

Stochastisch Lineair Programmeren

Hoe om te gaan met onzekerheid

Wietske Bastiaansen
Jan-Tino Brethouwer
Dennis Swart

Bacheloropdracht Technische Wiskunde
Begeleider: Prof. dr. R.J. Boucherie
20 juni 2016

UNIVERSITEIT TWENTE.

Voorwoord

De afgelopen negen weken hebben wij ons gebogen over *Stochastische Lineair Programmeren*. Ons onderzoek was gericht op het vinden van oplosmethoden hiervoor. Het verslag dat voor u ligt is het resultaat hiervan. Met deze opdracht sluiten wij onze bachelor Technische Wiskunde aan de Universiteit Twente te Enschede af.

In het bijzonder willen we graag Richard Boucherie bedanken voor de begeleiding bij de opdracht en alle wijze levenslessen. Ook willen we vriend en studiegenoot Hidde Wieringa bedanken voor de feedback op ons verslag.

Wietske Bastiaansen, s1304259

Jan-Tino Brethouwer, s1319167

Dennis Swart, s1367781

Samenvatting

Het probleem bij een stochastisch lineair programma is dat het lastig is om eenduidige uitspraken te doen over de toelaatbaarheid van oplossingen en optimaliteit. In dit literatuuronderzoek zijn verschillende oplosmethoden voor een stochastisch lineair programma bekeken.

Allereerst is er gekeken naar de *Oplosbaarheid en optimaliteit* van een stochastisch lineair programma. De kans op toelaatbaarheid en optimaliteit van een oplossing wordt behandeld en er worden betrouwbaarheidsintervallen afgeleid voor toelaatbaarheid en doelfunctie waarde van een zekere oplossing.

Vervolgens is er gekeken naar *Stochastic Programming*. Dit is een verzamelnaam voor methoden die gebruik maken van kennis van de kansverdeling. De methode die hier behandeld worden zijn: stochastische programmeren met recourse, scenario-analyse en chance constrained programming.

Ook wordt *Robust Optimization* behandeld. Hierbij wordt alleen aangenomen dat de realisaties van de stochastische variabelen in een begreind interval liggen. Er zal ook gekeken worden naar de verdiepende methode budget of uncertainty.

Tot slot wordt *Stability and Sensitivity Analysis* en *Fuzzy Programming* kort aangestipt.

In de conclusie worden alle behandelde methodes vergeleken en valt te lezen welke methode zich het best leent voor welke situatie.

Inhoudsopgave

1 Inleiding	4
1.1 Probleembeschrijving	4
1.2 Theoretische achtergrond	5
1.2.1 Equivalente notatie lineair programma	5
1.2.2 Basis variabelen en het Simplex Algoritme	6
1.2.3 Opsplitsen van deterministische en stochastische variabele	7
1.2.4 Stochastiek verplaatsen	8
2 Oplosbaarheid en optimaliteit	10
2.1 Kans op toelaatbaarheid van een oplossing en optimaliteit	12
2.1.1 Kansdichtheidsfunctie voor een product van stochastische variabelen	13
2.2 Een exact betrouwbaarheidsinterval voor de oplossing en doelfunctie	14
2.2.1 Onbekende variantie	15
2.2.2 Bekende variantie	18
2.3 Benadermethode voor kansdichtheidsfunctie en betrouwbaarheidsinterval	20
2.3.1 Oplossingsvector x	20
2.3.2 Doelfunctie	23
2.4 Een betrouwbaarheidsinterval voor niet-multivariaat normaal verdeelde elementen	24
2.4.1 Onafhankelijke en identiek verdeelde elementen	25
2.4.2 Onafhankelijke elementen met niet-identieke kansverdeling	26
2.4.3 Afhankelijke elementen met identieke kansverdeling	27
2.4.4 Afhankelijke niet-identiek verdeelde elementen	29
3 Stochastic Programming	30
3.1 Stochastisch programmeren met recourse	30
3.2 Scenario-analyse	31
3.3 The L-shaped Method	31
3.3.1 Bewijs van het algoritme	32
3.3.2 Het algoritme	33
3.3.3 Uitleg van het algoritme	34
3.4 Chance constrained programming	34
3.4.1 Sample Average Approximation	34
3.4.2 Chance constrained programming	35
3.5 Andere Stochastic Programming modellen	36
4 Fuzzy Programming	38
4.1 Het idee achter een Fuzzy number	38
4.2 Fuzzy Programming	39
4.2.1 Lineaire membership function	39
4.2.2 Driehoekige membership functie	39
5 Robust Optimization	40
5.1 Worst case scenario	40
5.2 Budget of uncertainty	41
6 Stability and Sensitivity Analysis	44
6.1 Analyse van de kansverdeling	44
6.2 Statistische analyse	44
7 Conclusie	45
8 Discussie	47
9 Symbolenlijst	48
10 Engelse begrippenlijst	50

1 Inleiding

In het algemeen kunnen optimalisatieproblemen met deterministische variabelen kunnen worden opgelost door te beginnen met het probleem om te schrijven naar de standaardvorm van een deterministisch lineair programma:

$$\begin{array}{ll} \min_x & c^T x \\ \text{s.t.} & Ax \leq b, \\ & x \geq 0, \end{array} \quad (1)$$

waarbij $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ en $x \in \mathbb{R}^n$

Maar helaas is het niet altijd het geval dat de matrix A en vectoren b en c deterministische variabelen zijn. Het kan ook voorkomen dat het stochastische variabelen zijn. Door de stochastische variant op dezelfde manier te beschrijven als de deterministische, kunnen ook optimalisatieproblemen waarbij onzekerheden een rol spelen opgelost worden door middel van een lineair programma.

Het voordeel van een lineair programma is dat het probleem inzichtelijk blijft en er bekende oplosmethoden voor zijn.

Als A , b en/of c stochastische variabelen zijn, wat betekent dit dan voor het probleem? Wat is dan een toegelaten oplossing en wat betekent optimaliteit? Wat voor technieken zijn er om een stochastisch lineair programma op te lossen? Dat zijn de vragen waar naar gekeken zal worden in dit verslag.

Op pagina 47 is een symbolenlijst te vinden en op pagina 49 een Engelse begrippenlijst. Engelse begrippen zullen schuin gedrukt staan, de betekenis is in de lijst te vinden.

1.1 Probleembeschrijving

In dit verslag zullen de volgende vragen beantwoord worden:

1. Wat betekent oplosbaarheid en optimaliteit voor een stochastisch lineair programma?
2. Wat voor technieken zijn er om een stochastisch lineair programma op te lossen?
3. In welke situaties is welke techniek aan te raden?

Minimaliseren heeft binnen stochastisch lineair programmeren een andere betekenis dan in het deterministische geval. Bij een deterministisch lineair programma weet je van te voren hoe de situatie in elkaar steekt en is er een, eventueel zelfs unieke, optimale oplossing. De kosten liggen namelijk vast en doordat ook de voorwaarden vast liggen, weet je ook of een oplossing toegelaten is. Bij stochastisch lineair programmeren ligt dit helaas een stuk lastiger. Voor de hand liggend zou zijn om het minimalisatieprobleem te benaderen met de verwachtingswaarde, om vervolgens de verwachte kosten te minimaliseren. Dan heb je op de lange termijn wellicht de beste oplossing gevonden, maar op de korte termijn zou het best kunnen dat risico nemen loont. Hoe veel risico wil je dan nemen? Wat is eigenlijk de kans op een optimale oplossing? Dit soort vragen worden ook meegenomen in dit verslag.

Zelfs als je een optimale oplossing gevonden zou hebben, kan het zijn dat de oplossing in sommige scenario's niet toegelaten is. Het is dan mogelijk om alsnog je oplossing toegelaten te maken, maar dit brengt natuurlijk ook kosten met zich mee, wat je optimale oplossing verandert. Dus ook hier wordt naar gekeken in dit verslag. Er wordt geen eenduidig antwoord gegeven op wat het begrip 'minimaal' precies is bij een stochastisch lineair programma. Wel wordt er naar verschillende situaties gekeken en beschreven hoe je met deze situaties om moet gaan. Een stochastisch lineair programma ziet er als volgt uit:

$$\begin{array}{ll} \min_x & c^T(\omega)x \\ \text{s.t.} & A(\omega)x \leq b(\omega), \\ & x \geq 0, \end{array} \quad (2)$$

met $x \in \mathbb{R}^n$ en $A(\omega)$, $b(\omega)$ en $c(\omega)$ multivariate stochastische variabelen.

Tenzij anders vermeld wordt de volgende definitie voor een multivariate stochastische variabele bedoeld:

Definitie 1: Multivariate stochastische variabelen

Een multivariate stochastische variabele $X(\omega)$ is een $m \times n$ matrix met elementen x_{ij} , $i = 1, \dots, m$ en $j = 1, \dots, n$ waarvoor geldt:

$$X(\omega) : \Omega \rightarrow \mathbb{R}^{m \times n}, \quad (3)$$

met Ω de uitkomstenruimte die alle mogelijk realisaties van ω bevat.

Definitie 2: Multivariate stochastische variabelen voor een stochastisch lineair programma

Voor het stochastisch lineair programma (2) zijn de stochastische variabelen A , b en c als volgt gedefinieerd:

$$[A(\omega), b(\omega), c(\omega)] : \Omega \rightarrow [\mathbb{R}^{m \times n}, \mathbb{R}^{m \times 1}, \mathbb{R}^{1 \times n}], \quad (4)$$

met uitkomstenruimte Ω die alle realisaties van ω bevat.

Als er in dit verslag gesproken wordt van een stochastische variabele, wordt deze definitie gebruikt.

1.2 Theoretische achtergrond

Om een stochastisch lineair programma te kunnen analyseren en oplostechieken te kunnen bekijken, zijn enkele theoretische aspecten van belang. Deze zullen in deze paragraaf behandeld worden.

Allereerst zal een equivalente notatie voor het stochastisch lineair programma (2) behandeld worden. Daarna zal het Simplex Algoritme van [17] bekeken worden. Ook zal behandeld worden hoe de stochastiek tussen $A(\omega)$, $b(\omega)$ en/of $c(\omega)$ verplaatst kan worden. Tot slot zal bewezen worden dat een multivariate stochastische variabele gesplitst kan worden in een deterministisch en stochastisch deel.

1.2.1 Equivalente notatie lineair programma

Neem het deterministisch lineair programma:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \geq 0, \end{aligned} \quad (5)$$

waarbij $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ en $c \in \mathbb{R}^n$.

Voor sommige toepassingen of methoden wordt de volgende notatie gebruikt:

$$\begin{aligned} \min \quad & c^* x \\ \text{s.t.} \quad & A^* x = b^*, \\ & x \geq 0. \end{aligned} \quad (6)$$

Om (5) om te schrijven naar (6), definieer een extra variabele x_{n+1} waarvoor geldt:

- $a_{i,n+1} = 1$, $i = 1, \dots, m$
- $c_{n+1} = 0$
- $x_{n+1} \geq 0$

Het deterministisch lineair programma (5) verandert dan als volgt in (6):

$$A^* = \begin{pmatrix} A & 0 \\ 0_{1 \times n} & 1 \end{pmatrix} \quad b^* = b \quad c^* = \begin{pmatrix} c \\ 0 \end{pmatrix}, \quad (7)$$

met $0_{1 \times n}$ een $1 \times n$ nulmatrix.

De doelfunctie is onveranderd gebleven, daardoor zal dezelfde optimale waarde in (5) en (6) gelden. De oplossing uit (6) is gelijk aan die in (5), al zal in (6) de variabele x_{n+1} een waarde hebben. Als deze waarde niet wordt meegenomen, wordt elke constraint weer een ongelijkheid, dus het deterministisch lineair programma (5) is equivalent aan het deterministisch lineair programma (6).

1.2.2 Basis variabelen en het Simplex Algoritme

Voor het oplossen van deterministisch lineair programma wordt vaak het Simplex Algoritme van [17] gebruikt. Dit algoritme maakt gebruik van basis en niet-basis variabelen. In deze paragraaf zal dit verder worden toege- licht en het Simplex Algoritme kort worden uitgelegd.

Basis- en niet-basisvariabelen

Beschouw het deterministisch lineair programma (6). Dan geldt:

Stelling 1.1

Een basisoplossing voor $Ax = b$ kan verkregen worden door $n - m$ variabelen gelijk te stellen aan 0 en het pro- bleem op te lossen voor de overgebleven m variabelen. Er wordt gebruikt gemaakt van het feit dat $n - m$ varia- belen gelijk aan 0 stellen ervoor zorgt dat er unieke waarde gevonden worden voor de overige m variabelen, wat erop neer komt dat de m overige variabelen lineair onafhankelijk zijn.

De m variabelen die niet gelijk zijn aan 0, zijn dan de basisvariabelen en de $n - m$ variabelen die gelijk zijn aan 0, zijn de niet-basisvariabelen.

Omdat de niet-basisvariabelen waarde 0 hebben, kan geschreven worden:

$$\begin{aligned} A_B x_B &= b, \\ c_B^T x_B &= z, \end{aligned} \tag{8}$$

met z de optimale waarde, A_B de kolommen uit A horend bij de basisvariabelen, x_B de basisvariabelen uit x , c_B^T de coëfficiënten uit de doelfunctie horende bij de basisvariabelen.

Het Simplex Algoritme

Het Simplex Algoritme bestaat uit de volgende stappen:

Stap 1 Schrijf het deterministisch lineair programma naar de vorm zoals gegeven in (6).

Stap 2 Zoek een basisoplossing x_B (deze is toegelaten, maar niet per se optimaal).

Stap 3 Bekijk of de huidige basisoplossing x_B optimaal is. Als de oplossing niet optimaal is wil dat zeggen dat de doelfunctie waarde z verlaagd kan worden door een niet-basisvariabelen toe te voegen aan de basis (waardoor een basis-variabelen de basis zal verlaten).

Stap 4 Als de huidige basisoplossing x_B niet optimaal is, bepaal welke niet-basisvariabelen in de basis moet komen om een betere oplossing¹ te krijgen.

Stap 5 Gebruik elementaire rij-operaties om de nieuwe basisoplossing x_B te vinden. Ga daarna terug naar stap 3.

Het Simplex Algoritme maakt dus gebruik van de notatie van het deterministisch lineair programma (6) met behulp van basis- en niet-basisvariabelen. Er geldt dus:

$$A_B x_B + A_N x_N = b \tag{9}$$

met A_N de kolommen uit A horend bij de niet-basisvariabelen en x_N de niet-basisvariabelen uit x .

A_B is een vierkante matrix $m \times m$ met lineair onafhankelijke kolommen, dus inverteerbaar. Er geldt:

$$x_B = A_B^{-1} b - A_B^{-1} A_N x_N \tag{10}$$

Zo kan ook de doelfunctie geschreven in termen van basis- en niet-basisvariabelen:

$$z = c_B x_B + c_N x_N \tag{11}$$

¹'een betere oplossing' wil zeggen een oplossing met een lagere waarde voor de doelfunctie.

met c_N de coëfficiënten uit c van de niet-basisvariabelen. Gebruik (10), hieruit volgt:

$$\begin{aligned} z &= c_B (A_B^{-1} b - A_B^{-1} A_N x_N) + c_N x_N \\ &= c_B A_B^{-1} b + (c_N - c_B A_B^{-1} A_N) x_N. \end{aligned} \quad (12)$$

Deze uitdrukking is minimaal dan en slechts dan als:

$$c_N \geq c_B A_B^{-1} A_N. \quad (13)$$

Immers, als (13) niet zou gelden, dan is er een minstens één element van x_N dat ongelijk aan nul gekozen kan worden, die ervoor zorgt dat de optimale waarde z kleiner is dan wanneer deze x_N gelijk is aan 0. Dus deze x_N moet in de basis en dus is de oplossing nog niet optimaal.

1.2.3 Opsplitsen van deterministische en stochastische variabele

Onder andere in [5] en [52] wordt vaak gebruikt gemaakt van een notatie voor het stochastisch lineair programma waarin de matrix $A(\omega)$ en vectoren $b(\omega)$ en $c(\omega)$ gesplitst worden in een deterministisch en stochastisch deel. Deze notatie wordt zo gekozen, omdat in vele praktische toepassingen er sprake is van een waarde (bijvoorbeeld een aankomsttijd van de bus) die kan fluctueren. Deze fluctuatie wordt weergegeven in een aparte matrix, respectievelijk vector. De stochastische component staat ook heel vaak voor de error in de deterministische waarde.

In deze paragraaf zal bewezen worden dat het splitsen van het deterministische en stochastische deel de kansverdeling van de gesplitste vector of matrix niet verandert. Als dat geldt, zullen de kansuitspraken gedaan over het gesplitste stochastisch lineair programma ook gelden voor (2).

Er wordt gebruik gemaakt van de volgende splitsing:

$$\begin{aligned} b(\omega) &= b + (b(\omega) - b) \\ &= b + \hat{b}(\omega), \end{aligned} \quad (14)$$

waarbij b deterministisch is en $\hat{b}(\omega)$ stochastisch met een bekende verdeling.

Gebruik²:

$$P(x + y \leq a) = F_{b+\hat{b}(\omega)}(a) = \int_{x+y \leq a} f_b(x) f_{\hat{b}(\omega)}(y) dx dy. \quad (15)$$

Omdat b geen stochastische variabele is, kan gebruikt worden dat $f_b(x) = b \forall x$. Dit geeft:

$$\begin{aligned} F_{b+\hat{b}(\omega)}(a) &= \int_{-\infty}^{a-x} \left(\int_{-\infty}^{\infty} f_b(x) dx \right) f_{\hat{b}(\omega)}(y) dy \\ &= \int_{-\infty}^{a-b} 1 \cdot f_{\hat{b}(\omega)}(y) dy \\ &= \int_{-\infty}^{a-b} f_{\hat{b}(\omega)}(y) dy \\ &= F_{\hat{b}(\omega)}(a - b). \end{aligned} \quad (16)$$

Dit betekent dat de cumulatieve kansverdelingsfunctie van $b + \hat{b}(\omega)$ niet beïnvloed wordt door het splitsen van de vector in een stochastisch en deterministisch deel. De stochastische vector is enkel verschoven.

■

Hetzelfde geldt voor het splitsen van $A(\omega)$ ³ en $c(\omega)$ uit (2).

²Om de notatie van integralen overzichtelijk te houden wordt bedoeld:

$$\int f_x(x) dx = \int \cdots \int f_x(x_1, \dots, x_n) dx_1 \cdots dx_n$$

voor een vector van lengte n .

³De matrix $A(\omega)$ kan geschreven worden als één vector van lengte $m \cdot n$ met alle elementen achter elkaar.

1.2.4 Stochastiek verplaatsen

Een standaard stochastisch lineair programma ziet er als volgt uit:

$$\begin{aligned} \min_x \quad & c^T(\omega)x \\ \text{s.t.} \quad & A(\omega)x \leq b(\omega), \\ & x \geq 0, \end{aligned} \tag{17}$$

met $A(\omega)$, $b(\omega)$ en $c(\omega)$ stochastische variabelen en $x \in \mathbb{R}^n$.

In sommige gevallen is het echter makkelijker om te werken met een stochastisch lineair programma waarin niet alle componenten $A(\omega)$, $b(\omega)$ of $c(\omega)$, stochastisch zijn. In deze paragraaf zal behandeld worden in welke gevallen het mogelijk is de stochastiek te verplaatsen van het ene naar het andere component. Indien de stochastiek verplaatst wordt, zal dit in het verslag bij de desbetreffende methode worden aangegeven.

Stochastiek van $b(\omega)$ naar $A(\omega)$

Het verplaatsen van de stochastiek van $b(\omega)$ naar $A(\omega)$ kan op de volgende manier gedaan worden: $A(\omega)x \leq b(\omega)$ bestaat uit m constraints:

$$a_{i1}x_1 + \dots + a_{in}x_n \leq b_i.$$

Deze worden voor $i = 1, \dots, m$ omgeschreven naar:

$$\begin{aligned} a_{i1}x_1 + \dots + a_{in}x_n - b_i x_{n+1} &\leq 0, \\ x_{n+1} &\leq 1, \\ -x_{n+1} &\leq -1, \end{aligned} \tag{18}$$

zoals beschreven in [10] en [52]. Er geldt dat $x_{n+1} = 1$, dus het nieuwe stochastisch lineair programma is volledig equivalent met het originele stochastisch lineair programma (2), indien $c_{n+1} = 0$.

Immers:

$$A^*(\omega) = \begin{pmatrix} A(\omega) & -b(\omega) \\ 0_{1 \times n} & 1 \\ 0_{1 \times n} & -1 \end{pmatrix} \quad b^* = \begin{pmatrix} 0_{m \times 1} \\ 1 \\ -1 \end{pmatrix} \quad c^*(\omega) = \begin{pmatrix} c(\omega) \\ 0 \end{pmatrix}, \tag{19}$$

waarbij $0_{i \times j}$ een $i \times j$ nulmatrix is.

Stochastiek van $c(\omega)$ naar $A(\omega)$

De stochastiek uit $c(\omega)$ is op vergelijkbare manier te verplaatsen naar de matrix $A(\omega)$, zoals beschreven in [9] en [52]. Dit is te doen door de kostenfunctie $c(\omega)^T x$ te veranderen naar z en de extra constraints $c(\omega)^T x \leq z$ en $c(\omega)^T x \geq z$ toe te voegen aan het stochastisch lineair programma. Het nieuwe stochastisch lineair programma ziet er dan als volgt uit:

$$\begin{aligned} \min_{x,z} \quad & z \\ \text{s.t.} \quad & A(\omega)x \leq b(\omega), \\ & c(\omega)^T x - z \leq 0, \\ & -c(\omega)^T x + z \leq 0, \\ & x \geq 0, z \in \mathbb{R}. \end{aligned} \tag{20}$$

Op deze manier is de stochastiek van $c(\omega)$ naar $A(\omega)$ verplaatst.

Er geldt hier wel dat z een dummy variabele is, die gelijk zal zijn aan $\sup c(\omega)^T x$. Dit is equivalent aan rekening houden met de slechtst mogelijke uitkomst van $c(\omega)$, wat terug zal komen bij hoofdstuk 5 over Robust Optimization.

Stochastiek van $A(\omega)$ en/of $b(\omega)$ naar $c(\omega)$

Dit is in het algemeen niet mogelijk. In [52] wordt uitgelegd dat dit niet zomaar kan, omdat de matrix $A(\omega)$ en vector $b(\omega)$ het oplossingsgebied van de vector x bepalen. Als deze matrix en/of vectoren worden veranderd, verandert dit gebied mee en is het verkregen stochastisch lineair programma niet meer equivalent aan (2).

De enige manier om de stochastiek van $b(\omega)$ naar $c(\omega)$ te krijgen, is om het duale probleem op te lossen, maar de interpretatie van de variabelen in het originele en duale probleem is anders dan bij het originele probleem.

Stochastiek van $A(\omega)$ naar $b(\omega)$

Het is mogelijk om de stochastiek van $A(\omega)$ naar $b(\omega)$ te verplaatsen, hierbij wordt gebruik gemaakt van de notatie van het stochastisch lineair programma in basis- en niet-basisvariabelen, zoals beschreven in paragraaf 1.2.2. Hier wordt ook gebruik gemaakt van het opsplitsen van een stochastische variabele in een deterministisch en een stochastisch gedeelte, zoals beschreven in paragraaf 1.2.3.

Er geldt dat:

$$\begin{aligned} x &\leq A_B^{-1}(\omega)b(\omega) \\ A_B x &\leq A_B A_B^{-1}(\omega)b(\omega). \end{aligned} \tag{21}$$

Dit is equivalent met stochastisch lineair programma (2), immers:

$$A^* = A_B \quad b^*(\omega) = A_B A_B^{-1}(\omega)b(\omega) \quad c^*(\omega) = c_B. \tag{22}$$

Echter werkt dit alleen als geldt in (2) dat $m = n$, anders is de matrix $A(\omega)$ niet vierkant en bestaat de inverse niet.

In dit verslag wordt enkel gebruik gemaakt van het verplaatsen van de stochastiek naar $A(\omega)$ van $b(\omega)$ en/of $c(\omega)$ dus zal op dit probleem niet verder ingegaan worden. Meer informatie hierover kan gevonden worden in [52] en [4].

2 Oplosbaarheid en optimaliteit

Bij het oplossen van een deterministisch lineair programma is een vector $x \geq 0$ een toegelaten oplossing als deze voldoet aan $Ax \leq b$. Het deterministisch lineair programma is oplosbaar als er een dergelijke x bestaat. Van alle vectoren x waarvoor dit geldt, is de oplossingsvector x optimaal als geldt dat deze de doelfunctie minimaliseert.

Bij een stochastisch lineair programma is oplosbaarheid en optimaliteit niet zo eenduidig gedefinieerd. Een vector x kan een toegelaten oplossing zijn voor één scenario van het stochastisch lineair programma, maar voor een ander juist weer niet. In dit hoofdstuk zal bekeken worden hoe toch de oplosbaarheid en de optimaliteit van een stochastisch lineair programma geanalyseerd kunnen worden.

Allereerst zal er gekeken worden naar de kans dat een zekere oplossing x toelaatbaar is en de kans op optimaliteit van een zekere oplossing, onder een zekere verdeling, en hoe deze kans bepaald kan worden. Vervolgens zal gekeken worden hoe, in het algemeen, de kansdichtheidsfunctie van een product van twee stochastische vectoren bepaald kan worden.

Daarna zullen twee methoden behandeld worden om een betrouwbaarheidsinterval op te stellen voor de toelaatbaarheid van een oplossing x en de waarde van de doelfunctie. Deze methoden maken gebruik van multivariaat normaal verdeelde variabelen in het stochastisch lineair programma (2). In de laatste paragraaf van dit hoofdstuk zal daarom gekeken worden naar de Centrale Limietstelling om ook problemen met niet-normaal verdeelde variabelen te kunnen analyseren.

Samenvatting hoofdstuk 2

In de onderstaande tabellen staat kort samengevat welke informatie bekend moet zijn over het stochastisch lineair programma (2) om de methoden, toegelicht in dit hoofdstuk, toe te passen. Verder staat vermeld wat het doel is van de methode en tot slot wat het een alternatief kan zijn om toch de methode te gebruiken indien niet de juiste informatie bekend is.

Oplossingsvector x

	Paragraaf 1	Paragraaf 2	Paragraaf 3	Paragraaf 4
Bekend	Basisoplossing x_B , kansdichtheidsfuncties voor $A^{-1}(\omega)$ en $b(\omega)$.	Normaal verdeelde elementen van $A(\omega)$.	$A(\omega)$ inverteerbaar, normaal verdeelde elementen van $A(\omega)$ en $b(\omega)$ met verwachting 0 en bekende variantie.	Stochastiek van $A(\omega)$ naar $b(\omega)$. Verwachting en variantie van $b^*(\omega)$. Voldoende constraints.
Resultaat	De kans dat x_B een toegelaten oplossing is.	Formule voor grenzen betrouwbaarheidsinterval voor toegelaten oplossingen x .	Benadering kansdichtheidsfunctie en formule voor grenzen betrouwbaarheidsinterval voor toegelaten oplossingen x .	Formule voor grenzen betrouwbaarheidsinterval voor toegelaten oplossingen x .
Alternatief	Kansdichtheidsfunctie benaderen. Zie [2] of paragraaf 3.	-	-	-

Tabel 1: Samenvatting methoden hoofdstuk 2 voor oplossingsvector x .

Doelfunctiewaarde $z = c^T(\omega)x$

	Paragraaf 1	Paragraaf 2	Paragraaf 3	Paragraaf 4
Bekend	Kansdichtheidsfunctie $f_{c(\omega)}(c_B, c_N)$.	Normaal verdeelde elementen van $c(\omega)$, σ^2 of schatter S^2 .	Normaal verdeelde elementen van $c(\omega)$ met verwachting 0 en bekende variantie.	Verwachting en variantie $c(\omega)$ en voldoende variabele.
Resultaat	De kans op optimaliteit voor een oplossing.	Formule voor betrouwbaarheidsinterval van doelfunctiewaarde.	Benadering kansdichtheidsfunctie en formule voor grenzen betrouwbaarheidsinterval van doelfunctiewaarde.	Formule voor grenzen betrouwbaarheidsinterval van doelfunctiewaarde.
Alternatief	Kansdichtheidsfunctie benaderen. Zie [2].	Centrale limietstelling, zie paragraaf 4.	-	-

Tabel 2: Samenvatting methoden hoofdstuk 2 voor de doelfunctiewaarde $c^T(\omega)x$.

De methode in paragraaf 1 vraagt kennis over de kansdichtheidsfuncties van $A^{-1}(\omega)$ en $b(\omega)$ maar geeft wel de kans op het toelaatbaar zijn van een oplossing, respectievelijk optimaliteit. De methoden in paragraaf 2 en 3 geven formules voor de grenzen van een betrouwbaarheidsinterval voor de toegelaten oplossingen en doelfunctiewaarde in $(1 - \alpha)\%$ van de realisaties ω . Het betrouwbaarheidsinterval voor de oplossingsvector x uit paragraaf 2 vertelt dat in $(1 - \alpha)\%$ van de realisaties de oplossing binnen de aangeven grenzen zal liggen. Als deze grenzen een toegelaten oplossing impliceren, is de oplossing dus in $(1 - \alpha)\%$ van de realisaties toelaatbaar. In paragraaf 3 geeft het gevonden betrouwbaarheidsinterval de grenzen voor elke variabele x_j waartussen deze moet liggen zodat de oplossing toelaatbaar is.

Hetzelfde geldt voor een betrouwbaarheidsinterval voor de doelfunctie. Deze vertelt dat in $(1 - \alpha)\%$ van de realisaties deze waarde tussen bepaalde grenzen zal liggen. Dit zegt niks over of deze doelfunctie waarde optimaal is voor bijbehorende x . Natuurlijk kan met deze kennis wel iets gezegd worden over optimaliteit. Voor een eindig aantal realisaties kan immers de voor elke realisatie de x gekozen worden met de laagst doelfunctiewaardegrenzen in $(1 - \alpha)\%$ van de gevallen.

In paragraaf 3 wordt ook beschreven hoe de kansdichtheidsfunctie van de oplossing x en doelfunctiewaarde benaderd kan worden. De kansdichtheidsfunctie van de oplossingsvector x kan gebruikt worden om de kans op toelaatbaarheid van een oplossing te bepalen, zoals gedaan in paragraaf 1.

In paragraaf 2 en 3 wordt gebruik gemaakt van normaal verdeelde elementen in de stochastische variabelen $A(\omega)$, $b(\omega)$ en $c(\omega)$. In paragraaf 4 wordt beschreven hoe de centrale limietstelling ingezet kan worden om voor niet-normaal verdeelde, onafhankelijke of afhankelijke, identiek of niet-identiek verdeelde elementen van $A(\omega)$, $b(\omega)$ en $c(\omega)$ toch de kansdichtheidsfunctie en grenzen van het betrouwbaarheidsinterval te kunnen bepalen voor toegelaten oplossing respectievelijk doelfunctiewaarde.

De verdeling van de elementen van $A(\omega)$, $b(\omega)$ en $c(\omega)$ en kennis van de kansdichtheidsfunctie spelen de grootste rol. Alle methoden in dit hoofdstuk kunnen omgaan met (on)afhankelijkheid tussen de elementen. Voor paragraaf 2 en 3 moet de verdeling normaal zijn en voor paragraaf 1 is de kansdichtheidsfunctie nodig. Met behulp van de centrale limietstelling in paragraaf 4 is het voor veel gevallen mogelijk de verdeling te benaderen met de standaardnormale verdeling, al zijn hier wel veel constraints en/of variabelen voor nodig.

2.1 Kans op toelaatbaarheid van een oplossing en optimaliteit

Eén van de grootste uitdagingen bij een stochastisch lineair programma is dat het niet eenduidig te bepalen is of een oplossing toegelaten is. Een oplossing kan immers in de ene realisatie van het probleem toegelaten zijn, maar in de andere niet. Niet iedere oplossing zal even vaak toegaten zijn. Om wat te kunnen zeggen over de toelaatbaarheid van een oplossing, zal in deze paragraaf gekeken worden wat de kans op toelaatbaarheid voor een oplossing x van het stochastisch lineair programma (2) is, zoals beschreven in [52].

Kans op toegelaten oplossing

Allereerst is er een basis oplossing x_B nodig. Deze kan verkregen worden door het Simplex Algoritme [17] te gebruiken op een realisatie van het stochastisch lineair programma (2). Indien een dergelijke x_B verkregen is, is het de vraag wat de kans is dat, voor alle realisaties van het stochastisch lineair programma, deze x_B een toegelaten oplossing is.

Veronderstel dat de kansdichtheidsfuncties voor matrix $A(\omega)$ en vector $b(\omega)$ bekend zijn: $f_{A(\omega)}(A)$ en $f_{b(\omega)}(b)$.

Gebruik dat geldt:

$$x_B = A_B^{-1}(\omega)b(\omega), \quad (23)$$

dus de kansdichtheidsfunctie van de oplossing x_B wordt volledig en alleen bepaald door de kansdichtheidsfuncties van $A(\omega)$ en $b(\omega)$. Merk op dat er hier gewerkt wordt met het Simplex Algoritme. Dit geeft de volgende stelling:

Stelling 2.1

Neem aan dat x_B een basis oplossing is voor een realisatie van (2), verkregen met behulp van het Simplex Algoritme met kansdichtheidsfunctie $f_{x_B}(x_B)$. De kans dat x_B toegelaten is, is:

$$P(x_B \geq 0) = \int_0^\infty f_{x_B}(x_B) dx_B. \quad (24)$$

Bewijs:

Het volgende moet gelden:

$$x_B \geq 0 \Rightarrow x_B \text{ toegelaten.} \quad (25)$$

Bekend is dat, voor een zekere $\omega^* \in \Omega$:

$$x \text{ is toegelaten} \Leftrightarrow A(\omega^*)x \leq b(\omega^*), \quad x \geq 0. \quad (26)$$

Gebruik dat:

$$A(\omega^*)x = A_B(\omega^*)x_B = A_B(\omega^*)A_B^{-1}(\omega^*)b = b(\omega^*), \quad (27)$$

zoals volgt uit (8). Hieruit volgt dat:

$$A(\omega^*)x = b(\omega^*) \quad \text{voor} \quad x_B \geq 0, \quad (28)$$

dus uit (26) volgt dat:

$$\text{Als } x_B \geq 0 \Rightarrow x_B \text{ is oplosbaar.} \quad (29)$$

■

Dus als de kansdichtheidsfunctie van een basisoplossing x_B bepaald kan worden, kan berekend worden wat de kans op oplosbaarheid voor alle $\omega \in \Omega$ van een basisoplossing x_B is die een toegelaten oplossing is voor een zekere ω^* . Het bepalen van de kansdichtheidsfunctie zal behandeld worden in de volgende paragraaf.

Kans op optimaliteit

Ook voor optimaliteit geldt dat voor de ene realisatie van het stochastisch lineair programma een oplossing optimaal kan zijn, terwijl deze voor een andere realisatie niet optimaal is (en soms niet eens een toegelaten oplossing is).

Het volgende optimaliteitscriterium wordt gegeven door [52], welke volgt uit het Simplex Algoritme.

$$x_B \text{ is optimaal} \Leftrightarrow c_N \geq c_B A_B^{-1} A_N, \quad (30)$$

waarbij c_B en c_N de coëfficiënten van de doelfunctie zijn na toepassingen van het Simplex Algoritme en voor de basis variabele en respectievelijk de niet-basis variabele. Dus de kans op optimaliteit is:

$$P(c_N \geq c_B A_B^{-1} A_N). \quad (31)$$

Er geldt nu:

Stelling 2.2

Neem aan dat $c(\omega)$ een kansdichtheidsfunctie $f_c(c_B, c_N)$ heeft. Veronderstel dat de matrix A deterministisch is. Dan is de kans dat een basis B optimaal is gegeven door:

$$P(c_N \geq c_B A_B^{-1} A_N) = \int \int_D f_c(c_B, c_N) dc_B dc_N, \quad (32)$$

waarbij geldt voor D :

$$D = \left\{ (c_B^1, \dots, c_B^m, c_N^1, \dots, c_N^{n-m}) \mid c_N^i \geq (A_B^{-1} A_N)_i c_B, \quad i = 1, \dots, m-n \right\}. \quad (33)$$

Bewijs:

Omdat (30) geldt en de kansdichtheidsfunctie wordt geïntegreerd over alle mogelijke waarden waarvoor dit criterium geldt, geeft dit de kans dat een zekere basis x_B optimaal is.

■

Dus als het mogelijk is om de kansdichtheidsfunctie $f_{c(\omega)}(c_B, c_N)$ te vinden, kan de kans op optimaliteit van een oplossing x_B bepaald worden.

2.1.1 Kansdichtheidsfunctie voor een product van stochastische variabelen

Om formule (24) toe te kunnen passen en dus de kans op toegelatenheid voor een oplossing x_B van stochastisch lineair programma (2) te kunnen bepalen, is de kansdichtheidsfunctie nodig van $x_B = A_B^{-1}(\omega)b(\omega)$. De kansdichtheidsfunctie van twee continue stochastische vectoren is door [40] gevonden.

Stelling 2.3

Afhankelijke stochastische vectoren X, Y waarvoor geldt: $U = XY$:

$$f_U(u) = \int_{-\infty}^{\infty} f_{X,Y}\left(x, \frac{u}{x}\right) \frac{1}{|x|} dx. \quad (34)$$

Onafhankelijke stochastische vectoren X, Y waarvoor geldt: $U = XY$:

$$f_U(u) = \int_{-\infty}^{\infty} f_X(x) f_Y\left(\frac{u}{x}\right) \frac{1}{|x|} dx. \quad (35)$$

Bewijs:

Gebruik:

$$P(U < u) = P(XY < u) = P\left(Y < \frac{u}{x}, X \in \mathbb{R}\right). \quad (36)$$

Gebruik de volgende transformatie:

$$T(x, y) = (xy, x), \quad (37)$$

met Jacobiaan (met componentsgewijze differentiatie):

$$\begin{aligned}
\det(J) &= \det \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial y} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial x} \end{pmatrix} \\
&= \frac{\partial x}{\partial u} \cdot \frac{\partial y}{\partial x} - \frac{\partial y}{\partial u} \cdot \frac{\partial x}{\partial x} \\
&= 0 \cdot \frac{1}{y} - \frac{1}{x} \cdot 1 \\
&= -\frac{1}{x}.
\end{aligned} \tag{38}$$

Dit geeft volgens [1]:

$$\begin{aligned}
P\left(Y < \frac{u}{x}\right) &= \int_{-\infty}^{\infty} \int_{-\infty}^u f_{X,Y}\left(x, \frac{w}{x}\right) |\det(J)| dw dx \\
P\left(Y < \frac{u}{x}\right) &= \int_{-\infty}^{\infty} \int_{-\infty}^u f_{X,Y}\left(x, \frac{w}{x}\right) \left|\frac{1}{x}\right| dw dx.
\end{aligned} \tag{39}$$

Na het verwisselen van de integratievolgorde geeft de binnenste integraal de kansdichtheidsfunctie:

$$P(U < u) = \int_{-\infty}^u \int_{-\infty}^{\infty} f_{X,Y}\left(x, \frac{u}{x}\right) \frac{1}{|x|} dx dw. \tag{40}$$

Voor het geval waarin X, Y onderling onafhankelijk zijn, merk op dat dan geldt dat de kansdichtheidsfunctie voor X, Y in (39) vervangen mag worden voor:

$$f_{X,Y}(x, y) = f_X(x) f_Y(y). \tag{41}$$

■

Voor het stochastisch lineair programma (2) geldt dat, zoals beschreven in 1.2.3, de stochastiek verplaatst kan worden van matrix $A(\omega)$ naar vector $b(\omega)$ en viceversa. Als één van deze variabelen deterministisch is, zullen $A_B^{-1}(\omega)$ en $b(\omega)$ altijd onderling onafhankelijk zijn. Formule (35) kan dus gebruikt worden om de kansdichtheidsfunctie van x_B te bepalen.

In paragraaf 2.3 zal een benadermethode voor de kansdichtheidsfunctie van de oplossing en doelfunctie uitgewerkt worden.

Meer over het bepalen van de product kansdichtheidsfunctie kan gevonden worden in [46] en [35].

Om de kans op optimaliteit in (32) te berekenen, is de kansdichtheidsfunctie $f_{c(\omega)}(c_B, c_N)$ nodig. Deze is niet te bepalen als een product van kansdichtheidsfuncties, tenzij voor elk element van $c(\omega)$ geldt dat ze onderling onafhankelijk zijn. Ook kan het voorkomen dat de kansdichtheidsfuncties van $A(\omega)$ of $b(\omega)$ niet bekend zijn, of de kansdichtheidsfunctie van $A^{-1}(\omega)$ niet te bepalen is uit die van $A(\omega)$, al zijn er verschillende methoden om de kansdichtheidsfunctie te kunnen schatten.

In [2] wordt een methode beschreven om kansdichtheidsfuncties te schatten, die simpeler en sneller is dan de oudere methodes, omdat er geen enkele aanname wordt gedaan over de kansdichtheidsfunctie. Methoden om kansdichtheidsfuncties te schatten vallen uiteen in parametrische en niet-parametrische. Parametrische schattingsmethoden voor de kansdichtheidsfunctie hebben als nadeel dat er informatie nodig is over de kansverdeling. Niet-parametrische schattingsmethoden zijn veelal langzaam als ze geïmplementeerd worden.

2.2 Een exact betrouwbaarheidsinterval voor de oplossing en doelfunctie

In deze paragraaf zal de methode beschreven in [15] behandeld worden om een exact betrouwbaarheidsinterval te bepalen voor de toegelaten oplossingen van een stochastisch lineair programma en de doelfunctiewaarde. Meer over dit onderwerp is te vinden in [14] en [25].

σ^2	Correlatie	Oplossingsvector x	Doelfunctie $c^T(\omega)x$
Onbekend	Ja	$\delta^T \delta = S^2 k F_\alpha(m, \phi) V$	$z^2 < \sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) S^2 F_{\frac{1}{2}\alpha}(1, \phi)$
(2.2.1)	Nee	$\sum_{i=1}^m \left(\frac{\sum_{j=1}^n (a_{ij} x_j)^2}{\sum_{j=1}^n v(a_{ij}) x_j^2} \right) = S^2 m F_\alpha(m, \phi)$	$z^2 < \left(\sum_{j=1}^n \text{var}(c_j) x_j^2 \right) S^2 F_{\frac{1}{2}\alpha}(1, \phi)$
	Nee, gelijke variantie	$(1 + x_1^2 + x_2^2 + \dots + x_n^2) v(a) S^2 m F_\alpha(m, \phi) \leq 0$	$z^2 < \left(\sum_{j=1}^n x_j^2 \right) \text{var}(c) S^2 F_{\frac{1}{2}\alpha}(1, \phi)$
Bekend	Ja	$\delta^T \delta = \chi_\alpha^2(m) \sigma^2 V$	$z^2 < \sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) \chi_{\frac{1}{2}\alpha}^2(1) \sigma^2$
(2.2.2)	Nee	$\sum_{i=1}^m \left(\frac{\sum_{j=1}^n (a_{ij} x_j)^2}{\sum_{j=1}^n v(a_{ij}) x_j^2} \right) = \chi_\alpha^2(m) \sigma^2$	$z^2 < \left(\sum_{j=1}^n \text{var}(c_j) x_j^2 \right) \chi_{\frac{1}{2}\alpha}^2(1) \sigma^2$
	Nee, gelijke variantie	$(1 + x_1^2 + x_2^2 + \dots + x_n^2) v(a) \chi_\alpha^2(m) \sigma^2 \leq 0$	$z^2 < \left(\sum_{j=1}^n x_j^2 \right) \text{var}(c) \chi_{\frac{1}{2}\alpha}^2(1) \sigma^2$

Tabel 3: Formules voor grenzen $(1 - \alpha)\%$ -betrouwbaarheidsinterval.

Er wordt gebruik gemaakt van het volgende systeem, waarbij het niet noodzakelijk is dat $m = n$:

$$\begin{array}{ccccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & - & b_1 & = & \delta_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & - & b_2 & = & \delta_2 \\
\vdots & & \vdots & & & & \vdots & & \vdots & & \\
a_{i1}x_1 & + & a_{i2}x_2 & + & a_{ij}x_j & + & a_{in}x_n & - & b_i & = & \delta_i \\
\vdots & & \vdots & & & & \vdots & & \vdots & & \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & - & b_m & = & \delta_m
\end{array} \quad , \quad (42)$$

waarbij $i = 1, \dots, m$ en $j = 1, \dots, n$ en geldt:

$$A(\omega)x - b(\omega) = \delta(\omega) \quad (43)$$

Dit systeem is equivalent aan het stochastisch lineair programma (2) als geldt $\delta(\omega) \leq 0$, waarbij de stochastiek naar A is verplaatst zoals beschreven in 1.2.3. Gebruik:

$$\begin{aligned}
-b_i &= a_{i0} \\
x_0 &= 1 \\
c_0 &= 0
\end{aligned} \quad (44)$$

Er wordt aangenomen dat alle a_{ij} , $i = 1, \dots, m$, $j = 0, \dots, n$ multivariaat normaal verdeeld zijn met covariantiematrix $\Omega\sigma^2$. De covariantiematrix is bekend en onafhankelijk van de factor σ^2 .

Er geldt nu dat:

$$x \text{ is een toegelaten oplossing} \Leftrightarrow \delta(\omega) \leq 0 \quad \text{Voor een zekere } \omega \in \Omega \quad (45)$$

In dit hoofdstuk zal onderscheid gemaakt worden tussen het geval waarin σ^2 bekend is en of de elementen in de matrix $A(\omega)$ en $c(\omega)$ gecorreleerd zijn. De formules voor de grenzen van een $(1 - \alpha)\%$ -betrouwbaarheidsinterval voor de toegelaten oplossing en doelfunctiewaarde staan samengevat in onderstaande tabel. De afleiding is te vinden in de bijbehorende sectie. Niet voor elk geval is het mogelijk een expliciete formule op te stellen.

2.2.1 Onbekende variantie

Betrouwbaarheidsinterval voor oplossingsvector x

Als σ^2 niet bekend is, moet de covariantiematrix $\Sigma\sigma^2$ geschat worden. Neem aan dat voor σ^2 geldt dat er een schatter S^2 – bekend is. Neem aan dat $S^2 \frac{\chi^2(\phi)}{\phi}$ verdeeld is met ϕ vrijheidsgraden. De notatie $\chi^2(\phi)$ betekent dat de variable chi-kwadraat verdeeld is met ϕ vrijheidsgraden.

De covariantiematrix $\Sigma\sigma^2$ is bekend. Er geldt dat:

$$\delta_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + a_{i0} \quad \forall i = 1, \dots, m. \quad (46)$$

De matrix $\Sigma\sigma^2 \in \mathbb{R}^{m(n+1) \times m(n+1)}$ ziet er als volgt uit:

$$\Sigma\sigma^2 = \begin{pmatrix} \text{cov}(a_{10}, a_{10}) & \text{cov}(a_{10}, a_{11}) & \cdots & \text{cov}(a_{10}, a_{mn}) \\ \text{cov}(a_{11}, a_{10}) & \text{cov}(a_{11}, a_{11}) & \cdots & \text{cov}(a_{11}, a_{mn}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(a_{mn}, a_{10}) & \text{cov}(a_{mn}, a_{11}) & \cdots & \text{cov}(a_{mn}, a_{mn}) \end{pmatrix}. \quad (47)$$

Met behulp van de covariantiematrix $\Sigma\sigma^2$ kan de covariantiematrix voor $\delta(\omega)$ berekend worden. De covariantiematrix $V\sigma^2 \in \mathbb{R}^{m \times m}$ van $\delta(\omega)$ ziet er dan als volgt uit:

$$V\sigma^2 = \begin{pmatrix} \text{cov}(\delta_1, \delta_1) & \text{cov}(\delta_1, \delta_2) & \cdots & \text{cov}(\delta_1, \delta_m) \\ \text{cov}(\delta_2, \delta_1) & \text{cov}(\delta_2, \delta_2) & \cdots & \text{cov}(\delta_2, \delta_m) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(\delta_m, \delta_1) & \text{cov}(\delta_m, \delta_2) & \cdots & \text{cov}(\delta_m, \delta_m) \end{pmatrix}. \quad (48)$$

Voor elke a_{ij} , $i = 1, \dots, m$, $j = 0, \dots, n$ geldt dat deze multivariaat normaal verdeeld is. Voor δ_i , $i = 1, \dots, m$ geldt ook dat deze multivariaat normaal verdeeld zijn, omdat deze een lineaire combinatie van multivariaat normaal verdeelde variabelen zijn. Hieruit volgt dat:

$$\begin{aligned} S^2 & \sim \frac{\chi^2(\phi)}{\phi}, \\ \delta^T(\omega) V^{-1} \delta(\omega) & \sim \chi^2(m), \end{aligned} \quad (49)$$

dus kan afgeleid worden dat:

$$\frac{\frac{\delta^T(\omega) V^{-1} \delta(\omega)}{m}}{\frac{S^2 \cdot \phi}{\phi}} = \frac{\delta^T(\omega) V^{-1} \delta(\omega)}{m S^2} \sim F(m, \phi), \quad (50)$$

waarbij $F(m, \phi)$ betekent dat $\frac{\delta^T(\omega) V^{-1} \delta(\omega)}{m S^2}$ een F-verdeling heeft met m en ϕ vrijheidsgraden.

Er geldt nu dat:

$$\frac{\delta^T(\omega) V^{-1} \delta(\omega)}{m S^2} < F_\alpha(m, \phi), \quad (51)$$

met kans $(1 - \alpha)$, waarbij $F_\alpha(m, \phi)$ het punt is met een kans van α in de $F(m, \phi)$ -verdeling.

Omdat $\frac{\delta^T(\omega) V^{-1} \delta(\omega)}{m S^2}$ een functie is van $A(\omega)$, $b(\omega)$, x en S^2 , definieert deze functie het gebied waarvoor geldt dat x voor $(1 - \alpha)\%$ van de realisaties ω een toegelaten oplossing is, als x in dit gebied ligt.

De grenzen van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval voor x kunnen als volgt geschreven worden:

$$\delta^T(\omega) V^{-1} \delta(\omega) = S^2 m F_\alpha(m, \phi). \quad (52)$$

Dit kan ook geschreven worden als:

$$\det \begin{pmatrix} S^2 m F_\alpha & \delta^T(\omega) \\ \delta(\omega) & V \end{pmatrix} = S^2 m F_\alpha \cdot V - \delta^T(\omega) \cdot \delta(\omega) = 0, \quad (53)$$

$$\delta^T(\omega) \delta(\omega) = S^2 m F_\alpha \cdot V. \quad (54)$$

De grenzen van het betrouwbaarheidsinterval worden gedefinieerd door precies die waarde voor x_j , $j = 0, \dots, n$ die ervoor zorgen dat formule (54) geldt. Deze formule zal nu verder uitgewerkt worden om tot een expliciete uitdrukking te komen.

Ongecorrleerde elementen

In het speciale geval dat alle elementen a_{ij} , $i = 1, \dots, m$, $j = 0, \dots, n$ ongecorrleerd zijn, kan er een meer expliciete vergelijking opgesteld worden voor het de grenzen van het betrouwbaarheidsinterval.

Als alle a_{ij} , $i = 1, \dots, m$, $j = 0, \dots, n$ ongecorreleerd zijn, is $V\sigma^2$ een diagonaalmatrix met op de diagonaal de varianties van elke δ_i , $i = 1, \dots, m$. Immers, $\text{cov}(a_{ij}, a_{ij}) = 0$ als $(i, j) \neq (i, j)$. Noem het element dat op de diagonaal van V staat $v(\delta_i)$. Noem het diagonaalelement van Σ : $v(a_{ij})$, $i = 1, \dots, m$, $j = 0, \dots, n$.

De formule voor de grenzen van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval voor x wordt dan:

$$\begin{aligned}\delta^T(\omega) V^{-1} \delta(\omega) &= \sum_{i=1}^m \frac{\delta_i^2}{v(\delta_i)} = S^2 m F_\alpha \\ &= \sum_{i=1}^m \left(\frac{\sum_{j=0}^n (a_{ij} x_j)^2}{\sum_{j=0}^n v(a_{ij}) x_j^2} \right) = S^2 m F_\alpha.\end{aligned}\quad (55)$$

Ongecorreleerde elementen met gelijke variantie

Indien voor alle a_{ij} , $i = 1, \dots, m$, $j = 0, \dots, n$ geldt dat de variantie gelijk is – noem deze variantie $v(a)$ – dan volgt de volgende formule:

$$\sum \delta_i^2 = \sum_{i=1}^m \left(\sum_{j=0}^n a_{ij} x_j \right)^2 = \sum_{j=0}^n x_j^2 v(a) s^2 m F_\alpha(m, \phi). \quad (56)$$

Er geldt dat x een toegelaten oplossing is wanneer $\delta(\omega) \leq 0$. Dus:

$$\sum_{j=1}^n x_j^2 v(a) s^2 m F_\alpha(m, \phi) \leq 0 \quad (57)$$

Dit verder uitwerken geeft:

$$(1 + x_1^2 + x_2^2 + \dots + x_n^2) v(a) s^2 m F_\alpha(m, \phi) \leq 0, \quad (58)$$

Dus voor elke oplossing X die binnen dit gebied valt, geldt dat deze in $(1 - \alpha)\%$ van de realisaties ω toegelaten is.

Betrouwbaarheidsinterval voor de doelfunctiewaarde

Naast het opstellen van een betrouwbaarheidsinterval voor de oplossingsvector x zoals gedaan in [15] kan op gelijke wijze een betrouwbaarheidsinterval opgesteld worden voor de doelfunctiewaarde $z = c^T(\omega)x$.

Gebruik dat:

$$z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \quad (59)$$

en veronderstel dat alle c_j , $j = 1, \dots, n$ multivariaat normaal verdeeld zijn met covariantiematrix $\Lambda\sigma^2$, met S^2 als schatter voor σ^2 .

Het enige wat in deze situatie anders is dan in het geval van de oplossingsvector x , is dat er nu slechts één waarde z is in plaats van m waarden $\delta(\omega)$. Dit betekent dat de covariantie-matrix van z enkel de variantie van z is. Dus er geldt dat:

$$\text{var}(z) = \sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) = v\sigma^2. \quad (60)$$

Hieruit volgt:

$$\frac{z^T v^{-1} z}{S^2} \sim F_{\frac{1}{2}\alpha}(1, \phi). \quad (61)$$

De grens van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval voor de doelfunctiewaarde $z = c^T(\omega)x$ wordt dus als volgt gedefinieerd voor een zekere x_j , $j = 1, \dots, n$:

$$z^2 < \sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) S^2 k F_{\frac{1}{2}\alpha}(1, \phi), \quad (62)$$

dit geeft:

$$-\sqrt{\sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) S^2 k F_{\frac{1}{2}\alpha}(1, \phi)} < z < \sqrt{\sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) S^2 k F_{\frac{1}{2}\alpha}(1, \phi)}, \quad (63)$$

dit betekent dat voor $(1 - \alpha)\%$ van de realisaties ω geldt dat de doelfunctiewaarde z binnen deze grenzen zal vallen voor een zekere x . Dit zegt nog niks over de optimaliteit van x of de optimale waarde van van het stochastisch lineair programma.

De oplossingsvector x die een toegelaten oplossing is voor een zekere realisatie met het laagste betrouwbaarheidsinterval voor de doelfunctiewaarde is in die realisatie optimaal in $(1 - \alpha)\%$ van de gevallen.

Deze formule wordt afgeleid op dezelfde manier als in (51), (52) en (53).

Ongecorrleerde elementen

In het geval dat alle c_j , $j = 1, \dots, n$ ongecorrleerd zijn, verandert de formule voor de variantie van z :

$$\text{var}(z) = \sum_{j=1}^n \text{var}(c_j) x_j^2. \quad (64)$$

De formule voor de grens van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval van de doelfunctiewaarde wordt dan:

$$z^2 < \left(\sum_{j=1}^n \text{var}(c_j) x_j^2 \right) S^2 F_{\frac{1}{2}\alpha}(1, \phi), \quad (65)$$

dit geeft:

$$-\sqrt{\left(\sum_{j=1}^n \text{var}(c_j) x_j^2 \right) S^2 F_{\frac{1}{2}\alpha}(1, \phi)} < z < \sqrt{\left(\sum_{j=1}^n \text{var}(c_j) x_j^2 \right) S^2 F_{\frac{1}{2}\alpha}(1, \phi)}. \quad (66)$$

Ongecorrleerde elementen met gelijke variantie

In het geval dat alle varianties van c_j , $j = 1, \dots, n$ gelijk zijn, wordt de formule voor de grens van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval voor de doelfunctiewaarde:

$$z^2 < \left(\sum_{j=1}^n x_j^2 \right) \text{var}(c) S^2 F_{\frac{1}{2}\alpha}(1, \phi), \quad (67)$$

dit geeft:

$$-\sqrt{\left(\sum_{j=1}^n x_j^2 \right) \text{var}(c) S^2 F_{\frac{1}{2}\alpha}(1, \phi)} < z < \sqrt{\left(\sum_{j=1}^n x_j^2 \right) \text{var}(c) S^2 F_{\frac{1}{2}\alpha}(1, \phi)}. \quad (68)$$

2.2.2 Bekende variantie

In deze sectie zullen de formules voor het $(1 - \alpha)\%$ -betrouwbaarheidsinterval van de oplossingsvector x en de doelfunctiewaarde $c^T(\omega)x$ worden afgeleid indien σ^2 bekend is.

Betrouwbaarheidsinterval voor oplossingsvector x

Er geldt dat alle a_{ij} , $i = 1, \dots, m$, $j = 0, \dots, n$ multivariaat normaal verdeeld zijn met covariantiematrix $\Sigma\sigma^2$ zoals in (47). Voor alle δ_i , $i = 1, \dots, m$ geldt dat deze een lineaire combinatie is van multivariaat normaal verdeelde stochastische variabelen en dus dat alle δ_i , $i = 1, \dots, m$ multivariaat normaal verdeeld zijn. Hieruit volgt dat:

$$\frac{\delta^T(\omega)\delta(\omega)}{V\sigma^2} \sim \chi^2(m). \quad (69)$$

Er geldt dan dat met kans $(1 - \alpha)$ de volgende formule waar is:

$$\frac{\delta^T(\omega)\delta(\omega)}{V\sigma^2} < \chi_\alpha^2(m). \quad (70)$$

De grenzen van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval voor de oplossingsvector x kunnen dan geschreven worden met behulp van de afleiding van (54) met formules (51)-(53):

$$\delta^T(\omega)\delta(\omega) = \chi_\alpha^2(m)\sigma^2 V, \quad (71)$$

$\forall x_j, j = 1, \dots, n$ en $x_0 = 1$ die ervoor zorgen dat bovenstaande formule geldt.

Voor de formules uit sectie 2.2.1 betekent dit dus dat $S^2 mF_\alpha(m, \phi)$ vervangen kan worden door $\chi_\alpha^2(m)\sigma^2$. Om die reden zijn de afleidingen van de formules in deze sectie zeer beknopt. De uitgebreide afleiding is in sectie 2.2.1 te vinden.

Ongecorrleerde elementen

In het geval dat alle $a_{ij}, i = 1, \dots, m, j = 0, \dots, n$ ongecorrleerd zijn, kan de volgende formule voor het $(1 - \alpha)\%$ -betrouwbaarheidsinterval van de oplossingsvector x afgeleid worden:

$$\begin{aligned} \delta^T(\omega) V^{-1} \delta(\omega) &= \sum_{i=1}^m \frac{\delta_i^2}{v(\delta_i)} = \chi_\alpha^2(m)\sigma^2 \\ &= \sum_{i=1}^m \left(\frac{\sum_{j=0}^n (a_{ij}x_j)^2}{\sum_{j=0}^n v(a_{ij})x_j^2} \right) = \chi_\alpha^2(m)\sigma^2, \end{aligned} \quad (72)$$

met $v(\cdot)$ de waarde op de diagonaal van de matrices V respectievelijk Ω .

Ongecorrleerde elementen met gelijke variantie

Als de elementen van de matrix $A(\omega)$ ongecorrleerd zijn en gelijke variantie hebben, dan geldt de volgende formule voor het $(1 - \alpha)\%$ -betrouwbaarheidsinterval van de oplossingsvector x :

$$\sum \delta_i^2 = \sum_{i=1}^m \left(\sum_{j=0}^n a_{ij}x_j \right)^2 = \sum_{j=0}^n x_j^2 v(a) \chi_\alpha^2(m)\sigma^2. \quad (73)$$

Er moet dus gelden dat:

$$\sum_{j=0}^n x_j^2 v(a) \chi_\alpha^2(m)\sigma^2 \leq 0, \quad (74)$$

dit is te schrijven als:

$$(1 + x_1^2 + x_2^2 + \dots + x_n^2) v(a) \chi_\alpha^2(m)\sigma^2 \leq 0, \quad (75)$$

Betrouwbaarheidsinterval voor de doelfunctie

Ook hier kan het werk van [15] uitgebreid worden voor de doelfunctie.

Gebruik dat:

$$z = c_1 x_1 + c_2 + x_2 + \dots + c_n x_n \quad (76)$$

en veronderstel dat alle $c_j, j = 1, \dots, n$ multivariaat normaal verdeeld zijn met covariantiematrix $\Lambda\sigma^2$. Wederom kan gebruikt worden dat de covariantiematrix van z slechts bestaat uit de variantie van z , zoals vermeld in (60). Dit geeft het volgende resultaat:

$$\frac{z^T v z}{\sigma^2} \sim \chi_{\frac{1}{2}\alpha}^2(1). \quad (77)$$

De grens van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval voor de doelfunctie $c^T(\omega)x$ wordt dus gedefinieerd door de waarde van $x_j, j = 1, \dots, n$ die ervoor zorgen dat de onderstaande gelijkheid geldt:

$$z^2 < \sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) \chi_{\frac{1}{2}\alpha}^2(1) \sigma^2, \quad (78)$$

dit geeft voor een zekere x het volgende $(1 - \alpha)\%$ -betrouwbaarheidsinterval voor de doelfunctiewaarde:

$$-\sqrt{\sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) \chi_{\frac{1}{2}\alpha}^2(1) \sigma^2} < z < \sqrt{\sum_{j=1}^n \sum_{\hat{j}=1}^n \text{cov}(c_j x_j, c_{\hat{j}} x_{\hat{j}}) \chi_{\frac{1}{2}\alpha}^2(1) \sigma^2}. \quad (79)$$

Ongecorrleerde elementen

In het geval dat alle $c_j, j = 1, \dots, n$ ongecorrleerd zijn, verandert de formule voor de variantie van z :

$$\text{var}(z) = \sum_{j=1}^n \text{var}(c_j) x_j^2. \quad (80)$$

De formule voor de grens van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval wordt dan:

$$z^2 < \left(\sum_{j=1}^n \text{var}(c_j) x_j^2 \right) \chi_{\frac{1}{2}\alpha}^2 (1) \sigma^2, \quad (81)$$

dit geeft:

$$-\sqrt{\left(\sum_{j=1}^n \text{var}(c_j) x_j^2 \right) \chi_{\frac{1}{2}\alpha}^2 (1) \sigma^2} < z < \sqrt{\left(\sum_{j=1}^n \text{var}(c_j) x_j^2 \right) \chi_{\frac{1}{2}\alpha}^2 (1) \sigma^2}. \quad (82)$$

Ongecorrelleerde elementen met gelijke variantie

In het geval dat alle varianties van c_j , $j = 1, \dots, n$ gelijk zijn, wordt de formule voor de grens van het $(1 - \alpha)\%$ -betrouwbaarheidsinterval:

$$z^2 < \left(\sum_{j=1}^n x_j^2 \right) \text{var}(c) \chi_{\frac{1}{2}\alpha}^2 (1) \sigma^2, \quad (83)$$

dit geeft:

$$-\sqrt{\left(\sum_{j=1}^n x_j^2 \right) \text{var}(c) \chi_{\frac{1}{2}\alpha}^2 (1) \sigma^2} < z < \sqrt{\left(\sum_{j=1}^n x_j^2 \right) \text{var}(c) \chi_{\frac{1}{2}\alpha}^2 (1) \sigma^2} \quad (84)$$

2.3 Benadermethode voor kansdichtheidsfunctie en betrouwbaarheidsinterval

In deze paragraaf zal de methode uit [5] om een benadering te maken van de kansdichtheidsfunctie van de oplossing en doelfunctie beschreven worden. In sommige gevallen is het ook mogelijk om met deze methode een exacte kansdichtheidsfunctie te krijgen. Meer informatie over deze methode is te vinden in [16], [27] en [36].

Allereerst zal de methode voor het benaderen dan wel bepalen van de kansdichtheidsfunctie van de oplossingsvector x worden behandeld, daarna zal deze methode worden toegepast op de doelfunctie. Het doel van deze benadering is om, naast het bepalen van de kansdichtheidsfunctie, een betrouwbaarheidsinterval voor de oplossingsvector x van het stochastisch lineair programma en voor de doelfunctie op te kunnen stellen.

In tabel 4 staan de in deze paragraaf afgeleide formules samengevat. De afleiding is te vinden in de bijbehorende secties.

	Oplossingsvector x (2.3.1)
Kansdichtheidsfunctie	$f_{x_k}(x_k) = \frac{1}{\sqrt{2\pi}} \frac{\beta \sigma_k^2 - \delta_k \sigma_{\hat{A}(\omega)k} + x_k (\delta_k \sigma_{\hat{A}(\omega)}^2 - \beta \sigma_{\hat{A}(\omega)k})}{(\sigma_k^2 - 2\sigma_{\hat{A}(\omega)k} x_k + \sigma_{\hat{A}(\omega)}^2)^{3/2}} \cdot e^{-1/2 [x_k \beta - \delta_k]^2 / [\sigma_N^2 - 2\sigma_{\hat{A}(\omega)k} x_k + \sigma_{\hat{A}(\omega)}^2 x_k^2]}$
Grenzen betrouwbaarheidsinterval	$x_k = \frac{-2\beta \delta_k + \gamma^2 \sigma_{\hat{A}(\omega)k} \pm \sqrt{(-2\beta \delta_k - \gamma^2 \sigma_{\hat{A}(\omega)k})^2 + 4(\beta^2 - \gamma^2 \sigma_{\hat{A}(\omega)}^2)(\delta_k^2 - \gamma^2 \sigma_k^2)}}{2(\delta_k^2 - \gamma^2 \sigma_k^2)}$
	Doelfunctie $z = c^T(\omega)x$ (2.3.2)
Kansdichtheidsfunctie	$f_z(z) = \frac{1}{\sqrt{2\pi}} \frac{\beta \sigma_{N(z)}^2 - \delta_z \sigma_{\hat{A}(\omega)N(z)} + y (\delta_k \sigma_{\hat{A}(\omega)}^2 - \beta \sigma_{\hat{A}(\omega)N(z)})}{(\sigma_{N(z)}^2 - 2\sigma_{\hat{A}(\omega)N(z)} y + \sigma_{\hat{A}(\omega)}^2)^{3/2}} \cdot e^{-1/2 [z \beta - \delta_z]^2 / [\sigma_N^2 - 2\sigma_{\hat{A}(\omega)N(z)} z + \sigma_{\hat{A}(\omega)}^2 z^2]}$
Grenzen betrouwbaarheidsinterval	$z = \frac{-2\beta \delta_z + \gamma^2 \sigma_{\hat{A}(\omega)N(z)} \pm \sqrt{(-2\beta \delta_z - \gamma^2 \sigma_{\hat{A}(\omega)N(z)})^2 + 4(\beta^2 - \gamma^2 \sigma_D^2)(\delta_z^2 - \gamma^2 \sigma_{N(z)}^2)}}{2(\delta_z^2 - \gamma^2 \sigma_{N(z)}^2)}$

Tabel 4: Formules kansdichtheidsfunctie en betrouwbaarheidsinterval.

2.3.1 Oplossingsvector x

Om de kansverdeling van de oplossingsvector x te bepalen wordt het volgende systeem gebruikt:

$$(A + \hat{A}(\omega))x = (b + \hat{b}(\omega)), \quad (85)$$

waarbij $A \in \mathbb{R}^{m \times m}$ een niet-singuliere matrix is met deterministische constanten, $\hat{A}(\omega) : \Omega \rightarrow \mathbb{R}^{m \times m}$ een stochastische matrix, $b \in \mathbb{R}^m$ een vector met deterministische constanten en $\hat{b}(\omega) : \Omega \rightarrow \mathbb{R}^m$ een stochastische

vector. Voor deze methode is het dus van belang dat matrix $A(\omega)$ uit (2) inverteerbaar is.

Benadering kansverdeling x

Veronderstel dat de stochastische matrix $\hat{A}(\omega)$ een bekende kansverdeling heeft die er als volgt uitziet voor elk element van $\hat{A}(\omega)$:

$$\begin{aligned} E(\hat{a}_{ij}) &= 0, \\ E(\hat{a}_{ij}^2) &= \sigma_{ij}^2, \\ \text{var}(\hat{a}_{ij}) &= E(\hat{a}_{ij}^2) + (E(\hat{a}_{ij}))^2 \\ &= \sigma_{ij}^2 + 0^2 \\ &= \sigma_{ij}^2, \end{aligned} \tag{86}$$

voor $i, j = 1, 2, \dots, m$. Ook hier wordt aangenomen dat de matrix $(A + \hat{A}(\omega))$ niet-singulier is.

Op een gelijke manier is de kansverdeling van $\hat{b}(\omega)$ bepaald:

$$\begin{aligned} E(\hat{b}_i) &= 0, \\ E(\hat{b}_i^2) &= \tau_i^2, \\ \text{var}(\hat{b}_i) &= E(\hat{b}_i^2) + E(\hat{b}_i)^2 \\ &= \tau_i^2 + 0^2 \\ &= \tau_i^2, \end{aligned} \tag{87}$$

voor $i = 1, 2, \dots, m$.

De oplossingsvector x kan bepaald worden met Cramer's regel⁴, dus de oplossingsvector x wordt:

$$x_k = \frac{\det(D^k + d^k)}{\det(\hat{B} + \hat{b})}, \tag{88}$$

waarbij $(D^k + d^k)$ de matrix is waarin de k^e kolom van $(A + \hat{A}(\omega))$ is vervangen door de kolomvector $(b + \hat{b}(\omega))$

Omdat $\hat{A}(\omega)$ en $\hat{b}(\omega)$ stochastisch zijn, is een methode nodig om de determinanten te benaderen. Hiervoor wordt de volgende benadering gebruikt:

$$\begin{aligned} \det(D^k + d^k) &\approx \det(D^k) + \sum_{i=1}^m \beta_{ik} \hat{b}_i + \sum_{i=1}^m \sum_{j=1, j \neq k}^m D_{ij}^k \hat{a}_{ij} = N(x_k), \\ \det(A + \hat{A}(\omega)) &\approx \det(A) + \sum_{i=1}^m \sum_{j=1}^m \beta_{ij}^k \hat{a}_{ij} = D(x), \end{aligned} \tag{89}$$

waarbij β_{ij} de cofactor⁵ van element $\hat{a}_{ij}$ is in het uitrekenen van de determinant van A .

Deze benadering is als volgt afgeleid:

$$\begin{aligned} \det(A + \hat{A}(\omega)) &= \sum_{j=1}^m (-1)^{i+j} M_{ij} (a_{ij} + \hat{a}_{ij}) && \text{Ontwikkeling langs rij } i \\ &= \sum_{j=1}^m \beta_{ij} a_{ij} + \sum_{j=1}^m (-1)^{i+j} M_{ij} \hat{a}_{ij} \\ &= \det(A) + \sum_{j=1}^m \left(\sum_{i=1}^m M_{ij} \right) \hat{a}_{ij} \\ &\cong \det(A) + \sum_{j=1}^m \sum_{i=1}^m \beta_{ij} \hat{a}_{ij}, && \text{Gebruik } \beta_{ij} \text{ als benadering voor } (-a)^{i+j} M_{ij} \end{aligned}$$

$$\begin{aligned} \det(D^k + d^k) &= \sum_{i=1}^m (-1)^{i+k} M_{ik} (D^k + d^k) && \text{Ontwikkeling langs rij } i \\ &= \sum_{i=1}^m \beta_{ik} D^k + \sum_{i=1}^m (-1)^{i+k} M_{ik} d^k \\ &= \det(D^k) + \sum_{i=1}^m \left(\sum_{k=1}^m M_{ik} \right) d^k \\ &\cong \det(D^k) + \sum_{i=1}^m \beta_{ik} \hat{b}_i + \sum_{j=1, j \neq k}^m D_{ij}^k \hat{a}_{ij}, && \text{Gebruik } \beta_{ik} \text{ als benadering voor } (-a)^{i+k} M_{ik} \text{ in} \\ &&& \text{kolom } k \text{ en dat behalve in kolom geldt } k \ d^k = \hat{A}(\omega) \end{aligned}$$

⁴Voor $Ax = \hat{b}$ geldt dat $\forall k \ x_k = \frac{\det(M_k)}{\det(A)}$ als $\det(A) \neq 0$. M_k is de matrix waarin kolom k van matrix A vervangen is door vector $\hat{B}$

⁵Cofactor: $\beta_{ij} = (-1)^{i+j} \cdot M_{ij}$

Deze benadering is exact als:

1. $\hat{A}(\omega) = 0$ en de elementen van $\hat{b}(\omega)$ ongecorreleerd zijn.
2. Alle elementen in $\hat{A}(\omega)$ nul zijn behalve die in de k de kolom, dan is de benadering van x_k exact.

Het is mogelijk om de stochastiek van $A(\omega)$ naar $b(\omega)$ te verplaatsen zoals beschreven in 1.2.3 en als alle elementen van $\hat{b}(\omega)$ ongecorreleerd zijn, is deze methode dus exact. In alle andere gevallen geeft deze methode dus een benadering van de verdeling van de oplossingsvector x .

Het is nu zaak de verdeling te bepalen van de benadering voor x_k :

$$x_k = \frac{N(x_k)}{D(x)} \quad (90)$$

De verdeling van x_k volgt uit:

$$\begin{aligned} E(N(x_k)) &= E \left[\det(D^k) + \sum_{i=1}^m \beta_{ik} \hat{b}_i + \sum_{i=1}^m \sum_{j=1, j \neq k}^m D_{ij}^k \hat{a}_{ij} \right] \\ &= \det(D^k) \\ &= \delta_k, \\ \text{var}(N(x_k)) &= \text{var} \left[\det(D^k) + \sum_{i=1}^m \beta_{ik} \hat{b}_i + \sum_{i=1}^m \sum_{j=1, j \neq k}^m D_{ij}^k \hat{a}_{ij} \right] \\ &= \sum_{i=1}^m (\beta_{ik})^2 \tau_i^2 + \sum_{i=1, j \neq k}^m (D_{ij}^k)^2 \sigma_{ij}^2 \\ &= \sigma_k^2, \\ E(D(x)) &= E \left[\det(A) + \sum_{i=1}^m \sum_{j=1}^m \beta_{ij}^k \hat{a}_{ij} \right] \\ &= \det(\hat{A}) \\ &= \beta, \\ \text{var}(D(x)) &= \text{var} \left[\det(A) + \sum_{i=1}^m \sum_{j=1}^m \beta_{ij}^k \hat{a}_{ij} \right] \\ &= \sum_{i=1}^m \sum_{j=1}^m (\beta_{ij}^k)^2 \sigma_{ij}^2 \\ &= \sigma_{\hat{A}^2}, \\ \text{cov}(\det(D^k + d^k), \det(A + \hat{A}(\omega))) &\cong \sum_{i=1}^m \sum_{j=1, j \neq k}^m D_{ij}^k \beta_{ij} \sigma_{ij}^2 \\ &= \sigma_{\hat{A}(\omega)} k. \end{aligned}$$

Indien de kansverdelingen van de vectoren $\hat{A}(\omega)$ en $\hat{b}(\omega)$ bekend zijn, kan de kansverdeling van het quotiënt $\frac{N(x_k)}{D(x)}$ bepaald worden. In [5] wordt gebruikt dat $\hat{A}(\omega)$ en $\hat{b}(\omega)$ normaal verdeeld zijn, dan zijn $N(x_k)$ en $D(x)$ lineaire functies van normaal verdeelde variabelen, dus zijn ze zelf ook normaal verdeeld.

Om de kansverdeling van het quotiënt te benaderen wordt de volgende stelling van [27] gebruikt:

Stelling 2.4

Als N en D normaal verdeelde variabelen zijn met verwachtingswaarde $E(N)$, $E(D)$, varianties σ_N^2 , σ_D^2 en covariantie σ_{ND} dan is de expressie:

$$\frac{E(D)Z - E(N)}{\sqrt{(\sigma_N^2 - 2\sigma_{ND}Z + \sigma_D^2 Z^2)}} \quad (91)$$

bij benadering standaard normaal verdeeld voor het quotiënt $Z = \frac{N}{D}$ als $E(D) > 3\sigma_D$.

Het bewijs van deze stelling is te vinden in [27].

Kansdichtheidsfunctie

Met behulp van (91) kan nu de kansdichtheidsfunctie voor x_k in (90) bepaald worden.

De kansdichtheidsfunctie van Z volgt uit:

$$f_Z(Z) = \frac{1}{\sqrt{2\pi}} \frac{E(D)\sigma_N^2 - E(N)\sigma_{ND} + Z(E(N)\sigma_D^2 - E(D)\sigma_{ND})}{(\sigma_N^2 - 2\sigma_{ND}Z + \sigma_D^2)^{3/2}} \cdot e^{-1/2[Z(E(D)-E(N))]^2 / [\sigma_N^2 - 2\sigma_{ND}Z + \sigma_D^2]}. \quad (92)$$

Gebruik dat: $Z = x_k$, $E(D) = \beta$, $E(N) = \delta_k$, $\sigma_D^2 = \sigma_{\hat{A}(\omega)}^2$, $\sigma_{ND} = \sigma_{\hat{A}(\omega)k}$ en $\sigma_N^2 = \sigma_k^2$ dan:

$$f_{x_k}(x_k) = \frac{1}{\sqrt{2\pi}} \frac{\beta\sigma_k^2 - \delta_k\sigma_{\hat{A}(\omega)k} + x_k(\delta_k\sigma_{\hat{A}(\omega)}^2 - \beta\sigma_{\hat{A}(\omega)k})}{(\sigma_k^2 - 2\sigma_{\hat{A}(\omega)k}x_k + \sigma_{\hat{A}(\omega)}^2)^{3/2}} \cdot e^{-1/2[x_k\beta - \delta_k]^2 / [\sigma_k^2 - 2\sigma_{\hat{A}(\omega)k}x_k + \sigma_{\hat{A}(\omega)}^2]}. \quad (93)$$

Met behulp van deze kansdichtheidsfunctie kan nu voor een vector x_B voor elke component $k = 1, \dots, m$ de kans bepaald worden dat deze toegelaten is met behulp van (24).

Benadering betrouwbaarheidsinterval x

Het betrouwbaarheidsinterval kan bepaald worden uit:

$$P \left[\left| \frac{E(D)Z - E(N)}{\sqrt{(\sigma_N^2 - 2\sigma_{ND}Z + \sigma_D^2)}} \right| \leq \gamma \right] = \alpha, \quad (94)$$

hierbij kan $\gamma = N_\alpha(0, 1)$ op gezocht worden in een tabel voor de standaard normale verdeling, waarbij γ de waarde is waarvoor de vergelijking geldt.

Om het betrouwbaarheidsinterval van x_k te bepalen, moet de volgende vergelijking worden opgelost:

$$(E(D)^2 - \gamma^2\sigma_D^2)Z^2 - 2(E(D) \cdot E(N) - \gamma^2\sigma_{ND})Z - (E(N)^2 - \gamma^2\sigma_N^2) = 0, \quad (95)$$

waarbij $E(D) = \beta$, $E(N) = \delta_k$, $\sigma_D^2 = \sigma_{\hat{A}(\omega)}^2$, $\sigma_{ND} = \sigma_{\hat{A}(\omega)k}$ en $\sigma_N^2 = \sigma_k^2$. De oplossing van de volgende vergelijking geeft dus een benadering voor het betrouwbaarheidsinterval voor toegelaten x_k in $(1 - \alpha)\%$ van de realisaties:

$$x_k = \frac{-2\beta\delta_k + \gamma^2\sigma_{\hat{A}(\omega)k} \pm \sqrt{(-2\beta\delta_k - \gamma^2\sigma_{\hat{A}(\omega)k})^2 + 4(\beta^2 - \gamma^2\sigma_{\hat{A}(\omega)}^2)(\delta_k^2 - \gamma^2\sigma_k^2)}}{2(\delta_k^2 - \gamma^2\sigma_k^2)}. \quad (96)$$

2.3.2 Doelfunctie

De afleiding van de benadering voor de kansverdeling van de doelfunctie $c^T(\omega)x$ en het bijbehorende betrouwbaarheidsinterval is vrijwel gelijk aan die van de oplossingsvector x . De volgende notatie wordt gebruikt:

$$(c + \hat{c}(\omega))^T x \quad (97)$$

met $c \in \mathbb{R}^m$ een deterministische vector en $\hat{c}(\omega) : \Omega \rightarrow \mathbb{R}^m$ een stochastische vector.

Benadering kansverdeling doelfunctie

Er wordt gebruik gemaakt van de volgende notatie:

$$\begin{aligned} z &= \sum_{r=1}^m (c_r + \hat{c}_r)x_r \\ &= \frac{1}{\det(A + \hat{A}(\omega))} \sum_{r=1}^m (c_r + \hat{c}_r) \cdot \det(D^r + d^r), \end{aligned} \quad (98)$$

waarbij $E(\hat{c}_i) = 0$ en $E(\hat{c}_i^2) = v_r^2$.

Wederom moeten de determinanten benaderd worden en van de benadering de verwachtingswaarde en variantie bepaald worden, dit gaat als volgt:

$$\begin{aligned}
N(z) &\cong \sum_{r=1}^m (c_r + \hat{c}_r) \cdot \det(D^r + d^r) \\
&= \sum_{r=1}^m c_r \cdot \det(D^r) + \sum_{r=1}^m c_r \left(\sum_{i=1}^m \beta_{ir} \hat{b}_i \right) + \sum_{r=1}^m c_r \sum_{i=1}^m \sum_{j=1, j \neq r}^m D_{ij}^r \hat{a}_{ij} + \sum_{r=1}^m \det(D^r) \hat{c}_r, \\
E(N(z)) &= \sum_{r=1}^m c_r \cdot \det(D^r) \\
&= \delta_y, \\
\text{var}(N(z)) &\cong \sum_{r=1}^m \left(\sum_{i=1}^m c_i \beta_{ri} \right)^2 \tau_r^2 + \sum_{r=1}^m c_r^2 \sum_{i=1}^m \sum_{j=1, j \neq r}^m (D_{ij}^r)^2 \sigma_{ij}^2 + \sum_{r=1}^m \det(D^r)^2 v_r^2 \\
&= \sigma_{N(z)}^2, \\
\text{cov}(N(z), D(z)) &\cong \sum_{r=1}^m c_r \sum_{i=1}^m \sum_{j=1, j \neq r}^m D_{ij}^r \beta_{ij} \sigma_{ij}^2 \\
&= \sigma_{\hat{A}(\omega)N(z)}, \\
E(D(z)) &= E(D(x)) \\
&= \beta, \\
\text{var}(D(z)) &= \text{var}(D(x)) \\
&= \sigma_{\hat{A}(\omega)}^2.
\end{aligned}$$

Het is nu zaak om de verdeling te benaderen van:

$$z = \frac{N(z)}{D(z)}. \quad (99)$$

Dit is equivalent aan de afleiding gegeven in de vorige paragraaf.

De kansdichtheidsfunctie wordt:

$$f_z(z) = \frac{1}{\sqrt{2\pi}} \frac{\beta \sigma_{N(z)}^2 - \delta_z \sigma_{\hat{A}(\omega)N(z)} + z \left(\delta_k \sigma_{\hat{A}(\omega)}^2 - \beta \sigma_{\hat{A}(\omega)N(z)} \right)}{\left(\sigma_{N(z)}^2 - 2 \sigma_{\hat{A}(\omega)N(z)} y + \sigma_{\hat{A}(\omega)}^2 \right)^{3/2}} \cdot e^{-1/2 [z\beta - \delta_z]^2 / [\sigma_N^2 - 2 \sigma_{\hat{A}(\omega)N(z)} y + \sigma_{\hat{A}(\omega)}^2 z^2]}. \quad (100)$$

Het betrouwbaarheidsinterval van de doelfunctiewaarde z voor een zekere x wordt dan:

$$y = \frac{-2\beta\delta_z + \gamma^2 \sigma_{\hat{A}(\omega)N(z)} \pm \sqrt{(-2\beta\delta_z - \gamma^2 \sigma_{\hat{A}(\omega)N(z)})^2 + 4(\beta^2 - \gamma^2 \sigma_D^2)(\delta_z^2 - \gamma^2 \sigma_{N(z)}^2)}}{2(\delta_z^2 z - \gamma^2 \sigma_{N(z)}^2)}, \quad (101)$$

De doelfunctiewaarde z zal dus in $(1 - \alpha)\%$ van de realisaties, voor een zekere x , tussen deze grenzen liggen.

2.4 Een betrouwbaarheidsinterval voor niet-multivariaat normaal verdeelde elementen

In de voorgaande secties werd steeds gebruikt gemaakt van het feit dat de elementen van $A(\omega)$, $b(\omega)$ en $c(\omega)$ normaal verdeeld waren. In deze sectie zal gekeken worden naar het opstellen van een betrouwbaarheidsinterval voor de oplossingsvector x en doelfunctiewaarde z , indien de elementen uit matrix $A(\omega)$ en vector $c(\omega)$ in (2) niet multivariaat normaal verdeeld zijn. Hiervoor wordt de centrale limietstelling gebruikt.

Stelling 2.5: de centrale limietstelling

Neem de rij onafhankelijke, stochastische variabelen $X_1, X_2, \dots, X_n$ met dezelfde kansverdeling, elk met gemiddelde μ en variantie σ^2 . De verdeling van: $\frac{X_1 + X_2 + \dots + X_n - n\mu}{\sigma\sqrt{n}}$ convergeert naar de standaard normale verdeling als $n \rightarrow \infty$. Dit betekent:

$$P\left(\frac{X_1 + X_2 + \dots + X_n - n\mu}{\sigma\sqrt{n}} \leq a\right) \rightarrow \frac{1}{\sqrt{2\pi}} \int_{-\infty}^a e^{-\frac{x^2}{2}} dx. \quad (102)$$

Dit geeft het volgende betrouwbaarheidsinterval:

$$\left| \frac{X_1 + X_2 + \dots + X_n - n\mu}{\sigma\sqrt{n}} \right| < N_\alpha(0, 1), \quad (103)$$

met kans $(1 - \alpha)$, waarbij $N_\alpha(0, 1)$ de grens is van het $(1 - \alpha)$ -interval in de $N(0, 1)$ -verdeling.

Dit betrouwbaarheidsinterval kan ook geschreven worden als:

$$\left| \frac{\frac{1}{n} (X_1 + X_2 + \dots + X_n) - \mu}{\frac{\sigma}{\sqrt{n}}} \right| < N_\alpha(0, 1). \quad (104)$$

Met behulp van de centrale limietstelling zal in deze paragraaf een betrouwbaarheidsinterval voor de oplossingsvectoren en doelfunctiewaarde opgesteld worden, indien er sprake is van niet-multivariaat normaal verdeelde elementen.

	Identieke kansverdeling	Niet-identieke kansverdeling
Onafhankelijk	Centrale limietstelling vergelijking (102) (2.4.1)	Lindebergs conditie vergelijking (116), (118) (2.4.2)
Afhankelijk	Mixing condition vergelijking (124), (126) (2.4.3)	Mixing condition (2.4.4)

Tabel 5: Theorieën en condities voor benaderen niet-multivariaat normaal verdeelde variabelen.

2.4.1 Onafhankelijke en identiek verdeelde elementen

Eerst zal gekeken worden hoe er een betrouwbaarheidsinterval opgesteld kan worden, als de elementen van $A(\omega)$, $b(\omega)$ en $c(\omega)$ per matrix/vector een identieke kansverdeling hebben en de elementen onafhankelijk zijn.

Een betrouwbaarheidsinterval voor toegelaten oplossingsvector x

Maak gebruik van paragraaf 1.2.4 en verplaats alle stochastiek naar de linkerkant. Dit geeft:

$$A(\omega)x - b(\omega) \leq \delta(\omega) \quad (105)$$

Een oplossing is toegelaten $\Leftrightarrow \delta(\omega) \leq 0$.

De kansverdeling van $\delta(\omega)$ is identiek voor iedere rij i , omdat alle elementen van $A(\omega)$ en van $b(\omega)$ dezelfde kansverdeling hebben. Noem deze verdeling $D_{\delta_i}(\mu_{\delta_i(\omega)}, \sigma_{\delta_i(\omega)}^2)$, $i = 1, \dots, m$. In het geval van identiek verdeelde elementen zal de verwachtingswaarde en variantie gelijk zijn voor elke i .

Als $m \rightarrow \infty$, dan geldt:

$$P\left(\frac{\delta_1 + \delta_2 + \dots + \delta_m - m\mu_{\delta(\omega)}}{\sigma_{\delta(\omega)}\sqrt{m}} \leq a\right) \rightarrow \frac{1}{\sqrt{2\pi}} \int_{-\infty}^a e^{-\frac{\delta_i^2}{2}} d\delta_i, \quad (106)$$

dit geeft het volgende betrouwbaarheidsinterval:

$$\left| \frac{\frac{1}{m} (\delta_1 + \delta_2 + \dots + \delta_m) - \mu_{\delta(\omega)}}{\frac{\sigma_{\delta(\omega)}}{\sqrt{m}}} \right| < N_\alpha(0, 1). \quad (107)$$

Dus als $m \rightarrow \infty$, dan zal in $(1 - \alpha)\%$ van de gevallen gelden dat $\frac{1}{m}\delta_i = \bar{\delta}$, $i = 1, \dots, m$ in (107) ligt.

Dit is te schrijven als:

$$\bar{\delta} < N_\alpha(0, 1) \frac{\sigma_{\delta(\omega)}}{\sqrt{m}} + \mu_{\delta(\omega)}. \quad (108)$$

Uit (105) volgt nu:

$$A(\omega)x - b(\omega) \leq \bar{\delta}(\omega) < N_\alpha(0, 1) \frac{\sigma_{\delta(\omega)}}{\sqrt{m}} + \mu_{\delta(\omega)}. \quad (109)$$

Er geldt dat x een toegelaten oplossing is dan en slechts dan als:

$$A(\omega)x - b(\omega) \leq \bar{\delta}(\omega) < N_\alpha(0, 1) \frac{\sigma_{\delta(\omega)}}{\sqrt{m}} + \mu_{\delta(\omega)} \leq 0. \quad (110)$$

Dus als geldt:

$$\bar{\delta}(\omega) < N_\alpha(0, 1) \frac{\sigma_{\delta(\omega)}}{\sqrt{m}} + \mu_{\delta(\omega)} \leq 0, \quad (111)$$

dan zal voor een zekere x gelden dat deze in $(1 - \alpha)\%$ van de realisaties ω een toegelaten oplossing is.

Een betrouwbaarheidsinterval voor de doelfunctie

Neem aan dat geldt: $c_j \sim D_{c(\omega)}(\mu_{c(\omega)}, \sigma_{c(\omega)}^2)$, $j = 1, \dots, n$ waarbij $D_{c(\omega)}$ een zekere verdeling is met verwachtingswaarde $\mu_{c(\omega)}$ en variantie $\sigma_{c(\omega)}^2$ en de elementen $c_j(\omega)$ onderling onafhankelijk zijn. Als $n' \rightarrow \infty$ dan vertelt de centrale limietstelling:

$$\text{Als } n' \rightarrow \infty, \text{ dan } P\left(\frac{c_1 x_1 + c_2 x_2 + \dots + c_n x_n - n' \mu_{c(\omega)}}{\sigma_{c(\omega)} \sqrt{n'}} \leq a\right) \rightarrow \frac{1}{\sqrt{2\pi}} \int_{-\infty}^a e^{-\frac{t^2}{2}} dt, \quad (112)$$

met $n' = \sum_{j=1}^n x_j$ van elke c_j komt immers x_j keer voor in de som.

Dit geeft het volgende betrouwbaarheidsinterval:

$$\left| \frac{c_1 x_1 + c_2 x_2 + \dots + c_n x_n - n' \mu_{c(\omega)}}{\sigma_{c(\omega)} \sqrt{n'}} \right| < N_{\frac{1}{2}\alpha}(0, 1). \quad (113)$$

Merk op dat geldt:

$$z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n. \quad (114)$$

Dus als het aantal variabelen j naar oneindig gaat⁶, geeft (113) een $(1 - \alpha)\%$ -betrouwbaarheidsinterval:

$$n' \mu_{c(\omega)} - \sigma_{c(\omega)} \sqrt{n} N_{\frac{1}{2}\alpha}(0, 1) < z < \sigma_{c(\omega)} \sqrt{n'} N_{\frac{1}{2}\alpha}(0, 1) + n' \mu_{c(\omega)}. \quad (115)$$

Dit betekent dat voor een zekere oplossing x de doelfunctiewaarde in $(1 - \alpha)\%$ van de realisaties ω binnen dit interval zal liggen.

De oplossing x met het in zijn geheel laagste betrouwbaarheidsinterval voor een zekere realisatie ω^* zal in $(1 - \alpha)\%$ van de gevallen optimaal zijn.

2.4.2 Onafhankelijke elementen met niet-identieke kansverdeling

Bij een stochastisch lineair programma kan het voorkomen dat niet alle elementen van $A(\omega)$, $b(\omega)$ en $c(\omega)$ dezelfde kansverdeling hebben. Iedere constraint heeft een andere betekenis, dus de kansverdelingen hoeven niet hetzelfde te zijn. In deze paragraaf zal gekeken worden hoe de centrale limietstelling aangepast kan worden om ook een betrouwbaarheidsinterval op te kunnen stellen in het geval van onafhankelijke elementen in de stochastische variabelen $A(\omega)$, $b(\omega)$ en $c(\omega)$ die niet dezelfde kansverdeling hebben.

Als het mogelijk is om $\bar{\delta}$ en z uit (108) en (114) toch standaardnormaal te benaderen dan kunnen op gelijke wijze als de voorgaande paragraaf betrouwbaarheidsinterval opgesteld worden voor de toegelaten oplossingen en doelfunctiewaarde.

In [12] wordt Lindebergs conditie geformuleerd:

⁶De centrale limietstelling geldt ook als voor een zekere x_j geldt dat deze naar oneindig gaat, maar dat is in de context van een stochastisch lineair programma triviaal

Stelling 2.6: Lindebergs conditie

Neem $X_{ni}(\omega)$ een triangular array⁷ van onafhankelijke stochastische variabelen. Veronderstel:

- $E[X_{ni}(\omega)] = 0$
- $\sigma_{ni}^2 = E[X_{ni}(\omega)^2]$
- $S_n = \sum_{i=1}^{r_n} X_{ni}(\omega)$
- $s_n^2 = \sum_{i=1}^{r_n} \sigma_{ni}^2$
- Voor voldoende grote n , $s_n > 0$

als geldt:

$$\lim_{n \rightarrow \infty} \sum_{i=1}^{r_n} \frac{1}{s_n^2} \int_{|X_{ni}(\omega)| \geq \varepsilon s_n} X_{ni}^2(\omega) dP = 0 \quad \forall \varepsilon > 0, \quad (116)$$

dan volgt:

$$\frac{S_n}{s_n} \rightarrow N(0, 1). \quad (117)$$

Dit geeft het volgende betrouwbaarheidsinterval:

$$\begin{aligned} \frac{S_n - 0}{s_n \sqrt{n \cdot r_n}} &< N_\alpha(0, 1) \\ -N_\alpha(0, 1) s_n \sqrt{n \cdot r_n} &< S_n < N_\alpha(0, 1) s_n \sqrt{n \cdot r_n}. \end{aligned} \quad (118)$$

Betrouwbaarheidsinterval voor toegelaten oplossing x

Uit (118) volgt dus dat $\tilde{\delta}(\omega)$ de volgende grenzen voor het betrouwbaarheidsinterval voor een zekere oplossing x geeft:

$$-N_\alpha(0, 1) \frac{\sum_{i=1}^{r_n} \sigma_{\delta_i(\omega)}}{\sqrt{n \cdot r_n}} < \tilde{\delta} < N_\alpha(0, 1) \frac{\sum_{i=1}^{r_n} \sigma_{\delta_i(\omega)}}{\sqrt{n \cdot r_n}}. \quad (119)$$

Als geldt dat:

$$\tilde{\delta} < N_\alpha(0, 1) \frac{\sum_{i=1}^{r_n} \sigma_{\delta_i(\omega)}}{\sqrt{n \cdot r_n}} \leq 0, \quad (120)$$

voor een zekere oplossing x , die voldoet aan de Lindeberg conditie, dan is deze in $(1 - \alpha)\%$ van de realisaties ω een toegelaten oplossing.

Betrouwbaarheidsinterval voor doelfunctiewaarde z

Voor z geeft dit het de volgende grenzen voor het betrouwbaarheidsinterval voor een zekere oplossing x :

$$-N_{\frac{1}{2}\alpha}(0, 1) \left(\sum_{i=1}^{r_n} \sigma_{c_i(\omega)} \right) \sqrt{n \cdot r_n} < z < N_{\frac{1}{2}\alpha}(0, 1) \left(\sum_{i=1}^{r_n} \sigma_{c_i(\omega)} \right) \sqrt{n \cdot r_n}, \quad (121)$$

als de elementen van $c(\omega)$ voldoen aan de Lindeberg conditie.

2.4.3 Afhankelijke elementen met identieke kansverdeling

Naast de kansverdeling van de elementen van de stochastische variabelen $A(\omega)$, $b(\omega)$ en $c(\omega)$, speelt ook de afhankelijkheid van de elementen van deze stochastische variabelen een rol. In deze paragraaf zal beschreven worden hoe een betrouwbaarheidsinterval voor de toegelaten oplossing en doelfunctiewaarde met afhankelijke en identiek verdeelde elementen opgesteld kan worden.

In [12] worden verschillende soorten afhankelijkheidscriteria beschreven. Met behulp van afhankelijkheids-criteria wordt beschreven hoe de afhankelijkheid een rol speelt. Het meest gebruikt is een *a-mixing criterium*:

$$P(X \cap Y) - P(X)P(Y) \leq \alpha. \quad (122)$$

⁷Onderling onafhankelijke rij $X_{n1}, \dots, X_{nr_n}$ waarbij $n = 1, \dots, n$ steeds staat voor een andere verdeling. Van elke verdeling n zijn er r_n elementen.

Dit betekent dat het verschil tussen de kans op X en Y bij afhankelijke en onafhankelijke X, Y kleiner dan of gelijk moet zijn aan a . Meer hierover kan gevonden worden in [34] en [43].

Een ander veel gebruikt afhankelijkheids criterium is *b-dependent*. Dit betekent dat voor $i > b$, $X_i(\omega)$ niet meer afhankelijk is. Meer hierover kan gevonden worden in [7].

Het afhankelijkheids criterium van [29] is een α -mixing criterium. Dit criterium is anders dan elders in de literatuur te vinden, omdat hier het exacte verschil tussen de som van afhankelijke en onafhankelijke variabelen in de staart van de rij stochastische variabelen berekend wordt. Om die reden zal dit afhankelijkheids criterium hier toegelicht worden, andere afhankelijkheids criteria zijn te vinden in [34] en [43].

Het afhankelijkheids criterium van [29] is als volgt:

Stelling 2.7

Neem $\{X_i(\omega) : \Omega \rightarrow [0, 1]; 1 \leq i \leq n\}$ een rij van gelijk verdeelde afhankelijke stochastische variabelen die voldoen aan het volgende afhankelijkheids criterium:

$$\int_{[0,1]^n} \left| \mathbb{P} \left(\bigcap_{i=1}^n \{X_{v_i}(\omega) \leq x_{v_i}\} \right) - \prod_{i=1}^n \mathbb{P}\{X_{v_i}(\omega) \leq x_{v_i}\} \right| dx_{v_1} \cdots dx_{v_n} \leq \left(1 - k^{-\delta/2}\right)^{k - k^{\delta/2} - n}, \quad (123)$$

voor een δ , $0 < \delta < 1$ en elke keuze voor indices $v_1, \dots, v_n$ zodanig dat geldt: $k^{\delta/2} < v_1 < v_2 < \dots < v_n \leq k$.

Door de integraal wordt de mate van afhankelijkheid bepaald over het gemiddelde, terwijl in voorgaande literatuur vaak gesproken wordt over afhankelijkheid tussen elke, elkaar in tijd voorgaande variabelen. Voor de rechterkant van de ongelijkheid geldt dat voor een zekere k met een relatief klein aantal stochastische variabelen n , de rechterkant 0 zal worden. Dit betekent dat er geen sprake is van afhankelijkheid. Voor een waarde van $n \approx k - k^{\delta/2}$ zal de rechterkant 1 worden. Dit betekent dat er sterke afhankelijkheid mag zijn in een grote groep van stochastische variabelen en meer onafhankelijkheid in een kleinere groep.

Centrale limietstelling voor afhankelijke identiek verdeelde stochastische variabelen

Met behulp van het afhankelijkheids criterium (123) bewijst [29] de volgende centrale limietstelling voor afhankelijke stochastische variabelen met dezelfde kansverdeling:

Stelling 2.8

Neem $X_i(\omega)$ een rij van stochastische variabelen die dezelfde kansverdeling hebben, zodanig dat geldt: $E(X_1)^{2+\varepsilon} < +\infty$ voor een $\varepsilon > 0$. Laat ε_1 een reëel getal zijn zodanig dat geldt: $\varepsilon_1 < \frac{\varepsilon}{2(1+\varepsilon)}$. Veronderstel dat voor een voldoende grote k de ongelijkheid:

$$\sup \left\{ \left| \mathbb{P} \left(\bigcap_{i=1}^n \{X_{v_i}(\omega) \leq x_{v_i}\} \right) - \prod_{i=1}^n \mathbb{P}\{X_{v_i}(\omega) \leq x_{v_i}\} \right| : (x_{v_1}, \dots, x_{v_n}) \in \mathbb{R}^n \right\} \leq (1 - k^{-\varepsilon_1})^{k - k^{\varepsilon_1} - j} \quad (124)$$

geldt, met $v_1, \dots, v_n$ elke keuze van indices kan zijn zolang geldt:

$$k^{\varepsilon_1} < v_1 < \dots < v_n \leq k. \quad (125)$$

Dan geldt:

$$\text{Als } k \rightarrow \infty, \quad \frac{1}{\sigma\sqrt{k}} \sum_{i=1}^k (X_i(\omega) - E(X_i(\omega))) \rightarrow N(0, 1). \quad (126)$$

Als het aantal stochastische variabelen k oneindig groot wordt, moet voor de staart van de rij stochastische variabelen (125) gelden. Als dit het geval is, kan de verdeling van de afhankelijke elementen met dezelfde kansverdeling van de stochastische variabelen normaal benaderd worden en is het mogelijk een betrouwbaarheidsinterval op te stellen voor de toegelaten oplossing en de doelfunctiewaarde.

Betrouwbaarheidsinterval voor toegelaten oplossing

Voor een zekere oplossing x geldt dat deze een toegelaten oplossing is in $(1 - \alpha)\%$ van de realisaties ω , als geldt:

$$\bar{\delta} < N_{\alpha}(0, 1) \frac{\sigma_{\delta(\omega)}}{\sqrt{k}} + \mu_{\delta(\omega)} \leq 0, \quad (127)$$

en de rij $\delta_1, \delta_2, \dots$ voldoet aan de voorwaarde (123).

Betrouwbaarheidsinterval voor doelfunctiewaarde

Voor een zekere oplossing x geldt dat de doelfunctiewaarde in $(1 - \alpha)\%$ van de realisaties ω zal liggen tussen:

$$-N_{\frac{1}{2}\alpha}(0, 1) \sigma_{\delta(\omega)} \sqrt{k} + \mu_{\delta(\omega)} < z < N_{\frac{1}{2}\alpha}(0, 1) \sigma_{\delta(\omega)} \sqrt{k} + \mu_{\delta(\omega)}, \quad (128)$$

als de rij $c_1, c_2, \dots$ voldoet aan de voorwaarde van (123).

2.4.4 Afhangelijke niet-identiek verdeelde elementen

Tot slot het geval waarin de elementen van de stochastische variabelen $A(\omega)$, $b(\omega)$ en $c(\omega)$ afhankelijk zijn en niet-identiek verdeeld zijn. In [29] wordt de bovenstaande theorie verder uitgebreid om ook voorwaarde voor een centrale limietstelling voor afhangelijke stochastische variabelen die een niet-identieke kansverdeling hebben op te stellen.

Het afhankelijkheids criterium (123) wordt gebruik in combinatie met de Lindeberg Conditie (116). Hieruit wordt de centrale limietstelling afgeleid voor afhangelijke niet-identieke verdeelde stochastische variabelen afgeleid. Deze afleiding wordt niet meegenomen in dit onderzoek. De afleiding is samen met de criteria te vinden in [29].

3 Stochastic Programming

In dit hoofdstuk wordt Stochastic Programming behandeld. Bij Stochastic Programming wordt aangenomen dat de kansverdeling bekend is. Stochastic Programming is een verzamelnaam voor het oplossen van stochastisch lineair programma's met behulp van de kansverdeling. In dit hoofdstuk worden een aantal oplossmethoden die binnen deze categorie vallen besproken.

In paragraaf 3.1 en 3.2 wordt uitgegaan van een model waarin het mogelijk is om niet aan alle constraints te voldoen. Na de originele beslissing x moet nog een beslissing y uitgevoerd worden om eventuele overtredingen van de constraints te compenseren. Deze tweede actie y wordt in de literatuur *recourse* genoemd.

In paragraaf 3.3 wordt een algoritme gegeven om Stochastic Programming met *recourse* op te lossen. Als er veel mogelijke scenario's zijn, ontstaan er namelijk erg veel constraints. Het stochastisch lineair programma kan op deze manier te groot worden om volledig op te lossen, daarom is een efficiënt algoritme nodig.

In paragraaf 3.4 wordt een stochastisch lineair programma op een andere manier benaderd. Hier wordt vanuit gegaan dat met kans $(1 - \alpha)$ de constraints overschreden mogen worden. Op deze manier wordt er dus geaccepteerd dat er een kans is op dat x niet toegelaten is. Het uiteindelijke probleem is lastig op te lossen, daarom wordt *Sample Average Approximation* gebruikt om de oplossing van het verkregen probleem te benaderen.

Tot slot worden er overige methoden beschouwd in paragraaf 3.5. Deze methoden zijn beknopt beschreven met verwijzingen naar literatuur voor verder onderzoek.

3.1 Stochastisch programmeren met recourse

In deze paragraaf zal het principe van *recourse* worden uitgelegd en daarna worden oplossmethoden aangedragen. Deze sectie is gebaseerd op [28] en [44]. Het idee van *recourse* is dat, na de initiële beslissing, er later een tweede actie plaatsvindt – de zogeheten *recourse* – om te verbeteren wat er fout ging of een boete voor hoe ver de initiële beslissing naast de grenzen zit. Het gaat als volgt in zijn werk. Een normaal stochastisch lineair programma ziet er als volgt uit:

$$\begin{aligned} \min_x \quad & c^T(\omega)x \\ \text{s.t.} \quad & A(\omega)x \leq b(\omega), \\ & x \geq 0, \end{aligned} \tag{129}$$

met $A(\omega)$, $b(\omega)$ en $c(\omega)$ stochastische variabelen en $x \in \mathbb{R}^n$. In dit probleem moet een beslissing x worden gemaakt maar door de stochastische natuur van $A(\omega)$, $b(\omega)$ en $c(\omega)$ kan het voorkomen dat x achteraf niet voldoet aan de voorwaarden $A(\omega)x \leq b(\omega)$. Dit is waar de *recourse* zijn entree maakt. De *recourse* is een tweede actie y , die plaatsvindt nadat x is bepaald en $A(\omega)$, $b(\omega)$ en $c(\omega)$ hun waarde hebben aangenomen. Afhankelijk van het probleem wordt deze y dan bijvoorbeeld extra inkoop of boete. Deze y zorgt er dan voor dat er weer aan de voorwaarden wordt voldaan:

$$A(\omega)x + M(\omega)y \leq b(\omega). \tag{130}$$

Hierbij is $M(\omega)$ een stochastische variabele die de interactie tussen de originele beslissing x en de *recourse* y beschrijft. Er zijn uiteraard ook kosten om y te laten plaatsvinden. De kosten van y worden beschreven in de stochastische variabele $q(\omega)$. De kosten voor y zijn dan $q^T(\omega)y$. Dit verandert het probleem naar de volgende vorm:

$$\begin{aligned} \min_{x,y} \quad & E[c^T(\omega)x] + E[q^T(\omega)y|x] \\ \text{s.t.} \quad & A(\omega)x + M(\omega)y \leq b(\omega), \\ & x, y \geq 0. \end{aligned} \tag{131}$$

Voordat de stochastische variabelen hun waarden aannemen, moet x worden besloten en y wordt erna besloten. Het doel is de verwachte kosten te minimaliseren. Hier wordt de verwachtingswaarde geïntroduceerd, omdat y van te voren vastgelegd moet worden, zonder de uitkomsten van de stochastische variabelen bekend zijn.

3.2 Scenario-analyse

In dit hoofdstuk zal er gekeken worden naar een oplosmethode om het probleem met recourse zo goed mogelijk op te lossen. Deze sectie is ook gebaseerd op [44] en [28]. In scenario-analyse wordt gekeken naar de gevallen waarbij er maar een eindig aantal scenario's voor de stochastische variabelen mogelijk zijn en de kansverdeling van de stochastische variabelen bekend zijn.

Aangezien er een gelimiteerd aantal scenario's $\omega_i := (c_i, q_i, A_i, M_i, b_i)$, $i = 1, \dots, K$ zijn, kan voor elk van deze ω_i een optimale y_i die alleen afhankelijk is van x bepaald worden. Dit geeft de volgende K minimalisatieproblemen voor y :

$$\begin{aligned} \min_y \quad & q_i^T y \\ \text{s.t.} \quad & A_i x + M_i y \leq b_i \\ & y \geq 0 \end{aligned} \tag{132}$$

We noteren met y_i de optimale oplossing van dit probleem. Voor elke x is er dus een y_i voor elk scenario ω_i . Omdat elke kans p_i op ω_i bekend is, valt hiermee voor elke x de verwachtingswaarde van de recourse y , oftewel $E[q^T(\omega)y|x]$, uit te rekenen. Op deze manier is het stochastisch lineair programma aan te passen naar een deterministisch lineair programma, waarmee de optimale x gevonden kan worden. Dit wordt gedaan door (132) voor elk scenario toe te voegen aan (131). Het uiteindelijke deterministisch lineair programma heeft dan de volgende vorm:

$$\begin{aligned} \min_x \quad & \sum_{i=1}^K p_i (c_i^T x + q_i^T y_i) \\ \text{s.t.} \quad & A_i x + M_i y_i \leq b_i, \quad i = 1, \dots, K \\ & x \geq 0, y_i \geq 0 \end{aligned} \tag{133}$$

Dit systeem kan voor alle gevallen met eindig aantal scenario's de optimale oplossing vinden, maar de oplostijd neemt wel lineair toe met het aantal scenario's.

3.3 The L-shaped Method

Het oplossen van een stochastisch lineair programma met recourse neemt veel tijd in beslag als er veel verschillende scenario's mogelijk zijn. The L-shaped Method is een efficiënt algoritme dat stochastisch programmeren met recourse oplost. De referentie die hierbij het belangrijkste is, is [13]. Het algoritme begint met het oplossen van het oorspronkelijke probleem, om vervolgens extra constraints toe te voegen als de oplossing niet toegelaten (*feasibility cut*) of niet optimaal (*optimality cut*) is.

Het stochastisch lineair programma in (131) is equivalent met het volgende deterministisch lineair programma:

$$\begin{aligned} \min_x \quad & c^T x + Q(x) \\ \text{s.t.} \quad & x \geq 0, \end{aligned} \tag{134}$$

met $Q(x)$ als volgt gedefinieerd:

$$Q(x) := E_{\xi} Q(x, \xi(\omega)), \tag{135}$$

$$\begin{aligned} Q(x, \xi(\omega)) := \min_y \quad & q(\omega)^T y \\ \text{s.t.} \quad & A(\omega)x + M(\omega)y \leq b(\omega), \\ & y \geq 0. \end{aligned} \tag{136}$$

Het algoritme gebruikt θ om de functie van de kosten $Q(x)$ van recourse te schatten. Er wordt gebruikt dat $\theta \geq Q(x)$. De door het algoritme nieuw gegenereerde optimality constraints geven de grenzen van θ aan.

Om het algoritme beter te begrijpen, wordt eerst het bewijs gegeven. Hieruit valt af te leiden waar de constraints in (146) vandaan komen.

3.3.1 Bewijs van het algoritme

In dit bewijs wordt gebruik gemaakt van notatie (6). Er wordt constructief bewezen dat de constraints $E_m x + \theta \geq e_m$, $m = 1, \dots, s$ *supporting hyperplanes* van de functie $Q(x)$ zijn en dat het algoritme convergeert naar de optimale oplossing. Dit bewijs komt uit [13], waarbij gebruik is gemaakt van [48].

Merk op dat het stochastisch lineair programma in (134) equivalent is met het oplossen van

$$\begin{aligned} \min_{x, \theta} \quad & c^T x + \theta \\ \text{s.t.} \quad & Q(x) \leq \theta, \\ & x \geq 0, \quad \theta \in \mathbb{R}. \end{aligned} \quad (137)$$

In *step 3* van het algoritme wordt (149) opgelost voor elke $i = 1, \dots, K$. Dit geeft de optimale simplex multipliers S_i^ν , $i = 1, \dots, K$. Vanuit de dualiteit van lineair programmeren volgt dat $\forall i$:

$$Q(x^\nu, \xi_i) = (S_i^\nu)^T (b_i - A_i x^\nu), \quad (138)$$

En aangezien $Q(x, \xi_i)$ een convexe functie is, volgt uit de subgradiënt ongelijkheid dat:

$$Q(x, \xi_i) \geq (S_i^\nu)^T b_i - (S_i^\nu)^T A_i x. \quad (139)$$

De verwachting van deze twee ongelijkheden nemen geeft respectievelijk:

$$Q(x^\nu) = E(S^\nu)^T (b - A x^\nu) = \sum_{i=1}^K p_i \cdot (S_i^\nu)^T (b_i - A_i x^\nu) \quad (140)$$

en

$$Q(x) \geq E(S^\nu)^T (b - A x) = \sum_{i=1}^K p_i (S_i^\nu)^T b_i - \left(\sum_{i=1}^K p_i (S_i^\nu)^T A_i \right) x. \quad (141)$$

Uit het feit dat $\theta \geq Q(x)$, volgt dat (x, θ) toegelaten is voor (137) dan en slechts dan als $\theta \geq E(S^\nu)^T (b - M x)$, wat precies overeenkomt met de tweede rij constraints in (146) met E_m en e_m gedefinieerd zoals in (150). Dit definieert dus de volgende constraints:

$$\begin{aligned} & E_m x + \theta \geq e_m \\ \text{met} \quad & E_m = \sum_{i=1}^K p_i (S_i^\nu)^T A_i \\ \text{en} \quad & e_m = \sum_{i=1}^K p_i (S_i^\nu)^T b_i. \end{aligned} \quad (142)$$

Hierbij komen de simplex multipliers S_i^ν uit iteratie ν , behorende bij scenario i . E_m en e_m worden geconstrueerd als er reeds $m - 1$ constraints van deze vorm aanwezig zijn. m is simpelweg een teller in het algoritme

Als (x^ν, θ^ν) een optimale oplossing is voor (137), geldt dat $Q(x^\nu) = \theta^\nu$, omdat θ geen andere restricties heeft dan $\theta \geq Q(x)$. Dit gebeurt wanneer $\theta^\nu = E(S^\nu)^T (b - A x^\nu)$, wat hetzelfde is als het criterium om te stoppen in *step 3*. Want $\theta^\nu \geq w^\nu = e_{m+1} - E_{m+1} x^\nu = E(S^\nu)^T (b - A x^\nu) = Q(x^\nu)$.

Dus in elke iteratie geldt òf $\theta^\nu \geq Q(x^\nu)$, wat betekent dat je klaar bent, òf $\theta^\nu < Q(x^\nu)$. In het laatste geval geeft geen van de al bestaande optimality cuts $\theta \geq Q(x)$, dus wordt een nieuwe verzameling van simplex multipliers S_i^ν gedefinieerd om een geschikte constraint te genereren. Het algoritme convergeert vanuit het feit dat er maar een eindig aantal verschillende combinaties K van de multipliers S_i zijn, omdat elk bij één van de eindig aantal verschillende opties hoort.

Nu moet nog bewezen worden dat er een eindig aantal constraints $D_l x \geq d_l$ nodig zijn zo dat geldt:

$$\begin{aligned} A_i x + M_i y &= b_i \\ M_i y &= b_i - A_i x. \end{aligned} \quad (143)$$

Dit betekent dat

$$b_i - A_i x \in \text{pos}^8 M(\omega), \quad i = 1, \dots, K. \quad (144)$$

⁸ $\text{pos } M = \{t M y = t, y \geq 0\}$

In stap 2 wordt het deelprobleem (147) opgelost. Daarin wordt getest of $b_i - A_i x^v \in \text{pos } M(\omega)$ voor $i = 1, \dots, K$. Als dit niet het geval is, betekent dit dat voor sommige $i = 1, \dots, K$, $b_i - A_i x^v \notin \text{pos } M(\omega)$. Dan moet er dus een hyperplane zijn die $b_i - A_i x$ scheidt van $\text{pos } M(\omega)$. Deze hyperplane moet voldoen aan $\sigma^T t \leq 0, \forall t \in \text{pos } M(\omega)$ en $\sigma^T (b_i - A_i x^v) > 0$. In stap 2 wordt deze hyperplane verkregen door σ te vervangen door σ^v , de simplex multipliers van het deelprobleem (147).

Door sterke dualiteit geldt in stap 2 dat $w' > 0$ hetzelfde is als $(\sigma^v)^T (b_i - A_i x^v) > 0$. Daarnaast wordt voldaan aan $(\sigma^v)^T M(\omega) \leq 0$, omdat σ^v optimale simplex multipliers zijn en, bij het optimum, de gereduceerde kosten geassocieerd met y niet negatief moeten zijn.

Dus heeft σ^v de gewenste eigenschappen. Daaruit volgt dat een noodzakelijke voorwaarde voor x toegelaten, is $(\sigma^v)^T M(\omega) \leq 0$, wat precies overeenkomt met de eerste rij constraints in (146) met D_l en d_l gedefinieerd zoals in (148). Dit definieert dus de volgende constraints:

$$\begin{aligned} & D_l x \geq d_l \\ \text{met} \quad & D_l = (\sigma^v)^T A_i \\ \text{en} \quad & d_l = (\sigma^v)^T b_i. \end{aligned} \tag{145}$$

Er zijn een eindig aantal van deze constraints, omdat er maar een eindig aantal optimale basissen zijn voor het deelprobleem (147).

3.3.2 Het algoritme

In dit gedeelte wordt het algoritme gegeven. In 3.3.3 worden de stappen van het algoritme uitgelegd. Het algoritme ziet er als volgt uit:

Stap 0: Neem $r = s = v = 0$.

Stap 1: Neem $v = v + 1$. Los het volgende deterministisch lineair programma op:

$$\begin{aligned} \min_{x, \theta} \quad & c^T x + \theta \\ \text{s.t.} \quad & D_l x \geq d_l, \quad l = 1, \dots, r \\ & E_m x + \theta \geq e_m, \quad m = 1, \dots, s \\ & x \geq 0, \quad \theta \in \mathbb{R}. \end{aligned} \tag{146}$$

Neem (x^v, θ^v) een optimale oplossing. Als er geen constraint $E_m x + \theta \geq e_m$ aanwezig is, neem $\theta^v = -\infty$ en neem deze niet mee in de berekening voor x^v .

Stap 2: Voor $i = 1, \dots, K$, los het volgende deterministisch lineair programma op:

$$\begin{aligned} \min_{v^+, v^-, y} \quad & w' = e^T v^+ + e^T v^- \\ \text{s.t.} \quad & M_i y + I v^+ - I v^- \leq b_i - A_i x^v \\ & y \geq 0, \quad v^+ \geq 0, \quad v^- \geq 0, \end{aligned} \tag{147}$$

met $e^T = (1, \dots, 1)$, totdat voor een k de optimale waarde $w' > 0$. Neem in dit geval σ^v de simplex multipliers⁹ van de optimale oplossing van (147) en definieer

$$\begin{aligned} D_{l+1} &= (\sigma^v)^T A_i \\ d_{l+1} &= (\sigma^v)^T b_i \end{aligned} \tag{148}$$

om een nieuwe constraint (feasibility cut) te genereren. Voeg deze constraint toe aan het systeem (146) en neem $r = r + 1$. Als $\forall i, w' = 0$, ga naar *Stap 3*.

⁹De simplex multipliers $(S_1, \dots, S_m)$ van een oplossing zijn de veelvouden van hun initiële systeem van vergelijkingen dusdanig dat, als al deze vergelijkingen worden vermenigvuldigd met hun simplex multiplier en van de doelfunctie afgehaald worden, de coëfficiënten van de basisvariabelen nul zijn. Dus $(S_1 a_{1j} + S_2 a_{2j} + \dots + S_m a_{mj} = c_j)$.

Stap 3: Voor $i = 1, \dots, K$, los het volgende deterministisch lineair programma op:

$$\begin{aligned} \min_y \quad & w = q_i^T y \\ \text{s.t.} \quad & M_i y \leq b_i - A_i x^\nu \\ & y \geq 0 \end{aligned} \tag{149}$$

Neem nu S_i^ν de simplex multipliers van de optimale oplossing van probleem i van (149) en definieer

$$\begin{aligned} E_{m+1} &= \sum_{i=1}^K p_i \cdot (S_i^\nu)^T A_i \\ e_{m+1} &= \sum_{i=1}^K p_i \cdot (S_i^\nu)^T b_i \end{aligned} \tag{150}$$

Neem $w^\nu = e_{m+1} - E_{m+1} x^\nu$. Als $\theta^\nu \geq w^\nu$, stop; x^ν is een optimale oplossing. Anders, neem $s = s + 1$, gebruik (150) om een nieuwe constraint (optimality cut) te genereren en ga terug naar *Stap 1*.

3.3.3 Uitleg van het algoritme

Omdat een probleem met recourse erg veel oplostijd kan vergen als er veel mogelijke scenario's zijn, lost dit algoritme het probleem in hapbare stukken op.

In de eerste stap wordt er een optimale oplossing gezocht voor de doelfunctie zonder de oorspronkelijke constraints. Er zijn weinig tot geen constraints aanwezig, waardoor dit snel gebeurt.

In *stap 2* wordt er gekeken of deze optimale oplossing toegelaten is. Dit gebeurt door te minimaliseren over v^+ , v^- en y . Hierbij is de kandidaat-oplossing (x^ν, θ^ν) al bekend en gaat het dus vooral om y . (147) vindt dus een optimale oplossing (v^+, v^-, y) gebaseerd op x^ν . Omdat $e^T = (1, \dots, 1)$, is het minimum voor w' gelijk aan 0. Dit minimum wordt bereikt als $v^+ = v^- = 0$. Dan is x^ν dus toegelaten voor scenario i , omdat dan geldt

$$M_i y + I v^+ - I v^- = M_i y \leq b_i - A_i x^\nu. \tag{151}$$

Dus als de gevonden optimale oplossing (x^ν, θ^ν) voor een scenario i niet toegelaten is, geldt $w' > 0$ en moet er een nieuwe constraint toegevoegd worden, zodat deze oplossing niet meer gevonden kan worden. Vandaar de naam 'feasibility cut'.

In *stap 3* wordt er gekeken of de gevonden optimale oplossing van *stap 1* daadwerkelijk optimaal is in alle gevallen. Hier draait het vooral om de kosten van de recourse ($q_i^T y$). In het bewijs van het algoritme wordt er laten zien waarom er aan de voorwaarde $\theta^\nu \geq w^\nu$ moet worden voldaan om x^ν een optimale oplossing te laten zijn.

3.4 Chance constrained programming

Een ander soort probleem is een *chance constrained* probleem. Dit is een probleem waar gezocht wordt naar de optimale x , die met een maximale kans van α niet toegestaan is. Dit ziet er als volgt uit:

$$\begin{aligned} \min_x \quad & c^T(\omega) x \\ \text{s.t.} \quad & P[A(\omega)x \leq b(\omega)] \geq 1 - \alpha, \\ & x \geq 0. \end{aligned} \tag{152}$$

In deze sectie zal met behulp van *Sample Average Approximation* de optimale oplossing voor dit probleem benaderd worden. Eerst wordt uitgelegd wat Sample Average Approximation is en vervolgens zal het toegepast worden op een chance constrained probleem. Deze sectie is gebaseerd op [38] en [37].

3.4.1 Sample Average Approximation

Scenario-analyse wordt niet alleen toegepast op problemen met recourse, het kan ook op simpele wijze worden toegepast op standaard stochastisch lineair programma's, zolang deze een eindig aantal scenario's hebben.

Hiermee kunnen de verwachte kosten worden geminimaliseerd. In het geval dat de stochastische variabelen oneindig veel mogelijke uitkomsten hebben, kan met Sample Average Approximation een steekproef genomen worden om het aantal scenario's terug te brengen naar een eindig aantal en daarmee een benadering van de optimale oplossing te kunnen vinden. Meer informatie over Sample Average Approximation is te vinden in [31].

Eerst zal er laten zien worden hoe scenario-analyse op een stochastisch lineair programma met eindig aantal scenario's gaat. Er wordt gezocht naar de x die de laagste verwachting geeft. Het probleem per scenario ziet er als volgt uit:

$$\begin{aligned} \min_x \quad & c^T(\omega)x \\ \text{s.t.} \quad & A(\omega)x \leq b(\omega), \\ & x \geq 0. \end{aligned} \tag{153}$$

Hierbij is wederom $\omega := (c, A, b) \in \Omega$ een scenario en Ω de eindige verzameling van alle scenario's. De scenario's worden genummerd met indices $i = 1, \dots, K$. Nu kan de verwachtingswaarde van de kostenfunctie geminimaliseerd worden:

$$\begin{aligned} \min_x \quad & E(c^T(\omega)x) = \sum_{i=1}^K (p_i c_i^T) x \\ \text{s.t.} \quad & A_i x \leq b_i \quad i = 1 \dots K, \\ & x \geq 0, \end{aligned} \tag{154}$$

met p_i de kans op scenario i . Echter komen vaak de realisaties van de stochastische variabelen $A(\omega)$, $b(\omega)$ en $c(\omega)$ niet uit een verzameling met een eindig aantal scenario's, maar uit een interval of een continue kansverdeling. Wanneer dit bekeken wordt met scenario-analyse, zouden er oneindig veel scenario's zijn, wat het oplossen onmogelijk maakt.

Dit kan echter wel opgelost worden met Sample Average Approximation. Bij deze methode wordt er met Monte Carlo een steekproef van n scenario's genomen, om vervolgens de x te optimaliseren voor deze verzameling scenario's verkregen door de steekproef. Er wordt aangenomen dat deze scenario's identiek verdeeld en onafhankelijk zijn.

Het oplossen van zo'n systeem gaat op de vergelijkbare wijze als het stochastisch lineair programma in (153), alleen, omdat het een steekproef is, is er hierbij geen p_i bekend. Daarom wordt aan elk scenario de kans $1/n$ toegewezen, om de kansverdeling te benaderen. Uiteindelijk wordt het volgende lineair programma verkregen:

$$\begin{aligned} \min_x \quad & E(c^T(\omega)x) = \frac{1}{n} \sum_{i=1}^n (c_i^T) x \\ \text{s.t.} \quad & A_i x \leq b_i \quad i = 1 \dots n, \\ & x \geq 0. \end{aligned} \tag{155}$$

Dit is een deterministisch lineair programma, dus efficiënt op te lossen. Hiermee zal een benadering voor de optimale x gevonden kunnen worden. Hoe meer scenario's er gecreëerd worden, hoe beter de kansverdeling benaderd wordt en dus hoe beter de gevonden oplossing de optimale oplossing zal benaderen. Echter kost het nemen en berekenen van meer scenario's ook meer tijd.

Verder is het interessant op te merken dat deze manier ook te gebruiken is voor alle systemen waarbij de kansverdeling volledig onbekend is, maar het wel mogelijk is om een steekproef te doen.

3.4.2 Chance constrained programming

Omdat Sample Average Approximation zo algemeen is, kan dit op nog veel meer soorten stochastisch lineair programma's toegepast worden. Zo is het bijvoorbeeld ook erg handig voor stochastisch lineair programma's met chance constraints. Een lineair programma met chance constraints ziet er als volgt uit:

$$\begin{aligned} \min_x \quad & c^T(\omega)x \\ \text{s.t.} \quad & P[A(\omega)x \leq b(\omega)] \geq 1 - \alpha, \\ & x \geq 0. \end{aligned} \tag{156}$$

Hier is α de toegestane kans dat de oplossing niet toegelaten is. Met $P[A(\omega)x \leq b]$ wordt de kans bedoeld dat de constraint $A(\omega)x \leq b$ voldoet. Verder wordt hier aangenomen dat alle stochastiek in $A(\omega)$ zit. Het probleem kan herschreven worden als:

$$\begin{aligned} \min_x \quad & c^T(\omega)x \\ \text{s.t.} \quad & p(x) \leq \alpha, \\ & x \geq 0, \end{aligned} \tag{157}$$

waarbij:

$$p(x) := P[A(\omega)x \not\leq b(\omega)]. \tag{158}$$

Op deze manier geeft $p(x)$ de kans aan dat x niet toegelaten is. Als $A(\omega)$ een oneindig aantal scenario's heeft, zijn dit soort problemen zijn erg lastig op te lossen. Zelfs numeriek kan het veel tijd kosten. Daarom is een benadering met gebruik van Sample Average Approximation een goede optie.

Sample Average Approximation maakt wederom gebruik van een steekproef om de kansverdeling af te schatten. In dit geval moet $p(x) \leq \alpha$ afgeschat worden. Dit kan gedaan worden door te kijken naar het gedeelte van de steekproeven waar x toegelaten is. Om dit te doen, worden op de volgende manieren de constraints omgeschreven:

$$G(x, \omega) = \max(A(\omega)x - b(\omega)). \tag{159}$$

Als $G(x, \omega) \leq 0$, dan is $A(\omega)x \leq b$ en dan is x dus toegelaten. Hierdoor is met één getal te controleren of x toegelaten is. Dit is nodig voor de indicator functie. De indicatorfunctie $\mathbb{1}_{(0, \infty)}(t)$ is als volgt gedefinieerd:

$$\mathbb{1}_{(0, \infty)}(t) := \begin{cases} 1 & \text{als } t > 0 \\ 0 & \text{als } t \leq 0 \end{cases}. \tag{160}$$

Er geldt dus dat $\mathbb{1}_{(0, \infty)}(G(x, \omega)) = 0$, als x toegelaten is en $\mathbb{1}_{(0, \infty)}(G(x, \omega)) = 1$ als x niet toegelaten is. Het idee is om $p(x)$ te benaderen met het gedeelte van de steekproeven waar x toegelaten is. Wederom wordt aan alle n scenario's van de steekproef de kans $\frac{1}{n}$ toegewezen. De constraint wordt dan omgeschreven naar:

$$p_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{(0, \infty)}(G(x, \omega_i)). \tag{161}$$

$p_n(x)$ is hier het gedeelte van de scenario's van de steekproef waarbij x toegelaten is. Deze $p_n(x)$ kan dan gebruikt worden om de functie $p(x)$ te benaderen. Het uiteindelijke probleem wordt dan:

$$\begin{aligned} \min_x \quad & \frac{1}{n} \sum_{i=1}^n c_i^T x \\ \text{s.t.} \quad & p_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{(0, \infty)}(G(x, \omega_i)) \leq \gamma, \\ & x \geq 0. \end{aligned} \tag{162}$$

Voor deze benadering kan dus het toegelaten gedeelte niet toegelaten gevallen, γ , anders zijn dan het toegelaten gedeelte uit het originele probleem, α . Over het algemeen geldt dat als $\gamma < \alpha$, met grote kans geldt dat alle toegelaten x van (162) ook toegelaten zijn in (156). In [37] staat meer informatie over de keuze van γ en de invloed daarvan.

3.5 Andere Stochastic Programming modellen

Naast de al genoemde manieren zijn er nog een paar andere modellen van Stochastic Programming. Deze sectie volgt het overzicht van [13]. In deze sectie zal nog een beknopte beschrijving gegeven worden van *Multistage Stochastic Programming* en *Stochastic Integer Programming*.

Multistage Stochastic Programs with Recourse

Stochastic Multistage Programs lijken op de stochastisch lineair programma's met recourse zoals beschreven in paragraaf 3.1. In het standaard two-stage recourse model zijn er twee momenten waar een beslissing moet worden genomen, de originele beslissing x en de recourse y . Tussen deze momenten wordt ω bekend, waardoor de stochastische variabelen hun waarden aannemen. In Multistage Stochastic Programming zijn er niet twee, maar $T + 1$ momenten waarop een beslissing x_i moet worden genomen. Tussen al deze momenten in wordt steeds een nieuwe ω_i bekend, waardoor steeds meer variabelen hun waarden aannemen. Uiteindelijk wordt er gezocht naar een *Recourse function*, die er als volgt uit ziet:

$$x(\omega) = (x_0, x_1(\omega_1), x_2(\omega_1, \omega_2), \dots, x_T(\omega_1, \dots, \omega_T)).$$

Belangrijk is de eigenschap *nonanticipativity*, wat inhoudt dat de beslissing op moment t , x_t , alleen af mag hangen van alle x_j en ω_j met $j \leq t$. Oftewel, de beslissing op een moment t mag alleen afhangen van alle eerdere beslissingen en alle bekende ω_i op dat moment. Het probleem ziet er dan als volgt uit:

$$\begin{array}{llll}
\min_x & c_0^T x_0 + E(c_1^T(\omega_1)x_1 + \dots + c_T^T(\omega_1, \dots, \omega_n)x_T) & & \\
\text{s.t.} & A_0 x_0 & \leq & b_0 \\
& M_0(\omega_1)x_0 + A_1(\omega_1)x_1 & \leq & b_1(\omega_1) \\
& \quad \ddots & & \vdots \\
& M_{T-1}(\omega_1, \dots, \omega_T)x_{T-1} + A_T(\omega_1, \dots, \omega_T)x_T & \leq & b_T(\omega_T) \\
& x_i \geq 0, \quad i = 0 \dots T. & &
\end{array} \tag{163}$$

De optimale oplossing is dus de recourse functie met de laagste verwachte waarde. De uitgebreide oplossmethoden voor dit probleem staan beschreven in [13] en [19].

Stochastic Integer Programming met Recourse

Stochastic Integer Programming met Recourse zijn in principe hetzelfde als de normale systemen met recourse, alleen zijn hier variabelen die geheel of gedeeltelijk in integers gegeven moeten worden. Een voorbeeldprobleem ziet er als volgt uit:

$$\begin{array}{ll}
\min_{x,y} & E(c(\omega)^T x + q(\omega)^T y) \\
\text{s.t.} & A(\omega)x + M(\omega)y \leq b(\omega) \\
& x \in \mathbb{R}_+^{n_1-z_1} \times \mathbb{Z}_+^{z_1}, \\
& y \in \mathbb{R}_+^{n_2-z_2} \times \mathbb{Z}_+^{z_2}.
\end{array} \tag{164}$$

Hierbij zijn $n_1, n_2, z_1, z_2 \in \mathbb{Z}_+$ met $z_1 \leq n_1, z_2 \leq n_2$ de getallen die vastleggen hoeveel elementen van x en y integers moeten zijn. Van de totale n_1 elementen in x moeten er z_1 integers zijn en van de totale n_2 elementen in y moeten er z_2 integers zijn. Meer over Stochastic Integer Programming met Recourse kan worden gevonden in [3] en [13].

4 Fuzzy Programming

In dit verslag wordt vooral gewerkt met een bepaalde soort onzekerheid. Er wordt vooral gewerkt met de kans dat een stochastische variabele een bepaalde waarde aanneemt. De stochastische variabele kan dus verschillende waarden aannemen en daarover bestaat onzekerheid: er is van te voren niet bekend welke waarde de variabele aan gaat nemen. Maar niet elke onzekerheid gaat over waar of niet waar.

Neem als voorbeeld een weersvoorspelling. Het is algemeen bekend dat er niet met zekerheid gezegd kan worden wat het weer van morgen gaat worden. Het zou bijvoorbeeld regenachtig of zonnig en warm of koud kunnen worden. Maar, wat is dan de definitie van zonnig? Is dat als er helemaal geen wolken zijn, als het maximaal 20% bewolkt is? Dit zijn menselijke intuïties die wiskundig gezien lastig te beschrijven zijn. Dit is waar de *Fuzzy numbers* hun intrede maken. Fuzzy numbers geven een gewicht aan de mate waarin een bepaalde waarde tot een groep behoort. In het geval van of het weer zonnig is of niet, zou de waarde 1 betekenen dat er 0% bewolking is en 0,5 dat er bijvoorbeeld 25% bewolking is.

Bij Fuzzy Programming hoort dus een andere kansverdeling en interpretatie van het probleem dan bij Stochastic Programming. Hier is niet duidelijk of iets wel of niet waar is, maar wordt een waarde gegeven aan in hoeverre een bewering waar is. Dit heeft met name toepassing op problemen met subjectieve onderwerpen, zoals leeftijd in het eerder genoemde voorbeeld.

In dit hoofdstuk zal eerst kort beschreven worden wat Fuzzy numbers zijn en waar ze voor worden gebruikt. Vervolgens wordt Fuzzy Programming bekeken. Dit hoofdstuk is gebaseerd op [33] en [50].

4.1 Het idee achter een Fuzzy number

In deze sectie wordt een andere soort verzameling gebruikt dan de algemeen gebruikte universele verzameling U . Hier wordt gebruik gemaakt van een *Fuzzy verzameling* F . Om uit te kunnen leggen wat een Fuzzy verzameling is, is eerst een ander begrip nodig: een *membership function*.

Een membership function μ_F van een Fuzzy verzameling F is een functie die aan een element uit een universele verzameling een waarde hecht, die aangeeft in hoeverre het element bij de Fuzzy verzameling hoort. De functie beeldt doorgaans de elementen uit de universele verzameling af op het interval $[0, 1]$,

$$\mu_F : U \rightarrow [0, 1]. \quad (165)$$

Zoals eerder aangegeven, zou dit bijvoorbeeld het percentage bewolking kunnen zijn, als men wil weten hoe zonnig het is. Hierbij geldt: hoe hoger, hoe beter. Een membership functie kan verschillende vormen aannemen. Zo kan het bijvoorbeeld een lineaire functie zijn, maar ook een normale verdeling.

Zie het volgende voorbeeld om te verduidelijken hoe een Fuzzy verzameling eruit ziet. Het gaat hier om het begrip jong, middelbare leeftijd of oud. Hierbij is het dus lastig om te zeggen wanneer iemand jong, middelbaar of oud is. Een persoon van 15 zal iemand van 40 misschien al wel oud vinden, terwijl de persoon van 40 het daar ongetwijfeld niet mee eens is. Daarom zou je de volgende Fuzzy verzamelingen kunnen definiëren:

$$\begin{aligned} \text{Jong:} \quad F_1(x) &= \begin{cases} 1 & x \leq 20 \\ \frac{35-x}{15} & 20 < x < 35 \\ 0 & x \geq 35, \end{cases} \\ \text{Middelbare leeftijd:} \quad F_2(x) &= \begin{cases} 0 & x \leq 20 \text{ of } x \geq 60 \\ \frac{x-20}{15} & 20 < x < 35 \\ 1 & 35 \leq x \leq 45 \\ \frac{60-x}{15} & 45 < x < 60, \end{cases} \\ \text{Oud:} \quad F_3(x) &= \begin{cases} 0 & x \leq 45 \\ \frac{x-45}{15} & 45 < x < 60 \\ 1 & x \geq 60. \end{cases} \end{aligned} \quad (166)$$

4.2 Fuzzy Programming

Met Fuzzy Programming wordt vaak gebruik gemaakt van een boven- en een ondergrens. De ondergrens geeft aan welke waarde je minimaal wilt hebben, de bovengrens geeft de maximale toegestane waarde aan. In (166) is dit goed terug te zien. In het geval van de Fuzzy verzameling voor jong is de 'ondergrens' 20 en de 'bovengrens' 35. De b_i worden dan vervangen door de ondergrens b'_i en de bovengrens b''_i .

4.2.1 Lineaire membership function

Met b'_i als ondergrens en b''_i als bovengrens, kan de volgende membership functie opgesteld worden voor elke constraint:

$$\mu(A_i x) = \begin{cases} 1 & A_i x \leq b'_i \\ 1 - \frac{A_i x - b'_i}{b''_i - b'_i} & b'_i < A_i x \leq b''_i \\ 0 & A_i x > b''_i. \end{cases} \quad (167)$$

Hetzelfde kan gedaan worden voor de doelfunctie $c^T(\omega)x$. Er wordt gesteld dat $z' \leq c^T(\omega)x \leq z''$. Op dezelfde manier als bij de constraints, krijgt de doelfunctie de volgende membership function:

$$\mu(c(\omega)^T x) = \begin{cases} 0 & c^T(\omega)x < z' \\ \frac{c^T(\omega)x - z'}{z'' - z'} & z' \leq c^T(\omega)x \leq z'' \\ 1 & c^T(\omega)x > z''. \end{cases} \quad (168)$$

Dus, hoe hoger de waarde van de membership function, hoe beter het probleem wordt opgelost. Daarom wordt gezocht naar een waarde van x waarvoor alle membership functions een zo groot mogelijke waarde hebben. Het minimalisatieprobleem wordt daarom:

$$\alpha = \max_x \min[\mu(c^T(\omega)x), \mu(A_1 x), \mu(A_2 x), \dots, \mu(A_m x)]. \quad (169)$$

Dit is equivalent met

$$\begin{aligned} \max_{x, \alpha} \quad & \alpha \\ \text{s.t.} \quad & \mu(c^T(\omega)x) \geq \alpha, \\ & \mu(A_i x) \geq \alpha, \quad i = 1, \dots, m, \\ & \alpha \in [0, 1], \quad x \geq 0. \end{aligned} \quad (170)$$

Zodra de membership functions zijn gedefinieerd, verandert (170) dus in een deterministisch lineair programma. Dit is efficiënt op te lossen.

4.2.2 Driehoekige membership functie

Zoals in (166) te zien is, is de membership functie van middelbare leeftijd niet lineair, maar driehoekig. Deze paragraaf is gebaseerd op de methoden gebruikt in [49]. Neem in het geval van een driehoekige membership functie $\bar{b}_i$ en $\hat{b}_i$ als midden respectievelijk afwijking van b_i . De membership functie voor de constraints wordt dan:

$$\mu(A_i x) = \begin{cases} 0 & A_i x < \bar{b}_i - \hat{b}_i, \quad A_i x > \bar{b}_i + \hat{b}_i \\ 1 - \frac{\bar{b}_i - A_i x}{\hat{b}_i} & \bar{b}_i - \hat{b}_i \leq A_i x \leq \bar{b}_i \\ 1 + \frac{\bar{b}_i - A_i x}{\hat{b}_i} & \bar{b}_i \leq A_i x \leq \bar{b}_i + \hat{b}_i \\ 1 & A_i x = \bar{b}_i. \end{cases} \quad (171)$$

Op dezelfde manier kan de membership functie voor $c^T(\omega)x$ bepaald worden. Meer over driehoekige membership functies en het oplossen van het bijbehorende probleem is te vinden in [49].

5 Robust Optimization

Bij stochastisch programmeren wordt aangenomen dat de kansverdeling bekend is, maar dit is niet altijd het geval. In de realiteit zijn er ook problemen waar zeer weinig tot niets over de kansverdeling bekend is. Ook kan het voorkomen dat achteraf blijkt dat de aangenomen kansverdeling niet geheel overeenkomt met de realiteit, waardoor de gevonden oplossing nutteloos zou kunnen worden. Om dit soort problemen op te lossen is *Robust optimization* ontwikkeld. Robust optimization zoekt naar oplossingen die niet afhangen van de kansverdeling. Het houdt rekening met alle mogelijke uitkomsten van de stochastische variabelen en zoekt naar de beste oplossing die zo vaak mogelijk toegelaten is. Het enige wat dus bekend moet zijn, zijn de mogelijke waarden die de variabelen aan kunnen nemen. Naast de weinig vereiste kennis over het probleem heeft robust optimization nog een ander voordeel, namelijk dat het vaak met efficiënte algoritmen opgelost kan worden, waardoor de oplossing snel berekend kan worden. Een uitgebreider overzicht van de theorie van robust optimization staat beschreven in [6] en [8].

5.1 Worst case scenario

Met robust optimization wordt geprobeerd een oplossing te vinden die immuun is tegen de stochastiek. Dit houdt in dat de oplossing mogelijk moet blijven voor alle mogelijke uitkomsten van de stochastische variabelen. Dit wordt gedaan door alleen te kijken naar de slechtst mogelijke uitkomst en wordt er geprobeerd deze oplossing zo goed mogelijk te maken. Het eerste model daarvan is ontwikkeld door Soyster in 1973 en staat beschreven in [47]. Dit model zoekt de beste oplossing die altijd mogelijk is. Er wordt uitgegaan dat alle stochastiek in $A(\omega)$ zit en dat alle kolommen van $A(\omega)$ onafhankelijk zijn. Het probleem ziet er dan als volgt uit:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & A(\omega)x \leq b, \quad \forall a_j \\ & x \geq 0. \end{aligned} \tag{172}$$

Hierbij is a_j kolom j van $A(\omega)$. Dit systeem is equivalent aan het volgende systeem:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & A^* x \leq b, \\ & x \geq 0. \end{aligned} \tag{173}$$

Hier zijn alle $a_{ij}^* = \sup(a_{ij})$. Immers geldt dat als x toegelaten is voor het supremum, x ook toegelaten is voor alle mogelijke uitkomsten van A . Verder is A^* deterministisch, waardoor alle stochastiek verdwenen is en een deterministisch lineair programma overblijft. Dit is efficiënt op te lossen en de oplossing die dan verkregen wordt is de beste oplossing die altijd toegelaten is.

Dit systeem kan nog iets uitgebreid worden voor het geval dat de voorwaarde dat de kolommen onafhankelijk zijn, wordt losgelaten. Als alle constraints rijen zijn en de kolommen onafhankelijk, is het mogelijk overal het supremum van te nemen. Om het supremum te mogen nemen is het namelijk nodig dat alle coëfficiënten van een constrained onafhankelijk zijn. Omdat elke coëfficiënt uit een andere kolom komt, zijn ze onafhankelijk wanneer de kolommen afhankelijk zijn. Op dezelfde manier geldt dat als de kolommen dus afhankelijk zijn, er een constraint is die afhankelijke coëfficiënten heeft. Daarbij kan het dus voorkomen dat het supremum van alle coëfficiënten een slechter scenario dan het slechts mogelijke scenario is. Daarom moet bij afhankelijke kolommen rekening worden gehouden met alle mogelijke uitkomsten van een constraint en niet alleen het slechtste geval. Het veralgemeniseerde systeem ziet er als volgt uit:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b, \quad \forall a_i \ i = 1 \dots n, \\ & x \geq 0. \end{aligned} \tag{174}$$

Hier is a_i rij i van $A(\omega)$. Dus x moet toegelaten zijn voor alle mogelijke uitkomsten van de rijen van $A(\omega)$. (174) wordt een *robust counterpart* van een lineair programma genoemd, omdat deze alleen rekening houdt met het slechtste geval. Wanneer dit wordt opgelost, wordt namelijk de *robust solution* van het bijbehorende lineair programma gevonden. Dit is de oplossing die toegestaan is voor alle mogelijke uitkomsten van de stochastische variabelen en het beste slechtste geval heeft.

Deze manier van oplossen is de meest algemene manier van robust optimization. Het is namelijk altijd mogelijk een robust counterpart van een stochastisch lineair programma op te stellen, zolang alle stochastische variabelen zijn. De robust counterpart is over het algemeen efficiënt op te lossen.

5.2 Budget of uncertainty

Het probleem waar Robust Optimization vaak tegenaan loopt, is dat de gevonden oplossingen te conservatief zijn. Omdat er alleen rekening wordt gehouden met de slechtste uitkomst, wordt er vaak een oplossing gekozen die in vele gevallen een slechter resultaat geeft dan andere oplossingen en slechts in een paar gevallen een betere. Om dit probleem aan te pakken zijn er enkele methoden bedacht binnen Robust Optimization.

Eén van deze methoden heet *budget of uncertainty*. Deze methode zoekt naar een oplossing die rekening houdt met een kleiner gedeelte van de mogelijke uitkomsten van $A(\omega)$, in plaats van alle mogelijkheden. Deze methode staat beschreven in [11] [51]. Deze methode is ontworpen voor *conic optimization*, terwijl het probleem zoals beschreven in 1.1 een lineair programma is. Dit houdt in dat geldt $l \leq x \leq u$ in plaats van $x \geq 0$. Dit heeft echter redelijke invloed op de oplosmethode. Daarom zal eerst beschreven worden hoe deze methode werkt voor conic optimization en vervolgens zal de methode aangepast worden voor een lineair programma. Budget of uncertainty werkt als volgt:

De matrix $A(\omega)$ is stochastisch, dus bevat het elementen a_{ij} die stochastisch zijn. Hierbij is k_i het aantal stochastisch verdeelde a_{ij} van rij i . De a_{ij} worden gemodelleerd als begrensde, symmetrische en onafhankelijke variabelen, met mogelijke waarden uit het interval $[\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$, waarbij $\bar{a}_{ij}$ de nominale waarde van a_{ij} is en $\hat{a}_{ij} > 0$ de maximale afwijking. De relatieve afwijking wordt gedefinieerd als:

$$z_{ij} = \frac{a_{ij} - \bar{a}_{ij}}{\hat{a}_{ij}}, \quad |z_{ij}| \leq 1. \quad (175)$$

Hierbij geldt dat als a_{ij} zijn nominale waarde aanneemt, dan is $z_{ij} = 0$ en bij maximale afwijking is $|z_{ij}| = 1$. Dit is waar de budget of uncertainty Γ_i geïntroduceerd wordt. Er moet gelden:

$$\sum_{j=1}^{k_i} |z_{ij}| \leq \Gamma_i, \quad \forall i. \quad (176)$$

De keuze van Γ_i bepaalt dus de verschillende waarden van $A(\omega)$ waarmee rekening gehouden wordt. Bij $\Gamma_i = 0, \forall i$ zijn alle a_{ij} gelijk aan hun nominale waarde $\bar{a}_{ij}$ en heeft $A(\omega)$ dus maar 1 mogelijkheid, terwijl bij $\Gamma_i = k_i, \forall i$ wordt er rekening gehouden met alle mogelijkheden van $A(\omega)$. Met de keuze van het budget of uncertainty Γ_i kan er dus een afweging gemaakt worden tussen een betere oplossing en risico op een niet toegelaten oplossing.

Nu moet het alleen nog geïmplementeerd worden in het stochastisch lineair programma. Omdat de stochastiek alleen in $A(\omega)$ zit en alle a_{ij} een nominale waarde hebben, kan het systeem als volgt worden opgeschreven:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & \bar{a}_i x + \sum_j (\hat{a}_{ij} |x_j| z_{ij}) \leq b_i, \quad \forall i, \\ & l \leq x \leq u. \end{aligned} \quad (177)$$

Hier is $\bar{a}_i \in \mathbb{R}^n$ de nominale waarde van rij i . Op deze manier is de nominale waarde en de afwijking van elke a_{ij} gesplitst. Omdat het systeem bestand moet zijn tegen het slechtste geval mogelijk voor bepaalde Γ_i , moeten de z_{ij} slim gekozen worden. Daarvoor wordt de functie $\beta_i(x, \Gamma_i)$ geïntroduceerd:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & \bar{a}_i x + \beta_i(x, \Gamma_i) \leq b_i, \quad \forall i, \\ & l \leq x \leq u. \end{aligned} \quad (178)$$

De afwijking in het systeem is nu vervangen door de functie $\beta_i(x, \Gamma_i)$. Deze functie zoekt de maximale afwijking van de variabelen in rij i , bij gegeven x en Γ_i . Het probleem in (178) is dan ook een robust counterpart, omdat het alleen nog maar rekening houdt met het slechtste geval. De functie $\beta_i(x, \Gamma_i)$ is als volgt gedefinieerd:

$$\begin{aligned} \beta_i(x, \Gamma_i) = \max_z \quad & \sum_j \hat{a}_{ij} |x_j| z_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^k z_{ij} \leq \Gamma_i, \\ & 0 \leq z_{ij} \leq 1, \quad \forall i, j. \end{aligned} \quad (179)$$

Hierbij is $z = [z_{i1}, z_{i2}, \dots, z_{ik_i}]$. De functie $\beta_i(x, \Gamma_i)$ zoekt dus voor een gegeven x de grootste waarde van rij i van $A(\omega)$, onder het budget of uncertainty Γ_i . Als x toegelaten is voor dit maximum, is x immers toegelaten voor alle $A(\omega)$ mogelijk voor deze Γ_i . De oplossing zal altijd bestaan uit $\lfloor \Gamma_i \rfloor$ ¹⁰ variabelen met de waarde 1, één variabele met waarde $(\Gamma_i - \lfloor \Gamma_i \rfloor)$ en de rest 0, wat het uiteindelijke rekenwerk alleen maar makkelijker maakt.

Met het gebruik van sterke dualiteit is dit geheel om te schrijven naar een deterministisch lineair programma. Hiervoor is eerst de dual van (179) nodig. Deze ziet er als volgt uit:

$$\begin{aligned} \min \quad & \sum_j p_{ij} + \Gamma_i z_i \\ \text{s.t.} \quad & z_i + p_{ij} \geq \hat{a}_{ij} |x_j|, \quad \forall i, j, \\ & p_{ij} \geq 0, \quad \forall i, j, \\ & z_i \geq 0, \quad \forall i. \end{aligned} \quad (180)$$

Hier zijn p_{ij} en z_i de duale variabelen van (179). Omdat (179) toegelaten en begrensd is, is zijn duale dat ook, er geldt immers sterke dualiteit. Ook hebben ze hetzelfde optimum, dus kan dit systeem gesubstitueerd worden in (178). Dit geeft het volgende deterministisch lineair programma:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \bar{a}_i^T x + z_i \Gamma_i + \sum_j p_{ij} \leq b_i, \quad \forall i, \\ & z_i + p_{ij} \geq \hat{a}_{ij} y_j, \quad \forall i, j, \\ & -y_j \leq x \leq y_j \quad \forall j, \\ & l_j \leq x_j \leq u_j \quad \forall j, \\ & p_{ij} \geq 0, \quad \forall i, j, \\ & z_i \geq 0, \quad \forall i, \\ & y_j \geq 0, \quad \forall j. \end{aligned} \quad (181)$$

Het systeem wat overblijft is een deterministisch lineair programma, dat rekening houdt met de slechtste mogelijke uitkomst voor de gekozen Γ_i . Dit systeem is dus door keuze van Γ_i af te stellen op conservativiteit en toch efficiënt te berekenen.

Budget of uncertainty voor een lineair programma

Zoals opgemerkt in de vorige paragraaf, kan budget of uncertainty iets versimpeld worden, als de aanname $x \leq 0$ wordt gedaan in plaats van $l \leq x \leq u$. Om te beginnen kan het interval waaruit a_{ij} afkomstig is veranderd worden. Omdat alle $x > 0$, is het slechtste geval altijd dat de a_{ij} de waarde $\bar{a}_{ij} + \hat{a}_{ij}$ aanneemt. De onderste helft van het interval wordt nooit gebruikt. Daarom kan voor een lineair programma het inval met mogelijke waarden van a_{ij} gedefinieerd worden als $[\bar{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$.

Vervolgens heeft dit tot gevolg dat de relatieve afwijking z_{ij} niet langer negatief kan zijn. Dit verandert de definitie van z_{ij} als volgt:

$$z_{ij} = \frac{a_{ij} - \bar{a}_{ij}}{\hat{a}_{ij}}, \quad 0 \leq z_{ij} \leq 1. \quad (182)$$

Er hoeft dus niet langer de absolute waarde van z_{ij} genomen te worden. De verdere afleiding is hetzelfde als in de de vorige paragraaf, met het verschil dat in (176), (178), (179) en (180) niet langer de absolute waarde genomen hoeft te worden, en de voorwaarde $l \leq x \leq u$ vervangen wordt door $x \geq 0$. Het uiteindelijke deterministisch lineair programma ziet er dan als volgt uit:

¹⁰ $\lfloor x \rfloor = \max(m \in \mathbb{Z} | m \leq x)$

$$\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & \bar{a}_i^T x + z_i \Gamma_i + \sum_j p_{ij} \leq b_i, \quad \forall i, \\
& z_i + p_{ij} \geq \hat{a}_{ij} x_j, \quad \forall i, j, \\
& p_{ij} \geq 0, \quad \forall i, j, \\
& z_i \geq 0, \quad \forall i, \\
& x_j \geq 0, \quad \forall j
\end{aligned} \tag{183}$$

Hierbij is (183) een versimpelde versie van (181). Dit systeem blijft door keuze van Γ_i af te stellen op conservativiteit en toch efficiënt te berekenen.

6 Stability and Sensitivity Analysis

Volgens [6] en [51] valt het oplossen van een stochastisch lineair programma uiteen in drie hoofdrichtingen, Stochastisch Programmeren, Robust Optimization en tot slot Stability and Sensitivity Analysis. In dit hoofdstuk zal Stability and Sensitivity Analysis behandeld worden.

Waar Stochastisch Programmeren en Robust Optimization gefocust is op het vinden van een strategie om het stochastisch lineair programma op te lossen, gaat het bij Stability and Sensitivity Analysis over het analyseren van de gebruikte methoden. Bij een stochastisch lineair programma worden aannames gedaan over de onzekere informatie. Bij Robust Optimization worden geen aannames gedaan over de kansverdeling van $A(\omega)$, $b(\omega)$ en $c(\omega)$. Deze methode wordt dan ook niet behandeld bij Stability and Sensitivity Analysis.

Stability and Sensitivity Analysis kijkt of de aannames over de onzekere informatie na het toepassen van een methode kloppen en of de aannames beter zouden kunnen om de methoden te kunnen verbeteren.

Bij een stochastisch lineair programma worden aannames gedaan over de onzekere informatie, ofwel de stochastiek in de parameters $A(\omega)$, $b(\omega)$ en $c(\omega)$. Hoe goed een oplosmethode werkt hangt, volgens [24], sterk samen met hoe goed deze aannames zijn. De twee hoofdrichtingen van de Stability and Sensitivity Analysis zijn de gevoeligheidsanalyse van de kansverdeling en de statische analyse van de kansverdeling.

Omdat Stability and Sensitivity Analysis bij het stochastisch lineair programma kijkt naar het toepassen van een methode, zal in dit hoofdstuk alleen een overzicht gegeven worden van relevante literatuur over Stability and Sensitivity Analysis ter ondersteuning van oplosmethode in dit verslag.

6.1 Analyse van de kansverdeling

De kansverdeling van $A(\omega)$, $b(\omega)$ en $c(\omega)$ zal in veel gevallen niet (volledig) bekend zijn en geschat worden. In deze schatting en daarbij gemaakte aannames zit een fout, hoe deze fout het stochastisch lineair programma beïnvloed behandeld [41].

Meer over de stabiliteit en gevoeligheid van verschillende oplosmethoden voor een stochastisch lineair programma met betrekking tot de kansverdeling kan gevonden worden in:

Methode	Literatuur	Paragraaf
Stochastische programmeren met recourse	[20], [21], [26], [53] en [18]	(3.1)
Scenario-analyse	[39]	(3.2)
Chance constrainend programming	[22]	(3.4)

Tabel 6: Literatuuroverzicht stabiliteit en gevoeligheid met betrekking tot de kansverdeling.

6.2 Statistische analyse

Een andere manier om de stabiliteit en gevoeligheid van een oplosmethode voor stochastisch lineair programma te analyseren, is statische analyse van de kansverdeling. Gebaseerd op empirische data van het stochastisch lineair programma kan een schatter gemaakt worden voor de kansverdeling, hoe goed deze schatter is, kan geanalyseerd worden met begrippen uit de statistiek.

Dit wordt behandeld in:

Onderwerp	Literatuur
Optimale oplossing	[23], [30] en [54]
Betrouwbaarheidsintervallen	[32] en [45]

Tabel 7: Literatuuroverzicht statische analyse van de kansverdeling van de parameters van een stochastisch lineair programma.

7 Conclusie

In dit onderzoek is gekeken naar de mogelijkheden om een stochastisch lineair programma te analyseren en op te lossen. In dit hoofdstuk zullen de vragen geformuleerd in de probleembeschrijving beantwoord worden.

1. Wat betekent oplosbaarheid en optimaliteit voor een stochastisch lineair programma?

Of een oplossing x toegelaten is, is verschillend per realisatie van het stochastisch lineair programma. Om hier toch uitspraken over te kunnen doen, is in hoofdstuk 2 allereerst gekeken naar manieren om de kans op toelaatbaarheid van een oplossing x te bepalen.

Naast het bepalen van de kans op toelaatbaarheid, is er ook gekeken naar betrouwbaarheidsintervallen voor toegelaten waarden van de oplossing x en het opstellen van betrouwbaarheidsintervallen die aangeven of x in $(1 - \alpha)\%$ van de realisaties toegelaten is.

Ook is onderzocht hoe de kans op optimaliteit berekend kan worden en is deze methode gebruikt om een betrouwbaarheidsinterval op te stellen voor de waarde van de doelfunctie. Deze betrouwbaarheidsintervallen zeggen nog niks over optimaliteit, ze zeggen enkel in welk gebied de doelfunctiewaarde in $(1 - \alpha)\%$ van de gevallen zal liggen.

Oplosbaarheid en optimaliteit van een oplossing x voor een stochastisch lineair programma betekent dat voor $(1 - \alpha)\%$ van de realisaties een oplossing toegelaten en/of optimaal kan zijn.

2. Wat voor technieken zijn er om een stochastisch lineair programma op te lossen?

De drie meest bekende manieren om een stochastisch lineair programma te lossen zijn Stochastic Programming, Robust Optimization en Stability and Sensitivity Analysis.

Stochastic Programming maakt gebruik van kennis over de kansverdeling van de realisaties van een stochastisch lineair programma. De bekendste oplosmethoden maken gebruik van recourse, een tweede actie om niet-toelaatbaarheid van een oplossing te compenseren als gevolg van de stochastiek. Allereerst is er gekeken naar scenario-analyse, waarbij voor elk van het eindige aantal scenario's de optimale recourse wordt bepaald. Als het aantal scenario's groot wordt kan gebruik worden gemaakt van Sample Average Approximation, waarbij gebruik wordt gemaakt van een steekproef onder de mogelijke scenario's. Dit wordt ook gebruikt bij chance constrained programming, waarbij gezegd wordt dat aan een constraint voldaan moet worden in $(1 - \alpha)\%$ van de realisaties.

Robust Optimization doet bijna geen enkele aanname over de kansverdeling van de realisaties van het stochastisch lineair programma, er wordt enkel gebruik gemaakt van het feit dat de realisaties in een begrensde interval liggen. Robust Optimization zoekt een oplossing die in elk van de scenario's toelaatbaar zal zijn, het nadeel hiervan is dat dit soms onnodig hoge 'optima' geeft. Daarom wordt ook gekeken naar een methode die gebruik maakt van budget of uncertainty, waarbij een vooraf te kiezen risico wordt genomen om zo betere oplossingen te vinden.

Stability and Sensitivity Analysis is erg beknopt behandeld, omdat dit zich vooral bezig houdt met het analyseren van de uitkomsten gevonden door een oplosmethode voor een stochastisch lineair programma. Bij Stability and Sensitivity Analysis wordt gekeken naar de schatting en aannames die gedaan worden over de kansverdeling van de realisaties en het effect hiervan op de oplossing gegeven door *Stochastic Programming*.

Tot slot wordt *Fuzzy Programming* nog kort toegelicht in het verslag, om om te kunnen gaan met situaties waarin de formulering van het stochastisch lineair programma met stochastische variabelen niet eenduidig is.

3. In welke situaties is welke techniek aan te raden?

De vraag welke oplostechniek in welke situatie het meest aan te raden is, hangt erg sterk samen met hoeveel betrouwbare informatie bekend is over de kansverdeling van de stochastische variabelen. Als er betrouwbare informatie bekend is over de kansverdeling van de realisaties, verdient *Stochastic Programming* de voorkeur boven *Robust Optimization*, omdat gebruik maken van de kennis van deze verdeling een beter resultaat geeft.

Het verschil tussen *Stochastic Programming* met recourse en chance constrained programming, is de formulering van het probleem. *Stochastic Programming* met recourse zoekt naar een recourse voor elke realisatie bij een gekozen x die naar verwachting de laagste kosten geeft. Het moet wel mogelijk zijn om een tweede, compenserende actie uit te voeren. Chance constrained programming versoepelt de constraints van het stochastisch lineair programma door te zeggen dat deze met kans $(1 - \alpha)$ moeten gelden.

Stochastic Programming accepteert dat een gekozen oplossing x niet voor alle realisaties ω toelaatbaar zal zijn. Dit wordt, indien mogelijk, opgelost door recourse of door het versoepelen van de constraints, zoals bij chance constrained programming. Moet echter voor elke realisaties aan de constraints voldaan worden, dan is *Robust Optimizat*on een betere aanpak.

Het is ook aan te raden om *Robust Optimization* te gebruiken wanneer er weinig tot geen informatie is over de kansverdeling. Hoe kleiner het interval waarin de realisaties liggen, hoe beter *Robust Optimization* zal werken. Als dit interval te groot wordt, gaat de onbekende kansverdeling een te grote rol spelen. Het kan immers best zo zijn dat de uiterst rechtste realisaties nauwelijks voorkomen en toch houdt *Robust Optimization* dit dan aan als grens. Hiervoor kan budget of uncertainty ingezet worden.

Stability and Sensitivity Analysis kan gebruikt worden om te bepalen of de beschikbare kennis van de kansverdeling van de realisaties al dan niet betrouwbaar was bij het gebruik maken van *Stochastic Programming*. Indien blijkt dat de beschikbare kennis niet betrouwbaar was, is het aan te raden om gebruik te maken van *Robust Optimization*.

Als deze kennis wel betrouwbaar was, kunnen de technieken uit hoofdstuk 2 erg goed van pas komen. De kans op toelaatbaarheid en optimaliteit van een oplossing kunnen dan bepaald worden. Ook kunnen er betrouwbaarheidsintervallen opgesteld worden voor de toelaatbaarheid van een oplossing en de waarde van de doelfunctie voor deze oplossing.

Of het mogelijk is om deze betrouwbaarheidsintervallen te bepalen en deze kansen te berekenen, hangt volledig af van de kansverdeling van de realisaties. Deze moeten aan de voorwaarden voldoen (identiek verdeeld, onafhankelijk, etc.) en deze kennis moet bekend zijn.

Samengevat geeft dit de volgende adviezen:

1. Gebruik *Stochastic Programming* als er betrouwbare informatie is over de kansverdeling van de realisaties.
 - Gebruik scenario-analyse in het geval van een eindig en niet te groot aantal scenario's.
 - Gebruik Sample Average Approximation om bij een groot of oneindig aantal scenario's een steekproef te doen.
 - Gebruik recourse als het mogelijk is om het probleem zo te formuleren, dat er sprake is van een tweede actie.
 - Als een probleem niet geformuleerd kan worden voor recourse, gebruik dan chance constrained programming.
2. Gebruik *Robust Optimization* als er geen informatie is over de kansverdeling van de realisaties en dat de gevonden oplossing moet gelden voor elke realisatie.
 - Gebruik Budget of uncertainty als het nemen van wat risico voor het probleem geen grote gevolgen heeft. Het genomen risico wordt beloond met betere oplossingen.
3. Gebruik *Stability and Sensitivity Analysis* om te controleren of *Stochastic Programming* werkt. Is de schatting van de informatie over de kansverdeling van de realisaties toch niet betrouwbaar genoeg, gebruik dan *Robust Optimization*.
4. Gebruik de technieken uit hoofdstuk 2 om het probleem te analyseren, indien de juiste informatie beschikbaar is. Dit geeft een erg goed beeld over de invloed van de stochastiek en een beeld over de oplossing.

8 Discussie

In dit onderzoek zijn een aantal zaken beknopt behandeld. Zo zou meer aandacht voor Stability and Sensitivity Analysis en Fuzzy Programming wellicht tot nieuwe inzichten kunnen leiden.

Het duale probleem is ook niet geanalyseerd in de context van een stochastisch lineair programma, terwijl dit wel veel gebruikt wordt bij het oplossen van een deterministisch lineair programma. Verder kan er nog gekeken worden naar een Integer stochastisch lineair programma.

In het geval van de oplosmethoden, zou het interessant kunnen zijn om te kijken wat in de praktijk beter werkt als er veel informatie bekend is over de kansverdeling van de realisaties. Stochastic Programming of Robust Optimization met als interval de betrouwbaarheidsintervallen gevonden met behulp van deze informatie.

De focus bij de analyse van een stochastisch lineair programma in dit onderzoek lag vooral bij het opstellen van betrouwbaarheidsintervallen voor de toelaatbaarheid van een oplossing, waarde van de oplossing en doelfunctiewaarde. Dit zou ook uitgebreid kunnen worden naar het berekenen van de kansen op toelaatbaarheid en optimaliteit in een bredere setting. Verder is het betrouwbaarheidsinterval voor de waarde van de oplossing slechts een schatting, waarbij gebruikt wordt gemaakt van normaal verdeelde elementen. Als het mogelijk is om de theorie uit 2.3 verder uit te breiden kan er nog meer kennis verzameld worden over een stochastisch lineair programma.

Budget of uncertainty versus chance constrained programming

Tot slot is het interessant op het verschil tussen budget of uncertainty bij Robust Optimization en chance constrained programming bij Stochastic Programming te bekijken, omdat deze twee methoden op dezelfde soort problemen kunnen worden toegepast. Bij beide methoden wordt een bepaald risico genomen op ontoelaatbaarheid van de oplossing x . Bij chance constrained programming wordt met een bepaalde kans toegestaan dat de oplossing niet toelaatbaar is en worden vervolgens de verwachte kosten geminimaliseerd door middel van een steekproef uit de scenario's. Doordat er een voldoende grote steekproef wordt genomen, wordt de kansverdeling benaderd. Bij budget of uncertainty wordt de oplossingsruimte vergroot door het interval van bepaalde stochastische variabelen kleiner te maken, waardoor er meer en dus waarschijnlijk betere oplossingen mogelijk zijn.

Zo hebben beide methoden voor- en nadelen. Chance constrained programming is op zichzelf erg moeilijk te berekenen en wordt daarom opgelost met een steekproef. Dit geeft dan uiteraard slechts een benadering van de oplossing. Ook moet de steekproef zelf voldoende groot zijn, wat dan weer zorgt dat er in het benaderingsprobleem relatief veel constraints zijn. Verder geeft het wel een goed beeld van de kans op ontoelaatbaarheid. Budget of uncertainty daarentegen, geeft een deterministisch lineair programma met relatief weinig met constraints. Echter zegt deze methode weinig over de kans op de scenario's die worden weggelaten. Het zou kunnen dat deze methode in 2% of 20% van de gevallen een niet toegelaten oplossing geeft.

Ons advies zou zijn om budget of uncertainty te gebruiken, als de kans op ontoelaatbaarheid erg klein is, zodat het aantal weggelaten scenario ook klein is. Hiermee wordt de kans dat veel voorkomende scenario's worden weggelaten op zijn beurt ook klein. Wanneer de kans op toelaatbaarheid groter wordt, zouden wij chance constrained programming aanraden, omdat deze een veel beter beeld geeft van de kans op ontoelaatbaarheid.

9 Symbolenlijst

Symbol	Definitie
$A(\omega)$	De coëfficiënten-matrix uit het stochastisch lineair programma (2)
A	Deterministisch gedeelte van $A(\omega)$
$\hat{A}$	Stochastisch gedeelte van $A(\omega)$
A^*	Aangepaste matrix A .
A_B	De matrix bestaand uit de kolommen van A horend bij de basisvariabelen x_B
A_N	De matrix bestaand uit de kolommen van A horend bij de niet-basisvariabelen x_N
a_{ij}	Het element op rij i en kolom j van matrix A
$\bar{a}_{ij}$	De nominale waarde van a_{ij}
$\hat{a}_{ij}$	De afwijking van a_{ij} van zijn nominale waarde
a_i	Rij i van matrix A
a_j	Kolom j van matrix A
$b(\omega)$	Right hand side stochastisch lineair programma (2)
b	Deterministisch gedeelte van $b(\omega)$
$\hat{b}$	Stochastisch gedeelte van $b(\omega)$
b^*	Aangepaste vector b
$c(\omega)$	Doelfunctiecoëfficiënten van het stochastisch lineair programma (2)
c	Deterministisch gedeelte van $c(\omega)$
$\hat{c}$	Stochastisch gedeelte van $c(\omega)$
c^8	Aangepaste vector c
c_B	De coëfficiënten van c horend bij de basisvariabelen x_B
c_N	De coëfficiënten van c horend bij de niet-basisvariabelen x_N
$\text{cov}(\cdot)$	Covariantie
D^k	$m \times m$ matrix waarbij kolom k is vervangen door een zekere kolomvector
$D(\cdot), N(\cdot)$	stochastische functies
$E(\cdot)$	Verwachtingswaarde
F	Een Fuzzy verzameling
$F(\cdot)$	Cumulatieve verdelingsfunctie
$F_\alpha(k, \phi)$	Grenswaarden van een $(1 - \alpha)$ -betrouwbaarheidsinterval van een F-verdeelde variable met vrijheidsgraden k en ϕ
$f(\cdot)$	Kansdichtheidfunctie
i, j, k, r	Index
k, ϕ	Vrijheidsgraden
k_i	Het aantal stochastische variabelen in rij i van $A(\omega)$
M	De interactie tussen de originele beslissing x en de recourse y
m, n	Lengte vector/matrix
p_i	De kans op scenario i
$P(\cdot)$	De kans op $\cdot$
$Q(x)$	De verwachting van de recourse kosten onder de constraints
q	De kostenfunctie van de recourse y
S	Simplex multiplier behorende bij een oplossing
S^2	Schatting voor σ^2
t	Standaard normaal verdeelde variabelen
U	Universele verzameling
V	Factor in covariantie matrix van $b(\omega)$
$\text{var}(\cdot)$	Variantie
U, X, Y	Stochastische variabelen
x	Oplossingsvector stochastisch lineair programma
x_B	Vector van lengte m met alle variabele uit x die in de basis vormen
x_N	Vector van lengte $n - m$ met alle variabele uit x die niet in de basis zitten
y	De recourse na de originele beslissing x
Z	Het quotiënt van $N(\cdot)$ en $D(\cdot)$
z	Uitkomst van $c^T x$
z_{ij}	Relatieve afwijking van a_{ij}

Symbool	Definitie
α	Overschrijdingskans
β	Verwachtingswaarde van $D(\cdot)$
$\beta_i(x, \Gamma_i)$	De functie die de mogelijke afwijkingen van a_{ij} modelleert
β_{ij}	Cofactor van B_{ij}
γ	Grenswaarde betrouwbaarheidsinterval
$\delta(\omega)$	Stochastische variabele
Δ, Σ	Factor in covariantiematrix
Γ_i	Budget of uncertainty van rij i .
θ	Een bovengrens voor de functie $Q(x)$: $\theta \geq Q(x)$
λ	Constante
σ	Standaardafwijking, Simplex multiplier behorende bij een oplossing
σ^2, τ^2	Variantie
Ω	De verzameling van alle scenario's ω
ω	Een scenario, waarin de waarden van c , q , A , M en b zijn vastgelegd
v	Verwachtingswaarde van $N(\cdot)$
$v(\cdot)$	Waarde op diagonaal van $V\sigma^2$ of $\Sigma\sigma^2$

10 Engelse begrippenlijst

Budget of uncertainty

Het gedeelte van de mogelijke oplossingen waar de oplossingsmethode rekening mee houdt, 41

Chance constraint

Een constraint die met een bepaalde kans $(1 - \alpha)$ moet gelden, 34

Constraint

Voorwaarde waar de oplossingsvector x aan moet voldoen, om zo het oplossingsgebied te definiëren, 8

Error

Afwijking of fout in waarde, 7

Feasibility cut

Een extra constraint om een oplossing dat niet toegelaten is uit te sluiten, 31

Fuzzy number

Een Fuzzy number is een generalisatie van een normaal, reëel getal. Een Fuzzy number heeft niet één specifieke waarde, maar is een verzameling met mogelijke waarden die het getal aan kan nemen, meestal tussen 0 en 1, 38

Fuzzy verzameling

Een Fuzzy verzameling is een verzameling gedefinieerd door *membership functions*, 38

Optimality cut

Een extra constraint om een oplossing dat niet optimaal is uit te sluiten, 31

Recourse

Een recourse is een tweede actie om te zorgen dat er alsnog aan de voorwaarden wordt voldaan., 30

Robust counterpart

De tegenhanger van een bepaald lineair programma, die alleen rekening houdt met het slechte geval van dat lineair programma, 40

Robust optimization

Een oplossingsmethode waarbij de kansverdeling van de stochastische variabelen niet bekend hoeven te zijn, 40

Robust solution

De oplossing van een de robust counterpart van een lineair programma, 40

Sample Average Approximation

Een methode waarmee er via Monte Carlo Sampling een aantal scenario's wordt geselecteerd, om daar vervolgens de oplossing op te baseren, 34

Supporting hyperplane

Een 'separating hyperplane' is een hyperplane dat een gesloten, convexe verzameling C scheidt van een punt x_0 . Een separating hyperplane heet 'supporting' als deze hyperplane een punt $x_0 \in C$ van C scheidt, 32

Triangular array

Onderling onafhankelijke rij $X_{n1}, \dots, X_{nr_n}$ waarbij $n = 1, \dots$ steeds staat voor een andere verdeling. Van elke verdeling n zijn er r_n elementen, 27

Referenties

- [1] Adams, R.A., Essex, C. (2010). *Calculus, a complete course* seventh edition, p.814.
- [2] Alghalith, M. (2016). *Novel and simple non-parametric methode of estimating the joint and marginal densities* Department of Economics, University of the West Indies, St Augustine, Trinidad and Tobago, Physica A 454 (2016) 94-96.
- [3] Ahmed, S. (2010). *Two-Stage Stochastic Integer Programming: A Brief Introduction*. Wiley Encyclopedia of Operations Research and Management Science.
- [4] Avrachenkov, K.E. (1999). *Analytic perturbation theory and its applications*.
- [5] Babbar, M. M. (1955). *Distributions of Solutions of a Set of Linear Equations (with an Application to Linear Programming)*. Journal of the American Statistical Association, 50(27), 854-869.
- [6] Ben-Tal, A., Ghaoui, L. E., & Nemirovskiĭ, A. S. (2009). *Robust optimization*. Princeton: Princeton University Press.
- [7] Berk, Kenneth (1973). *A Central Limit Theorem for m-Dependent Random Variables with Unbounded m* Ann. Probab. 1 (1973), no. 2, 352-354. doi:10.1214/aop/1176996992. <http://projecteuclid.org/euclid.aop/1176996992>.
- [8] Bertsimas, D., Brown, D.B., Caramanis C. (2007). *Theory and application of robust optimization* SIAM Review, 53 (3) (2007), pp. 464-501.
- [9] Bertsimas, D., Thiele, A. (2006). *A robust optimization approach to inventory theory*, Oper. Res. 54 (January(1)), 150-168.
- [10] Bertsimas, D., Sim, M. (2003). *Robust discrete optimization and network flows*, Math. Program.98 (September(1-3)), 49-71.
- [11] Bertsimas, D., Sim, M., (2004). *The price of robustness*, Oper. Res.52(1), 35-53.
- [12] Billingsley, P (1995). *Probability and Measure* Wiley series in probability -and mathematical statistics. Probability and mathematical statistics, p.359-365.
- [13] Birge, J. R., & Louveaux, F. (1997). *Introduction to Stochastic Programming*, New York: Springer-Verlag, 155-163.
- [14] Bose, R.C., Roy, S.N. (1953). *Simultaneous confidence interval estimation* Ann. Math. Statist. 13, 322.
- [15] Box, G., & Hunter, J. (1954). *A Confidence Region for the Solution of a Set of Simultaneous Equations with an Application to Experimental Design*. Biometrika, 41(1), 190-199.
- [16] Craig, C.C., (1929). *The frequency function of y/x* Annals of Mathematics 30 (1929), 471-86.
- [17] Dantzig, G.B., (1965). *Linear programming and Extensions* Princeton University Press.
- [18] Dantzig, G.B. & Glynn, P.W. (1989). *Parallel processors for planning under uncertainty* 5th Int.Conf. on Stochastic Programming, Ann Arbor (August 13-18, 1989).
- [19] Dantzig, G. B., & Infanger, G. (1993). *Multi-stage stochastic linear programs for portfolio optimization* Annals of Operations Research, 45(1), 59-76.
- [20] Dupačová, J. (1984). *Stability in stochastic programming with recourse-estimated parameters* Mathematical Programming 28 (1984) 72-83.
- [21] Dupačová, J. (1986). *Stability in stochastic programming with recourse. Contaminated distributions* Mathematical Programming Study 27 (1986) 133-144.
- [22] Dupačová, J. (1986). *Stability in stochastic programming-probabilistic constraints* in: V.I. Arkin, A. Shiriaev and R. Wets, eds., Proceedings of the International Conference on Stochastic Optimization, Kiev. Lecture Notes in Control and Information Sciences No. 81 (Springer, Berlin, 1986) pp. 314-325.

- [23] Dupačová, J. (1989). *On statistical sensitivity analysis in stochastic programming* 5th Int. Conf. on Stochastic Programming, Ann Arbor (August 13-18, 1989).
- [24] Dupačová, J. (1990). *STABILITY AND SENSITIVITY-ANALYSIS FOR STOCHASTIC PROGRAMMING* Annals of Operations Research 27 (1990) 115-142.
- [25] Fieller, E.C. (1940). *The biological standardization of insulin* J.R. Statist. Soc. Suppl. 7,1.
- [26] Garstka, G.J. (1974). *Distribution functions in stochastic programs with recourse: A parametric analysis* Mathematical Programming 6 (1974) 339-351.
- [27] Geary, R.C. (1930). *The frequency distribution of the quotient of two normal Variates* Journal of the Royal Statistical Society, 93 (1930), 422-6.
- [28] Jensen, P. A., & Bard, J. F. (2003). *Stochastic Optimization. In Operations research: Models and methods.* Hoboken, NJ: Wiley.
- [29] Kaminski, M. (2007). *Central limit theorem for certain classes of dependent random variables* Theory probab. appli. vol.51, no.02, pp.335-342.
- [30] Kaňková, V. (1977). *Optimum solution of a stochastic optimization problem with unknown parameters* Trans. 7th Prague Conf. 1974 (Academia, Prague, 1977) pp. 239-244.
- [31] Kim, S., Pasupathy, R., & Henderson, S. G. (2014). *A Guide to Sample Average Approximation* Handbook of Simulation Optimization International Series in Operations Research & Management Science, 207-243.
- [32] King, A.J. (1988). *Asymptotic distributions for solutions in stochastic optimization and generalized M-estimation* WP-88-58, IIASA, Laxenburg.
- [33] Klir, G., Yuan, B. (1995). *Fuzzy sets and fuzzy logic (Vol. 4)* New Jersey: Prentice hall.
- [34] Kuelbs, J., Philipp, W. (1980). *Almost sure invariance principles for sums of mixing B-valued random variables* Ann. Probab. 8 (1980), pp.1003-1036.
- [35] Lomnicki, Z.A. (1967). *On the Distribution of Products of Random Variables* Journal of the Royal Statistical Society Series B (Methodological) Vol. 29, No3, pp. 513-52.
- [36] Lonseth, A.T., (1942). *Systems of linear equations with coefficients subject to error* Annals of Mathematical Statistics, 13 (1942), 33-27.
- [37] Luedtke, J., & Ahmed, S. (2008). *A Sample Approximation Approach for Optimization with Probabilistic Constraints*. SIAM J. Optim. SIAM Journal on Optimization, 19(2), 674-699.
- [38] Pagnoncelli, B. K., Ahmed, S., & Shapiro, A. (2009). *Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications*. Journal of Optimization Theory and Applications, 142(2), 399-416.
- [39] Robinson, S.M. & Wets, R.J.B. (1988). *Stability in two-stage stochastic programming* SIAM J. Control Optim. 25 (1988) 1409-1416.
- [40] Rohatgi, V.K. (1976). *An introduction to Probability Theory Mathematical Statistics* Wiley, New York, p.141.
- [41] Römich, W & Schultz, R. (1987). *Distribution sensitivity in stochastic programming* Mathematical Programming 50 (1991) 197-226.
- [42] Ross, S. M. (1972). *Introduction to probability models*. New York: Academic Press, 80.
- [43] Rosenblatt, M. (1956). *A central limit theorem and strong mixing conditions* Proc. Nat. Acad. Sci. USA 42 (1956) pp.245-250.
- [44] Ruszczyński, A. P., & Shapiro, A. (2003). *Stochastic programming*, Amsterdam: Elsevier.
- [45] Shapiro, A. (1989). *Asymptotic properties of statistical estimators in stochastic programming* Research Report 55/87 (17), University of South Africa, Ann. Statist. 17 (1989) 841-858.

- [46] Springer, M.D. (1979). *The Algebra of Random Variables* Wiley, New York.
- [47] Soyster, A. L. (1973). *Convex programming with set-inclusive constraints and applications to inexact linear programming*, Oper. Res. 21 1154-1157.
- [48] Still, G. (2015). *Mathematical Optimization*, Enschede, Netherlands: Department of Applied Mathematics.
- [49] Tanaka, H., Guo, P., & Zimmermann, H. J. (2000). *Possibility distributions of fuzzy decision variables obtained from possibilistic linear programming problems*. Fuzzy Sets and Systems, 113(2), 323-332.
- [50] Tsakiris, G. & Spiliotis, M. (2004). *Fuzzy linear programming for problems of water allocation under uncertainty*. European Water, 7(8), 25-37.
- [51] Varas, M., Maturana, S., Pascual, R., Vargas, I., & Vera, J. (2014). *Scheduling production for a sawmill: A robust optimization approach*. International Journal of Production Economics, 150, 37-51.
- [52] Velde van der, M. (2013). *Analyzing the solution of a Linear Program*, Enschede, Netherlands: Thesis Applied Mathematics Faculty of Electrical Engineering, Mathematics and Computer Science.
- [53] Wang, J. (1985). *Distribution sensitivity analysis for stochastic programs with complete recourse* Mathematical Programming 31 (1985) 286-297.
- [54] Wets, R.J.B. (1979). *Wets, A statistical approach to the solution of stochastic programs with (convex) simple recourse* Working Paper, University of Kentucky.