

Stopproblemen en Speltheorie

De wiskunde
achter de optimale
inbraak

T.P.S. Heijnders



Stopproblemen en Speltheorie

De wiskunde achter de optimale inbraak

door

T.P.S. Heijnders

ter verkrijging van de graad van Bachelor of Science
aan de Technische Universiteit Delft,
online te verdedigen op 21 augustus.

Datum: 3 juli 2020
Studentnummer: 4362659
Beoordelingscommissie: Dr. R.J. Fokink, TU Delft, begeleider
Drs. E. M. van Elderen, TU Delft

Voorwoord

Voor u ligt het bacheloreindproject dat ten behoeve van het behalen van het bachelordiploma Technische Wiskunde te TU Delft is geschreven.

Dit project is uitgevoerd onder de afdeling toegepaste kansrekening van het Delft institute of Applied Mathematics onder leiding van R.J. Fokkink.

Het was erg interessant om wiskundig de denkwijze achter inbraken te benaderen. Het project werd relevanter dan ooit toen er bij mij thuis werd ingebroken. In dit geval was de buit van de dief mijn laptop met hierop de Matlab code voor de numerieke resultaten. Deze moest ik destijds dan ook opnieuw schrijven.

Ik wil graag mijn begeleider R.J. Fokkink bedanken voor al zijn begeleiding, hulp en geduld. Daarnaast wil ik ook E.M. van Elderen bedanken voor het plaatsnemen in de beoordelingscommissie.

*T.P.S. Heijnders
Delft, juli 2020*

Abstract

In dit verslag wordt het gedrag van een dief die huizen besteelt wiskundig bekeken. Het dievenspel is de aanleiding van dit verslag. In dit spel bepaalt een dief zijn optimale strategie voor het bestellen van één huis uit verschillende huizen met verschillende waarden. Daarnaast is er een agent die op zoek is naar zijn optimale strategie om de huizen te bewaken.

Dit verslag duikt in de speltheorie om het dievenspel te ontleden. Het dievenspel wordt opgelost en vervolgens uitgebreid naar het geval dat de dief meerdere huizen kan bestellen en er meerdere agenten zijn. Er worden twee gevallen onderscheiden, waar huizen dezelfde waarde hebben en waar huizen verschillende waarde hebben.

Voor de situatie met huizen van dezelfde waarde wordt er gebruik gemaakt van de theorie over stopproblemen om een optimale strategie voor de dief te bepalen. Eerder wordt de theorie van stopproblemen uitgelicht. Hier worden interessante voorbeelden en optimale stopregels besproken.

Voor de situatie met huizen van verschillende waarde zal de strategie van de dief en de agent numeriek worden geoptimaliseerd. Er zal blijken dat het voordeliger is voor de dief om meer huizen van lagere waarde te bestellen en voor de agenten om meer huizen van hogere waarden te bewaken.

Inhoudsopgave

1	Stoppen op het laatste succes	1
1.1	Inleiding	1
1.2	Stopproblemen met onafhankelijke Bernoulli variabelen	2
1.2.1	Het voorspellen van het laatste succes bij het gooien van een dobbelsteen	2
1.2.2	Het secretaresseprobleem	4
1.3	Optimale stopregels	7
1.3.1	Sum-the-odds.	7
1.3.2	Voorbeelden Sum-the-odds	9
1.3.3	One-stage look-ahead	10
1.3.4	One-stage look-ahead identiek aan Sum-the-odds	11
1.4	Stopprobleem met afhankelijke Bernoulli variabelen	13
2	Speltheorie	15
2.1	Inleiding	15
2.2	Nulsomspel	15
2.3	Introductie dievenspel	17
2.4	Indifference	17
2.5	Oplossing dievenspel.	19
2.5.1	Algemene oplossing Dievenspel.	20
3	Het dievenspel als stopprobleem	23
3.1	Huizen met dezelfde waarde.	23
3.2	Huizen met verschillende waardes	25
3.2.1	Greedy stopregel	25
3.2.2	Numerieke resultaten.	27
3.3	Optimale stopregel dief.	28
3.3.1	Numerieke resultaten.	30
3.4	Conclusie en discussie	32
A	Matlab code	33
A.1	Functies Greedy stopregel	33
A.2	Code figuur 3.1	34
A.3	Code figuur 3.2	34
A.4	Code figuur 3.3 en 3.4	36
A.5	Code figuur 3.5 en 3.6	38
	Bibliografie	41

Stoppen op het laatste succes

1.1. Inleiding

In mijn studententijd heb ik wel eens een casino avond georganiseerd met een commissie. Tijdens deze avond konden veel spellen gespeeld worden zoals pokeren, blackjack en roulette. Bij roulette valt een balletje op een willekeurige cijfer van 0 tot en met 36 die ook de kleur zwart of rood hebben. De 0 heeft als enige de kleur groen. Vervolgens kan je geld inzetten op bepaalde cijfers of kleuren. Wanneer het balletje hier op belandt krijg je de inzet een aantal keer terug.

Op een gegeven moment waagde ik me aan de roulette en hier was ik niet weg te slaan. Ook al was de inleg van de avond maar 2 euro, je kon daarmee wel fiches inzetten. Dit kon dus ook bij de roulette, waar je op allerlei manieren je fiches in kon zetten en terugverdienen. Die avond vroeg ik mij af: wanneer is nu het beste moment om te stoppen? Ik bleef fiches inzetten. Soms won ik wat en soms verloor ik ook wat, maar wanneer moet ik dan stoppen?

Dit is een voorbeeld van een **stopprobleem**. Bij een stopprobleem moet een bepaalde actie worden ondernomen op een bepaalde tijd zodat een beloning zo hoog mogelijk of het verlies zou laag mogelijk is. In het geval van roulette moet je dus stoppen met spelen wanneer de winst zo hoog mogelijk is. Zo zijn er nog veel meer voorbeelden in de praktijk. Zoals bij het handelen op de aandelenbeurs, wanneer moet een aandeel verkocht worden zodat de winst het hoogst is? Of een drugsmokkelaar moet bedenken wanneer hij drugs mee smokkelt en wanneer niet, wanneer moet hij stoppen? In dit hoofdstuk zullen we verschillende stopproblemen bekijken en verschillende methodes om deze op te lossen.

Verschillende stopproblemen hebben verschillende eigenschappen. In het geval van de roulette kan de speler zo lang doorgaan als hij wil, wanneer er altijd een dealer is. In dat geval is het probleem *oneindig*. Wanneer de dealer die avond naar huis moet heeft de speler maar een beperkte tijd om te spelen en is het probleem *eindig*. Elke keer dat de roulette wordt gedraaid is de kans hetzelfde dat het balletje op een bepaald getal komt. De kansen zijn dus *onafhankelijk*. Wanneer de dealer de speler probeert tegen te werken en de kansen op winst verkleint als het goed gaat, dan is het probleem *afhankelijk*. In het geval van de roulette kan de speler stoppen na elke beurt of hij nu verlies of winst heeft gemaakt. We zullen verder vooral stopproblemen tegenkomen waar er alleen gestopt kan worden bij winst.

1.2. Stopproblemen met onafhankelijke Bernoulli variabelen

Bij een stopprobleem observeert een speler een rij random variabelen $X_1, X_2, \dots$ en krijgt hij bij bepaalde momenten de keuze om te stoppen of door te gaan. Als de speler stopt krijgt hij een beloning die afhankelijk is van de hele rij variabelen.

Voor elk stopprobleem zoeken we naar een **stopregel**. Dit is een regel om te beslissen om te stoppen of door te gaan met het observeren van een rij random variabelen van een stopprobleem op basis van de huidige en vorige observaties.

Bij een stopprobleem met beloning B zoeken we naar een **optimale** stopregel. Een stopregel is optimaal wanneer de verwachting van B zo hoog mogelijk is.

We zullen nu twee stopproblemen bekijken. Deze zullen te beschrijven zijn met onafhankelijke Bernoulli variabelen $X_1, X_2, \dots, X_n$. Een Bernoulli variabele X_i met $1 \leq i \leq n$ kan gelijk zijn 0 of aan 1. We noemen $X_i = 1$ een **succes**.

1.2.1. Het voorspellen van het laatste succes bij het gooien van een dobbelsteen

We spelen een spel door het gooien van een dobbelsteen. We gooien 100 keer met een eerlijke dobbelsteen en het doel is om te stoppen op de worp waar voor het laatst een 6 wordt gegooid. We mogen alleen stoppen als er een 6 wordt gegooid en winnen het spel als we stoppen bij een 6 en dit ook de laatste is. Als we niet stoppen of stoppen bij een 6 die niet de laatste is, verliezen we het spel. Bij winst krijgen we een beloning van 1 en bij verlies 0. Bij dit stopprobleem bekijken we dus een eindige rij onafhankelijke variabelen met als beloning 1 of 0.

We bekijken een rij onafhankelijke Bernoulli variabelen $X_1, \dots, X_{100}$. Deze variabelen zijn dan representatief voor de worpen van de dobbelsteen. Er geldt $X_k = 1$ als er een 6 wordt gegooid voor $1 \leq k \leq 100$ en dit noemen we een succes. Er geldt dan $\mathbb{P}(X_k = 1) = \frac{1}{6}$. Nu willen we stoppen bij de hoogste k waarvoor geldt $X_k = 1$. Hiervoor zoeken we de optimale stopregel. Het is logisch om een stopregel te kiezen van de vorm: *stop bij de eerste 6 die wordt gegooid na k worpen*.

Stel we nemen de stopregel: *stop bij de eerste 6 die gegooid wordt*.

Deze stopregel geeft ons de winst als deze 6 de enige 6 is die gegooid wordt. Deze kans is gelijk aan $\frac{1}{6} * \frac{5}{6}^{99} * \binom{100}{1} \approx 0,00000024$. Deze stopregel lijkt dus niet optimaal.

Stel we nemen de stopregel: *stop bij de eerste 6 die wordt gegooid na de eerste 50 worpen*.

De kans dat dit de laatste keer is dat er 6 wordt gegooid is dan gelijk aan $\frac{1}{6} * \frac{5}{6}^{49} * \binom{50}{1} \approx 0,001$. Deze stopregel is al beter dan de vorige, maar is waarschijnlijk nog steeds niet optimaal.

Nu stoppen we bij de eerste 6 die gegooid wordt na k worpen en bepalen de optimale waarde van k . Noem K_k het aantal keren dat er een 6 wordt gegooid na k worpen. Dan geldt dat K_k binomiaal verdeeld is met $K_k \sim B(100 - k, \frac{1}{6})$. De kans dat er nog maar één 6 wordt gegooid na k worpen is gelijk aan:

$$\mathbb{P}(K_k = 1) = \frac{1}{6} * \frac{5}{6}^{100-k-1} * \binom{100-k}{1}. \quad (1.1)$$

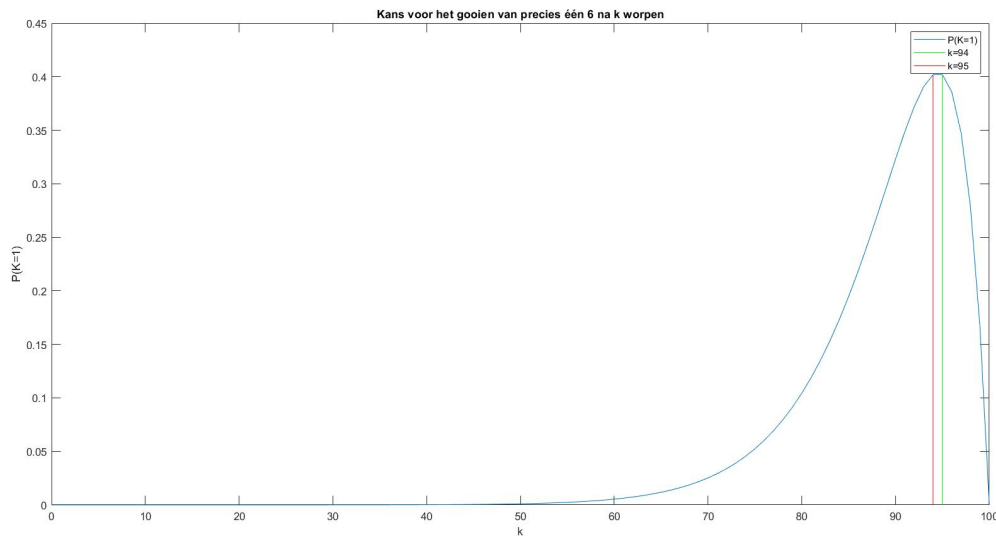
Nu willen we de k vinden waarvoor de kans om precies één 6 te gooien na worp $100 - k$ het hoogst is. We bekijken de afgeleide van $\mathbb{P}(K_k = 1)$ naar k en stellen deze gelijk aan 0 om het maximum te bepalen. Neem $k \in \mathbb{R}$ zodat:

$$\mathbb{P}(K_k = 1) = \frac{1}{6} * \frac{5^{99}}{6} * \frac{6^k}{5} * (100 - k) = 0 \quad (1.2)$$

$$\frac{d\mathbb{P}(K_k = 1)}{dk} = 0 \Leftrightarrow \frac{1}{6} * \frac{5^{99}}{6} * \frac{6^k}{5} * \ln\left(\frac{6}{5}\right)(100 - k) - \frac{1}{6} * \frac{5^{99}}{6} * \frac{6^k}{5} = 0 \quad (1.3)$$

$$\Leftrightarrow \ln\left(\frac{6}{5}\right)(100 - k) = 1 \Leftrightarrow k = 100 - \frac{1}{\ln\left(\frac{6}{5}\right)} \approx 94,52. \quad (1.4)$$

De afgeleide van $\mathbb{P}(K_k = 1)$ is gelijk aan 0 wanneer $k = 94,52$. In onderstaande grafiek zien we inderdaad dat dit een maximum is. De vraag is daarnaast of we moeten stoppen na worp 94 of worp 95. Ook dit zien we in onderstaande grafiek.



Figuur 1.1: We zien $\mathbb{P}(K_k = 1)$ voor $1 \leq k \leq 100$. $k = 94$ en $k = 95$ zijn met verticale lijnen afgebeeld.

Hieruit blijkt dat de kans $\mathbb{P}(K_k = 1)$ gelijk is aan 0,4019 voor $k = 94$ en $k = 95$. Dit betekent dat de optimale stopregel in dit geval niet uniek is. Er zijn twee stopregels die een even grote kans geven voor het stoppen bij het laatste succes, namelijk:

- Stop bij de eerste 6 die wordt gegooit na 94 worpen.
- Stop bij de eerste 6 die wordt gegooit na 95 worpen.

1.2.2. Het secretaresseprobleem

Stel je zoekt een partner voor het leven en je kan kiezen uit een n aantal kandidaten. Hoe zorg je dat je met de grootste kans de beste kandidaat kiest? Deze vraag gaan we beantwoorden aan de hand van het secretaresseprobleem, een bekend probleem uit de speltheorie.

Het probleem luidt als volgt. Een werkgever wil een nieuwe secretaresse aannemen en hij krijgt n kandidaten in willekeurige volgorde over de vloer.¹ Één van deze kandidaten is de beste kandidaat voor de baan en verder kunnen alle kandidaten gerangschikt worden op basis van hun geschiktheid. Na het gesprek moet de werknemer direct bepalen of hij de kandidaat aanneemt of niet. De keuze kan niet worden teruggedraaid en de rest van de kandidaten zullen niet meer in aanmerking komen. Het probleem luidt: *Hoe vindt de werkgever de beste kandidaat als secretaresse?*

Dit probleem kan beschouwd worden als een stopprobleem en we zoeken naar de optimale stopregel voor dit probleem. Het is logisch om te zien dat we niet meteen de eerste kandidaat moeten kiezen omdat we nog geen vergelijkingsmateriaal hebben. We moeten dus eerst een aantal kandidaten afwijzen om vergelijkingsmateriaal te hebben.

We nemen de volgende stopregel: *Wijs eerst $n/2$ kandidaten af en kies dan de eerste kandidaat uit de tweede helft die beter is dan iedereen van de eerste helft.*

Deze stopregel geeft de beste kandidaat wanneer de een-na-beste beste kandidaat bij de eerste helft kandidaten zit, en de beste bij de tweede helft. De kans dat de een-na-beste beste kandidaat bij de eerste helft kandidaten zit is ongeveer $1/2$. De kans dat de beste kandidaat bij de tweede helft kandidaten zit is ook ongeveer $1/2$. Deze kansen zijn ongeveer $1/2$ omdat deze twee kansen afhankelijk van elkaar zijn en omdat n oneven kan zijn. De kans dat met deze stopregel de beste kandidaat wordt gevonden is dan ongeveer $1/4$. Met deze redelijk simpele stopregel kunnen we dus al de beste kandidaat kiezen met kans ongeveer $1/4$. We zullen nu een stopregel bekijken die een nog beter resultaat geeft.

Noem n het aantal kandidaten en random variabelen $X_1, X_2, \dots, X_n$ representatief voor de kandidaten. Neem $X_k = 1$ voor $1 \leq k \leq n$ als deze kandidaat de beste is tot dan, dus wanneer $X_k > X_j$ voor alle $j < k$. Dit noemen we een succes en bij geen succes geldt $X_k = 0$. Er geldt dan dat X_k Bernoulli verdeeld is met $\mathbb{P}(X_k = 1) = \frac{1}{k}$. De rij variabelen is dan onafhankelijk en n is eindig.

We bekijken de stopregel: *wijs de eerste r kandidaten af en stop op het eerst volgende succes.*

We bepalen de r zodat deze stopregel optimaal is. Noem S_r de kandidaat die gekozen wordt gebruikmakend van bovenstaande stopregel en B de beste kandidaat. We willen de kans $\mathbb{P}(S_r = B)$ maximaliseren voor r . Dit is de kans dat deze stopregel ons de beste kandidaat als secretaresse geeft. We bekijken eerst de kans dat kandidaat i met $0 \leq i \leq n$ de beste is en ook gekozen wordt. Deze kans is gelijk aan:

$$\mathbb{P}(S_r = i \cap B = i) = \mathbb{P}(S_r = i | B = i) \mathbb{P}(B = i) = \mathbb{P}(S_r = i | B = i) \frac{1}{n}. \quad (1.5)$$

We zijn nu op zoek naar $\mathbb{P}(S_r = i | B = i)$. Merk op dat we onderscheid kunnen maken tussen de gevallen dat kandidaat i bij de eerste r kandidaten zit of bij de laatste $r + 1$ kandidaten zit.

Als $i \leq r$, gegeven dat kandidaat i de beste is, wordt deze kandidaat volgens onze stopregel nooit gekozen en er geldt $\mathbb{P}(S_r = i | B = i) = 0$.

Voor $i \geq r + 1$, gegeven dat kandidaat i de beste is, wordt kandidaat i alleen gekozen als de beste kandidaat van de eerste $i - 1$ kandidaten bij de eerste r kandidaten zit. Er geldt dan $\mathbb{P}(S_r = i | B = i) = \frac{r}{i-1}$.

¹In het klassieke geval wordt er gesproken over een mannelijke werkgever en een secretaresse. Natuurlijk is het niet meer van deze tijd om dit aan te nemen.

Merk nu op dat $\mathbb{P}(S_r = B)$ gelijk is aan de som van $\mathbb{P}(S_r = i | B = i)$ over alle $1 \leq i \leq n$. We zien:

$$\mathbb{P}(S_r = B) = \sum_{i=1}^n \mathbb{P}(S_r = i | B = i) \frac{1}{n} \quad (1.6)$$

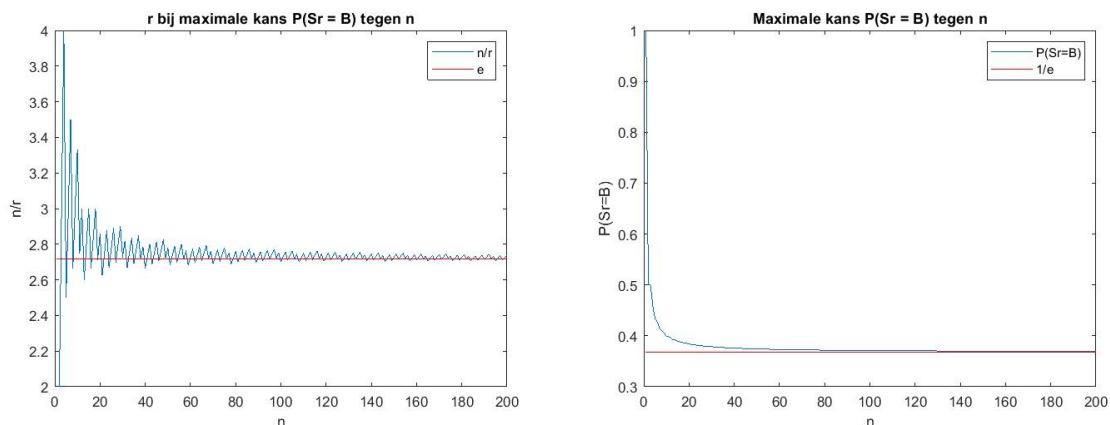
$$= \left(\sum_{i=1}^r 0 + \sum_{i=r+1}^n \frac{1}{n} \frac{r}{i-1} \right) \quad (1.7)$$

$$= \frac{r}{n} \sum_{i=r+1}^n \frac{1}{i-1}. \quad (1.8)$$

Als $r = 0$ wijzen we geen kandidaten af, de kans dat we dan de beste kiezen is $1/n$. Vergelijking (1.8) geldt dan ook alleen voor $r > 0$. Daarnaast herkennen we de harmonische reeks in (1.8). We weten dat $H_n = \sum_{i=1}^n \frac{1}{i}$ en we zien:

$$\mathbb{P}(S_r = B) = \frac{r}{n} (H_{n-1} - H_{r-1}). \quad (1.9)$$

Gebruik makend van Matlab kunnen we nu voor alle n kijken voor welke r de kans $\mathbb{P}(S_r = B)$ maximaal is. Het resultaat is te zien in onderstaande grafieken.



Figuur 1.2: Voor n met $1 \leq n \leq 200$ zien we r waarvoor $\mathbb{P}(S_r = B)$ maximaal is. Het resultaat is weergegeven als n/r . Daarnaast zien we de lijn e . Het lijkt erop dat n/r naar e convergeert. Daarnaast zien we de bijbehorende maximale kans $\mathbb{P}(S_r = B)$ voor $1 \leq n \leq 200$. We zien ook de lijn $1/e$, het lijkt er op dat $\mathbb{P}(S_r = B)$ naar $1/e$ convergeert.

Hieruit blijkt dat n/r lijkt te convergeren naar e als $n \rightarrow \infty$. De bijbehorende kans $\mathbb{P}(S_r = B)$ lijkt daarbij te convergeren naar $1/e$ wanneer $n \rightarrow \infty$.

Stelling 1.2.1. $\mathbb{P}(S_r = B)$ is maximaal voor $r = \frac{n}{e}$ als $n \rightarrow \infty$. $\mathbb{P}(S_r = B)$ is dan gelijk aan $\frac{1}{e}$.¹

Bewijs. We kunnen de som van $\mathbb{P}(S_r = B)$ benaderen met een integraal. Merk eerst op dat:

$$\mathbb{P}(S_r = B) = \frac{r}{n} (H_{n-1} - H_{r-1}) = \frac{r}{n} \left(H_n - \frac{1}{n} - H_{r-1} \right). \quad (1.10)$$

¹het bewijs is gebaseerd op het bewijs dat te lezen is op Wikipedia, zie <https://nl.wikipedia.org/wiki/Secretaressesprobleem>

Uit $n \rightarrow \infty$ volgt dat $-\frac{1}{n} \rightarrow 0$. We weten van harmonische reeks H_n dat deze te benaderen is met integraal $\int_1^n (1/t) dt$. Dus $\mathbb{P}(S_r = B)$ kan worden benaderd met de volgende integraal:

$$\mathbb{P}(S_r = B) = \frac{r}{n} \int_r^n \frac{1}{t} dt = \frac{r}{n} (\ln(n) - \ln(r)) = -\frac{r}{n} (\ln(r) - \ln(n)) = -\frac{r}{n} \ln\left(\frac{r}{n}\right). \quad (1.11)$$

We vinden het maximum van $\mathbb{P}(S_r = B)$ door de afgeleide voor r gelijk te stellen aan 0. We krijgen:

$$\frac{d\mathbb{P}(S_r = B)}{dr} = 0 \Leftrightarrow -\frac{1}{n} \ln\left(\frac{r}{n}\right) - \frac{r}{n} \frac{1}{r} = 0 \Leftrightarrow -\frac{1}{n} \ln\left(\frac{r}{n}\right) = \frac{1}{n} \Leftrightarrow \ln\left(\frac{r}{n}\right) = -1 \Leftrightarrow \frac{r}{n} = e \quad (1.12)$$

Dus $\mathbb{P}(S_r = B)$ is maximaal voor $r = \frac{n}{e}$ als $n \rightarrow \infty$. Wanneer we $r = \frac{n}{e}$ substitueren in (1.11) zien we dat $\mathbb{P}(S_r = B) = \frac{1}{e}$. $\square$

Met onze gekozen stopregel zien we dus dat de we met kans $\frac{1}{e} \approx 0,37$ de beste kandidaat kunnen kiezen. Waarom deze stopregel optimaal is zullen we in de volgende paragraaf laten zien.

1.3. Optimale stopregels

We hebben nu enkele stopproblemen gezien en daarvoor een stopregel bepaald. Dit is niet altijd even makkelijk. Daarom bekijken we algemene stopregels die in bepaalde gevallen altijd optimaal zullen zijn. We zullen net als in de voorbeelden uit de vorige paragraaf vooral gebruik maken van Bernoulli variabelen $X_1, X_2, \dots, X_n$ en voor alle $1 \leq k \leq n$ noemen we $X_k = 1$ een succes. Het doel is om te stoppen op het laatste succes.

1.3.1. Sum-the-odds

Een van deze optimale stopregels is **Sum-the-odds**, geïntroduceerd door Bruss [1].

Voor gebeurtenissen beschreven met onafhankelijke Bernoulli variabelen $X_1, X_2, \dots, X_n$ kunnen we met de Sum-the-odds stopregel een $1 \leq s \leq n$ vinden zodat de speler moet stoppen bij i waarvoor geldt $X_i = 1$ en $i \geq s$.

Definitie 1.3.1. *Neem $X_1, X_2, \dots, X_n$ een rij Bernoulli variabelen. Noem $p_k = \mathbb{P}(X_k = 1)$ en $q_k = p_k - 1$. We definiëren de **odds** van X_k als $r_k = \frac{p_k}{q_k}$ voor $1 \leq k \leq n$.*

Met gebrek aan een goede vertaling voor "odds", zullen we dit woord blijven gebruiken.

De Sum-the-odds stopregel zoekt naar moment s vanaf waar het stoppen op het eerst volgende succes het laatste succes is. Neem r_i de odds van moment i , volgens de Sum-the-odds stopregel is s de maximale s waarvoor geldt: $r_n + r_{n-1} + r_{n-2} + \dots + r_s \geq 1$. Als deze s niet bestaat geldt $s = 1$. Bruss definieert de stopregel als volgt:

Stelling 1.3.1 (Sum-the-odds door Bruss). *Voor een eindige rij onafhankelijke Bernoulli variabelen $X_1, X_2, \dots, X_n$ met r_k de odds van X_k voor alle $1 \leq k \leq n$, bestaat er een optimale stopregel voor het stoppen op het laatste succes. Namelijk door te stoppen op de eerste $k \geq s$ zodat $X_k = 1$ en:*

$$s = \sup \left\{ 1, \sup \left\{ 1 \leq k \leq n : \sum_{j=k}^n r_j \geq 1 \right\} \right\}. \quad (1.13)$$

De stelling is als eerst bewezen door Bruss. We zullen gedurende dit hoofdstuk een generalisatie van de Sum-the-odds stopregel gebruiken. Deze is gegeven door Ferguson [3].

Definitie 1.3.2 (Sum-the-odds door Ferguson). *Voor een eindige rij onafhankelijke Bernoulli variabelen $X_1, X_2, \dots, X_n$ met r_k de odds van X_k voor $1 \leq k \leq n$, stop bij k waarvoor geldt:*

$$k = \min \left\{ k \geq 1 : X_k = 1 \text{ en } \sum_{i=k+1}^n r_i \leq 1 \right\}. \quad (1.14)$$

Wanneer de som gelijk is aan 1 voor een bepaalde k zegt de stopregel dat het niet uit maakt of we stoppen of doorgaan op dit punt.

Daarnaast kan n ook oneindige waarde aannemen. Als $n \rightarrow \infty$ kan het voorkomen dat er geen laatste succes voorkomt. In dit geval zou de stopregel nooit stoppen en zou deze niet meer optimaal zijn. Hier-voor gebruikt Ferguson de Borel-Cantelli lemma.

Stelling 1.3.2 (Borel-Cantelli). *Voor een rij gebeurtenissen $A_1, A_2, \dots \in \mathcal{F}$ met kansruimte $(\Omega, \mathcal{F}, \mathbb{P})$ geldt: wanneer $\sum_{k=1}^{\infty} \mathbb{P}(A_k) < \infty$ dan $\mathbb{P}(\limsup_{k \rightarrow \infty} A_k) = 0$.*

Met $\limsup_{k \rightarrow \infty} A_k = \bigcap_{n=1}^{\infty} \bigcup_{k=n}^{\infty} A_k$. Dit betekent dat $s \in \limsup_{k \rightarrow \infty} A_k$ dan en slechts dan wanneer $s \in A_k$ voor oneindig veel $k \in \mathbb{N}$.

In woorden zegt Borel-Cantelli wanneer de som van de kansen van $A_1, A_2 \dots$ eindig is, is de kans dat deze gebeurtenissen oneindig vaak voorkomen gelijk aan 0.

Bewijs. Merk op dat $\mathbb{P}$ een maat is. Nu kunnen we onze kennis van maten uit Reële Analyse [2] gebruiken om deze stelling te bewijzen.

Neem $E_n = \bigcup_{k=n}^{\infty} A_k$ dan geldt $E_1 \supseteq E_2 \supseteq E_3 \dots$ dus E_n is een dalende rij met $\lim_{n \rightarrow \infty} E_n = \bigcap_{n=1}^{\infty} E_n$. Daarnaast geldt door de σ -subadditiviteit van $\mathbb{P}$ dat:

$$\mathbb{P}(E_1) = \mathbb{P}\left(\bigcup_{k=1}^{\infty} A_k\right) \leq \sum_{k=1}^{\infty} \mathbb{P}(A_k) < \infty.$$

Door een stelling uit Reële Analyse over dalende rijen weten we omdat $\lim_{n \rightarrow \infty} E_n = \bigcap_{n=1}^{\infty} E_n$ en $\mathbb{P}(E_1) < \infty$ dat er geldt:

$$\mathbb{P}\left(\bigcap_{n=1}^{\infty} E_n\right) = \mathbb{P}\left(\lim_{n \rightarrow \infty} E_n\right) = \lim_{n \rightarrow \infty} \mathbb{P}(E_n) = \lim_{n \rightarrow \infty} \mathbb{P}\left(\bigcup_{k=n}^{\infty} A_k\right) \leq \lim_{n \rightarrow \infty} \sum_{k=n}^{\infty} \mathbb{P}(A_k) = 0.$$

Hieruit volgt dat $\mathbb{P}\left(\bigcap_{n=1}^{\infty} E_n\right) = \mathbb{P}\left(\bigcap_{n=1}^{\infty} \bigcup_{k=n}^{\infty} A_k\right) = 0$ □

Voor de rij Bernoulli variabelen $X_1, X_2, \dots, X_n$ kunnen we de gebeurtenissen onderscheiden dat $X_i = 1$ een succes is voor $1 \leq i \leq n$. De kans dat deze gebeurtenissen oneindig vaak voorkomen is gelijk aan 0 wanneer $\sum_{i=1}^{\infty} p_i < \infty$. Als dit geldt kan n oneindige waarde aannemen.

We zullen in de volgende paragraaf bewijzen dat de Sum-the-odds stopregel optimaal is, gebruikmakende van een andere optimale stopregel.

1.3.2. Voorbeelden Sum-the-odds

De voorbeelden die eerder behandeld zijn kunnen we nu opnieuw oplossen met de Sum-the-odds stop-regel (STO). We zullen zien dat dit een stuk makkelijker gaat.

- Bij het voorbeeld van de dobbelsteen met onafhankelijke Bernoulli variabelen $X_1, X_2, \dots, X_n$ geldt $p_k = \mathbb{P}(X_k = 1) = \frac{1}{6}$ en de odds zijn gelijk aan $\frac{1}{5}$. De STO zegt dat we moeten stoppen voor de minimale k zodat $X_k = 1$ en

$$\sum_{i=k+1}^{100} \frac{1}{5} \leq 1.$$

Wanneer $k = 95$ is de som gelijk aan 1. Dit betekent dat de kansen om te stoppen of doorgaan gelijk zijn en we kunnen kiezen voor $k \geq 95$ en $k \geq 96$. Ook hieruit blijkt dus dat er twee optimale stopregels zijn, namelijk: *stop bij het eerste succes als $k \geq 95$* en *stop bij het eerste succes als $k \geq 96$* . Dit geeft dezelfde resultaten die we eerder zagen: stop na worp 94 en stop na worp 95.

- Voor het secretaresseprobleem kunnen we het bekende resultaat opnieuw aantonen met de STO. Het probleem vraagt ons om de beste kandidaat te vinden en dus niet per se het laatste succes. Zoals eerder genoemd kunnen we een succes in dit geval beschouwen wanneer de beste kandidaat tot zover gevonden wordt. Dan is het laatste succes gelijk aan de beste kandidaat, vervolgens kunnen we de STO gebruiken. We bekijken de gebeurtenissen dat n kandidaten op sollicitatiegesprek komen. Dit kunnen we beschrijven met een rij onafhankelijke Bernoulli variabelen $X_1, X_2, \dots, X_n$ met $X_i = 1$ voor $1 \leq i \leq n$ als kandidaat i de beste kandidaat is tot zover. Dan geldt $p_i = \frac{1}{i}$ en de odds zijn gelijk aan $r_i = \frac{1}{i-1}$.

Volgens de STO moeten we stoppen bij de k waarvoor geldt:

$$\min \left\{ k \geq 1 : X_k = 1 \text{ en } \sum_{i=k+1}^n \frac{1}{i-1} \leq 1 \right\}. \quad (1.15)$$

We zoeken nu eigenlijk naar r zodat $\sum_{i=r+1}^n \frac{1}{i-1} \leq 1$. Dan geldt dat we moeten stoppen voor de eerste $k \geq r$ waarvoor geldt $X_k = 1$. Merk op dat:

$$\sum_{i=r+1}^n \frac{1}{i-1} \leq 1 \Leftrightarrow \frac{r}{n} \sum_{i=r+1}^n \frac{1}{i-1} \leq \frac{r}{n}. \quad (1.16)$$

Vanwege (1.8) is (1.16) equivalent aan $\mathbb{P}(S_r = B) \leq \frac{r}{n}$. Voor $n \rightarrow \infty$ geldt dat we $\mathbb{P}(S_r = B)$ kunnen benaderen met (1.11) en er volgt:

$$\mathbb{P}(S_r = B) \leq \frac{r}{n} \Leftrightarrow -\frac{r}{n} \ln\left(\frac{r}{n}\right) \leq \frac{r}{n} \Leftrightarrow \quad (1.17)$$

$$\ln\left(\frac{r}{n}\right) \geq -1 \Leftrightarrow \frac{r}{n} \geq \frac{1}{e} \Leftrightarrow r \geq \frac{n}{e}. \quad (1.18)$$

De STO geeft nu dat we moeten stoppen bij k waarvoor geldt $k \geq r \geq n/e$ en $X_k = 1$. Met in dit geval een succes de beste kandidaat tot zover. Zo hebben we dus het klassieke geval bewezen met de STO.

1.3.3. One-stage look-ahead

Naast de Sum-the-odds stopregel bekijken we een andere stopregel voor een stopprobleem waar we moeten stoppen bij het laatste succes. De stopregel heet de One-stage look-ahead (1-sla) en is geïntroduceerd door Ferguson [3]. Deze stopregel is nog algemener en is goed te gebruiken voor afhankelijke random variabelen. De enige restrictie is nu dat het stoppen op geen succes verboden is. Dit was tot nu toe nooit een logische optie dus eigenlijk is het geen beperking.

We bekijken weer een rij Bernoulli variabelen $X_1, X_2, \dots$ waar de rij nu ook afhankelijk en oneindig kan zijn. Omdat stoppen op een niet succes verboden is definieert Ferguson een **stage**. Dit is korter en makkelijker te gebruiken, maar omdat de notatie verwarrend kan zijn definiëren we eerst I .

Definitie 1.3.3. *Neem Bernoulli variabelen $X_1, X_2, \dots$. We noemen I de **indicator verzameling van alle successen** met $I = \{i \in \mathbb{N} : X_i = 1\}$.*

Definitie 1.3.4. *Neem de Bernoulli random variabelen $X_1, X_2, \dots$. Neem I de indicator verzameling van alle successen en $i \in I$. Een **stage** is een verzameling Z_i van Bernoulli variabelen na een succes $i - 1$ en tot en met succes i . Als er minder dan i successen zijn, dan zeggen we $Z_i = 0$.*

Dit betekent zodra er geen successen in een verzameling Z_i zitten, we zeggen dat $Z_i = 0$. Nu kunnen we dus stoppen op een stage en is het stoppen op geen succes niet meer mogelijk.

We nemen aan dat het aantal successen eindig is en dus geldt voor de rij $Z_1, Z_2, \dots, Z_n$ met $n \in I$ dat deze altijd eindigt met $Z_n = 0$. Wanneer we te maken hebben met een oneindige rij Bernoulli variabelen is de rij van stages eindig omdat het aantal successen eindig is.

Laat $Z_1, Z_2, \dots, Z_n$ een rij van stages met $n \in I$. Nu willen we één stage van te voren voorspellen wanneer $Z_n = 0$. Net als voorheen winnen we 1 als we correct voorspellen en 0 als we incorrect voorspellen. Verder maakt Ferguson de beloning voor niet stoppen anders. Als de rij 0 bereikt voor onze voorspelling, is het laatste succes al geweest en kunnen we niet meer stoppen. In dit geval winnen we ω . Als omega groter is dan 1 is het altijd voordeliger om niet te stoppen, dus er geldt $\omega < 1$.

Stel we willen stoppen in stage n . Dan kan het zijn dat $Z_n = 0$ en er dus geen successen meer zijn, in dit geval krijgen we ω . Als er geldt $Z_n \neq 0$ willen we dat $Z_{n+1} = 0$ in dat geval geldt dat het laatste succes in Z_n zit. De beloning is dan gelijk aan 1.

Neem $I(A)$ de indicatorfunctie van het event A en $\mathcal{G}_n = \sigma(Z_1, Z_2, \dots, Z_n)$ de sigma-algebra gegenereerd door de Bernoulli variabelen in de stages tot en met succes n . Dit zijn alle mogelijke gebeurtenissen die we al gezien kunnen hebben. Het stoppen in stage n geeft ons de volgende functie voor de beloning Y_n :

$$Y_n = \omega I(Z_n = 0) + I(Z_n \neq 0) \mathbb{P}(Z_{n+1} = 0 | \mathcal{G}_n), \quad (1.19)$$

$$Y_\infty = \omega. \quad (1.20)$$

Om een stage van te voren te kunnen voorspellen, kijken we naar de voorwaardelijke verwachting van Y_{n+1} . Hiervoor geldt:

$$\mathbb{E}(Y_{n+1} | \mathcal{G}_n) = \omega \mathbb{P}(Z_{n+1} = 0 | \mathcal{G}_n) + \mathbb{P}(Z_{n+1} \neq 0, Z_{n+2} = 0 | \mathcal{G}_n). \quad (1.21)$$

De 1-sla zegt dat we moeten stoppen wanneer de beloning in stage n groter is dan de verwachte beloning in stage $n + 1$. Dus we stoppen wanneer $Y_n \geq \mathbb{E}(Y_{n+1} | \mathcal{G}_n)$. Als $Z_n = 0$ zijn we te laat gestopt en is de beloning ω . Als we kijken naar $Z_n \neq 0$ krijgen we de volgende ongelijkheid:

$$\mathbb{P}(Z_{n+1} = 0 | \mathcal{G}_n) \geq \omega \mathbb{P}(Z_{n+1} = 0 | \mathcal{G}_n) + \mathbb{P}(Z_{n+1} \neq 0, Z_{n+2} = 0 | \mathcal{G}_n) \quad \Leftrightarrow \quad (1.22)$$

$$(1 - \omega) \mathbb{P}(Z_{n+1} = 0 | \mathcal{G}_n) \geq \mathbb{P}(Z_{n+1} \neq 0, Z_{n+2} = 0 | \mathcal{G}_n) \quad \Leftrightarrow \quad (1.23)$$

$$\frac{\mathbb{P}(Z_{n+1} \neq 0, Z_{n+2} = 0 | \mathcal{G}_n)}{\mathbb{P}(Z_{n+1} = 0 | \mathcal{G}_n)} \leq 1 - \omega. \quad (1.24)$$

Hieruit volgt de definitie van de one-stage look-ahead stopregel.

Definitie 1.3.5. De one-stage look-ahead stopregel zegt dat we moeten stoppen in stage Z_n waarvoor geldt:

$$N_1 = \min \left\{ n : Z_n = 0 \text{ of } (Z_n \neq 0 \text{ en } \frac{\mathbb{P}(Z_{n+1} \neq 0, Z_{n+2} = 0 | \mathcal{G}_n)}{\mathbb{P}(Z_{n+1} = 0 | \mathcal{G}_n)} \leq 1 - \omega) \right\}. \quad (1.25)$$

We zien dus dat de 1-sla de kans op geen successen meer, vergelijkt met de kans op nog maar één succes. Ferguson toont aan dat de stopregel optimaal is wanneer vergelijking (1.24) niet stijgt. Dit klinkt logisch. Wanneer de regel zegt te stoppen, is de vergelijking kleiner dan $1 - \omega$. Als de vergelijking later weer groter is kan deze waarde weer boven $1 - \omega$ komen.

De 1-sla is dus een optimale stopregel. Hiermee kunnen we aantonen dat ook de Sum-the-odds stopregel optimaal is.

1.3.4. One-stage look-ahead identiek aan Sum-the-odds

Omdat de 1-sla optimaal is, kunnen we hiermee aantonen dat de STO optimaal is. Sterker nog, we zullen bewijzen dat voor onafhankelijke Bernoulli variabelen de 1-sla en STO identiek zijn.

Met een voorbeeld kunnen we al zien dat de 1-sla en STO dezelfde resultaten geven.

Stel we gooien 100 keer met een oneerlijke munt met $X_1, \dots, X_{100}$ Bernoulli variabelen zodat $X_k = 1$ als worp k kop geeft en $p_k = \mathbb{P}(X_k = 1) = \frac{1}{25}$. De worp kop wordt als een succes gezien. De kans dat worp k munt geeft is dan q_k . De odds r_j van gooi j zijn dan gelijk aan $r_j = (1/25)/(24/25) = 1/24$ en volgens de STO moeten we stoppen bij X_k waarvoor geldt $X_k = 1$ en $\sum_{i=k+1}^{100} 1/24 \leq 1$. We moeten dus stoppen wanneer $k \geq 76$ en $X_k = 1$.

Met de 1-sla stopregel vergelijken we de kans op een succes nu en een succes later. Neem een succes op moment k . De kans dat dit het laatste succes is $(24/25)^{100-k}$ en de kans op nog precies één succes is $(1/25)(24/25)^{100-k-1}(100-k)$.

Volgens de 1-sla moeten we stoppen bij $X_K = 1$ met k waarvoor geldt

$$\frac{\frac{1}{25} \frac{24}{25}^{100-k-1} \binom{100-k}{1}}{\frac{24}{25}^{100-k}} \leq 1 \quad (1.26)$$

$$\Leftrightarrow \frac{1}{24} (100-k) \leq 1. \quad (1.27)$$

Dit geeft dus dezelfde stopregel als de STO met $k \geq 76$.

Stelling 1.3.3. Voor een rij onafhankelijke Bernoulli variabelen $X_1, X_2, \dots, X_n$ geldt: de 1-sla is identiek aan de STO.

Bewijs. We definiëren de 1-sla als in (1.25) en we bekijken stage Z_i met $i \in I$ en $Z_i \neq 0$. We bekijken eerst het geval dat $Z_{i+1} \neq 0$. We nemen $Z_{i+1} = \{X_{k+1}, X_{k+2}, \dots, X_m\}$ en $Z_{i+2} = \{X_{m+1}, X_{m+2}, \dots, X_n\}$ met $k+1 \leq m \leq n$ willekeurig. We nemen vervolgens voor alle $1 \leq j \leq n$ aan dat $\mathbb{P}(X_j = 1) = p_j$ en $q_j = 1 - p_j$. Er geldt:

$$\begin{aligned} \mathbb{P}(Z_{i+1} \neq 0, Z_{i+2} = 0 | \mathcal{G}_i) &= \sum_{m=k+1}^n \mathbb{P}(X_{k+1} = \dots = X_{m-1} = 0, X_m = 1, X_{m+1} = X_{m+2} = \dots = X_n = 0 | \mathcal{G}_i) \\ &= \sum_{m=k+1}^n \prod_{j=k+1}^{m-1} q_j p_m \prod_{j=m+1}^n q_j. \end{aligned}$$

Vervolgens bekijken we het geval wanneer $Z_{i+1} = 0$ en we nemen $Z_{i+1} = \{X_{k+1}, X_{k+2}, \dots, X_n\}$. Er volgt:

$$\mathbb{P}(Z_{i+1} = 0 | \mathcal{G}_i) = \mathbb{P}(X_{k+1} = X_{k+2} = \dots = X_n = 0 | \mathcal{G}_i) = \prod_{j=k+1}^n q_j.$$

Volgens de 1-sla (1.25) bekijken we:

$$\begin{aligned} \frac{\mathbb{P}(Z_{i+1} \neq 0, Z_{i+2} = 0 | \mathcal{G}_i)}{\mathbb{P}(Z_{i+1} = 0 | \mathcal{G}_i)} &= \frac{\sum_{m=k+1}^n \prod_{i=k+1}^{m-1} q_i p_m \prod_{i=m+1}^n q_i}{\prod_{i=k+1}^n q_i} \\ &= \sum_{m=k+1}^n \left(\frac{\prod_{i=k+1}^{m-1} q_i p_m \prod_{i=m+1}^n q_i}{\prod_{i=k+1}^n q_i} \right) = \sum_{m=k+1}^n \left(\frac{\prod_{i=k+1}^{m-1} q_i p_m \prod_{i=m+1}^n q_i}{\prod_{i=k+1}^{m-1} q_i q_m \prod_{i=m+1}^n q_i} \right) \\ &= \sum_{m=k+1}^n \frac{p_m}{q_m} \end{aligned}$$

Neem $r_m = \frac{p_m}{q_m}$. Er volgt uit de 1-sla dat we moeten stoppen als $X_k = 1$ en $\sum_{m=k+1}^n r_m \leq 1$ wanneer $\omega = 0$. Dit geeft ons de STO. $\square$

1.4. Stopprobleem met afhankelijke Bernoulli variabelen

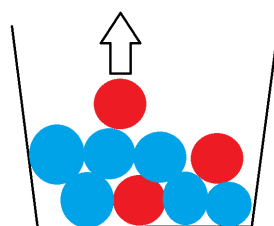
Eerder bekeken we alleen nog spellen met onafhankelijke Bernoulli variabelen. Nu bekijken we een spel waar de Bernoulli variabelen afhankelijk zijn.

We bekijken een spel met een vaas met blauwe en rode knikkers zoals in Ferguson [3] beschreven staat. De speler trekt n knikkers en hij wil stoppen bij de laatste rode knikker. Neem $X_1, X_2, \dots, X_n$ de Bernoulli variabelen representatief voor het trekken van knikkers. Het trekken van een rode knikker kunnen we dan zien als een succes en we zeggen $X_i = 1$ als trekking i een rode knikker geeft. Noem a het aantal rode knikkers, b het aantal blauwe knikkers en n het aantal knikkers dat wordt getrokken. Er geldt dan $a + b \geq n$. Succesvol stoppen bij de laatste rode knikker geeft beloning 1. Verkeerd en niet stoppen geeft beloning 0.

Dit probleem kunnen we niet oplossen met de STO. We kunnen namelijk geen $p_k = \mathbb{P}(X_k = 1)$ voor trekking k vinden. Deze kans is namelijk afhankelijk van hoeveel blauwe en rode knikkers er al zijn getrokken en dit weten we niet. We zullen de 1-sla stopregel gebruiken.

Noem a_k en b_k het aantal rode en blauwe knikkers respectievelijk in de vaas na k keer trekken. Merk op dat er moet gelden $b_k \geq n - k$. Wanneer er minder blauwe knikkers zijn dan het aantal overige trekkingen stop je natuurlijk niet omdat er dan sowieso nog een rode knikker wordt getrokken.

We bekijken eerst een voorbeeld voor we het algemene geval bekijken. Stel we hebben een vaas met $a = 3$, $b = 6$ en $n = 5$. We willen stoppen bij de laatste rode knikker die wordt getrokken. Bij $k = 1$, trekken we een rode knikker. Nu moeten we beslissen, is dit het laatste succes of niet. Hiervoor bekijken we de kansen met de hypergeometrische verdeling:



Figuur 1.3: De vaas met rode en blauwe knikkers

We bekijken eerst de kans dat de volgende $n - k = 4$ knikkers die we trekken blauw zijn. Merk op $a_k = 2$ en $b_k = 6$. Er volgt door de hypergeometrische verdeling.

$$p_b = \frac{\binom{6}{4}}{\binom{8}{4}} \approx 0,21$$

en

$$p_{a1} = \frac{\binom{2}{1} \binom{6}{3}}{\binom{8}{4}} \approx 0,58.$$

Met p_b de kans dat er alleen nog maar blauwe knikkers worden getrokken en p_{a1} de kans dat er nog één rode knikker wordt getrokken en verder alleen blauwe knikkers. We zien $p_b > p_{a1}$ en volgens de 1-sla moeten we nog niet stoppen.

We gaan verder met het spel en bij de volgende twee trekkingen trekken we een blauwe knikker. Nu bij de vierde trekking, trekken we weer een rode knikker. De kans dat we bij de laatste trekking een blauwe knikker trekken is nu $4/5$ en een rode $1/5$. Omdat $p_b < p_{a1}$ moeten we volgens de 1-sla stoppen.

We bekijken nu het algemene geval en definiëren:

$$V_k = \mathbb{P}(\text{De volgende trekkingen zijn blauw} \mid a_k, b_k) \quad (1.28)$$

$$= \frac{\binom{b_k}{n-k}}{\binom{a_k+b_k}{n-k}} \quad (1.29)$$

en

$$W_k = \mathbb{P}(\text{De volgende trekkingen zijn blauw en 1 rood} \mid a_k, b_k) \quad (1.30)$$

$$= \frac{\binom{b_k}{n-k-1} \binom{a_k}{1}}{\binom{a_k+b_k}{n-k}}. \quad (1.31)$$

Voor de 1-sla bekijken we het quotiënt:

$$\frac{W_k}{V_k} = \frac{a_k(n-k)}{b_k - (n-k-1)}. \quad (1.32)$$

De 1-sla voor het stopprobleem met de vaas wordt dan:

$$N_1 = \min\{k \leq n : X_k = 1 \text{ en } (a_k + 1)(n-k) \leq b_k + 1\}. \quad (1.33)$$

Stelling 1.4.1. *De 1-sla voor het stopprobleem met de vaas is een optimale stopregel.*

Bewijs. Om dit te bewijzen moeten we aantonen dat (1.32) niet stijgt voor k . In het geval dat er een blauwe knikker wordt getrokken geldt:

$$\frac{W_{k+1}}{V_{k+1}} = \frac{a_k(n-k-1)}{b_k - 1 - (n-k-2)} = \frac{a_k(n-k-1)}{b_k - (n-k-1)} \leq \frac{a_k(n-k)}{b_k - (n-k-1)} = \frac{W_k}{V_k}.$$

In het geval dat er een rode knikker wordt getrokken geldt:

$$\frac{W_{k+1}}{V_{k+1}} = \frac{(a_k - 1)(n-k-1)}{b_k - (n-k-2)} \leq \frac{a_k(n-k)}{b_k - (n-k-1)} = \frac{W_k}{V_k}.$$

We zien dus dat (1.32) niet stijgt voor k dus de 1-sla is optimaal. □

2

Speltheorie

2.1. Inleiding

Speltheorie is een relatief onbekend tak binnen de wiskunde waar onder andere beslissingstheorie en optimaliseren veel wordt gebruikt. Bij speltheorie wordt het spelen van spellen op een wiskundige manier geanalyseerd. Voor deze spellen kunnen vervolgens wiskundige modellen gemaakt worden.

Bekijk bijvoorbeeld het spel boter-kaas-en-eieren. In dit bekende spel kan speler 1 kruisjes plaatsen in 9 vakjes en speler 2 kan vervolgens rondjes plaatsen. De speler die als eerst drie van zijn eigen vorm op een rij heeft wint. Speler 1 kan kiezen op welke van de lege vakjes hij een kruisje zet, dit zijn keuzes die bepaald worden door een **strategie** van speler 1. Voor zijn eerste zet zijn er negen mogelijkheden, vervolgens heeft speler 2 acht mogelijkheden afhankelijk van wat speler 1 gedaan heeft. Vervolgens kan speler 1 weer zeven posities kiezen voor zijn kruisje, afhankelijk van wat hij de eerste beurt gedaan heeft en wat speler twee gedaan heeft. Voor de eerste twee zetten kan speler 1 dus al kiezen uit $9 \cdot 8 \cdot 7$ mogelijke strategieën. Uiteindelijk kunnen beide speler 1 en 2 kiezen uit $9!$ strategieën. Dit is extreem veel en deze zouden we nooit allemaal overzichtelijk kunnen uitschrijven.

De $9!$ strategieën die speler 1 en 2 kunnen doen zijn niet allemaal logisch. Bij sommige strategieën wordt de drie op een rij al eerder bereikt. Ook zou je het speelveld kunnen draaien en zullen bepaalde strategieën hetzelfde zijn door de symmetrie van het speelveld. In speltheorie gaat men er dan ook vanuit dat de spelers alleen logische en rationele strategieën kiezen. In werkelijkheid ligt het aantal strategieën dus iets lager dan $9!$ maar zijn er nog steeds heel veel!

Beide spelers zullen op zoek gaan naar hun **optimale strategie**, onafhankelijk van wat de andere speler doet. Een strategie is optimaal wanneer de verwachte beloning zo hoog mogelijk is. De verwachte beloning van de spelers noemen we de **waarde van het spel**. In dit hoofdstuk gebruiken we stellingen uit de speltheorie om een optimale strategie voor een dief te vinden in het dievenspel.

2.2. Nulsomspel

De persoon die het meest wordt geassocieerd bij de grondlegging van de speltheorie is John Neumann. Het was Neumann die begon met een twee personen **nulsomspel**. Deze spellen worden gespeeld door twee spelers waar de ene speler wint wat de andere speler verliest. In het geval van boter-kaas-en-eieren kunnen we zeggen dat de speler die wint 1 krijgt en de speler die verliest -1 krijgt. Als het gelijkspel is, krijgen beide spelers 0. Boter-kaas-en-eieren is dan ook een nulsomspel. We bekijken de definitie van een nulsomspel en stellingen die hier bij komen kijken. Voor achtergrondinformatie en precieze definities verwijzen we naar Ferguson's notes voor Game Theory [4].

Definitie 2.2.1. De strategische vorm van een twee personen nulsomspel is gegeven door ruimte (X, Y, A) , waar

1. X is een niet lege verzameling met strategieën van speler 1
2. Y is een niet-lege verzameling met strategieën van speler 2
3. A is de pay-off, een reële functie gedefinieerd op $X \times Y$ (dus $A(x, y)$ is een reël getal voor elke $x \in X$ en $y \in Y$)

Een twee personen nulsomspel heet eindig als beide strategieverzamelingen X en Y eindig zijn.

De definitie betekent het volgende: Wanneer een spel wordt gespeeld kiest speler 1 een strategie $x \in X$ en speler 2 een strategie $y \in Y$. Speler 1 wint vervolgens $A(x, y)$ en speler 2 verliest $A(x, y)$. We kiezen speler 1 en speler 2 vast gedurende dit hoofdstuk zodat speler 1 altijd $A(x, y)$ wint en speler 2 altijd $A(x, y)$ verliest. Wanneer $A(x, y)$ negatief is verliest speler 1 en wint speler 2. Daarnaast kan $A(x, y)$ ook worden beschreven als een matrix.

Definitie 2.2.2. Een eindig nulsomspel in strategische vorm wordt beschreven door een matrix. Als $X = \{x_1, \dots, x_m\}$ en $Y = \{y_1, \dots, y_n\}$ dan is de **spel matrix**:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \text{ met } a_{ij} = A(x_i, y_j).$$

Speler 1 kiest dus een rij uit de matrix en speler 2 kiest een kolom. De pay-off wordt gegeven door de bijbehorende waarde in de matrix.

Een speler kan ook met een bepaalde kansverdeling tussen strategieën kiezen, dit wordt een **gemengde strategie** genoemd.

Definitie 2.2.3 (Gemengde strategie). Een gemengde strategie is een kansverdeling $p = (p_1, \dots, p_m)$ over de strategie verzameling van een speler.

Als speler 1 de gemengde strategie $p = (p_1, \dots, p_m)$ gebruikt dan is zijn gemiddelde winst gelijk aan $\sum_{i=1}^m p_i a_{ij}$, gegeven dat speler 2 kiest voor strategie y_j . Hetzelfde geldt voor speler 2 als deze kiest voor $q = (q_1, \dots, q_n)$. Het gemiddelde verlies van speler 2 is dan gelijk aan $\sum_{j=1}^n q_j a_{ij}$ als speler 1 kiest voor strategie x_i .

Belangrijk om op te merken dat is beide spelers hun strategie kiezen wetende wat de andere speler voor strategie kiest.

Stelling 2.2.1 (De Minimax Stelling). Voor alle eindige zero-sum games is er

1. een getal V , dit is de waarde van het spel.
2. een (gemengde) strategie voor speler 1 zodat zijn gemiddelde winst minstens V is onafhankelijk van wat speler 2 doet.
3. een (gemengde) strategie voor speler 2 zodat zijn gemiddelde verlies niet meer is dan V onafhankelijk van wat speler 1 doet.

Met deze theorie kunnen we nu het dievenspel bekijken.

2.3. Introductie dievenspel

Een dief heeft de keuze om 5 huizen te beroven: huis A, B, C, D en E met een respectievelijke buit van 1000, 500, 400, 300 en 50. De dief kiest één huis uit om te bestelen en houdt deze buit. Echter is er ook een politie agent in het spel die één huis bewaakt. Als de dief het huis met de agent kiest is hij de buit kwijt. De functie A wordt gepresenteerd door een matrix.

Voor de dief is de verzameling van strategieën, de huizen waar hij kan inbreken. Er geldt $X = \{A, B, C, D, E\}$. Voor de politieagent is de verzameling van de strategieën gelijk aan de huizen die hij kan bewaken, $Y = \{A, B, C, D, E\}$. De functie $A(x_i, y_i)$ is gelijk aan de winst van de dief en het verlies van de politie agent wanneer de dief kiest voor huis x_i en de politieagent voor huis y_i . De spel matrix is nu als volgt.

	A	B	C	D	E
A	0	1000	1000	1000	1000
B	500	0	500	500	500
C	400	400	0	400	400
D	300	300	300	0	300
E	50	50	50	50	0

(2.1)

De dief kiest dus een rij uit en de politieagent kiest een kolom. Kiest de dief bijvoorbeeld voor rij D, dan is zijn buit in de meeste gevallen gelijk aan 300. Behalve natuurlijk wanneer de politie agent ook voor huis D kiest. Het is niet moeilijk om te zien dat de optimale oplossing voor de dief een gemengde strategie zal zijn. Als de dief enkel kiest voor één huis, zal de strategie van de agent zijn om bij dit huis te gaan staan. We gaan dus op zoek naar de gemengde strategie voor beide spelers.

Grote spelmatrices kunnen soms worden gereduceerd naar kleinere matrices. Dit kan door rijen en kolommen te verwijderen die irrelevant zijn voor de speler om ze te gebruiken.

Definitie 2.3.1. Rij i van een matrix $A = (a_{ij})$ **domineert** rij k als $a_{ij} \geq a_{kj}$ voor alle j . Rij i **domineert strikt** rij k als $a_{ij} < a_{kj}$.

Vergelijkbaar, kolom j van A **domineert (domineert strikt)** kolom k als $a_{ij} \leq a_{ik}$ (resp. $a_{ij} < a_{ik}$) voor alle i .

Wanneer een rij een andere rij domineert kan de gedomineerde rij worden verwijderd uit de matrix omdat speler 1 een strategie kan kiezen die beter of even goed is. Hetzelfde geldt voor de kolommen, wanneer een kolom een andere kolom domineert zal speler 2 nooit meer voor de gedomineerde kolom kiezen. De waarde van het spel verandert niet als deze gedomineerde rijen of kolommen worden verwijderd. Er kan daarin tegen wel een optimale strategie bestaan die gebruik maakt van de verwijderde rij of kolom. In dit geval zal er nog wel een andere optimale strategie zijn die even goed is. Bij een verwijderde rij die strikt gedomineerd wordt, verandert de verzameling van optimale strategieën niet.

Daarnaast kan een rij (kolom resp.) ook worden verwijderd als deze wordt gedomineerd door een kans verdeling van andere rijen (kolommen resp.). Als voor een $0 < p < 1$ geldt dat $pa_{ij} + (1-p)a_{kj} \geq a_{hj}$ voor alle j dan is rij h gedomineerd door de gemengde strategie die gebruik maakt van kansen p en $1-p$ met rij i en k . De speler kan minstens even goed deze gemengde strategie gebruiken in plaats van rij k en daarom kan deze rij worden verwijderd. Hetzelfde geldt voor de kolommen.

2.4. Indifference

Als speler 1 de gemengde strategie $\mathbf{p} = (p_1, \dots, p_m)$ gebruikt en speler 2 kiest voor kolom j dan is de gemiddelde beloning van speler 1 gelijk aan $\sum_{i=1}^m p_i a_{ij}$. De waarde van het spel is V en wanneer deze strategie optimaal is geldt door de Minimax stelling dat de waarde $\sum_{i=1}^m p_i a_{ij}$ minstens V is onafhankelijk van wat j is. Er geldt dus $\mathbf{p} = (p_1, \dots, p_m)$ is optimaal dan en slechts dan:

$$\sum_{i=1}^m p_i a_{ij} \geq V \text{ voor alle } j = 1, \dots, n. \quad (2.2)$$

Vergelijkbaar geldt strategie $q = (q_1, \dots, q_n)$ is optimaal voor speler 2 dan en slechts dan:

$$\sum_{j=1}^n a_{ij} q_j \leq V \text{ voor alle } i = 1, \dots, m. \quad (2.3)$$

Stelling 2.4.1. Wanneer beide spelers hun optimale strategie gebruiken geldt dat $V = \sum_i \sum_j p_i a_{ij} q_j$.

Bewijs. We gebruiken het feit dat $\sum_{j=1}^n q_j = \sum_{i=1}^m p_i = 1$. Er geldt dan:

$$V = \sum_{j=1}^n V q_j \leq \sum_{j=1}^n q_j \left(\sum_{i=1}^m p_i a_{ij} \right) = \sum_{i=1}^m \sum_{j=1}^n p_i a_{ij} q_j = \sum_{i=1}^m p_i \left(\sum_{j=1}^n a_{ij} q_j \right) \leq \sum_{i=1}^m p_i V = V. \quad (2.4)$$

Waaruit volgt dat $V \leq \sum_i \sum_j p_i a_{ij} q_j \leq V$. Dus $V = \sum_i \sum_j p_i a_{ij} q_j$. $\square$

Gebruikmakend van bovenstaande stelling vinden we de stelling voor indifference

Stelling 2.4.2 (Indifference). Neem een spel met $m \times n$ matrix A en waarde V . Laat $p = (p_1, \dots, p_m)$ een optimale strategie voor speler 1 en $q = (q_1, \dots, q_n)$ een optimale strategie voor speler 2. Dan geldt:

$$\sum_{i=1}^m p_i a_{ij} = V \text{ voor alle } j \text{ waarvoor geldt } q_j > 0 \quad (2.5)$$

en

$$\sum_{j=1}^n a_{ij} q_j = V \text{ voor alle } i \text{ waarvoor geldt } p_i < 0. \quad (2.6)$$

We zeggen dus dat de strategie van speler 1 "indifference" is met de keuze van speler 2. Dit betekent dat het niet uitmaakt welke strategie speler 2 kiest. De strategie van speler 1 zal altijd hetzelfde resultaat geven. Vanwege een gebrek aan een goede vertaling zullen we het Engelse woord indifference gebruiken.

Bewijs. Neem een willekeurige k waarvoor geldt $q_k \geq 0$. Neem nu aan dat $\sum_{i=1}^m p_i a_{ik} \neq V$, dan leiden we een tegenstelling af.

Er geldt door (2.2) dat $\sum_{i=1}^m p_i a_{ik} > V$ en uit (2.4) volgt:

$$V = \sum_{j=1}^n q_j \left(\sum_{i=1}^m p_i a_{ij} \right) > \sum_{j=1}^n q_j V = V.$$

Bij term k volgt er $V > V$ dus we hebben een tegenspraak en (2.5) geldt.

Vergelijkbaar nemen we een k zodat $p_k \geq 0$. Neem nu aan dat $\sum_{j=1}^n a_{kj} q_j \neq V$, dan leiden we een tegenspraak af. Uit 2.3 volgt dat $\sum_{j=1}^n a_{kj} q_j < V$ en uit 2.4 volgt dat:

$$V = \sum_{i=1}^m p_i \left(\sum_{j=1}^n a_{ij} q_j \right) < \sum_{i=1}^m p_i V = V.$$

Bij term k volgt er $V < V$ dus we hebben een tegenspraak en (2.6) geldt. $\square$

2.5. Oplossing dievenspel

We beginnen met het reduceren van spel matrix (2.1). Neem bijvoorbeeld $\frac{1}{3}a_{1j} + \frac{2}{3}a_{2j}$. Dit resulteert in een rij waar de minimale waarde gelijk is aan 333,33. Deze rij is dus groter dan rij a_{4j} en rij a_{5j} . Deze rijen kunnen dus verwijderd worden. Vervolgens hebben de kolommen a_{i4} en a_{i5} geen element 0 meer en er geldt dat $a_{i4}, a_{i5} \geq a_{ij}$ voor alle j . Dus ook deze twee kolommen kunnen worden verwijderd.

Dit betekent dat de dief met bovenstaande gemengde strategie altijd een beter resultaat heeft dan het bestellen van huis D of E. Vervolgens zal de agent niet huis D en E bewaken omdat de dief hier nooit zal inbreken.

De gereduceerde matrix ziet er als volgt uit:

	A	B	C
A	0	1000	1000
B	500	0	500
C	400	400	0

(2.7)

Met behulp van stelling 2.4.2 kunnen we de optimale strategie van de dief vinden. Neem strategie van de dief $\mathbf{p} = (p_1, p_2, p_3)$ met p_1 de kans dat de dief huis A besteeft, p_2 de kans dat de dief huis B besteeft en p_3 de kans dat de dief huis C besteeft. Neem V de waarde van het spel, er volgt:

$$0p_1 + 500p_2 + 400p_3 = V \quad (2.8)$$

$$1000p_1 + 0p_2 + 400p_3 = V \quad (2.9)$$

$$1000p_1 + 500p_2 + 0p_3 = V \quad (2.10)$$

$$p_1 + p_2 + p_3 = 1. \quad (2.11)$$

Hieruit volgt dat:

$$V = \frac{4000}{11} \text{ en } \mathbf{p} = \left(\frac{2}{11}, \frac{4}{11}, \frac{5}{11}\right). \quad (2.12)$$

Hetzelfde kunnen we doen voor de agent. Neem $\mathbf{q} = (q_1, q_2, q_3)$ met q_1 de kans dat de agent huis A bewaakt, q_2 de kans dat de agent huis B bewaakt en q_3 de kans dat de agent huis C bewaakt. Dan volgt door stelling (2.4.2) dat:

$$0q_1 + 1000q_2 + 1000q_3 = V \quad (2.13)$$

$$500q_1 + 0q_2 + 500q_3 = V \quad (2.14)$$

$$400q_1 + 400q_2 + 0q_3 = V \quad (2.15)$$

$$q_1 + q_2 + q_3 = 1. \quad (2.16)$$

Hieruit volgt dat:

$$V = \frac{4000}{11} \text{ en } \mathbf{q} = \left(\frac{7}{11}, \frac{3}{11}, \frac{1}{11}\right). \quad (2.17)$$

Dit betekent dat de dief met kans $2/11$ kiest voor huis A, met kans $4/11$ kiest voor huis B en met kans $5/11$ voor huis C. We zien dus dat de dief met de meeste kans kiest voor huizen waar minder te halen valt.

De agent kiest voor kans $7/11$ voor huis A, met kans $3/11$ voor huis B en met kans $1/11$ voor huis C. De agent bewaakt dus met de meeste kans de huizen waar het meest te halen valt.

Vervolgens geldt voor de waarde van het spel dat $V = \frac{4000}{11}$. Dit betekent dat de dief gemiddeld 364 verdient en de agent dit bedrag verliest.

2.5.1. Algemene oplossing Dievenspel

Neem huizen $1, 2, \dots, n$ met waarde respectievelijk $u_1, u_2, \dots, u_n$, zodat $u_1 \geq u_2 \geq u_3 \geq \dots \geq u_n$. De dief wil een huis beroven en de agent bewaakt één huis. De spelmatrix ziet er als volgt uit:

$$\begin{pmatrix} 0 & u_1 & u_1 & \dots & u_1 \\ u_2 & 0 & u_2 & \dots & u_2 \\ u_3 & u_3 & 0 & \dots & u_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_n & u_n & u_n & \dots & 0 \end{pmatrix}. \quad (2.18)$$

Vervolgens nemen we $k \leq n$, de huizen waar het meest te halen valt. Net als in het dievenspel zal de dief alleen bij deze k huizen inbreken en de agent zal alleen deze bewaken.

Neem V_k de waarde van het spel bij het kiezen uit alleen deze k huizen. De agent heeft strategie $q = \{q_1, q_2, \dots, q_k, 0, \dots, 0\}$. Door indifference geldt dat:

$$\sum_{j=1}^k q_j u_{ij} = V_k \text{ voor alle } i \leq k.$$

Voor deze som geldt $u_{ij} = u_i$ voor alle $1 \leq j \leq k$ behalve voor $j = i$ dan geldt $u_{ij} = 0$. Ook weten we dat $\sum_{j=1}^k q_j = 1$. Er geldt

$$V_k = u_i \sum_{j \neq i}^k q_j = u_i(1 - q_i)V_k \text{ voor alle } i \leq k. \quad (2.19)$$

$$\frac{V_k}{u_i} = 1 - q_i \text{ voor alle } i \leq k. \quad (2.20)$$

$$q_i = 1 - \frac{V_k}{u_i} \text{ voor alle } i \leq k. \quad (2.21)$$

Hieruit volgt de oplossing voor q_i . De oplossing voor V_k volgt uit vergelijking (2.20) door beide kanten te sommeren over i . Merk op dat $\sum_{i=1}^k 1 - q_i = k - 1$. Er volgt:

$$V_k = \frac{k - 1}{\sum_{i=1}^k 1/u_i}. \quad (2.22)$$

Speler 1 heeft strategie $p = \{p_1, p_2, \dots, p_k, 0, \dots, 0\}$. Dan geldt vanwege indifference dat:

$$\sum_{i=1}^k p_i u_{ij} = V_k \text{ voor alle } j \leq k.$$

Omdat $u_j = 0$ voor alle i geldt dat

$$V_k = \sum_{i=1}^k p_i u_i - p_j u_j \text{ voor alle } j \leq k.$$

Als we (2.22) invullen voor V_k invullen krijgen we:

$$\begin{aligned} p_j u_j &= \sum_{i=1}^k p_i u_i - \frac{k - 1}{\sum_{i=1}^k 1/u_i} = \frac{\sum_{i=1}^k \sum_{i=1}^k p_i u_i (1/u_i)}{\sum_{i=1}^k 1/u_i} - \frac{k - 1}{\sum_{i=1}^k 1/u_i} \\ &= \frac{k}{\sum_{i=1}^k 1/u_i} - \frac{k - 1}{\sum_{i=1}^k 1/u_i} = \frac{1}{\sum_{i=1}^k 1/u_i}. \end{aligned}$$

Hieruit volgt:

$$p_j = \frac{1/u_j}{\sum_{i=1}^k 1/u_i} \text{ voor alle } j \leq k. \quad (2.23)$$

Er geldt $p_j > 0$ voor alle $1 \leq j \leq k$ en $q_i > 0$ wanneer $u_i > V_k$. Omdat $u_k \leq u_i$ voor alle i , moet er dus gelden dat $u_k > V_k$. Dus we zoeken naar de maximale k zodat geldt:

$$\frac{k-1}{u_k} \geq \frac{1}{\sum_{i=1}^k 1/u_i}. \quad (2.24)$$

Uit de resultaten zien we dat p_i daalt naarmate u_i stijgt. Oftewel, de dief zal met grotere kans de huizen bestellen waar het minst te halen valt. Daar in tegen zien we dat q_i stijgt naarmate u_i stijgt. De agent zal dus met meer kans de huize bewaken waar meer te halen valt. Dit is interessant om te zien, breekt een dief in de werkelijkheid vaker in bij arme wijken? En bewaken agenten vooral rijke wijken?

In dit geval kan de dief slechts één huis bestellen en is er maar één agent die maar één huis bewaakt. In werkelijkheid zijn er natuurlijk meerdere agenten die zich verdelen over verschillende wijken. De dief kan ook meerdere huizen bestellen, in dit geval begint het spel op een stopprobleem te lijken. Dit geval zullen we in het volgende hoofdstuk bekijken en dan zullen we zien of de dief en de agenten hetzelfde gedrag vertonen.

Het dievenspel als stopprobleem

In dit hoofdstuk breiden we het probleem van hoofdstuk 2 uit zodat er meer agenten zijn en de dief meerdere huizen kan beroven. Zo proberen we door wat we weten over stopproblemen een sociologisch spel uit de speltheorie op te lossen.

3.1. Huizen met dezelfde waarde

In het vorige hoofdstuk kon de dief maar één huis bestellen. We vragen ons nu af wat er gebeurt als de dief meer huizen kan bestellen en of we in dit geval de theorie kunnen gebruiken voor een stopprobleem. Neem n het aantal huizen en a het aantal agenten om deze huizen te bewaken. Er geldt $1 \leq a < n$. We nemen daarnaast aan dat de huizen dezelfde waarde hebben, voor het gemak nemen we aan dat alle huizen waarde 1 hebben. In Lidbetter en Lin [5] wordt dit probleem ook opgelost. We kijken hier of het probleem ook als stopprobleem op te lossen is. Wanneer de dief een huis heeft berooft, mag hij kiezen of hij doorgaat of niet. Als de dief stopt, houdt hij zijn verkregen buit. De dief kan ook doorgaan met huizen bestellen. Als de dief een huis kiest waar een agent staat is hij al zijn buit kwijt. We praten over een nulsomspel, de buit die de dief wint is wat de agenten verliezen. We nemen dan ook niet mee de eventuele schade die de huizen oplopen door de inbraak.

We zoeken nu naar de optimale stopregel voor de dief. De strategie van de agent is nu logisch aangezien alle huizen dezelfde waarde hebben. De agenten zullen dan een huis bewaken met kans $1/n$. De dief bestelt m huizen. Omdat de huizen allemaal waarde 1 hebben, is de buit van de dief ook gelijk aan m .

We willen de 1-sla toepassen om een stopregel te bepalen. Neem $X_1, X_2, \dots, X_n$ de Bernoulli variabelen representatief voor het bestellen van huizen $1, 2, \dots, n$. We noemen $X_i = 1$ een succes voor $1 \leq i \leq n$ als het huis niet bewaakt wordt en de dief door kan gaan met het bestellen van huizen. Die dief wil nu stoppen bij het laatste succes en er geldt $\mathbb{P}(X_i = 1) = \frac{n-i-a}{n-i}$. We hebben dus te maken met een eindige rij afhankelijke Bernoulli variabelen, maar toch kunnen we de 1-sla niet zomaar toepassen. Wanneer er een niet-succes voorkomt, stopt het spel en is de dief al zijn buit kwijt. We moeten dus de vergelijkingen voor de verwachte beloning van de speler opnieuw bekijken.

De dief krijgt buit m wanneer hij m huizen bestelt en deze niet worden bewaakt. Anders is zijn buit 0. De verwachte buit van een dief als hij m huizen bestelt is dan gelijk aan m vermenigvuldigd met de kans dat de agenten de huizen $n - m$ bewaken. Er geldt voor de verwachte buit Y_m :

$$Y_m = \frac{\binom{n-m}{a}}{\binom{n}{a}} m. \quad (3.1)$$

Dit is vergelijkbaar met de verwachte beloning Y_n in vergelijking (1.19) die we nodig hadden bij de 1-sla.

Nu kijken we naar de verwachting van Y_{m+1} welke te vergelijken is met vergelijking (1.21). Wanneer we m in vergelijking (3.1) vervangen met $m + 1$ zien we:

$$Y_{m+1} = \frac{\binom{n-(m+1)}{a}}{\binom{n}{a}}(m+1). \quad (3.2)$$

Volgens de 1-sla moeten we stoppen wanneer $Y_m \geq Y_{m+1}$. We nemen aan dat de dief wat risico neemt, dus wanneer $Y_m = Y_{m+1}$ zal de dief doorgaan met het bestellen van de huizen. We bekijken nu $\frac{Y_{m+1}}{Y_m}$, dit is gelijk aan:

$$\frac{Y_{m+1}}{Y_m} = \frac{(n-m-a)(m+1)}{(n-m)m}. \quad (3.3)$$

Hieruit volgt dat $Y_m > Y_{m+1}$ wanneer:

$$m > \frac{n-m-a}{n-m}(m+1). \quad (3.4)$$

Het uitwerken van bovenstaande vergelijking geeft de stopregel voor de dief. Stop zodra:

$$m > \frac{n-a}{a+1}. \quad (3.5)$$

Uit vergelijking (3.4) blijkt dat de dief stopt wanneer zijn huidige buit m hoger is dan dan zijn verwachte buit $\mathbb{P}(X_m = 1)(m+1)$. De dief kijkt dus elke keer wat hoger is: zijn huidige buit of zijn buit plus 1 vermenigvuldigd met de kans op een succes.

Uit de stopregel kunnen we een paar conclusies trekken. Als $a < n$ zal de dief altijd minstens één huis bestellen. Namelijk de kleinste kans op een succes is nog steeds beter dan geen buit. We houden dan ook geen rekening met de consequenties voor de dief als hij gepakt wordt. In werkelijkheid zal de dief het huis misschien niet bestellen als hij het risico loopt om de gevangenis in te gaan.

Wanneer het aantal agenten de helft van de huizen bezet zal de dief ook maar 1 huis bestellen. Want neem $a = \frac{1}{2}n$ dan wordt de stopregel: stop zodra $m \geq \frac{\frac{1}{2}n}{\frac{1}{2}n+1}$

Wanneer het aantal agenten gelijk is aan 1, stopt de dief zodra $m > \frac{n-1}{2}$. De dief zal dus nooit meer dan de helft van de huizen bestellen.

Merk op dat wanneer de dief m huizen bestelt, de buit van de dief alleen gelijk is aan m als hij niet wordt gepakt door de agent. De verwachte buit van de dief, en daarmee de waarde van het spel, is dan ook gelijk aan Y_m gegeven m . Deze is altijd lager dan m omdat hij bij elk huis kans heeft om een agent tegen te komen.

Neem bijvoorbeeld $n = 100$ en $a = 1$. Er volgt dan dat $m = 50$ en $Y_m = 25$. De dief zal dus 50 huizen willen bestellen. Stel hij doet dit meerdere malen dan zien we dat de dief maar in de helft van de gevallen geen agent tegenkomt.

In de volgende paragraaf bekijken we het gedrag van de dief wanneer hij huizen kan bestellen met verschillende waardes.

3.2. Huizen met verschillende waardes

In deze paragraaf proberen we het probleem op te lossen wanneer de huizen verschillende waardes hebben. Dit wordt niet behandeld door Lidbetter en Lin, dus we zoeken zelf naar een oplossing.

Net als in het vorige hoofdstuk zijn we geïnteresseerd in het gedrag van de dief als hij kan kiezen uit huizen van verschillende waarde. We maken onderscheid tussen twee soorten huizen: huizen met waarde y en huizen met waarde x , waarbij $y > x$. Voor het gemak noemen we dit rijke en arme huizen respectievelijk. In dit geval is de verzameling van de strategieën voor de dief erg groot. Hij kan eerste eerst een arm huis bestellen, dan een rijk huis of weer een arm huis. Zo zijn er veel verschillende mogelijkheden voor de dief. Ook is de verdeling van de agent nu moeilijker te bepalen. Zij moeten zich slim verdelen over de rijke en arme huizen zodat de buit van de dief zo klein mogelijk blijft.

Neem n het aantal rijke huizen en k het aantal arme huizen. De agenten zullen zich vervolgens verdelen over deze huizen. Het aantal agenten bij de rijke huizen noemen we a en bij de arme huizen b . Het totaal aantal agenten is dan $a + b$ en is constant.

Neem i het aantal bestolen arme huizen en j het aantal bestolen rijke huizen. De huidige buit van de dief is nu gelijk aan $yj + xi$. Vervolgens kan de dief beslissen of hij een arm huis wil bestellen, een rijk huis bestellen of stoppen. We nemen vergelijking (3.4) en bekijken de aparte gevallen. Wanneer die dief alleen rijke huizen besteelt moet hij stoppen wanneer:

$$yj + xi > \frac{n - a - j}{n - j}(yj + xi + y). \quad (3.6)$$

Als de dief alleen de arme huizen besteelt moet hij stoppen wanneer:

$$yj + xi > \frac{k - b - i}{k - i}(yj + xi + x). \quad (3.7)$$

De dief kan nu ook veranderen van zijn keuze, hij kan eerst arme huizen beroven en vervolgens rijke, en ook andersom. Hij vergelijkt dus steeds zijn huidige buit en de verwachte opbrengst van het bestellen van een nieuw huis.

De buit van de dief $yj + xi$ is zijn buit als hij geen agent tegenkomt. Zijn verwachte buit is dan ook anders, zoals we zagen in de vorige paragraaf. Omdat in dit geval lastig aan te tonen is wat de verwachte buit is en we vooral geïnteresseerd zijn in het gedrag van die dief, gaan we hier niet verder op in.

3.2.1. Greedy stopregel

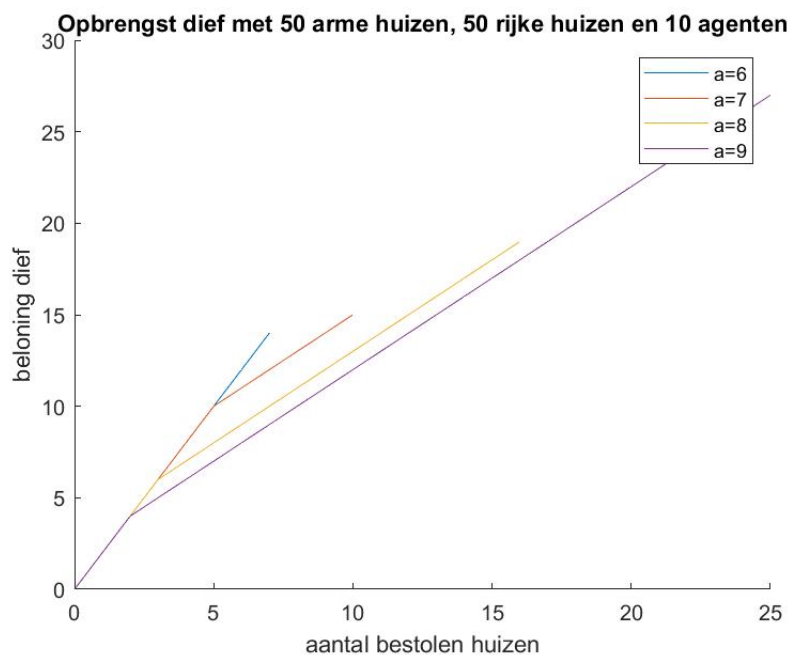
Zoals al eerder werd genoemd is de verzameling van strategieën van de dief erg groot. We kiezen eerst een voor de hand liggende stopregel voor de dief om zijn buit te maximaliseren. We nemen het volgende: de dief houdt zich aan stopregels (3.6) en (3.7). Alleen vergelijkt hij ook de verwachte buit onderling. Elke keer zal de dief kijken wat hoger is in verwachting: het bestellen van een rijk huis R , een arm huis A of stoppen H . Dit zal hij doen met onderstaande waardes:

$$\begin{cases} H &= yj + ix \\ R &= (y + yj + xi) \frac{n - a - j}{n - j} \\ A &= (x + yj + xi) \frac{k - b - i}{k - i} \end{cases} \quad (3.8)$$

Vervolgens volgt de dief een algoritme. Hij zal kijken welke van de drie opties het beste resultaat geeft. Kiest hij voor een rijk huis dan volgt $j = j + 1$ en dan kiest de dief weer opnieuw. Hetzelfde geldt wanneer de dief een arm huis besteelt, dan geldt $i = i + 1$. Wanneer het huidige bedrag groter is dan de optie om een arm of rijk huis te bestellen zal de dief daarvoor kiezen en geen huizen meer bestellen.

Dit algoritme wordt ook wel een **Greedy Algoritme** genoemd. Dit is een algoritme waar de optimale oplossing gezocht wordt door het kiezen van een lokaal optimum. We zullen deze stopregel van de dief dan ook de Greedy stopregel noemen. We bekijken het gedrag van de dief aan de hand van een voorbeeld.

We nemen het aantal rijke en arme huizen gelijk aan $n = k = 50$. Daarnaast nemen we de waarde van de arme huizen gelijk aan $x = 1$ en de waarde van de rijke huizen gelijk aan $y = 2$. Het totaal aantal agenten is gelijk aan 10. Er geldt dan $b = 10 - a$ en we bekijken de werkwijze van de dief voor verschillende waarden van a .



Figuur 3.1: We zien de beloning van de dief bij verschillende waarde voor a . Daarnaast zien we het aantal huizen dat de dief besteelt. Wanneer de lijn het steilst is, berooft de dief een rijk huis.

Wanneer de lijn het steilst is betekent dit dat de dief steelt van de rijke huizen. Wordt de lijn minder steil steelt de dief bij arme huizen.

De buit van de dief is weergegeven voor verschillende waarde van a . Dit is de buit van de dief als hij geen agenten tegenkomt. Zijn verwachte buit zal dan ook lager liggen omdat de dief in sommige gevallen een agent zal tegenkomen en zijn buit gelijk is aan 0.

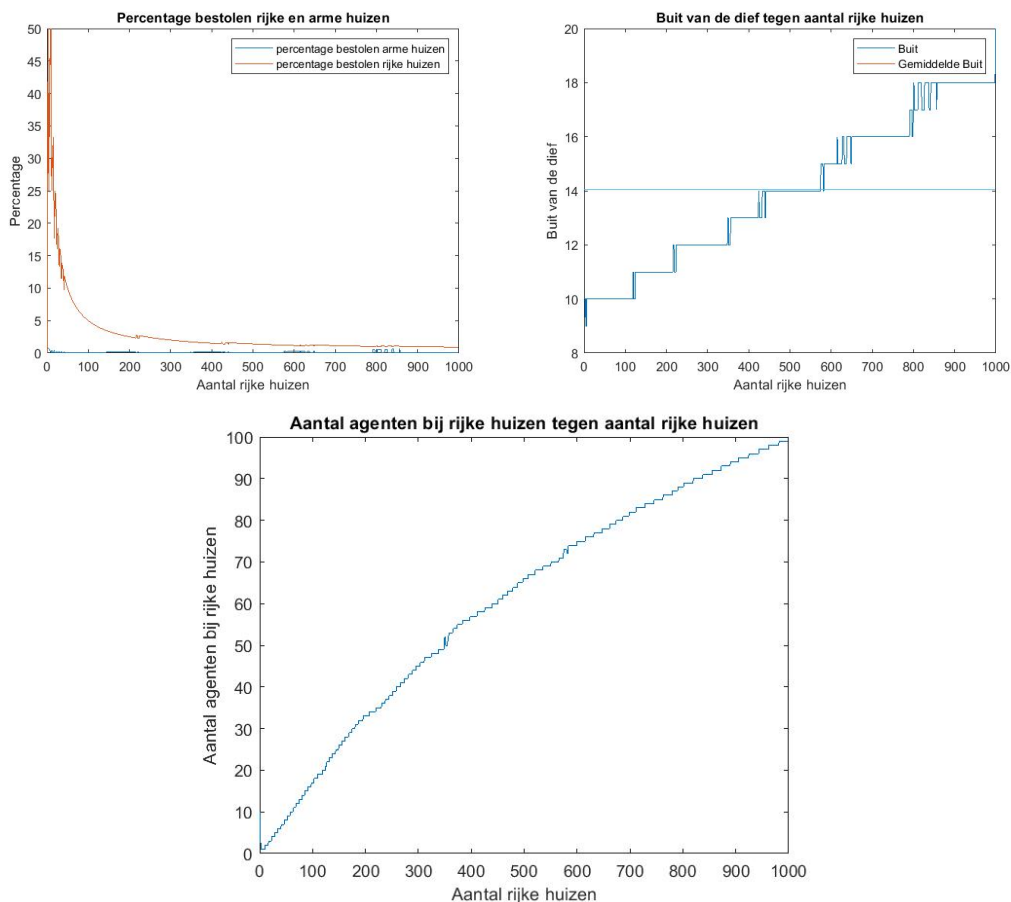
Uit het gedrag van de dief vallen een paar dingen op. In het geval als $a \leq 6$ zal de dief alleen maar rijke huizen beroven. Daartegenover met $b = 1$ en $a = 9$ zal de dief nog steeds twee rijke huizen bestellen. De dief lijkt dus al snel een voorkeur te hebben voor het bestellen van rijke huizen met deze stopregel. Daarnaast lijkt het wanneer de dief een arm huis berooft, hij nooit meer terug gaat naar het bestellen van een rijk huis. Dit omdat, wanneer de dief een arm huis besteelt H sneller stijgt dan R . Dit is een vermoeden wat lastig is om te bewijzen voor alle gevallen, en laten we bij een vermoeden.

De agenten willen zich zo verdelen zodat de buit van de dief minimaal is wanneer hij geen agenten tegenkomt. In bovenstaand voorbeeld zullen de agenten dus zich verdelen met $a = 6$ en $b = 4$.

3.2.2. Numerieke resultaten

Wanneer de dief gebruik maakt van de Greedy stopregel willen de agenten zich zo verdelen dat de buit van de dief minimaal is wanneer de dief geen agent tegenkomt. Hoe zullen de agenten zich verdelen over de huizen? Voor een gegeven aantal rijke en arme huizen, en een vast aantal agenten, kunnen de agenten zich zo verdelen over a en b zodat de buit van die dief zo laag mogelijk is. We bekijken deze verdeling met behulp van Matlab.

We nemen het aantal huizen gelijk aan 1000 en het aantal agenten gelijk aan 100. We laten het aantal rijke huizen, n , lopen van 1 tot en met 999. Het aantal arme huize is dan gelijk aan $k = 1000 - n$. Voor elke $1 \leq n \leq 999$ bekijken de agenten met welke waarde voor $1 \leq a \leq 99$ de buit van de dief minimaal is. Met $b = 100 - a$, $x = 1$ en $y = 2$. Verder telt het programma voor elke n en elke a hoe vaak de dief na het bestellen van een arm huis nog een rijk huis besteelt. De resultaten zijn te zien in onderstaande grafiek.



Figuur 3.2: Met het aantal huizen gelijk aan 1000, het aantal agenten gelijk aan 100 en de dief die de Greedy stopregel gebruikt. Zien we de verdeling van de agenten voor voor $1 \leq n \leq 999$. Daarnaast zien we de buit van de dief bij deze agentenverdeling als hij geen agenten tegenkomt met zijn gemiddelde buit. Ook zien we het percentage van de rijke huizen die wordt bestolen en het percentage van de arme huizen.

Als eerst geeft de code dat het aantal keer dat de dief een rijk huis besteelt na het bestellen van een arm huis gelijk is aan 0. Het vermoeden lijkt te kloppen.

Verder, net als in het vorige hoofdstuk zien we in figuur 3.2 dat de agenten meer bij de rijke huizen zullen staan.

Daarnaast zien we dat de buit van de dief wanneer hij geen agenten tegenkomt een stap functie lijkt. Dit komt doordat de agenten zich zo verdelen dat de buit van een dief een tijd constant blijft wanneer de rijke huizen toenemen. Dit tot een punt waar de buit van de dief hoger wordt omdat er meer rijke huizen zijn.

De dief steelt vooral bij rijke huizen en amper bij de arme huizen. Dit is in tegenstelling met de verwachting uit het vorige hoofdstuk. Dit heeft te maken met de stopregel omdat we eerder zagen dat de dief voorkeur heeft voor rijke huizen en daarna niet altijd nog een arm huis besteelt. Dit is terug te zien in de resultaten.

We hebben nu een stopregel gekozen voor de dief die werkt volgens het Greedy algoritme. Het Greedy algoritme is echter vaak niet optimaal. We zullen voor de dief een andere stopregel moeten verzinnen die een hogere buit geeft en wellicht meer arme huizen besteelt.

3.3. Optimale stopregel dief

Een optimale stopregel bepalen voor de dief is moeilijk omdat deze, zoals eerder gezegd, veel keuze heeft.

We zagen eerder met de Greedy stopregel dat de dief vooral begon met bestellen van rijke huizen en daarna met het bestellen van arme huizen. Dit resulteerde in het bestellen van veel rijke huizen en weinig arme huizen. Nu willen we een stopregel vinden voor de dief zodat hij kijkt naar alle opties voor het bestellen van eerst rijke huizen en vervolgens arme huizen, en alle opties voor het bestellen van eerst arme huizen en vervolgens rijke huizen. Vervolgens moet hij kiezen voor de combinatie van huizen bestellen zodat zijn beloning het hoogst is.

Dit is niet per se een stopregel die logisch zal klinken of te beschrijven is in één regel. Echter geeft dit wel een goed beeld met welke combinatie de dief zijn buit kan optimaliseren.

We schrijven (3.6) en (3.7) om en krijgen een bovengrens voor het aantal rijke en arme huizen die de dief kan bestellen. De dief moet stoppen met het bestellen van arme huizen wanneer:

$$i > \frac{(k - b) - \frac{y}{x}jb}{1 + b}. \quad (3.9)$$

De dief moet stoppen met het bestellen van rijke huizen wanneer:

$$j > \frac{(n - a) - \frac{x}{y}ia}{1 + a}. \quad (3.10)$$

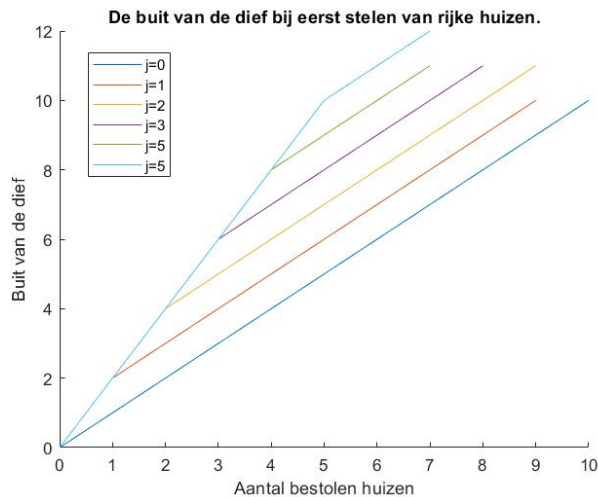
Gebruik makend van deze stopregels kunnen we zoveel mogelijk combinaties van het bestellen bekijken voor de dief. Namelijk uit (3.9) blijkt dat voor alle waarde voor j (het aantal rijke huizen die de dief bestolen heeft) er een maximale waarde is voor i (het aantal arme huizen die de dief nog kan bestellen). Hetzelfde geldt door (3.10). Voor alle waarde van i (het aantal arme huizen die de dief berooft) is er een maximale waarde voor j (het aantal rijke huizen die de dief besteelt).

Met een voorbeeld kunnen we zien hoe de dief te werk gaat. Neem $x = 1$, $y = 2$, $n = 50$, $k = 50$, $a = 6$ en $b = 4$ (dezelfde waardes als het eerdere voorbeeld met de Greedy stopregel). We kunnen de bovengrens voor i bekijken voor verschillende waarde van j . Of hoewel, hoeveel arme huizen kan de dief nog bestellen als deze eerst al rijke huizen bestolen heeft. Ook hier bekijken we alleen de buit van de dief als hij geen agenten tegenkomt. De bovengrens voor i is dan:

j	0	1	2	3	4	5
i	9,2	7,6	6	4,4	2,8	1,2

(3.11)

Hieruit zien we bijvoorbeeld dat wanneer de dief al drie rijke huizen bestolen heeft, de bovengrens voor i gelijk is aan 4,4. Dit betekent dat de dief nog vijf arme huizen kan bestellen voordat hij moet stoppen. Zijn buit is dan $2j + i = 2 * 4 + 5 = 13$. De uiteindelijke buit voor verschillende waardes van j is te zien in onderstaande grafiek:



Figuur 3.3: De dief besteelt eerst rijke huizen en dan arm. Voor verschillende waarde van j is de buit weergegeven. De maximale buit is hier 12.

Hetzelfde kunnen we doen voor het eerst bestelen van arme huizen. We bekijken de bovengrens voor j bij verschillende waarden van i .

i	0	1	2	3	4	5	6	7	8	9	10	
j	6,29	5,86	5,43	5	4,57	4,14	3,71	3,29	2,86	2,43	2	(3.12)

We hoeven in dit geval alleen de bovengrenzen van j te bekijken tot $i = 10$. Omdat we al eerder zagen dat i niet hoger kan zijn dan 10.

De buit voor verschillende waarden van i is te zien in onderstaande grafiek.



Figuur 3.4: De dief besteelt eerst arme huizen en dan arm. Voor verschillende waarde van i is de buit weergegeven. De maximale buit is hier 16.

Uit de grafieken zien we dat de maximale buit van de dief, wanneer hij geen agenten tegenkomt, gelijk is aan 16. Dit is wanneer $i = 10$ en $j = 3$. De dief zal dus eerst 10 arme huizen bestelen en dan nog 3 rijke huizen. Als we hiervoor de Greedy stopregel zouden gebruiken, zou de dief eerst 7 rijke huizen bestelen en daarna stoppen, de buit is dan dus 14.

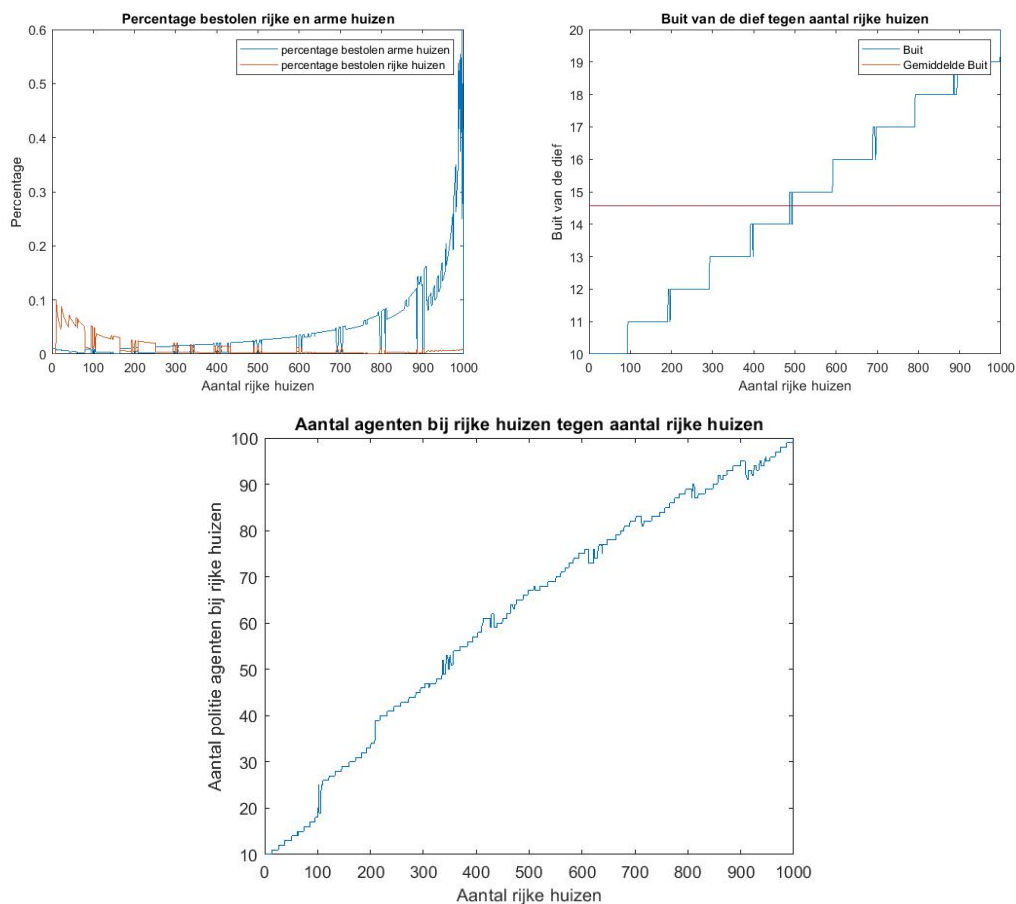
We verwachten dan ook dat bovenstaande methode altijd een beter of even goed resultaat voor de dief zal geven als het Greedy algoritme.

Met bovenstaande methode bekijkt de dief niet al zijn mogelijke strategieën. Hij kan namelijk veel vaker wisselen tussen het bestellen van rijk en arm in plaats van maar één keer. Toch is het vermoeden dat de dief op deze manier zijn maximale buit kan verkrijgen. We bekijken bovenstaand voorbeeld, stel de dief steelt eerst 2 keer bij een arm huis en vervolgens één keer bij een rijk huis. Nu wil de dief weer een arm huis beroven. De bovengrens voor i bij $j = 1$ is dan 7,6 en die dief kan 8 arme huizen beroven. Echter heeft de dief al twee arme huizen beroofd. Dit betekent dat hij nog maar 6 arme huizen kan bestellen. In dit geval zitten we dus weer bij $j = 1$ en $i = 8$.

Dit vermoeden is echter moeilijk te bewijzen voor alle gevallen, dus het blijft bij een vermoeden. We noemen bovenstaande methode dan ook de optimale stopregel van de dief.

3.3.1. Numerieke resultaten

Nu we een optimale stopregel hebben voor de dief, kunnen de agenten zich weer zo verdelen dat de buit van de dief minimaal is. Dit berekenen we op dezelfde manier als eerder met het Greedy algoritme. Dus neem het aantal huizen gelijk aan 1000 en het aantal agenten gelijk aan 100. We laten het aantal rijke huizen n lopen van 1 tot en met 999 met $k = 1000 - n$. Voor elke $1 \leq n \leq 999$ bekijken de agenten met welke waarde voor $1 \leq a \leq 99$ de buit van de dief minimaal is wanneer deze geen agenten tegenkomt. Met $b = 100 - a$. Daarnaast nemen we $x = 1$ en $y = 2$. De resultaten zijn als volgt:

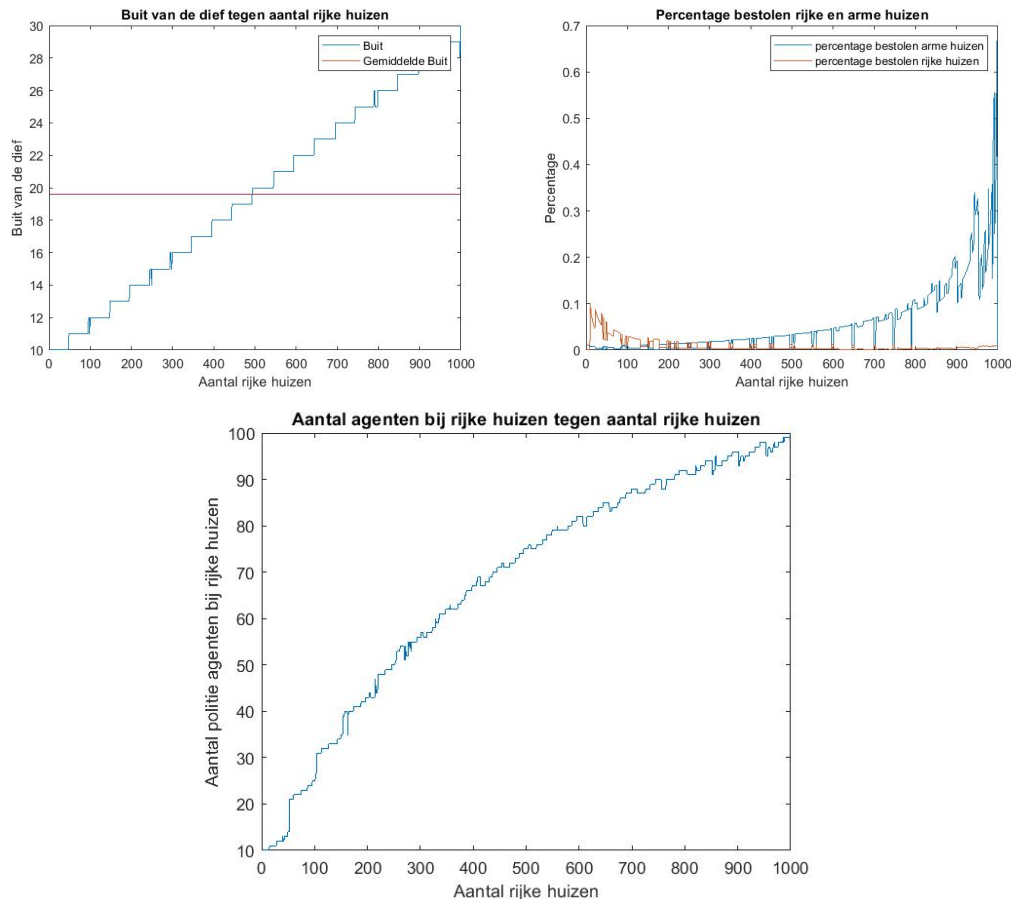


Figuur 3.5: Met het aantal huizen gelijk aan 1000, het aantal agenten gelijk aan 100 en de dief die de optimale stopregel gebruikt zien we de verdeling van de agenten voor $1 \leq n \leq 100$. Daarnaast zien we de buit van de dief bij deze agentenverdeling met zijn gemiddelde buit. Ook zien we het percentage van de rijke huizen die wordt bestolen en het percentage van de arme huizen.

Er vallen een aantal dingen op aan de resultaten. Als eerste is de gemiddelde buit van de dief wanneer hij geen agenten tegenkomt, hoger dan bij het Greedy algoritme. De optimale stopregel is in dit geval dus inderdaad beter.

Daarnaast zien we dat de dief nu wel meer bij de arme huizen inbreekt. Het percentage rijke huizen komt deze keer niet boven de 10%. Bij de verdeling van de agenten is nog steeds wel een bolling te zien. De agenten lijken zich dus ook meer bij de rijke huizen te verdelen maar het is minder duidelijk te zien dan bij het Greedy algoritme.

Om dit te verduidelijken nemen we nu waarde $x = 1$ en $y = 3$. Het verschil in rijk en arm is nog groter en de vraag is wat er verandert in het gedrag van de dief en de agenten. Verder nemen we alle variabelen hetzelfde. De resultaten zijn als volgt:



Figuur 3.6: Met het aantal huizen gelijk aan 1000, het aantal agenten gelijk aan 100 en de dief die de optimale stopregel gebruikt zien we de verdeling van de agenten voor $1 \leq n \leq 100$. Daarnaast zien we de buit van de dief bij deze agentenverdeling met zijn gemiddelde buit. Ook zien we het percentage van de rijke huizen die wordt bestolen en het percentage van de arme huizen.

Het is bijzonder om te zien dat wanneer het verschil in rijk en arm nog groter is, de dief weer vooral voor de arme huizen kiest. De agenten daar in tegen bewaken vooral de rijke huizen. Wanneer de dief mag kiezen uit verschillende huizen met verschillende waarden en hij de optimale stopregel gebruikt zien we hetzelfde gedrag voor de dief en de agenten zoals eerder. De agenten bewaken vooral de rijke huizen en de dief steelt vooral van de arme huizen.

3.4. Conclusie en discussie

Als eerst hebben we gekeken naar het geval waar de dief één huis mag bestellen uit verschillende huizen met verschillende waarde en waar er maar een agent is. Dit probleem was op te lossen gebruikmakende van stellingen uit de speltheorie. Hieruit bleek dat de optimale strategie van de agenten is om met grotere kans bij huizen van hogere waarde staan terwijl de optimale strategie van de dief was om met grotere kans een huis te bestellen van kleinere waarde. Dit spel was te beschouwen als een nulsomspel met twee spelers, de agenten en de dief, en daarom nog goed op te lossen. De volgende gevallen bleken al snel lastiger te zijn om op te lossen.

We hebben vervolgens gekeken naar het geval waar de dief door kan gaan met het bestellen van huizen. Dit is een lastiger probleem waar de eerder gebruikte stellingen niet meer konden helpen. De theorie voor de stopproblemen konden we gebruiken maar helaas minder goed dan gedacht. Dit was jammer omdat we stopproblemen met de bijbehorende stopregels uitgebreid behandeld hebben.

Voor de huizen met dezelfde waarde konden we een optimale stopregel vinden voor de dief met een logische strategie voor de agenten. Ook hebben we hier een beeld geschetst over de verwachte buit voor de dief. Dit hebben we in de verdere uitbreiding weggelaten omdat dit op het eerste gezicht een lastige opgave leek en we dit niet nodig hebben om het gedrag van de dief en agenten te bepalen.

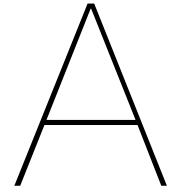
Het geval van verschillende waardes voor de huizen is een stuk lastiger en we hebben er daarom voor gekozen enkel te kijken naar twee verschillende waardes, rijk en arm. Vervolgens hebben we twee stopregels van de dief onderzocht. In werkelijkheid zijn er talloze stopregels te bedenken. Echter is de verwachting dat onze uiteindelijke stopregel optimaal is.

Met onze gekozen variabelen voor een totaal aantal huizen van 1000 met 10% agenten bezetting en een verhouding tussen rijk en arm van 1 op 3 bezettingen, zijn we weer op het resultaat gekomen dat de dief voorkeur heeft voor arme huizen en de agenten voor rijke huizen. Natuurlijk zijn er nog meer variabelen mogelijk om dit te onderzoeken maar dit hebben we niet bekeken.

De eerste vraag die bij deze conclusie komt kijken is of dit representatief is voor de werkelijkheid. Dit is discutabel vanwege onze aannames. We hebben aangenomen dat de dief in zijn beslissing geen rekening houdt met de consequenties van een straf. Het contrast met werkelijkheid was interessant geweest om uit te zoeken. Vanwege het niet kunnen vinden van geschikte datasets voor inbraakcijfers van verschillende wijken van steden, hebben we ervoor gekozen dit niet verder te onderzoeken. Daarnaast was het ook al gauw lastig geweest conclusies te trekken met hoe de agenten zich in de werkelijkheid zouden verdelen over deze wijken.

Wel is er een artikel beschikbaar van het CBS die concludeert dat inbrekers vaker actief zijn in duurdere buurten ¹. Dit is in tegenstelling met onze conclusie. Dit kan komen doordat zowel de politie agenten als de dieven niet hun optimale strategie gebruiken. Voor een dief kan het logischer klinken om bij een huis uit een dure buurt in te breken. Vervolgens kan het zijn dat de agenten niet hun optimale strategie gebruiken en deze dure buurten niet meer bewaken dan minder dure buurten. Mijn advies aan de agenten zou dan ook zijn om deze dure buurten meer te bewaken.

¹zie: <https://www.cbs.nl/nl-nl/nieuws/2019/10/inbrekers-vaker-actief-in-duurdere-buurten>



Matlab code

A.1. Functies Greedy stopregel

```
1 function p = armehuizen(k,b,i,j)
2     p = (1+2*j+i)*((k-i-b)/(k-i));
3     %p = (k-i-b)/b;
4     if k-i == 0
5         p = 0
6     end
7 end
```

```
1 function q = rijkehuizen(n,a,i,j)
2     q = (2+2*j+i)*((n-j-a)/(n-j));
3     if n-j == 0
4         q = 0
5     end
6     %q = (2*(n-a-j))/a;
7 end
```

```
1 function d = huidigbedrag(i,j)
2     d = i + 2*j;
3 end
```

A.2. Code figuur 3.1

```
1  totaalhuizen = 100;
2  totaalagenten = 10;
3  n=50;
4  k=totaalhuizen-n;
5  for a = 6:9
6      b=totaalagenten-a;
7      i=0;
8      j=0;
9      array=[0];
10
11     while huidigbedrag(i,j)< (2*n+k)
12         %huidigbedrag(i,j)
13         %armehuizen(k,b,i,j)
14         %rijkehuizen(n,a,i,j)
15         if rijkehuizen(n,a,i,j) ≥ huidigbedrag(i,j) & rijkehuizen(n,a,i,j) ≥ ...
16             armehuizen(k,b,i,j)
17             j=j+1
18         elseif armehuizen(k,b,i,j) ≥ huidigbedrag(i,j)
19             i=i+1
20         else
21             break
22         end
23         array=[array huidigbedrag(i,j)]
24     end
25     X=0:(i+j);
26     hold on
27     plot(X, array)
28 end
29 title('Opbrengst dief met 50 arme huizen, 50 rijke huizen en 10 agenten')
30 xlabel('aantal bestolen huizen')
31 ylabel('beloning dief')
32 hold off
33 legend('a=6','a=7','a=8','a=9')
```

A.3. Code figuur 3.2

```
1  totaalhuizen = 1000
2  totaalagenten = 100
3  X=[];
4  Y=[];
5  Y1=[];
6  Y2=[];
7  alowarray=[];
8  Built=[];
9  gembuit=0;
10 rijknaarm=0;
11 for n = 1: (totaalhuizen-1)
12     k=totaalhuizen-n;
13     reversea= 1:(totaalagenten-1);
14     arraylow=[2*n+k];
15     alow=totaalagenten;
16     for a = totaalagenten-reversea
17         b=totaalagenten-a;
18         j=0;
19         i=0;
20         array=[0];
21         while huidigbedrag(i,j)< (2*n+k)
22             if rijkehuizen(n,a,i,j) ≥ huidigbedrag(i,j) & rijkehuizen(n,a,i,j) ≥ ...
23                 armehuizen(k,b,i,j)
24                 j=j+1;
25                 if i ≠ 0
26                     rijknaarm=rijknaarm+1;
27                 end
28             elseif armehuizen(k,b,i,j) ≥ huidigbedrag(i,j)
```

```

28         i=i+1;
29     else
30         break
31     end
32     array=[array huidigbedrag(i,j)];
33 end
34 %a
35 %array
36 if max(array) < max(arraylow)
37     arraylow = array;
38     alow=a;
39     J=j;
40     I=i;
41     buit=max(array);
42 elseif max(array) == max(arraylow);
43     if length(array) < length(arraylow)
44         arraylow = array;
45         alow=a;
46         J=j;
47         I=i;
48         buit=max(array);
49     end
50 end
51 end
52
53     alowarray = [alowarray alow];
54     X=[X n];
55     Y=[Y (alow*100/totaalagenten)];
56     %Y1=[Y1 max(arraylow)];
57     Y1=[Y1 (I/k)*100];
58     Y2=[Y2 (J/n)*100];
59     gembuit=gembuit + buit;
60     Buit=[Buit buit];
61 end
62 %X
63 %Y
64 figure
65 plot(X,Y)
66 title('Aantal agenten bij rijke huizen tegen aantal rijke huizen')
67 xlabel('Aantal rijke huizen')
68 ylabel('Aantal agenten bij rijke huizen')
69
70 figure
71 plot(X,Y1)
72 hold on
73 plot(X,Y2)
74 title('Percentage bestolen rijke en arme huizen')
75 xlabel('Aantal rijke huizen')
76 ylabel('Percentage')
77 hold off
78 legend('percentage bestolen arme huizen', 'percentage bestolen rijke huizen')
79
80 figure
81 plot(X, Buit)
82 hold on
83 plot(X, ones(totaalhuizen-1)*gembuit/n)
84 title('Buit van de dief tegen aantal rijke huizen')
85 xlabel('Aantal rijke huizen')
86 ylabel('Buit van de dief')
87 hold off
88 legend('Buit', 'Gemiddelde Buit')
89
90
91 %title('Beloning dief tegen aantal dure huizen')
92 %xlabel('Percentage dure huizen')
93 %ylabel('Beloning van de dief')

```

A.4. Code figuur 3.3 en 3.4

```
1  Totaalhuizen = 100
2  Totaalagenten = 10
3  n = 50
4  k = Totaalhuizen - n
5  a = 6
6  b = Totaalagenten - a
7  x=1
8  y=2
9
10 bovangrensi=[];
11 bovangrensj=[];
12 waardearray=[];
13 J=[];
14 I=[];
15
16
17 %maak een array met daarin de bovengrens van i. Dus hoeveel arme huizen
18 %kunnen er bestolen worden
19 for loopj=0:n
20     i=((k-b)-loopj*(y/x)*b)/(1+b);
21     if i<0
22         break
23     end
24     bovangrensi=[bovangrensi i];
25 end
26
27 %wanneer i=8,5 zal de dief 9 huizen bestellen, dus maxwaarde is bovengrens +
28 %1 en deze naar onderafgerond. Om lege arrays te voorkomen is de maxwaarde
29 %0 als er geen huis wordt bestolen
30
31 maxwaardei=floor(bovangrensi+1);
32 if isempty(maxwaardei)==1
33     maxwaardei=0;
34 end
35
36
37 for loopi = 0:k
38     j= ((n-a)-loopi*(x/y)*a)/((a+1));
39     if j<0
40         break
41     end
42     bovangrensj=[bovangrensj j];
43 end
44 maxwaardej=floor(bovangrensj+1);
45 if isempty(maxwaardej)==1
46     maxwaardej=0;
47 end
48
49
50 for j=0:max(maxwaardej);
51     if j>(length(maxwaardei)-1)
52         break
53     end
54     i=maxwaardei(j+1);
55     J= [J j];
56     I= [I i];
57     waardei=y*j+x*i;
58     waardearray = [waardearray waardei];
59 end
60
61 for i=0:max(maxwaardei)
62     if i>(length(maxwaardej)-1)
63         break
64     end
65     j=maxwaardej(i+1);
66     J= [J j];
67     I= [I i];
```

```

68     waardej=y*j+x*i;
69     waardearray = [waardearray waardej];
70 end
71 if isempty(waardearray) ==1
72     waardearray=[1000];
73 end
74
75 figure
76 for k = 1:6
77     j=0: J(k)
78     i=1: I(k)
79     resj = [y*j max(y*j)+i]
80     hold on
81     plot(0:(J(k)+I(k)), resj)
82
83 end
84 title('De buit van de dief bij eerst stelen van rijke huizen.')
85 xlabel('Aantal bestolen huizen')
86 ylabel('Buit van de dief')
87 hold off
88 legend('j=0','j=1','j=2','j=3','j=5','j=5')
89
90 figure
91 for k = 7:17
92     j=1: J(k)
93     i=0: I(k)
94     resi = [i (max(i)+y*j)]
95     hold on
96     plot(0:(J(k)+I(k)), resi)
97
98 end
99 title('De buit van de dief bij eerst stelen van arme huizen.')
100 xlabel('Aantal bestolen huizen')
101 ylabel('Buit van de dief')
102 hold off
103 legend('i=0','i=1','i=2','i=3','i=4','i=5','i=6','i=7','i=8','i=9','i=10')
104 waardearray
105 J
106 I

```

A.5. Code figuur 3.5 en 3.6

```
1  totaalhuizen = 1000
2  totaalagenten = 100
3  arrayhuizen=1:totaalhuizen;
4  x= 1%waarde arme huizen
5  y= 3%waarde rijke huizen
6  Besta=[];
7  N=1:totaalhuizen;
8  arrayI=[];
9  arrayJ=[];
10 Buit=[];
11 gembuit=0;
12
13 for n=N
14     k= totaalhuizen -n;
15     reversea= 1:(totaalagenten-1);
16     besta = 0;
17     waardearraya=[100000];
18     Ia=0;
19     Ja=0;
20     suma=10000;
21     for a = totaalagenten - reversea
22         b = totaalagenten -a;
23         bovengrensi=[];
24         bovengrensj=[];
25         waardearray=[];
26         J=[];
27         I=[];
28
29
30         %maak een array met daarin de bovengrens van i. Dus hoeveel arme huizen
31         %kunnen er bestolen worden
32         for loopj=0:n
33             i=((k-b)-loopj*(y/x)*b)/(1+b);
34             if i<0
35                 break
36             end
37             bovengrensi=[bovengrensi i];
38         end
39
40         %wanneer i=8,5 zal de dief 9 huizen bestellen, dus maxwaarde is bovengrens +
41         %1 en deze naar onderafgerond. Om lege arrays te voorkomen is de maxwaarde
42         %0 als er geen huis wordt bestolen
43
44         maxwaardei=floor(bovengrensi+1);
45         if isempty(maxwaardei)==1
46             maxwaardei=0;
47         end
48
49
50         for loopi = 0:k
51             j= ((n-a)-loopi*(x/y)*a)/((a+1));
52             if j<0
53                 break
54             end
55             bovengrensj=[bovengrensj j];
56         end
57
58         maxwaardej=floor(bovengrensj+1);
59         if isempty(maxwaardej)==1
60             maxwaardej=0;
61         end
62
63
64         for j=0:max(maxwaardej);
65             if j>(length(maxwaardei)-1)
66                 break
67             end
```



```

68     i=maxwaardei(j+1);
69     J= [J j];
70     I= [I i];
71     waardej=y*j+x*i;
72     waardearray = [waardearray waardej];
73 end
74
75     for i=0:max(maxwaardei)
76         if i>(length(maxwaardej)-1)
77             break
78         end
79         j=maxwaardej(i+1);
80         J= [J j];
81         I= [I i];
82         waardej=y*j+x*i;
83         waardearray = [waardearray waardej];
84     end
85
86
87     %hierboven worden de arrays: Waardearray, J en I gemaakt.
88     if isempty(waardearray) ==1
89         waardearray=[1000];
90     end
91
92
93     %agent vergelijkt max waardearrays
94     %wanner de maximale waarde gelijk is kiest agent de verdeling
95     %waarbij dief de minste opties heeft.
96     if max(waardearray) < max(waardearraya)
97         waardearraya = waardearray;
98         besta=a;
99         Ja=J;
100        Ia=I;
101        suma = sum(waardearray == (max(waardearray)));
102    elseif max(waardearray) == max(waardearraya);
103        if sum(waardearray == (max(waardearray))) < suma
104            waardearraya = waardearray;
105            besta=a;
106            Ja=J;
107            Ia=I;
108            suma = sum(waardearray == (max(waardearray)));
109        end
110    end
111    %voor a
112    end
113    %hebben nu besta, waardearraya, Ia, Ja
114
115    gemI = (Ia(waardearraya == (max(waardearraya))));
116    gemJ = (Ja(waardearraya == (max(waardearraya))));
117
118    arrayI = [arrayI gemI(end)/k];
119    arrayJ = [arrayJ gemJ(end)/n];
120    Besta = [Besta besta];
121    Buit=[Buit max(waardearraya)];
122    gembuit = gembuit + max(waardearraya);
123    %voor n
124    end
125    figure
126    plot(N,Besta)
127    title('Aantal agenten bij rijke huizen tegen aantal rijke huizen')
128    xlabel('Aantal rijke huizen')
129    ylabel('Aantal politie agenten bij rijke huizen')
130
131    figure
132    plot(N, arrayI)
133    hold on
134    plot(N, arrayJ)
135    title('Percentage bestolen rijke en arme huizen')
136    xlabel('Aantal rijke huizen')
137    ylabel('Percentage')
138    hold off

```

```

139 legend('percentage bestolen arme huizen', 'percentage bestolen rijke huizen')
140
141 figure
142 plot(N,Buit)
143 hold on
144 plot(N,ones(totaalhuizen)*gembuit/n)
145 title('Buit van de dief tegen aantal rijke huizen')
146 xlabel('Aantal rijke huizen')
147 ylabel('Buit van de dief')
148 hold off
149 legend('Buit','Gemiddelde Buit')
150
151 %plot(arrayhuizen,totaalwaarde)
152 %plot(0:(aantalj-1),arrayi)
153 %hold on
154 %plot(arrayj,0:(aantali-1))
155 %title('Bovengrens i en j')
156 %xlabel('j')
157 %ylabel('i')
158 %legend('bovengrensi','bovengrensj')

```

Bibliografie

- [1] F Thomas Bruss. Sum the odds to one and stop. *Annals of Probability*, pages 1384–1391, 2000.
- [2] Neal L Carothers. *Real analysis*. Cambridge University Press, 2000.
- [3] Thomas S Ferguson. The sum-the-odds theorem with application to a stopping game of sakaguchi. *Mathematica Applicanda*, 44(1), 2016.
- [4] Thomas S. Ferguson. Game theory, second edition, 2014. *Mathematics Department, UCLA*, 2020.
- [5] Thomas Lidbetter and Kyle Y Lin. A search game on a hypergraph with booby traps. *Theoretical Computer Science*, 2020.