

Triage op de spoedeisende hulp

BWI Werkstuk, augustus 2008

Matthijs Kooy

Vrije Universiteit Amsterdam

Faculteit der Exacte Wetenschappen

De Boelelaan 1081a

1081 HV Amsterdam



Voorwoord

Een van de laatste onderdelen van de studie Bedrijfswiskunde en Informatica is het schrijven van het BWI-werkstuk. Het schrijven hiervan vindt plaats in de periode vlak voor de eindstage. Het doel van dit werkstuk is om een (literatuur)onderzoek te doen naar een onderwerp dat tenminste twee van de drie BWI componenten bevat (Bedrijfskunde, Wiskunde en Informatica).

Dit werkstuk zal gaan over triage op de spoedeisende hulp van ziekenhuizen. Door middel van het triagemodel worden mensen die op de spoedeisende hulp binnen komen, geordend naar urgentie van de klacht. Een binnenkomende patiënt die een klacht heeft met een hoge urgentie, zal daardoor eerder worden behandeld dan een patiënt met een klacht met lage urgentie, ook al kwam de patiënt met de hoge urgentie op een later tijdstip aan dan de patiënt met lage urgentie.

Ik zou graag Ger Koole willen bedanken voor zijn suggestie voor het onderwerp en René Bekker zou ik graag willen bedanken voor zijn begeleiding tijdens het schrijven van dit werkstuk. Verder zou ik de lezer veel plezier willen wensen bij het lezen van dit werkstuk.

Matthijs Kooy,
Augustus 2008

Samenvatting

Om patiënten op de spoedeisende hulp het beste te kunnen helpen, bestaan er zogenaamde triagemodellen. Deze modellen selecteren en ordenen de patiënten op basis van hun verwondingen. De patiënten met de zwaarste verwondingen die snel behandeld moeten worden krijgen de hoogste prioriteit en de patiënten die nog wel even kunnen wachten met hun verwondingen, krijgen een lagere prioriteit.

In Engeland hebben ze hier al geruime tijd ervaring mee en is een model ontworpen dat het Manchester Triage System heet. Het systeem heeft vijf prioriteitsklassen en voor iedere prioriteitsklasse staat een maximale wachttijd voorgeschreven.

Naar aanleiding van het Manchester Triage Systeem en verkregen data zijn verschillende wachtrijmodellen opgesteld en doorgerekend. Het blijkt dat in tegenstelling tot de laagste drie prioriteitsklassen, de maximale wachttijd voor de hoogste twee prioriteitsklassen moeilijker zijn te bereiken als we het aantal behandelruimtes groter maken.

De belangrijkste conclusie die we uit dit onderzoek kunnen halen is dat triage alleen zinvol is wanneer de bezettingsgraad hoog is. Wanneer de bezettingsgraad laag is, zijn de wachttijden zo laag, dat men beter het “see and treat” principe kan gebruiken, omdat dit eenvoudiger en net zo effectief is.

Inhoudsopgave

VOORWOORD	I
SAMENVATTING	II
INHOUDSOPGAVE	III
INLEIDING	1
Wat is triage?	1
Triage in Nederland	1
Manchester Triage System	2
Onderzoek naar triage in de literatuur	4
WACHTRIJMDELLEN	6
Notatie en wachtrij-eigenschappen	6
Kendall notatie	6
De formules van Little	7
De PASTA eigenschap	8
De M/M/1 wachtrij	8
De M/M/c wachtrij	9
De M/M/1 en M/M/c wachtrij toegepast	10
Wachtrijmodellen met prioriteiten	14
M/M/1 wachtrij met non-preemptive prioriteiten	14
M/M/c wachtrij met non-preemptive prioriteiten met gelijke behandeltijd	15
M/M/c wachtrij met non-preemptive prioriteiten met ongelijke behandeltijd	15
De wachtrijmodellen met prioriteiten toegepast	16
CONCLUSIE	20
MOGELIJKHEDEN TOT VERVOLGONDERZOEK	21
APPENDIX A	22
De M/M/1 wachtrij	22
De M/M/c wachtrij	24
APPENDIX B	26
BRONNEN	36
BRONNEN	36
Literatuur	36
Websites	36

Inleiding

Wat is triage?

Het begrip triage is afgeleid van het Franse woord *trier*, wat volgens het woordenboek *uitzoeken* of *sorteren* betekend. In medische context werd het begrip voor het eerst gebruikt door Dominique-Jean Larrey, een chirurg in het leger van Napoleon. Er vielen in een korte tijd een groot aantal gewonden, terwijl het aantal medische faciliteiten zeer beperkt was. Larrey ontwikkelde een systeem waarbij de gewonde soldaten werden geselecteerd en geordend, ongeacht hun status of rang. Allereerst werd bekeken wie er überhaupt voor medische behandeling in aanmerking kwam. Als de verwondingen zodanig waren dat de kans op genezing nihil was, werd de gewonde aan zijn lot overgelaten. Vervolgens keek men wie als eerst moest worden behandeld en wie nog wel een tijdje met zijn verwondingen kon wachten voordat hij werd geholpen. Het revolutionaire aan dit systeem was dat de selectie enkel gebaseerd was op de ernst van de verwondingen. Daarvoor keek men vooral naar de status en de rang van de gewonde; mensen met een hoge rang werden als eerste geholpen.

Triage heeft dus betrekking op selectie van patiënten. Een belangrijk criterium waarop wordt geselecteerd is urgentie: heeft de patiënt direct medische hulp nodig of is het minder dringend. Tegenwoordig wordt het begrip triage gebruikt bij acute zorg, zoals bij calamiteiten en binnen de afdeling spoedeisende hulp in ziekenhuizen. Ook hier is het aanbod van patiënten wisselend en kan de tijd die mensen op een medische behandeling moeten wachten oplopen. De ene patiënt heeft direct medische hulp nodig en bij de ander zijn de verwondingen niet meteen van levensgevaar.

Triage in Nederland

In 2005 heeft de Nederlandse Vereniging spoedeisende hulp Verpleegkundigen in samenwerking met verschillende andere verenigingen op het gebied van de zorg het initiatief genomen om een richtlijn op te stellen voor triage op de spoedeisende hulp [7]. Directe aanleiding hiervoor was dat in verschillende ziekenhuizen in Nederland de spoedeisende hulp verstopt raakt doordat er een grote stijging is ontstaan in het aantal mensen dat naar de spoedeisende hulp komt. De oorzaken hiervan zijn onder andere de vergrijzing en het feit dat mensen om diverse redenen de huisarts passeren en zelf direct naar de spoedeisende hulp van het ziekenhuis gaan. Hierdoor zou het voor kunnen komen dat mensen die een klacht hebben die een grote urgentie vereist, te lang zouden moeten wachten in de wachtruimte waardoor de gezondheid van deze mensen ernstige schade zou kunnen leiden.

Op het moment van schrijven van dit initiatief bestond er geen systematische werkwijze voor de behandeling van binnenkomende patiënten. Patiënten met een hoge urgentie, bijvoorbeeld mensen die met een ambulance of met een traumahelikopter bij de spoedeisende hulp arriveren worden uiteraard direct behandeld. De opvang van de andere patiënten wordt uitgevoerd door een baliemedewerker die op basis van ervaring en intuïtie bepaald of de patiënt directe hulp

nodig heeft of dat de patiënt nog wel even in de wachtkamer zou kunnen wachten. De behandeling van deze laatste groep mensen begint dan pas als de patiënt uit de wachtkamer wordt gehaald. Dit wordt gedaan op basis van de regel wie het eerst komt, wie het eerst maalt. Het gevolg hiervan is dat patiënten geïrriteerd of soms zelf agressief kunnen reageren omdat ze niet weten waar ze aan toe zijn. Een ander belangrijk knelpunt van deze werkwijze is dat er geen sprake is van procesbeheersing, waardoor wacht- en doorlooptijden niet te beïnvloeden zijn.

Een triagemodel zou voor deze problemen een goede oplossing kunnen zijn. Triage begint namelijk meteen bij de voordeur, in een aparte kamer tussen de wachtkamer en de behandelruimte in. Een triageverpleegkundige bekijkt de patiënt en binnen enkele minuten krijgt deze te horen hoe urgent zijn situatie is. Vervolgens krijgt de patiënt reële informatie over de wachttijd. Patiënten waarvan de klachten niet urgent zijn, kunnen kiezen: blijven wachten of weggaan. Het classificeren van de patiënten gebeurt aan de hand van vaste stroomschema's en op basis van observaties, metingen, kennis en ervaring. Elke urgentiecategorie heeft een maximaal toegestane wachttijd op de arts waardoor patiënten meteen weten waar ze aan toe zijn. Op deze manier worden mensen over het algemeen een stuk minder geïrriteerd of agressief. Ook kunnen er eenvoudig prioriteiten worden gesteld aan zorg, is er een grotere controle van de patiëntenstroom en is er een betere overdracht van patiënten naar een geschikte behandelruimte. Verder kan er meteen worden begonnen met een diagnostisch onderzoek en zal er bij de patiënten een afname van angst, bezorgdheid en frustratie worden gerealiseerd. Minder voor de hand liggende voordelen van de invoering van een triagemodel zijn afname van zowel schriftelijke en mondelinge klachten en een stijging van de werknemerstevredenheid. Nadelen zijn er echter ook. Niet alle triagesystemen lijken altijd even betrouwbaar te zijn. De systemen zijn wel sensitief genoeg om kritiek zieke patiënten te identificeren, maar dan moet het systeem wel gehanteerd worden als dynamisch systeem waarbij de toestand van patiënten in de wachtkamer regelmatig wordt geëvalueerd.

Bij het schrijven van de richtlijn heeft men verschillende triagemodellen van over de hele wereld onder de loep genomen. Hierbij is gekeken naar de mate waarin het model toepasbaar is in Nederland en naar de wetenschappelijke onderbouwing van de modellen. De belangrijkste punten waren vooral dat het triagemodel voor iedereen, voor jong en oud, mannen en vrouwen toepasbaar moet zijn. Ook moet het model werken volgens een verpleegkundige methodiek. Verpleegkundigen in Nederland hebben nu eenmaal geen puur medische opleiding genoten. Daarom heeft het geen zin een model te gebruiken waarbij er kennis en vaardigheden worden verwacht die in Nederland niet onder het verpleegkundige domein vallen.

Manchester Triage System

Uiteindelijk is aanbevolen om het Manchester Triage System (MTS) op de spoedeisende hulp afdeling van ziekenhuizen in Nederland te implementeren. Dit systeem stelt duidelijk dat de zorgbehoefte van de patiënt het hoofddoel is en dat klachten en symptomen belangrijker zijn dan de diagnose. Dit is belangrijk omdat dit ervoor zorgt dat men snel de urgentie van de klacht kan vaststellen. Ook is het zo dat men bij triage niet volledig objectief kan zijn en dat men niet

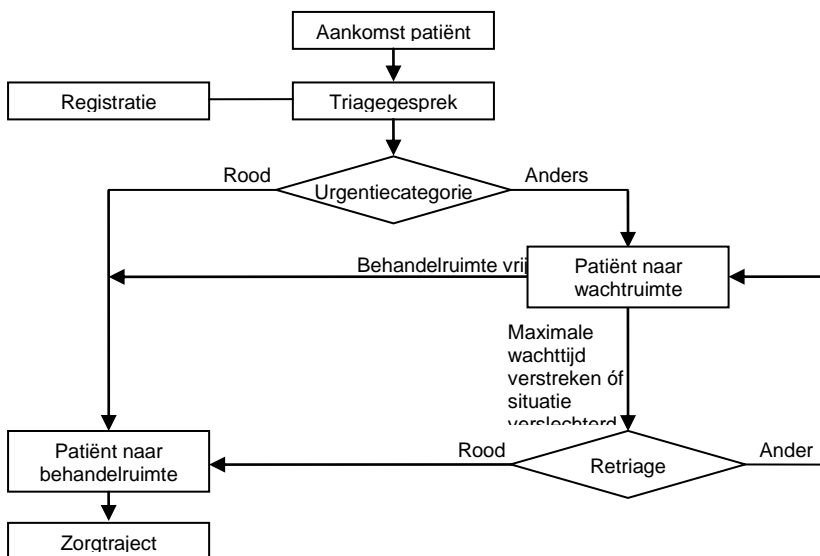
historisch volledig kan zijn. Dit houdt in dat een triageverpleegkundige die een patiënt ziet, niet altijd weet welke aandoeningen de patiënt in het verleden heeft gehad. Verder is het zo dat dit systeem praktisch toepasbaar is, dat het voldoende valide en betrouwbaar is en dat de gebruikte methodiek in het verlengde ligt van wat er in de huidige situatie van de verpleegkundigen op de spoedeisende hulp wordt verwacht. Dit komt doordat de Engelse gezondheidszorg meer op de Nederlandse situatie lijkt in verhouding tot landen als Amerika, Canada en Australië, waar men ook een eigen triagesysteem heeft.

Het systeem is in 1997 in Manchester ontwikkeld en is het enige Europese triagesysteem. Het systeem werkt met vijf urgentieniveaus waar binnenkomende patiënten worden ingedeeld. Elk urgentieniveau heeft een eigen kleur en een maximaal toegestane wachttijd op een arts (zie tabel 1). Zo worden mensen die worden gecategoriseerd als

Naam	Kleur	Wachttijd
Acuut	Rood	0 minuten
Zeer urgent	Oranje	10 minuten
Urgent	Geel	60 minuten
Standaard	Groen	120 minuten
Niet urgent	Blauw	240 minuten

Tabel 1: Urgentietabel behorende bij het MTS

“acuut” direct geholpen en hebben prioriteit boven alle andere patiënten. Iemand die wordt gecategoriseerd als “standaard” kan naar de wachtkamer met de wetenschap dat de maximale wachttijd kan oplopen tot 2 uur. Deze persoon zal iedereen met een hoger urgentieniveau voor moeten laten gaan, ook al kwamen deze mensen later op de spoedeisende hulp. Op deze manier kan het voorkomen dat wanneer alleen iemand de in wachtkamer zit die is gecategoriseerd als “urgent” en daarna komt een patiënt binnen die wordt gecategoriseerd als “acuut”, dat deze patiënt eerder wordt geholpen dan de “urgente” patiënt. Binnen een bepaald urgentieniveau geldt het principe van wie het eerst komt, wie het eerst maalt. Op het moment dat de maximale wachttijd is verstreken, of dat blijkt dat de situatie van de patiënt dusdanig is verslechterd, kan de patiënt nogmaals door de triageverpleegkundige worden gezien en een zogenaamde retriage uitvoeren. Hierbij kan men het urgentieniveau eventueel aanpassen, waardoor men ervoor kan zorgen dat deze patiënten eerder aan de beurt komen. Het hele triageproces vanaf de aankomst van de patiënt tot en met de behandeling is in onderstaande figuur samen te vatten.



Figuur 1: Het triageproces

Onderzoek naar triage in de literatuur

Hoewel triage in sommige landen al een redelijk lange tijd wordt gebruikt, is er niet al te veel literatuur aanwezig over de invloed van triage op de wachttijden. Wel zijn er een aantal artikelen te vinden die andere gevolgen van het invoeren van triage op de spoedeisende hulp behandelen. De resultaten van een aantal artikelen hierover zal in het kort worden beschreven.

Evaluation of nurse triage in a British accident and emergency department [3]

Dit onderzoek is uitgevoerd in Groot-Brittannië in 1990, dus nog vóór de ontwikkeling en implementatie van het Manchester Triage System. Bij dit onderzoek is de invloed van triage op de wachttijden van binnenkomende patiënten onderzocht. Het zes weken durende onderzoek is bij één ziekenhuis gehouden. Vijf jaar voor de uitvoering van het onderzoek heeft dit ziekenhuis een niet nader beschreven triagemodel met vier verschillende urgentiecategorieën ingevoerd. In de zes weken heeft men van ruim 5000 patiënten de wachttijden verzameld en geanalyseerd. Men heeft de tijd in twee periodes opgedeeld; een periode waarin men geen triagemodel gebruikte en een periode waarin men het model met vier urgentiecategorieën gebruikte. Uit dit onderzoek kwam naar voren dat de gemiddelde wachttijd van patiënten die in de periode kwamen waarin men een triagemodel gebruikte, langer was dan in de periode waarin men geen triagemodel gebruikte. Volgens dit onderzoek heeft triage dus een negatief effect op de wachttijd van de patiënten.

The effect of low-complexity patients on emergency department waiting times [9]

Voor een triagesysteem met drie verschillende prioriteitsklassen, is onderzocht wat het effect is van het aantal mensen in de laagste prioriteitsklasse vallen, hebben op de wachttijd van de mensen die in de hoogste prioriteitsklasse vallen. Met behulp van een grote dataset hebben ze voor de hoogste prioriteitsklasse berekend met hoeveel tijd de wachttijd toeneemt wanneer er iemand van de laagste prioriteitsklasse naar de spoedeisende hulp komt. Het resultaat van dit onderzoek is dat de wachttijd maar minimaal toeneemt naar mate er meer laaggeclassificeerde patiënten naar de spoedeisende hulp komen.

An advance triage system [1]

In dit artikel wordt een methode beschreven om de tijd die patiënten moeten wachten effectief te benutten door diagnostische tests, zoals laboratoriumonderzoek, in gang te zetten. Tegen de tijd dat de patiënt door de behandelaar wordt gezien, zijn de eerste testresultaten binnen. Dit systeem wordt door de schrijver het "advance triage system" genoemd. Dit systeem wordt uitgevoerd met behulp van eigen ontwikkelde "advance triage algorithms" en staat los van het initiële triageproces. Wel heeft het betrekking op de effectiviteit van het verdere zorgproces en kan daarmee de doorlooptijd van patiënten inkorten door effectief gebruik te maken van de wachttijd van de patiënt.

NHS Direct: consistency of triage outcomes [5]

NHS Direct is een hulplijn die mensen in Engeland kunnen bellen. Mensen vertellen wat de klachten zijn en krijgen advies wat ze het beste kunnen doen. De triage gebeurt dus op basis van wat de mensen via de telefoon vertellen. Omdat het belangrijk is dat patiënten met dezelfde klachten ook dezelfde triageuitkomst krijgen, hebben ze hier een onderzoek naar gedaan. Vooraf zijn 119 verschillende scenario's gemaakt die gebaseerd zijn op praktijkvoorbeelden. Vervolgens zijn deze scenario's voorgelegd aan de mensen die normaal gesproken de triage uitvoeren en is gekeken in welke mate er verschillen optreden in de triageuitkomst. Iedere triageverpleegster gebruikte verschillende soorten software bij het classificeren van de verschillende scenarios. Tussen de triageverpleegsters onderling waren geen grote verschillen te ontdekken. Wel waren er verschillen te zien in de triageuitkomsten tussen de verschillende software die gebruikt werd.

Effects of actual waiting time, perceived waiting time, information delivery and expressive quality on patient satisfaction in the emergency department [2]

Het onderzoek dat in dit artikel naar voren komt gaat over de effecten van de werkelijke wachttijd, de wachttijd die de patiënten ervaren, de informatie over de wachttijd die aan de patiënt worden gegeven en hoe men met de mensen omgaat op de patiënttevredenheid. Om dit te meten hebben ze mensen die naar de spoedeisende hulp zijn gekomen, na afloop een aantal vragen gesteld. De conclusie van het artikel is dat de werkelijke wachttijd niet van invloed is op de patiënttevredenheid, maar alle andere zaken wel. Het zal daarom effectiever zijn om vriendelijker naar de patiënt te zijn en hem meer en goede informatie over de wachttijd te geven, dan de wachttijd daadwerkelijk te verlagen.

Wachtrijmodellen

Omdat we willen gaan kijken wat de invloed van triage nu precies is op de wachttijd van binnenkomende patiënten, willen we de wachttijd uiteraard gaan vergelijken met de situatie dat er geen triage systeem gebruikt wordt. Dit doen we door het wachtproces van deze situaties te modelleren. Voordat we hier mee gaan beginnen, bekijken we eerst welke notatie er gebruikt wordt bij het modelleren van wachtrijmodellen en daarnaast bekijken we een aantal belangrijke eigenschappen hiervan. Hierna gaan we bepalen wat de capaciteit van de spoedeisende hulp zal moeten zijn als we weten hoeveel mensen de spoedeisende hulp per dag bezoeken. Vervolgens kunnen we de gemiddelde wachttijden vergelijken in het geval er een triagesysteem gebruikt wordt.

Notatie en wachtrij-eigenschappen

Kendall notatie

In het dagelijks leven maken we iedere dag wel een aantal keer mee dat we in een wachtrij staan. Deze wachtrijen bestaan omdat er gewoonweg niet altijd genoeg middelen voor handen zijn om iedereen tegelijk van dienst te zijn. Simpele voorbeelden zijn wachten in de rij bij de supermarkt of wachten bij het bellen naar een call center. Wachtrijmodellen zijn er dus in vele vormen en doordat er een aantal verschillende factoren zijn die een rol spelen in het modelleren van een wachtrij kan het modelleren soms erg complex zijn.

De meest gebruikte notatie bij het beschrijven van wachtrijen is de Kendall notatie. Deze notatie heeft de volgende vorm: $a/b/c/d/e$. Deze symbolen beschrijven de meest belangrijke factoren waarmee rekening moet worden gehouden bij het modelleren van wachtrijmodellen. Deze factoren zijn de volgende:

- a*: Aankomst proces van klanten (in ons geval dus binnenkomende patiënten)
- b*: Behandeltijden van klanten
- c*: Het aantal behandelruimtes (of bij call centers het aantal telefonisten)
- d*: Het aantal plaatsen in het systeem, dus zowel het aantal mensen in de wachtrij als het aantal mensen dat tegelijk behandeld kan worden.
- e*: De service discipline

Veel voorkomende aankomstprocessen zijn:

- M: Markovian, als het aankomstproces een Poisson proces is (de tijden tussen twee opeenvolgende aankomsten zijn exponentieel verdeeld)
- D: deterministic, de tussenaankomsttijden zijn constant
- G: general, de verdeling van de tussenaankomsttijden is niet gespecificeerd

Voor de behandeltijd worden vaak deze symbolen gebruikt:

- M: als de behandeltijden exponentieel verdeeld zijn
- D: de behandeltijden zijn constant
- G: de verdeling van de behandeltijden is niet gespecificeerd

Als het aantal plaatsen in het systeem niet is gespecificeerd, dan kan worden aangenomen dat deze oneindig groot is. Voor het M/M/1 wachtrijmodel geldt dus dat de tijd tussen twee opeenvolgende aankomsten exponentieel verdeeld is, dat de behandeltijden ook exponentieel verdeeld zijn en dat er 1 behandelruimte beschikbaar is. Verder is het zo dat er geen limieten aan het aantal plaatsen in de wachtruimte en aan het totaal aantal klanten in het systeem zijn.

Naast de notatie van Kendall voor de classificering van de diverse wachtrijmodellen, gebruiken we ook nog de volgende notatie voor de verschillende parameters die we in dit hoofdstuk zullen gebruiken.

λ : de parameter voor het Poisson aankomstproces

S: de behandeltijd

μ : de parameter van de verdeling van de behandeltijd in het geval deze exponentieel verdeeld is

β : is de verwachting van de behandeltijd (dus $\beta = E(S)$ en $\beta = \frac{1}{\mu}$ in het geval deze exponentieel verdeeld is)

c: het aantal behandelruimtes

ρ : de bezettingsgraad (dus $\rho = \frac{\lambda}{c\mu}$), vaak voorkomende voorwaarde is dat deze kleiner moet zijn dan 1, omdat anders het aantal klanten dat binnenkomt groter is dan wat het personeel aankan.

$E(W_Q)$: de tijd dat een patiënt in een stationaire situatie gemiddeld wacht in de wachtkamer voordat hij wordt behandeld.

$E(W)$: de tijd dat een patiënt in een stationaire situatie zich gemiddeld in het systeem bevindt, dus de gemiddelde tijd dat de patiënt wacht *plus* de gemiddelde tijd dat de patiënt wordt behandeld. (dus $E(W) = E(W_Q) + \beta$)

$E(L_Q)$: het gemiddeld aantal patiënten in de wachtrij.

$E(L)$: het gemiddeld aantal patiënten in het systeem, dus het gemiddeld aantal in de wachtrij *plus* het gemiddeld aantal dat behandeld wordt (dus $E(L) = E(L_Q) + c\rho$).

π : de stationaire verdeling van het aantal patiënten in het systeem.

De formule van Little

De formule van Little geeft een belangrijke relatie tussen het gemiddeld aantal patiënten in het systeem en de gemiddelde verblijftijd van een patiënt in het systeem:

$$E(L) = \lambda E(W)$$

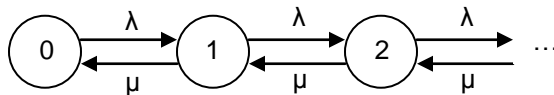
Of in woorden gezegd: als er gemiddeld λ patiënten per uur aankomen en elke patiënt blijft gemiddeld $E(W)$ uur, dan zal men gemiddeld $\lambda E(W)$ patiënten aantreffen. Deze formule is algemeen geldig. Deze hangt niet af van de bedieningsvolgorde en geldt ook voor systemen waarbij bepaalde wachtrijen prioriteit hebben boven andere. Ook geldt het voor het gemiddeld aantal patiënten in de wachtrij, waardoor de vergelijking $E(L_Q) = \lambda E(W_Q)$ ook op gaat.

De PASTA eigenschap

Het woord PASTA staat voor: Poisson Arrivals See Time Averages. Dit houdt in dat de fractie van het aantal binnenkomende patiënten dat een A aantal wachtende patiënten in de wachtruimte treft gelijk is aan de stationaire verdeling van het aantal personen A in de wachtrij. Deze eigenschap is alleen geldig in het geval we te maken hebben met een Poisson aankomstproces. Om in eerste instantie de invloed van triage op de wachttijd te kunnen vergelijken, bekijken we eerst het meest simpele (maar misschien niet het meest reële) model dat we kunnen verzinnen. De aanname dat binnenkomende patiënten met een Poisson proces binnenkomen, is een erg realistische aanname; dit omdat deze patiënten met spoed binnenkomen. Voorlopig houden we ook aan dat de behandeltijd exponentieel verdeeld is en dat er maar 1 behandelruimte beschikbaar is. We zullen nu eerst een aantal eigenschappen van dit wachtrijmodel onderzoeken en daarna zullen we gaan kijken hoe deze veranderen als we met prioriteiten gaan werken.

De M/M/1 wachtrij

Bij het M/M/1 wachtrijmodel gaan we er dus vanuit dat de zowel de tussenaankomsttijden (met intensiteit λ) als de behandeltijden (met intensiteit μ) exponentieel verdeeld zijn en dat we 1 behandelruimte tot onze beschikking hebben. Voor dit wachtrijmodel kunnen we het onderstaande toestandsdiagram voor het aantal patiënten in de wachtrij maken.



Figuur 2: Toestandsdiagram van het M/M/1 wachtrijmodel

Van dit wachtrijmodel noemen we de stationaire verdeling π . De bezettingsgraad van de wachtrij wordt als volgt berekend: $\rho = \frac{\lambda}{\mu}$. Voor het bestaan van een stationaire verdeling moet gelden dat $\rho < 1$, omdat er anders meer mensen komen dan dat er behandeld kunnen worden. Een afleiding van deze resultaten is te vinden in appendix A.

$$\pi(x) = \rho^x (1 - \rho)$$

$$E(L) = \frac{\rho}{1 - \rho}$$

$$E(L_Q) = \frac{\rho^2}{1 - \rho}$$

$$E(W) = \frac{1}{\mu(1 - \rho)}$$

$$E(W_Q) = \frac{\rho}{\mu(1 - \rho)}$$

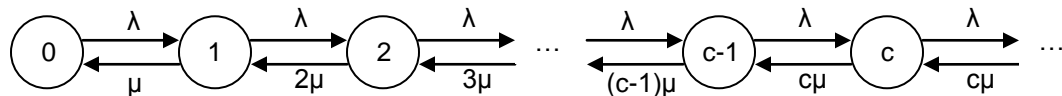
Verder zijn ook de verdelingen van de wachttijd in de wachtkamer en tijd in het systeem bekend:

$$P(W_Q > t) = \rho e^{-\mu(1-\rho)t}$$

$$P(W > t) = \mu(1-\rho)e^{-\mu(1-\rho)t}$$

De M/M/c wachtrij

De M/M/c wachtrij heeft dezelfde verdelingen van de aankomsttijden en behandelzeiten als bij de M/M/1 wachtrij, alleen hebben we in dit geval c ($c > 1$) behandelruimtes tot onze beschikking. Het toestanddiagram van dit wachtrijmodel staat hieronder weergegeven.



Figuur 3: Toestanddiagram van het M/M/c wachtrijmodel

Ook bij dit wachtrijmodel noemen we de stationaire verdeling π . De bezettingsgraad wordt in tegenstelling tot de M/M/1 wachtrij nu als volgt berekend: $\rho = \frac{\lambda}{c\mu}$. Ook hier is het zo dat voor het

bestaan van een stationaire verdeling moet gelden dat $\rho < 1$. Van onderstaande resultaten is een weer afleiding te vinden in Appendix A.

De stationaire verdeling:

$$\pi(x) = \frac{1}{x!} \left(\frac{\lambda}{\mu} \right)^x \pi(0) \text{ voor het geval dat } 0 \leq x \leq c$$

en
$$\pi(x) = \frac{1}{c! c^{x-c}} \left(\frac{\lambda}{\mu} \right)^x \pi(0) \text{ voor het geval dat } x > c$$

met
$$\pi(0) = \left[\sum_{x=0}^{c-1} \frac{(c\rho)^x}{x!} + \frac{(c\rho)^c}{c!(1-\rho)} \right]^{-1}$$

De kans dat we met dit wachtrijmodel moeten wachten noemen we $\pi_w(c)$ en is als volgt:

$$C(c,\rho) = \frac{(c\rho)^c}{c!(1-\rho)} \left[\sum_{x=0}^{c-1} \frac{(c\rho)^x}{x!} + \frac{(c\rho)^c}{c!(1-\rho)} \right]^{-1}$$

Verder weten we onderstaande resultaten:

$$E(L_Q) = \frac{\rho C(c,\rho)}{1-\rho} \quad E(L) = \frac{\rho C(c,\rho)}{1-\rho} + \frac{\lambda}{\mu}$$

$$E(W_Q) = \frac{C(c,\rho)}{c\mu - \lambda} \quad E(W) = \frac{C(c,\rho)}{c\mu - \lambda} + \frac{1}{\mu}$$

Ook weten we de verdeling van de wachttijd:

$$P(W_Q > t) = C(c,\rho) e^{-c\mu(1-\rho)t}$$

De M/M/1 en M/M/c wachtrij toegepast

Nu we de resultaten weten van de M/M/1 en de M/M/c wachtrij, kunnen we deze gegevens toepassen op de situatie van een bestaande spoedafdeling van een Nederlands ziekenhuis. Om een indicatie te krijgen van deze gegevens, heb ik eerst een aantal documenten op internet proberen te zoeken.

Uit de richtlijn Triage op de spoedeisende hulp [7] kon ik de verdeling van het aantal mensen per prioriteitsklasse halen. In de analyse van dit document hebben ze voor een periode van 2 weken bij diverse ziekenhuizen gegevens over triage verzameld. In totaal is voor ruim 10.000 mensen die door de verschillende spoedeisende hulp afdelingen zijn behandeld, vastgelegd wat de herkomst van deze mensen was en in welke groep de mensen werden ingedeeld na triage.

Verder staat in de meeste jaarverslagen van ziekenhuizen het aantal mensen dat op jaarbasis een bezoek brengt aan de spoedeisende hulp. Als we dit combineren met de verdeling van de mensen over de verschillende kleuren, kunnen we per ziekenhuis voor de verschillende prioriteitsklassen een voorspelling van het totaal aantal aankomsten geven.

Daarnaast is het van belang dat we een indicatie kunnen krijgen voor de behandelduur per prioriteitsklasse. Op internet kwam ik hiervoor een onderzoek van de plexus medical Group [10] tegen. Hierin is te vinden wat voor de laatste 4 prioriteitsklassen de gemiddelde behandeltijden zijn. Met deze gegevens in het achterhoofd ben ik allereerst naar het VUmc gegaan. Hier heb ik het afdelingshoofd van de spoedeisende hulp gesproken. Uit het gesprek bleek dat het Manchester Triage System bij het VUmc tot op zekere hoogte wordt toegepast, maar dat hierover niets wordt bijgehouden of gedocumenteerd. Gegevens wat betreft aankomsttijden, wachttijden en behandeltijden konden ze me dus niet geven.

Wel konden ze me vertellen dat het BovenIJ ziekenhuis, wat ook in Amsterdam gevestigd is, wel gegevens bij zou houden. Ook hier heb ik een afspraak gemaakt en gesproken met het afdelingshoofd van de spoedeisende hulp. Deze vertelde me dat er inderdaad een aantal dingen werden bijgehouden, maar de informatie bleek erg lastig te interpreteren en bovendien konden ze me geen informatie over de behandelduur van de verschillende prioriteitsklasse geven. Omdat ik het toch interessant en leuk vond om een aantal dingen door te rekenen, heb ik besloten om de gegevens die ik in eerste instantie als indicatie wilde gebruiken, hiervoor te gebruiken.

Zoals hierboven valt te lezen, staat in de richtlijn Triage op de spoedeisende hulp een verdeling van het aantal mensen per prioriteitsklasse. In de richtlijn staat ook een percentage waarvan de triageuitkomst onbekend is. Wanneer we deze personen buiten beschouwing laten, komen we tot de onderstaande verdeling.

<i>Naam</i>	<i>Kleur</i>	<i>Percentage</i>
<i>Acuut</i>	<i>Rood</i>	<i>1,17 %</i>
<i>Zeer urgent</i>	<i>Oranje</i>	<i>17,33 %</i>
<i>Urgent</i>	<i>Geel</i>	<i>38,54 %</i>
<i>Standaard</i>	<i>Groen</i>	<i>41,63 %</i>
<i>Niet urgent</i>	<i>Blauw</i>	<i>1,33 %</i>

Tabel 2: Verdeling van de urgentiekleuren in procenten

Als we als voorbeeld het VUmc nemen, kunnen we in het jaarverslag lezen wat het aantal mensen is dat jaarlijks een bezoek brengt aan de spoedeisende hulp. Volgens het jaarverslag waren dat 36669 mensen over het jaar 2006. De gegevens over het jaar 2007 waren op het moment van schrijven nog niet bekend. De verwachting is dat het voor het jaar 2007 niet veel zal verschillen. We zien dus dat er ongeveer 100 mensen per dag op de spoedeisende hulp binnen komen, als we aannemen dat de verdeling over de dagen per jaar gelijk zal zijn. Als we deze getallen dan toepassen op de gegevens uit tabel 2, komen we op de volgende verdeling van het aantal patiënten over het jaar 2006:

Naam	Kleur	Percentage	Patiënten per jaar	Patiënten per dag
<i>Acuut</i>	<i>Rood</i>	<i>1,17 %</i>	429 patiënten	1,18 patiënten
<i>Zeer urgent</i>	<i>Oranje</i>	<i>17,33 %</i>	6356 patiënten	17,41 patiënten
<i>Urgent</i>	<i>Geel</i>	<i>38,54 %</i>	14131 patiënten	38,72 patiënten
<i>Standaard</i>	<i>Groen</i>	<i>41,63 %</i>	15266 patiënten	41,82 patiënten
<i>Niet urgent</i>	<i>Blauw</i>	<i>1,33 %</i>	488 patiënten	1,34 patiënten

Tabel 3: Verdeling van de urgentiekleuren in procenten, per jaar en per dag

Hieruit kunnen we aflezen dat er gelukkig niet al te veel mensen op de spoedeisende hulp komen die in de hoogste prioriteitsklasse vallen. Ook het aantal patiënten wat eigenlijk beter door een huisarts kan worden behandeld valt heel erg mee.

Uit het onderzoek van de plexus medical group weten we voor de laatste 4 prioriteitsklassen de gemiddelde behandelzeiten. Uit het onderzoek kunnen we helaas geen gegevens halen over de gemiddelde behandelduur van patiënten die worden gecategoriseerd in de hoogste urgentie categorie ("acuut"). Er zit helaas dus niets anders op dan voor deze urgentie categorie de behandelduur te schatten. Dit heb ik op de volgende manier gedaan: de behandelduur van prioriteitsklasse urgent is 1,5 keer zo groot als de behandelduur van de prioriteitsklasse standaard. Daarnaast is het zo dat de behandelduur van prioriteitsklasse zeer urgent 1,5 keer zo groot is als de behandelduur van de prioriteitsklasse urgent. Daarom heb ik de behandelduur van zeer urgent vermenigvuldigd met 1,5 om de behandelduur van acuut te schatten. Door dit te doen komt de behandelduur van de hoogste prioriteitsklasse uit op 80 minuten. De behandelduur van de verschillende prioriteitsklassen is in onderstaande tabel weergegeven.

Naam	Kleur	Behandelduur
<i>Acuut</i>	<i>Rood</i>	<i>80 minuten¹</i>
<i>Zeer urgent</i>	<i>Oranje</i>	<i>53 minuten</i>
<i>Urgent</i>	<i>Geel</i>	<i>35 minuten</i>
<i>Standaard</i>	<i>Groen</i>	<i>23 minuten</i>
<i>Niet urgent</i>	<i>Blauw</i>	<i>19 minuten</i>

Tabel 4: Behandelduur van de prioriteitsklassen

Nu we deze gegevens hebben, kunnen we deze op de M/M/c wachtrij toepassen en voor verschillende aantallen behandelruimten een berekening maken van de gemiddelde wachttijden en de gemiddelde lengte van de wachtrij. Vervolgens kunnen we dit vergelijken met de M/M/1 wachtrij waarvoor de gemiddelde behandelduur gedeeld wordt door het aantal behandelruimten.

¹ geschat

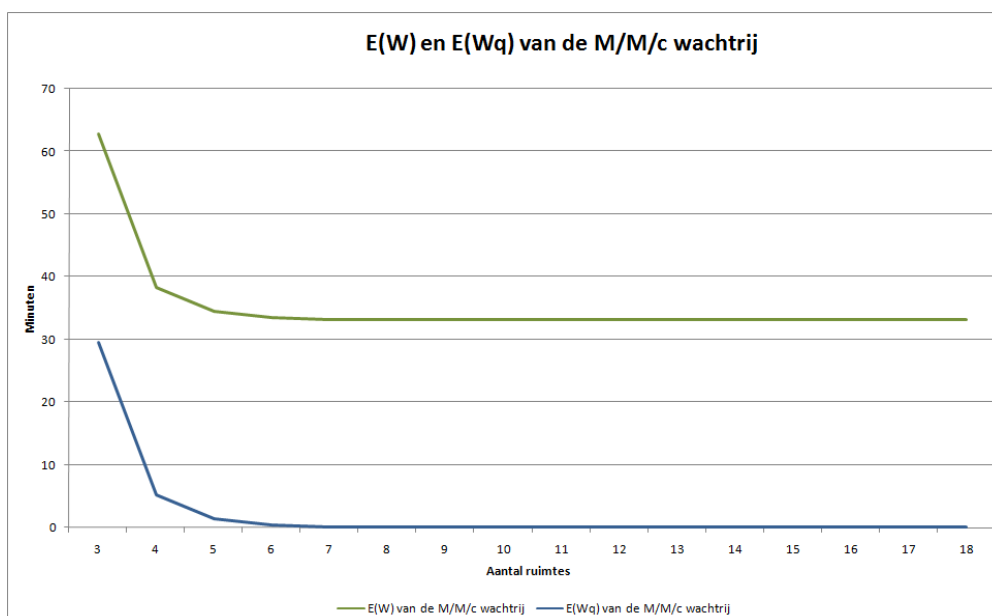
Voordat we dit allemaal kunnen doen, moeten we eerst de input berekenen. De twee gegevens die we nodig hebben zijn de aankomstintensiteit en de gemiddelde behandelduur. Voor de aankomstintensiteit hebben we gezien dat dit er ongeveer 100 per dag zijn. Per uur komen er dan 4,19 patiënten op de spoedeisende hulp. Dit gebruiken we dan ook als verwachte aankomstintensiteit. De parameter λ is daardoor: $\lambda = 4,19$. Voor de behandelduur berekenen we een gewogen gemiddelde. Hierdoor zal de behandelduur van de grote bulk patiënten het zwaarst meewegen. Als parameter voor de behandelduur krijgen we dan: 33 minuten en 15 seconden. Dit is dan een parameter van 0,56 per uur. We kunnen dus concluderen dat $\beta = 0,56$ en daardoor $\mu = \frac{1}{0,56} = 1,79$. Nu we deze twee parameters weten, kunnen we deze gaan gebruiken voor de

M/M/1 en M/M/c wachtrijmodellen.

De resultaten van de M/M/c wachtrij staan in onderstaande tabel en is de figuur daaronder weergegeven. We beginnen bij het aantal van 3 behandelruimten, omdat bij minder behandelruimten de bezettingsgraad groter wordt dan 1 en hierdoor het systeem overbelast raakt.

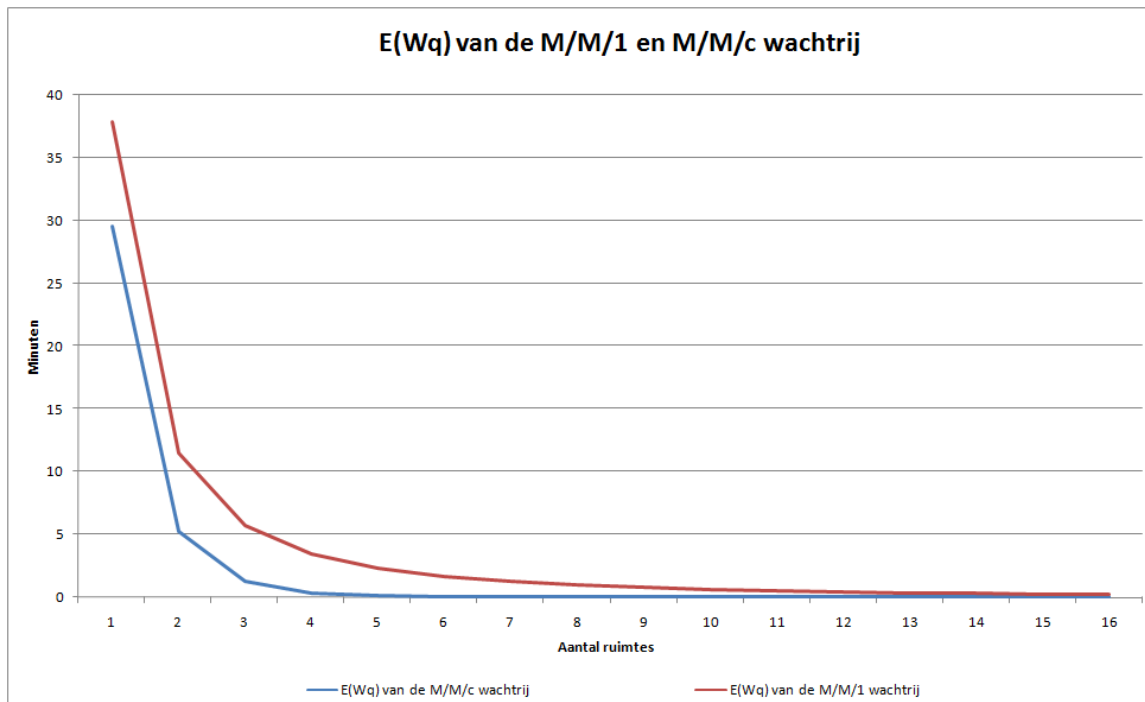
# Behandelruimten	$E(W_q)$	$E(W)$	$E(L_q)$	$E(L)$
3	29,58	62,84	2,06	4,38
4	5,19	38,45	0,36	2,68
5	1,25	34,51	0,09	2,41
6	0,31	33,57	0,02	2,34
7	0,07	33,33	0,01	2,33
8	0,02	33,27	0,00	2,32
9	0,00	33,26	0,00	2,32
10	0,00	33,26	0,00	2,32

Tabel 5: Resultaten van de M/M/c wachtrij



Figuur 4: Resultaten van de M/M/c wachtrij

We kunnen zien dat de wachttijd snel daalt naarmate er meer behandelruimtes gebruikt worden. Wanneer er 5 of meer wachtruimtes gebruikt worden, is de wachttijd te verwaarlozen. Deze resultaten willen we gaan vergelijken met de resultaten van de M/M/1 wachtrij. Om een goede vergelijking te kunnen maken, delen we de gemiddelde behandelduur met het aantal behandelruimtes. Hierdoor kunnen we zien wat het verschil in wachttijd tussen de M/M/1 en de M/M/c wachtrij is wanneer we het aantal behandelruimtes veranderen. De resultaten van de twee wachtrijmodellen zijn weergegeven in onderstaande grafiek.



Figuur 5: Resultaten van zowel de M/M/1 en de M/M/c wachtrij

We zien dat de wachttijd van de M/M/1 wachtrij veel langzamer daalt naarmate de aankomsttijd groter wordt. Pas als we de behandelduur met 8 vermenigvuldigen, is het verschil in wachttijd pas minder dan 2 minuten.

In het volgende hoofdstuk zullen we zien hoe dit zal zijn als we onderscheid gaan maken tussen de verschillende prioriteitsklassen.

Wachtrijmodellen met prioriteiten

Als er in de literatuur wordt gesproken over wachtrijmodellen met prioriteiten, dan wordt er meestal over twee varianten van de prioriteitsregel gesproken. Bij de eerste variant is het zo dat als er eenmaal begonnen is met behandelen van een van de prioriteiten, deze eerst wordt afgemaakt voordat men overgaat tot het behandelen van een hogere prioriteit. Dit type prioriteit wordt ook wel *non-preemptive* genoemd. Bij de tweede variant kan het behandelen van werk met een bepaalde prioriteit onderbroken worden door werk met een hogere prioriteit. Als al het werk van hogere prioriteit vervolgens is afgehandeld, gaat men weer verder waar men gestopt is. Dit type prioriteit heet in de literatuur *preemptive-resume*. De eerste variant zullen we hieronder uitwerken aan de hand van de mean value analyse. Voor de tweede variant is dit echter niet mogelijk, omdat er dan ook rekening gehouden moet worden met het feit dat behandeling op een later tijdstip hervat wordt.

M/M/1 wachtrij met non-preemptive prioriteiten

Bij deze wachtrij gaan we uit van een model waarbij we verschillende typen patiënten kunnen behandelen. Voor elk type patiënt geldt dat zowel de tussenaankomsttijden als de behandelzeiten exponentieel verdeeld zijn. Alleen is het zo dat intensiteiten van deze tijden per type patiënt verschillend zijn. Het type patiënt met de hoogste prioriteit noemen we een patiënt van type 1, het type patiënt met de een na hoogste prioriteit noemen we een patiënt van type 2 enzovoorts. Voor de patiënt met de hoogste prioriteit geldt hierdoor dat de tussenaankomsttijden exponentieel verdeeld zijn met intensiteit λ_1 en dat de behandelzeiten exponentieel verdeeld zijn met intensiteit μ_1 . Verder hebben we 1 behandelruimte tot onze beschikking en kan een patiënt die eenmaal in behandeling is niet worden onderbroken. Voor deze wachtrij kunnen we vervolgens onderstaande resultaten afleiden.

$$E(W_{Q1}) = \frac{\sum_{i=1}^r \frac{\rho_i}{\mu_i}}{1 - \rho_1} \quad E(W_1) = \frac{\sum_{i=1}^r \frac{\rho_i}{\mu_i}}{1 - \rho_1} + \frac{1}{\mu_1} \quad \text{waarbij } r \text{ het aantal prioriteiten}$$

$$E(L_{Q1}) = \frac{\lambda_1 \sum_{i=1}^r \frac{\rho_i}{\mu_i}}{1 - \rho_1} \quad E(L_1) = \frac{\lambda_1 \sum_{i=1}^r \frac{\rho_i}{\mu_i}}{1 - \rho_1} + \rho_1$$

En voor alle typen met een lagere prioriteit geldt dat:

$$E(W_{Qi}) = \frac{\sum_{j=1}^r \frac{\rho_j}{\mu_j}}{\left(1 - \sum_{j=1}^i \rho_j\right) \left(1 - \sum_{j=1}^{i-1} \rho_j\right)} \quad E(W_i) = \frac{\sum_{j=1}^r \frac{\rho_j}{\mu_j}}{\left(1 - \sum_{j=1}^i \rho_j\right) \left(1 - \sum_{j=1}^{i-1} \rho_j\right)} + \frac{1}{\mu_i}$$

$$E(L_{Qi}) = \frac{\lambda_i \sum_{j=1}^r \frac{\rho_j}{\mu_j}}{\left(1 - \sum_{j=1}^i \rho_j\right) \left(1 - \sum_{j=1}^{i-1} \rho_j\right)} \quad E(L_i) = \frac{\lambda_i \sum_{j=1}^r \frac{\rho_j}{\mu_j}}{\left(1 - \sum_{j=1}^i \rho_j\right) \left(1 - \sum_{j=1}^{i-1} \rho_j\right)} + \rho_i$$

Voor een afleiding van deze resultaten zou ik de lezer willen verwijzen naar de pagina's 205-210 van het boek van Gross en Harris [4], of naar het dictaat van Boxma [8] waar een afleiding voor het meer algemenere M/G/1 wachtrijmodel met prioriteiten wordt gegeven.

M/M/c wachtrij met non-preemptive prioriteiten met gelijke behandeltime

De analyse voor deze wachtrij is redelijk gelijk aan die van de vorige wachtrij (M/M/1 wachtrij met non-preemptive prioriteiten) behalve dat we nu helaas niet kunnen eisen dat ieder type patiënt een verschillende intensiteit heeft voor de behandeltime. Als we dit wel zouden willen afdwingen, zouden we op ieder moment precies moeten weten welke typen patiënten we in behandeling hebben. De analyse hiervan wordt daarom te ingewikkeld. Hierdoor moeten we er dus vanuit gaan dat iedere type patiënt dezelfde verdeling heeft voor de behandeltime. Voor de afleiding van $E(W_Q)$ zou ik de lezer weer willen verwijzen naar het boek van Gross en Harris [4], de pagina's 210-211. De resultaten van $E(W)$, $E(L)$ en $E(L_Q)$ kunnen we vervolgens krijgen aan de hand van de formules van Little en door gebruik te maken van de regels $E(L) = E(L_Q) + c\rho$ en $E(W_Q) + 1/\mu$.

$$E(W_{Q_i}) = \frac{C(c, \rho)}{\left(1 - \sum_{j=1}^i \rho_j\right) \left(1 - \sum_{j=1}^{i-1} \rho_j\right)} \quad E(W_i) = \frac{C(c, \rho)}{\left(1 - \sum_{j=1}^i \rho_j\right) \left(1 - \sum_{j=1}^{i-1} \rho_j\right)} + \frac{1}{\mu}$$

$$E(L_{Q_i}) = \frac{\lambda_i C(c, \rho)}{\left(1 - \sum_{j=1}^i \rho_j\right) \left(1 - \sum_{j=1}^{i-1} \rho_j\right)} \quad E(L_i) = \frac{\lambda_i C(c, \rho)}{\left(1 - \sum_{j=1}^i \rho_j\right) \left(1 - \sum_{j=1}^{i-1} \rho_j\right)} + \frac{\lambda_i}{\mu}$$

Waarbij $\rho_i = \frac{\lambda_i}{c\mu}$ en $C(c, \rho) = \left[c!(1-\rho)(c\mu) \sum_{n=0}^{c-1} \frac{(c\rho)^{n-c}}{n!} + c\mu \right]^{-1}$ weer de kans dat we moeten wachten noemen.

M/M/c wachtrij met non-preemptive prioriteiten met ongelijke behandeltime

Omdat we ook graag de resultaten zouden willen zien van de M/M/c wachtrij met non-preemptive prioriteiten en een ongelijke behandeltime, hebben we een simulatieprogramma gemaakt in C++. Het programma is getest door de resultaten ervan te vergelijken met de resultaten van de M/M/c wachtrij met non-preemptive prioriteiten met gelijke behandeltime en de M/M/1 wachtrij met non-preemptive prioriteiten.

De broncode van dit simulatieprogramma is terug te vinden in Appendix B.

Met de gegevens die we eerder al gebruikt hebben, kunnen we nu de drie hierboven behandelde wachtrijmodellen onderling vergelijken. Verder is het nu mogelijk om de wachttijden van deze verschillende wachtrijmodellen te vergelijken met de richtlijn voor triage op de spoedeisende hulp [7].

De wachtrijmodellen met prioriteiten toegepast

Om niet iedere keer terug te hoeven bladeren naar het hoofdstuk waar we de gegevens hebben toegepast op de M/M/1 en M/M/c wachtrij, hebben we hieronder de benodigde gegevens nog een keer weergegeven.

Naam	Kleur	Percentage	Patiënten per jaar	Patiënten per dag	Behandelduur
<i>Acuut</i>	<i>Rood</i>	<i>1,17 %</i>	429 patiënten	1,18 patiënten	<i>80 minuten²</i>
<i>Zeer urgent</i>	<i>Oranje</i>	<i>17,33 %</i>	6356 patiënten	17,41 patiënten	<i>53 minuten</i>
<i>Urgent</i>	<i>Geel</i>	<i>38,54 %</i>	14131 patiënten	38,72 patiënten	<i>35 minuten</i>
<i>Standaard</i>	<i>Groen</i>	<i>41,63 %</i>	15266 patiënten	41,82 patiënten	<i>23 minuten</i>
<i>Niet urgent</i>	<i>Blauw</i>	<i>1,33 %</i>	488 patiënten	1,34 patiënten	<i>19 minuten</i>

Tabel 6: Verdeling van de urgentiekleuren in procenten, per jaar en per dag. Daarnaast de behandelduur van de urgentiekleuren

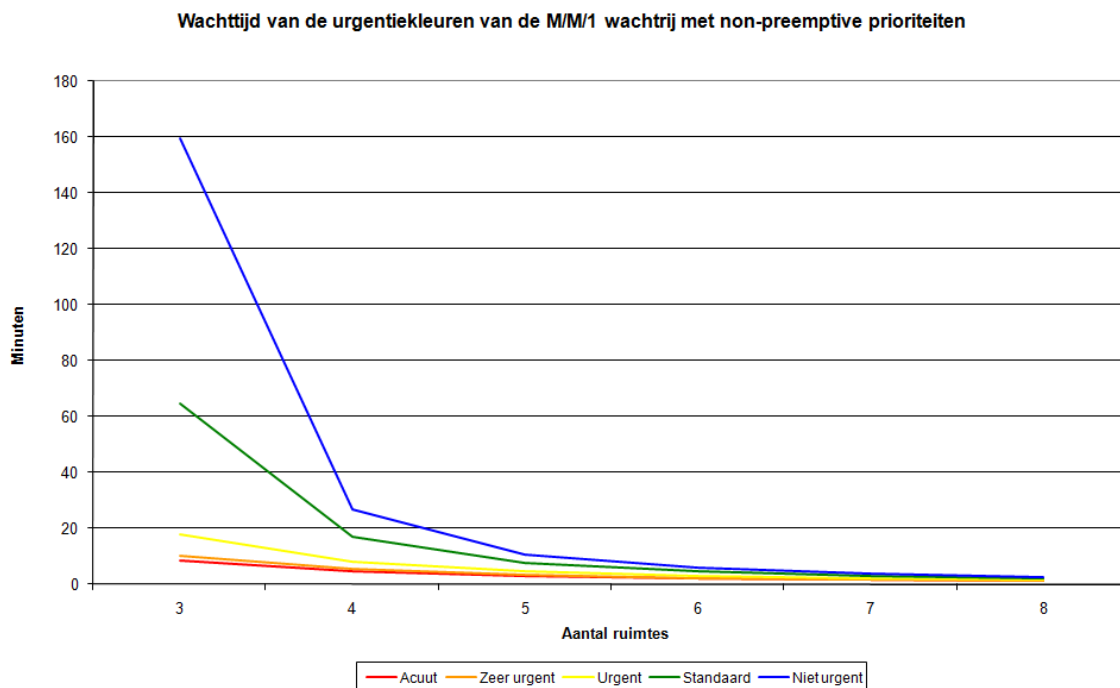
Deze gegevens hebben we vervolgens toegepast op zowel de M/M/1 wachtrij met non-preemptive prioriteiten als de M/M/c wachtrij met non-preemptive prioriteiten en gelijke behandeltijd. Later zullen we deze gegevens toepassen op de M/M/c wachtrij met non-preemptive prioriteiten en ongelijke behandeltijd. Voor de M/M/1 wachtrij kunnen we aankomsttijd en behandeltijd per urgentiekleur gebruiken. Voor de M/M/c wachtrij gebruiken we wel de aankomsttijd per urgentiekleur, maar niet de behandeltijd. Voor de behandeltijd zullen we weer een gewogen gemiddelde nemen. Verder zullen we voor de M/M/1 wachtrij de gemiddelde behandelduur weer delen door het aantal ruimtes, dit om een goede vergelijking te kunnen maken.

Omdat we vooral geïnteresseerd zijn in de wachttijd van de patiënten in de wachtkamer, kijken we alleen naar de $E(W_{Q_i})$. De resultaten van de M/M/1 wachtrij staan in onderstaande tabel en grafiek. De tabel en grafiek beginnen weer bij 3 ruimtes, omdat bij minder behandelruimten de bezettingsgraad groter wordt dan 1 en het systeem overbelast raakt.

# Behandelruimten	$E(W_{Q1})$	$E(W_{Q2})$	$E(W_{Q3})$	$E(W_{Q4})$	$E(W_{Q5})$
3	8,65	10,10	17,90	64,78	159,77
4	4,86	5,44	8,07	16,85	26,86
5	3,10	3,40	4,59	7,74	10,62
6	2,15	2,32	2,96	4,45	5,65
7	1,58	1,68	2,07	2,89	3,50
8	1,21	1,28	1,53	2,02	2,38
9	0,96	1,00	1,17	1,50	1,72
10	0,77	0,81	0,93	1,15	1,30
11	0,64	0,67	0,75	0,92	1,02
12	0,54	0,56	0,62	0,74	0,82
13	0,46	0,47	0,53	0,62	0,67
14	0,39	0,41	0,45	0,52	0,56
15	0,34	0,35	0,39	0,44	0,48
16	0,30	0,31	0,34	0,38	0,41
17	0,27	0,27	0,30	0,33	0,36
18	0,24	0,24	0,26	0,29	0,31

Tabel 7: Resultaten van de M/M/1 wachtrij met non-preemptive prioriteiten

² Geschat, zie pagina 11



Figuur 6: Wachttijd van de urgentiekleuren van de M/M/1 wachtrij met non-preemptive prioriteiten

Wat we goed kunnen zien is dat als we drie of meer behandelruimtes hebben, de wachttijd voor de laagste drie urgentiekleuren (geel, groen en blauw) binnen de maximaal voorgestelde wachttijd van het Manchester Triage System blijven. Voor de prioriteitsklasse “zeer urgent” is dit vanaf 4 ruimtes. De eis van 0 minuten wachttijd voor de rode kleur blijkt alleen haalbaar wanneer er enorm veel ruimtes zijn. We kunnen ook zien dat de wachttijd van laagste drie urgentiekleuren behoorlijk hard daalt naar mate het aantal ruimtes groter wordt, terwijl dat voor deze urgentiekleuren eigenlijk helemaal niet zo belangrijk is. Voor een klein aantal behandelruimtes voldoen we namelijk al aan de eisen van het Manchester Triage System.

Voor de M/M/c wachtrij met non-preemptive prioriteiten en gelijke behandelduur kunnen en voor de M/M/c wachtrij met non-preemptive prioriteiten en ongelijke behandelduur zien we ongeveer dezelfde resultaten. In tegenstelling tot de M/M/1 wachtrij met non-preemptive prioriteiten dalen de wachttijden echter nog harder. Wanneer er vijf behandelruimtes worden gebruikt zien we dat voor alle urgentiekleuren de wachttijd kleiner dan 2 minuten is. De bezettingsgraad van de complete wachtrij (ρ) is dan bij beide ook kleiner dan 50%. Verder maakt het voor de wachttijd niet heel veel uit of we nu een gelijke of een ongelijke behandelduur gebruiken. Dit komt waarschijnlijk ook doordat de patiënten überhaupt niet lang hoeven te wachten.

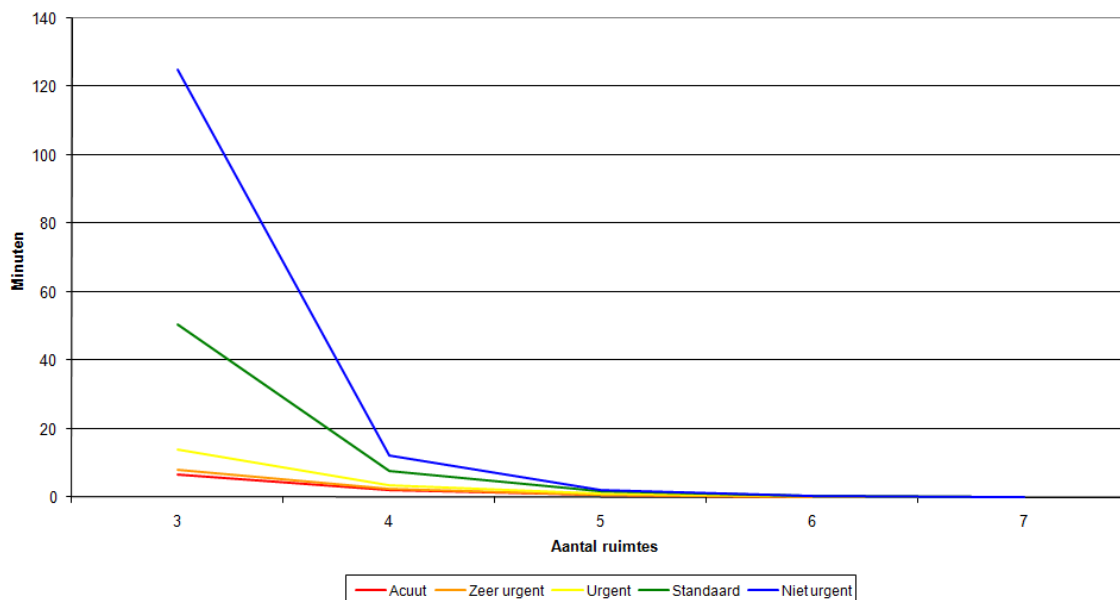
De resultaten van deze wachtrijmodellen zijn op de volgende pagina weergegeven.

De resultaten van de M/M/c wachtrij met non-preemptive prioriteiten en gelijke behandelduur:

# Behandelruimten	$E(W_{Q1})$	$E(W_{Q2})$	$E(W_{Q3})$	$E(W_{Q4})$	$E(W_{Q5})$
3	6,76	7,89	13,99	50,60	124,83
4	2,19	2,46	3,65	7,62	12,14
5	0,68	0,74	1,00	1,69	2,31
6	0,19	0,21	0,26	0,40	0,51
7	0,05	0,05	0,07	0,09	0,11

Tabel 8: Resultaten van de M/M/c wachtrij met non-preemptive prioriteiten en gelijke behandelduur

Wachttijd van de urgentiekleuren van de M/M/c wachtrij met non-preemptive prioriteiten en gelijke behandelduur

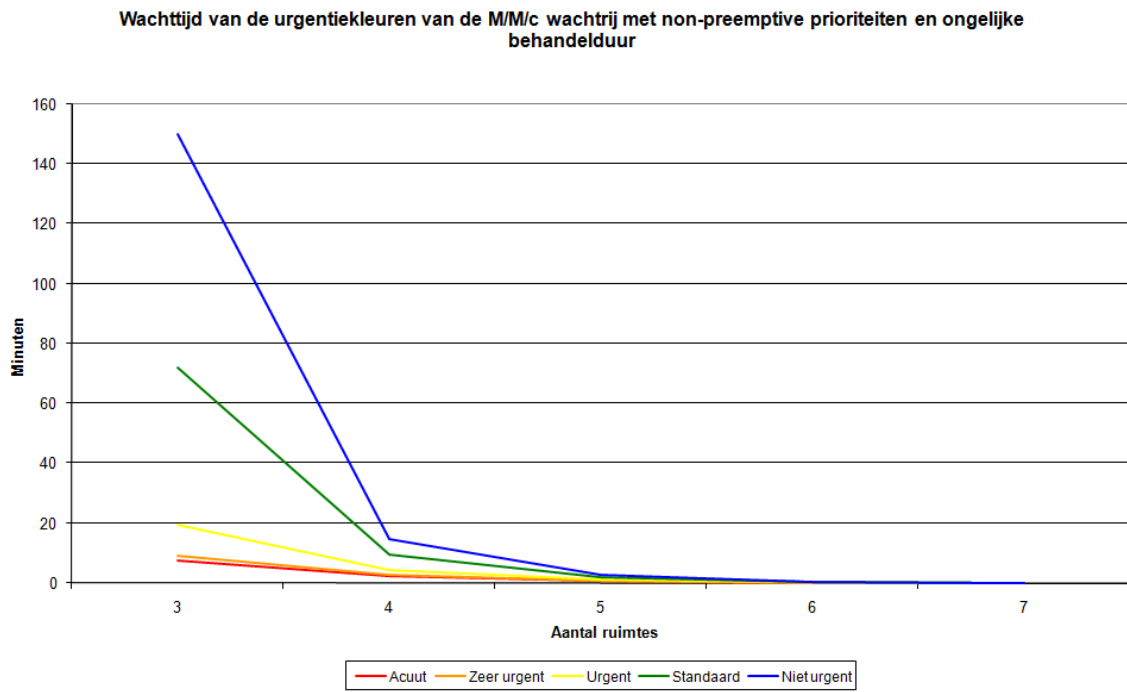


Figuur 7: Wachttijd van de urgentiekleuren van de M/M/c wachtrij met non-preemptive prioriteiten en gelijke behandelduur

De resultaten van de M/M/c wachtrij met non-preemptive prioriteiten en ongelijke behandelduur

# Behandelruimten	$E(W_{Q1})$	$E(W_{Q2})$	$E(W_{Q3})$	$E(W_{Q4})$	$E(W_{Q5})$
3	7,65	9,11	19,43	71,86	149,86
4	2,52	2,74	4,41	9,63	14,66
5	0,78	0,81	1,13	1,99	2,74
6	0,23	0,22	0,29	0,45	0,59
7	0,06	0,06	0,07	0,10	0,12

Tabel 9: Resultaten van de M/M/c wachtrij met non-preemptive prioriteiten en ongelijke behandelduur



Figuur 8: De wachttijd van de urgentiekleuren van de M/M/c wachtrij met non-preemptive prioriteiten en ongelijke behandelduur

Conclusie

Voor het onderzoek in dit werkstuk is het erg jammer dat het niet mogelijk was om een grote betrouwbare dataset te analyseren. Er wordt niet al te veel data bijgehouden en als dit wel het geval is, is de vraag hoe betrouwbaar deze data is.

Verder blijkt dat het best lastig is om de data vervolgens in een wachtrijmodel te gieten. Als we namelijk eens goed naar de resultaten kijken, zien we dat niet al te veel behandelruimtes nodig hebben om ervoor te zorgen dat de wachttijd van overgrote deel van de patiënten groter dan enkele minuten te laten worden. In de praktijk blijkt echter dat mensen vele malen langer wachten. Hier kunnen we een aantal redenen voor bedenken:

Ten eerste: We gaan er nu vanuit dat als een behandelruimte leeg is, de behandeling meteen kan beginnen. In de praktijk kan het nog als eens voorkomen dat er eerst een specialist moet worden opgeroepen voordat men met de behandeling van de patiënt begint. Hierdoor moet de patiënt nog wat langer wachten voordat hij geholpen kan worden. Ook moeten mensen vaak wachten op het maken van een röntgenfoto of voor het maken van scans. Het zou kunnen zijn dat deze tijd in de behandelruimte is opgenomen, maar dat is niet duidelijk uit de data te halen.

Ten tweede: De verdeling over de dag. Nu hebben we aangenomen dat het aantal patiënten dat op spoedeisende hulp binnenkomt over de gehele dag gelijk is. In de praktijk is dit natuurlijk anders. Overdag gebeuren er meer ongevallen en zal het op de spoedeisende hulp ook drukker zijn. Voor het verloop van de wachttijd voor de verschillende urgentiekleuren maakt dit echter niet veel uit. De grafieken zullen dezelfde vorm hebben, alleen zal het aantal behandelruimtes groter moeten zijn.

Ten derde: De verdeling van het aantal patiënten over het jaar. Ook hier hebben we aangenomen dat de verdeling over de dagen van het jaar gelijk zijn. Het is echter goed voor te stellen dat er in het weekend meer mensen op de spoedeisende hulp komen dan op werkdagen. Ook kunnen er bepaalde piekdagen/piekmomenten zijn, bijvoorbeeld met oud en nieuw als er veel ongelukken met vuurwerk gebeuren.

Wat we verder aan de resultaten kunnen zien is dat de gemiddelde wachttijd van een patiënt van de drie laagste prioriteitsklassen (geel, groen en blauw) eigenlijk veel harder daalt dan we eigenlijk zouden willen. We zien liever dat de wachttijd van de eerste twee prioriteitsklassen (rood en oranje) sterk afneemt naar mate er meer behandelruimtes gebruikt worden. Dit omdat deze mensen over het algemeen zeer zware verwondingen hebben. Wat we hiermee kunnen concluderen is dat triage dus alleen zin heeft op het moment dat de bezettingsgraad groot is. Wanneer de bezettingsgraad namelijk laag is, hoeft eigenlijk niemand lang te wachten en is het eenvoudiger en net zo effectief om het “see and treat” principe aan te houden: het direct behandelen van een aandoening.

Mogelijkheden tot vervolgonderzoek

Naast het onderzoek dat in dit werkstuk naar voren is gekomen, zijn er nog enkele mogelijkheden om vervolgonderzoek te doen.

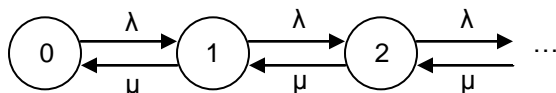
Zoals in de conclusie al naar voren is gekomen zou het interessant zijn om te kijken naar de verdeling van het aantal patiënten over de dag en de verdeling van het aantal patiënten over de dagen in het jaar. Voornamelijk voor het inplannen van het personeel zou dit heel handig kunnen zijn. Het is dan beter te voorspellen hoeveel behandelruimtes nodig zijn, wanneer je de patiënten gemiddeld niet te lang wilt laten wachten.

Waar in dit werkstuk ook geen aandacht aan is besteed is de kans dat een patiënt toch langer moet wachten dan de voorgeschreven maximale tijd die door het Manchester Triage System is voorgeschreven. De gemiddelde tijd kan namelijk wel onder de voorgeschreven tijd liggen, maar het is dan alsnog goed mogelijk dat er patiënten bij zitten die weldegelijk langer moeten wachten dan de voorgeschreven maximale wachttijd.

Appendix A

De M/M/1 wachtrij

Bij het M/M/1 wachtrijmodel gaan we er dus vanuit dat de zowel de tussenaankomsttijden (met intensiteit λ) als de behandeltijden (met intensiteit μ) exponentieel verdeeld zijn en dat we 1 behandelruimte tot onze beschikking hebben. Voor dit wachtrijmodel kunnen we het onderstaande toestandsdiagram voor het aantal patiënten in de wachtrij maken.



Figuur 9: Toestandsdiagram van het M/M/1 wachtrijmodel

Met behulp van dit toestandsdiagram kunnen we de evenwichtsverdeling berekenen. Deze kunnen we berekenen door de stroom die uit een bepaalde toestand gaat, gelijk te stellen aan de stroom die bij dezelfde toestand binnenkomt. Op deze manier krijgen we de volgende vergelijkingen:

$$\begin{aligned}\lambda \pi(0) &= \mu \pi(1) \\ (\lambda + \mu) \pi(1) &= \lambda \pi(0) + \mu \pi(2) \\ (\lambda + \mu) \pi(2) &= \lambda \pi(1) + \mu \pi(3) \\ &\dots \\ (\lambda + \mu) \pi(x) &= \lambda \pi(x - 1) + \mu \pi(x + 1) \\ &\dots\end{aligned}$$

Deze vergelijkingen kunnen recursief uitgedrukt worden in $\pi(0)$:

$$\pi(x) = \left(\frac{\lambda}{\mu}\right)^x \pi(0) = \rho^x \pi(0); \text{ hierbij geldt dat } \rho = \frac{\lambda}{\mu} \text{ en dat } \rho < 1$$

Omdat $\pi(x)$ kansen zijn, moeten deze optellen tot 1:

$$\sum_{x=0}^{\infty} \pi(x) = 1$$

Hieruit kunnen we afleiden dat:

$$\sum_{x=0}^{\infty} \rho^x \pi(0) = \pi(0) \sum_{x=0}^{\infty} \rho^x = \pi(0) \frac{1}{1-\rho} = 1, \text{ dus } \pi(0) = 1-\rho \text{ en } \pi(x) = \rho^x (1-\rho)$$

Nu we de kansverdeling van het aantal patiënten in het systeem weten, kunnen we het gemiddeld aantal patiënten in het systeem berekenen door middel van:

$$E(L) = \sum_{x=0}^{\infty} x \pi(x) = \pi(0) \sum_{x=0}^{\infty} x \rho^x = (1-\rho) \frac{\rho}{(1-\rho)^2} = \frac{\rho}{1-\rho}$$

Omdat we weten dat $E(L) = E(L_Q) + \rho$, kunnen we $E(L_Q)$ dus ook heel gemakkelijk berekenen:

$$E(L_Q) = E(L) - \rho = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}$$

Met behulp van de formules van Little kunnen we nu ook de gemiddelde tijd dat mensen zich in het systeem bevinden en de gemiddelde tijd dat ze in de wachtkamer moeten plaatsnemen uitrekenen.

$$E(W) = \frac{E(L)}{\lambda} = \frac{\rho}{1-\rho} \cdot \frac{1}{\lambda} = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu - \lambda}$$

$$E(W_Q) = E(W) - \frac{1}{\mu} = \frac{1}{\mu(1-\rho)} - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}$$

We kunnen de verdeling van de wachttijd in de wachtkamer als volgt berekenen:

$$P(W_Q > t | L = j+1) = P(\text{minderdan } j+1 \text{ vertrekken in } t) = \sum_{k=0}^j \frac{(t\mu)^k}{k!} e^{-t\mu}$$

$$\begin{aligned} P(W_Q > t) &= \sum_{j=0}^{\infty} P(W_Q > t | L = j+1) P(L = j+1) \\ &= \sum_{j=0}^{\infty} \sum_{k=0}^j \frac{(t\mu)^k}{k!} e^{-t\mu} (1-\rho) \rho^{j+1} \\ &= \sum_{k=0}^{\infty} \sum_{j=k}^{\infty} \frac{(t\mu)^k}{k!} e^{-t\mu} (1-\rho) \rho^{j+1} \\ &= \sum_{k=0}^{\infty} \frac{(t\mu)^k}{k!} e^{-t\mu} (1-\rho) \sum_{j=k}^{\infty} \rho^{j+1} = \sum_{k=0}^{\infty} \frac{(t\mu)^k}{k!} e^{-t\mu} (1-\rho) \rho^{k+1} \frac{1}{1-\rho} \\ &= \rho e^{-t\mu} \sum_{k=0}^{\infty} \frac{(t\mu)^k}{k!} = \rho e^{-t\mu} e^{t\mu} = \rho e^{-t\mu(1-\rho)} \end{aligned}$$

Over de verdeling van de tijd in het systeem kunnen we het volgende zeggen: als een aankomende patiënt k patiënten voor hem in het systeem vindt, dan bestaat de totale tijd in het systeem uit $k+1$ onafhankelijke exponentiele behandel tijden. De som van $k+1$ onafhankelijke en identiek verdeelde behandel tijden zijn hetzelfde als een Erlang verdeling met parameters $k+1$ en μ , met dichtheidsfunctie:

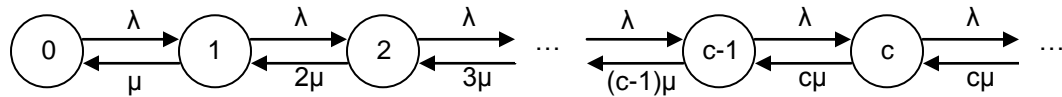
$$f_{k+1}(t) = \mu \frac{(\mu t)^k}{k!} e^{-\mu t}$$

Vanwege de PASTA eigenschap weten we dat de kans dat een aankomende patiënt k andere patiënten in het systeem tegenkomt gelijk is aan $\pi(k)$. De totale dichtheid wordt hierdoor:

$$f(t) = \sum_{k=0}^{\infty} \pi(k) f_{k+1}(t) = \sum_{k=0}^{\infty} (1-\rho) \rho^k \mu \frac{(\mu t)^k}{k!} e^{-\mu t} = \mu(1-\rho) e^{-\mu(1-\rho)t}$$

De M/M/c wachtrij

De M/M/c wachtrij heeft dezelfde verdelingen van de tussenaankomsttijden en behandeltijden als bij de M/M/1 wachtrij, alleen hebben we in dit geval c ($c > 1$) behandelruimtes tot onze beschikking. Ook voor dit wachtrijmodel kunnen we weer een toestandsdiagram van het aantal patiënten in het systeem maken. Deze staat hieronder weergegeven.



Figuur 10: Toestandsdiagram van het M/M/c wachtrijmodel

Nu gaan we op dezelfde manier als bij de M/M/1 wachtrij de evenwichtsvergelijking opstellen. De vergelijkingen die we nu krijgen als we de stroom die uit een bepaalde toestand gelijk stellen aan de stroom die bij dezelfde toestand binnenkomt zijn:

$$\begin{aligned} \lambda \pi(0) &= \mu \pi(1) \\ (\lambda + \mu) \pi(1) &= \lambda \pi(0) + 2\mu \pi(2) \\ (\lambda + 2\mu) \pi(2) &= \lambda \pi(1) + 3\mu \pi(3) \\ (\lambda + 3\mu) \pi(3) &= \lambda \pi(2) + 4\mu \pi(4) \\ &\dots \\ (\lambda + (c-1)\mu) \pi(c-1) &= \lambda \pi(c-2) + c\mu \pi(c) \\ (\lambda + c\mu) \pi(c) &= \lambda \pi(c-1) + c\mu \pi(c+1) \\ (\lambda + c\mu) \pi(c+1) &= \lambda \pi(c) + c\mu \pi(c+2) \\ &\dots \end{aligned}$$

Als we deze vergelijking weer uitdrukken in $\pi(0)$ krijgen we:

$$\pi(x) = \frac{1}{x!} \left(\frac{\lambda}{\mu} \right)^x \pi(0) \text{ voor het geval dat } 0 \leq x \leq c$$

$$\text{en } \pi(x) = \frac{1}{c! c^{x-c}} \left(\frac{\lambda}{\mu} \right)^x \pi(0) \text{ voor het geval dat } x > c$$

Nu moeten alle kansen weer optellen tot 1:

$$\begin{aligned} \sum_{x=0}^{c-1} \frac{1}{x!} \left(\frac{\lambda}{\mu} \right)^x \pi(0) + \sum_{x=c}^{\infty} \frac{1}{c! c^{x-c}} \left(\frac{\lambda}{\mu} \right)^x \pi(0) &= 1 \\ \frac{1}{\pi(0)} &= \sum_{x=0}^{c-1} \frac{1}{x!} \left(\frac{\lambda}{\mu} \right)^x + \sum_{x=c}^{\infty} \frac{1}{c! c^{x-c}} \left(\frac{\lambda}{\mu} \right)^x \end{aligned}$$

Hiervan kunnen we het laatste deel nog omschrijven:

$$\sum_{x=c}^{\infty} \frac{1}{c! c^{x-c}} \left(\frac{\lambda}{\mu} \right)^x = \left(\frac{\lambda}{\mu} \right)^c \frac{1}{c!} \sum_{x=c}^{\infty} \frac{1}{c^{x-c}} \left(\frac{\lambda}{\mu} \right)^{x-c}$$

Nu voeren we in dat $k = x - c$ en $c\rho = \frac{\lambda}{\mu}$:

$$\left(\frac{\lambda}{\mu}\right)^c \frac{1}{c!} \sum_{x=c}^{\infty} \frac{1}{c^{x-c}} \left(\frac{\lambda}{\mu}\right)^{x-c} = \frac{(c\rho)^c}{c!} \sum_{k=0}^{\infty} \frac{(c\rho)^k}{c^k} = \frac{(c\rho)^c}{c!} \sum_{k=0}^{\infty} \rho^k = \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} = \frac{(c\rho)^c}{c!(1-\rho)}$$

Uiteindelijk kunnen we nu afleiden dat:

$$\pi(0) = \left[\sum_{x=0}^{c-1} \frac{(c\rho)^x}{x!} + \frac{(c\rho)^c}{c!(1-\rho)} \right]^{-1}$$

De kans dat we met dit wachtrijmodel moeten wachten wordt in de literatuur ook wel $C(c, \rho)$ genoemd. Deze kans kunnen we als volgt berekenen:

$$C(c, \rho) = \pi(c) + \pi(c+1) + \pi(c+2) + \dots$$

$$\begin{aligned} C(c, \rho) &= \sum_{x=c}^{\infty} \pi(x) = \sum_{x=c}^{\infty} \frac{1}{c! c^{x-c}} \left(\frac{\lambda}{\mu}\right)^x \pi(0) = \sum_{x=c}^{\infty} \frac{(c\rho)^x}{c! c^{x-c}} \pi(0) \\ &= \sum_{x=c}^{\infty} \frac{(c\rho)^x}{c! c^{x-c}} \left[\sum_{x=0}^{c-1} \frac{(c\rho)^x}{x!} + \frac{(c\rho)^c}{c!(1-\rho)} \right]^{-1} = \frac{(c\rho)^c}{c!(1-\rho)} \left[\sum_{x=0}^{c-1} \frac{(c\rho)^x}{x!} + \frac{(c\rho)^c}{c!(1-\rho)} \right]^{-1} \end{aligned}$$

$E(W_Q)$ en $E(L_Q)$ kunnen we berekenen aan de hand van de mean value analyse en de formule van Little. Als niet alle behandelruimtes bezet zijn, dan is de wachttijd van een binnenkomende patiënt gelijk aan nul. Als alle behandelruimtes wel bezet zijn en er zijn nul of meer patiënten aan het wachten in de wachtkamer, dan moet een aankomende patiënt eerst wachten totdat een van de behandelkamers vrij komt en dan moet hij dit nog net zo vaak doen als dat er patiënten in de wachtkamer zaten. De tijd tussen twee opeenvolgende vertrekken is het minimum van c exponentiële (residuele) behandel tijden met een gemiddelde van $1/\mu$ en dus is dit exponentieel met gemiddelde $1/c\mu$. Hierdoor kunnen we het volgende opstellen:

$$E(W_Q) = C(c, a) \frac{1}{c\mu} + E(L_Q) \frac{1}{c\mu}$$

Als we nu de formule van Little gebruiken krijgen we het volgende:

$$E(W_Q) = C(c, a) \frac{1}{c\mu} + \lambda E(W_Q) \frac{1}{c\mu}$$

$$E(W_Q) - \lambda E(W_Q) \frac{1}{c\mu} = C(c, a) \frac{1}{c\mu}$$

$$E(W_Q)(1 - \rho) = C(c, a) \frac{1}{c\mu}$$

$$E(W_Q) = C(c, a) \frac{1}{c\mu(1-\rho)} = \frac{C(c, a)}{c\mu - \lambda}$$

Voor $E(L_Q)$ kunnen we een soortgelijke berekening maken. Voor $E(L_Q)$ krijgen we dan de volgende uitdrukking:

$$E(L_Q) = \frac{\rho C(c, a)}{1-\rho}$$

Door deze twee resultaten krijgen we voor $E(W)$ en $E(L)$ de volgende uitdrukkingen:

$$E(W) = \frac{C(c, a)}{c\mu - \lambda} + \frac{1}{\mu} \qquad E(L) = \frac{\rho C(c, a)}{1-\rho} + \frac{\lambda}{\mu}$$

Appendix B

Dit is het simulatieprogramma dat we gebruiken hebben voor de M/M/c wachtrij met non-preemptive prioriteiten met ongelijke behandeltijd.

```
#include <stdlib.h>
#include <math.h>
#include <iostream.h>

const int c =9;
const float labda1 =0.048557123;
const float labda2 =0.725426678;
const float labda3 =1.613268562;
const float labda4 =1.742614692;
const float labda5 =0.056091849;

const float mu1 =0.75;
const float mu2 =1.153846;
const float mu3 =1.714286;
const float mu4 =2.608696;
const float mu5 =3.157895;

const int aankomsten = 250000; //het aantal aankomsten per herhaling
const float uur =60;
int SEED = 12345;
const int HERHALINGEN = 100;

float EWQrij[HERHALINGEN];
float EWQrij1[HERHALINGEN];
float EWQrij2[HERHALINGEN];
float EWQrij3[HERHALINGEN];
float EWQrij4[HERHALINGEN];
float EWQrij5[HERHALINGEN];

float ELQrij[HERHALINGEN];
float ELQrij1[HERHALINGEN];
float ELQrij2[HERHALINGEN];
float ELQrij3[HERHALINGEN];
float ELQrij4[HERHALINGEN];
float ELQrij5[HERHALINGEN];

float rhorij[HERHALINGEN];
float rhorij1[HERHALINGEN];
float rhorij2[HERHALINGEN];
float rhorij3[HERHALINGEN];
float rhorij4[HERHALINGEN];
float rhorij5[HERHALINGEN];

enum kind {arr,dep};

struct event_node {
    float time;
    kind type;
    int prioriteit;
    event_node * next;
};

struct wachtRij_node{
    float time;
    float service;
    wachtRij_node * next;
};
```

```

// classes
class WachtRij {
    wachtRij_node * first;
    wachtRij_node * last;
public:
    WachtRij() {first = NULL; last = NULL;};
    ~WachtRij();
    void enwachtRij(float, float);
    bool serve(float&, float&);
};

WachtRij::~WachtRij() {
    wachtRij_node *p;
    while (first != NULL) {
        p = first;
        first=first->next;
        delete(p);
    }
};

void WachtRij::enwachtRij(float t, float s) {
    wachtRij_node * nieuwe_node = new wachtRij_node;
    if (nieuwe_node == NULL){
        cout << "out of memory\n";
        exit(1);
    };
    nieuwe_node->time =t;
    nieuwe_node->service =s;
    nieuwe_node->next =NULL;

    if (first == NULL){ //eerste klant in de rij
        first = nieuwe_node;
        last = nieuwe_node;
    }
    else {
        last->next = nieuwe_node;
        last = nieuwe_node;
    }
};

bool WachtRij::serve(float& t, float& s) {
    if (first ==NULL) return false; //rij leeg
    else {
        t = first->time;
        s = first->service;
        wachtRij_node * p = first;
        first = first->next;
        delete p;
        return true;
    }
};

class EventList {
    event_node * first;
public:
    EventList() {first = NULL;}
    ~EventList();
    void schedule_event(float, kind, int);
    bool get_next_event(float&, kind&, int&);
};

EventList::~EventList() {
    event_node *p;
    while (first != NULL) {
        p = first;
        first=first->next;
        delete(p);
    }
};

```

```

void EventList::schedule_event(float t, kind soort, int p) {
    event_node * nieuwe_node = new event_node;
    if (nieuwe_node == NULL){
        cout << "out of memory\n";
        exit(1);
    };
    nieuwe_node->time =t;
    nieuwe_node->type =soort;
    nieuwe_node->prioriteit =p;
    if (first == NULL){ //lege lijst
        first = nieuwe_node;
        nieuwe_node->next =NULL;
    }
    else if (t <= first->time){ //toevoegen aan begin
        nieuwe_node->next = first;
        first = nieuwe_node;
    }
    else {
        event_node * p = first;
        event_node * q;
        while (t > p->time) { // op juiste plaats tussenvoegen
            q = p;
            p = p->next;
            if (p == NULL) break; // toevoegen aan einde
        }
        q->next = nieuwe_node;
        nieuwe_node->next =p;
    }
};

bool EventList::get_next_event(float& t, kind& soort,int& p) {
    if (first == NULL) return false; //lijst leeg
    else {
        t = first->time;
        soort = first->type;
        p = first->prioriteit;
        event_node * p = first;
        first = first->next;
        delete p;
        return true;
    }
};

//globale variabelen
float simtime=0;
int bezet =0; //aantal behandelruimtes bezet
int bezet1=0; //aantal behandelruimtes door type1 bezet
int bezet2=0; //aantal behandelruimtes door type2 bezet
int bezet3=0; //aantal behandelruimtes door type3 bezet
int bezet4=0; //aantal behandelruimtes door type4 bezet
int bezet5=0; //aantal behandelruimtes door type5 bezet

int rijlengte = 0;
int rijlengte1 = 0;
int rijlengte2 = 0;
int rijlengte3 = 0;
int rijlengte4 = 0;
int rijlengte5 = 0;

int p;
float tijd;
kind soort;
EventList evt_list;
WachtRij type1; //heeft prioriteit boven alle anderen
WachtRij type2;
WachtRij type3;
WachtRij type4;
WachtRij type5; //de laagste prioriteit

```

```

int n =0; //totaal aantal klanten
int n1=0; //totaal aantal klanten type1
int n2=0; //totaal aantal klanten type2
int n3=0; //totaal aantal klanten type3
int n4=0; //totaal aantal klanten type4
int n5=0; //totaal aantal klanten type5

float EWqueue =0; //totale wachttijd in rij van alle klanten
float EWqueue1 = 0; //totale wachttijd in rij van klanten type1
float EWqueue2 = 0; //totale wachttijd in rij van klanten type2
float EWqueue3 = 0; //totale wachttijd in rij van klanten type3
float EWqueue4 = 0; //totale wachttijd in rij van klanten type4
float EWqueue5 = 0; //totale wachttijd in rij van klanten type5

float ELqueue=0; //schatteer gemiddelde rijlengte
float ELqueue1=0; //schatteer gemiddelde rijlengte type1
float ELqueue2=0; //schatteer gemiddelde rijlengte type2
float ELqueue3=0; //schatteer gemiddelde rijlengte type3
float ELqueue4=0; //schatteer gemiddelde rijlengte type4
float ELqueue5=0; //schatteer gemiddelde rijlengte type5

float util_grd=0; //schatteer utilisatiegraad
float util_grd1=0; //schatteer utilisatiegraad type1
float util_grd2=0; //schatteer utilisatiegraad type2
float util_grd3=0; //schatteer utilisatiegraad type3
float util_grd4=0; //schatteer utilisatiegraad type4
float util_grd5=0; //schatteer utilisatiegraad type5

//functies
float draw_exp(float l){//random trekking uit exponentiele verdeling
    long int dummy;
    dummy =rand()+1; //willekeurig uit 1..RAND_MAX+1
    return -(1/l)*log((float)dummy/(RAND_MAX+2));
};

void initialize() {
    float a,b,c,d,e; // eerste aankomsttijd type1, type2, type3, type4, type5
    srand(SEED); // initialiseren van de random generator
    a = draw_exp(labda1/uur);
    b = draw_exp(labda2/uur);
    c = draw_exp(labda3/uur);
    d = draw_exp(labda4/uur);
    e = draw_exp(labda5/uur);
    evt_list.schedule_event(a,arr,1);
    n1++;
    n++;
    evt_list.schedule_event(b,arr,2);
    n2++;
    n++;
    evt_list.schedule_event(c,arr,3);
    n3++;
    n++;
    evt_list.schedule_event(d,arr,4);
    n4++;
    n++;
    evt_list.schedule_event(e,arr,5);
    n5++;
    n++;
};

void handle_arr(int p) {
    float a,s; //tussenaankomsttijd, behandeltijd
    switch(p){
        case 1:
            a = draw_exp(labda1/uur);
            s= draw_exp(mu1/uur);
            break;
        case 2:
            a = draw_exp(labda2/uur);
            s= draw_exp(mu2/uur);

```

```

        break;
    case 3:
        a = draw_exp(labda3/uur);
        s= draw_exp(mu3/uur);
        break;
    case 4:
        a = draw_exp(labda4/uur);
        s= draw_exp(mu4/uur);
        break;
    case 5:
        a = draw_exp(labda5/uur);
        s= draw_exp(mu5/uur);
        break;
};
if (aankomsten > n ) { // er worden nog patienten toegelaten
    evt_list.schedule_event(simtime+a,arr,p);
    switch(p){
        case 1:n1++; break;
        case 2:n2++; break;
        case 3:n3++; break;
        case 4:n4++; break;
        case 5:n5++; break;
    };
    n++;
};

if (bezet < c) { // er zijn nog wachtruimtes vrij
    switch(p){
        case 1: bezet1++; break;
        case 2: bezet2++; break;
        case 3: bezet3++; break;
        case 4: bezet4++; break;
        case 5: bezet5++; break;
    };
    bezet++;
    evt_list.schedule_event(simtime+s,dep,p);
} else { //in de wachtrij gaan staan
    switch(p){
        case 1:
            type1.enwachtRij(simtime,s);
            rijlengte1++;
            break;
        case 2:
            type2.enwachtRij(simtime,s);
            rijlengte2++;
            break;
        case 3:
            type3.enwachtRij(simtime,s);
            rijlengte3++;
            break;
        case 4:
            type4.enwachtRij(simtime,s);
            rijlengte4++;
            break;
        case 5:
            type5.enwachtRij(simtime,s);
            rijlengte5++;
            break;
    };
    rijlengte++;
};
};

void handle_dep(int p) {
    float arr_time, s;
    if (type1.serve(arr_time, s)) { //iemand van hoogste prioriteit
        rijlengte--;
        rijlengte1--;
        EWqueue += (simtime - arr_time);
        EWqueue1 += (simtime- arr_time);
    }
}

```

```

    evt_list.schedule_event(simtime+s,dep,1);

    switch(p){
        case 1: break;
        case 2: bezet2--; bezet1++; break;
        case 3: bezet3--; bezet1++; break;
        case 4: bezet4--; bezet1++; break;
        case 5: bezet5--; bezet1++; break;
    };
}
else if (type2.serve(arr_time, s)) { //iemand van prioriteit type 2
    rijlengte--;
    rijlengte2--;
    EWqueue+=(simtime - arr_time);
    EWqueue2+=(simtime-arr_time);
    evt_list.schedule_event(simtime+s,dep,2);
    switch(p){
        case 1:bezet1--; bezet2++; break;
        case 2:break;
        case 3:bezet3--; bezet2++; break;
        case 4:bezet4--; bezet2++; break;
        case 5:bezet5--; bezet2++; break;
    };
}
else if (type3.serve(arr_time, s)) { //iemand van prioriteit type 3
    rijlengte--;
    rijlengte3--;
    EWqueue+=(simtime - arr_time);
    EWqueue3+=(simtime-arr_time);
    evt_list.schedule_event(simtime+s,dep,3);
    switch(p){
        case 1:bezet1--; bezet3++; break;
        case 2:bezet2--; bezet3++; break;
        case 3:break;
        case 4:bezet4--; bezet3++; break;
        case 5:bezet5--; bezet3++; break;
    };
}
else if (type4.serve(arr_time, s)) { //iemand van prioriteit type 4
    rijlengte--;
    rijlengte4--;
    EWqueue+=(simtime - arr_time);
    EWqueue4+=(simtime-arr_time);
    evt_list.schedule_event(simtime+s,dep,4);
    switch(p){
        case 1:bezet1--; bezet4++; break;
        case 2:bezet2--; bezet4++; break;
        case 3:bezet3--; bezet4++; break;
        case 4:break;
        case 5:bezet5--; bezet4++; break;
    };
}
else if (type5.serve(arr_time, s)) { //iemand van prioriteit type 5
    rijlengte--;
    rijlengte5--;
    EWqueue+=(simtime - arr_time);
    EWqueue5+=(simtime-arr_time);
    evt_list.schedule_event(simtime+s,dep,5);
    switch(p){
        case 1:bezet1--; bezet5++; break;
        case 2:bezet2--; bezet5++; break;
        case 3:bezet3--; bezet5++; break;
        case 4:bezet4--; bezet5++; break;
        case 5:break;
    };
}
else { //rijen leeg
    switch(p){
        case 1:bezet1--; break;
        case 2:bezet2--; break;
        case 3:bezet3--; break;
        case 4:bezet4--; break;
        case 5:bezet5--; break;
    };
}
bezet--;

```

```

};

};

void handle_event(float time, kind soort, int p){
    ELqueue += (tijd-simtime)*rijlengte;
    util_grd += (tijd - simtime)*bezet;
    ELqueue1 += (tijd-simtime)*rijlengte1;
    util_grd1 += (tijd-simtime)*bezet1;
    ELqueue2 += (tijd-simtime)*rijlengte2;
    util_grd2 += (tijd-simtime)*bezet2;
    ELqueue3 += (tijd-simtime)*rijlengte3;
    util_grd3 += (tijd-simtime)*bezet3;
    ELqueue4 += (tijd-simtime)*rijlengte4;
    util_grd4 += (tijd-simtime)*bezet4;
    ELqueue5 += (tijd-simtime)*rijlengte5;
    util_grd5 += (tijd-simtime)*bezet5;
    simtime = tijd;
    switch(soort){
        case arr:handle_arr(p); break;
        case dep:handle_dep(p); break;
    };
};

void report() {
    float EWQ = 0;
    float EWQ1 = 0;
    float EWQ2 = 0;
    float EWQ3 = 0;
    float EWQ4 = 0;
    float EWQ5 = 0;
    float ELQ =0;
    float ELQ1 =0;
    float ELQ2 =0;
    float ELQ3 =0;
    float ELQ4 =0;
    float ELQ5 =0;
    float rho =0;
    float EW=0;
    float EW1=0;
    float EW2=0;
    float EW3=0;
    float EW4=0;
    float EW5=0;
    float EL=0;
    float EL1=0;
    float EL2=0;
    float EL3=0;
    float EL4=0;
    float EL5=0;
    float labda=labda1+labda2+labda3+labda4+labda5;
    float overalServiceRate = 1/(labda1/(labda*mu1)+labda2/(labda*mu2)+
labda3/(labda*mu3)+labda4/(labda*mu4)+labda5/(labda*mu5));

    for(int i = 0; i < HERHALINGEN; i++) {
        EWQ += EWQrij[i];
        ELQ += ELQrij[i];
        EWQ1 += EWQrij1[i];
        ELQ1 += ELQrij1[i];
        EWQ2 += EWQrij2[i];
        ELQ2 += ELQrij2[i];
        EWQ3 += EWQrij3[i];
        ELQ3 += ELQrij3[i];
        EWQ4 += EWQrij4[i];
        ELQ4 += ELQrij4[i];
        EWQ5 += EWQrij5[i];
        ELQ5 += ELQrij5[i];
        rho += rhorij[i];
        EL = EL + rhorij[i]*c + ELQrij[i];
        EL1 = EL1 + rhorij1[i]*c + ELQrij1[i];
        EL2 = EL2 + rhorij2[i]*c + ELQrij2[i];

```

```

        EL3 = EL3 + rhorij3[i]*c + ELQrij3[i];
        EL4 = EL4 + rhorij4[i]*c + ELQrij4[i];
        EL5 = EL5 + rhorij5[i]*c + ELQrij5[i];
    }
    //gemiddelden berekenen
    EWQ /= HERHALINGEN;
    ELQ /=HERHALINGEN;
    rho /=HERHALINGEN;
    EW = EWQ + overalServiceRate*60;
    EL /= HERHALINGEN;
    EWQ1 /= HERHALINGEN;
    ELQ1 /=HERHALINGEN;
    EW1 = EWQ1 + (1/mu1)*60;
    EL1 /= HERHALINGEN;
    EWQ2 /= HERHALINGEN;
    ELQ2 /=HERHALINGEN;
    EW2 = EWQ2 + (1/mu2)*60;
    EL2 /= HERHALINGEN;
    EWQ3 /= HERHALINGEN;
    ELQ3 /=HERHALINGEN;
    EW3 = EWQ3 + (1/mu3)*60;
    EL3 /= HERHALINGEN;
    EWQ4 /= HERHALINGEN;
    ELQ4 /=HERHALINGEN;
    EW4 = EWQ4 + (1/mu4)*60;
    EL4 /= HERHALINGEN;
    EWQ5 /= HERHALINGEN;
    ELQ5 /=HERHALINGEN;
    EW5 = EWQ5 + (1/mu5)*60;
    EL5 /= HERHALINGEN;

    cout << "Simulatie M/M/c wachtrij" << endl;
    cout << "======" << endl;
    cout << endl;
    cout << "c" : " << c << endl;
    cout << "lambda1" : " << labda1 << " per uur" << endl;
    cout << "mu1" : " << mu1 << " per uur" << endl;
    cout << "lambda2" : " << labda2 << " per uur" << endl;
    cout << "mu2" : " << mu2 << " per uur" << endl;
    cout << "lambda3" : " << labda3 << " per uur" << endl;
    cout << "mu3" : " << mu3 << " per uur" << endl;
    cout << "lambda4" : " << labda4 << " per uur" << endl;
    cout << "mu4" : " << mu4 << " per uur" << endl;
    cout << "lambda5" : " << labda5 << " per uur" << endl;
    cout << "mu5" : " << mu5 << " per uur" << endl;
    cout << "overall service rate" : " << overalServiceRate << " per uur"
<<endl;
    cout << "aantal aankomsten per iteratie" : " << aankomsten << endl;
    cout << "aantal iteraties" : " << HERHALINGEN << endl;
    cout << endl << endl;
    cout << "Resultaten complete wachtrij:" << endl << endl;
    cout << "Utilisatiegraad (rho)" : " << rho*100 <<"%" << endl;
    cout << "Gemiddeld # klanten in systeem (EL)" : " << EL << endl;
    cout << "Gemiddelde rijlengte (ELQ)" : " << ELQ << endl;
    cout << "Gemiddelde tijd in systeem (EW)" : " << EW;
    if (EW < 2.0) cout << " minuut" << endl;
    else
        cout << " minuten"<< endl;
        cout << "Gemiddelde wachttijd in rij (EWQ)" : " << EWQ;
    if (EWQ < 2.0) cout << " minuut" << endl;
    else
        cout << " minuten"<< endl << endl;
        cout << "Resultaten prioriteitsklasse 1" << endl << endl;
        cout << "Gemiddeld # klanten in systeem (EL)" : " << EL1 << endl;
        cout << "Gemiddelde rijlengte (ELQ)" : " << ELQ1 << endl;
        cout << "Gemiddelde tijd in systeem (EW)" : " << EW1;
    if (EW1 < 2.0) cout << " minuut" << endl;
    else
        cout << " minuten"<< endl;
        cout << "Gemiddelde wachttijd in rij (EWQ)" : " << EWQ1;

```

```

if (EWQ1 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl << endl;
    cout << "Resultaten prioriteitsklasse 2" << endl << endl;
    cout << "Gemiddeld # klanten in systeem (EL) : " << EL2 << endl;
    cout << "Gemiddelde rijlengte (ELQ) : " << ELQ2 << endl;
    cout << "Gemiddelde tijd in systeem (EW) : " << EW2;
if (EW1 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl;
    cout << "Gemiddelde wachttijd in rij (EWQ) : " << EWQ2;
if (EWQ2 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl << endl;
    cout << "Resultaten prioriteitsklasse 3" << endl << endl;
    cout << "Gemiddeld # klanten in systeem (EL) : " << EL3 << endl;
    cout << "Gemiddelde rijlengte (ELQ) : " << ELQ3 << endl;
    cout << "Gemiddelde tijd in systeem (EW) : " << EW3;
if (EW1 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl;
    cout << "Gemiddelde wachttijd in rij (EWQ) : " << EWQ3;
if (EWQ3 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl << endl;
    cout << "Resultaten prioriteitsklasse 4" << endl << endl;
    cout << "Gemiddeld # klanten in systeem (EL) : " << EL4 << endl;
    cout << "Gemiddelde rijlengte (ELQ) : " << ELQ4 << endl;
    cout << "Gemiddelde tijd in systeem (EW) : " << EW4;
if (EW1 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl;
    cout << "Gemiddelde wachttijd in rij (EWQ) : " << EWQ4;
if (EWQ4 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl << endl;
    cout << "Resultaten prioriteitsklasse 5" << endl << endl;

    cout << "Gemiddeld # klanten in systeem (EL) : " << EL5 << endl;
    cout << "Gemiddelde rijlengte (ELQ) : " << ELQ5 << endl;
    cout << "Gemiddelde tijd in systeem (EW) : " << EW5;
if (EW2 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl;
    cout << "Gemiddelde wachttijd in rij (EWQ) : " << EWQ5;
if (EWQ5 < 2.0) cout << " minuut" << endl;
else
    cout << " minuten"<< endl;
    cout << "======" << endl;
};

// hoofdprogramma
int main() {
    for(int i = 0; i < HERHALINGEN; i++) {
        initialize();
        while (evt_list.get_next_event(tijd, soort, p)) {
            handle_event(tijd, soort,p);
        };
        EWqueue /=n;
        ELqueue /=simtime;
        EWqueue1 /=n1;
        ELqueue1 /=simtime;
        EWqueue2 /=n2;
        ELqueue2 /=simtime;
        EWqueue3 /=n3;
        ELqueue3 /=simtime;
        EWqueue4 /=n4;
        ELqueue4 /=simtime;
        EWqueue5 /=n5;
        ELqueue5 /=simtime;
    }
}

```

```

    util_grd /=simtime;
    util_grd /=c;
    util_grd1 /=simtime;
    util_grd1 /=c;
    util_grd2 /=simtime;
    util_grd2 /=c;

    util_grd3 /=simtime;
    util_grd3 /=c;
    util_grd4 /=simtime;
    util_grd4 /=c;
    util_grd5 /=simtime;
    util_grd5 /=c;
    EWQrij[i]=EWqueue;
    ELQrij[i]=ELqueue;
    EWQrij1[i]=EWqueue1;
    ELQrij1[i]=ELqueue1;
    EWQrij2[i]=EWqueue2;
    ELQrij2[i]=ELqueue2;
    EWQrij3[i]=EWqueue3;
    ELQrij3[i]=ELqueue3;
    EWQrij4[i]=EWqueue4;
    ELQrij4[i]=ELqueue4;
    EWQrij5[i]=EWqueue5;
    ELQrij5[i]=ELqueue5;
    rhorij[i]=util_grd;
    rhorij1[i]=util_grd1;
    rhorij2[i]=util_grd2;
    rhorij3[i]=util_grd3;
    rhorij4[i]=util_grd4;
    rhorij5[i]=util_grd5;
    SEED+=1;
    simtime=0;
    bezet =0;
    bezet1 =0;
    bezet2=0;
    bezet3=0;
    bezet4=0;
    bezet5=0;
    rijlengte = 0;
    rijlengte1= 0;
    rijlengte2= 0;
    rijlengte3= 0;
    rijlengte4= 0;
    rijlengte5= 0;
    n =0;
    n1 = 0;
    n2 = 0;
    n3 = 0;
    n4 = 0;
    n5 = 0;
    EWqueue =0;
    ELqueue =0;
    EWqueue1 =0;
    ELqueue1 =0;
    EWqueue2 =0;
    ELqueue2 =0;
    EWqueue3 =0;
    ELqueue3 =0;
    EWqueue4 =0;
    ELqueue4 =0;
    EWqueue5 =0;
    ELqueue5 =0;
    util_grd=0;
    cout << i << endl;
}
report();
cin.get();
};

```

Bronnen

Literatuur

- [1] Cheung W.W.H., Heeney L., Pound J.L. (2002), *An advance triage system*, Accident and Emergency Nursing 2002:10:10-16, New York
- [2] Thompson D.A., Yarnold P.R., Williams D.R., Adams S.L. (1996), *Effects of actual waiting time, perceived waiting time, information delivery and expressive quality on patient satisfaction in the emergency department*, Annals of Emergency Medicine 1996:28:(6):657-665, New York
- [3] George S., Read S., Westlake L., Williams B., Fraser-Moodie A., Pritty P. (1992), *Evaluation of nurse triage in a British accident and emergency department*, BMJ 1992:304:876-878, Londen
- [4] Gross D., Harris C. M. (1985), *Fundamentals of queueing theory 2nd edition*, John Wiley en Sons, New York
- [5] O’Cathain A., Webber E., Nicholl J., Munro J., Knowles E. (2003), *NHS Direct: consistency of triage outcomes*, Emergency Medical Journal 2003:20:289-292, Londen
- [6] Maas P.M. e.a. (1996), *Prisma woordenboek Frans - Nederlands*, Uitgeverij Het Spectrum B.V., Utrecht
- [7] Coenen I.G.A., Hagemeyer A., Caluwé R. de, Voeght F.J. de, Jochems P.J.J. (2005), *Richtlijn Triage op de spoedeisende hulp*, Van Zuiden Communications B.V., Alphen aan den Rijn
- [8] Boxma O.J. (2002), *Stochastic Performance Modelling*, Department of Mathematics and Computer Science Eindhoven University of Technology, Eindhoven
- [9] Schull M.J., Kiss A., Szalai J. (2007), *The effect of low-complexity patients on emergency department waiting times*, Annals of Emergency Medicine 2007:49:(3):257-264, New York
- [10] Maas M. (2007), *SEH-arts biedt meerwaarde*, Medisch Contact 9 februari 2007:251-253, Utrecht

Websites

Wikipedia:

<http://en.wikipedia.org/wiki/Triage>

<http://nl.wikipedia.org/wiki/Triage>