Übung 6

1. Aufgabe

In einem grossen Betrieb, der spezialisiert ist auf die Reparatur von Elektrogeräten, beschaffen sich die Monteure die benötigten Ersatzteile am Schalter des internen Lagers. Pro Stunde erscheinen durchschnittlich 8 Monteure am Schalter und die mittlere Bedienzeit beträgt 6 Minuten. Beim Schalter befinden sich zwei Sitzplätze zum Warten. Es wird angenommen, dass sowohl die Zwischenankunfts- als auch die Bedienzeiten exponentiell verteilt sind.

a)

Berechnen Sie die folgenden Systemmerkmale:

Ankunfts-8/h und Bedienrate10/h, Auslastungsgrad=0.8

Wahrscheinlichkeit, dass kein () (einer (), zwei ()) Monteur(e) beim Schalter ist   
p0=rho^0\*(1-rho)=0.2  
p1= rho^1\*(1-rho)=0.16  
p2= rho^2\*(1-rho)=0.128

Wahrscheinlichkeit, dass mindestens ein Monteur stehend warten muss (d.h. dass mindestens 4 Personen im System sind)  
1-(p0+p1+p2+p3)=0.4096

mittlere Anzahl Monteure beim Schalter (in Bedienung oder am Warten)  
E[N] = rho/(1-rho)=4

mittlere Aufenthaltszeit beim Schalter (in Bedienung oder am Warten)  
E[W]=E[N]/lambda = 0.5

mittlere Wartezeit beim Schalter   
E[Wq]=E[W]-1/mu=0.4

mittlere Anzahl Monteure am Warten  
E[Nq]=lambda\*E[Wq]=3.2

b)

Der Betriebschef möchte die Wartezeiten reduzieren und fordert, dass sich mit einer Wahrscheinlichkeit von mindestens 95% nicht mehr als 3 Monteure beim Schalter befinden. Um wie viel müsste die mittlere Bedienzeit verkürzt werden, damit diese Forderung erfüllt wird?  
p0+p1+p3>=0.95  
rho^0\*(1-rho)+ rho^1\*(1-rho)+ rho^2\*(1-rho)+ rho^3\*(1-rho)>=0.95

Benützung der Matlab Symbolic Toolbox zur Lösung der Gleichung:

>> syms rho

>> eqn = rho^0\*(1-rho)+ rho^1\*(1-rho)+ rho^2\*(1-rho)+ rho^3\*(1-rho)==0.95

eqn =

1 - rho^2\*(rho - 1) - rho\*(rho - 1) - rho == 19/20

>> solRho = solve(eqn,rho)

solRho =

-20^(3/4)/20

20^(3/4)/20

-(20^(3/4)\*1i)/20

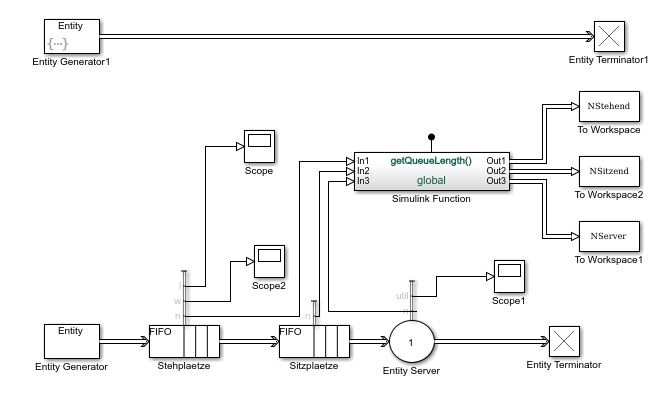
(20^(3/4)\*1i)/20

rho = 0.4729

muMin = lambda/rho = **16.9169**

c)

Überprüfen Sie Ihre Resultate mit einem SimEvents-Modell.



%a)

t = Nq.Time;

QLength = Nq.Data;

NumberInServer = Ns.Data;

NumberInSystem = QLength+NumberInServer;

% Wahrscheinlichkeit, dass kein Monteur beim Schalter ist

p0 = sum((NServer + NSitzend + NStehend) == 0)/length(NServer)

% p0 =

%

% 0.2021

% Wahrscheinlichkeit, dass ein Monteur beim Schalter ist

p1 = sum((NServer + NSitzend + NStehend) == 1)/length(NServer)

% p1 =

%

% 0.1610

% Wahrscheinlichkeit, dass zwei Monteure beim Schalter sind

p2 = sum((NServer + NSitzend + NStehend) == 2)/length(NServer)

% p2 =

%

% 0.1291

% Wahrscheinlichkeit, dass mindestens ein Monteur stehend warten muss

p = sum(NStehend>0)/length(NStehend)

% p =

%

% 0.4059

% mittlere Anzahl Monteure beim Schalter (in Bedienung oder am Warten)

mean(NServer + NSitzend + NStehend)

% ans =

%

% 3.9784

% mittlere Aufenthaltszeit beim Schalter (in Bedienung oder am Warten)

% Siehe Scopes: Mittlere Wartezeit in den Warteschlangen + mittlere

% Wartezeit im Server

% 0.255 + 0.144 + 1/mu = 0.499

% mittlere Wartezeit beim Schalter

% Siehe Scopes: 0.399

% mittlere Anzahl Monteure am Warten

mean(NSitzend + NStehend)

% ans =

%

% 3.1805

%b)

p = sum(NServer + NSitzend + NStehend<=3)/length(NServer)

% p =

%

% 0.5941

% Momentan ist die Wahrscheinlichkeit ca. 59 Prozent.

% Schrittweises Anpassen des Parameters für die Bearbeitungsrate liefert:

% Bei mu = 16.9169 ergibt sich das folgende Resultat für die

% Wahrscheinlichkeit:

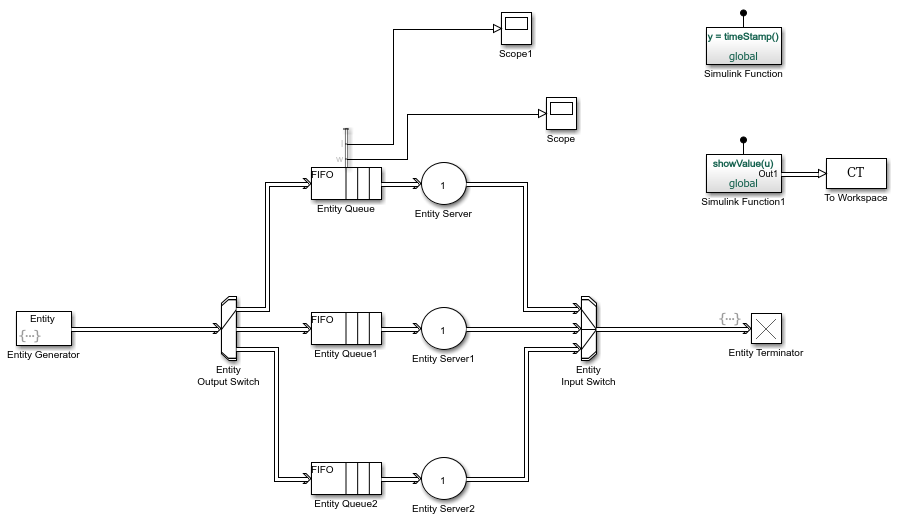
% p =

%

% 0.9501

1. Aufgabe (SimEvents Routing)

Ein Computer hat drei Prozessoren, zwei schnelle und einen langsameren. Die Prozessoren 1 und 2 brauchen für einen Job eine exponentialverteilte Bearbeitungszeit mit Mittelwert 1 s. Prozessor 3 benötigt eine Bearbeitungszeit, die gleichverteilt im Intervall [0.5, 2] Sekunden ist. Die Jobs kommen als Poissonprozess an mit einer Rate von 2 Jobs/Sekunde.   
Wenn ein Job ankommt, wird er auf Prozessor i (i=1,2,3) geroutet mit einer Wahrscheinlichkeit pi=1/3. Wenn der Prozessor, auf den geroutet wird, beschäftigt ist, wartet der Job an diesem Prozessor in einer unendlich grossen Warteschlange.

1. Bauen Sie ein SimEvents-Simulationsmodell. Schätzen Sie mittels Ihrer Simulation die mittlere Systemzeit für einen Job. Vergewissern Sie sich, dass Ihr Schätzfehler klein genug ist (wiederholen Sie die Simulation ein paar Mal und beobachten Sie die Variabilität).  
     
     
     
   
2. Ändern Sie die Routing-Policy, indem Sie den verschiedenen Prozessoren verschiedene Routing-Wahrscheinlichkeiten zuordnen. Bestimmen Sie die optimalen Routing-Wahrscheinlichkeiten, d.h. diejenigen Wahrscheinlichkeiten, die die kleinste mittlere Durchlaufzeit ergeben.   
     
   Optimierung der Wahrscheinlichkeiten: Aus Symmetriegründen müssen die Wahrscheinlichkeiten für die beiden ersten Prozessoren gleich sein. Deshalb hat man als einzige Unbekannte die Wahrscheinlichkeit, auf Prozessor drei zu routen.  
     
   Es ergibt sich ein optimaler Wert von etwa 0.3 für Prozessor 3 mit einer minimalen Systemzeit von etwa 3.38.