. علاوه بر این امکان مشاهده الگوریتم شبیه سازی (زمانبندی و جلوبری زمان) وجود دارد. نیاز به امکانات خاصی ندارد.

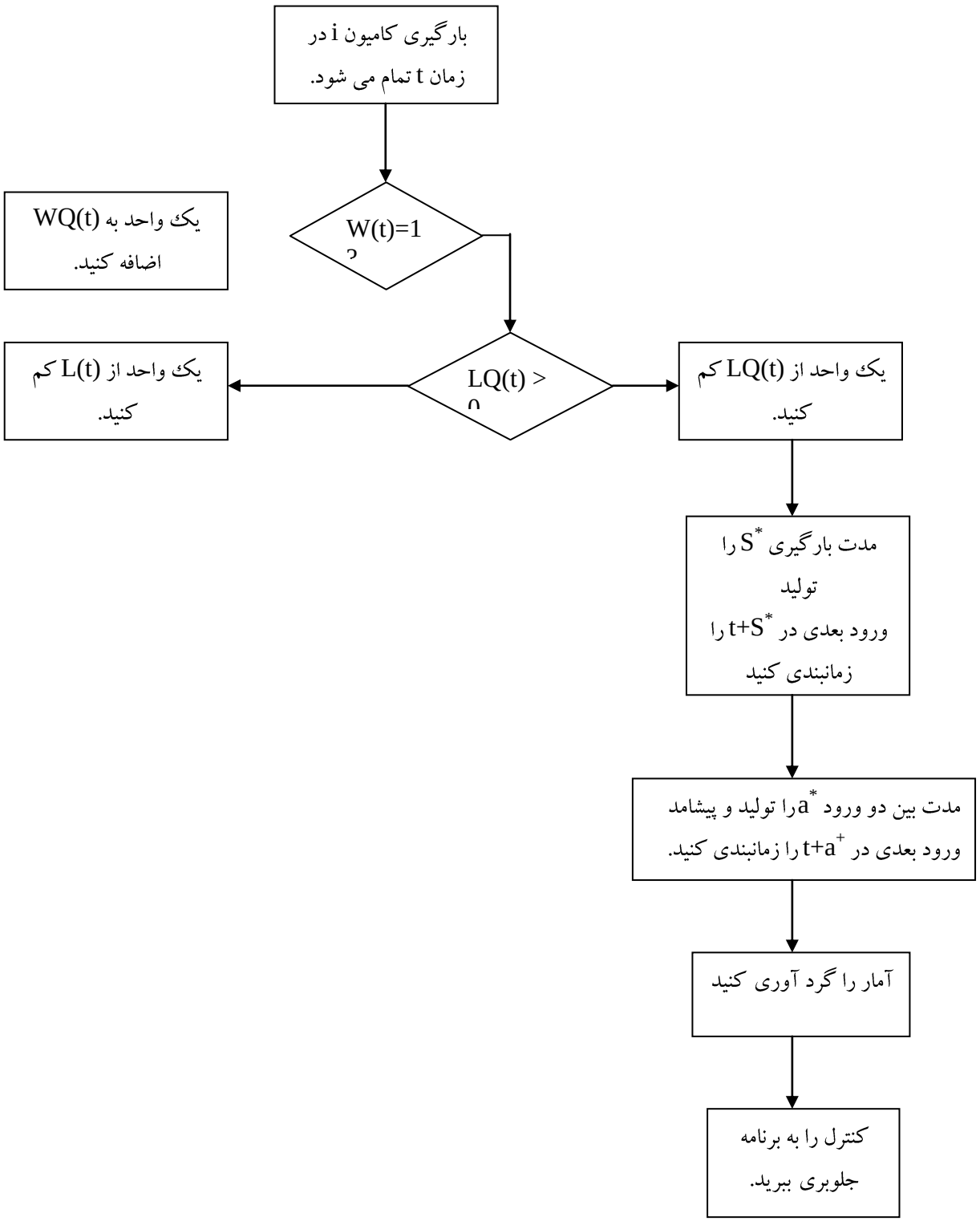
در نقطه مقابل شبیه سازی کامپیوتری سرعت و دقت بالاتری دارد. امکان اجرای مدل برای تعداد بسیار زیاد ورودیها و زمان طولانی وجود دارد. امکان بسط زمان و فشرده سازی آن به شکل ساده تری در دسترس است. هزینه کمتری دارد.

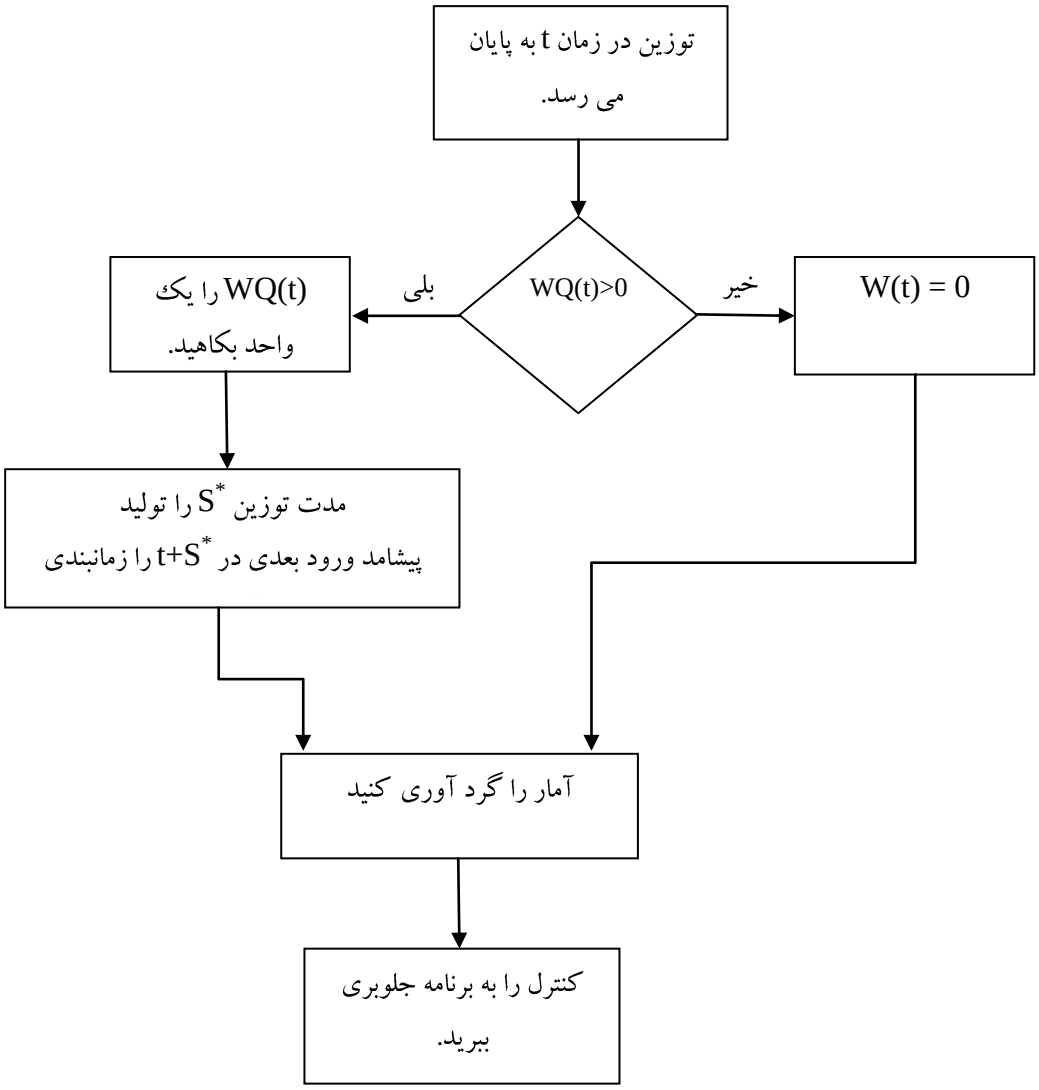
۲. دیاگرام های منطق پیشامد را برای مسأله کامیونهای کمپرسی، مثال ۳-۴، ایجاد کنید.

پیشامد ورود به صف بارگیری



پیشامد اتمام بارگیری




توزین در زمان t به پایان
می رسد.
WQ(t)>0
بلی
خیر
WQ(t) را یک
واحد بکاهید.
W(t) = 0
مدت توزین S* را تولید
پیشامد ورود بعدی در t+S* را زمانبندی
آمار را گرد آوری کنید
کنترل را به برنامه جلوبری
ببرید.

۳. در مسأله کامیون های کمپرسی در مثال ۳-۴، برآورد میانگین مدت پاسخ و درصد مدت های پاسخ بزرگتر از ۳۰ دقیقه مورد نظر است. هر مدت پاسخ برای یک کامیون از لحظه ورود کامیون به صف بارگیری آغاز و در لحظه ای که توزین پایان می یابد، تمام می شود. اجزاء مدل و آمار تجمعی مورد نیاز به منظور براورد این دو معیار عملکرد سیستم را اضافه کنید و شبیه سازی را به مدت ۸ ساعت انجام دهید.

پاسخ: اگر میانگین مدت پاسخ سیستم یا نسبت کامیونهایی را که بیش از ۳۰ دقیقه در سیستم می مانند، براورد می کردیم به طوری که سیستم صف بارگیری و دستگاه های بارگیری و صف توزین و باسکول را در برگیرد، نهادهای کامیون (DT_i) به اضافه یک ویژگی یعنی زمان ورود به صف بارگیری مورد نیاز خواهد بود. با ترک باسکول توسط یک کامیون مدت پاسخ این کامیون را ممکن بود به صورت زمان کنونی شبیه سازی (t) منهای ویژگی زمان ورود محاسبه نمود. این مدت پاسخ جدید برای تازه کردن آمار تجمعی مورد استفاده قرار می گیرد:

S: مجموع مدت پاسخ تمام کامیون هایی که سراسر سیستم را پیموده اند و

F: تعداد مدتهای پاسخ کامیونها که بیش از ۳۰ دقیقه است.

۴. مثال ۲-۲ (مسأله اتورستوران هابیل و خباز) را با استفاده از رهیافت زمانبندی پیشامدها دوباره با دست شبیه سازی کنید.

- حالت سیستم (L(t), LQ(t)) که در آن LQ(t) تعداد افراد منتظر و L(t) تعداد افرادی که خدمتدهی می کنند را نشان می دهد.
- نهادها: اگر آمار مربوط به کارکرد هریک از افراد مدنظر باشد، باید برای هابیل نهادی مثل (H, t) و برای خباز نهادی مثل (B, t) را در نظر گرفت. در غیر اینصورت نیازی نیست.
- فعالیت ها: به شرح جدول های ۱۱، ۱۲ و ۱۳ از فصل ۲.
- پیشامدها:

(A,t,H): یعنی هابیل در زمان t خدمت دهی را شروع کرده است.

(D,t,H): یعنی هابیل در زمان t خدمت دهی را به پایان رسیده است.

(D,t,B): یعنی خباز در زمان t خدمت دهی را شروع کرده است.

(D,t,B): یعنی خباز در زمان t خدمت دهی را به پایان رسانده است.

- مجموعه زمان شروع خدمتدهی

۸. مثال ۲-۴ (سیستم موجودی (M,N)) را با استفاده از رهیافت زمانبندی پیشامدها دوباره شبیه سازی نمایید.

حالت سیستم: (ME(t),R(t)) که در آن R(t) نشاندهنده تقاضاست و ME(t) نشاندهنده موجودی است.

فعالیتها: جداول ۱۹ و ۲۰ فصل دوم

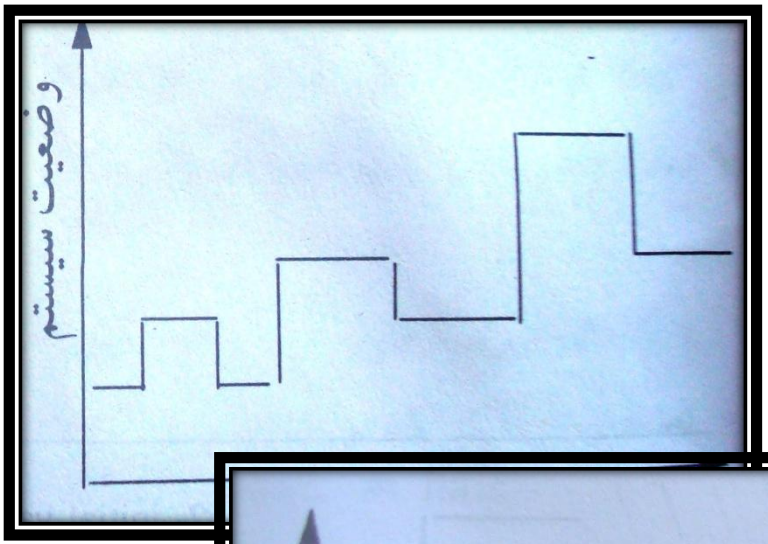
آمار: MP موجودی در پایان دوره است و F مقدار سفارش و S میزان کمبود است.

پیشامدها: ((M,R,t) یعنی M واحد تقاضا در زمان t رسیده است.

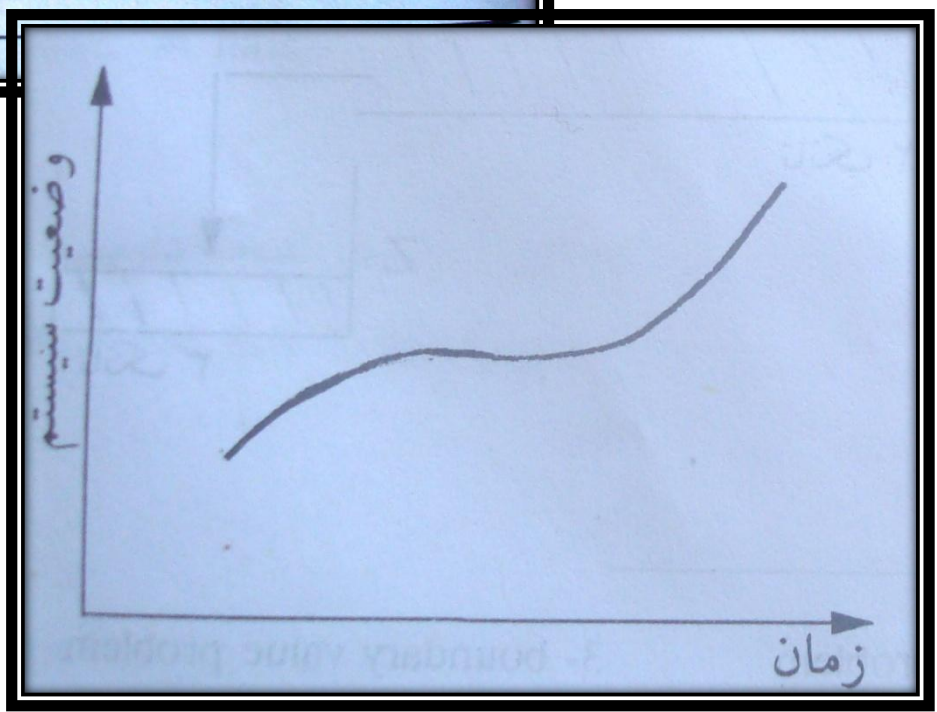
سیستم های گسسته و پیوسته:

همراه با گذر زمان مقدار و بعضی از مشخصه های اشیاء سیستم (متغیرها) تغییر می یابد. بعبارت دیگر بعضی از اشیاء و در نتیجه سیستم تغییر وضعیت می دهند. این تغییرات ممکن است بصورت پیوسته (نسبت به زمان) یا بصورت ناپیوسته و در لحظات متمایزی از زمان باشند. بعنوان مثال یک سیستم بانک را در نظر بگیرید، تعداد مشتری ها در بانک یکی از مشخصه های سیستم است. تغییرات این مشخصه بصورت ناپیوسته با ورود و خروج مشتری ها صورت می گیرد، بطوریکه منحنی آن حالت پله ای دارد. یک ورود باعث افزایش و یک خروج باعث کاهش آنی آن می گردد. بعبارت دیگر تغییر وضعیت سیستم بصورت ناپیوسته است.

اکنون یک سیستم تصفیه خانه مایعات را در نظر بگیرید. منابع مایع تصفیه نشده و تصفیه شده از اشیاء سیستم بوده و مقدار مایع درون هریک از آنها مشخصه ای برای هریک می باشد. تغییرات این مشخصه ها نسبت به گذر زمان متصل است. شکل های ۱ و۲ تغییر وضعیت را در سیستم های گسسته و پیوسته نشان می دهد:



شکل ۱: وضعیت سیستم های گسسته



شکل ۲: وضعیت سیستم های پیوسته

وضعیت مدل سیستم های پیوسته :

مثال : بررسی رشد جمعیت خرگوش و روباه: این مدل بصورت فرض می شود:

R_0 وF_0 ثابت می باشند و تعدا خرگوش و روباه را در شروع (t=0) شبیه سازی تعیین می کنند.

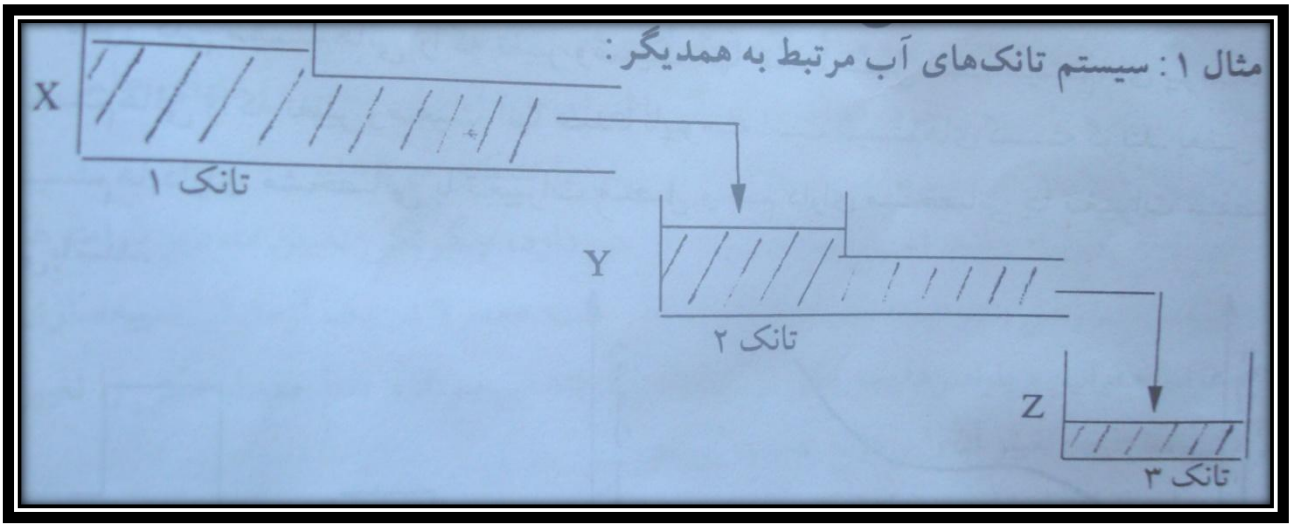
أ. دو معادله فوق تنها شامل مشتق های معمولی و نه مشتق های جزئی می باشد، در صورت نتیجه مدل فوق، مدل با معادلات معمولی نامیده می شود.

ب. در مدل فوق، تعداد خرگوش ها و تعداد روباه ها در شروع شبیه سازی تعیین گردیده است. چنین مدل هایی، مسئله با مقدار اولیه نامیده می شود.

ج. دسته دیگری از مسائل که در آنها، شرایط در دو لحظه زمانی تعیین می شود ، مسائل با مقدار مرزی نامیده می شود. برای مثال ، برای یک تعداد خرگوش داده شده، تعداد روباهی که در ابتدا نیاز است تا تعداد خرگوش ها در زمان مشخصی در آینده تعداد معینی باشد، مسئله با مقدار مرزی دو بخشی نامیده می شود.

مثال های بیشتری راجع به سیستم های پیوسته و گسسته:

مثال۱: سیستم تانک های آب مرتبط به همدیگر:



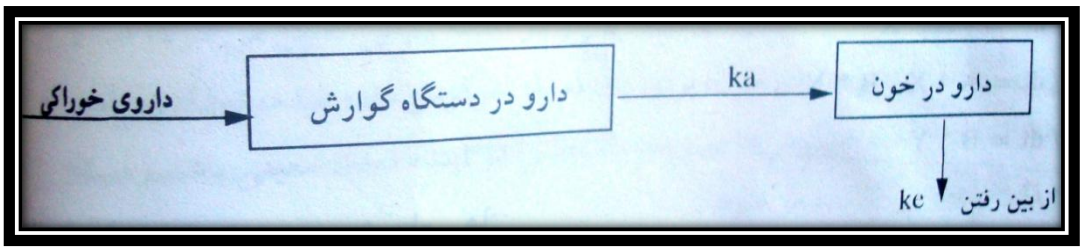
Z,Y,X ارتفاع مایعات را در سه تانک نشان می دهد. مقطع تانک ها یکسان است.

مدل ریاضی:

A وB ثابت هستند.

اگر تمام آب از تانک ۱ شروع شود چگونه ارتفاع آب در سه تانک تغییر می یابد؟ و ماکزیمم مقدار برای Y چقدر است؟

مثال: بررسی تعیین غلظت دارو در خون



معادله تعیین مقدار دارو در دستگاه گوارش:

معادله تعیین مقدار دارو در خون:

بهتر است به جای تعیین مقدار دارو در خون، غلظت دارو در خون اندازه گیری شود.

هدف از شبیه سازی انجام آزمایش های زیر می باشد:

۱. مقدار G و C را بعد از ۱۲ ساعت بدست آورید.
۲. مقدار C در چه موقعی ماکزیمم است.
۳. مقدار دارو را از ۲۰ تا ۲۰۰ واحد هربار ۲۰ واحد تغییر داده و مشاهده نمایید مواقعی که C ماکزیمم می شود.
۴. برای یک مقدار معین C بعد از ۶ ساعت چه مقدار دارو باید به فرد مریض داده شود.
۵. برای مقدار ماکزیمم C بعد از ۶ ساعت چه مقدار دارو باید به فرد مریض داده شود.

مثال: بررسی فراگیر یک بیماری واگیردار

جمعیت یک اجتماع N و فرضیات زیر را در نظر بگیرید:

- هر فردی که مبتلا شده است از ابتلای مجدد مصون است.
- هر فردی که مبتلا نشده است، آمادگی ابتلا توسط فرد مبتلا را دارد.
- افراد مبتلا برای یک مدت ثابت می باشد.

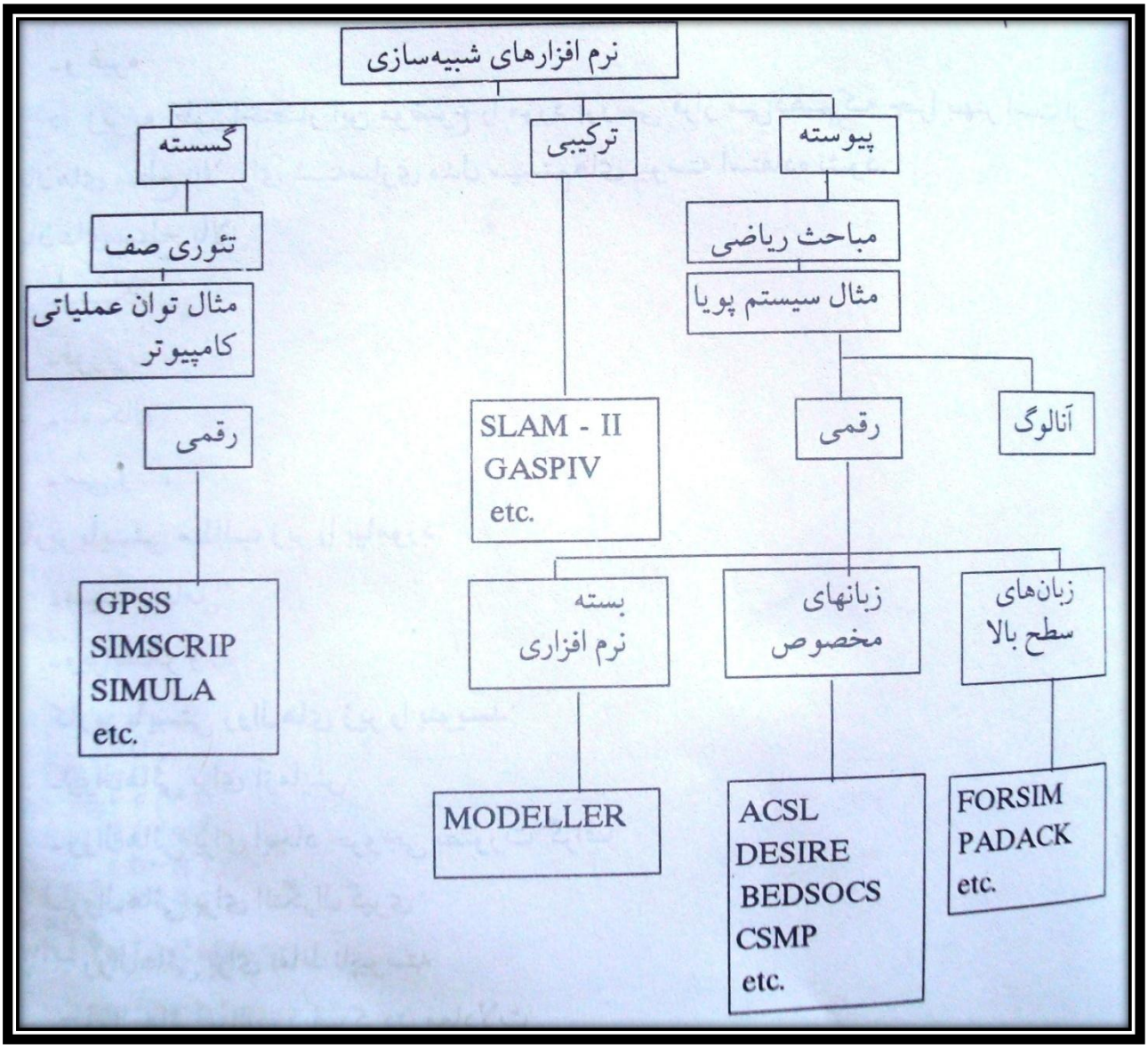
مدل ریاضی:

جمعیت در ابتدا ۱۰۰۰ نفر، I(0)=20،R(0)=80 ، U(0))=900، به سوالات زیر پاسخ دهید:

أ. ماکزیمم افرادی که مبتلا شده اند، چند نفر می باشند؟

ب. اگر T زمانی باشد که این بیماری پایان می یابد، یعنی $I(T)=0$، در این موقع نشان دهید که چند نفر از این جمعیت به این بیماری مبتلا نشده اند. توجه نمایید در زمان T مردم از ابتلا به این بیماری مصون می باشند.

نرم افزارهای شبیه سازی:



در زیر به اختصار امکاناتی که نرم افزارهای شبیه سازی بایستی برای شبیه سازی مدل سیستم های پیوسته بوجود آورند مورد بررسی قرار گرفته اند. این امکانات بایستی مدل های پیچیده همراه با آزمایش برروی آن (از قبیل بهینه سازی) را شبیه سازی نمایند بدون آنکه کاربر از نظر برنامه نویسی مهارت کافی داشته باشد. این امکانات عبارتند از:

- روال های کتابخانه ای، همانند زبانهای سطح بالا

- روال های کتابخانه ای ، برای آزمایش
- روال های مربوط به ایجاد خروجی بصورت گراف
- روال های انتگرال گیری
- روال های مربوط به زیر مدل ها
- روال های مربوط به نقاط ناپیوسته
- و غیره

در زیر به طور اختصار این موضوع را مورد بررسی قرار می دهیم که چرا بهتر است از زبان های سطح بالا برای شبیه سازی مدل سیستم های پیوسته استفاده نشود.

زبان های سطح بالا:

- بیسیک
- فورترن
- پاسکال
- C و ...

کاربرد بایستی مطالب زیر را بیاموزد:

- سیستم عامل
- ویرایشگر و ...

کاربرد بایستی روال های زیر را بنویسد:

- روال هائی برای آزمایش
- روال هائی برای ایجاد خروجی بصورت گراف
- روال هائی برای انتگرال گیری
- روال هائی برای نقاط ناپیوسته
- روال هائی برای ردیف کردن معادلات
- روال کتابخانه ای و...

حال اجازه بفرمایید مثال ساده مربوط به سیستم تانک های آب مرتبط به همدیگر را با زبان سطح بالای C^{++} شبیه سازی نمائیم. در این برنامه توجه نمایید که کاربر بایستی از قابلیت بالایی در برنامه نویسی برخوردار باشد تا بتواند مثلاً در این نثال روالهایی برای خروجی بصورت نمودار و روالهایی برای انتگرال گیری بنویسد، در حالی که اگر از زبان های مخصوص شبیه سازی استفاده نماید نیازی به نوشتن این روالها نیست. توجه نمایید که استفاده کنندگان شبیه سازی را پژوهشگران تمام رشته ها از جمله رشته های علوم انسانی و علوم پایه تشکیل می دهند که ممکن است هیچ علاقه ای به یادگیری زبان نویسی نداشته باشند.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
#include <iostream.h>
#define FALSE 0
#define TRUE  1
float f(float xs,float ys);
type_results();
display_graph();
display_tanks();
drawtanks();
int num=1;
float yy;
int flag=FALSE;
   int k,M=40,gd=DETECT,gm,choice;
   float h,t[55],x[55],y[55],z[55],a=0,b=20;
main()
{

   clrscr();
   y[0]=0;
   z[0]=0;
   printf("\ninitial amount of water in Tank 1(0-70):")
   scanf("%f",&x[0]);
   h = (b-a)/M;
   t[0] = a;
   k=0;
   while(k<=M-1)
   {
      switch(num){
         case 1:
            x[k+1] = x[k]+h*f(x[k],y[k]);
            break;
         case 2:
            y[k+1] = y[k]+h*f(x[k],y[k]);
            break;
         case 3:
            z[k+1] = z[k]+h*f(x[k],y[k]);
            k++;
            break;
      }
      t[k+1] = a+h*(k+1);
   }
   clrscr();
   cout<<"\n1.Type results";
   cout<<"\n2.Display graph";
   cout<<"\n3.Display tanks";
   cout<<"\n4.Exit";
   cout<<"\nEnter choice:";
   cin>>choice;
```

برنامه سیستم تانک های آب مرتبط به همدیگر به زبان C++

```
while(choice!=4){
   switch(choice){
      case 1:
          type_results();
          break;
      case 2:
          display_graph();
          break;
      case 3:
          display_tanks();
          break;
   }/*switch*/
   //  putimage(91,150-x[k],buf1,COPY_PUT);
     buf2=malloc(imagesize(241,249-y[k],349,249));
     getimage(241,249-y[k],349,249,buf2);
     bar(241,249-y[k],349,249);
     delay(30);
   //  putimage(241,249-y[k],buf2,COPY_PUT);
     buf3=malloc(imagesize(391,349-z[k],499,349));
     getimage(391,349-z[k],499,349,buf3);
     bar(391,349-z[k],499,349);
     delay(30);
     getche();
     putimage(91,150-x[k],buf1,COPY_PUT);
     putimage(241,249-y[k],buf2,COPY_PUT);
     putimage(391,349-z[k],buf3,COPY_PUT);
     free(buf1);
     free(buf2);
     free(buf3);
   }
   bar(391,349-z[k-1],499,349);
   outtextxy(500,450,"press a key ...");
   getch();
   closegraph();
}

float f(float xs,float ys)
{
   switch(num){
      case 1:
        num++;
```

```
drawtanks()
{
   int gd=DETECT,gm;

   cleardevice();
   line(90,80,90,150);
   line(90,150,250,150);
   line(200,130,200,80);
   line(200,130,250,130);
   line(200,130,250,"Tank 1");
   outtextxy(140,160,250);
   line(240,180,240,250);
   line(240,250,400,250);
   line(350,230,350,180);
   line(350,230,400,230);
   line(350,230,400,"Tank 2");
   outtextxy(290,260,390,350);
   line(390,280,390,350);
   line(390,350,500,350);
   line(500,350,500,280);
   outtextxy(425,360,"Tank 3");
}
   clrscr();
   cout<<"\n1.Type results";
   cout<<"\n2.Display graph";
   cout<<"\n3.Display tanks";
   cout<<"\n4.Exit";
   cout<<"\nEnter choice:";
   scanf("%d",&choice);
}/*while*/
clrscr();
}
/////////* END MAIN *///////////

type_results()
{
   clrscr();
   cout<<"Time    Tank 1      Tank 2      Tank 3\n";
   for(k=0;k<=22;k++)
      printf("\n%.2f   %f   %f   %f",t[k],x[k],y[k],z[k]);

   getch();
   for(k=22;k<=M;k++)
      printf("\n%.2f   %f   %f   %f",t[k],x[k],y[k],z[k]);
   gotoxy(60,25);
   cout<<"press a key ...";
   getch();
}

display_graph()
{
   char *str;
   initgraph(&gd,&gm,"d:\\tc\\bgi");
   line(50,40,50,400);
   line(50,400,640,400);
   outtextxy(0,40,"liquid");
   outtextxy(0,55,"height");
   outtextxy(600,410,"time");
```

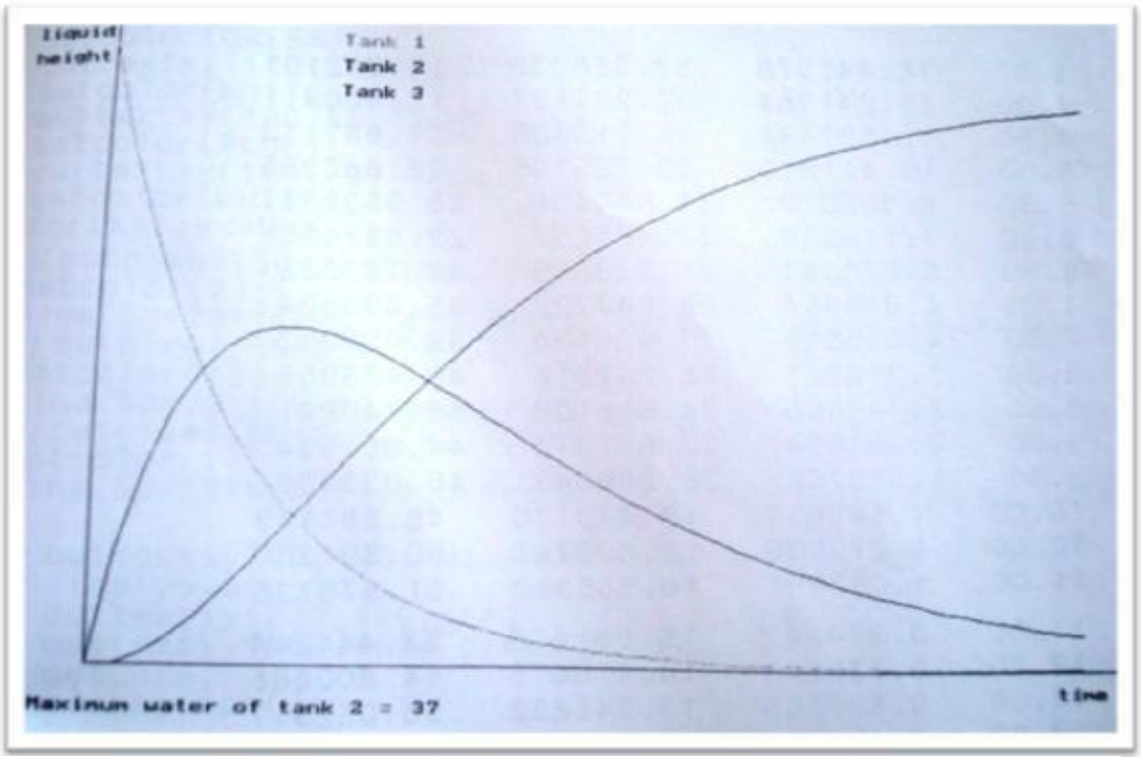
```
setcolor(GREEN);
outtextxy(180,40,"Tank 1");
setcolor(WHITE);
outtextxy(180,55,"Tank 2");
setcolor(BLUE);
outtextxy(180,70,"Tank 3");
setcolor(WHITE);
for(k=1;k<=M-1;k++){
//putpixel(50+29*t[k],400-5*x[k],LIGHTGREEN);
setcolor(2);
line(50+29*t[k-1],400-5*x[k-1],50+29*t[k],400-5*x[k]);
//putpixel(50+29*t[k],400-5*y[k],WHITE);
setcolor(15);
line(50+29*t[k-1],400-5*y[k-1],50+29*t[k],400-5*y[k]);
//putpixel(50+29*t[k],400-5*z[k],LIGHTBLUE);
setcolor(1);
line(50+29*t[k-1],400-5*z[k-1],50+29*t[k],400-5*z[k]);
 }
 outtextxy(20,420,"Maximum water of tank 2 = ");
 itoa(yy,str,10);
 outtextxy(230,420,str);
 outtextxy(500,450,"press a key...");
 getch();
 closegraph();

isplay_tanks()

 void *buf1,*buf2,*buf3;
 initgraph(&gd,&gm,"d:\\tc\\bgi");
 drawtanks();
 setfillstyle(1,BLUE);
 for(int k=0;k<=M;k++){
    buf1=malloc(imagesize(91,150-x[k],199,149));
    getimage(91,150-x[k],199,149,buf1);
    bar(91,150-x[k],199,149);
    delay(30);
```

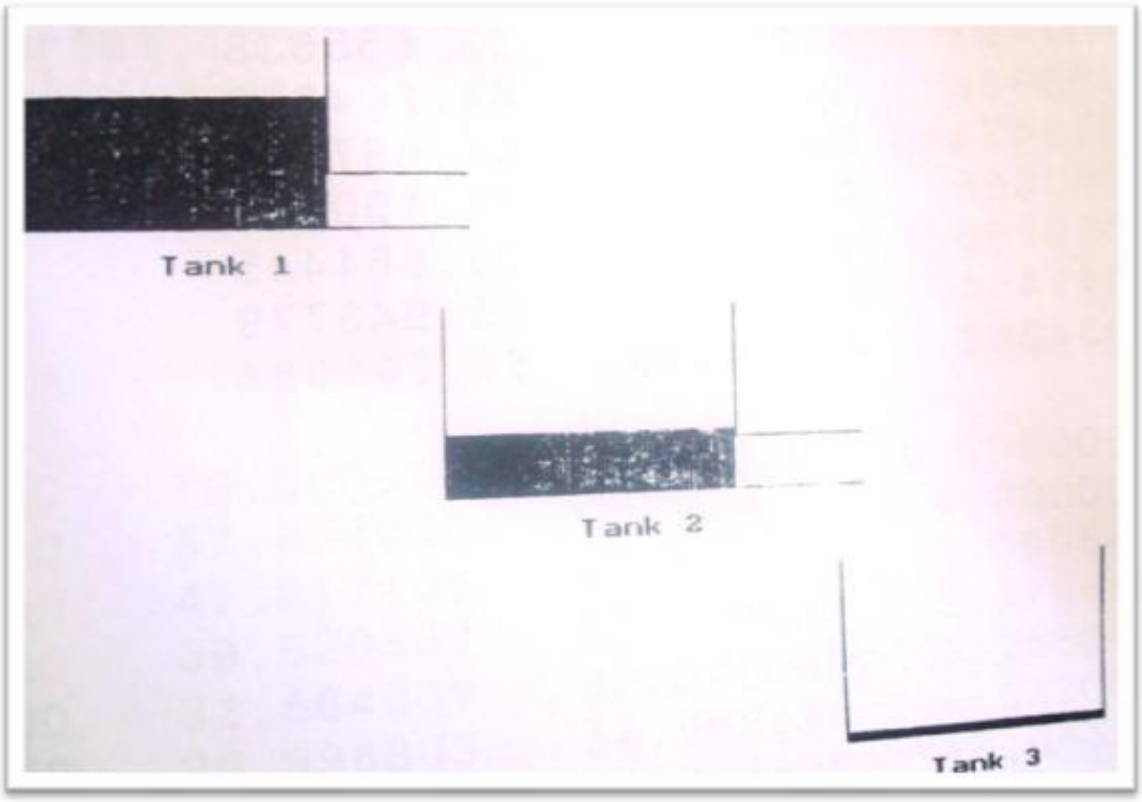
خروجی عددی سیستم تانک‌های آب مرتبط به همدیگر

Time	Tank 1	Tank 2	Tank 3
0.00	70.000000	0.000000	0.000000
0.50	57.855000	12.145000	0.000000
1.00	47.817158	21.132299	1.050543
1.50	39.520882	27.600632	2.878486
2.00	32.664009	32.070049	5.265941
2.50	26.996803	34.963196	8.040000
3.00	22.312859	36.622826	11.064317

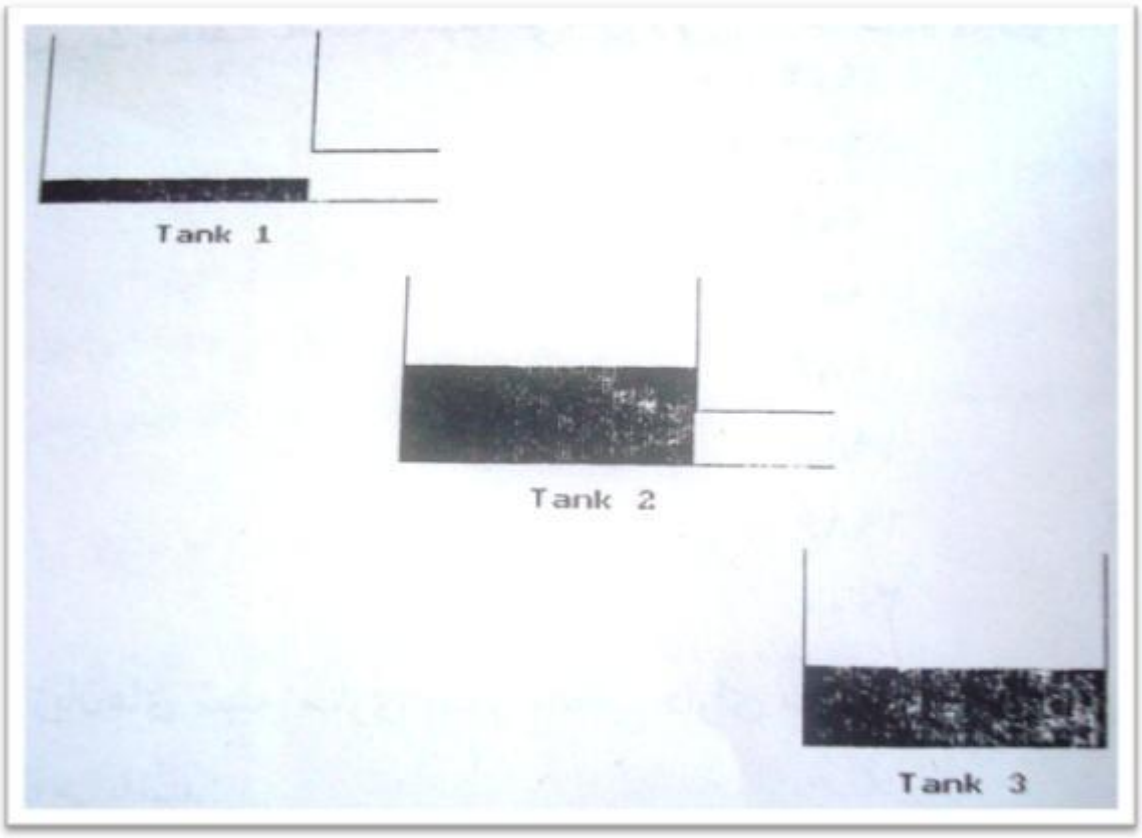
		37.326233	14.232191
3.50	18.441578	37.297127	17.460911
4.00	15.241964	36.715405	20.687113
4.50	12.597484	35.725185	23.862995
5.00	10.411820	34.441406	26.953224
5.50	8.605370	32.955257	29.932405
6.00	7.112338	31.338619	32.783035
6.50	5.878347	29.647722	35.493824
7.00	4.858454	27.926136	38.058353
7.50	4.015512	26.207216	40.473965
8.00	3.318821	24.516108	42.740891
8.50	2.743006	22.871376	44.861534
9.00	2.267094	21.286343	46.839909
9.50	1.873753	19.770170	48.681179
10.00	1.548657	18.328743	50.391300
10.50	1.279965	16.965380	51.976738
11.00	1.057891	15.681418	53.444244
11.50	0.874347	14.476675	54.800686
12.00	0.722648	13.349822	56.052917
12.50	0.597268	12.298689	57.207676
13.00	0.493642	11.320499	58.271511
13.50	0.407995	10.412064	59.250732
14.00	0.337208	9.569925	60.151375
14.50	0.278703	8.790482	60.979172
15.00	0.230348	8.070070	61.739548
15.50	0.190382	7.405041	62.437611
16.00	0.157351	6.791805	63.078148
16.50	0.130051	6.226878	63.665638
17.00	0.107487	5.706902	64.204262
17.50	0.088838	5.228668	64.697906
18.00	0.073424	4.789127	65.150185
18.50	0.060685	4.385397	65.564445
19.00	0.050156	4.014762	65.943779
19.50	0.041454	3.674678	66.291054
20.00	0.034262		

شکل ۶- نمودار سیستم تانک های آب مرتبط به همدیگر

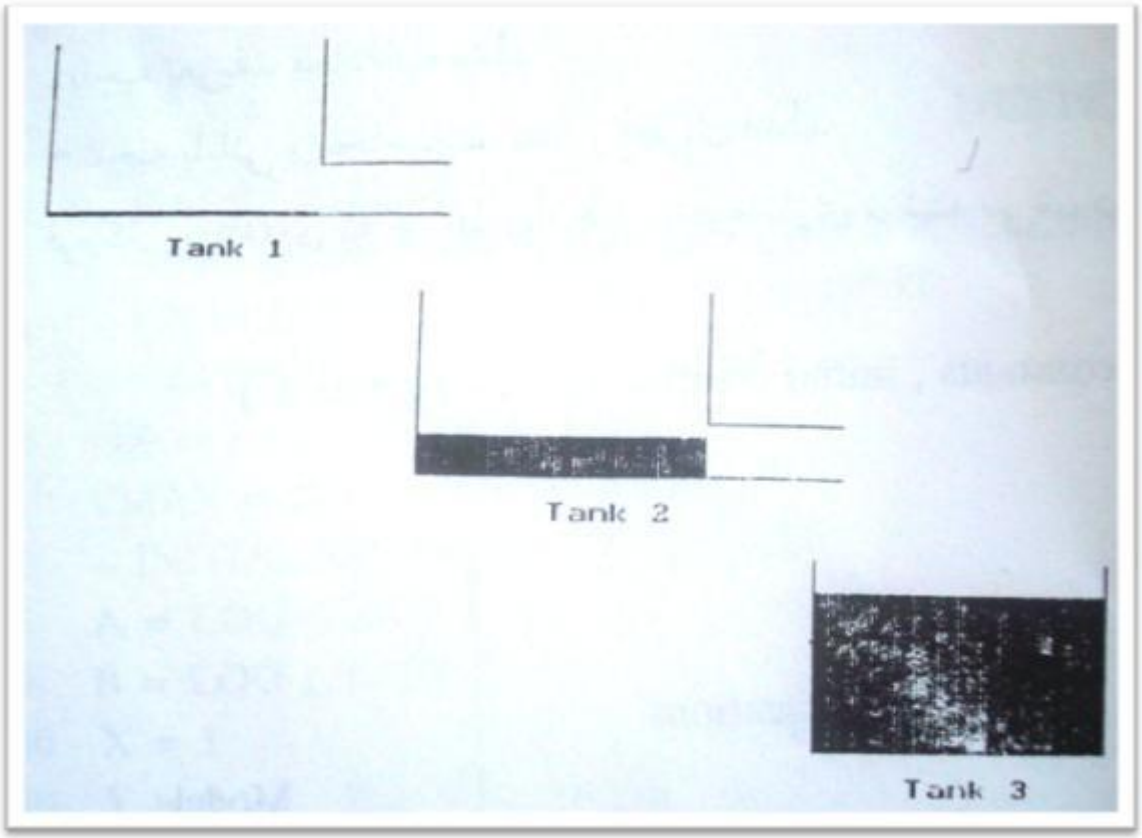

liquid height
Tank 1
Tank 2
Tank 3
Maximum water of tank 2 = 37
time


Tank 1
Tank 2
Tank 3

شکل ۷



شکل ۸



شکل ۹

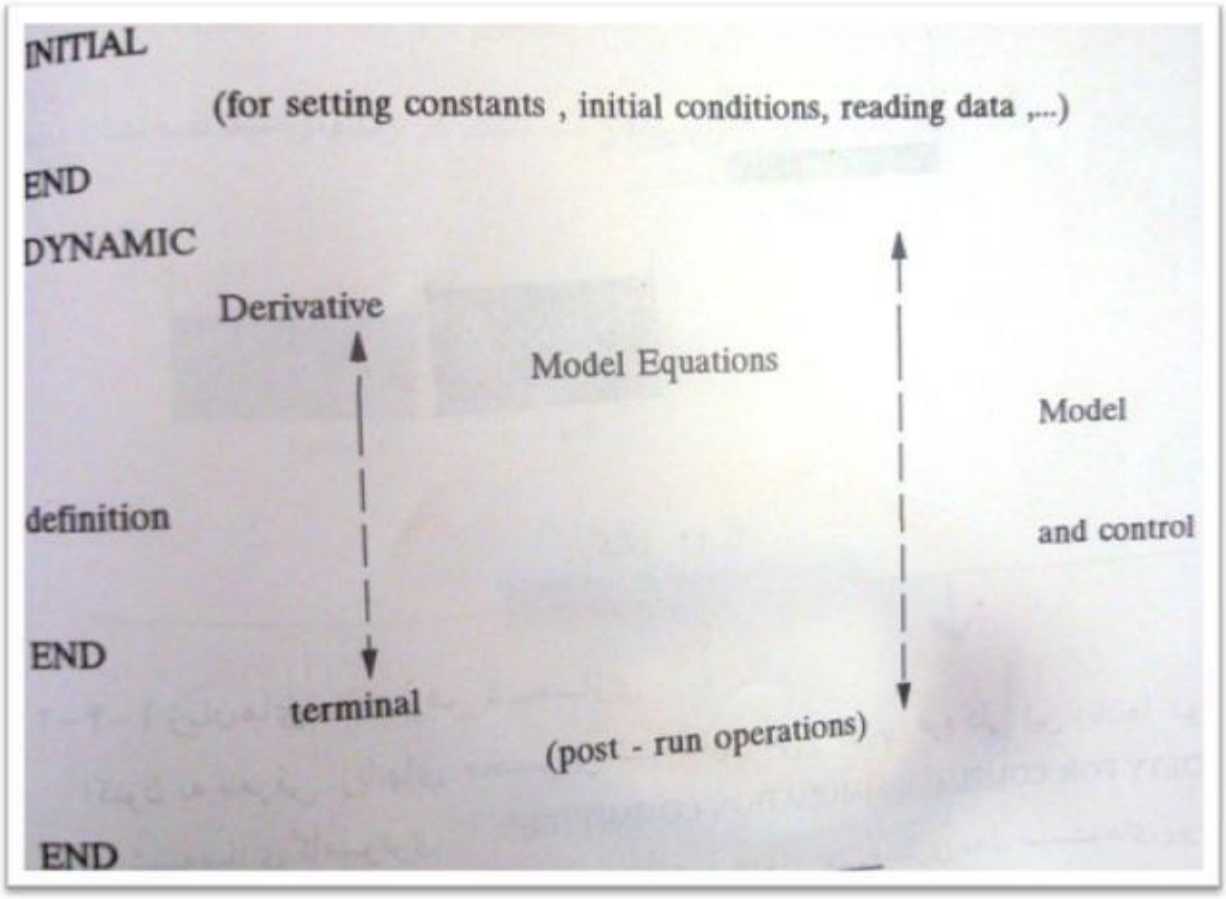
زبان های مخصوص شبیه سازی

اکنون به معرفی زبان های مخصوص شبیه سازی می پردازیم. فرم کلی این زبان ها توسط کمیته شبیه سازی کامپیوتری:"SOCIETY FOR COMPUTER SIMULATION COMMITTEE" در سال ۱۹۶۷ طراحی گردیده است. زبان های شبیه سازی مدل سیستم های پیوسته بصورت مخفف CSSL (Computer System Simulation Language)نامیده می شود و تعدادی از آنها عبارتند از:

MIMIC	1972
ACSL	1972
DESIRE	2000

زبان های شبیه سازی بطور منطقی دارای سه ناحیه برای تعریف مدل و آزمایش برروی مدل می باشند:

- ناحیه تعیین مقادیر اولیه (ثابت ها، متغیرها و ...).
- ناحیه تعریف معادلات مدل.
- ناحیه پایانی و محاسبات بعد از اجرای مدل.



فرم کلی برنامه ای که توسط شبیه سازی نوشته می شود بصورت زیر می باشد:

آشنایی با نرم افزار شبیه سازی DESIRE

نرم افزار DESIRE

نسخه جدید DESIRE/2000 برروی کامپیوترهای شخصی تحت ویندوزهای ۹۵،۹۸، ۲۰۰۰ و NT نصب می گردد، همچنین نسخه ای از این نرم افزار برروی کامپیوترهای Sun SPARC تحت سیستم عامل Solaris 2.x. نصب می شود.

نسخه DESIRE/NEUNET بخشی از نرم افزار DESIRE/2000 می باشد که مدل های حاوی شبکه های عصبی و منطقی فازی را حل می نماید.

نسخه قدیمی DESIRE تحت DOS می باشد و برای ورود به آن دستورات زیر را تایپ کنید، فرض بر این است که قبلاً کلیه فایل های DESIRE در دایرکتوری با این نام نصب گردیده است.

- cd desire
- djsc 0

برنامه های همراه نرم افزار DESIRE

برای اجرای برنامه های از قبل ذخیره شده نسخه جدید DESIRE/2000 تحت ویندوز به طریق زیر عمل نمایید.

برنامه های از قبل ذخیره شده در پنجره Samples می باشند، اشاره گر ویندوز برروی آن برده و یا دوبار کلیک دکمه سمت چپ ماوس و یا کلیک دکمه سمت چپ و کشیدن نام برنامه، آن را در داخل DESIRE Editor قرار دهید. سپس به ترتیب دکمه های OK و ADD را در این پنجره فشار دهید و در پنجره DESIRE- DESIRE دکمه سمت چپ ماوس را کلیک نمایید تا برنامه اجرا گردد. در صورتیکه از نسخه قدیمی DESIRE تحت DOS استفاده می کنید، برای اجرای برنامه های از قبل ذخیره شده دستور زیر را تایپ کنید:

- oldاسم فایل

لیست تعدادی از مدل های از قبل ذخیره شده همراه DESIRE در جدول ۱ آمده است.

دستورهای مقدماتی برای نوشتن و اجرای برنامه

جدول ۱-۳-: لیست تعدادی از برنامه‌های موجود همراه نرم‌افزار DESIRE

نام فایل	برنامه
ZLINE	شبیه‌سازی یک مدار الکتریکی (۱۰۰ معادله دیفرانسیل)
VDP	شبیه‌سازی مدل ون - دِر - پُل
PENDU...	شبیه‌سازی حرکت یک آونگ
CANNON	شبیه‌سازی مسیر یک جسم پرتاب شونده
ORBIT	شبیه‌سازی یک قمر مصنوعی
WORLD	شبیه‌سازی جهان
EPIDEM	شبیه‌سازی یک بیماری واگیردار
BOUNCER	شبیه‌سازی پرتاب یک توپ
DELAY	شبیه‌سازی تأخیر در یک مدار
PHYSB...	شبیه‌سازی مدل جریان خون در بدن
GLUC	شبیه‌سازی تغییرات گلوکز در خون
PILOT X	شبیه‌سازی مطالعه پرتاب خلبان از یک هواپیمای شکاری
HOWE	شبیه‌سازی کنترل ارتفاع یک قمر مصنوعی

برای نوشتن برنامه به زبان DESIRE ابتدا سه ناحیه زیر را در نظر می گیریم.

- ناحیه تعیین مقادیر اولیه (ثابت ها، متغیرها و ...).
- ناحیه تعریف معادلات مدل.
- ناحیه پایانی و محاسبات بعد از اجرای مدل.

با بکارگیری ناحیه ای فوق، ابتدا مقادیر ثابت ها، متغیر ها، مدت زمان شبیه سازی، نوع روش انتگرال گیری، دستورالعمل های مربوط به تعداد اجرای شبیه سازی تعریف می شود، سپس در ناحیه DYNAMIC معادلات مدل و سرانجام در ناحیه پایانی دستورالعمل های مربوط به نوع خروجی شبیه سازی مشخص می شود.

در ادامه این بخش کلیدهای تابعی و دستورات اجرایی نسخه قدیمی DESIRE تحت DOS آمده است که در موقع اجرای DESIRE/2000 تحت ویندوز نیازی به آنها نمی باشد و از طریق گزینه های منوهای پنجره DESIRE-Editor کلیه راهنمایی های لازم را برای تایپ و ویرایش برنامه ها صورت می پذیرد.

برا ی مشاهده کردن لیست برنامه از کلید F7 یا دستورهایlist ، + list استفاده نمایید. برای اجرای برنامه از کلید F1 یا دستور run استفاده نمایید. برا ی متئقف کردن اجرای برنامه از کلید space bar استفاده شود.

برای وارد کردن یک خط از دستورات بصورت ترکیبی، از علامت | برای جدا سازی دستورات استفاده نمایید برا ی مثال ذکر شده در بالا داریم :

oldاسم برنامهrun | + list |

در حین استفاده از دستورات DESIRE می توان از عملوندهای DOS استفاده کرد بدون آنکه از محیط DESIRE خارج شویم. برای این منظور علامت @ و یک ستون فضای خالی را همراه با دستور DOS بکار می بریم، برا ی مثال:

@ copy b:canon.prc c:\project.prc

برای بازگشتن به محیط DESIRE و خروج از DOS دستور Exit را تایپ کنید.برای تایپ یک برنامه جدید از دستور new , filename > استفاده نمایید.

برای تایپ برنامه جدید می توانید علامت / را در شروع سطر تایپ نموده و کلید Enter را فشار دهید . زبان DESIRE بطور اتوماتیک اولین سطر برنامه را با شماره ۲۱۰ شروع می کند و هر بار برای هر سطر جدید ۱۰ شماره به آن اضافه می کند. برنامه زبان DESIRE همانند برنامه زبان بیسیک سطر به سطر تفسیر می شود و اگر سطری خطا داشته باشد به سطر جدید نمی رود و پیغام خطا بررو ی صفحه نمایش ظاهر می شود.

اگر بعد از وارد نمودن برنامه مجدداً علامت / را در شروع سطر جدید تایپ نموده و کلید Enter را فشار دهید ، از حالت شماره سطر اتوماتیکٍ خارج می شوید. برای حذف سطر ، کافی است تنها شماره سطر را تایپ کنید و کلید Enter را فشار دهید. اگر بخواهید سطر جدید داشته باشید شماره سطر و دستورات مورد نظر را تایپ کنید، یعنی همانند زبان برنامه نویسی بیسیک عمل نمایید یا از دستور erase برای حذف سطرها استفاده نمایید.

برای تصحیح برنامه می توانید از دستور ed یا کلید F10 (نیاز به ENTER نمی باشد) استفاده کنید . این دستور برای رفتن به ویرایشگر خطی می باشد. هر تعداد سطری می توانید ویرایشگر خطی تصحیح نمایید، ولی بعد از ویرایش هر سطر بایستی از کلید F10 یا ENTER استفاده نمایید. وقتی که کار ویرایش برنامه تمام شد با دستور Ctrl -Z به محیط DESIRE برگردانده می شوید.

همچنین برای تصحیح برنامه می توانید از دستور edit یا کلید F9 (نیاز به ENTERنمی باشد) استفاده نمایید. این دستورها برای رفتن به ویرایشگر تمام صفحه می باشد. وقتی که کار ویرایش برنامه تمام شد با دستور Ctrl-z به محیط DESIRE برگردانده می شوید در موقع برگشت از ویرایشگر های فوق، یک کپی از آخرین نسخه تصحبح شده برنامه بصورت مد اسکی با نام SYSPIC.LST برروی شاخه جاری دیسک قرار می گیرد. این فایل به صورت فایل پشتیبان عمل می نماید.

برای ذخیره کردن برنامه از کلید F5 یا دستور save به فرم زیر استفاده نمایید.

برا ی مثال:

<save filename.prc

برا ی خارج شدن از نرم افزار DESIRE، از دستور bye یا دستور Ctrl- c استفاده نمایید. در این حالت برنامه جاری در فایل SYSPIC.LST در شاخه جاری دیسک ذخیر ه می شود. سپس می توانید با بارکردن فایل SYSPIC.PRC به این برنامه دسترسی پیدا کنید . در صورتیکه با دستور Ctrl-c از نرم افزار DESIRE خارج شوید، برنامه ذخیره نمی شود .

خلاصه دستورات و کلیدهای تابعی :

دستور	کلید تابعی
run	F1
help	SHIFT + F1
reset	F2
time	SHIFT + F2
drunr	F3
pic	SHIFT + F3
drun	F4
note	SHIFT + F4
save	F5
size	SHIFT + F5
old	F6
dump	SHIFT + F6
list	F7
list +	SHIFT + F7
go	F8
traec	SHIFT + F8
edit	F9
edit	SHIFT + F9
ed	F10
ed	SHIFT + F10

دستورهای برنامه نویسی

حال اجازه بفرمایید در این بخش بطور اختصار تعدادی از دستورالعمل های زبان شبیه سازی DESIRE را مورد بررسی قرا ر دهیم. با توجه به اینکه دستور العمل های این زیان همانند زبانهای سطح بالای بیسیک و پاسکال می باشد و فرض بر این است که خواننده با حداقل یکی از زبانهای فوق آشنایی داشته باشد لذا از جزئیات ساختار دستورالعمل ها صرفنظر گردیده است .

دستور write:

برای نمایش دادن نتیجه یک متغیر، عبارت محاسباتی یا یک رشته برروی صفحه نمایش از دستور write استفاده کنید.

مثال:

```
write x
write ‘the elevator deflection is small’
```

همچنین می توانید در دستور write چندین عنصر را روی یک سطر داشته باشید مثال:

```
write ‘x=’; x, ‘y=’ y; ‘z=’; z
```

علامت ; در عبارت فوق باعث می گردد، عنصر بعدی بلافاصله بعد از عنصر قبلی نوشته شود، در حالیکه علامت، همانند کلید tap تعداد ۸ کاراکتر فاصله ایجاد می کند.

برای نمایش دادن مقدار یک متغیر، عبارت محاسباتی یا رشته در درون یک فایل داده یا چاپگر ، مثل زبان نویسی بیسیک از دستورهای connect یا disconnect استفاده کنید.

```
connect ‘MOD.DAT’ as output 3
connect ‘lpt1’ as output 4
write #4, ‘x has value’;c
disconnect 3, 4
```

دستور if در زبان DESIRE:

دستور if در برنامه اجازه تصمیم گیری می دهد،در صورتیکه شرط صادق باشد عبارتهای بین then و else اجرا می شود، در غیر اینصورت برنامه با دستور های بعد از else ادامه پیدا می کند. به مثالهایی زیر توجه نمایید:

```
if a<0 then x=x+1 | else proceed
if a<6 then x=x+1 | y=y+1
else x=x-1 | write ‘wee’
proceed
```

در زبان DESIRE هر if دارای یک else و یک Proceed می باشد.

در زبان شبیه ساز ی DESIRE مانند زبان های سطح بالا می توان از ifهای تو در تو استفا ده کرد، به مثال زیر توجه نمایید:

```
a=10 | b=1
if a=10 then if b=0 then write ‘a=10,b=0’
else write ‘a=10,b < > 0’ | proceed
else if b=0 then write ‘a < > 10,b=0’
else write ‘ a < > 10, b < > 0’
proceed
proceed
write ‘done’
```

در زبان DESIRE دستور go to، موجب انتخاب بدون شرط می شود مثال:

```
Go to 230
```

نظیردستور go to دستور lable را داریم یک lable در اولین جمله یک سطر قرار می گیرد و نام آن هم به دنبال lable در همان سطر مثال:

```
i =10
label looper
write i | i=i+4
if i<50 then go to looper
else proceed
write ‘done’
```

دستور های for, while, repeat برای ایجاد حلقه :

برای تکرار اجراهای متعدد شبیه سازی یا عملیات محاسباتی از حلقه استفاده می شود. زبان شبیه سازی DESIRE از دستورهای فوق که ساختار آنهه همانند زبان های برنامه نویسی بیسیک و پاسکل می باشد استفاده می نماید. به مثالهای زیر توجه نمایید:

```
for x = 0 to PI /2 step PI / 20
write ‘x’; x, ’sin(x) =’; sin(x)
next
```

```
i=1
while i<= 100
write I, i*i, 1/i
i  = i +1
end while
```

```
i = 0
repeat | write i, i*i,1/i | i = i+10 | until i>= 100
```

حلقه ها می توانند تو در تو باشند، به مثال زیر توجه نمایید:

```
for i=1 to 100
k = 1000
        repeat
        write i + k
        k=k+1000
        until k>5000
write i
next
```

متغیرهای زیر نویس دار و آرایه ها:

در زبان شبیه سازی DESIRE می توان از متغیرهای زیر نویس دار همانند زبان های سطح بال مثل بیسیک استفاده نمود. ابتدا بایستی هر آرایه با عبارت’ dimension‘ اعلان شود. به مثالهایی زیر توجه شود.

```
dimension a[5]
dimension A[3,3], vectr1[100], bb[3,4,2]
```

```
N=100
dimension vectr1[2  * N]
For i = 1 to N
        vectr1[i] = i
```

```
        vectr1 [i + N] = cos(i)
next
```

دستورهای data, read:

دستور data مقادیر برای دستور read ایجاد می کند بنحوی که یک دستور read به اولین جمله از اولین data در لیست اشاره می کند و برای جستجو data بعدی، data قبلی از لیست خارج می شود
دستور read به نحو صحیح بصورت ستونی عنصرهای یک آرایه را می خواند.

دستورهای برنامه نویسی برای اجرای شبیه سازی

دستور drun:

دستور drun برای اجراء برنامه می باشد، کامپایلر DESIREبه دستور drun که رسید بطور اتوماتیک به ناحیه DYNAMIC می رود و این قسمت را برای زمان t0 تا TMAX اجراء می نماید و اگر مقدار t0 را مشخص نکرده باشید، بطور پیش فرض t0=0 گرفته می شود.

روش های انتگرال گیری:

دستور irule n برای تعیین یک نوع روش انتگرال گیری است.
اگر در برنامه از این دستور استفاده نشود، DESIRE بطور پیش فرض irule 1 را در نظر می گیرد که همان روش انتگرال گیری رونگ – کوتا مرتبهٔ دوم می باشد.
روش های انتگرال گیری DESIRE مطابق جدول ۳ است که با تغییر مقدار n بین ۱ تا ۸ یکی از آنها انتخاب می شود.
اگر irule را بدون آرگومان تایپ کنید، DESIRE روش انتگرال گیری جاری را می دهد.

جدول۳- روش های انتگرال گیری Desire

ناحیه

irule 1	رونگ – کوتا– هیون، مرتبه دوم (روش پیش فرض)
irule 2	اویلر، مرتبه اول
irule 3	رونگ – کوتا، مرتبه چهارم
irule 4	رونگ – کوتا، مرتبه چهارم، طول گام متغیر
irule 7	رونگ – کوتا، مرتبه دوم، طول گام متغیر
irule 8	رونگ – کوتا– نیس، طول گام متغیر

DYNAMIC:

در این ناحیه معادلات مدل تعریف می شوند که این معادلات می توانند معادلات جبری یا معادلات دیفرانسیل معمولی مرتبه اول باشند و اگر معادله دیفرانسیلی دارای مرتبه n باشد به n معادلهٔ مرتبه اول تبدیل شود، مثال:

```
d/dt  x = xdot
d/dt xdot = -ww * x – r * xdot + A * sin(b * t)
```

متغیر و پارامترهایی که در معادلات مدل فوق مورد استفاده قرار گرفتهاند بایستی قبلاً در ناحیه DYNAMIC یا در ناحیه مقادیر اول قبل از دستور drun تعیین مقدار شده باشند.

دستور display:

اگر از display 1 استفاده شود، هر اجرا در یک صفحه نمایش داده می شود.

اگر از display 2 استفاده شود، تمام اجراها در یک صفحه نمایش داده می شود.

نمایش خروجهای برنامه:

۱ - برای نمایش گرافیکی حداکثر چهار متغیر در مقابل t برای محور افقی از دستور العمل زیر استفاده شود.

```
dispt  x
```

یا:

```
disp x1, x2, …
```

۲ - برای نمایش گرافیکی حداکثر ۳ متغیر در مقابل اولین متغیر برای محور افقی از دستورالعمل زیر استفاده شود.

```
dispxy x, y
```

یا:

```
dispxy  x, y1, y2, ..
```

۳ - برای نمایش عددی t (زمان) و حداکثر چهار متغیر از دستورالعمل زیر استفاده شود.

```
type x
```

یا:

```
type x1, x2, …
```

متغیر TMAX:

کلمه ذخیره شده است که برای نشان دادن حداکثر زمان شبیه سازی استفاده می شود.

متغیر DT:

کلمه ذخیره شده است که برا ی مشخص کردن طول گام استفاده می شود.

```
DT = TMAX/(NN-1)
```

هر چه NN را بزرگتر انتخاب کنید طول گام کوچک تر و در نتیجه تعداد نقاط گراف بیشتر خواهد شد. پیش فرض برای NN، عدد ۲۵۱است.

متغیر Scale:

در موقع رسم گراف مقادیر متغیرهای وابسته سیستم بایستی در فاصله (- scale, scale) باشند.

در صورتیکه مقادیر وابسته در فاصله (-scale,scale) نباشند، خطا پیش می آیدکه بایستی با تغییر مقیاس نسبت به سرریز شدن مقادیر متغیرهای وابسته از صفحه نمایش جلوگیری نمود. با قرار دادن زیر روال () procedure scrun در اول برنامه، بطور اتوماتیک scale مناسب ایجاد خواهد شد و مانع از بروز پیام خطا overload scale می شود. زیر روال فوق بصورت زیر می باشد.

```
procedure scrun()
XCCC = 1
drunr
while  XCCC < 0
XCCC = -XCCC
scale = 3 * scale
drunr
end while
end
```

در زیر روال فوق XCCC یک متغیر سیستم است که در موقع سرریز شدن مقادیر متغیرها منفی می شود و باعث می گردد مقدار متغیر scale تغییر یابد.

دستورTerm:

این دستور باعث می گردد که اجرای شبیه سازی قبل از شرط t=IMAX پایان پذیرد. فرم کلی این دستور بصورت زیر است:

```
term expression
```

در دستور فوق وقتی مقدار عبارت بیشتر از صفر شود، شبیه سازی پایان می پذیرد.

دستور OUT:

این دستور در ناحیه پایانی DYNAMIC قرار می گیرد و هر جمله ائی که بعد از این دستور نوشته شود تنها در نقاط مکاتباتی [1] اجرا می شود .

دستور Stash:

این دستور در ناحیه DYNAMIC قرار می گیرد و می توان با بکارگیری این دستور مقادیر متغیرها را در یک فایل اسکی نگهداری نمود. برای مثال:

```
stash x1, x2, …
```

مقادیر x1,x2,… را درمقادیر مکاتباتی در یک فایل با پسوند T1M. ذخیره می سازد.

دستور recover:

این دستور در ناحیه DYNAMIC قرار می گیرد و می توان برای بازیابی مقادیر متغیرهائی که توسط دستور stash در یک فایل با پسوند T1M. قرار گرفته اند، استفاده نمود. به مثال زیر توجه نمایید.

[1]) communication points

```
connect ‘MYPROGOO.T1M’ as input 9
t = 0.0 | TMAX = 20 | NN = 100
drun
DYNAMIC
recover x1, x2, …
dispt x2, x3,  یا type x1, x2, x4
```

دستور چاپ برنامه برروی چاپگر :

برای چاپ لیست برنامه DESIRE برروی چاپگر، دستور زیر را تایپ کنید:

```
list + ‘LPT1’
```

و سپس کلید prtsc را فشار دهید.

دستور چاپ خروجی بصورت گراف برروی چاپگر :

در صورتی که خروجی به شکل گراف باشد و بخواهید آن را برروی چاپگر بیاورید، ابتدا قبل از ورود به DESIREدر محیط DOS دستور GRAPHICS را تایپ نمائید . این دستور را می توانید در محیط DESIRE با تایپ GRAPHICS @ < انجام دهید و سپس هنگامیکه گراف برروی صفحه نمایش می باشد کلید prtsc را فشار دهید. توجه داشته باشید که چاپگر روشن باشد.

توقف اجرای برنامه :

با فشار دادن کلید Space bar می توان اجرای برنامه را متوقف ساخت.

تکنیک های پیشرفته در برنامه نویسی

در این بخش تعدادی از تکنیک های پیشرفته زبان DESIRE توضیح داده می شود . این تکنیک ها در مثال های کاربردی مورد استفاده قرار می گیرند.

اعلان و فراخوانی توابع و زیر روالها:

در زبان شبیه سازی DESIRE هم مانند دیگر زبانهای سطح بالا می توان از توابع و زیر روالها استفاده نمود. توابع و زیر روالها قبل از ناحیه DYNAMIC تعریف و صدا زده می شوند. به مثالهای زیر در مورد اعلان توابع و فراخوانی آنها توجه نمایید.

```
FUNCTION        cot (xx)  cos(xx) / sin (xx)

FUNCTION        min (aa, bb) = aa – lim (aa – bb)
FUNCTION        sqr (pp) = pp * pp
FUNCTION        vabs (uu, vv) = sqr (sqr(uu) + sqr(vv))
```

فراخوانی توابع :

```
val7 = cot(n) + cot(11.3-y) +  vabs(r,s)/r
MAX = max(a,b-15)
```

فرم کلی زیر روالها به شکل زیر می باشد.

```
PROCEDURE procname (x,y,…;X, Y, …)
```

در دستور فوق علامت ; بیانگر این است که پارامترهای x,y,… که قبل از این علامت آمده اند با مقدار [۲] نامیده می شوند. همچنین پارامترهای X,Y,… که بعد از این علامت آمده اند با نام [۳] (در زبان پاسکال قبل از آنها دستور VAR قرارمی گیرد) می باشند. به مثال زیر توجه نمایید.

```
PROCEDURE        polrec(r, theta; x, y)
        x = r * cos(theta)
        y = r * sin(theta)
end
```

زیر روالها می توانند تنها پارامترهای با مقدار یا تنها پارامترهای با نام و یا اصلاً پارامتری نداشته باشند به مثالهای زیر توجه نمایید.

```
PROCEDURE        Proc1(p, q)
PROCEDURE Proce2(;g, h, i)
PROCEDURE Proces3()
```

در حالتی که اصلاً پارامتری وجود ندارد، نمی توان بعد از نام زیر روال پرانتز های باز و بسته را حذف نمود.

فرخوانی زیر روالها:

```
call polrec(r1, theta1 ; x1, y1)
```

یا

```
call polrec(a + 12.34, 4 * PI/180 ; x2, y2)
```

اعلان و فراخوانی زیر مدلها:

زیر مدلها حاوی بخش های مختلفی از معادلات مدل یک سیستم می باشند و در ناحیه DYNAMIC مورد استفاده قرا رمی گیرند. اعلان یک زیر مدل توسط کلمه MACRO و در ناحیه اولیه قبل از دستور drun می باشد. به مثال زیر توجه نمایید.

```
MACRO  servo (x, xdot ; gain, dampg, input, error)
        Error = x – input
        d/dt x = xdot
        d/dt xdot = -gain * error – dampg * xdot
end
```

بدنه MACRO یک زیر مدل را تعریف می کند و کدهائی را می توان در داخل آن نوشت که در ناحیه DYNAMIC قابل قبول باشد.

بعد از نام MACRO در داخل پرانتز قبل از علامت ; متغیر های حالت [۴] قرا ر می گیرند. تمام آرگومانها چه قبل از علامت ; و چه بعد از آن پارامترهای با نام می باشند (مثل آرگومانهائی که در زبان پاسکال قبل از آنها کلمه VAR قرار گرفته باشد).

[2] -by value
[3] - by name

در مثال های زیر فرم های مختلفی از اعلان MACRO (با متغیرهای حالت ، بدون متغیرهای حالت یا بدون آرگومان) آمده است.

```
MACRO  mac1(p,q)
MACRO  mac2(;g, h, i)
MACRO  mac3()
```

یک زیر مدل را در هرجائی از ناحیه DYNAMIC می توان بصورت زیر فراخوانی نمود.

```
invoke  macroname(a, b, …; A, B, …)
```

جائیکه b,a ,… متغیرهای واقعی حالت و A,B,… متغیر های واقعی مدل سیستم می باشند.

قسمت های چندگانه در ناحیه DYNAMIC

ناحیه DYNAMIC می تواند حاوی چند قسمت مختلف باشد که یکی بعد از دیگری اجرا می شوند. بغیر از اولین قسمت ، هر قسمت با کلمه label شروع و نام آن قسمت بلافاصله بعد از آن قرا ر می گیرد. سپس این نام در جلو دستور drunیا drunr مربوطه قرا رمی گیرد. به مثال زیر توجه نمایید:

```
k = 100  |  r = 4                                   برای قسمت اول
c = 0.99
-----------------------------------------------------------
K = 1  |  R = 0.05                                  برای قسمت دوم
X = -0.7
-----------------------------------------------------------
TMAX = 2.5  |  DT = 0.01  |  irule 3               برای قسمت اول
drunr
display 2
-----------------------------------------------------------
TMAX = 10  |  DT = 0.05  |  irule 1                برای قسمت دوم
drunr BOO
-----------------------------------------------------------
DYNAMIC
    d/dt x = xdot  |  d/dt  xdot = -k * x - r * xdot
    dispt x
-----------------------------------------------------------
label BOO
    d/dt X = XDOT  |  d/dt  XDOT = -K * X - R * XDOT
    dispxy X , XDOT
```

ترکیب قسمت های یک برنامه در یک فایل

در موقع نوشتن برنامه های بزرگ به زبان DESIRE، می توان قسمت های قابل استفاده مجدداً برنامه [5] را مثل PROCEDURE, FUNCTIONو MACRO را در فایل های مختلفی با پسوند SRC. ذخیره نمود و سپس این قسمت ها را بهمراه یک برنامه DESIRE مثلاً بنام Prog در یک برنامه جدید با نام دلخواه COMBO.PRC ترکیب و ذخیره نمود . به مثال زیر توجه نمایید:

```
new
load  ‘seg1’
load  ‘seg2’
```

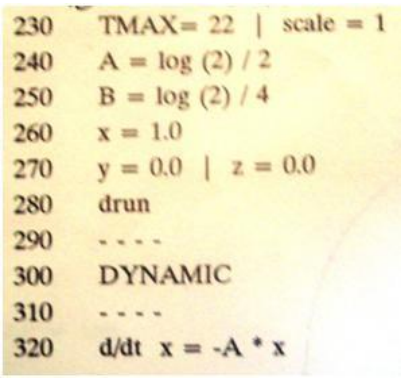
[4] - State Variables
[5] - reusable program segments

.
.
.
load ‘prog’
PIC ‘combo’
save

اکنون در ادامه به برنامه نویسی و اجرای دو مدل نمونه به زبان شبیه سازی DESIRE اکتفا می کنیم.

مدل نمونه – سیستم تانک های مایعات مرتبط به همدیگر :

این مدل در اول بحث توضیح داده شده است . برنامه این مدل در زیر نوشته شده است . هدف از اجرای این برنامه بررسی ارتفاع مایع در سه تانک در زمان های مختلف می باشد.




```
330     d/dt  y = A * x – B * y
340     d/dt  z = B * y
350     ---
360     dispt x, y, z
/--
```

اجازه بفرمایید در ارتباط با این برنامه توضیحاتی داده شود.

۱ -ماکزیمم زمان شبیه سازی در این برنامه ۲۲ واحد زمان است که می توان آن را به هر مقدار دلخواه تغییر داد.

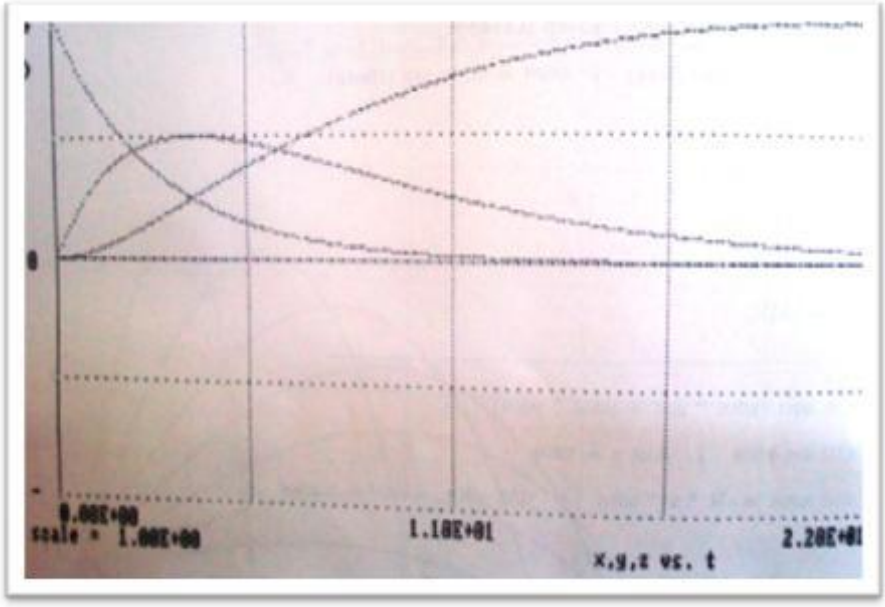
۲ - مقدار مقیاس در این برنامه یک است، در صورتیکه در موقع رسم نمودار پیغام display – scale overloads روی صفحه نمایش ظاهر شود بایستی نسبت به تغییر مقیاس اقدام نمود. یا اینکه زیر روال مربوط به مقیاس اتوماتیک را در ابتدای برنامه قرار داد.

۳ - در ادامه ناحیه اولیه، مقادیر پارامترهای B,A و مقادیر اولیه متغیرهای حالت مدل یعنی z, x, y تعیین شده اند.

۴ -برای اجرای برنامه دستور drun قبل از ناحیه DYNAMIC قرار گرفته است.

۵ - معادلات مدل سیستم در قسمت DYNAMIC تعریف شده است.

۶ -سرانجام نمودار تغییرات مقادیر مایع در سه تانک در زمانهای مختلف در شکل زیر رسم شده است.

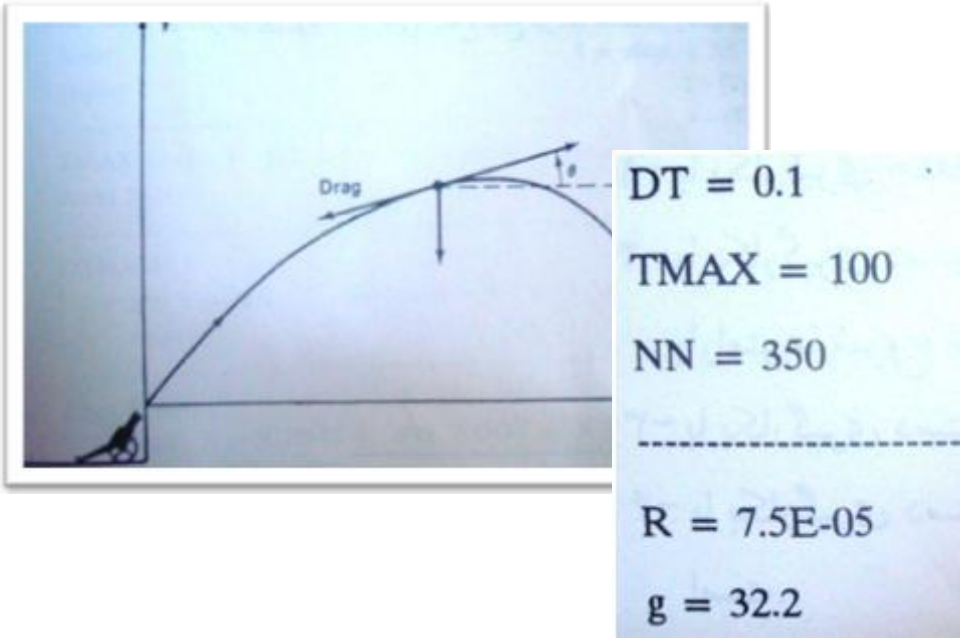


نمودار تغیرات مایع در سه تانک

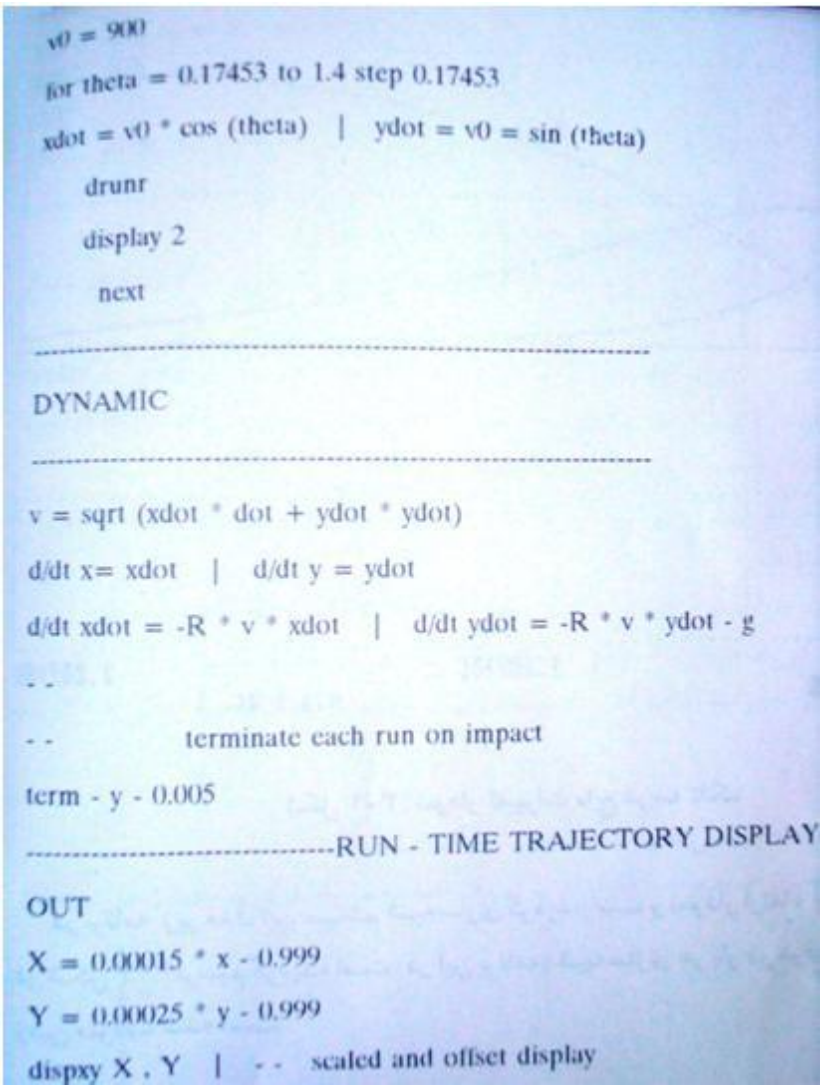
مدل نمودار- بررسی

مسیر گلوله توپ از یک اسلحه :

این مدل نمونه مربوط به پرتاپ از یک اسلحه می باشد که به دفعات با زاویه های مختلف پرتاپ می شود.

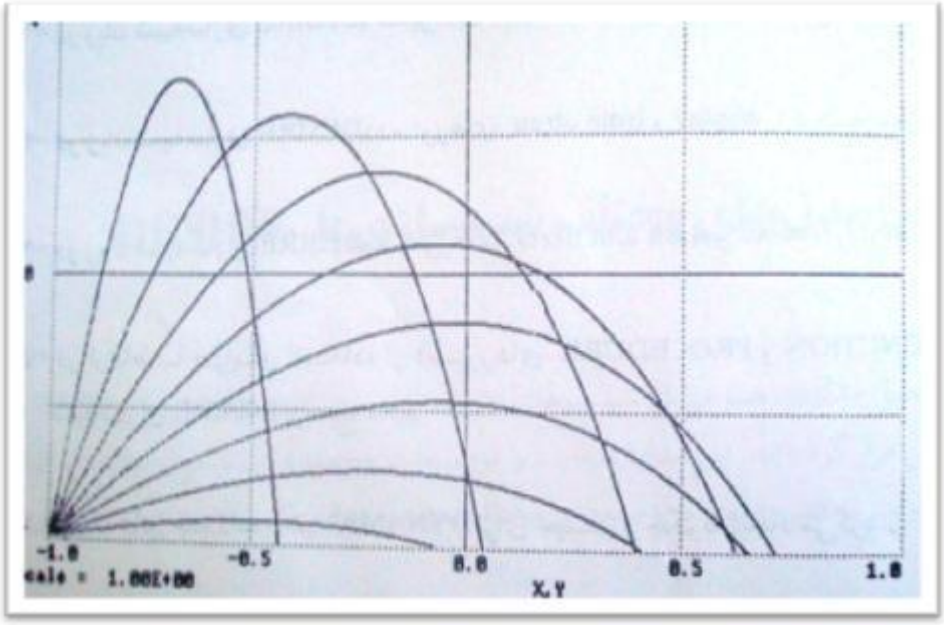


در برنامه زیر مدل این سیستم شبیه سازی گردیده است.



```
v0 = 900
for theta = 0.17453 to 1.4 step 0.17453
xdot = v0 * cos (theta)   |   ydot = v0 = sin (theta)
    drunr
    display 2
     next
-----------------------------------------------------------
DYNAMIC
-----------------------------------------------------------
v = sqrt (xdot * dot + ydot * ydot)
d/dt x= xdot   |   d/dt y = ydot
d/dt xdot = -R * v * xdot   |   d/dt ydot = -R * v * ydot - g
- -
- -          terminate each run on impact
term - y - 0.005
----------RUN - TIME TRAJECTORY DISPLAY
OUT
X = 0.00015 * x - 0.999
Y = 0.00025 * y - 0.999
dispxy X , Y   |   - -   scaled and offset display
```

نمودار ارتفاع آن در مقابل برد آن در شکل ۳ ترسیم گردیده است. در این برنامه ، شبیه سازی هر بار در موقع برخورد جسم به زمین متوقف شده است.



شکل ۳- نمودار مسیر یک جسم پرتاپ شونده- چندین اجرا

حل مدل های مختلف شبیه سازی با DESIRE

بررسی مدلهای متنوع شبیه سازی برا ی سیستم های پیوسته:

این مدلها به چند دسته زیر طبقه بندی و برنامه های شبیه سازی برای آنها نوشته شده است :

۱ -مدلهایی با یک اجرای ساده: اصولاً ایجاد برنامه شبیه سازی برای این مدلها کار آسانی است و آزمایش برروی این مدلها مستلزم اجرای یکبار مدل است.

۲ -مدل هایی با چندین اجرا: اصولاً برای این مدلها، آزمایش های مورد نظر مستلزم دنباله ایی از اجراهای شبیه سازی است. برای مثال در رابطه با جسم پرتاپ شونده، مقدار زاویه پرتاپ برای اجراهای مختلف عوض می شود تا یک دنباله از جوابها حاصل گردد.

۳ - مدلهای ناپیوسته : اصولاً نرم افزارهای شبیه سازی دارای توابع کتابخانه ایی غیر خطی جهت بالا بردن قابلیت های نرم افزار برای شبیه سازی مدل های ناپیوسته مربوط به سیستم های پیچیده می باشند. در این بخش با طرح مثالهایی، این قابلیت ها مورد بررسی قرار می گیرد. برای مثال مدل مربوط به کنترل درجه حرارت مایع در یک تانک که یک دستگاه گرم کننده بطور ناگهانی بایستی روشن و خاموش شود تا درجه حرارت مایع را ثابت نگه دارد.

۴ - مدل های نسبتاً بزرگ: در این بخش مدلهای کاربردی نسبتاً بزرگ مورد بررسی قرار می گیرند. برای مثال مدل مربوط به تأخیر ولتاژ وجریان در یک مدار الکتریکی که حاوی تعداد ۱۰۰ معادله دیفرانسیل می باشد.

مدلهایی با یک اجرای ساده:

مدل بررسی رشد جمعیت در سه شهر:

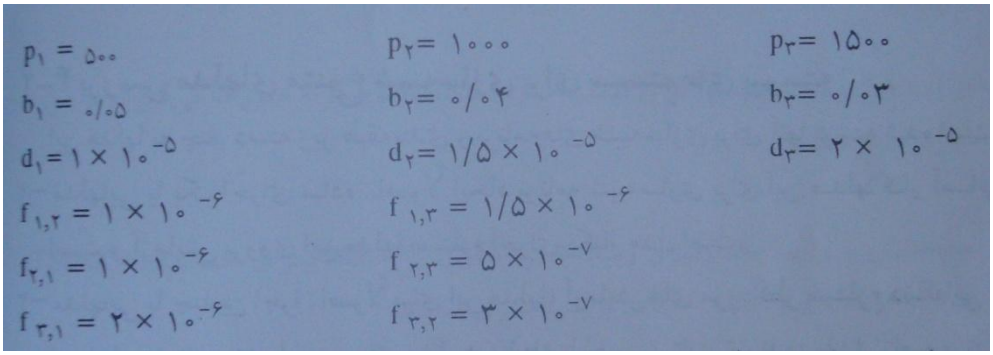
فرض کنید که جمعیت هر شهر با متغیر های P_1 وP_2 وP_3 نشان داده شده است. همچنین مقادیر اولیه جمعیت با p_k ، نرخ توالید با b_k و نرخ مرگ و میر با d_k ، که k برابر با ۳،۲،۱ می باشد . نرخ مرگ و میر براثر جنگ داخلی بصورت زیر می باشد:

برای p1 $f_{1,2} P_1P_2 + f_{1,3}P_1P_3$

برای p2 $f_{2,1} P_2P_1 + f_{2,3} P_2P_3$

برای p3 $f_{3,1}P_3P_1 + f_{3,2} P_3P_2$

رشد جمعیت شهر ها را باتوجه به فرض های زیر برای ۲/۵ و ۵۰۰ سال بعد از شروع ، شبیه سازی نمائید.



برنامه شبیه سازی

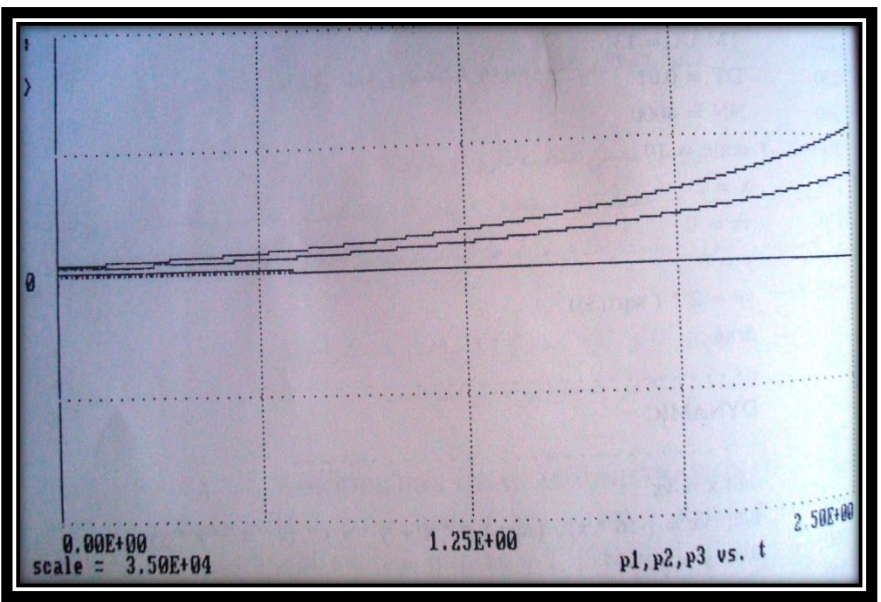
مثال فوق به زبان DESIRE برای ۲/۵ سال و نمودار آن در شکل ۴ نشان داده شده است.

```
THREE POPULATION  ------          210
TMAX = 2.5                        220
DT = 0.001                        230
NN = 5000                         240
                                  250
                                  scale =
                                  35000
                                  p1          260
                                  500 | p2 =
                                  1000 | p3 =
                                  1500 | ----
```

=

```
270  b1 = 0.05  | b2 = 0.04  | b3 = 0.03  | - - - - birth rate - - - - - - - - - -
280  d1 = 10E-5 | d2 = 5 * 10E -5 | d3 = 2*10E-5 | - - natural death rate
290  f12 = 10E - 6 | f13 = 5 * 10E - 6
300  f21 = 10E - 6 | f23 = 5 * 10E - 7
310  f31 = 2 * 10E - 6 | f32 = 3 * 10E - 7
320  drunr
330  DYNAMIC
340  d/dt  p1 =(p1+(b1 *p1)) - ((d1 *p1+(f 12 *p1 *p2) + (f 13 *p1* p3)))
350  d/dt  p2 =(p2+(b2 *p2)) - ((d2 *p2+(f 21 *p2 *p1) + (f 23 *p2* p3)))
360  d/dt  p3 =(p3 + (b3 *p3)) - ((d3 *p3+(f 31 *p3 *p1) + (f 32 *p3* p2)))
370  dispt  p1 , p2 , p3
```

POPULATION INITIAL

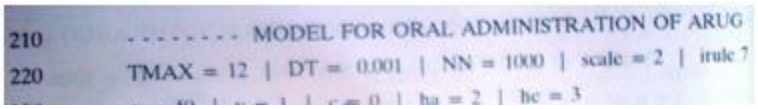


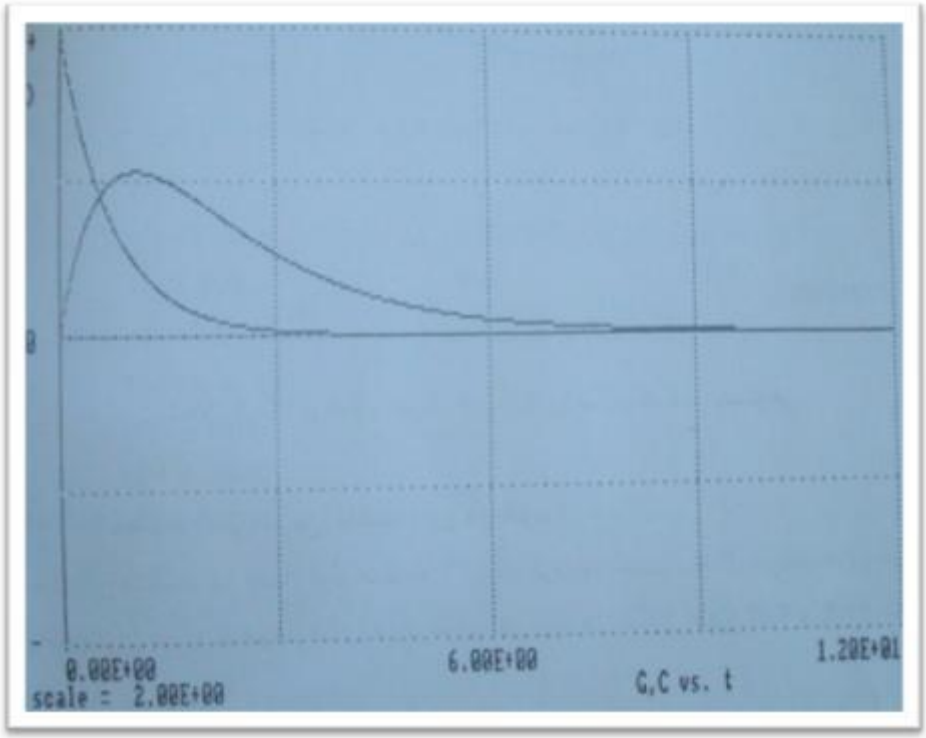
شکل ۴: برنامه و نمودار شبیه سازی رشد جمعیت در سه شهر

مدل بررسی تعیین غلظت دارو در خون

آزمایش ۱: مقدار G و C را بعد از ۱۲ ساعت بدست آورید.

آزمایش ۲: مقدار C در چه موقعی ماکزیمم است.





شکل: برنامه و نمودار شبیه سازی بررسی تعیین غلظت دارو در خون- آزمایش ۱

```
210     - - - - - - - MODEL FOR ORAL ADMINISTRATION OF ARUG (e.1)
220     TMAX = 12 | DT = 0.001 | NN = 1000 | scale = 2 | irule 7
230     g = 40 | v = 1 | c = 0 | ha = 2 | he = 3 | prc = -.05
240     ke = ln (2) / he
250     ka = ln (2) / ha
260     drun
270     write " VALUES OF PEAK : " ; "  t = " ; t ; "   c = " ; c
280     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
290     DYNAMIC
300     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
310     d/dt  g = -ka * g
320     d/dt  c = (ka * g) / v - ke * c
330     G = g * .05  |  C = c * .05
340     term prc - c
350     prc = c
```


```
210     - - - - - - - MODEL FOR ORAL ADMINISTRATION OF ARUG (e.1)
220     TMAX = 12 | DT = 0.001 | NN = 1000 | scale = 2 | irule 7
230     g = 40 | v = 1 | c = 0 | ha = 2 | he = 3 | prc = -.05
240     ke = ln (2) / he
250     ka = ln (2) / ha
```



```
210     - - - - - - - MODEL FOR ORAL ADMINISTRATION OF ARUG (e.1)
220     TMAX = 12 | DT = 0.001 | NN = 1000 | scale = 2 | irule 7
230     g = 40 | v = 1 | c = 0 | ha = 2 | he = 3 | prc = -.05
240     ke = ln (2) / he
250     ka = ln (2) / ha
```



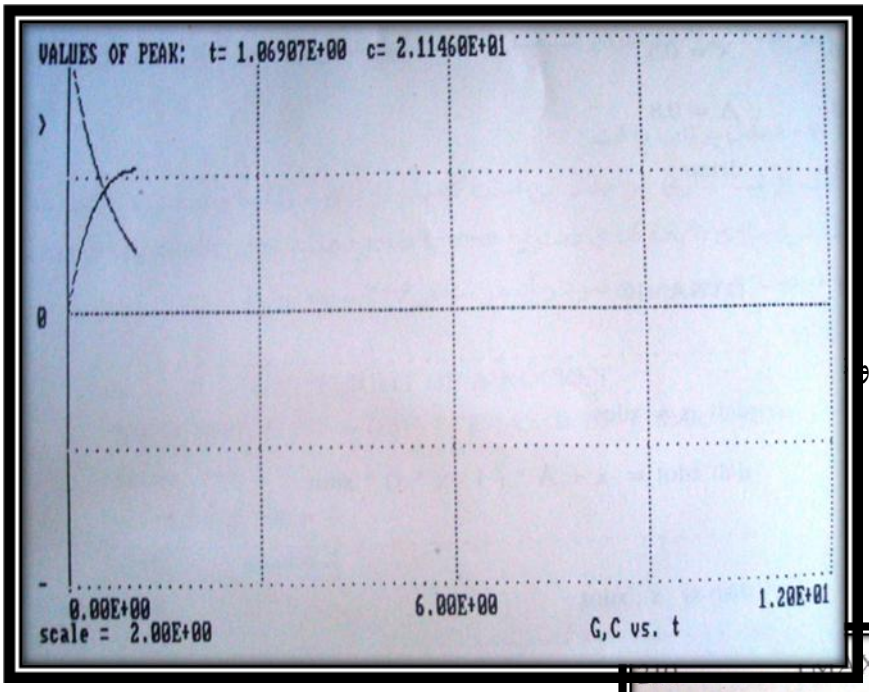
```
210     - - - - - - - MODEL FOR ORAL ADMINISTRATION OF ARUG (e.1)
220     TMAX = 12 | DT = 0.001 | NN = 1000 | scale = 2 | irule 7
230     g = 40 | v = 1 | c = 0 | ha = 2 | he = 3 | prc = -.05
240     ke = ln (2) / he
250     ka = ln (2) / ha
```



```
210     - - - - - - - MODEL FOR ORAL ADMINISTRATION OF ARUG (e.1)
220     TMAX = 12 | DT = 0.001 | NN = 1000 | scale = 2 | irule 7
230     g = 40 | v = 1 | c = 0 | ha = 2 | he = 3 | prc = -.05
240     ke = ln (2) / he
250     ka = ln (2) / ha
```



```
210     - - - - - - - MODEL FOR ORAL ADMINISTRATION OF ARUG (e.1)
220     TMAX = 12 | DT = 0.001 | NN = 1000 | scale = 2 | irule 7
230     g = 40 | v = 1 | c = 0 | ha = 2 | he = 3 | prc = -.05
240     ke = ln (2) / he
250     ka = ln (2) / ha
```



شکل: برنامه و

مدل های بازگشتی:

برنامه اول: فاکتوریل

```
220     t = 1   |   x = 1
230     drun
240     - - - - - - - - - - - -
250     DYNAMIC
260     - - - - - - - - - - - -
270     x = x * t
280     type x
```

برنامه دوم: ایجاد یک دایره

```
210     TMAX = 1    |   NN = 10000
220     h = 0.005
230     x = 1    |    y = 0
240     drun
250     - - - - - - - - - - - -
260     DYNAMIC
270     - - - - - - - - - - - -
280     x = x - y * h    |    y = y + x * h
290     dispxy y , x
```