
Handboek simulatie

W.M.P. van der Aalst

*Faculteit Wiskunde en Informatica Technische Universiteit Eindhoven,
Postbus 513, 5600 MB, Eindhoven.*

Recente ontwikkelingen op het gebied van hardware openen nieuwe perspectieven ten aanzien van simulatie. Zo kunnen de hoge resolutie van de huidige generatie beeldschermen alsook de muisbediening gebruikt worden om op een snelle en inzichtelijke wijze een simulatiemodel te bouwen. Ook lenen deze beeldschermen zich voor animatie tijdens het uitvoeren van de simulatie. Door deze ontwikkelingen bestaat het gevaar dat de analyse van de betrouwbaarheid van de gegenereerde resultaten wat op de achtergrond raakt. Keek men vroeger argwanend naar de getallen die uit een simulatie kwamen, nu is men vanwege de mooie grafiek bereid om de resultaten van een simulatie voor waar aan te nemen. Omdat een simulatiestudie vaak uitgevoerd wordt om een belangrijke strategische beslissing te kunnen nemen, is er dus sprake van een groot gevaar. Omdat het onverstandig is om dit soort beslissingen te baseren op ongefundeerde resultaten, moet er meer aandacht besteed worden aan de statistische aspecten van simulatie. Met dit in het achterhoofd is dit handboek ontstaan.

Inhoudsopgave

1	Introductie	4
2	Het construeren van een simulatiemodel	6
3	Trekkingen uit verdelingen	12
3.1	Random en pseudo-random getallen	12
3.2	Enkele verdelingen	14
3.2.1	Bernoulli-verdeling	15
3.2.2	Discrete homogene verdeling	15
3.2.3	Binomiale verdeling	16
3.2.4	Geometrische verdeling	16
3.2.5	Poisson-verdeling	16
3.2.6	Uniforme verdeling	17
3.2.7	Negatief-exponentiële verdeling	18
3.2.8	Normale verdeling	19
3.2.9	Gamma-verdeling	20
3.2.10	Erlang-verdeling	22
3.2.11	χ^2 -verdeling	22
3.2.12	Beta-verdeling	24
4	Verwerken van de resultaten	27
4.1	Gemiddelde en variantie	27
4.2	Subruns en voorlooprun	29
4.2.1	Noodzaak	29
4.2.2	Subruns en aanloopverschijnselen	30
4.3	Analyse van subruns	33
4.3.1	De situatie met meer dan 30 subruns	33
4.3.2	De situatie met minder dan 30 subruns	36
4.4	Variantie-reductie	37
4.5	Gevoeligheidsanalyse	39
5	Valkuilen	40
6	Slotwoord	44

A	Elementaire eigenschappen	46
A.1	Markov's ongelijkheid	46
A.2	Chebyshev's ongelijkheid	46
A.3	Centrale limietstelling	47
A.4	Spreiding normale verdeling	47
B	Overzicht kansverdelingen	48
B.1	Discrete kansverdelingen	48
B.2	Continue kansverdelingen	48
C	Wachtrijmodellen	49

Doelgroep:

Dit handboek is bedoeld voor personen die betrokken zijn bij simulatiestudies maar relatief weinig ervaring hebben met de statistische kanten van simulatie. Het boek is ook geschikt als naslagwerk en als aanvulling op de handleiding van het simulatiegereedschap dat gebruikt wordt.

1 Introductie

Stel dat u als eigenaar van een grote discotheek problemen hebt met de inzet van het personeel op zaterdagavond. Enerzijds is er op bepaalde uren sprake van overcapaciteit, anderzijds zijn er klachten van bezoekers die vinden dat ze te lang moeten wachten bij het weghangen van jassen en bij het bestellen van drankjes. Omdat U als eigenaar van de discotheek het gevoel heeft dat U teveel personeel in dienst heeft en toch klanten kwijt dreigt te raken vanwege de lange wachttijden, besluit U tot het instellen van een diepgaand onderzoek. Voorbeelden van vragen die U beantwoord wilt zien zijn:

- Wat zijn de gemiddelde wachttijden van klanten aan de diverse bars en voor de garderobe?
- Wat is de bezettingsgraad van het barpersoneel?
- Worden de wachttijden substantieel kleiner indien er extra personeel ingezet wordt?
- Heeft het flexibel inzetten van personeel (een personeelslid is bijvoorbeeld niet langer toegewezen aan één bar) zin?
- Wat voor effect heeft de invoering van consumptiebonnen op de gemiddelde wachttijd?
- Heeft het aanstellen van obers zin?
- Wat zijn de effecten van het door middel van een ‘happy hour’ spreiden van de aankomsten van bezoekers?

Simulatiemodel

Om deze en andere vragen te beantwoorden, kan een *simulatiemodel* gebruikt worden. Een simulatiemodel is een afbeelding van de werkelijkheid die gebruikt kan worden om diezelfde werkelijkheid na te bootsen met behulp van een computer. Zoals een architect op basis van een bouwtekening inzicht probeert te verwerven in het gemodelleerde huis, gebruikt een systeemanalist een simulatiemodel om inzicht te verwerven in het gemodelleerde bedrijfsproces.

Redenen voor simulatie

Wanneer is het nu zinvol om een simulatiemodel op te zetten? Een aantal voor de hand liggende redenen zijn:

- Men wil *inzicht* krijgen in een bestaande of nog te realiseren situatie. Door het in kaart brengen van bijvoorbeeld een bedrijfsproces, worden belangrijke verbanden en nutteloze handelingen zichtbaar.

- Een echt experiment is *te duur*. Simulatie is een goedkope manier om een aantal alternatieven te analyseren. Het daadwerkelijk aannemen van extra personeel of het invoeren van een consumptiebonnen-systeem is te duur om vrijblijvend te onderzoeken of het een verbetering tot gevolg heeft. Men wil van tevoren weten of een bepaalde maatregel het gewenste effect heeft. Met name als het gaat om de inrichting van een nieuw bedrijfsproces, kan men door simulatie veel geld besparen.
- Een echt experiment is *te gevaarlijk*. Soms is het niet verantwoord om bepaalde experimenten in het echt uit te voeren. Voordat de Nederlandse Spoorwegen besluit om een nieuw verkeersbegeleidingssysteem in te voeren, wil men eerst vertrouwen hebben in de veiligheid van dit systeem. Hetzelfde geldt voor allerlei processen waar veiligheid een grote rol speelt, bijvoorbeeld in de luchtvaart of in een kerncentrale.

In plaats van een uitgebreide simulatiestudie kan men soms ook volstaan met een wiskundig model, ook wel analytisch model genoemd. In de Operations Research (OR) zijn vele modellen ontwikkeld die zonder simulatie geanalyseerd kunnen worden. We kunnen hierbij denken aan wachtrijmodellen, combinatorische optimaliseringsmodellen, stochastische modellen, enzovoorts. Deze modellen zijn vaak toegesneden op specifieke situaties of zijn alleen maar bruikbaar voor het beantwoorden van specifieke vragen. Het spreekt voor zich dat men pas aan een simulatiestudie moet beginnen op het moment dat men ervan overtuigd is dat een eenvoudig analytisch model niet voldoet. Voor een analytisch model zijn over het algemeen minder gegevens nodig. Ook vergt een analytisch model over het algemeen minder rekenkracht om tot betrouwbare resultaten te komen.

Sterke punten simulatie

Sterke punten van simulatiemodellen ten opzichte van analytisch modellen zijn:

- Simulatie is een flexibele analyse-techniek. Bijna elke denkbare situatie, hoe complex dan ook, kan met behulp van simulatie onderzocht worden.
- Simulatie kan gebruikt worden voor het beantwoorden van zeer uiteenlopende vragen. Met behulp van hetzelfde model kunnen bijvoorbeeld vragen over wachttijden, bezettingsgraden, en storingspercentages beantwoord worden.
- Simulatie is eenvoudig te begrijpen. In feite is simulatie niets meer dan het naspelen van een gemodelleerde situatie. In tegenstelling tot vele analytische modellen is geen specialistische kennis nodig om het model te begrijpen.

Aan simulatie zijn helaas ook wat nadelen verbonden:

- Een simulatiestudie kan erg tijdrovend zijn. Soms zijn er erg lange simulatieruns nodig om betrouwbare resultaten te verkrijgen.
- Bij de interpretatie van simulatieresultaten dient men erg voorzichtig te werk te gaan. Het vaststellen van de betrouwbaarheid van de verkregen resultaten kan erg verraderlijk zijn.
- Simulatie levert geen waterdicht bewijs. Het feit dat er in een simulatie een bepaald effect niet optreedt, wil niet zeggen dat dit effect niet op kan treden.

Simulatie wordt vaak gebruikt om grote strategische beslissingen te onderbouwen met getallen. Vanwege de beschikbaarheid van eenvoudig te bedienen gereedschappen voor simulatie, is men in staat om in korte tijd een simulatiemodel ter ondersteuning van het nemen van deze belangrijke beslissingen te construeren. Hierin schuilt tegelijk het gevaar dat er ten onrechte beslissingen genomen op basis van een foutief model of een verkeerde interpretatie van de resultaten. In dit handboek richten we onze aandacht dan ook op het valideren van een simulatiemodel en het op een juiste manier omgaan met de simulatieresultaten.

De indeling van dit handboek is als volgt. Eerst besteden we in hoofdstuk 2 nog wat aandacht aan het construeren van een simulatiemodel. In hoofdstuk 3 richten we onze aandacht op de ‘input-kant’ van simulatie. De invoer van een simulatie bestaat voor een belangrijk deel uit geparаметeriseerde kansverdelingen. We laten in dit hoofdstuk vooral zien hoe we trekkingen uit een bepaalde kansverdeling kunnen genereren. Hoofdstuk 4 is gericht op de ‘output-kant’ van een simulatiestudie. Hier laten we zien waar we op moeten letten bij het interpreteren van de resultaten. In hoofdstuk 5 laten we tenslotte een aantal veel gemaakte fouten de revue passeren.

2 Het construeren van een simulatiemodel

Simulatiegereedschappen Voor de constructie van een simulatiemodel maken we gebruik van een *gereedschap* dat ervoor zorgt dat de computer aan de hand van dit model de in kaart gebrachte situatie na kan spelen. We kunnen hierbij onderscheid maken tussen twee soorten simulatiegereedschappen:

Simulatietalen Een *simulatietaal* is een programmeertaal die voorzien is van speciale voorzieningen om het maken van een simulatiemodel te vereenvoudigen. Voorbeelden van simulatietalen zijn SIMULA, GPSS, SIMSCRIPT, SIMPAS, SIMON, MUST en GASP.

Simulatiepakketten Een *simulatiepakket* is een gereedschap dat voor een bepaald toepassingsgebied over een set bouwstenen beschikt, waardoor men in korte tijd, en meestal op een grafische manier, een simulatiemodel kan maken. Voorbeelden van simulatiepakketten die speciaal voor productieprocessen zijn ontwikkeld, zijn: SIMFACTORY, WITNESS en TAYLOR.

Het voordeel van een simulatietaal is dat bijna elke situatie gemodelleerd kan worden. Het nadeel is dat men gedwongen is om in termen van een programmeertaal de situatie in kaart te brengen. Dit heeft tot gevolg dat het maken van een model een tijdrovende bezigheid is en dat het opgeleverde model weinig inzichtelijk is. Een simulatiepakket biedt de mogelijkheid om in een korte tijd en op een inzichtelijke wijze een model te bouwen. Omdat men gedwongen is het model op te bouwen uit kant-en-klare bouwstenen is het toepassingsgebied beperkt. Zodra men buiten het specifieke toepassingsdomein treedt, moet men veel moeite doen om de situatie correct te modelleren. Vaak is het zelfs onmogelijk om bijvoorbeeld een afwijkende besturing te modelleren.

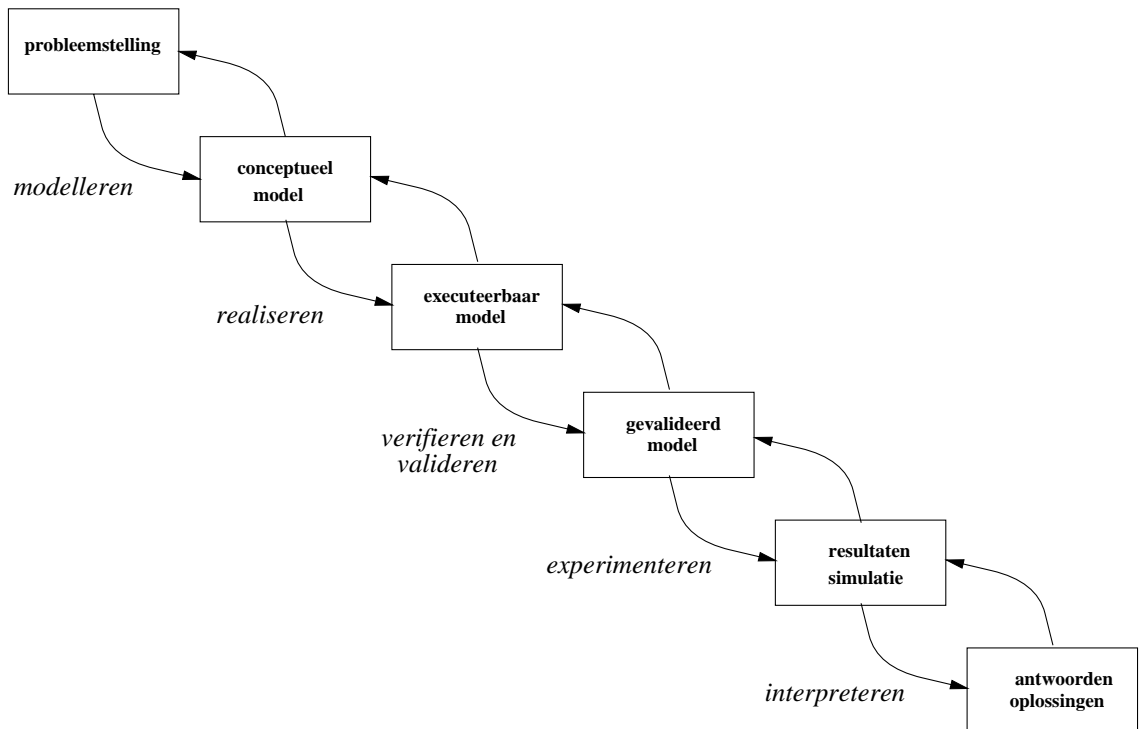
In de afgelopen jaren zijn er ook enkele gereedschappen op de markt gekomen die de voordelen van een simulatietaal en een simulatiepakket proberen te combineren. Voorbeelden zijn ExSpect, Design/CPN, Extend en ARENA (voorheen SIMAN/CINEMA). Deze gereedschappen koppelen een grafische ontwikkelomgeving aan een soort van programmeertaal en bieden de mogelijkheid tot animatie. ExSpect en Design/CPN zijn gebaseerd op hoog-niveau Petri-netten waardoor er behalve simulatie ook nog andere analyse-technieken gebruikt kunnen worden. Gereedschappen als Extend en ARENA zijn gebaseerd op zelf ontwikkelde concepten en daardoor ongeschikt voor verdere analyse. Doordat pakketten als Extend en ARENA over eigen ‘proprietary’ bouwstenen beschikken, is het moeilijk simulatiemodellen tussen deze pakketten uit te wisselen. Standardisatie op basis van een formele beschrijvingswijze als hoog-niveau Petri-netten zou de uitwisselbaarheid tussen de diverse gereedschappen ten goede komen.

Fasering

Het ontbreken van een algemeen geaccepteerde beschrijvingswijze maakt het onmogelijk om voor al deze simulatiegereedschappen aan te geven op welke wijze het model geconstrueerd moet worden. Wel is de *fasering* van een simulatiestudie voor al deze gereedschappen min of meer hetzelfde. In figuur 1 is de fasering van een typische simulatiestudie weergegeven.

Probleemstelling

Het simulatietraject start met een goede *probleemstelling*. Deze probleemstelling geeft aan wat de doelstellingen van de simulatiestudie zijn. Daarnaast wordt in de probleemstelling het proces dat onderzocht gaat worden precies afgebakend. Op deze manier ligt vast wat wel en wat niet tot het simulatiemodel zal behoren. De probleemstelling moet ook



Figuur 1: De fasering van een simulatiestudie.

aangeven welke vragen beantwoord moeten gaan worden. Bij voorkeur gaat het hier om vragen die kwantificeerbaar zijn. De vraag: “Zijn de klanten tevreden?” kan beter vervangen worden door vragen als: “Hoe lang moeten de klanten gemiddeld wachten?”.

Modelleren *Conceptueel model*

Na het opstellen van de probleemstelling vindt er een fase van *modelvorming* plaats. Deze modelvorming moet leiden tot een *conceptueel model*. In het conceptuele model wordt aangegeven uit welke objecten het model bestaat en wat de samenhang is tussen de diverse objecten. In het geval van de discotheek zijn de te onderscheiden objecten onder andere de diverse personeelsleden, de diverse bars, de garderobe en de klanten. Van al deze objecten worden de relevante kenmerken (eigenschappen) geregistreerd. We kunnen hierbij onderscheid maken tussen *kwalitatieve kenmerken* en *kwantitatieve kenmerken*. Voorbeelden van kwalitatieve kenmerken zijn rijdisciplines (In welke volgorde worden de klanten geholpen, bijvoorbeeld FIFO (First-In-First-Out) of LIFO (Last-In-First-Out)?), sequenties (Wat is de routing van een klant?) en beslisregels (Wanneer gaat er een extra bar open?). Kwantitatieve kenmerken hebben vooral betrekking op de capaciteit (Hoeveel klanten kunnen er tegelijk bediend worden?) en snelheid (Hoelang duurt de bediening van één klant?) van objecten. Indien we te maken hebben met objecten die tot dezelfde klasse behoren, dan beschrijven we deze kenmerken voor de gehele klasse

(eventueel geparparameteriseerd). Om met name de verbanden tussen de diverse objecten of objectklassen weer te geven, wordt vaak gebruik gemaakt van een tekentechniek om (delen van) het model grafisch weer te geven. We kunnen hierbij denken aan toestandsdiagrammen, dataflow-diagrammen of flowdiagrammen specifiek voor simulatiedoeleinden.

Tijdens de constructie van het conceptuele model komen waarschijnlijk een aantal vaagheden en tegenstrijdigheden in de probleemstelling boven water. Ook kunnen er tijdens de modelvorming nieuwe vragen opduiken die men met de simulatiestudie wil beantwoorden. In beide gevallen zal de probleemstelling bijgesteld moeten worden.

Realiseren

Executeerbaar model

Nadat het, op basis van het conceptuele model, duidelijk is hoe het simulatiemodel eruit moet zien, is de *realisatie-fase* aangebroken. In deze fase moet het conceptuele model afgebeeld worden op een *executeerbaar model*. Het executeerbare model is een model dat door de computer zonder menselijke tussenkomst nagespeeld kan worden. Hoe dit model gerealiseerd dient te worden, hangt sterk af van het simulatiegereedschap dat gebruikt wordt. In het geval van een simulatietaal, is er sprake van een echte implementatie-fase. In het geval van een simulatiepakket dat past bij de probleemstelling, betreft het slechts de instelling van een aantal parameters. Bij het gebruik van een simulatiegereedschap gebaseerd op een formeel executeerbaar model (bijvoorbeeld Petri-netten) vallen de modelleringsfase en de realisatie-fase feitelijk samen!

Verificatie

Op het moment dat men beschikt over een executeerbaar model, moet het model nog correct bevonden worden. In de eerste plaats moet het model *geverifieerd* worden. Verificatie van het model is nodig om te onderzoeken of het model nog logische of statistische fouten bevat. We kunnen hierbij denken aan programmeerfouten of aan verkeerde parameterinstellingen. Voor verificatiedoeleinden kunnen kleine proefruns met de hand nagespeeld worden om te kijken of de resultaten kloppen, ook kan men het model een *stress-test* laten ondergaan. In de stress-test wordt het model aan een aantal extreme situaties blootgesteld. We kunnen bijvoorbeeld meer klanten aan laten komen dan dat er bediend kunnen worden. In dat geval zullen de gemeten wachttijden gaandeweg sterk toe moeten nemen. Sommige gereedschappen ondersteunen meer geavanceerde vormen van verificatie. Met behulp van een op Petri-netten gebaseerd simulatietool, kan men bijvoorbeeld allerlei logische eigenschappen van het model bewijzen. Men kan bijvoorbeeld bepaalde behoudswetten (het aantal personeelsleden is constant) en andere logische eigenschappen (afwezigheid van deadlock) afleiden.

Valideren

Behalve verificatie is ook *validatie* van het model noodzakelijk. Door middel van validatie wordt nagegaan of het simulatiemodel een correcte weergave van de werkelijkheid is. In het geval dat men een bestaande

situatie simuleert, kan men de resultaten van een simulatierun vergelijken met waarnemingen uit de werkelijkheid. In dat geval wordt gebruik gemaakt van historische data. Ook als het simulatiemodel een nog te realiseren situatie betreft, kan men de simulatieresultaten vergelijken met de resultaten die men verwacht op basis van eenvoudige berekeningen. Verificatie en validatie kunnen leiden tot aanpassingen van het executeerbare simulatiemodel. Het is zelfs mogelijk dat vernieuwde inzichten leiden tot een aanpassing van de probleemstelling en/of het conceptuele model. Nadat het simulatiemodel correct bevonden is spreken we ook wel over een *gevalideerd model*.

Gevalideerd model

Experimenteren

Uitgaande van het gevalideerde model, kunnen *experimenten* uitgevoerd worden. Deze experimenten moeten zo uitgevoerd worden dat er op een zo efficiënt mogelijke wijze betrouwbare resultaten bereikt worden. Er moeten in dit stadium bijvoorbeeld beslissingen genomen worden die betrekking hebben op het aantal simulatieruns en de lengte van elke run.

Resultaten simulatie

Interpreteren

De simulatieresultaten moeten geïnterpreteerd worden voordat er een terugkoppeling naar de probleemstelling plaats kan vinden. Voor de diverse metingen die tijdens de simulatie verzameld zijn, moeten betrouwbaarheidsintervallen opgesteld worden. Ook moeten de resultaten vertaald worden naar antwoorden op de vragen in de probleemstelling. Van elke antwoord moet duidelijk zijn wat de betrouwbaarheid is. Dit alles wordt samengevat in een eindrapport met antwoorden en oplossingen voor vraagstukken uit de probleemstelling.

Antwoorden en oplossingen

In figuur 1 zien we dat er vele terugkoppelingen mogelijk zijn. In de praktijk is het zelfs zo dat vele fases min of meer gelijktijdig uitgevoerd worden. Met name het experimenteren en het interpreteren van de resultaten zal in veel gevallen hand-in-hand gaan.

Alternatieven

Bij het opstellen van figuur 1 zijn we ervan uitgegaan dat er sprake is van één simulatiemodel. In de praktijk is dit zelden het geval omdat we meestal een aantal *alternatieve situaties* met elkaar willen vergelijken. In dat geval is er dus sprake van meerdere simulatiemodellen waarvan de bijbehorende simulatieresultaten met elkaar vergeleken worden. Een veel voorkomende situatie is de volgende: uitgaande van een bestaande situatie heeft men een aantal verbeteringen op het oog die men door middel van simulatie met elkaar wil vergelijken. In dat geval wordt er eerst een model van de huidige situatie gemaakt. Voor dit model worden de fases uit figuur 1 doorlopen. Daarna wordt het model aangepast zodanig dat het model een mogelijke toekomstige situatie voorstelt. In dat geval kunnen de fases uit figuur 1 in een versneld tempo uitgevoerd worden. We spreken in dit verband ook wel over een *what-if analyse*. Hetzelfde

What-if analyse

doen we voor de andere alternatieven. Op basis van deze *what-if* analyses kunnen we de verschillende alternatieven met elkaar vergelijken. Deze eindresultaten kunnen we gebruiken om vast te stellen of we de huidige situatie willen veranderen. Zo ja, dan kunnen we ook aangeven wat de meest succesvolle aanpassing lijkt.

Gebruiker

Systeemanalist

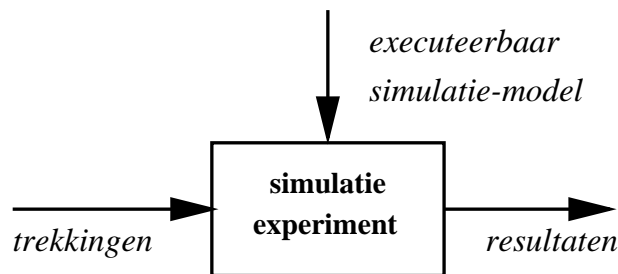
Programmeur

In figuur 1 zijn we ook voorbij gegaan aan het feit dat er bij een simulatiestudie verschillende personen betrokken kunnen zijn. In de eerste plaats hebben we te maken met de *gebruiker*, d.w.z. de persoon of de afdeling die met een probleem zit dat men door middel van simulatie wil onderzoeken. In de tweede plaats is er een *systeemanalist* die ervoor moet zorgen dat er een heldere probleemstelling op papier komt. Verder maakt de systeemanalist het conceptuele model. Afhankelijk van het gereedschap dat gebruikt wordt, kan het zijn dat de systeemanalist ondersteund wordt door een *programmeur* om het simulatiemodel te realiseren. Wie de experimenten uitvoert, hangt sterk af van het aantal simulaties. Betreft het simulaties die regelmatig uitgevoerd worden, dan lijkt een gebruiker de aangewezen persoon. Betreft het een simulatiestudie in het kader van een éénmalige strategische beslissing, dan is de systeemanalist of de programmeur de aangewezen persoon. Voor de correcte interpretatie van de simulatieresultaten is het van groot belang dat hier personen met voldoende kennis van de statistische aspecten van simulatie betrokken zijn.

Animatie

Omdat de gebruiker over het algemeen niet diegene is die het simulatiemodel maakt, is het van groot belang dat de gebruiker en de bouwers (d.w.z. de systeemanalist en de programmeur) op één lijn zitten. Het uiteindelijke model moet passen bij het model dat de gebruiker in gedachten had. Een manier om dit proces te ondersteunen is het gebruik van *animatie*. Animatie is het met behulp van een simulatiemodel *grafisch* naspelen van de gemodelleerde situatie. We kunnen hierbij denken aan bewegende objecten op het scherm die ook van vorm kunnen veranderen. Op deze manier kan een soort van ‘tekenfilm’ gemaakt worden die aansluit bij de belevingswereld van de gebruiker. Ook al is animatie een bruikbaar gereedschap voor bijvoorbeeld de validatie van het model samen met de gebruiker, toch moet het met de nodige voorzichtigheid toegepast worden. Juist een mooie animatie kan de gebruiker een onterecht gevoel van vertrouwen in het model geven, waardoor de uiteindelijke kwantitatieve resultaten voetstoots aangenomen worden.

In de rest van dit handboek richten we ons vooral op de statistische aspecten van het uitvoeren simulatie-experimenten. In figuur 2 is schematisch aangegeven wat de invoer en uitvoer van een simulatie-experiment is. De invoer bestaat uit een executeerbaar simulatiemodel en random trekkin-



Figuur 2: De invoer en uitvoer van een simulatie-experiment.

gen uit diverse geparameteriseerde kansverdelingen. De uitvoer bestaat uit simulatieresultaten. In hoofdstuk 3 richten we ons op de random trekkingen die nodig zijn om te experimenteren. In hoofdstuk 4 richten we ons op de verwerking van de simulatieresultaten.

3 Trekkingen uit verdelingen

3.1 Random en pseudo-random getallen

In essentie komt een simulatie-experiment neer op het domweg naspelen van een gemodelleerde situatie. Om deze situatie na te kunnen spelen met behulp van een computermodel, moeten we aannames maken ten aanzien van het gemodelleerde systeem zelf, maar ook ten aanzien van de *omgeving* waarin dit model opereert. Omdat we bepaalde zaken niet in detail kunnen of willen modelleren, zoeken we onze toevlucht in het *toeval!* Omdat we bijvoorbeeld niet weten wanneer en hoeveel klanten het postkantoor betreden, maar wel een idee van het gemiddelde en de spreiding hebben, laten we de computer schijnbaar willekeurige trekkingen uit één of andere kansverdeling doen. Omdat de computer van nature een deterministisch apparaat is, zijn er technieken ontwikkeld om zogenaamde *pseudo-random getallen* te genereren.

Omgeving

Toeval

Pseudo-random getal

Random generator

Een *random generator* is een stukje software dat pseudo-random getallen produceert. We spreken over pseudo-random getallen omdat de computer een deterministisch algoritme gebruikt om deze getallen te genereren. De meeste random generatoren genereren pseudo-random getallen tussen 0 en 1. Omdat elke waarde tussen 0 en 1 even waarschijnlijk is zeggen we ook wel dat de gegenereerde getallen uniform verdeeld zijn over het interval tussen 0 en 1. Het hangt van de random generator af of 0 en 1 zelf ook tot de mogelijke uitkomsten behoren. Omdat de kans op precies 0 of precies 1 gelijk is aan 0 (in theorie), is dit echter niet relevant. De meeste random generatoren genereren een reeks pseudo-random ge-

tallen $\frac{X_i}{m}$ volgens het voorschrift:

$$X_n = (aX_{n-1} + b) \text{ modulo } m$$

Voor elke i is X_i een getal uit de verzameling $\{0, 1, 2, \dots, m - 1\}$ en $\frac{X_i}{m}$ komt overeen met een trekking uit een uniforme verdeling tussen 0 en 1. De getallen a , b en m worden zo gekozen dat de reeks niet of nauwelijks van ‘echte random getallen’ onderscheiden kunnen worden. Zo moet de reeks X_i elk getal van de getallen $0, 1, 2, \dots, m - 1$ een keer aandoen. Verder moet m zo gekozen zijn dat de getallen X_i nog net gerepresenteerd kunnen worden in de computer. Er zijn diverse tests om de kwaliteit van een random generator te controleren (cf. [2, 12, 14, 9]):

- frequency test
- correlation test
- run test
- gap test
- poker test

Een goede random generator voor een 32-bit computer is:

$$X_n = 16807X_{n-1} \text{ modulo } (2^{31} - 1)$$

D.w.z. $a = 16807$, $b = 0$ en $m = 2^{31} - 1$. Voor een 36-bit machine is:

$$X_n = 3125X_{n-1} \text{ modulo } (2^{35} - 31)$$

een goede keuze.

Seed

Het eerste getal in de reeks (X_0) moet steeds gekozen worden. Dit eerste getal noemen we het *seed*. Het seed dient als uitgangspunt voor de te genereren reeks random getallen. Bij het gebruik van een goede random generator levert elk seed een andere reeks randomgetallen op. Soms kiest de computer zelf het seed (bijvoorbeeld op basis van de systeemklok). Het verdient echter de voorkeur dat de gebruiker zelf bewust kiest voor een bepaald seed. Op deze manier is het ook mogelijk om later een simulatie-experiment te *reproducen*. De reproduceerbaarheid van een simulatie-experiment is belangrijk op het moment dat men een onverwacht verschijnsel nader wil onderzoeken.

Reproduceerbaarheid

Omdat de meeste simulatietalen en pakketten een goede tot redelijke random generator beschikbaar hebben, kunnen we het genereren van pseudo-random getallen als een black-box beschouwen. Toch is enige voorzichtigheid geboden: pseudo-random getallen zijn per definitie niet

random! (Er wordt immers een deterministisch algoritme gebruikt om ze te genereren.) Daarom is het aan te bevelen om slechts één generator te gebruiken en zorgvuldig om te gaan met de keuze van het seed.

Ter illustratie van potentiële gevaren ten aanzien van het gebruik van random generatoren noemen we nog twee bekende valkuilen.

De eerste fout die wel eens gemaakt wordt, is het gebruik van de zogenaamde ‘lagere orde bits’ van een random reeks. Indien een random generator bijvoorbeeld het getal 0.1321734234 produceert, dan is het eerste deel (d.w.z. de hogere orde bits: 0.13217) van betere kwaliteit dan het tweede deel (d.w.z. de lagere orde bits: 34234). In het algemeen vertonen de minst significante cijfers een duidelijk cyclisch gedrag.

Een andere, vaak voorkomende fout is het dubbel gebruiken van een random getal. In een simulatie kan bijvoorbeeld de fout optreden dat een random getal in sommige gevallen meerdere malen gebruikt wordt voor het genereren van een trekking uit een kansverdeling. Op deze wijze sluipen er afhankelijkheden in het model die in de werkelijkheid niet aanwezig zijn, en op die manier tot zeer onbetrouwbare resultaten leiden.

3.2 Enkele verdelingen

Het komt slechts zelden voor dat we random getallen nodig hebben die uniform tussen 0 en 1 verdeeld zijn. Afhankelijk van de simulatie die we uit willen voeren hebben we trekkingen uit uiteenlopende *kansverdelingen* nodig (bijvoorbeeld: normale, negatief-exponentiële, gamma, Poisson of binomiale verdeling). Een kansverdeling specificeert welke waarden er mogelijk zijn en de waarschijnlijkheid van elke waarde, d.w.z. gegeven een kansverdeling ligt de *kans* op een bepaalde trekking vast.

Om het praten over kansverdelingen en trekkingen uit kansverdelingen wat te vergemakkelijken, introduceren we de term *stochastische grootheid*, kortweg *stochast* genoemd (Engels: *random variable*). Een stochast X is een variable die met een bepaalde kans bepaalde waarden aan kan nemen. Het werpen van een dobbelsteen kunnen we bijvoorbeeld modeleren door middel van een stochast X die de waarden 1, 2, 3, 4, 5 en 6 aan kan nemen. De kans op het werpen van één bepaalde waarde uit deze verzameling is gelijk aan $\frac{1}{6}$. Dit noteren we ook wel als volgt:

$$\mathbb{P}[X = a] = \begin{cases} \frac{1}{6} & \text{indien } a \in \{1, 2, 3, 4, 5, 6\} \\ 0 & \text{anders} \end{cases}$$

Gegeven een stochast X kunnen we spreken over de *verwachting* en de *variantie* van X . De verwachting van X , genoteerd als $\mathbb{E}[X]$, is het te verwachten gemiddelde van een groot aantal trekkingen uit X . We spreken daarom ook wel over het *gemiddelde* van X . De variantie, genoteerd

Kansverdeling

Kans

Stochastische grootheid

Verwachting

Variantie

$\mathbb{E}[X]$

Var[X]

als $\text{Var}[X]$, is een maat voor de gemiddelde afwijking van het gemiddelde (d.w.z. de verwachting) van X . Indien X een hoge variantie heeft zullen de trekkingen voor een belangrijk deel ver van het gemiddelde liggen. Een lage variantie geeft aan dat trekkingen over het algemeen dicht bij het gemiddelde liggen. De verwachting van een stochast ($\mathbb{E}[X]$) noteren we vaak met de letter μ , de variantie ($\text{Var}[X]$) noteren we als σ^2 . De relatie tussen verwachting en variantie is weergegeven in de volgende gelijkheid:

$$\text{Var}[X] = \mathbb{E}[(X - \mu)^2] = \mathbb{E}[X^2] - \mu^2$$

Standaardafwijking

Omdat $\text{Var}[X]$ de verwachting van het *kwadraat* van de afwijking van het gemiddelde is, is de wortel van $\text{Var}[X]$ een betere maat voor de afwijking van het gemiddelde. Daarom noemen we $\sigma = \sqrt{\text{Var}[X]}$ de *standaardafwijking* van X . Hoe groter σ wordt, hoe groter de afwijkingen van het gemiddelde. Indien de standaardafwijking van X groot is ten opzichte van het gemiddelde, dan zeggen we ook wel dat X *wild* verdeeld is. Voor meer details verwijzen we naar appendix A en de lijst met referenties.

In deze sectie laten we een aantal veelvuldig gebruikte kansverdelingen de revue passeren. We beginnen met een aantal discrete kansverdelingen.

3.2.1 Bernoulli-verdeling

Bernoulli-verdeling
 p

Laat X een stochast zijn welke verdeeld is volgens een *Bernoulli-verdeling* met parameter p . In dit geval:

$$\mathbb{P}[X = 0] = 1 - p \quad \text{en} \quad \mathbb{P}[X = 1] = p$$

De verwachting $\mathbb{E}[X]$, d.w.z. de gemiddelde waarde, is p . De variantie $\text{Var}[X]$ is gelijk aan $p(1 - p)$.

3.2.2 Discrete homogene verdeling

Homogene verdeling
 a, b

Een stochast X is verdeeld volgens een *discrete homogene verdeling* met ondergrens a en bovengrens b , indien elke gehele waarde tussen a en b even waarschijnlijk is en de kans op een waarde kleiner dan a of groter dan b gelijk aan 0 is. De ondergrens a en bovengrens b zijn gehele getallen. In dit geval is de kans op een bepaalde waarde k gelijk aan:

$$\mathbb{P}[X = k] = \begin{cases} \frac{1}{(b-a)+1} & \text{indien } k \in \{a, a+1, a+2, \dots, b-1, b\} \\ 0 & \text{anders} \end{cases}$$

Het werpen van een dobbelsteen komt dus overeen met een trekking uit een discrete homogene verdeling met ondergrens 1 en bovengrens 6. De verwachting ($\mathbb{E}[X]$) is gelijk aan $\frac{a+b}{2}$. De variantie ($\text{Var}[X]$) is gelijk aan $\frac{(b-a)((b-a)+2)}{12}$.

3.2.3 Binomiale verdeling

Binomiale verdeling
 n, p

Stel we voeren n experimenten uit. Per experiment is de kans op succes gelijk aan p . Wat is nu de kans op een bepaald aantal successen? Dit modelleren we door middel van een *binomiale verdeling* met parameters n en p . Stel X is binomiaal verdeeld met parameters n en p . Voor $x \in \{0, 1, \dots, n\}$ geldt dan het volgende:

$$\mathbb{P}[X = k] = \binom{n}{k} p^k (1 - p)^{n-k}$$

Het werpen van 10 geldstukken op een tafel en het daarna tellen van het aantal geldstukken met munt boven, komt dus overeen met een trekking uit een binomiale verdeling met parameters $n = 10$ en $p = 0.5$. De verwachting is gelijk aan np . De variantie is gelijk aan $np(1 - p)$. Een speciaal geval van de binomiale verdeling is de Bernoulli-verdeling ($n = 1$).

3.2.4 Geometrische verdeling

Geometrische verdeling
 p

Stel we herhalen een experiment met slagingskans p totdat we voor de eerste keer succes hebben. Het aantal experimenten dat nodig is om het eerste succes te hebben is een trekking uit een *geometrische verdeling* met parameter p . Stel X is geometrisch verdeeld met parameter p , dan is voor voor elk positief geheel getal k :

$$\mathbb{P}[X = k] = (1 - p)^{k-1} p$$

De verwachting is gelijk aan $\frac{1}{p}$. De variantie is gelijk aan $\frac{1-p}{p^2}$. Deze verdeling wordt ook wel de Pascal-verdeling genoemd.

3.2.5 Poisson-verdeling

Poisson-verdeling

λ

De *Poisson-verdeling* is sterk verbonden met de negatief-exponentiële verdeling die we later tegen zullen komen. Indien de tijd tussen twee opeenvolgende aankomsten van een klant in een supermarkt verdeeld is volgens een negatief-exponentiële verdeling met parameter λ , dan is het aantal klanten dat per tijdseenheid de supermarkt binnenkomt Poisson verdeeld met parameter λ .

Stel X is Poisson verdeeld met parameter λ , dan geldt voor elk natuurlijk getal k :

$$\mathbb{P}[X = k] = e^{-\lambda} \frac{\lambda^k}{k!}$$

De verwachting ($\mathbb{E}[X]$) is gelijk aan λ . De variantie ($\text{Var}[X]$) is ook gelijk aan λ .

Beschouw een supermarkt waar klanten volgens een negatief-exponentiële verdeling binnenkomen. De gemiddelde tijd tussen twee aankomsten is 15

minuten (0.25 uur), dat wil zeggen $\lambda = \frac{1}{0.25} = 4$. Het aantal aankomsten per uur is Poisson verdeeld met parameter $\lambda = 4$. Het gemiddeld aantal aankomsten is dus gelijk aan 4. Ook de variantie is gelijk aan 4.

We vervolgen deze sectie met de beschrijving van een aantal continue verdelingen. In tegenstelling tot een discrete verdeling kunnen we nu niet spreken over de kans op een bepaalde waarde ($\mathbb{P}[X = k]$). (De kans op een bepaalde waarde is per definitie gelijk aan 0.) Wel kunnen we spreken over de *kansdichtheid*. De kansdichtheid is een maat voor de relatieve waarschijnlijkheid van een bepaalde waarde.

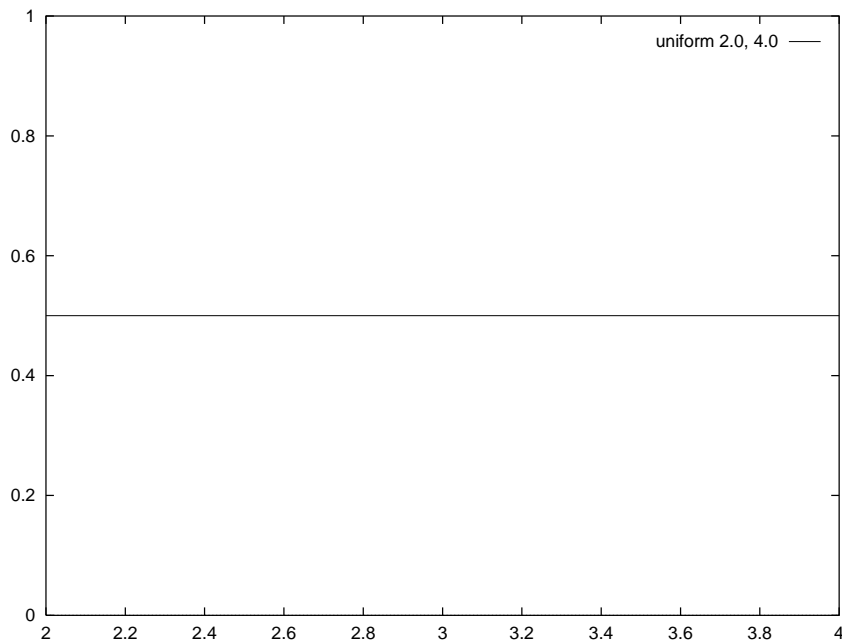
Kansdichtheid

3.2.6 Uniforme verdeling

Uniforme verdeling

a, b

De *uniforme verdeling*, ook wel homogene verdeling genoemd, is de meest eenvoudige continue kansverdeling die denkbaar is. Tussen een ondergrens a en een bovengrens b zijn alle waarden even waarschijnlijk. Stel dat stochast X uniform verdeeld is met parameters a en b . De bijbehorende kansdichtheid: $f_X(x) = \frac{1}{b-a}$ voor x tussen a en b ($a \leq x \leq b$). In figuur 3 is de uniforme verdeling voor $a = 2$ en $b = 4$ weergegeven. We



Figuur 3: De uniforme verdeling met parameters $a = 2$ en $b = 4$.

kunnen in deze figuur inderdaad zien dat elke waarde tussen a en b even waarschijnlijk is. De verwachting ($\mathbb{E}[X]$) is gelijk aan $\frac{a+b}{2}$. De variantie ($\text{Var}[X]$) is gelijk aan $\frac{(b-a)^2}{12}$.

3.2.7 Negatief-exponentiële verdeling

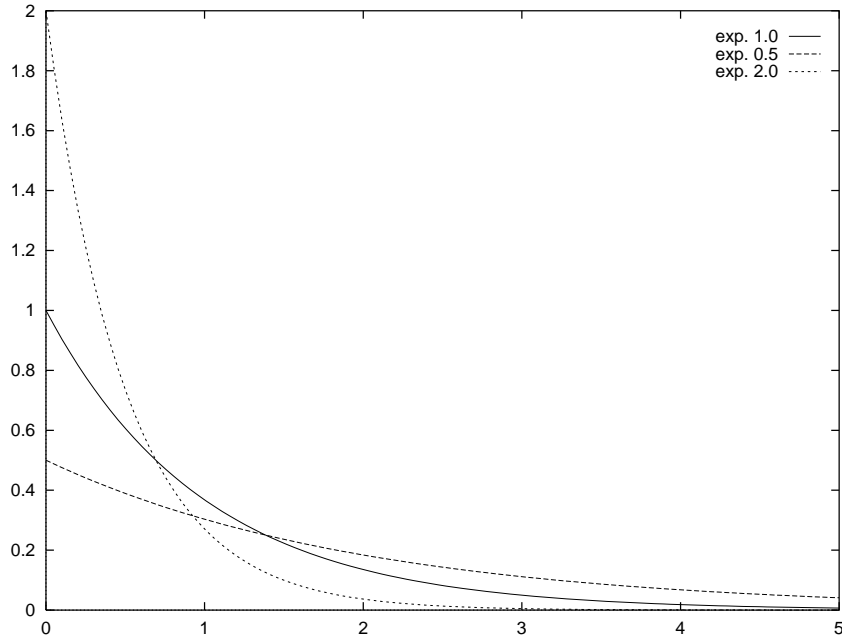
Negatief-exponentiële verdeling

λ

De *negatief-exponentiële verdeling* is een kansverdeling die vaak gebruikt wordt om een bepaald aankomstproces te modelleren. De negatief-exponentiële verdeling heeft één parameter: λ . Laat X een negatief-exponentieel verdeelde stochast zijn met parameter λ . De bijbehorende kansdichtheid is dan ($x \geq 0$):

$$f_X(x) = \lambda e^{-\lambda x}$$

In figuur 4 is deze kansdichtheid voor $\lambda = 1.0$, $\lambda = 0.5$ en $\lambda = 2.0$ weergegeven. De verwachting is gelijk aan $\frac{1}{\lambda}$. De variantie is gelijk aan $\frac{1}{\lambda^2}$.



Figuur 4: De negatief-exponentiële verdeling met parameter $\lambda = 1.0$, $\lambda = 0.5$ en $\lambda = 2.0$.

Intensiteit λ

Indien een negatief-exponentieel verdeelde stochast X met parameter λ gebruikt wordt voor het modelleren van een aankomstproces, dan wordt λ ook wel de *intensiteit* van het aankomstproces genoemd. De stochast X wordt gebruikt om de tijd tussen twee opeenvolgende aankomsten te modelleren. Hoe groter de intensiteit van het aankomstproces, d.w.z. hoe groter λ , hoe groter het gemiddeld aantal aankomsten per tijdseenheid. Indien $\lambda = 10$, dan is de verwachte tijd tussen twee opeenvolgende aankomsten gelijk aan 0.10 tijdseenheden. Het gemiddeld aantal aankomsten per tijdseenheid is in dit geval dus gelijk aan 10. In het geval $\lambda = 100$, is de verwachte tijd tussen twee opeenvolgende aankomsten gelijk aan 0.01 tijdseenheden en het gemiddeld aantal aankomsten per tijdseenheid 100.

3.2.8 Normale verdeling

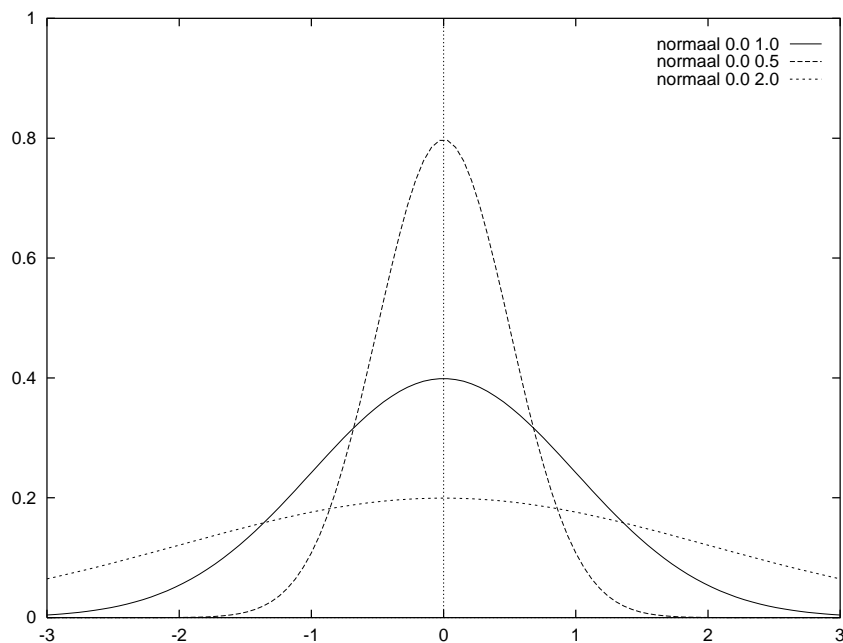
Normale verdeling

μ, σ

De *normale verdeling* kent vele toepassingen. Deze verdeling wordt onder andere gebruikt voor het modelleren van bewerkingstijden, response-tijden, voorraadhoogten en transporttijden. Ook wordt de normale verdeling gebruikt voor het benaderen van andere verdelingen. Een normaal verdeelde stochast X heeft twee parameters μ en σ . Voor een willekeurige x is de kansdichtheid als volgt gedefinieerd:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

In tegenstelling tot de negatief-exponentiële verdeling kan deze stochast ook negatieve waarden aannemen. In figuur 5 zien we voor een aantal parameterwaarden de bijbehorende kansdichtheden. De kansdichtheden



Figuur 5: De normale verdeling met parameters $\mu = 0.0$ en σ achtereenvolgens 1, 0.5 en 2.

rond μ zijn het hoogst, naarmate we verder van het gemiddelde komen neemt de kansdichtheid af. Hoe kleiner σ hoe sneller de kansdichtheid afneemt, d.w.z. bij een kleine waarde van σ is een waarde rond μ erg waarschijnlijk. De verwachting ($\mathbb{E}[X]$) is gelijk aan μ . De variantie ($\text{Var}[X]$) is gelijk aan σ^2 .

Standaard-normale verdeling

Indien $\mu = 0$ en $\sigma = 1$ dan spreken we over een *standaard-normale verdeling*. Indien Y een standaard-normaal verdeelde stochast, dan is $X = \mu + \sigma Y$ een normaal verdeelde stochast met parameters μ en σ . Omgekeerd geldt iets soortgelijks. Indien X een normaal verdeelde sto-

chast met parameters μ en σ , dan is $Y = \frac{X-\mu}{\sigma}$ een standaard-normaal verdeelde stochast.

Indien we een normaal verdeelde stochast gebruiken voor het modelleren van een bepaalde tijdsduur (bijvoorbeeld een bewerkingstijd, responsetijd of transporttijd), dan moeten we rekening houden met het feit dat deze stochast ook *negatieve* waarden aan kan nemen. In het algemeen zijn negatieve tijdsduren niet mogelijk. Om dit te voorkomen kunnen we bijvoorbeeld een nieuwe trekking doen in het geval dat een bepaalde trekking een negatieve waarde oplevert. Merk op dat dit gevolgen heeft voor het gemiddelde en de variantie. Deze oplossing is dus alleen aan te raden indien de kans op een negatieve waarde heel klein is. Als vuistregel kunnen de volgende regel aanhouden: “Indien $\mu - 2\sigma > 0$ kunnen we de gegenereerde negatieve waarden overslaan.”. (De kans dat er een negatieve waarde gegenereerd wordt, onder de voorwaarde dat $\mu - 2\sigma > 0$, is kleiner dan 0.023.)

3.2.9 Gamma-verdeling

Een kenmerk van de normale verdeling is het feit dat deze verdeling symmetrisch is. In de praktijk komt het echter vaak voor dat we een verdeling willen hebben die ‘scheef’ is, d.w.z. de kansdichtheden onder het gemiddelde zijn duidelijk anders verdeeld dan de kansdichtheden boven het gemiddelde. Is dit het geval dan ligt een *gamma-verdeling* voor de hand. Een stochast X is is gamma-verdeeld met parameters $r > 0$ en $\lambda > 0$ indien voor alle $x > 0$:

$$f_X(x) = \frac{\lambda(\lambda x)^{r-1} e^{-\lambda x}}{\Gamma(r)}$$

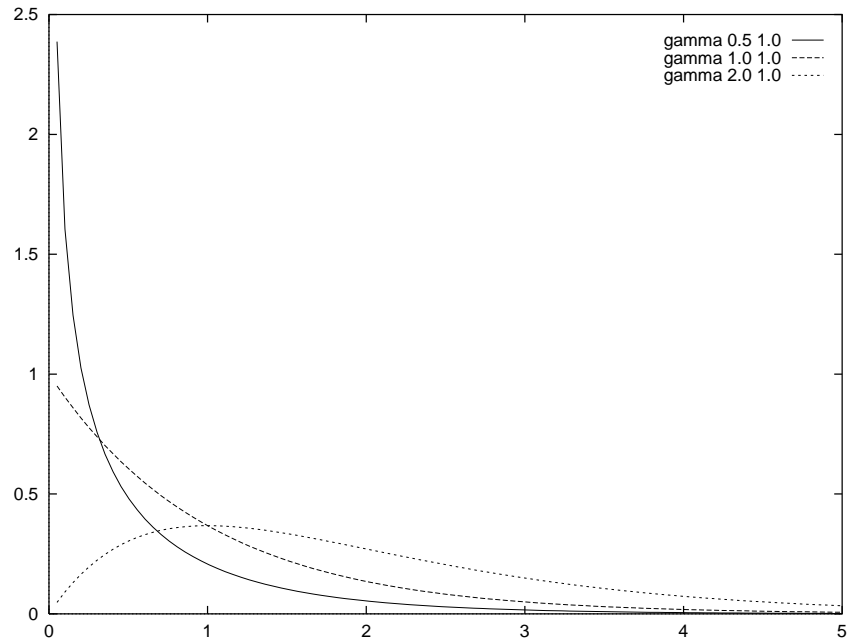
(De functie $\Gamma(r)$ is de zogenaamde gamma-functie.) In de figuren 6, 7 en 8 zijn voor een aantal parameterwaarden de bijbehorende kansdichtheden weergegeven. In deze figuren is duidelijk te zien dat de gamma-verdeling, afhankelijk van de parameterwaarden, vele verschijningsvormen heeft. Indien r kleiner of gelijk aan 1 is, is de f_X een monotoon dalende functie. Waarden dicht bij 0 zijn in dat geval het meest waarschijnlijk. Indien $r > 1$ stijgt de functie eerst tot een maximum, om daarna monotoon te dalen. De parameter r bepaalt dus de vorm van de verdeling. De parameter λ bepaalt de ‘uitgestrektheid’ van de verdeling. Hoe groter λ hoe groter de kans op een waarde dicht bij 0. De verwachting ($\mathbb{E}[X]$) is gelijk aan $\frac{r}{\lambda}$. De variantie ($\text{Var}[X]$) is gelijk aan $\frac{r}{\lambda^2}$.

Gamma-verdeling

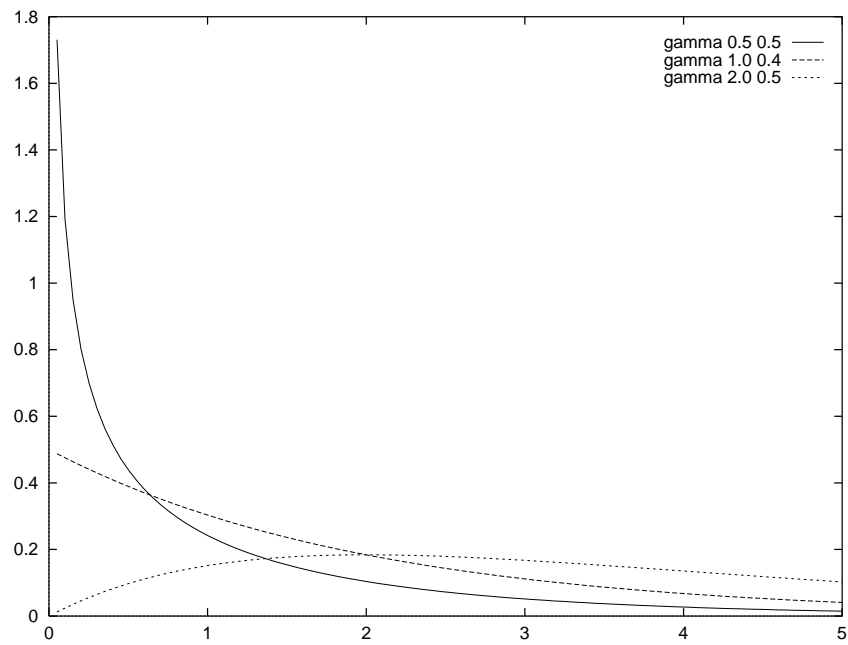
r, λ

Modus

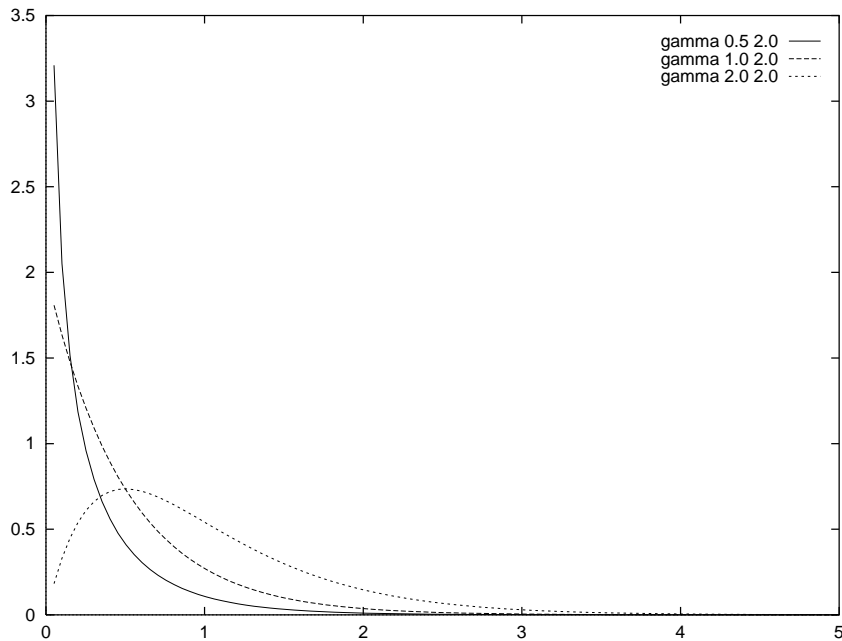
De *modus* van een kansverdeling is de waarde waarvoor de kansdichtheid maximaal is, d.w.z. m is de modus van een X indien voor elke x : $f_X(m) \geq f_X(x)$. De modus van een gamma-verdeelde stochast X hangt af van r



Figuur 6: De gamma-verdeling met parameter $\lambda = 1$ en r achtereenvolgens 1, 0.5 en 2.



Figuur 7: De gamma-verdeling met parameter $\lambda = 0.5$ en r achtereenvolgens 1, 0.5 en 2.



Figuur 8: De gamma-verdeling met parameter $\lambda = 2$ en r achtereenvolgens 1, 0.5 en 2.

en λ . Indien $r \leq 1$, dan is 0 de meest waarschijnlijke waarde, d.w.z. de modus van X is gelijk aan 0. Indien $r > 1$, dan is de modus gelijk aan $\frac{r-1}{\lambda}$. Een speciaal geval van de gamma-verdeling is de negatief-exponentiële verdeling ($r = 1$).

3.2.10 Erlang-verdeling

Erlang-verdeling

r, λ

De *Erlang-verdeling* is een speciaal geval van de gamma-verdeling. Een gamma-verdeling waarbij de parameter r geheeltallig is, noemen we ook wel een Erlang-verdeling. Een stochast die Erlang verdeeld is met parameters r (geheeltallig) en λ kunnen we beschouwen als de som van k onafhankelijke, negatief-exponentieel verdeelde stochasten met parameter λ . Stel X is Erlang-verdeeld met parameters r en λ , dan:

$$f_X(x) = \frac{\lambda(\lambda x)^{r-1} e^{-\lambda x}}{(r-1)!}$$

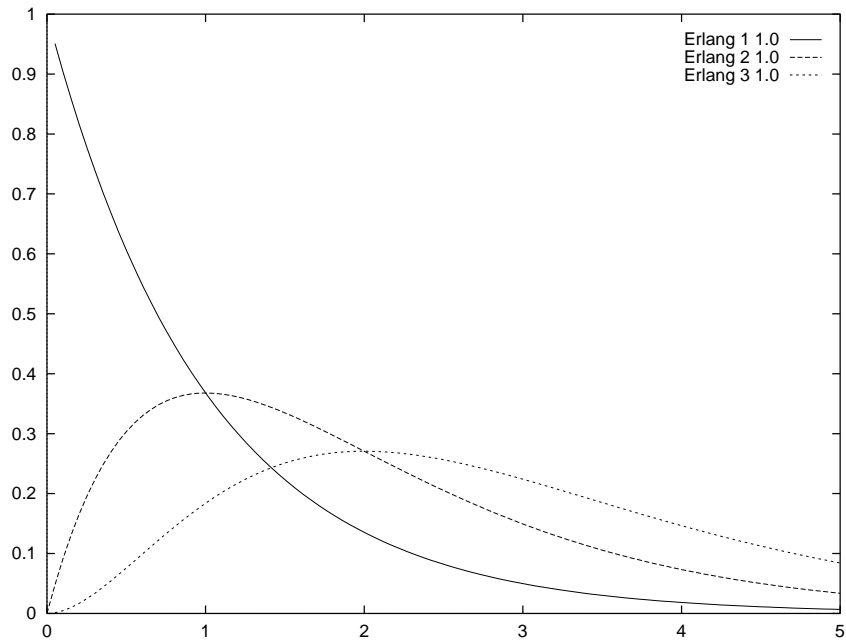
In figuur 9 is voor een aantal waarden van r en λ de functie f_X weergegeven. De verwachting is gelijk aan $\frac{r}{\lambda}$. De variantie is gelijk aan $\frac{r}{\lambda^2}$.

3.2.11 χ^2 -verdeling

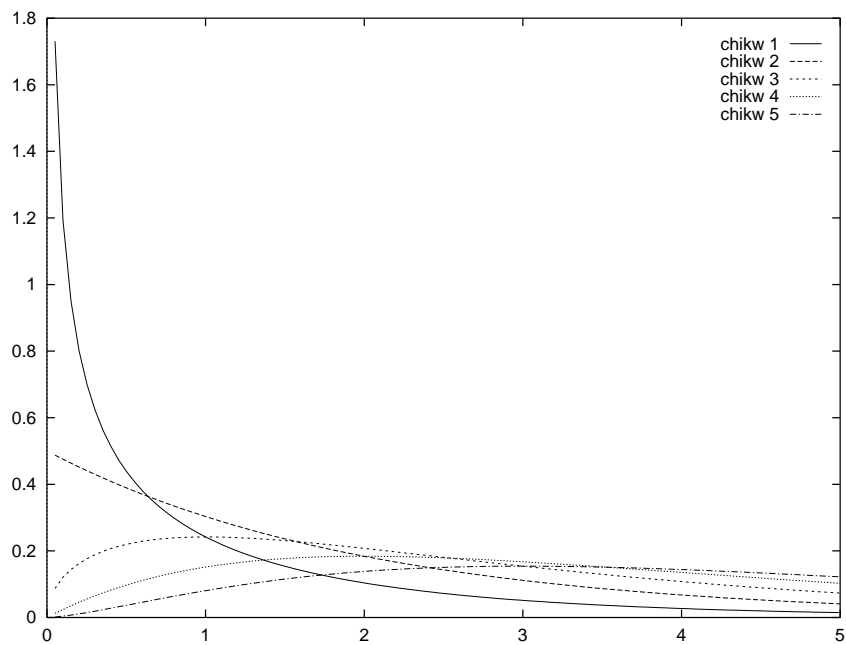
χ^2 -verdeling

Vrijheidsgraden v

De *χ^2 -verdeling* is ook een speciaal geval van de gamma-verdeling. Een χ^2 -verdeling heeft maar één parameter v . Deze v is een positief geheel getal en stelt het aantal *vrijheidsgraden* voor. Een stochast X die χ^2 -



Figuur 9: De Erlang-verdeling met parameter $\lambda = 2$ en r achtereenvolgens 1, 2 en 3.



Figuur 10: De χ^2 -verdeling met $v = 1, v = 2, v = 3, v = 4$ en $v = 5$.

verdeeld is met v vrijheidsgraden komt overeen met een stochast die gamma-verdeeld is met parameters $r = \frac{v}{2}$ en $\lambda = \frac{1}{2}$. In figuur 10 is voor een aantal waarden van v de kansdichtheid weergegeven. De verwachting is gelijk aan v . De variantie is gelijk aan $2v$. Er is ook een verband tussen de normale verdeling en de χ^2 -verdeling. Laat X_1, X_2, \dots, X_n onderling onafhankelijke standaard-normaal verdeeld stochasten zijn. In dat geval is de stochast $X = X_1^2 + X_2^2 + \dots + X_n^2$ precies χ^2 -verdeeld met parameter $v = n$. De χ^2 -verdeling wordt vooral gebruikt bij ‘goodness-of-fit’-tests.

3.2.12 Beta-verdeling

Beta-verdeling

a, b, r, s

De *beta-verdeling* is, net zoals de uniforme verdeling, verdeeld over een beperkt interval. We gebruiken deze dus op het moment dat we een ondergrens en bovengrens aan kunnen geven. Een stochast X die beta-verdeeld is heeft vier parameters a, b, r en s . De parameters a en b ($a < b$) stellen de onder- en bovengrens van de verdeling voor. De parameters r ($r > 0$) en s ($s > 0$) bepalen de vorm van de kansverdeling. Voor elke x ($a \leq x \leq b$) is de kansdichtheid als volgt gedefinieerd:

$$f_X(x) = \frac{1}{b-a} \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \left(\frac{x-a}{b-a}\right)^{r-1} \left(\frac{b-x}{b-a}\right)^{s-1}$$

Standaard-beta-verdeling

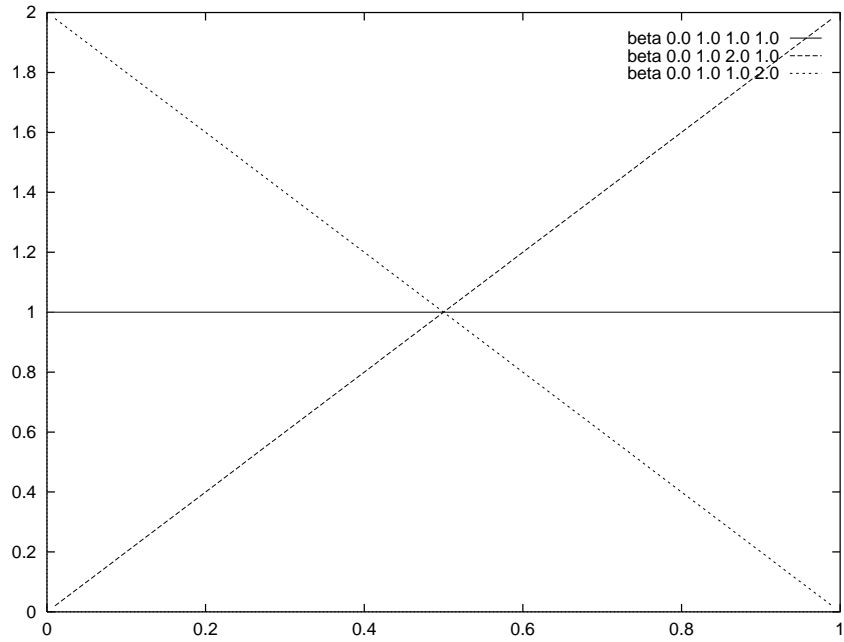
Indien $a = 0$ en $b = 1$ spreken we over een *standaard-beta-verdeling*. In de figuren 11, 12 en 13 zijn, uitgaande van een standaard-beta-verdeling, voor een aantal waarden van r en s de bijbehorende kansdichtheden afgebeeld. Zoals we in deze figuren kunnen zien, is de beta-verdeling een zeer veelzijdige kansverdeling. Indien r en s gelijk aan 1 zijn is X homogeen verdeeld met parameters a en b . Hoe groter r ten opzichte van s is, hoe groter de kans op een waarde die dicht bij b ligt. Hoe groter s ten opzichte van r is, hoe groter de kans op een waarde die dicht bij a ligt. Indien r kleiner dan 1, dan is de kansdichtheid dicht bij a groot. Indien s kleiner dan 1, dan is de kansdichtheid dicht bij b groot. Indien r en s beiden groter dan 1 zijn, ligt de modus (d.w.z. maximale kansdichtheid) ergens tussen a en b . De verwachting is gelijk aan:

$$\mathbb{E}[X] = a + (b-a) \frac{r}{r+s}$$

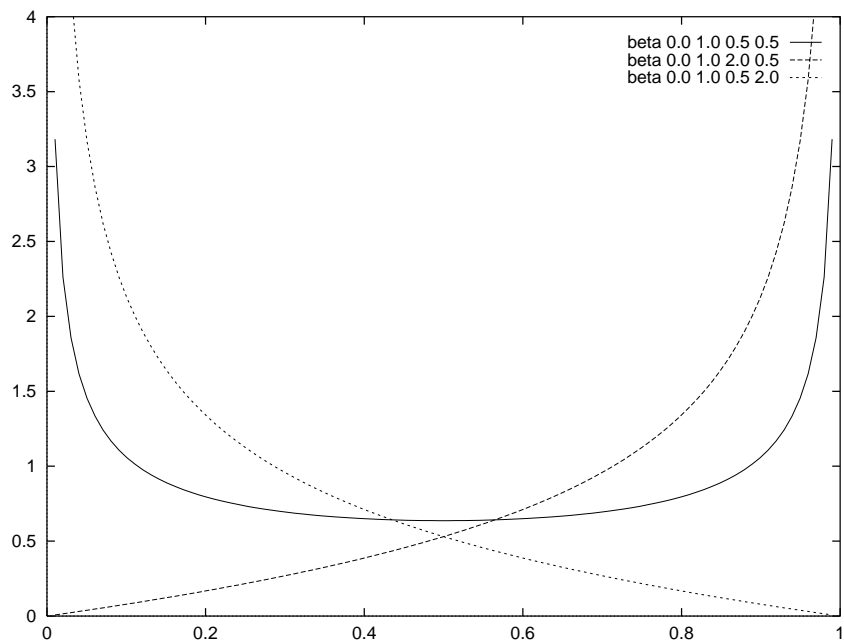
De variantie is gelijk aan:

$$\text{Var}[X] = \frac{rs(b-a)^2}{(r+s)^2(r+s+1)}$$

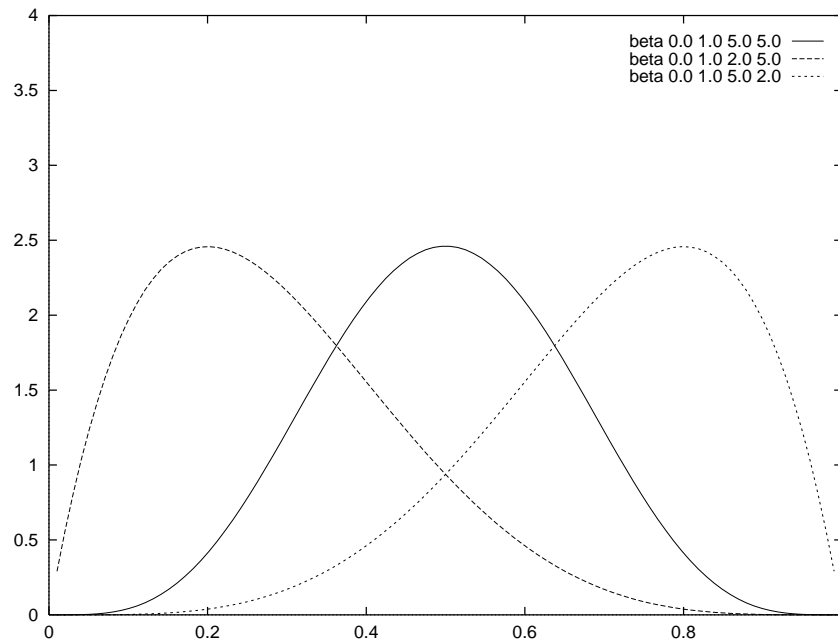
Een beta-verdeling is ‘scheef’ (d.w.z. niet symmetrisch) op het moment dat $r \neq s$. Het is daarom ook zeker niet altijd het geval dat de meest waarschijnlijke waarde (d.w.z. de modus) gelijk is aan de verwachting. Indien $r > 1$ en $s > 1$ is de modus gelijk aan $a + (b-a) \frac{r-1}{r+s-2}$. Indien



Figuur 11: De beta-verdeling met parameter $a = 0$ en $b = 1$ en r achtereenvolgens 1.0, 2.0 en 1.0, en s achtereenvolgens 1.0, 1.0 en 2.0.



Figuur 12: De beta verdeling met parameter $a = 0$ en $b = 1$ en r achtereenvolgens 0.5, 2.0 en 0.5, en s achtereenvolgens 0.5, 0.5 en 2.0.



Figuur 13: De beta verdeling met parameter $a = 0$ en $b = 1$ en r achtereenvolgens 5.0, 2.0 en 5.0, en s achtereenvolgens 5.0, 5.0 en 2.0.

$r < 1$, dan is er bij a sprake van een maximum. Is $s < 1$, dan wordt er een maximale kansdichtheid bereikt bij b .

Stel we willen een standaard-beta-verdeling met een verwachting μ en een variantie σ^2 . In dat geval moeten we r en s als volgt kiezen:

$$r = \frac{\mu^2(1 - \mu)}{\sigma^2} - \mu$$

$$s = \frac{(1 - \mu)^2\mu}{\sigma^2} - (1 - \mu)$$

Dit is alleen mogelijk indien voldaan is aan de volgende voorwaarde:

$$\sigma^2 < \mu(1 - \mu)$$

Omdat elke trekking tussen 0 en 1 ligt, is de variantie naar boven begrensd (kleiner dan 0.25).

Om een stochast Y te construeren met ondergrens a , bovengrens b , verwachting μ en variantie σ^2 , kan men uitgaan van een stochast X welke standaard beta-verdeeld is met verwachting $\frac{\mu-a}{b-a}$ en variantie $\frac{\sigma^2}{(b-a)^2}$. Stochast Y kunnen we dan als volgt definiëren: $Y = (b - a)X + a$.

PERT

Een bekende toepassing van de beta-verdeling is het gebruik van deze verdeling in *PERT* (Program Evaluation and Review Technique). *PERT* is een techniek die door projectmanagers gebruikt wordt om uitspraken te doen over de doorlooptijd van een project. *PERT* houdt van iedere activiteit drie gegevens bij over de duur van deze activiteit:

- (i) een optimistische schatting (d.w.z. een ondergrens),
- (ii) een pessimistische schatting (d.w.z. een bovengrens),
- (iii) een schatting van de meest waarschijnlijke tijdsduur (modus).

Indien we de duur van een zo'n activiteit modelleren door middel van een beta-verdeling, gebruiken we de eerste twee gegevens om de parameters a en b vast te stellen. Indien c de meest waarschijnlijke waarde is, dan worden de parameters r en s zodanig ingesteld dat de verwachting en variantie de volgende waarden aannemen:

$$\begin{aligned}\mu &= \frac{a + 4c + b}{6} \\ \sigma^2 &= \frac{(b - a)^2}{36}\end{aligned}$$

Gegeven een onder- en bovengrens ligt de variantie dus vast.

4 Verwerken van de resultaten

Omdat we simulatie willen gebruiken om inzicht te krijgen in de huidige of toekomstige situatie, moeten we tijdens de simulatie bepaalde zaken meten. Stel dat een bank de aanschaf van een nieuwe geldautomaat overweegt. Door middel van simulatie wil men inzicht krijgen in de wachttijden en het storingsgedrag. In de simulatie moeten we dus wachttijden en storingen registreren. In deze sectie laten we zien hoe we op basis van een simulatiestudie uitspraken kunnen doen over dit soort grootheden.

4.1 Gemiddelde en variantie

Waarneming

Tijdens de simulatie worden er steeds *waarnemingen* gedaan. We kunnen hierbij bijvoorbeeld denken aan een wachttijd, een doorlooptijd, een bewerkingsduur, een voorraadhoogte of een aantal.

voorbeeld

Een speler werpt 10 keer met een dobbelsteen. Achtereenvolgens worden de volgende waarnemingen gedaan: 3, 4, 6, 1, 1, 2, 5, 3, 3 en 2. In dit geval komt een waarneming dus overeen met het aantal ogen van de dobbelsteen.

In het algemeen is men niet geïnteresseerd in afzonderlijke waarnemingen. Meestal wil men een simulatie gebruiken om een indruk te krijgen van een bepaalde grootheid. Men wil bijvoorbeeld een uitspraak doen over de gemiddelde doorlooptijd van een order.

Stel dat we k opeenvolgende waarnemingen doen. Deze noemen we x_1, x_2, \dots, x_k . Deze waarnemingen noemen we ook wel een *steekproef*.

Steekproefgemiddelde

Het *gemiddelde* van een aantal waarnemingen noemen we ook wel het *steekproefgemiddelde*. Het steekproefgemiddelde van x_1, x_2, \dots, x_k noteren we door \bar{x} . \bar{x} vinden we door de waarnemingen op te tellen en te delen door k :

$$\bar{x} = \frac{\sum_{i=1}^k x_i}{k}$$

Merk op dat het steekproefgemiddelde een schatting van het werkelijke gemiddelde is.

Steekproefvariantie

De *variantie* van een aantal waarnemingen noemen we ook wel de *steekproefvariantie*. Deze variantie is een maat voor de afwijking van het gemiddelde. Hoe kleiner de variantie hoe dichter de waarnemingen bij het gemiddelde zullen liggen. De steekproefvariantie s^2 vinden we met behulp van de volgende formule:

$$s^2 = \frac{\sum_{i=1}^k (x_i - \bar{x})^2}{k - 1}$$

Deze formule kunnen we herschrijven tot:

$$s^2 = \frac{(\sum_{i=1}^k x_i^2) - \frac{1}{k} (\sum_{i=1}^k x_i)^2}{k - 1}$$

Steekproefstandaardafwijking

Indien we tijdens een simulatie (1) het aantal waarnemingen k , (2) de som van alle waarnemingen $\sum_{i=1}^k x_i$ en (3) de som van de kwadraten van alle waarnemingen $\sum_{i=1}^k x_i^2$ bijhouden, kunnen we het steekproefgemiddelde en de steekproefvariantie vaststellen. De wortel van de steekproefvariantie $s = \sqrt{s^2}$ wordt ook wel de *steekproefstandaardafwijking* genoemd. Over het algemeen geeft de standaardafwijking s een betere indruk van de afwijkingen van het gemiddelde dan de steekproefvariantie s^2 .

Mediaan

Behalve het gemiddelde en de variantie van een steekproef, kunnen we ook spreken over de *mediaan* van k waarnemingen x_1, x_2, \dots, x_k . De mediaan is de waarde van de waarneming die in het ‘midden’ ligt nadat we de waarnemingen op waarde gesorteerd hebben. Merk op dat we alleen kunnen spreken over de ‘middelste waarneming’ indien het aantal waarnemingen oneven is. In het geval van een even aantal waarnemingen nemen we het gemiddelde van de twee waarnemingen in het midden. Om na afloop van een simulatie-experiment de mediaan te berekenen, moeten we alle waarnemingen opslaan en sorteren.

voorbeeld

In een simulatie-experiment worden de volgende wachttijden geregistreerd: 2.3, 3.4, 2.2, 2.8, 5.6, 3.2, 6.8, 3.2, 5.3 en 2.1. Uitgaande van de steekproef kunnen we het steekproefgemiddelde en de steekproefvariantie bepalen. Het steekproefgemiddelde is 3.69 en de steekproefvariantie is 2.648. De mediaan is 3.2.

4.2 Subruns en voorlooprun

Gegeven een simulatie-experiment, kunnen we op een vrij eenvoudige wijze vaststellen wat het steekproefgemiddelde en de steekproefvariantie van een bepaalde grootte zijn. Het steekproefgemiddelde kunnen we als schatting gebruiken voor de verwachte waarde van deze grootte (bijvoorbeeld wachttijd). Helaas is op basis van één enkel getal voor het steekproefgemiddelde niet af te leiden wat de betrouwbaarheid van het systeem is. Dit is de reden dat een simulatie-experiment bestaat uit een aantal deel-experimenten (subruns) die het mogelijk maken om uitspraken te doen over de betrouwbaarheid van de simulatieresultaten.

4.2.1 Noodzaak

voorbeeld

Laten we als voorbeeld een postkantoor nemen met één loket. Klanten komen het postkantoor volgens een Poisson-proces met intensiteit 8 binnen. De tijd tussen twee opeenvolgende klanten is dus negatief-exponentieel verdeeld met een gemiddelde van 0.125 uur (7.5 minuten). Ook de bedieningstijd is negatief-exponentieel verdeeld. De gemiddelde bedieningstijd is 0.1 uur (6 minuten). De vraag is nu wat de gemiddelde wachttijd van klanten is.

Op basis van een simulatie-experiment waarin de wachttijden van 1000 opeenvolgende klanten geregistreerd vinden we een gemiddelde wachttijd van bijvoorbeeld 0.21 uur (ongeveer 12 minuten). (Het betreft hier het resultaat van een daadwerkelijk uitgevoerd simulatie-experiment.) De verwachte wachttijd kunnen we in dit geval echter ook berekenen door middel van een wiskundige formule. (Het betreft hier een M/M/1-wachtrij, zie appendix C.) Deze geeft aan dat de verwachte wachttijd gelijk is aan 0.15 uur (9 minuten). De schatting gebaseerd op het simulatie-experiment wijkt dus behoorlijk af van de werkelijke te verwachten wachttijd.

We hebben dus behoefte aan een mechanisme om erachter te komen wat de betrouwbaarheid van een bepaald resultaat is. Men zou kunnen denken dat de steekproefvariantie hier een direct antwoord op kan geven. Dit is echter niet het geval omdat de steekproefvariantie op basis van de wachttijden van 1000 opeenvolgende klanten slechts een schatting is van de gemiddelde afwijking van de gemiddelde wachttijd. Een manier om dit wel voor elkaar te krijgen is de introductie van *subruns*. In plaats van één lange simulatierun voeren we een aantal kleinere simulatieruns uit waarvan we de resultaten met elkaar vergelijken.

Subruns

Stel dat we in de plaats van één lange simulatierun met 1000 klanten, 10 opeenvolgende subruns met elk 100 klanten uitgevoerd hadden. In tabel 1 is voor elk van deze subruns een afzonderlijk steekproefgemiddelde weergegeven. Het gemiddelde over deze 10 subruns is weer ongeveer

0.21 uur. Deze tabel laat duidelijk zien dat er grote verschillen zijn tussen de diverse steekproefgemiddelden. De geschatte wachttijd loopt uiteen van 0.12 uur tot 0.31 uur. Op basis van deze gegevens kunnen we dus geen nauwkeurige schatting van de te verwachten wachttijd maken. Door het simulatie-experiment op te delen in subruns hebben we dus een indruk gekregen van de betrouwbaarheid. Later zullen we zien dat we de betrouwbaarheid ook kunnen kwantificeren.

subrun nummer	gemiddelde wachttijd
1	0.16
2	0.14
3	0.29
4	0.12
5	0.26
6	0.31
7	0.13
8	0.28
9	0.21
10	0.28

Tabel 1: Het steekproefgemiddelde uitgesplitst per subrun.

4.2.2 Subruns en aanloopverschijnselen

In plaats van één lange simulatierun maken we dus gebruik van een aantal kleinere subruns. We moeten hierbij echter onderscheid maken tussen twee situaties:

Stabiele situatie

- (i) Een *stabiele situatie*, d.w.z. de omstandigheden zijn constant. Het aankomstpatroon van bijvoorbeeld klanten is gedurende de hele simulatie hetzelfde. Het is dus niet zo dat er structurele pieken zijn. Ook zijn er geen opstart-effecten: we zijn geïnteresseerd in de *evenwichtssituatie* van een proces.

Instabiele situatie

- (ii) Een *instabiele situatie* in de zin dat de omstandigheden gedurende simulatie structureel veranderen. Er is bijvoorbeeld sprake van leegloop aan het begin en einde van de simulatie. Meestal is er sprake van een duidelijk begin en einde. Er wordt in dit verband ook wel gesproken over een *terminerend proces*.

Indien we de gemiddelde wachttijd van een klant in het postkantoor op een bepaald tijdstip op de dag willen analyseren, kunnen we spreken

van een stabiele situatie. We veronderstellen dat het aankomstproces en het bedieningsproces op dit tijdstip van de dag bepaalde kenmerken hebben. Willen we de gemiddelde wachttijd van een klant over een gehele dag analyseren, dan hebben we te maken met een instabiele situatie. Aan het einde van de dag hebben we wellicht te maken met een ander aankomstpatroon dan om 10 uur 's morgens.

De reden dat we onderscheid maken tussen stabiele en instabiele situaties is het feit dat dit van invloed is op de opzet van het simulatie-experiment.

In geval van een stabiele situatie

Indien we te maken hebben met een stabiele situatie, dan kunnen we de subruns direct na elkaar uitvoeren. Elke subrun, behalve de eerste, start in de toestand die de vorige subrun heeft achtergelaten. Dit betekent dat we subruns kunnen krijgen door één lange simulatierun uit te voeren en deze in gelijke delen (subruns) op te hakken. Van elke subrun hoeven we alleen de relevante resultaten te bewaren. De eerste run van zo'n simulatie-experiment moeten we zorgvuldig behandelen. De keuze van het seed heeft bijvoorbeeld een directe invloed op het eerste meetresultaat. Ook kan er sprake zijn van *aanloopverschijnselen*. We spreken over aanloopverschijnselen indien de resultaten beïnvloed worden door de gekozen beginsituatie. Als we starten met een leeg postkantoor, dan weten we dat de wachttijd van de eerste klant gelijk zal zijn aan 0. In dit geval is sprake van een beperkt effect. Er zijn echter ook situaties denkbaar waar de gekozen begintoestand nog lang invloed heeft op bepaalde metingen. Indien we bij het simuleren van een fabricage-afdeling een lege begintoestand (d.w.z. zonder onderhanden werk) kiezen, zal het enige tijd duren voordat de gemeten doorlooptijden representatief zijn voor de normale gang van zaken.

Aanloopverschijnselen

Voorlooprun

Om het effect van aanloopverschijnselen te minimaliseren starten we met een speciale subrun die we *voorlooprun* (ook wel aanlooprun genoemd) zullen noemen. De resultaten verkregen tijdens deze run nemen we niet mee in het schatten van een bepaalde grootte. De voorlooprun moet lang genoeg zijn om ervoor te zorgen dat de aanloopverschijnselen verdwenen zijn. In het geval van de simulatie van een fabricage-afdeling kunnen we de voorlooprun afsluiten op het moment dat de voorraad onderhanden werk een stabiel niveau heeft bereikt.

In geval van een instabiele situatie

Op het moment dat we te maken hebben met een instabiele situatie, dan kunnen we de subruns niet meer direct na elkaar uitvoeren. Eventuele aanloopverschijnselen maken nu een wezenlijk onderdeel uit van de simulatie. Elke subrun moet daarom starten vanuit een gelijke begintoestand. Omdat er voor elke subrun gebruik gemaakt wordt van een ander seed, zijn de resultaten van de diverse subruns toch onafhankelijk en kunnen tijdens de analyse met elkaar vergeleken worden. Indien we de wachttijden in een postkantoor over de gehele dag willen analyseren, dan komt

elke subrun overeen met één dag. Elk van deze subruns start vanuit de toestand dat er geen klanten in het postkantoor zijn, en eindigt nadat er aan het einde van de dag geen klanten meer in het postkantoor aanwezig zijn.

Relevante vragen tijdens het opzetten van een simulatie-experiment zijn:

- Is er een voorlooprun nodig?
- Hoe lang moet de voorlooprun zijn?
- Hoeveel subruns zijn er nodig?
- Hoelang dient elke subrun te zijn?

Is er een voorlooprun nodig?

Hoe lang moet de voorlooprun zijn?

Hoeveel subruns zijn er nodig?

Hoelang dient elke subrun te zijn?

Laten we deze vragen eens proberen te beantwoorden. Er is alleen maar een voorlooprun nodig indien we te maken hebben met de simulatie van een stabiele situatie waarbij de begintoestand afwijkt van doorsnee toestanden die tijdens de simulatie aangedaan worden. De lengte van een voorlooprun hangt af van de mate waarin de begintoestand afwijkt van een doorsnee toestand en de snelheid waarmee de simulatie naar een stabiele situatie gaat. We kunnen een aantal relevante grootheden die tijdens de simulatie gemeten worden in een grafiek uitzetten tegen de tijd en aan de hand van deze grafiek inschatten of een stabiele toestand is bereikt. Over het aantal benodigde subruns is moeilijk iets te zeggen. Het aantal hangt bijvoorbeeld sterk af van de gewenste betrouwbaarheid van de uiteindelijke resultaten en de lengte van elke subrun. Hier komen we later in meer detail op terug. De lengte van elke subrun ligt in het geval van een instabiele situatie meestal vast. De lengte is precies één dag, één ochtend of de tijd die nodig is om één verzoek volledig af te handelen. In het geval van een stabiele situatie moeten we ervoor zorgen dat elke subrun lang genoeg is om ervoor te zorgen dat de begintoestand van de ene subrun geheel onafhankelijk is van de begintoestand van de volgende subrun. Stel dat er aan het begin van een subrun sprake is van een enorme wachtrij. Indien we de subrunlengte niet lang genoeg kiezen is deze wachtrij in de daarop volgende subrun nog niet weggewerkt. In dat geval zijn de resultaten van de twee subruns verre van onafhankelijk, iets wat kan leiden tot een foute interpretatie van de resultaten. Een mogelijke vuistregel is dat er in elke subrun ten minste één regeneratiepunt bereikt wordt. Een regeneratiepunt is een bepaalde toestand waar het systeem om de zoveel tijd weer naar terugkeert. Een voorbeeld van een regeneratiepunt voor de simulatie van het postkantoor is de toestand waarbij er geen enkele klant binnen is.

4.3 Analyse van subruns

Stel we hebben n subruns uitgevoerd en voor elke subrun i een bepaald resultaat x_i geregistreerd. De waarde x_i is bijvoorbeeld de gemeten gemiddelde wachttijd in subrun i . Deze waarde kan echter ook de gemiddelde variantie van de gemeten wachttijden zijn. Ook de gemiddelde bezettingsgraad en de gemiddelde wachtrijslengte behoren tot de mogelijkheden. Het enige wat we weten van de waarden x_i is het feit dat ze onderling onafhankelijk zijn. (Dit hebben we bijvoorbeeld afgedwongen door de subrunlengte lang genoeg te kiezen.) Gegeven de resultaten x_1, x_2, \dots, x_n kunnen we ook hier weer spreken over het steekproefgemiddelde \bar{x} :

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

en de steekproefvariantie s^2 :

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

Let op!

Merk op dat we het steekproefgemiddelde en de steekproefvariantie over de resultaten van de diverse subruns niet moeten verwarren met het gemiddelde en de variantie van een aantal metingen *binnen één subrun!* We kunnen \bar{x} als een schatting van de waarde die we willen analyseren beschouwen. De gemiddelde afwijking van deze schatting $(\frac{s}{\sqrt{n}})^1$ geeft een indruk van deze betrouwbaarheid van deze schatting. Indien $\frac{s}{\sqrt{n}}$ klein is, is \bar{x} een goede schatting.

4.3.1 De situatie met meer dan 30 subruns

Indien er sprake is van een groot aantal subruns, kunnen we de schatting \bar{x} (vanwege de centrale limietstelling) als normaal verdeeld beschouwen. Daarom behandelen we de situatie met meer dan 30 subruns als een apart geval.

Wanneer stoppen we met het genereren van subruns?

Het feit dat $\frac{s}{\sqrt{n}}$ een maat is voor de kwaliteit van \bar{x} als een schatting van de waarde die we willen analyseren, kunnen we gebruiken om vast te stellen wanneer we kunnen stoppen met het genereren van subruns.

- (i) Kies een waarde d voor de toegestane standaardafwijking van de geschatte waarde \bar{x} .
- (ii) Genereer ten minste 30 subruns, en registreer per subrun de waarde x_i .

¹De variantie van de schatting \bar{x} is $\text{Var}[\bar{x}] = \text{Var}[\frac{1}{n} \sum_{i=1}^n x_i] = \frac{\sigma^2}{n}$. De standaardafwijking van \bar{x} is dus $\frac{\sigma}{\sqrt{n}}$. Omdat s een goede schatting van σ is, is $\frac{s}{\sqrt{n}}$ een goede schatting voor de standaardafwijking van \bar{x} .

- (iii) Genereer additionele subruns totdat $\frac{s}{\sqrt{n}} \leq d$, waar s steekproefstandaardafwijking is en n het aantal subruns dat is uitgevoerd.
- (iv) Het steekproefgemiddelde \bar{x} is nu een schatting van de te onderzoeken grootheid.

Er zijn twee redenen waarom er in dit geval ten minste 30 subruns uitgevoerd moeten worden. In de eerste plaats is \bar{x} pas bij een aanzienlijk aantal subruns bij benadering normaal verdeeld. Dit is een dwingende reden om ervoor te zorgen dat er ten minste 30 onderling onafhankelijke subruns zijn. Een andere reden om te kiezen voor een voldoende aantal subruns is het feit dat s als schatting van de werkelijke standaardafwijking bij meer subruns steeds beter wordt.

Betrouwbaarheidsinterval

Gegeven een groot aantal onafhankelijke subruns, kunnen we ook een *betrouwbaarheidsinterval* voor de te onderzoeken grootheid opstellen. Doordat \bar{x} het gemiddelde van een groot aantal onafhankelijke metingen is, mogen we aannemen dat \bar{x} bij benadering normaal verdeeld is (zie appendix A). Op basis van dit gegeven kunnen we afleiden dat de werkelijke waarde van de te onderzoeken grootheid met een bepaalde kans in een zogenaamd betrouwbaarheidsinterval ligt. De werkelijke waarde noteren we met μ . (μ is bijvoorbeeld niet de gemeten gemiddelde wachttijd maar de werkelijke gemiddelde wachttijd.) Gegeven het steekproefgemiddelde \bar{x} en de steekproefstandaardafwijking s voldoet de werkelijke waarde μ met betrouwbaarheid $(1 - \alpha)$ aan de volgende vergelijking:

$$\bar{x} - \frac{s}{\sqrt{n}} z\left(\frac{\alpha}{2}\right) < \mu < \bar{x} + \frac{s}{\sqrt{n}} z\left(\frac{\alpha}{2}\right)$$

waarbij $z\left(\frac{\alpha}{2}\right)$ als volgt gedefinieerd is. Laat Z een standaard-normaal-verdeelde stochast zijn, dan is $\mathbb{P}[Z > z(x)] = x$. Voor een aantal waarden van x is $z(x)$ in tabel 2 weergegeven. De waarde α geeft de onbetrouwbaarheid aan, d.w.z. de kans dat μ niet voldoet aan de vergelijking. Typische waarden voor α lopen uiteen van 0.001 tot 0.100. Het interval

$$\left[\bar{x} - \frac{s}{\sqrt{n}} z\left(\frac{\alpha}{2}\right), \bar{x} + \frac{s}{\sqrt{n}} z\left(\frac{\alpha}{2}\right) \right]$$

noemen we ook wel het $(1 - \alpha)$ -betrouwbaarheidsinterval voor de geschatte waarde μ .

voorbeeld

Het voorgaande illustreren we aan de hand van het volgende voorbeeld. Een bedrijf maakt zich zorgen over de bezettingsgraad van de helpdesk. Deze is zo hoog geworden dat het ziekteverzuim sterk is toegenomen. Om deze situatie te onderzoeken is gekozen voor een simulatiestudie om vast te stellen hoe dat de bezettingsgraad verlaagd kan worden. Om de bezettingsgraad van de huidige situatie te controleren is

x	z(x)
0.001	3.090
0.005	2.576
0.010	2.326
0.050	1.645
0.100	1.282

Tabel 2: $\mathbb{P}[Z > z(x)] = x$ waarbij Z standaard-normaal-verdeeld is.

een simulatie-experiment bestaande uit 30 subruns uitgevoerd. Elke subrun komt overeen met één werkdag. De gemiddelde bezettingsgraad per subrun is weergegeven in tabel 3. Het steekproefgemiddelde is 0.9408

subrun nummer	gemiddelde bezettingsgraad	subrun nummer	gemiddelde bezettingsgraad	subrun nummer	gemiddelde bezettingsgraad
1	0.914	11	0.894	21	0.898
2	0.964	12	0.962	22	0.912
3	0.934	13	0.973	23	0.943
4	0.978	14	0.984	24	0.953
5	0.912	15	0.923	25	0.923
6	0.956	16	0.932	26	0.914
7	0.958	17	0.967	27	0.923
8	0.934	18	0.924	28	0.936
9	0.978	19	0.945	29	0.945
10	0.976	20	0.936	30	0.934

Tabel 3: De gemiddelde bezettingsgraad per subrun.

en de steekproefvariantie is 0.000617. Alle gegevens die nodig om een $(1 - \alpha)$ -betrouwbaarheidsinterval om te stellen zijn dus bekend: $n = 30$, $\bar{x} = 0.9408$, $s^2 = 0.000617$ en dus $s = 0.02485$. Indien we α gelijk aan 0.010 nemen vinden we het volgende betrouwbaarheidsinterval:

$$\left[0.9408 - \frac{0.02485}{\sqrt{30}} z\left(\frac{0.010}{2}\right), 0.9408 + \frac{0.02485}{\sqrt{30}} z\left(\frac{0.010}{2}\right) \right]$$

Aangezien $z(0.005) = 2.576$ is dit dus het interval $[0.9291, 0.9525]$. Hoe groter de onbetrouwbaarheid α , hoe kleiner het bijbehorende betrouwbaarheidsinterval. Voor bijvoorbeeld $\alpha = 0.10$ vinden we het betrouwbaarheidsinterval $[0.9333, 0.9483]$. Op basis van de resultaten kunnen we dus veilig stellen dat de bezettingsgraad van de helpdesk vrij hoog is!

4.3.2 De situatie met minder dan 30 subruns

In sommige gevallen kunnen we ook volstaan met minder dan 30 subruns. Een eis die we dan moeten stellen is dat de resultaten voor de afzonderlijke subruns (x_i) bij benadering normaal verdeeld zijn. Indien het resultaat van een subrun x_i het gemiddelde is van een groot aantal waarnemingen, dan is (volgens de centrale limietstelling) x_i bij benadering normaal-verdeeld. Op het moment dat x_i dus de gemiddelde wachttijd, gemiddelde bedieningstijd of gemiddelde doorlooptijd van een groot aantal klanten is, is x_i bij benadering normaal-verdeeld. Door gebruik te maken van deze eigenschap, kunnen we gegeven n subruns met een steekproefgemiddelde \bar{x} , steekproefstandaardafwijking s en een betrouwbaarheid $(1 - \alpha)$ het volgende betrouwbaarheidsinterval afleiden:

$$\left[\bar{x} - \frac{s}{\sqrt{n}} t_{n-1}\left(\frac{\alpha}{2}\right), \bar{x} + \frac{s}{\sqrt{n}} t_{n-1}\left(\frac{\alpha}{2}\right) \right]$$

Student-verdeling

waarbij $t_v(x)$ de kritieke waarde is van een *Student-verdeling*, ook wel *t-verdeling* genoemd, met v vrijheidsgraden. In tabel 4 is voor een aantal waarden van v en x de kritieke waarde $t_v(x)$ opgenomen.

$t_v(x)$	$x =$			
	0.100	0.050	0.010	0.001
$v = 1$	3.08	6.31	31.82	318.31
2	1.89	2.92	6.96	22.33
3	1.64	2.35	4.54	10.21
4	1.53	2.13	3.75	7.17
5	1.48	2.02	3.37	5.89
6	1.44	1.94	3.14	5.21
7	1.41	1.89	3.00	4.79
8	1.40	1.86	2.90	4.50
9	1.38	1.83	2.82	4.30
10	1.37	1.81	2.76	4.14
15	1.34	1.75	2.60	3.73
20	1.33	1.72	2.53	3.55
25	1.32	1.71	2.49	3.45
50	1.30	1.68	2.40	3.26
100	1.29	1.66	2.35	3.17
∞	1.28	1.64	2.33	3.09

Tabel 4: De kritieke waarden voor een Student-verdeling met v vrijheidsgraden.

In tegenstelling tot de eerder besproken methode kunnen we het zojuist gegeven betrouwbaarheidsinterval ook toepassen in het geval dat we over

een beperkt aantal subruns (bijvoorbeeld 10) beschikken. Indien we beschikken over een groter aantal subruns en we ervan overtuigd zijn dat de resultaten per subrun normaal-verdeeld zijn, dan kunnen we het beste het zojuist geschetste $(1 - \alpha)$ -betrouwbaarheidsinterval toepassen. Voor grote n is het betrouwbaarheidsinterval op basis van $t_{n-1}(\frac{\alpha}{2})$ nauwkeuriger dan dat op basis van $z(\frac{\alpha}{2})$. Bij het bepalen van het betrouwbaarheidsinterval op basis van $t_{n-1}(\frac{\alpha}{2})$ wordt immers geen beroep gedaan op de centrale limietstelling ten aanzien van het aantal subruns.

Indien we op basis van de resultaten in tabel 2 een uitspraak doen over de betrouwbaarheid van \bar{x} , dan vinden we voor $\alpha = 0.100$ het volgende betrouwbaarheidsinterval:

$$\left[0.9408 - \frac{0.02485}{\sqrt{30}} t_{29}\left(\frac{0.100}{2}\right), 0.9408 + \frac{0.02485}{\sqrt{30}} t_{29}\left(\frac{0.100}{2}\right) \right]$$

Aangezien $t_{29}(0.050) = 1.699$ is dit dus het interval $[0.9331, 0.9485]$. Het interval is dus bij benadering hetzelfde als het 0.90-betrouwbaarheidsinterval dat we eerder al afgeleid hebben. Merk op dat we het betrouwbaarheidsinterval op basis van de t -verdeling alleen maar mogen gebruiken, indien we ervan overtuigd zijn dat de gemiddelde bezettingsgraad per subrun bij benadering normaal-verdeeld is. Zeker voor kleinere aantallen subruns is deze voorwaarde van groot belang!

4.4 Variantie-reductie

Met behulp van de zojuist beschreven technieken is het mogelijk om simulatieresultaten te analyseren. Er zijn echter nog vele geavanceerde technieken om met behulp van korte simulatieruns zoveel mogelijk informatie te verwerven. Zoals we zojuist gezien hebben is er een lineair verband tussen de lengte van het betrouwbaarheidsinterval en de standaardafwijking van de gemeten resultaten per subrun. Als de standaardafwijking twee keer zo groot is, is het bijbehorende betrouwbaarheidsinterval ook twee keer zo groot. Om het moment dat we de standaardafwijking tussen de subrun-resultaten kunnen verkleinen, zullen de grenzen van het betrouwbaarheidsinterval scherper worden. De standaardafwijking is de wortel van de variantie, daarom zal een verkleining van de variantie zorgen voor een aanscherping van het betrouwbaarheidsinterval. Er bestaan een groot aantal technieken die gericht zijn op het verlagen van deze variantie. Deze technieken worden ook wel *variantie-reducerende technieken* genoemd. Een aantal bekende technieken zijn:

- antithetische random getallen
- gemeenschappelijke random getallen

Variantie-reducerende technieken

- control variates
- conditioning
- stratified sampling
- importance sampling

Het voert te ver om al deze technieken volledig te behandelen. Daarom zullen we alleen de eerste twee kort toelichten.

Antithetische random getallen

In een simulatie-experiment worden doorlopend random getallen gebruikt. Door gebruik te maken van een goede random generator zijn deze getallen onafhankelijk van elkaar. Het is echter niet nodig om voor elke subrun een geheel nieuwe reeks random getallen te genereren. Indien r_1, r_2, \dots, r_n random getallen zijn, dan zijn $(1 - r_1), (1 - r_2), \dots, (1 - r_n)$ ook prima random getallen! Deze getallen noemen we ook wel antithetische random getallen. Indien we voor elke oneven subrun nieuwe random getallen genereren en voor elke even subrun de antithetische random getallen gebruiken, dan kunnen we dus volstaan met de helft van het aantal random getallen (indien het totale aantal subruns even is). Er is echter nog een andere prettige bijkomstigheid. De resultaten voor subrun $2k - 1$ en subrun $2k$ zijn meestal negatief gecorreleerd. Indien x_{2k-1} en x_{2k} de resultaten van twee opvolgende subruns voorstellen, dan is $\text{Cov}[x_{2k-1}, x_{2k}]$ naar alle waarschijnlijkheid kleiner dan nul. (Dit resultaat is gebaseerd op het feit dat de subrun meestal een monotone functie is ten opzichte van de verbruikte set random getallen.) Dit heeft tot gevolg dat de variantie van het gemiddelde van x_{2k-1} en x_{2k} kleiner wordt. Immers:

$$\text{Var}\left[\frac{X + Y}{2}\right] = \frac{1}{4}(\text{Var}[X] + \text{Var}[Y] + 2\text{Cov}[X, Y])$$

De totale steekproefvariantie zal daarom ook afnemen, wat een aanscherping van het betrouwbaarheidsinterval tot gevolg heeft.

Gemeenschappelijke random getallen

Als men een aantal (zeg 2) alternatieven met elkaar wil vergelijken, dan is het intuïtief duidelijk dat men dit het beste onder zoveel mogelijk dezelfde omstandigheden kan doen. Dit pleit ervoor om de simulatieruns van de alternatieven zoveel mogelijk te voorzien van dezelfde random getallen. Zo kan men in geval van het simuleren van verschillende inrichtingen van het postkantoor de diverse alternatieven voeden met dezelfde random getallen voor tussenaankomsttijden en bedieningstijden. In dat geval zal de variantie van het verschil tussen beide alternatieven waarschijnlijk aanzienlijk kleiner zijn.

Het moge duidelijk zijn dat er nog talloze technieken zijn om nog meer informatie uit een simulatie-experiment te persen. Het betreft hier meestal

geavanceerde technieken die zorgvuldig gebruikt moeten worden. Voor een verdere behandeling van deze technieken verwijzen we naar [2, 12, 13].

4.5 Gevoeligheidsanalyse

Door het zorgvuldig gebruik van subruns en het berekenen van betrouwbaarheidsintervallen kan men, gegeven een bepaalde situatie, vrij nauwkeurig bepalen wat de te verwachten wachttijden, bezettingsgraden, storingsfrequenties, enzovoorts zijn. Omdat deze resultaten gebaseerd zijn op een specifieke situatie, is het echter niet duidelijk hoe *gevoelig* de resultaten zijn. Indien we voor een aankomstproces met een gemiddelde tussenaankomsttijd van 5 minuten een geschatte gemiddelde wachttijd van 10 minuten vinden, wat is dan de gemiddelde wachttijd indien de tussenaankomsttijd niet 5 maar 6 minuten is? Een model bevat over het algemeen een aantal *parameters*; dit zijn instelbare grootheden als gemiddelde tussenaankomsttijd, gemiddelde bedieningstijd en gemiddelde responsetijd. Voor een simulatie-experiment wordt elke van deze grootheden op een bepaalde waarde gezet. Het betreft hier meestal een geschatte waarde, omdat de precieze waarde niet bekend is. Ook de gekozen kansverdeling zal alleen op hoofdlijnen overeenkomen met de werkelijke verdeling van waarden. Het is dus daarom van groot belang om te weten hoe gevoelig de resultaten zijn voor variaties in de instellingen van de modelparameters.

Gevoeligheid

Modelparameters

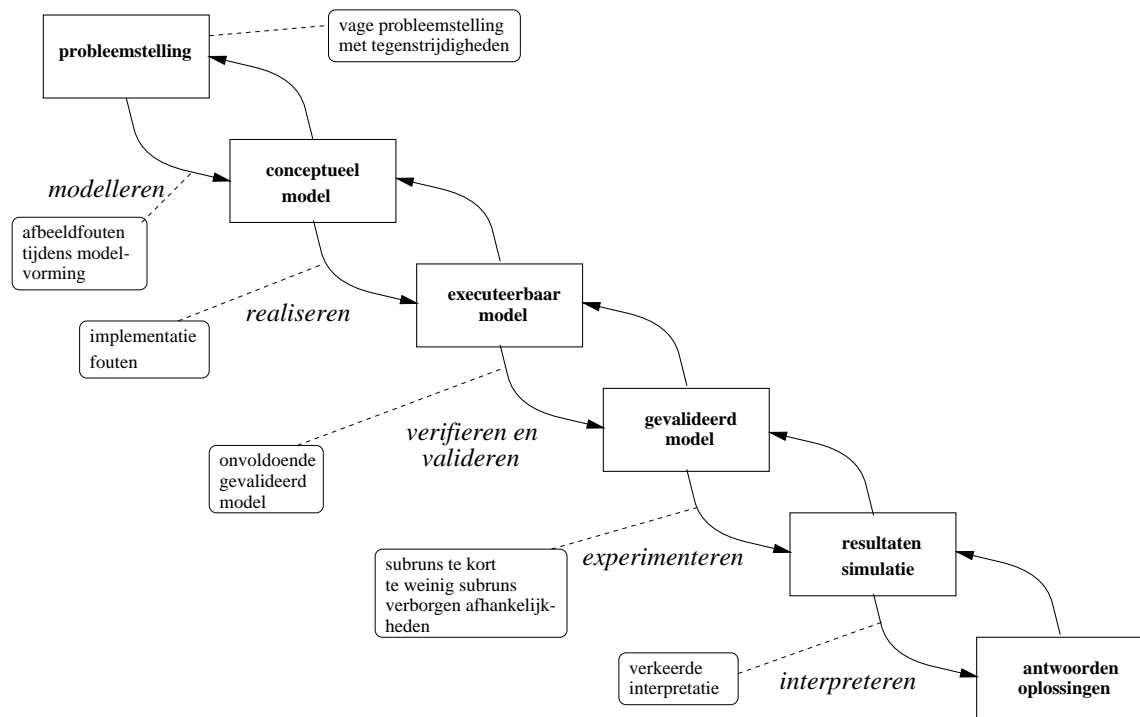
Gevoeligheidsanalyse

Door het uitvoeren van een *gevoeligheidsanalyse* krijgt men inzicht in afhankelijkheden tussen de modelparameters en de resultaten. Om de gevoeligheid te testen worden er een aantal experimenten uitgevoerd met licht afwijkende parameter-instellingen. Op deze manier krijgt men een indruk van de mate waarin kleine variaties het uiteindelijke resultaat kunnen beïnvloeden. Het kan zijn dat een iets aangepaste instelling van een bepaalde parameter weinig invloed heeft op het uiteindelijke resultaat. Het kan echter ook zijn een kleine aanpassing van deze parameter geheel andere resultaten tot gevolg heeft. Een proces met een hoge bezettingsgraad zal gevoeliger zijn voor fluctuaties in het aankomstproces, dan een proces met een lage bezetting. In dit verband spreken we ook wel over de *robuustheid* van een model. Een model is robuust indien licht afwijkende parameter instellingen nauwelijks invloed hebben op het uiteindelijke resultaat.

Robuustheid van een model

5 Valkuilen

Zoals in de introductie reeds is aangegeven, wordt simulatie veelal gebruikt ter ondersteuning van het maken van belangrijke strategische beslissingen. Daarom is het van groot belang dat er tijdens het uitvoeren van de simulatie niet allerlei fouten in het model en de interpretatie van de resultaten kunnen sluipen. Voordat we een aantal veel gemaakte fouten op een rij zetten, kijken we op op welke momenten in de life-cycle van een simulatiestudie er fouten geïntroduceerd kunnen worden. In figuur 14 is de fasering van een simulatiestudie en de mogelijke foutenbronnen weergegeven.



Figuur 14: De fasering van een simulatiestudie.

Mogelijke foutenbronnen

Reeds tijdens het opstellen van de probleemstelling kunnen onjuistheden en allerlei vaagheden ontstaan. Omdat het maken van het conceptuele simulatiemodel door een systeemanalist gedaan wordt en niet door de gebruiker zelf, kunnen er tijdens de modelvorming allerlei afbeeldfouten ontstaan. Indien het conceptuele model geïmplementeerd moet worden in een simulatietaal, dan kunnen er ook tijdens deze afbeelding fouten geïntroduceerd worden. Indien de validatie door de verkeerde mensen of met onvoldoende zorg uitgevoerd wordt, blijven de eerder geïntroduceerde fouten bestaan. Bij voorkeur dient het model ook door de gebruiker zelf gevalideerd te worden. Ook tijdens het experimenteren kunnen er fouten ontstaan doordat subruns te kort gekozen worden, er te wei-

nig subruns uitgevoerd worden of er verborgen afhankelijkheden tussen de verschillende subruns zitten. Ook kan de voorlooprun te kort gekozen zijn waardoor de meetresultaten fouten bevatten. Ook als al het voorgaande foutloos is uitgevoerd, dan kan het tijdens de interpretatie nog misgaan doordat er verkeerde conclusies getrokken worden uit de verkregen resultaten.

Typische fouten

De ervaring leert dat er een aantal typische fouten zijn, die veelvuldig voorkomen. Daarom is het zinvol deze ‘valkuilen’ eens op een rij te zetten.

Fout 1: probleemstelling wordt eenzijdig opgesteld

Een simulatiestudie gaat al fout van start als de probleemstelling eenzijdig door de gebruiker of de systeemanalist wordt opgesteld. De gebruiker weet veel van het probleemdomain, maar weinig van de eisen waaraan de probleemstelling moet voldoen. De systeemanalist weet daarentegen goed welke elementen er in de probleemstelling horen te zitten, maar weet weinig van de achtergronden van de specifieke probleemsituatie. De systeemanalist is ook goed op de hoogte van wat er wel en wat er niet mogelijk is met simulatie. De gebruiker, die over het algemeen weinig van simulatie weet, is daarentegen niet of nauwelijks op de hoogte van de (on)mogelijkheden van simulatie. Voor een succesvolle simulatiestudie is het dus van belang dat beide partijen nauw betrokken zijn bij het opstellen van de probleemstelling. De probleemstelling kunnen we in dit verband ook wel als een ‘contract’ tussen de gebruiker en de modelbouwer zien. Een *vuistregel* die we hierbij kunnen hanteren is:

vuistregel

“Begin pas met een simulatiestudie op het moment dat het voor beide partijen duidelijk is welke vragen beantwoord dienen te worden!”.

Fout 2: keuze van een verkeerd niveau van detaillering

Bij het maken van een simulatiemodel, is men gedwongen om te kiezen voor een bepaald niveau van detaillering. In het geval van een simulatiemodel voor een produktieafdeling, kunnen we kiezen om een machine te modelleren door middel van een object met als enige kenmerk een bedieningstijd of we kunnen de machine in detail modelleren waarbij we aspecten als omsteltijden, storingsen, tool-loading, onderhoudsintervallen, enzovoorts meenemen. Veel simulatiestudies worden halverwege afgebroken omdat men in het begin een verkeerd niveau van detaillering heeft gekozen. Teveel detaillering heeft tot gevolg dat het model onnodig complex wordt en dat de benodigde detail-informatie (parameters) ontbreekt. Een gebrek aan voldoende detaillering kan leiden tot een simulatiemodel waarmee de essentiële vragen in de probleemstelling niet beantwoord

Kenmerken van een goed detailleringsniveau

kunnen worden. Een goed gekozen niveau van detaillering kenmerkt zich door:

- (1) de benodigde informatie om met simulatiemodel te kunnen experimenteren is aanwezig,
- (2) de belangrijke vragen uit de probleemstelling kunnen met het model beantwoord worden en
- (3) de complexiteit van het model is voor alle partijen nog behapbaar.

Indien het niet mogelijk is een detailleringsniveau te kiezen dat voldoet aan deze voorwaarden, dan moet de probleemstelling aangepast worden.

Fout 3: verborgen aannames

Tijdens het modelleren en realiseren van een simulatiemodel, moeten er voortdurend allerlei aannames gemaakt worden. Er kunnen aannames gemaakt worden omdat de probleemstelling op bepaalde punten vaag is of omdat het simulatiemodel bewust eenvoudig gehouden moet worden. Vaak worden deze aannames niet of slecht gedocumenteerd. We spreken in dit verband ook wel over ‘verborgen aannames’. Deze verborgen aannames kunnen er in een laat stadium toe leiden dat het simulatiemodel en de bijbehorende resultaten door de gebruiker niet geaccepteerd worden. Het is daarom belangrijk om aannames goed te documenteren en deze met enige regelmaat samen met de gebruiker door te spreken. Op deze manier kunnen verrassingen achteraf voorkomen worden.

Fout 4: validatie door de verkeerde personen

Soms wordt in verband met tijdnood of een desinteresse van de gebruiker, het simulatiemodel uitsluitend gevalideerd door de maker van het model. Eventuele discrepanties tussen het model en de inzichten van de gebruiker kunnen daardoor blijven bestaan. Het is daarom aan te raden dat de gebruiker betrokken wordt bij de validatie van het simulatiemodel voordat er allerlei experimenten uitgevoerd worden.

Fout 5: kloppend maken van het model

In de validatiefase komt het nogal eens voor dat de uitkomsten van het simulatiemodel niet kloppen met historische data of met waarnemingen in de werkelijkheid. Op dat moment bestaat de neiging om het model ‘kloppend’ te maken door het veranderen van bepaalde parameterwaarden. Er wordt net zolang met de instellingen gespeeld totdat de simulatieresultaten overeenkomen met de historische data of met de waarnemingen in de werkelijkheid. Dit is echter een zeer gevaarlijke bezigheid, omdat dit niet leidt tot een simulatiemodel dat een goede afspiegeling van de werkelijkheid is. Pas als de oorzaak van een afwijking tussen het model en de

werkelijkheid boven water is, mag het model aangepast worden. Op deze manier wordt voorkomen dat fouten in het model, bewust of onbewust, verdoezeld worden.

Fout 6: gevoeligheid model onderbelicht

Vaak wordt er uitgegaan van één specifieke instelling van een bepaalde modelparameter (bijvoorbeeld de intensiteit van het aankomstproces). Ook al zijn de uitspraken voor deze instelling statistisch gezien betrouwbaar, toch kan het zijn dat kleine variaties in bijvoorbeeld het aankomstproces alle uitspraken ongeldig maken. Het is daarom van groot belang dat er tijdens de simulatiestudie ook aandacht besteed wordt aan de gevoeligheid van het model voor kleine aanpassingen van de instellingen.

Fout 7: achterwege laten van subruns

Er zijn mensen die zeggen: “Als ik maar lang genoeg simuleer, dan kloppen de gemeten resultaten wel!”. Deze mensen laten een simulatierun een hele nacht lopen en hebben daarna een blindelings vertrouwen in bijvoorbeeld de gemeten gemiddelde wachttijd. Natuurlijk is dit erg gevaarlijk omdat men dan geen inzicht krijgt in de betrouwbaarheid van dit resultaat. Andere mensen kijken naar de gemeten gemiddelde wachttijd en de gemeten gemiddelde variantie van de wachttijden en leiden daaruit een betrouwbaarheidsinterval af. Ook deze mensen gaan in de fout omdat de gemeten gemiddelde variantie van de wachttijden (in principe) niets te maken heeft met de betrouwbaarheid van de geschatte gemiddelde wachttijd. Er zijn immers afhankelijkheden tussen wachttijden van opeenvolgende klanten. De enige manier om in dit geval te komen tot onafhankelijke metingen, is het opdelen in subruns! In dit verband is het goed om nog eens te benadrukken dat simulatie alleen maar schattingen oplevert, en geen harde waarheden.

Fout 8: onzorgvuldige presentatie van de resultaten

Om de resultaten van een simulatiestudie te interpreteren, zijn vaak ingewikkelde statistische analyses nodig. Dit kan een bron van fouten zijn. Ook het vertalen van de uitkomsten van deze statistische analyses naar een voor de gebruiker begrijpelijke taal kan een bron van misverstanden zijn. In het boek “How to lie with statistics” ([4]) van Darrel Huff vinden we talloze voorbeelden van onzorgvuldige presentaties waardoor de lezer op het verkeerde been gezet wordt. Laten we als voorbeeld eens een mogelijke conclusie bekijken die in het eindrapport van een simulatiestudie kan staan. De conclusie luidt: “De wachttijden zullen met 10 procent dalen” en bevat dus veel vaagheden. Uit deze zin is immers niet duidelijk wat de betrouwbaarheid van de conclusie is. Beter is het om een betrouwbaarheidsinterval te geven. Ook suggereert de conclusie dat

voor elke klant de wachttijden met 10 procent zullen dalen. Dit hoeft echter niet het geval te zijn. Indien de gemiddelde wachttijd met 10 procent daalt, dan kan het nog zijn dat de wachttijd voor de meeste mensen stijgt en voor slechts een paar mensen (sterk) daalt.

Fout 9: verkeerd gebruik animatie

Moderne simulatiegereedschappen kenmerken zich door een mooie presentatie van de resultaten en animatie-faciliteiten. Hierdoor is het mogelijk om beter met de gebruiker te communiceren. Er schuilt echter ook een groot gevaar in animatie. Omdat de animatie alleen het tastbare deel van het simulatiemodel laat zien, bestaat het gevaar dat de gebruiker een mogelijk ongefundeerd vertrouwen in het model krijgt. Zaken als beslisregels, bedieningstijden, enzovoorts zijn immers niet of nauwelijks zichtbaar in de animatie. Ook maakt een mooie animatie een degelijke statistische analyse zeker niet overbodig.

Fout 10: simulatie als doel in plaats van middel

Simulatie is een flexibel en veelzijdig analyse-gereedschap, daarom bestaat bij sommige mensen de neiging om simulatie te pas en te onpas toe te passen. Veelal kan echter ook volstaan worden met een eenvoudig wiskundig model (bijvoorbeeld een wachtrijmodel) of een eenvoudig spreadsheet programma. In zo'n geval is simulatie een vorm van 'overkill'. Simulatie moet alleen maar ingezet worden op het moment dat de situatie dat vereist. Simulatie is een middel en geen doel op zich!

6 Slotwoord

Ondanks de vele waarschuwingen tegen het verkeerd gebruik van simulatie, is simulatie een veelzijdig analyse-gereedschap met vele prettige eigenschappen. Het is een hulpmiddel dat ingezet kan worden op het moment dat andere technieken tekort schieten.

Voor de lezer die na het lezen van dit handboek meer wil weten over simulatie, zijn er vele boeken beschikbaar. Een klassiek boek over simulatie is het boek van Naylor et al. [11]. Andere standaardwerken op het gebied van simulatie zijn: Bratley, Fox en Schrage [2], Law en Kelton [9], Ross [13] en Shannon [14]. Met name de boeken van Ross ([13]) en Bratley, Fox en Schrage ([2]) zijn aan te raden voor mensen die zich verder willen verdiepen in simulatie. Voor mensen die behoefte hebben aan een Nederlandstalig boek over simulatie verwijzen we naar Kleijnen en Groenendaal [6] en Kerbosch en Sierenberg [5].

Referenties

- [1] F. Baskett, K.M. Chandy, R.R. Muntz, and F.G. Palacios. Open, Closed and Mixed Networks of Queues with Different Classes of Customers. *Journal of the Association of Computing Machinery*, 22(2):248–260, April 1975.
- [2] P. Bratley, B.L. Fox, and L.E. Schrage. *A guide to simulation*. Springer-Verlag, Berlin, 1983.
- [3] D. Gross and C.M. Harris. *Fundamentals of queueing theory*. Wiley, London, 1985.
- [4] D. Huff. *How to lie with statistics*. Penguin Books, New York, 1954.
- [5] J.A.G.M. Kerbosch and R.W. Sierenberg. *Discrete simulatie met behulp van ALGOL, FORTRAN, PL/1*. Samsom, Alphen aan den Rijn, 1973.
- [6] J. Kleijnen and W. van Groenendaal. *Simulatie: technieken en toepassingen*. Academic Service, Schoonhoven, 1988.
- [7] J. Kleijnen and W. van Groenendaal. *Simulation: a statistical perspective*. John Wiley and Sons, New York, 1992.
- [8] L. Kleinrock. *Queueing systems, Vol. 1: Theory*. Wiley-Interscience, London, 1975.
- [9] A.M. Law and D.W. Kelton. *Simulation modeling and analysis*. McGraw-Hill, New York, 1982.
- [10] M. Ajmone Marsan, G. Balbo, and G. Conte. *Performance Models of Multiprocessor Systems*. The MIT Press, Cambridge, 1986.
- [11] T.H. Naylor, J.L. Balintfy, D.S. Burdick, and Kong Chu. *Computer simulation techniques*. Wiley, New York, 1966.
- [12] M. Pidd. *Computer modelling for discrete simulation*. John Wiley and Sons, New York, 1989.
- [13] S.M. Ross. *A course in simulation*. Macmillan, New York, 1990.
- [14] R.E. Shannon. *Systems simulation: the art and science*. Prentice-Hall, Englewood Cliffs, 1975.

A Elementaire eigenschappen

In deze appendix gaan we in op een aantal elementaire eigenschappen van stochasten. X en Y zijn stochasten met respectievelijk verwachting μ_X en variantie σ_X^2 , en verwachting μ_Y en variantie σ_Y^2 .

Ten aanzien van optelling en vermenigvuldiging van stochasten gelden de volgende universele eigenschappen:

$$\begin{aligned}\mathbb{E}[X + Y] &= \mu_X + \mu_Y \\ \mathbb{E}[aX + b] &= a\mu_X + b\end{aligned}$$

De variantie van een stochast X ($\text{Var}[X] = \sigma_X^2$) kunnen we uitdrukken in termen van de verwachting van X en X^2 .

$$\text{Var}[X] = \mathbb{E}[(X - \mu_X)^2] = \mathbb{E}[X^2] - \mu_X^2$$

Voor de variantie is de volgende eigenschap van belang:

$$\text{Var}[aX + b] = a^2\sigma_X$$

De covariantie van X en Y ($\text{Cov}[X, Y]$) noteren we ook wel als σ_{XY}^2 .

$$\text{Cov}[X, Y] = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = \mathbb{E}[XY] - \mu_X\mu_Y$$

Er is ook een relatie tussen variantie en covariantie:

$$\text{Var}[X + Y] = \sigma_X^2 + \sigma_Y^2 + 2\text{Cov}[X, Y]$$

Indien X en Y onafhankelijk:

$$\begin{aligned}\text{Cov}[X, Y] &= 0 \\ \text{Var}[X + Y] &= \sigma_X^2 + \sigma_Y^2\end{aligned}$$

A.1 Markov's ongelijkheid

Indien stochast X altijd een niet negatieve waarde oplevert, dan is voor elke $a > 0$:

$$\mathbb{P}[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$$

A.2 Chebyshev's ongelijkheid

Gegeven een stochast X met gemiddelde μ en variantie σ^2 , dan geldt voor elke $k > 0$:

$$\mathbb{P}[|X - \mu| \geq k\sigma] \leq \frac{1}{k^2}$$

A.3 Centrale limietstelling

Voor een reeks X_1, X_2, \dots, X_n van onafhankelijke gelijk verdeelde stochasten met verwachting μ en variantie σ^2 convergeert de stochast

$$\frac{\frac{1}{n}(X_1 + X_2 + \dots + X_n) - \mu}{\frac{\sigma}{\sqrt{n}}}$$

voor $n \rightarrow \infty$ naar een standaard normale verdeling.

Het gemiddelde van een groot aantal onafhankelijke stochasten is dus bij benadering normaal verdeeld. Ook de som van een groot aantal onafhankelijke stochasten is dus bij benadering normaal verdeeld. Bij het interpreteren van simulatieresultaten mogen we er dus onder bepaalde omstandigheden vanuit gaan dat de som van n onafhankelijke stochasten X_i met verwachting μ en variantie σ^2 , bij benadering normaal verdeeld is met verwachting $n\mu$ en variantie $n\sigma^2$.

A.4 Spreiding normale verdeling

Omdat de som en het gemiddelde van een groot aantal onafhankelijke stochasten bij benadering normaal verdeeld is, speelt de normale verdeling een grote rol bij het interpreteren van simulatieresultaten. Gegeven een normale verdeling met parameters μ (verwachting) en σ (standaardafwijking), is het daarom vaak interessant om te weten wat de kans is dat een bepaalde trekking tussen $\mu - \sigma$ en $\mu + \sigma$ ligt. Deze kans is ongeveer 0.683. In de onderstaande tabel is voor een aantal voor de hand liggende intervallen rond μ aangegeven wat de kans is dat een trekking uit een normale verdeling met parameters μ en σ in dit interval ligt.

interval	kans
$[\mu - \frac{\sigma}{2}, \mu + \frac{\sigma}{2}]$	0.383
$[\mu - \sigma, \mu + \sigma]$	0.683
$[\mu - 2\sigma, \mu + 2\sigma]$	0.954
$[\mu - 3\sigma, \mu + 3\sigma]$	0.997

De kans dat een bepaalde trekking tussen $\mu - 3\sigma$ en $\mu + 3\sigma$ ligt is dus ongeveer 0.997.

B Overzicht kansverdelingen

B.1 Discrete kansverdelingen

verdeling	domein	$\mathbb{P}[X = k]$	$\mathbb{E}[X]$	$\text{Var}[X]$
Bernoulli $0 \leq p \leq 1$	$k \in \{0, 1\}$	$\begin{cases} 1-p & k=0 \\ p & k=1 \end{cases}$	p	$p(1-p)$
homogeen $a < b$	$k \in \{a, \dots, b\}$	$\frac{1}{(b-a)+1}$	$\frac{a+b}{2}$	$\frac{(b-a)((b-a)+2)}{12}$
binomiaal $0 \leq p \leq 1$ $n \in \{1, 2, \dots\}$	$k \in \{0, 1, \dots, n\}$	$\binom{n}{k} p^k (1-p)^{n-k}$	np	$np(1-p)$
geometrisch $0 \leq p \leq 1$	$k \in \{1, 2, \dots\}$	$(1-p)^{k-1} p$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Poisson $\lambda > 0$	$k \in \{0, 1, \dots\}$	$\frac{\lambda^k}{k!} e^{-\lambda}$	λ	λ

B.2 Continue kansverdelingen

verdeling	domein	$f_X(x)$	$\mathbb{E}[X]$	$\text{Var}[X]$
uniform $a < b$	$a \leq x \leq b$	$\frac{1}{b-a}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
exponentieel $\lambda > 0$	$x \geq 0$	$\lambda e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
normaal $-\infty < \mu < +\infty$ $\sigma > 0$	$-\infty < x < +\infty$	$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	μ	σ^2
gamma $r, \lambda > 0$	$x > 0$	$\frac{\lambda(\lambda x)^{r-1} e^{-\lambda x}}{\Gamma(r)}$	$\frac{r}{\lambda}$	$\frac{r}{\lambda^2}$
Erlang $\lambda > 0$ $r \in \{1, 2, \dots\}$	$x > 0$	$\frac{\lambda(\lambda x)^{r-1} e^{-\lambda x}}{(r-1)!}$	$\frac{r}{\lambda}$	$\frac{r}{\lambda^2}$
χ^2 $v \in \{1, 2, \dots\}$	$x > 0$	zie gamma $r = \frac{v}{2}$ en $\lambda = \frac{1}{2}$	v	$2v$
beta $a < b$ $r, s > 0$	$a \leq x \leq b$	$\frac{1}{b-a} \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \left(\frac{x-a}{b-a}\right)^{r-1} \left(\frac{b-x}{b-a}\right)^{s-1}$	$a + (b-a) \frac{r}{r+s}$	$\frac{rs(b-a)^2}{(r+s)^2(r+s+1)}$

C Wachtrijmodellen

Wachtrijmodellen

Voor bepaalde klassen van systemen kan men allerlei aspecten zonder simulatie analyseren. We spreken in dit verband ook wel over analytische modellen, d.w.z. modellen die direct, zonder simulatie, geanalyseerd kunnen worden. Een bekende klasse van systemen waarvoor vele resultaten bekend zijn, zijn de zogenaamde *wachtrijmodellen*. In deze appendix beperken we ons tot de situatie waar er sprake is van één wachtrij. Dit neemt niet weg dat er ook voor netwerken van wachtrijen belangrijke resultaten geboekt zijn (zie bijvoorbeeld Baskett et al. [1] en Marsan et al. [10]).

A/B/c-notatie

Een wachtrij wordt vaak gekarakteriseerd door $A/B/c$ waarbij A refereert naar het aankomstproces, B naar de verdeling van de bedieningstijden en c naar het aantal parallele identieke servers. De letter M wordt gebruikt om aan te geven dat er sprake is van een negatief-exponentiële verdeling. De letter E_r wijst op een Erlang-verdeling. En G wordt gebruikt om aan te geven dat het gaat om een willekeurige verdeling.

Een van de meest eenvoudige wachtrijmodellen is de $M/M/1$ -wachtrij, d.w.z. een wachtrij met een Poisson-aankomstproces (de tussenaankomsttijden zijn negatief-exponentieel-verdeeld), negatief-exponentieel-verdeelde bedieningstijden en slechts 1 server (op elke moment wordt er ten hoogste één klant bediend).

Formule van Little

Voordat we voor een aantal eenvoudige wachtrijmodellen wat resultaten presenteren, behandelen we de *formule van Little*. De formule van Little is een formule die geldig is voor elke wachtrij waar sprake is van een stabiele situatie zonder afhankelijkheden tussen het aankomstproces en het bedieningsproces. De formule luidt:

$$L = \lambda S$$

waarbij L het gemiddeld aantal klanten in het systeem is, λ het gemiddeld aantal klanten dat per tijdseenheid binnenstroomt en S de gemiddelde systeemtijd (d.w.z. de tijd die de klant gemiddeld in het systeem verblijft, dus de wachttijd en de bedieningstijd samen).

M/M/1-wachtrij

De $M/M/1$ -wachtrij wordt gespecificeerd door middel van twee parameters λ en μ . Voor het aankomstproces wordt gebruik gemaakt van een negatief-exponentiële verdeling met parameter λ . De gemiddelde tussenaankomsttijd is dus $\frac{1}{\lambda}$. Voor het bedieningsproces wordt gebruik gemaakt van een negatief-exponentiële verdeling met parameter μ . De gemiddelde bedieningstijd is dus $\frac{1}{\mu}$. De bezettingsgraad ρ is:

$$\rho = \frac{\lambda}{\mu}$$

De toestand van de wachtrij kunnen we noteren door middel van een natuurlijk getal k dat het totale aantal klanten in het systeem voorstelt. De kans dat de wachtrij op een bepaald moment in toestand k is, noteren we als p_k :

$$p_k = (1 - \rho)\rho^k$$

Deze kansen worden ook wel de *evenwichtskansen* genoemd. Het gemiddeld aantal klanten in het systeem is L :

$$L = \frac{\rho}{1 - \rho}$$

De gemiddelde systeemtijd is S :

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

De gemiddelde wachttijd W is het verschil van de systeemtijd en de bedieningstijd:

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

M/E_r/1-wachtrij

De $M/E_r/1$ -wachtrij wordt gespecificeerd door middel van drie parameters λ , r en μ . De bedieningstijd is nu Erlang-verdeeld met parameters λ en r . Voor de bezettingsgraad ρ , het gemiddeld aantal klanten L , de gemiddelde systeemtijd S en de gemiddelde wachttijd W vinden we de volgende waarden:

$$\begin{aligned} \rho &= \frac{r\lambda}{\mu} \\ L &= \frac{\lambda\rho(r+1)}{2\mu(1-\rho)} + \rho \\ S &= \frac{\rho(r+1)}{2\mu(1-\rho)} + \frac{r}{\mu} \\ W &= \frac{\rho(r+1)}{2\mu(1-\rho)} + \frac{r-1}{\mu} \end{aligned}$$

M/G/1-wachtrij

Over de $M/G/1$ -wachtrij is duidelijk minder bekend. Het aankomstproces wordt weer gespecificeerd door de parameter λ . De gemiddelde bedieningstijd is μ en de variantie van de bedieningstijd is σ^2 . De variatiecoëfficiënt van de bedieningstijd C definiëren we als volgt: $C = \frac{\sigma}{\mu}$.

Voor de bezettingsgraad ρ , het gemiddeld aantal klanten L , de gemiddelde systeemtijd S en de gemiddelde wachttijd W vinden we de volgende waarden:

$$\begin{aligned}\rho &= \frac{\lambda}{\mu} \\ L &= \rho + \frac{\rho^2}{2(1-\rho)}(1+C^2) \\ S &= \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)}(1+C^2) \\ W &= \frac{\rho}{2\mu(1-\rho)}(1+C^2)\end{aligned}$$

De formule $L = \rho + \frac{\rho^2}{2(1-\rho)}(1+C^2)$ staat ook wel bekend als de formule van Pollaczek-Khinchin.

M/G/∞-wachtrij

De *M/G/∞*-wachtrij wordt gespecificeerd door middel van twee parameters λ en μ . De variantie van de verdeling G is blijkbaar niet relevant. Omdat er in de *M/G/∞*-wachtrij altijd voldoende vrije servers zijn is de bezettingsgraad niet te definiëren. Feitelijk is de bezettingsgraad gelijk aan 0. We stellen ρ nu gelijk aan de hoeveelheid werk die per tijdseenheid het systeem binnenkomt: $\rho = \frac{\lambda}{\mu}$. De evenwichtskansen p_k zijn nu:

$$p_k = \frac{\rho^k}{k!} e^{-\rho}$$

Voor het gemiddeld aantal klanten L , de gemiddelde systeemtijd S en de gemiddelde wachttijd W vinden we de volgende waarden:

$$\begin{aligned}L &= \rho \\ S &= \frac{1}{\mu} \\ W &= 0\end{aligned}$$

Voor meer informatie over dit onderwerp verwijzen we naar Kleinrock [8] en Gross en Harris [3].