Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

Het ellipsoïdealgoritme en een toepassing daarvan op de Lovász ϑ functie.

(Engelse titel: The ellipsoid method and an application to the Lovász ϑ function.)

Verslag ten behoeve van het
Delft Institute of Applied Mathematics
als onderdeel ter verkrijging

van de graad van

BACHELOR OF SCIENCE
in
TECHNISCHE WISKUNDE

door

Sander Gribling

Delft, Nederland
Juli 2013

BSc verslag TECHNISCHE WISKUNDE

"Het ellipsoïdealgoritme en een toepassing daarvan op de Lovász ϑ functie."

(Engelse titel: "The ellipsoid method and an application to the Lovász ϑ function.")

Sander Gribling

Technische Universiteit Delft

Begeleider

Prof. Dr. F. Vallentin

Overige commissieleden

Dr. ing. D. Jeltsema

Dr. M. C. Veraar

Dr. J. G. Spandaw

Juli, 2013

Delft

Samenvatting

In dit onderzoek is het ellipsoïdealgoritme bestudeerd. Het ellipsoïdealgoritme is een algoritme dat het optimum van een lineaire doelfunctie over een gegeven deelverzameling van $\mathbb{R}^n$ tot op een epsilon nauwkeurig kan bepalen. De gegeven verzameling moet hierbij compact, convex en volledig dimensionaal zijn.
Dit algoritme kan in polynomiale tijd uitgevoerd worden. Daarom is het dan ook van groot theoretisch belang gebleken; zeer veel problemen zijn te schrijven in een vorm die oplosbaar is met het ellipsoïdealgoritme.
In dit verslag wordt een bewijs van de correctheid van het ellipsoïdealgoritme gegeven, het bewijs dat behandeld wordt is oorspronkelijk gegeven door Grötschel, Lovász en Schrijver in [4].
Vervolgens zal het belang van het algoritme gedemonstreerd door voor een aantal problemen te laten zien dat ze te schrijven zijn in een vorm die oplosbaar is met het ellipsoïdealgoritme. Allereerst bekijken we een 'eenvoudige' klasse van compacte, convexe verzamelingen: polytopen. Hierbij zal ook de vergelijking worden getrokken met het welbekende SIMPLEX algoritme. Vervolgens zal een veel interessanter probleem beschouwd worden: de Lovász ϑ functie. De Lovász ϑ functie is een functie die aan een graaf een getal toekent. Dit getal kan van grote waarde zijn aangezien het een bovengrens op de Shannon-capaciteit van een graaf geeft. De Shannon-capaciteit van een graaf is over het algemeen erg lastig te berekenen. Dit probleem zal in veel detail worden behandeld, aangezien ook de implementatie van het onderdeel van dit project is. Tot slot zal er dan ook nog aandacht besteed worden aan deze implementatie, in het bijzonder aan de problemen die hierbij ontstonden en de oplossing daarvan. Het grootste probleem bleek zoals verwacht de enorme precisie die in theorie vereist is.

Voorwoord

Dit verslag is het eindproduct van mijn bachelorproject ter afsluiting van de bachelor Technische Wiskunde aan de Technische Universiteit Delft.
Gedurende de bachelor heb ik een groot aantal onderwerpen gezien, een flink aantal daarvan vond ik erg interessant. Waaronder de vakken in de leerlijn Optimalisering. Ik was dan ook op zoek naar een project wat deels theoretisch en deels toegepast was. Het ellipsoïdealgoritme is daarvoor een goed onderwerp gebleken. Het is niet alleen theoretisch interessant en belangrijk, het algoritme was ook nog (bijna) nooit geïmplementeerd. Een mooie uitdaging dus. Zodoende is het onderwerp van mijn bachelorproject dan ook geworden: "Het ellipsoïdealgoritme en een toepassing daarvan op de Lovász ϑ functie".
Allereerst wil ik mijn begeleider Prof. Dr. F. Vallentin bedanken voor de introductie tot dit onderwerp en de verdere begeleiding tijdens het project. Daarnaast wil ik ook de rest van de aanwezigen tijdens het seminar bedanken voor hun bijdrage.

Inhoudsopgave

1. Inleiding

In dit project is het ellipsoïdealgoritme bestudeerd. De invoer van het algoritme is een compacte, convexe deelverzameling van $\mathbb{R}^n$ en een lineaire doelfunctie. De uitvoer is het optimum van de doelfunctie over de verzameling tot op een epsilon nauwkeurig. Epsilon kan hierbij vrij gekozen worden in $\mathbb{R}_{>0}$. Het bewijs van de correctheid van het algoritme is meerdere malen gegeven, onder andere door Grötschel, Lovász en Schrijver in [4]. Hun bewijs zal hier uitgebreid behandeld worden. Dit bewijs bestaat grofweg uit drie delen:

(1) Aantonen dat het algoritme in elke stap een ellips oplevert.
(2) De ellips in stap $k+1$ moet het 'goede' deel van de ellips in stap k bevatten.
(3) De door het algoritme gegeven oplossing voldoet aan de gewenste eigenschappen.

Het laatste deel vereist uiteraard het meeste werk. Grötschel, Lovász en Schrijver waren namelijk niet er uitgebreid in hun argumentatie. Het eerste deel van dit onderzoek zal dan ook bestaan uit het begrijpbaar maken van hun bewijs.
Vervolgens zullen een aantal toepassingen van dit algoritme worden bekeken. Als eerste zal de toepassing van het algoritme op polytopen beschouwd worden. Hierbij zal een vergelijking getrokken worden met het bekende SIMPLEX algoritme ter demonstratie van de voordelen van het ellipsoïdealgoritme.
De tweede toepassing ligt in de grafentheorie. Een belangrijke eigenschap van een graaf is de grootte van de grootst mogelijke onafhankelijke verzameling punten in een graaf, dit noteren we met $\alpha(G)$ waarbij G een graaf is. Hierbij heet een verzameling punten onafhankelijk als voor elk tweetal punten uit de verzameling geldt dat de punten in de graaf niet verbonden zijn door een lijn. Hieraan gerelateerd is hetzelfde getal voor het sterke product van een graaf met zichzelf, $\alpha(G^n)$. Indien je de punten in een graaf als letters ziet en een lijn tussen twee letters interpreteert als het kunnen verwarren van die twee letters tijdens communicatie dan geeft $\alpha(G^n)$ je de maximale grootte van een woordenboek met woorden van n letters waarin twee woorden niet met elkaar te verwarren zijn. Het supremum over alle n van de n-de machtswortel van $\alpha(G^n)$ is in de informatietechnologie een belangrijk begrip en wordt de Shannon-capaciteit genoemd. De Shannon-capaciteit van een graaf is over het algemeen erg lastig te berekenen, als die al te berekenen is. De Lovász ϑ functie is een functie die aan een graaf een getal toekent. Dit getal blijkt een bovengrens op de Shannon-capaciteit van een graaf. De Lovász ϑ functie van een graaf wordt bepaald door een semidefiniet programma op te lossen. Dit semidefiniete programma blijkt oplosbaar met het ellipsoïdealgoritme.
Het tweede deel van dit onderzoek bestaat dan ook uit twee delen:

(1) Het aantonen dat de Lovász ϑ functie te benaderen is met het ellipsoïdealgoritme,
(2) Het algoritme implementeren om zo voor een aantal grafen de Lovász ϑ functie te kunnen benaderen.

Het zal blijken dat vooral dit laatste deel enige problemen oplevert in verband met de vereiste precisie.

2. Benodigde voorkennis

Voor we beginnen met de formulering van de problemen en definities is het handig om eerst wat meer over een ellips te weten te komen. Een ellips is gedefinieerd met behulp van een positief definiete matrix. We zullen eerst de definitie van een positief definiete matrix geven. Vervolgens zullen we een handige equivalentie daarvan aantonen.

2.1. Positief definiete matrices.

Definitie 1 (Positief definiete matrix)**.** *Een reële $n \times n$-matrix A heet positief definiet als A symmetrisch is en $x^T Ax > 0$ voor alle $0 \neq x \in \mathbb{R}^n$.*

Direct uit deze definitie is het volgende duidelijk:

Gevolg 1. *Laat A, B twee positief definiete $n \times n$-matrices zijn en $0 \leq c \in \mathbb{R}$. Dan geldt ook dat cA en $A + B$ positief definiete matrices zijn.*

Dan volgt nu een handige equivalentie:

Lemma 1. *Een symmetrische $n \times n$-matrix A is positief definiet dan en slechts dan als alle eigenwaarden van A groter dan 0 zijn.*

Bewijs. Neem aan dat A positief definiet is, dus $x^T Ax > 0$ voor alle $0 \neq x \in \mathbb{R}^n$. In het bijzonder geldt dan ook voor elke eigenvector v (bij eigenwaarde λ) dat $v^T Av > 0$. Maar $v^T Av = v^T(Av) = \lambda v^T v = \lambda \langle v, v \rangle = \lambda \|v\|^2$. Waarbij we gebruiken dat het inproduct van v met zichzelf de (euclidische!) norm van v in het kwadraat is. Aangezien $\|v\| > 0$ volgt ook $\lambda > 0$. Dus alle eigenwaarden van A zijn groter dan 0.
Neem nu aan dat alle eigenwaarden van A groter dan 0 zijn. We zullen laten zien dat $x^T Ax > 0$ voor alle $0 \neq x \in \mathbb{R}^n$. Merk eerst op dat uit bovenstaande ook voor alle eigenvectoren v (bij eigenwaarde $\lambda > 0$) volgt dat, aangezien $\lambda > 0$ en $\|v\| > 0$, ook $v^T Av > 0$. Laat nu $v_1, \ldots, v_n$ een orthogonaal stelsel eigenvectoren van A zijn (dit kan, een reële symmetrische matrix is orthogonaal diagonaliseerbaar). Aangezien alle eigenwaarden groter dan 0 zijn geldt $\text{Span}\{v_1, \ldots, v_n\} = \mathbb{R}^n$. Dus we kunnen elke vector $x \in \mathbb{R}^n$ schrijven als een lineaire combinatie van de $v_1, \ldots, v_n$. Laat nu $0 \neq x \in \mathbb{R}^n$ en neem $x = c_1 v_1 + c_2 v_2 + \ldots + c_n v_n$. Wegens de lineariteit van matrix-vector vermenigvuldiging kunnen we nu $x^T Ax$ als volgt schrijven:

$$\begin{aligned} x^T Ax &= (c_1 v_1 + c_2 v_2 + \ldots + c_n v_n)^T A(c_1 v_1 + c_2 v_2 + \ldots + c_n v_n) \\ &= \sum_{i=1}^{n} c_i v_i^T A(c_1 v_1 + c_2 v_2 + \ldots + c_n v_n) \\ &= \sum_{i,j=1}^{n} c_i v_i^T A(c_j v_j) \\ &= \sum_{i,j=1}^{n} c_i c_j v_i^T (Av_j). \end{aligned}$$

We kunnen nu gebruiken dat v_j een eigenvector is van A (bij eigenwaarde λ_j).

$$\begin{aligned} x^T Ax &= \sum_{i,j=1}^{n} c_i c_j v_i^T (\lambda_j v_j) \\ &= \sum_{i,j=1}^{n} \lambda_j c_i c_j v_i^T v_j \\ &= \sum_{i,j=1}^{n} \lambda_j c_i c_j \langle v_i, v_j \rangle \end{aligned}$$

Aangezien we een orthogonaal stelsel eigenvectoren hebben gekozen, dus $\langle v_i, v_j \rangle = 0$ als $i \neq j$ en $\langle v_i, v_i \rangle = \|v_i\|^2 = 1$, volgt nu

$$\begin{aligned} x^T Ax &= \sum_{i=1}^{n} \lambda_i c_i^2 \langle v_i, v_i \rangle \\ &= \sum_{i=1}^{n} \lambda_i c_i^2 \\ &> 0, \end{aligned}$$

waarbij we in de laatste stap gebruiken dat $x \neq 0$. Dit laat zien dat $x^T Ax > 0$ voor alle $0 \neq x \in \mathbb{R}^n$ en dus is A positief definiet. De gewenste equivalentie is dus bewezen. □

Analoog aan de definitie van een positief definiete matrix kunnen we ook een positief semidefiniete matrix definiëren:

Definitie 2 (Positief semidefiniete matrix)**.** *Een reële $n \times n$ matrix A heet positief semidefiniet als A symmetrisch is en $x^T Ax \geq 0$ voor alle $x \in \mathbb{R}^n$.*

Voor positief semidefiniete matrices geldt ook zoiets als Lemma 1, alleen wordt dan de rechter kant van de bewering dat alle eigenwaarden groter of gelijk aan 0 moeten zijn.

2.2. **Ellipsen.** We zullen nu de definitie van een ellips geven, vervolgens zullen we twee eenvoudige voorbeeldjes van ellipsen geven om de definitie te illustreren. Tot slot zullen we nog een paar handige eigenschappen van ellipsen noemen.

Definitie 3 (ellips)**.** *Een ellips rond $x_0 \in \mathbb{R}^n$ is gegeven door $\{x \in \mathbb{R}^n : (x^T - x_0^T)A^{-1}(x - x_0) \leq 1\}$, waarbij A een positief definiete matrix is.*

We zullen eerst aan de hand van twee eenvoudige voorbeelden wat meer gevoel krijgen voor deze definitie.

Voorbeeld 1. *Laat $n = 2$, $A = I$ en $x_0 = 0$. We zien allereerst dat A inderdaad een positief definiete matrix is. Verder geldt (waarbij we schrijven $x = (x_1, x_2)$):*

(2.1) $$\{x \in \mathbb{R}^n : (x^T - x_0^T)A^{-1}(x - x_0) \leq 1\} = \{x \in \mathbb{R}^2 : (x^T - 0)I^{-1}(x - 0) \leq 1\}$$

(2.2) $$= \{x \in \mathbb{R}^2 : x^T Ix \leq 1\}$$

(2.3) $$= \{x \in \mathbb{R}^2 : x_1^2 + x_2^2 \leq 1\}.$$

De ellips is dus de eenheidscirkel.

Voorbeeld 2. *Laat* $n = 2$, $A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ *en* $x_0 = 0$. *Wederom is het duidelijk dat* A *een positief definiete matrix is. We zien* $A^{-1} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix}$. *Verder geldt:*

(2.4)
$$\{x \in \mathbb{R}^n : (x^T - x_0^T)A^{-1}(x - x_0) \leq 1\} = \{x \in \mathbb{R}^2 : (x^T - 0)A^{-1}(x - 0) \leq 1\}$$

(2.5)
$$= \{x \in \mathbb{R}^2 : x^T \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix} x \leq 1\}$$

(2.6)
$$= \{x \in \mathbb{R}^2 : \frac{1}{2}x_1^2 + x_2^2 \leq 1\}.$$

De ellips is in dit geval de eenheidscirkelschijf geschaald met een factor $\sqrt{2}$ *in de* x_1*-richting.*

De oplettende lezer zal misschien opmerken dat in beide gevallen de 'assen' van de ellips samenvallen met de x_1- en x_2-as. Als we nu kijken naar de eigenvectoren van de matrices A dan zien we dat in beide voorbeelden dit de vectoren $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ en $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ zijn. Tevens zijn de eigenwaarden in het eerste voorbeeld beide 1, in het tweede voorbeeld 2 en 1. We zien dus dat de assen in de richting van de eigenvectoren liggen en lengte $\sqrt{\lambda}$ hebben, voor λ de betreffende eigenwaarde. Dit is geen toeval, zoals we ook in het volgende lemma zullen zien.

Lemma 2. *Gegeven een positief definiete matrix* A. *Laat* $\lambda_1, \ldots, \lambda_n$ *de eigenwaarden van* A *zijn en* $v_1, \ldots, v_n$ *de bijbehorende eigenvectoren met* $\|v_i\| = 1$ *voor alle* $1 \leq i \leq n$. *Er geldt dat* $\sqrt{\lambda_i}v_i \in \{x \in \mathbb{R}^n : x^T A^{-1} x \leq 1\}$ *en* $c\sqrt{\lambda_i}v_i \notin \{x \in \mathbb{R}^n : x^T A^{-1} x \leq 1\}$ *voor* $|c| > 1$.

Bewijs. Merk allereerst op dat als v een eigenvector van A is bij eigenwaarde λ dan is v een eigenvector van A^{-1} bij eigenwaarde $1/\lambda$. We weten immers dat A positief definiet is, dus reëel, symmetrisch en met eigenwaarden $\lambda_i > 0$ voor alle i. Dus A is diagonaliseerbaar. Dat wil zeggen er bestaan matrices P, D met D een diagonaalmatrix en P de matrix met bijbehorende eigenvectoren zó dat $A = PDP^{-1}$. Maar dan is $A^{-1} = (PDP^{-1})^{-1} = (P^{-1})^{-1}D^{-1}P^{-1} = PD^{-1}P^{-1}$. Dus is v ook een eigenvector van A^{-1}. We weten ook dat de inverse van een diagonaalmatrix weer een diagonaalmatrix is met $D_{ii}^{-1} = \frac{1}{D_{ii}}$ voor alle i. Dus v is een eigenvector van A^{-1} bij eigenwaarde $1/\lambda$.
We zullen nu laten zien dat $\sqrt{\lambda_i}v_i \in \{x \in \mathbb{R}^n : x^T A^{-1} x \leq 1\}$. Bekijk dus:

$$\begin{aligned}
(\sqrt{\lambda_i}v_i)^T A^{-1}(\sqrt{\lambda_i}v_i) &= \lambda_i v_i^T A^{-1} v_i \\
&= \lambda_i v_i^T (A^{-1} v_i) \\
&= \lambda_i v_i^T (1/\lambda_i v_i) \\
&= \frac{\lambda_i}{\lambda_i} v_i^T v_i \\
&= 1 \cdot \langle v_i, v_i \rangle \\
&= 1.
\end{aligned}$$

Waarbij we gebruiken dat $\langle v_i, v_i \rangle = \|v_i\|^2 = 1$. Dit laat zien dat $\sqrt{\lambda_i}v_i \in \{x \in \mathbb{R}^n : x^T A^{-1} x \leq 1\}$.

Merk op dat hieruit eenvoudig volgt dat voor $|c| > 1$ geldt $(c\sqrt{\lambda_i}v_i)^T A^{-1}(c\sqrt{\lambda_i}v_i) = c^2 > 1$ dus $c\sqrt{\lambda_i}v_i \notin \{x \in \mathbb{R}^n : x^T A^{-1} x \leq 1\}$ voor $|c| > 1$. □

Een gevolg van (het bewijs van) dit lemma is het volgende.

Gevolg 2. *Als A positief definiet is dan is ook A^{-1} positief definiet.*

We kunnen ook nog een andere karakterisering van een ellips geven, aan de hand van een inverteerbare affiene transformatie. We zullen eerst definiëren wat een inverteerbare affiene transformatie is en vervolgens de andere karakterisering van een ellips geven.

Definitie 4 (Een inverteerbare affiene transformatie)**.** *Een affiene transformatie $T : \mathbb{R}^n \to \mathbb{R}^n$ gegeven door $T(x) = x_0 + Qx$, waarbij $x_0 \in \mathbb{R}^n$ en Q een $n \times n$ matrix is, heet inverteerbaar als Q inverteerbaar is.*

Merk op dat deze definitie de naam inverteerbaar ook echt eer aandoet, we kunnen voor zo'n affiene transformatie T namelijk de inverse definiëren: $T^{-1}(x) = Q^{-1}(x - x_0)$. We zien $T \circ T^{-1}(x) = x_0 + Q(Q^{-1}(x - x_0)) = x_0 + x - x_0 = x$ en $T^{-1} \circ T(x) = Q^{-1}((x_0 + Qx) - x_0) = Q^{-1}(Qx + 0) = x$.

Lemma 3. *Laat T een inverteerbare affiene transformatie zijn, dan is het beeld van de eenheidsbol onder T een ellips.*

Bewijs. Laat $T(x) = x_0 + Qx$ zo'n affiene transformatie zijn, waarbij Q een $n \times n$ matrix is. Laat $B_1(0)$ de eenheidsbol in $\mathbb{R}^n$. We willen laten zien dat $T(B_1(0))$ een ellips is. Laat $B = QQ^T$. We zien

$$\begin{aligned}
&\{y = T(x) \in \mathbb{R}^n : (y - x_0)^T B^{-1}(y - x_0) \leq 1\} \\
=&\{y = T(x) \in \mathbb{R}^n : (Qx)^T B^{-1}(Qx) \leq 1\} \\
=&\{y = T(x) \in \mathbb{R}^n : x^T Q^T (Q^T)^{-1} Q^{-1} Qx \leq 1\} \\
=&\{y = T(x) \in \mathbb{R}^n : x^T x \leq 1\} \\
=&\{y = T(x) \in \mathbb{R}^n : \|x\| \leq 1\} = T(B_1(0)).
\end{aligned}$$

Dit is ook echt een ellips aangezien de transformatie T inverteerbaar is en dus elke $y \in \mathbb{R}^n$ te schrijven is als $y = T(x)$ voor een $x \in \mathbb{R}^n$. □

Het zal later in een aantal bewijzen handig blijken om op deze manier naar een ellips te kijken.

2.3. **Notatie.** Vervolgens zullen we nog wat afspraken maken over notatie. In het vervolg van dit artikel zal $\|x\|$ voor een vector $x \in \mathbb{R}^n$ de euclidische lengte van de vector x zijn. Evenzo zal ik voor een matrix A de volgende norm gebruiken $\|A\| = \max\{\|Ax\| : \|x\| = 1\}$. Merk op dat voor een positief (semi) definiete matrix A, gezien het bewijs van het bovenstaande lemma, eenvoudig volgt dat $\|A\|$ het maximum is van de eigenwaarden van A en tevens gelijk aan het $\max\{x^T Ax : \|x\| = 1\}$. Een heel eenvoudig voorbeeldje: laat $A = R^2 I$ met $0 \leq R \in \mathbb{R}$ en I de eenheidsmatrix. Het is eenvoudig in te zien dat $\|A\| = \|R^2 I\| = R^2 \|I\| = R^2$.
Voor een verzameling K en een $\delta > 0$ noteren we de verzameling punten die op afstand hooguit δ van K liggen als $S(K, \delta)$. Dus $S(K, \delta) = \{x \in \mathbb{R}^n : d(x, K) \leq \delta\}$ waarbij $d(x, K)$ de euclidische afstand van x tot K is. Dus $d(x, K) = \inf\{\|x - y\| : y \in K\}$. Merk op dat we zo de gesloten eenheidsbol ook kunnen schrijven als $S(0, 1)$.

3. Het ellipsoïdealgoritme

Allereerst bekijken we de definitie van het algemene optimaliseringsprobleem. Vervolgens zullen we laten zien dat dit algemene probleem niet geschikt is om met een computer op te lossen, in plaats daarvan zullen we een 'zwakke' vorm van het optimaliseringsprobleem geven. Deze zwakke vorm blijkt wel geschikt om algoritmisch op te lossen, dit lukt zelfs in polynomiale tijd. Dit kan met het ellipsoïdealgoritme. Dit algoritme zullen we vervolgens formuleren en in het volgende hoofdstuk zullen we de correctheid ervan aantonen. Dat wil zeggen we laten zien dat het algoritme ook daadwerkelijk een oplossing van het zwakke optimaliseringsprobleem geeft en wel in polynomiale tijd (dit laatste onder bepaalde voorwaarden).

3.1. **De formulering van het probleem.** Laat K een niet-lege convexe, compacte verzameling in $\mathbb{R}^n$ zijn.
We kunnen nu het sterke optimaliseringsprobleem op K definiëren:

Definitie 5 (Het sterke optimaliseringsprobleem)**.** *Gegeven een vector $c \in \mathbb{R}^n$, vind een vector $x \in K$ waarvoor $c^T x$ maximaal is op K.*

Een probleem dat hier sterk mee samenhangt, en heel belangrijk is voor het ellipsoïdealgoritme, is het separatieprobleem:

Definitie 6 (Het sterke separatieprobleem)**.** *Gegeven een vector $y \in \mathbb{R}^n$, bepaal of $y \in K$. Als $y \notin K$ vindt dan een vector $c \in \mathbb{R}^n$ met $\|c\| = 1$ zo dat $c^T y > \max\{c^T x : x \in K\}$.*

Merk op dat $c^T x = c^T y$ in dit geval een hypervlak is wat y van K scheidt.
We zullen nu een voorbeeld geven van een sterk optimaliseringsprobleem wat ook als motivatie zal dienen voor het formuleren van de zwakke vorm van bovenstaande problemen.

Voorbeeld 3. *Laat $K := \{(x,y) \in \mathbb{R}^2 : 0 \leq x^2 + y^2 \leq 1\}$, de gesloten eenheidsbol in $\mathbb{R}^2$, en $c = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Het is duidelijk dat K een convexe, compacte verzameling is. Bovendien is het eenvoudig te zien dat het optimum wordt aangenomen als $x = y$ en $x^2 + y^2 = 2x^2 = 1$. Dus we zien dat $\bar{x} := \frac{1}{\sqrt{2}}(1,1)$ de doelfunctie maximaliseert op K. Dus $\bar{x}$ is een oplossing van het sterke optimaliseringsprobleem.*

In het bovenstaande voorbeeld hebben we gezien dat het mogelijk is dat een oplossing irrationele waarden aanneemt. Een logische vraag is nu hoe zou een algoritme dit antwoord formuleren? Het antwoord is dat dit niet (eenduidig) kan, dit doordat de decimaalontwikkeling van bijvoorbeeld $\frac{1}{\sqrt{2}}$ oneindig veel niet-nul termen bevat. Mocht het algoritme je een antwoord geven dat op de eerste 500 termen gelijk is aan de decimaalontwikkeling van $\frac{2}{\sqrt{2}}$, kan je dan concluderen dat het optimum ook echt $\frac{2}{\sqrt{2}}$ is? Het antwoord is nee; wat je wel kan zeggen is dat je optimum binnen een straal van $\frac{2}{10^{500}}$ van $\frac{2}{\sqrt{2}}$ ligt. Dit geeft ons aanleiding om het volgende probleem te definiëren:

Definitie 7 (Het zwakke optimaliseringsprobleem)**.** *Gegeven een vector $c \in \mathbb{Q}^n$ en een $0 < \epsilon \in \mathbb{Q}$. Vind een vector $y \in \mathbb{Q}^n$ zo dat $d(y,K) \leq \epsilon$ en $c^T x \leq c^T y + \epsilon$ voor alle $x \in K$.*

Hierin staat $d(y,K)$ voor de euclidische afstand van y tot de verzameling K. We zeggen ook wel dat y bijna $c^T x$ maximaliseert.
Evenzo kunnen we ook het zwakke separatieprobleem opstellen:

Definitie 8 (Het zwakke separatieprobleem)**.** *Gegeven een vector $y \in \mathbb{Q}^n$ en een $0 < \epsilon \in \mathbb{Q}$, bepaal of $d(y, K) \leq \epsilon$. Als $d(y, K) > \epsilon$ geef dan een vector $c \in \mathbb{Q}^n$ met $\|c\| \geq 1$ zodanig dat $\max\{c^T x : x \in K\} \leq c^T y + \epsilon$.*

Deze twee problemen zijn met een computer wel goed op te lossen. Je krijgt immers een rationele output, en een breuk is uniek te representeren (neem de teller en noemer relatief priem).
We willen nu een algoritme vinden wat het zwakke optimaliseringsprobleem kan oplossen en die polynomiaal is in de input, in het bijzonder in de grootte van ϵ. We zullen in de volgende paragraaf laten zien dat het ellipsoïdealgoritme zo'n algoritme is. Het ellipsoïdealgoritme maakt gebruik van afronding op p binaire cijfers.

Voorbeeld 4 (Afronden op p binaire cijfers)**.** *Het afronden van een reëel getal x op p binaire cijfers gebeurt als volgt: deel x door de kleinste macht van 2 (zeg dat dit 2^k is) waarvoor het resultaat in het interval $[0, 2^p - 1]$ ligt. Rond dit resultaat vervolgens af op het dichtstbijzijnde gehele getal, dit is te schrijven als een binair getal van p cijfers, laat dit y zijn. Dan is $\hat{x} = y \cdot 2^k$ het getal x afgerond op p binaire cijfers. Laat bijvoorbeeld $p = 4$ en $x = 37$. We zien dat de kleinste macht van 2 waarvoor $x/2^k \in [0, 15]$ gelijk is aan 2^2. We krijgen zo $x/4 = \frac{37}{4} = 9.25$. Dus in dit geval is $y = 9$. Dus $\hat{x} = 9 \cdot 4 = 36$. De macht van 2 hoeft overigens niet positief te zijn. Nemen we bij dezelfde p bijvoorbeeld $x = 3.156$. Dan zien we dat 2^{-2} de laagste macht van 2 is waarvoor $x/2^k \in [0, 15]$. Immers is $\frac{x}{2^{-2}} = 3.156 \cdot 4 = 12.624$. Dit geeft dus $y = 13$ en dus $\hat{x} = 13 \cdot 2^{-2} = 3.25$.*

Bovendien maakt het ellipsoïdealgoritme gebruik van oplossingen van het separatieprobleem, het is echter helemaal niet duidelijk of dit probleem wel altijd oplosbaar is. Laten we daarom eerst nog even naar het separatieprobleem kijken.

De vraag is dus of er voor een gegeven een compacte, convexe verzameling K en een punt $x \notin K$ altijd een hypervlak bestaat wat x van K scheidt. Gelukkig is dit het geval!
In Hoofdstuk 5 van [2] wordt een belangrijke stelling behandeld, de stelling van *Hahn-Banach*. Wat deze stelling precies inhoudt is voor dit verhaal niet van belang, wat wel van belang is is een direct gevolg van deze stelling. Dit gevolg is ook wel bekend onder de naam *Separation Theorem* en dit gevolg geeft ons precies wat we willen (in [2] is dit stelling $5.30(b)$):

Stelling 1 (Separatie-stelling)**.** *Laat X een reële of complexe genormeerde vectorruimte zijn en laat $A, B \subset X$ niet-lege, disjuncte convexe verzamelingen zijn. Als A compact is en B gesloten dan bestaat er een $f \in X'$ zo dat voor een zekere $\gamma \in \mathbb{R}$ en een $\delta > 0$ geldt*

$$Re(f(a)) \leq \gamma - \delta < \gamma + \delta \leq Re(f(b)), \qquad \forall a \in A,\ \forall b \in B.$$

Hierin is X' de verzameling begrensde, lineaire transformaties van X naar het lichaam waarover gewerkt wordt: $\mathbb{F}$ (vaak dus $\mathbb{R}$ of $\mathbb{C}$). Oftewel, in het geval dat $X = \mathbb{R}^n$, $\mathbb{F} = \mathbb{R}$ en $f \in X'$ is de verzameling $f(x) = c$, voor een c vast, een hypervlak. Nemen we nu $A = K$ en $B = \{x\}$ dan zien we dat A convex en compact is en B gesloten, aangezien $x \notin K$ zijn A en B disjunct. Het is duidelijk dat B ook convex is. Deze stelling geeft ons dus dat er een $f \in X'$ bestaat zó dat $f(x) \geq \gamma + \delta > \gamma - \delta \geq f(y)$ voor alle $y \in K$. Dus $f(z) = f(x)$ is een hypervlak wat x van K scheidt.
Dit laat ook direct zien waarom de eis dat K convex moet zijn niet weggelaten kan worden, als K niet convex is hoeft zo'n hypervlak niet te bestaan. We kunnen hier een eenvoudig voorbeeldje van geven.

Voorbeeld 5. *Neem voor K de gesloten annulus met stralen $r = 1, R = 2$ en middelpunt $(0,0)$ in $\mathbb{R}^2$, dus $K = \{(x,y) \in \mathbb{R}^2 : 1 \leq x^2 + y^2 \leq 2\}$. Merk op dat K compact is, maar niet convex. Dan zien we dat $(0,0) \notin K$, maar er bestaat ook duidelijk geen hypervlak wat $(0,0)$ van K scheidt, ieder vlak door de oorsprong heeft immers niet-lege doorsnede met de annulus.*

3.2. **Het algoritme.** De input van het ellipsoïdealgoritme is als volgt: een compacte convexe verzameling K, een lineaire doelfunctie $c^T x$, een punt $a_0 \in \mathbb{R}^n$ en twee stralen $0 < r \leq R$ zo dat $S(a_0, r) \subseteq K \subseteq S(a_0, R)$. Waarbij $S(a_0, r)$ de euclidische bol van straal r rond a_0 is. Dit laatste wil simpelweg zeggen dat K volledig dimensionaal en begrensd is en dat we ook een expliciete bol binnen en om de verzameling K kennen. De aanname dat K volledig dimensionaal is is eenvoudig te verantwoorden, als K niet volledig dimensionaal is dan kunnen we hetzelfde probleem beschouwen op de kleinste (qua dimensie) affiene deelruimte van $\mathbb{R}^n$ waarin K in ligt. Vervolgens is er nog een gegeven nodig: een separatiealgoritme (SEP).
De input van SEP zal zijn een $x \in \mathbb{R}^n$, de verzameling K en een $\delta > 0$. Vervolgens zal dit algoritme ofwel bepalen dat $x \in S(K, \delta)$ ofwel dat $x \notin S(K, \delta)$. In het tweede geval zal je tevens een vector $d \in \mathbb{R}^n$ krijgen zo dat $\|d\| \geq 1$ en $\max\{d^T y : y \in K\} \leq d^T x + \delta$. In het tweede geval krijg je dus een hypervlak terug wat x van K scheidt.
Dan nu het algoritme. Laat

$$N = 5n^2 \lceil \log \frac{2R^2\|c\|}{r\epsilon} \rceil, \tag{3.1}$$

$$\delta = \frac{R^2 4^{-N}}{300n}, \tag{3.2}$$

$$p = 5N. \tag{3.3}$$

Hierbij staat $\lceil \log \frac{2R^2\|c\|}{r\epsilon} \rceil$ voor het kleinste gehele getal groter of gelijk aan $\log \frac{2R^2\|c\|}{r\epsilon}$.
Definieer nu een rij vectoren $x_0, x_1, \ldots x_N$ en matrices $A_0, A_1, \ldots, A_N$ als volgt. Laat $x_0 = a_0$ en $A_0 = R^2 I$. Als x_k en A_k bepaald zijn voer dan het separatiealgoritme (SEP) uit met als input $y = x_k$ en δ. Als je nu terugkrijgt dat $x_k \in S(K, \delta)$ dan noemen we k een toelaatbare index en nemen we $a = c$. Als SEP je een vector $d \in \mathbb{R}^n$ geeft zo dat $\|d\| \geq 1$ en $\max\{d^T x : x \in K\} \leq d^T x_k + \delta$ dan noemen we k een niet toelaatbare index en nemen we $a = -d$. Definieer nu:

$$b_k = \frac{A_k a}{\sqrt{a^T A_k a}}, \tag{3.4}$$

$$x_k^* = x_k + \frac{1}{n+1} b_k, \tag{3.5}$$

$$A_k^* = \frac{2n^2+3}{2n^2}\left(A_k - \frac{2}{n+1} b_k b_k^T\right). \tag{3.6}$$

Laat vervolgens

$$x_{k+1} \approx x_k^*, \qquad A_{k+1} \approx A_k^*$$

Waarbij $\approx$ betekent dat we de linkerkant verkrijgen door de rechterkant af te ronden op p binaire getallen achter de decimaal punt, waarbij we er wel op letten dat A_{k+1} symmetrisch is.
De rij x_k, k toelaatbaar, geeft een goede benadering van het optimum. Dat wil zeggen $\max\{c^T x_k : k \text{ toelaatbaar}\} \geq \max\{c^T x : x \in K\} - \epsilon$. Dit zullen we in Hoofdstuk 4 ook bewijzen. Voor we dat doen zullen we echter eerst nog even kijken naar het geometrische idee achter deze berekeningen. Dit zullen we aan de hand van een groot voorbeeld doen.

3.2.1. *Geometrische interpretatie.* Laat $n = 2$, $A = I$, $a = \begin{pmatrix}1\\0\end{pmatrix}$, $x_0 = \begin{pmatrix}0\\0\end{pmatrix}$. Laat $E = \{(x,y) \in \mathbb{R}^2 : (x,y)^T A^{-1}(x,y) \leq 1\}$. Dit betekent dus dat onze ellips E de eenheidsbol in $\mathbb{R}^2$ is en het scheidend hypervlak de y-as is. Door een stap van het algoritme uit te voeren verwachten we dus dat de rechter-helft van de eenheidsbol in de nieuwe ellips ligt. We zullen nu laten zien dat dit ook daadwerkelijk zo is.
Bereken eerst $b = Aa/\sqrt{a^T Aa} = I \cdot \begin{pmatrix}1\\0\end{pmatrix} \cdot \frac{1}{1} = \begin{pmatrix}1\\0\end{pmatrix} = a$. Vervolgens zien we dat $x_0^* = x_0 + \frac{1}{n+1}b = \frac{1}{3}\begin{pmatrix}1\\0\end{pmatrix}$.
Merk nu op dat dit betekent dat x_0^* opgeschoven is in de richting die loodrecht staat op het scheidend hypervlak. Bovendien zien we dat x_0^* op de lijn tussen x_0 en punt wat het verst van het hypervlak af ligt, $\begin{pmatrix}1\\0\end{pmatrix}$. En wel zo dat de verhouding $\|x_0^* - x\| : \|\begin{pmatrix}1\\0\end{pmatrix} - x_0^*\|$ gelijk is aan $1 : n$.
We bekijken nu de matrix A^*.

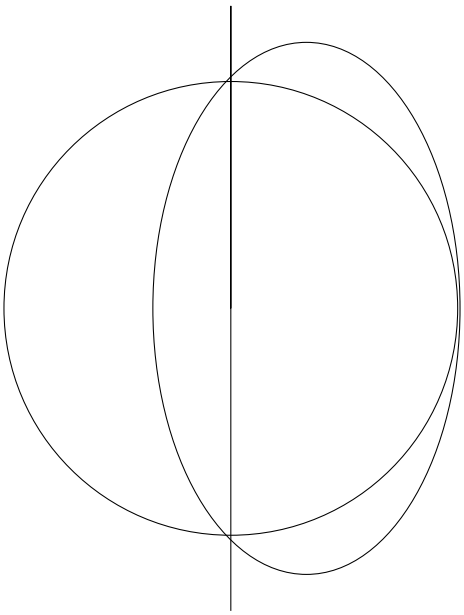
$$\begin{aligned}
A^* &= \frac{2n^2+3}{2n^2}\left(A - \frac{2}{n+1}bb^T\right)\\
&= \frac{2n^2+3}{2n^2}\left(I - \frac{2}{n+1}\begin{pmatrix}1&0\\0&0\end{pmatrix}\right)\\
&= \frac{2n^2+3}{2n^2}\begin{pmatrix}1-\frac{2}{n+1}&0\\0&1\end{pmatrix}\\
&= \frac{2n^2+3}{2n^2}\begin{pmatrix}\frac{n-1}{n+1}&0\\0&1\end{pmatrix}\\
&= \frac{11}{8}\begin{pmatrix}\frac{1}{3}&0\\0&1\end{pmatrix}
\end{aligned}$$

We kunnen deze ellipsen natuurlijk ook grafisch weergeven, zie hiervoor figuur 1. We zien hier de eenheidscirkel, samen met het gekozen hypervlak en de ellips gegeven door A^* en x_0^*.
We kunnen zien dat de nieuwe ellips de doorsnede van de rechterhalfruimte en de eenheidsbol volledig bevat, we kunnen dit echter ook algebraïsch laten zien. We zullen aantonen dat de 3 uiterste punten van de rechter-helft van de eenheidsbol in de ellips $E^* = \{(x,y) \in \mathbb{R}^2 : (x-1/3,y)^T (A^*)^{-1}(x-1/3,y) \leq 1\}$ liggen. Deze drie punten zijn $\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\-1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}$. Allereerst voor $\begin{pmatrix}0\\1\end{pmatrix}$.

$$\begin{aligned}
(-1/3,1)^T (A^*)^{-1}(-1/3,1) &= \begin{pmatrix}-1/3 & 1\end{pmatrix}\frac{8}{11}\begin{pmatrix}3&0\\0&1\end{pmatrix}\begin{pmatrix}-1/3\\1\end{pmatrix}\\
&= \frac{8}{11}(\frac{3}{9}+1) = \frac{8}{11}\frac{12}{9} = \frac{96}{99} \leq 1
\end{aligned}$$

Dus $\begin{pmatrix}0\\1\end{pmatrix} \in E^*$. Vanwege symmetrie zien we dat ook $\begin{pmatrix}0\\-1\end{pmatrix} \in E^*$.
Nu voor $\begin{pmatrix}1\\0\end{pmatrix}$.

$$\begin{aligned}
(1-1/3,0)^T (A^*)^{-1}(1-1/3,0) &= \begin{pmatrix}2/3 & 0\end{pmatrix}\frac{8}{11}\begin{pmatrix}3&0\\0&1\end{pmatrix}\begin{pmatrix}2/3\\0\end{pmatrix}\\
&= \frac{8}{11}(\frac{4}{9}\cdot 3) = \frac{8}{11}\frac{12}{9} = \frac{96}{99} \leq 1
\end{aligned}$$



FIGUUR 1. Afbeelding van de ellipsen E (de eenheidscirkel) en E^*, samen met het hypervlak.

Dus ook $\begin{pmatrix}1\\0\end{pmatrix} \in E^*$.

Merk op dat als we geen rekening met afrondfouten hadden gehouden en dus in de berekening van A^* de iets kleinere term $\frac{n^2}{n^2-1}$ hadden gebruikt in plaats van $\frac{2n^2+3}{2n^2}$ dan waren we in bovenstaande 3 punten op gelijkheid uitgekomen:

$$\begin{aligned}(1-1/3,0)^T(A^*)^{-1}(1-1/3,0) &= \begin{pmatrix}2/3 & 0\end{pmatrix}\frac{3}{4}\begin{pmatrix}3 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}2/3\\0\end{pmatrix}\\ &= \frac{3}{4}(\frac{4}{9}\cdot 3) = \frac{3}{4}\frac{12}{9} = 1\end{aligned}$$

De met deze factor verkregen ellips heet ook wel de *Löwner-John*-ellipsoïde. Meer informatie over dit interessante type ellipsoïden is te vinden in [5].

Een belangrijke eigenschap van de nieuwe ellips is dus dat hij het 'goede' deel van de oude ellips bevat. Natuurlijk is dit voor het algoritme niet genoeg, we willen namelijk ook graag dat het volume van de ellipsen afneemt. Het liefst een beetje snel. In Hoofdstuk 4 zullen we laten zien dat het volume van de nieuwe ellips kleiner is dan dat van de oude ellips en wel zo veel dat er nog een factor, zeg a (afhankelijk van de dimensie) tussen zit die kleiner dan 1 is. Dat betekent dat het volume van de ellips exponentieel afneemt in het aantal iteraties. We zien immers $\text{vol}(E_k) = a \cdot \text{vol}(E_{k-1}) = a^2 \cdot \text{vol}(E_{k-2}) = \ldots = a^k \cdot \text{vol}(E_0)$, met $0 < a < 1$.

4. De correctheid van het algoritme

We hebben in Hoofdstuk 3 gezien wat het ellipsoïdealgoritme is, in dit hoofdstuk zullen we de correctheid van het algoritme aantonen. In grote lijnen hebben we hiertoe drie dingen te bewijzen, elk zullen we in een eigen paragraaf doen. We zullen allereerst laten zien dat je in elke stap een ellips hebt. Vervolgens zullen we laten zien dat de verzameling toegelaten punten die een hogere waarde hebben dan de dan toe best gevonden oplossing in de huidige ellips ligt. Hiervoor zullen we de volgende notatie invoeren:

$$\xi_k = \max\{c^T x_j : 0 \leq j < k, j \text{ toelaatbaar}\} \tag{4.1}$$

$$K_k = K \cap \{x : c^T x \geq \xi_k\} \tag{4.2}$$

$$E_k = \{x : (x - x_k)^T A_k^{-1} (x - x_k) \leq 1\} \tag{4.3}$$

$$E_k^* = \{x : (x - x_k)^T (A_k^*)^{-1} (x - x_k) \leq 1\} \tag{4.4}$$

Waarbij (4.1) slechts gedefinieerd is voor $k > 0$, als gevolg daarvan is ook (4.2) slechts voor $k > 0$ gedefinieerd. We kunnen (4.2) echter ook definiëren voor $k = 0$ door $K_0 := K$. Wat we dus zullen laten zien is dat $K_k \subseteq E_k$. Tot slot zullen we nog aantonen dat we na N stappen ook daadwerkelijk een $y \in K$ hebben gevonden zó dat $c^T y \geq \max\{c^T x : x \in K\} - \epsilon$. Dit samen bewijst precies de correctheid van het algoritme.

4.1. Elke stap een ellips.

We zullen nu beginnen met het eerste deel, dit doen we in het volgende lemma. Merk op dat dit lemma ons precies geeft dat we in elke stap een ellips hebben, een ellips wordt immers gegeven door een positief definiete matrix A en een middelpunt x (zie ook Hoofdstuk 2).

Lemma 4. *De matrices $A_0, \ldots, A_N$ zijn positief definiet. Tevens geldt voor elke $0 \leq k \leq N$:*

$$\|x_k\| \leq \|a_0\| + R2^k, \qquad \|A_k\| \leq R^2 2^k, \qquad \|A_k^{-1}\| \leq R^{-2} 4^k. \tag{4.5}$$

Bewijs. We doen dit met inductie naar k. Merk op dat voor $k = 0$ de beweringen zeker waar zijn. Immers geldt $x_0 = a_0, \|A_0\| = \|R^2 I\| = R^2$ en $\|A_0^{-1}\| = \|R^{-2} I\| = R^{-2}$. Neem nu aan dat de beweringen waar zijn voor een zekere k. Dus $A_0, \ldots, A_k$ zijn positief definiet en voor $0 \leq i \leq k$ hebben we

$$\|x_i\| \leq \|a_0\| + R2^i, \qquad \|A_i\| \leq R^2 2^i, \qquad \|A_i^{-1}\| \leq R^{-2} 4^i.$$

We laten zien dat ze ook waar zijn voor $k + 1$.
We zullen allereerst laten zien dat A_k^* positief definiet is. Vanwege Gevolg 2 voldoet het te laten zien dat $(A_k^*)^{-1}$ positief definiet is.

Bewering 1. $(A_k^*)^{-1} = \frac{2n^2}{2n^2+3} \left(A_k^{-1} + \frac{2}{n-1} \cdot \frac{aa^T}{a^T A_k a} \right)$.

We zullen laten zien dat dit ook echt de inverse van A_k^* is. Merk op dat aangezien A_k^* en de potentiele inverse beide symmetrisch zijn het voldoet te laten zien dat $A_k^* \cdot (\frac{2n^2}{2n^2+3} \left(A_k^{-1} + \frac{2}{n-1} \cdot \frac{aa^T}{a^T A_k a} \right)) = I$.

Aldus beschouw $A_k^* \cdot (\frac{2n^2}{2n^2+3}\left(A_k^{-1} + \frac{2}{n-1} \cdot \frac{aa^T}{a^T A_k a}\right))$.

$$\begin{aligned}
A_k^* \cdot (A_k^*)^{-1} &= \frac{2n^2+3}{2n^2}\left(A_k - \frac{2}{n+1}b_k b_k^T\right) \\
&\quad \cdot \frac{2n^2}{2n^2+3}\left(A_k^{-1} + \frac{2}{n-1} \cdot \frac{aa^T}{a^T A_k a}\right) \\
&= (A_k - \frac{2}{n+1}b_k b_k^T)(A_k^{-1} + \frac{2}{n-1}\frac{aa^T}{a^T A_k a}) \\
&= I + \frac{2}{n-1}A_k\frac{aa^T}{a^T A_k a} - \frac{2}{n+1}b_k b_k^T A_k^{-1} - \frac{4}{(n-1)(n+1)}b_k b_k^T\frac{aa^T}{a^T A_k a} \\
&= I + \frac{2}{n-1}\frac{A_k aa^T}{a^T A_k a} - \frac{2}{n+1}\frac{A_k aa^T}{a^T A_k a} - \frac{4}{(n-1)(n+1)}b_k b_k^T\frac{aa^T}{a^T A_k a} \\
&= I + (\frac{2}{n-1} - \frac{2}{n+1})\frac{A_k aa^T}{a^T A_k a} - \frac{4}{(n-1)(n+1)}b_k b_k^T\frac{aa^T}{a^T A_k a} \\
&= I + \frac{4}{(n+1)(n-1)}\frac{A_k aa^T}{a^T A_k a} - \frac{4}{(n-1)(n+1)}b_k b_k^T\frac{aa^T}{a^T A_k a} \\
&= I + \frac{4}{(n+1)(n-1)}\frac{A_k aa^T}{a^T A_k a} - \frac{4}{(n-1)(n+1)}\frac{A_k aa^T A_k}{a^T A_k a}\frac{aa^T}{a^T A_k a} \\
&= I + \frac{4}{(n+1)(n-1)}\frac{A_k aa^T}{a^T A_k a} - \frac{4}{(n-1)(n+1)}\frac{A_k aa^T A_k aa^T}{(a^T A_k a)^2} \\
&= I + \frac{4}{(n+1)(n-1)}\frac{A_k aa^T}{a^T A_k a} - \frac{4}{(n-1)(n+1)}\frac{A_k a(a^T A_k a)a^T}{(a^T A_k a)^2} \\
&= I + \frac{4}{(n+1)(n-1)}\frac{A_k aa^T}{a^T A_k a} - \frac{4}{(n-1)(n+1)}\frac{A_k aa^T}{a^T A_k a} \\
&= I
\end{aligned}$$

Merk op dat A_k^{-1} positief definiet is vanwege de inductiehypothese en Gevolg 2. Er volgt dat $(A_k^*)^{-1}$ positief definiet is, aangezien het de som van twee positief definiete matrices is (zie ook Gevolg 1). Zoals al eerder opgemerkt weten we dat dan ook A_k^* positief definiet is. We kunnen dit nu gebruiken om de norm van A_k^* af te schatten. Er volgt immers:

$$\|A_k^*\| = \frac{2n^2+3}{2n^2}\|A_k - \frac{2}{n+1}b_k b_k^T\| \leq \frac{2n^2+3}{2n^2}\|A_k\| \leq \frac{2n^2+3}{2n^2}R^2 2^k$$

Waarbij we in de laatste stap gebruik hebben gemaakt van de inductieveronderstelling ($\|A_k\| \leq R^2 2^k$).
Merk nu op dat we $\|A_{k+1} - A_k^*\|$ kunnen afschatten op $n2^{-p}$. Immers staat op elke positie hooguit 2^{-p}. Laat nu $x' \in \mathbb{R}^n$ de vector met op elke positie een 1 zijn en neem $x = x'/\sqrt{n}$. We zien $\|x\| = 1$. We zien $x^T(A_{k+1} - A_k^*)x \leq \frac{1}{n}\sum_{i,j=1}^n x_i x_j 2^{-p}$. Merk nu op dat aangezien $\|x\| = 1$ zeker geldt dat iedere $x_i \leq 1$. Dus $x^T(A_{k+1} - A_k^*)x \leq \frac{1}{n}\sum_{i,j=1}^n x_i x_j 2^{-p} \leq \frac{1}{n}n^2 2^{-p} = n2^{-p}$. Zoals in Hoofdstuk 2 is opgemerkt wil dit zeggen dat $\|A_{k+1} - A_k^*\| \leq n2^{-p}$. We zullen nu bovenstaande gebruiken om de norm van A_{k+1} af te schatten.

$$\|A_{k+1}\| \leq \|A_k^*\| + \|A_{k+1} - A_k^*\| \leq \frac{2n^2+3}{2n^2}R^2 2^k + n2^{-p}$$

Merk nu op dat $\frac{2n^2+3}{2n^2} = 1 + \frac{3}{2n^2}$ en $n2^{-p} \leq R^2\frac{2n^2-3}{2n^2}2^k$, dit laatste volgt eenvoudig door p in te vullen. We zien dan immers $n2^{-p} = n2^{-20n^2\lceil \log \frac{2R^2\|c\|}{r\epsilon}\rceil} \leq n\left(\frac{2R^2\|c\|}{r\epsilon}\right)^{-20n^2}$ en dat is vele malen kleiner dan $R^2\frac{2n^2-3}{2n^2}2^k$.
Dus $\|A_{k+1}\| \leq \frac{2n^2+3}{2n^2}R^2 2^k + n2^{-p} \leq R^2 2^{k+1}$.
Bekijk nu de norm van b_k:

$$\|b_k\| = \frac{\|A_k a\|}{\sqrt{a^T A_k a}} = \sqrt{\frac{a^T A_k^2 a}{a^T A_k a}}$$

Dit laatste willen wij graag afschatten op $\sqrt{\|A_k\|}$. We zullen laten zien dat dat kan.
Bekijk A_k^2. We weten dat A_k te schrijven is als $A_k = PDP^T$ voor D een diagonaal matrix en $PP^T = P^TP = I$. Evenzo weten we dat $\|A_k\|$ de grootste eigenwaarde van A is en dus $\|A_k\| = \max_i D_{ii}$. Maar dan volgt $D_{ii}^2 \leq D_{ii}\|A_k\|$. Dus we zien dat voor elke vector v geldt $v^TA_k^2v = v^TPD^2P^Tv \leq v^TP\|A_k\|DP^Tv = \|A_k\|v^TA_kv$. Links en rechts door v^TA_kv delen (merk op dat $v^TA_kv > 0$ want A_k is positief definiet) geeft nu de gewenste ongelijkheid. Dus

$$\|b_k\| = \sqrt{\frac{a^T A_k^2 a}{a^T A_k a}} \leq \sqrt{\|A_k\|}$$

Dit kunnen we nu gebruiken om de norm van x_{k+1} af te schatten:

$$\|x_{k+1}\| \leq \|x_k^* + x_{k+1} - x_k^*\| \leq \|x_k^*\| + \|x_{k+1} - x_k^*\| \leq \|x_k\| + \frac{1}{n+1}\|b_k\| + \|x_{k+1} - x_k^*\|$$

Invullen wat we al weten geeft:

$$\|x_{k+1}\| \leq \|a_0\| + R2^k + \frac{1}{n+1}R2^k + \sqrt{n}2^{-p} \leq \|a_0\| + R2^{k+1}$$

Waarbij de laatste afschatting weer op eenzelfde manier als eerder gaat.
Beschouw nu de norm van $(A_k^*)^{-1}$. Rechtstreeks uit de definitie vinden we met de driehoeksongelijkheid:

$$\begin{aligned}\|(A_k^*)^{-1}\| &\leq \frac{2n^2}{2n^2+3}\left(\|A_k^{-1}\| + \frac{2}{n-1}\frac{\|a\|^2}{a^TA_ka}\right)\\ &\leq \frac{2n^2}{2n^2+3}\left(\|A_k^{-1}\| + \frac{2}{n-1}\|A_k^{-1}\|\right)\\ &= \frac{2n^2}{2n^2+3}\left(\frac{n+1}{n-1}\|A_k^{-1}\|\right) < \frac{n+1}{n-1}\|A_k^{-1}\|\end{aligned}$$

Hierbij gebruiken we in de tweede ongelijkheid dat $\|a\|^2 \leq \|A_k^{-1}\|a^TA_ka$. Deze ongelijkheid kunnen we op de volgende manier bewijzen: $\|a\|^2 = a^Ta \leq a^TAa$ waarbij A een positief definiete matrix is met eigenwaarden groter of gelijk aan 1. Bewering: $A_k\|A_k^{-1}\|$ is zo'n matrix. Bewijs: we kunnen wederom schrijven $A_k = PDP^T$. Merk op dat $\|A_k^{-1}\| = \frac{1}{\lambda_0}$, waarbij λ_0 de kleinste eigenwaarde van A_k is. Maar dan geldt dus $D_{ii}\|A_k^{-1}\| = \frac{\lambda_i}{\lambda_0} \geq 1$ voor alle i. Dus $A_k\|A_k^{-1}\|$ is inderdaad zo'n matrix. De gewenste ongelijkheid is dus gerechtvaardigd.

We zullen al het bovenstaande nu gebruiken om een ondergrens voor de kleinste eigenwaarde van A_{k+1} te vinden. Laat λ_0 de kleinste eigenwaarde van A_{k+1} zijn en v een bijbehorende eigenvector met $\|v\| = 1$. We zien: $\lambda_0 = v^TA_{k+1}v =$

$v^T A_k^* v + v^T (A_{k+1} - A_k^*)v$. Merk nu op dat $v^T A_k^* v \geq \lambda_0^*$, waarbij λ_0^* de kleinste eigenwaarde van A_k^* is. Maar $\lambda_0^* = \|(A_k^*)^{-1}\|^{-1}$. Bovendien geldt $|v^T(A_{k+1} - A_k^*)v| \leq \|A_{k+1} - A_k^*\|$, dus $v^T(A_{k+1} - A_k^*)v \geq -\|A_{k+1} - A_k^*\|$. Dit gebruikende vinden we:

$$\begin{aligned}\lambda_0 &\geq \|(A_k^*)^{-1}\|^{-1} - \|A_{k+1} - A_k^*\| \\ &\geq \frac{n-1}{n+1}\|A_k^{-1}\|^{-1} - n2^{-p} \\ &\geq \frac{n-1}{n+1}R^2 4^{-k} - n2^{-p} > R^2 4^{-(k+1)}\end{aligned}$$

Dit laat zien dat A_{k+1} positief definiet is en dat $\|A_{k+1}^{-1}\| = \frac{1}{\lambda_0} \leq R^{-2}4^{k+1}$. □

4.2. **Elke stap een 'goede' ellips.** We hebben nu dus laten zien dat het algoritme in elke stap ook daadwerkelijk een ellips geeft. We zullen nu laten zien dat we in elke stap het 'goede' deel van de verzameling K bewaren. Dat wil zeggen we willen bewijzen dat $K_k \subseteq E_k$ voor $k = 0, \ldots, N$.

Lemma 5. *$K_k \subseteq E_k$ voor $k = 0, \ldots, N$.*

Bewijs. We zullen dit met inductie naar k bewijzen. Voor $k = 0$ geldt de uitspraak overduidelijk, per aanname geldt immers $K_0 = K \subseteq S(a_0, R) = E_0$. Neem nu aan dat de uitspraak geldt voor k. Laat $x \in K_{k+1}$, we willen aantonen dat $x \in E_{k+1}$. Herinner:

$$\xi_k = \max\{c^T x_j : 0 \leq j < k, j \text{ toelaatbaar}\}$$
$$K_k = K \cap \{x : c^T x \geq \xi_k\}$$

Het is duidelijk dat $\xi_{k+1} \geq \xi_k$ voor alle k, het maximum over een grotere verzameling kan immers niet kleiner worden. Maar dat wil zeggen dat $K_{k+1} \subseteq K_k$. Dus $x \in K_k$ en, aangezien de uitspraak voor k geldt, dus ook $x \in E_k$. Bovendien weten we dat

$$a_k^T x \geq a_k^T x_k - \delta \tag{4.6}$$

Waarbij a_k de a is die in stap k is gebruikt. Merk op dat als k een toelaatbare index is we zelfs kunnen zeggen dat $a_k^T x \geq a_k^T x_k$, maar dan geldt zeker bovenstaande ongelijkheid.
Schrijf nu $x = x_k + y + tb_k$ waarbij $a_k^T y = 0$. Merk op dat dit kan aangezien a_k en b_k niet orthogonaal zijn, immers

$$a_k^T b_k = \frac{a_k^T A_k a_k}{\sqrt{a_k^T A_k a_k}} > 0$$

waarbij we gebruiken dat A_k positief definiet is.
We weten dat $x \in E_k$ dus $1 \geq (x - x_k)^T A_k^{-1}(x - x_k) = (y + tb_k)^T A_k^{-1}(y + tb_k) = y^T A_k^{-1} y + t^2 b_k^T A_k^{-1} b_k$. Waarbij we in de laatste stap gebruiken dat $a_k^T y = 0$, hieruit volgt immers

$$y^T A_k^{-1} b_k = y^T A_k^{-1} \frac{A_k a_k}{\sqrt{a_k^T A_k a_k}} = \frac{y^T a_k}{\sqrt{a_k^T A_k a_k}} = 0.$$

Verder zien we $t^2 b_k^T A_k^{-1} b_k = t^2$ door eenvoudig uit te schrijven. We vinden dus dat $1 \geq y^T A_k^{-1} y + t^2$ en aangezien $y^T A_k^{-1} y > 0$ zien we dus ook dat $t^2 \leq 1$.
Aan de andere kant vinden we door vergelijking 4.6 ook dat $-\delta \leq a_k^T(x - x_k) = a_k^T(y + tb_k) = 0 + ta_k^T b_k = ta_k^T b_k = t\frac{a_k^T A_k a_k}{\sqrt{a_k^T A_k a_k}} = t\sqrt{a_k^T A_k a_k}$.

We zullen nu aantonen dat $(x - x_{k+1})^T A_{k+1}^{-1}(x - x_{k+1}) \leq 1$. Allereerst schrijf $(x - x_{k+1})^T A_{k+1}^{-1}(x - x_{k+1}) = (x - x_k^*)^T (A_k^*)^{-1}(x - x_k^*) + R_1$ waarbij R_1 een restterm is. Merk op dat $R_1 \approx n^2 2^{-p}$ waarbij we hogere orde termen van 2^{-p} verwaarlozen. Invullen geeft

$$R_1 \approx n^2 2^{-5 \cdot 4n^2 \log \frac{2R^2\|c\|}{r\epsilon}} = n^2 \left(\frac{2R^2\|c\|}{r\epsilon}\right)^{-4n^2}$$

We kunnen inzien dat het geen sterke aanname is dat $\left(\frac{R^2\|c\|}{r\epsilon}\right)^{n^2} > n$ er volgt dus

$$R_1 < n^2 \frac{1}{(2n)^4} < \frac{1}{12n^2}.$$

We zullen nu de 'hoofdterm' afschatten:

(4.7)
$$(x - x_k^*)^T (A_k^*)^{-1}(x - x_k^*) = \frac{2n^2}{2n^2+3}\left(\left(t - \frac{1}{n+1}\right)b_k + y\right)^T \cdot$$

(4.8)
$$\cdot \left(A_k^{-1} + \frac{2}{n-1} \cdot \frac{a_k a_k^T}{a_k^T A_k a_k}\right) \cdot \left(\left(t - \frac{1}{n+1}\right)b_k + y\right)$$

(4.9)
$$= \frac{2n^2}{2n^2+3}\left(\left(t - \frac{1}{n+1}\right)^2 + y^T A_k^{-1} y + \frac{2}{n-1}\left(t - \frac{1}{n+1}\right)^2\right)$$

(4.10)
$$\leq \frac{2n^2}{2n^2+3}\left(\frac{n^2}{n^2-1} - \frac{2t(1-t)}{n-1}\right)$$

(4.11)
$$\leq \frac{2n^4}{2n^4+n^2-3} + \frac{4\delta}{(n-1)\sqrt{a_k^T A_k a_k}} \frac{2n}{2n^4+n^2-3} + \frac{4\delta}{n-1}\|A_k^{-1}\|$$

(4.12)
$$\leq \frac{2n^4}{2n^4+n^2-3} + \frac{4\delta R^{-2} 4^N}{n-1} \leq 1 - \frac{1}{12n^2}$$

Waarbij we in regel 4.10 het volgende hebben gebruikt:

$$\frac{n+1}{n-1}\left(t - \frac{1}{n+1}\right)^2 + y^T A_k^{-1} y = \frac{(n+1)t^2 - 2t}{n-1} + \frac{1}{n^2-1} + y^T A_k^{-1} y \leq \frac{n^2}{n^2-1} - \frac{2t(1-t)}{n-1}$$

Waarbij de laatste stap gerechtvaardigd is omdat $y^T A_k^{-1} y \leq 1 - t^2$ en dus

$$\begin{aligned}\frac{(n+1)t^2 - 2t}{n-1}\frac{1}{n^2-1} + y^T A_k^{-1} y &\leq \frac{(n+1)t^2 - 2t}{n-1} + \frac{1}{n^2-1} + \frac{n^2-1}{n^2-1} - t^2\frac{n-1}{n-1} \\ &= \frac{n^2}{n^2-1} - \frac{2t(1-t)}{n-1}\end{aligned}$$

Deze afschatting geeft samen met de afschatting voor R_1 geeft dit $(x - x_{k+1})^T A_{k+1}^{-1}(x - x_{k+1}) \leq 1$, dus $x \in E_{k+1}$. Aangezien $x \in K_{k+1}$ willekeurig was laat dit zien $K_{k+1} \subseteq E_{k+1}$. □

4.3. **Bereikt het algoritme zijn doel?** In deze paragraaf zullen we de belangrijkste stelling bewijzen, we laten namelijk zien dat we na N stappen ook daadwerkelijk een $y \in K$ hebben gevonden zo dat $c^T y \geq \max\{c^T x : x \in K\} - \epsilon$. Voor we dit kunnen doen hebben we echter nog een aantal kleine lemma's nodig. Zoals we al eerder hebben genoemd willen we graag iets weten over het volume van de ellipsen. In het bijzonder de verhouding tussen het volume van de ellips in stap $k+1$ ten opzichte van het volume van de ellips in stap k.

Lemma 6. *Laat μ het n-dimensionale volume voorstellen. Dan geldt er*

$$\frac{\mu(E_{k+1})}{\mu(E_k)} < e^{-1/(5n)} \tag{4.13}$$

Bewijs. Merk allereerst op dat we alleen het geval hoeven te beschouwen dat $x_0 = 0, A_k = I, a = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n$.
Dit is als volgt in te zien met behulp van Lemma 3. Beschouw zo'n affiene transformatie $T(x) = x_0 + Qx$ met Q een inverteerbare $n \times n$ matrix. We kunnen E_k met de inverse transformatie T^{-1} transformeren naar de eenheidsbol, aangezien die rotatiesymmetrisch is kunnen we daar als scheidend hypervlak $a^T x = 0$ nemen, het algoritme toepassen geeft zo een ellips E'. Vervolgens kunnen we op E' weer dezelfde affiene transformatie T toepassen om E_{k+1} te krijgen. Maar van de affiene transformatie T weten we wat die met het volume doet, namelijk $\mu(T(B_1(0))) = \det(Q)\mu(B_1(0))$. Zo zien we

$$\frac{\mu(E_{k+1})}{\mu(E_k)} = \frac{\det(Q)\mu(E')}{\det(Q)\mu(B_1(0))} = \frac{\mu(E')}{\mu(B_1(0))}$$

Het is dus voldoende om alleen het eerder genoemde geval te behandelen.
Laat nu dus $A = I, x = 0$ en $a = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n$. We zien:

$$b = Aa/\sqrt{a^T Aa} = I \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} / \sqrt{a^T Ia} = I \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} / \sqrt{\|a\|^2} = a \tag{4.14}$$

$$x^* = 0 + \frac{1}{n+1} b = \frac{1}{n+1} a \tag{4.15}$$

(4.16)

$$A^* = \frac{2n^2+3}{2n^2} \left(I - \frac{2}{n+1} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \right) = \frac{2n^2+3}{2n^2} \begin{pmatrix} 1 - \frac{2}{n+1} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 \end{pmatrix}$$

We zien dat we ook de nieuwe ellips als het beeld van de eenheidsbol door een affiene transformatie kunnen zien, namelijk de transformatie $T(x) = x^* + Qx$

waarbij

$$Q = \sqrt{A^*} = \sqrt{\frac{2n^2+3}{2n^2}} \begin{pmatrix} \sqrt{\frac{n-1}{n+1}} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 \end{pmatrix}.$$

Hieruit volgt

$$\frac{\mu(E')}{\mu(B_1(0))} = \frac{\det(Q)\mu(B_1(0))}{\mu(B_1(0))} = \det(Q)$$

De determinant van Q is eenvoudig te zien:

$$\det(Q) = \sqrt{\frac{n-1}{n+1}} \cdot \left(\frac{2n^2+3}{2n^2}\right)^{\frac{n}{2}}$$

Dus we zien dat

$$\frac{\mu(E_k^*)}{\mu(E_k)} = \sqrt{\frac{n-1}{n+1}} \cdot \left(\frac{2n^2+3}{2n^2}\right)^{\frac{n}{2}}.$$

Het volume van E_{k+1} hangt natuurlijk sterk samen met dat van E_k^*. We verkrijgen A_{k+1} door A_k^* af te ronden op p binaire getallen achter de decimaalpunt, dus voor onze matrix Q betekent dat dat alle posities met $\sqrt{1+2^{-p}}$ kunnen worden vermenigvuldigd. De determinant van Q wordt dan hooguit met $(1+2^{-p})^n$ vermenigvuldigd. Dus

$$\frac{\mu(E_{k+1})}{\mu(E_k)} = (1+2^{-p})^n \sqrt{\frac{n-1}{n+1}} \cdot \left(\frac{2n^2+3}{2n^2}\right)^{\frac{n}{2}}.$$

Het is niet moeilijk in te zien dat $(1+2^{-p})^n \leq (1+2^{-25})$, aangezien $-p = -25n^2\lceil \log \frac{2R^2\|c\|}{r\epsilon} \rceil$.
Gebruiken we nu de afschatting $1+x \leq e^x$ dan vinden we

$$\begin{aligned} \sqrt{\frac{n-1}{n+1}} \cdot \left(\frac{2n^2+3}{2n^2}\right)^{\frac{n}{2}} &= (1-\frac{2}{n+1})^{\frac{1}{2}} \left(1+\frac{3}{2n^2}\right)^{\frac{n}{2}} \\ &\leq \exp(-2/(n+1))^{\frac{1}{2}} \exp(3/(2n^2))^{\frac{n}{2}} \\ &= \exp(\frac{-1}{n+1} + \frac{3}{4n}) \\ &= \exp(\frac{3(n+1)-4n}{4n(n+1)}) \\ &= \exp(\frac{-1}{4n} + \frac{1}{n(n+1)}) \end{aligned}$$

We zien dat $\exp(\frac{-1}{4n} + \frac{1}{n(n+1)}) \leq \exp(\frac{-1}{5n})$ als $\frac{1}{n(n+1)} \leq \frac{1}{4n} - \frac{1}{5n} = \frac{1}{20n}$. Dus voor $n \geq 19$ kunnen we inderdaad zeggen dat de determinant van Q kleiner of gelijk aan $e^{\frac{-1}{5n}}$ is. Nemen we de erg kleine factor $(1+2^{-25})$ ook in beschouwing dan kunnen we vanaf $n = 20$ zeggen dat het volume van de nieuwe ellips kleiner is dan dat van de vorige maal $e^{\frac{-1}{5n}}$. Immers geldt voor $n \geq 20$

$$\frac{\exp(\frac{-1}{5n})}{\exp(\frac{-1}{4n} + \frac{1}{n(n+1)})} \leq (1+2^{-20})$$

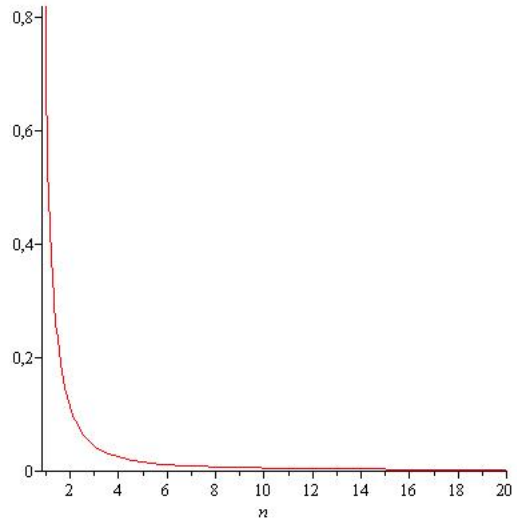
We moeten nu dus nog nagaan of we voor $1 \leq n < 20$ dezelfde afschatting kunnen maken. Maar dit is eenvoudig na te gaan door voor deze n de ongelijkheid

$$(1 + 2^{-25})\sqrt{\frac{n-1}{n+1}} \cdot \left(\frac{2n^2+3}{2n^2}\right)^{\frac{n}{2}} \leq \exp(\frac{-1}{5n})$$

te controleren. In figuur 2 is een plot van de rechterkant van de ongelijkheid min de linkerkant gegeven, we zien dat dit verschil groter dan 0 is. Dus ook voor $1 \leq n < 20$ geldt de ongelijkheid.
Het gewenste resultaat is dus bereikt. We kunnen concluderen dat

$$\frac{\mu(E_{k+1})}{\mu(E_k)} \leq \exp(\frac{-1}{5n})$$

□



FIGUUR 2. Een plot van de grafiek van $\exp(\frac{-1}{5n}) - (1 + 2^{-25})\sqrt{\frac{n-1}{n+1}} \cdot \left(\frac{2n^2+3}{2n^2}\right)^{\frac{n}{2}}$. Op de horizontale as loopt n van 1 tot 20.

Voor we de belangrijkste stelling bekijken is het handig om nog even naar het volume van een kegel te kijken en naar wat daarmee gebeurd als we de kegel doorsnijden met een vlak evenwijdig aan de basis van de kegel.

Beschouw een kegel A met basis B en hoogte H, hierbij is H de afstand van de top van de kegel tot het vlak waarin de basis ligt, het grondvlak. Het n-dimensionale volume van de kegel kunnen we berekenen door te integreren over de hoogte, waarbij de integrand het $(n-1)$-dimensionale volume van het grondvlak van de kegel doorsnede op hoogte x is. Als we de kegel doorsnijden met een vlak evenwijdig aan het grondvlak dan krijgen we een gelijkvormige kegel. Het $(n-1)$-dimensionale volume van het grondvlak is geschaald met een factor gelijk aan de verhouding van de hoogten tussen de twee kegels tot de macht $(n-1)$. Zo kunnen we het volume van de kegel A bepalen door de volgende integraal uit te rekenen:

$$\mu(A) = \int_0^H \mu(B)(\frac{x}{H})^{n-1}dx = \mu(B)\frac{H}{n} \tag{4.17}$$

Merk op dat deze vergelijking een uitbreiding naar hogere dimensies is van een zeer bekende vergelijking. Namelijk de vergelijking voor de oppervlakte van een driehoek: oppervlakte $\Delta = \frac{\text{(lengte grondvlak)} \times \text{hoogte}}{2}$.
Aan de hand van het bovenstaande is het niet lastig in te zien dat als we de kegel doorsnijden met een vlak evenwijdig aan de basis dat het volume van de nieuwe kegel K' gegeven bepaald wordt door de verhouding tussen de twee hoogtes H' van K' en H van K. En wel op de volgende manier:

$$\mu(K') = \mu(K) \left(\frac{H'}{H} \right)^n . \tag{4.18}$$

Waarbij de n-de macht komt door de dimensie waarin K en K' liggen.

We kunnen nu beginnen aan het bewijs van het belangrijkste resultaat. We zullen aantonen dat het algoritme na N stappen ook daadwerkelijk een goede oplossing voor het probleem geeft. Herinner: een index i heet toelaatbaar als $x_i \in S(K, \delta)$.

Stelling 2. *Laat j een toelaatbare index zijn waarvoor*

$$c^T x_j = \max\{c^T x_k : 0 \leq k < N, k \text{ toelaatbaar}\} \tag{4.19}$$

Dan geldt $c^T x_j \geq \max\{c^T x : x \in K\} - \epsilon$.

Bewijs. Laat V_n het volume van de n-dimensionale eenheidsbol zijn. Merk nu op dat uit de Lemma's 5 en 6 volgt:

$$\mu(K_N) \leq \mu(E_N) \leq e^{-N/5n} \mu(E_0) = e^{-N/5n} R^n V_n \tag{4.20}$$

Laat nu

$$\xi = \max\{c^T x : x \in K\} \tag{4.21}$$

en $y \in K$ zo dat $c^T y = \xi$. Beschouw de kegel G waarvan de basis de $(n-1)$-dimensionale bol rond x_0 met straal r in het vlak $c^T x = c^T x_0$ is en de top het punt y. Merk op dat de $(n-1)$-dimensionale bol rond x_0 met straal r in K ligt, aangezien x_0 en r zo gegeven waren dat de n-dimensionale bol met straal r rond x_0 in K ligt. Aangezien K convex is en $y \in K$ ligt dus de hele kegel in K.
Doorsnijdt deze kegel nu met de halfruimte $c^T x \geq c^T x_j$, noem de nieuwe kegel G'. Direct uit de definitie van K_N volgt dat deze nieuwe kegel in K_N ligt. We willen nu graag vergelijking (4.20) gebruiken, zoals al eerder opgemerkt moeten we daartoe de verhouding tussen de hoogte van de nieuwe kegel en de oude weten.
De hoogte van de oorspronkelijke kegel kunnen we als de afstand tussen de vlakken $c^T x = c^T x_0$ en $c^T x = \xi$ zien. Deze afstand is

$$\frac{\xi - c^T x_0}{\|c\|}.$$

Dit volgt uit de *Cauchy-Schwarz* ongelijkheid (zie bladzijde 57 van [2]). Deze ongelijkheid zegt namelijk dat $|\langle u, v \rangle| \leq \|u\| \|v\|$ voor alle $u, v \in \mathbb{R}^n$ waarbij gelijkheid geldt dan en slechts dan als u en v lineair afhankelijk van elkaar zijn. Kies nu een y' in het vlak $c^T x = \xi$ zo dat $y' - x_0$ loodrecht staat op het vlak $c^T x = c^T x_0$. Merk op dat $c^T y' = \xi$. Er volgt dan eenvoudig dat c en $y' - x_0$ lineair afhankelijk zijn en dus geldt $\langle c, y' - x_0 \rangle = \xi - c^T x_0 = \|c\| \|y' - x_0\|$, het is duidelijk dat $\|y' - x_0\|$ de afstand tussen de twee vlakken is. Op eenzelfde manier is natuurlijk ook te zien dat de afstand tussen de vlakken $c^T x = c^T x_j$ en $c^T x = \xi$ gelijk is aan $\frac{\xi - c^T x_j}{\|c\|}$

We kunnen nu het volume van de kegel G geven, gebruikmakend van vergelijking (4.17):

$$\mu(G) = \frac{V_{n-1} \cdot r^{n-1} \cdot (\xi - c^T x_0)}{n\|c\|} \tag{4.22}$$

Het volume van de kegel G' volgt dan eenvoudig uit (4.18) en het bovenstaande:

$$\begin{aligned} \mu(G') &= \frac{V_{n-1} \cdot r^{n-1} \cdot (\xi - c^T x_0)}{n\|c\|} \left(\frac{H'}{H}\right)^n \\ &= \frac{V_{n-1} \cdot r^{n-1} \cdot (\xi - c^T x_0)}{n\|c\|} \left(\frac{\xi - c^T x_j}{\|c\|} \frac{\|c\|}{\xi - c^T x_0}\right)^n \\ &= \frac{V_{n-1} \cdot r^{n-1} \cdot (\xi - c^T x_0)}{n\|c\|} \left(\frac{\xi - c^T x_j}{\xi - c^T x_0}\right)^n \end{aligned} \tag{4.23}$$

Zoals al eerder opgemerkt weten we $G' \subseteq K_N$ en dus $\mu(G) \leq \mu(K_N)$. Met (4.23) en (4.20) vinden we zo

$$\frac{V_{n-1} \cdot r^{n-1} \cdot (\xi - c^T x_0)}{n\|c\|} \left(\frac{\xi - c^T x_j}{\xi - c^T x_0}\right)^n \leq \mu(K_N) \leq e^{-N/5n} R^n V_n \tag{4.24}$$

Dit kunnen we omschrijven naar

$$\xi - c^T x_j \leq e^{-N/5n^2} R \left(\frac{\xi - c^T x_0}{r}\right)^{\frac{n-1}{n}} \left(\frac{nV_n}{V_{n-1}}\right)^{\frac{1}{n}} \|c\|^{\frac{1}{n}} \tag{4.25}$$

We willen nu nog een bovengrens op ξ vinden, dit doen we wederom met de *Cauchy-Schwarz* ongelijkheid:

$$|\xi - c^T x_0| = |c^T(y - x_0)| \leq \|c\| \cdot \|y - x_0\| \leq R\|c\| \tag{4.26}$$

Waarbij we gebruiken dat $y \in K \subseteq S(x_0, R)$. Vullen we dit in in vergelijking (4.25) dan zien we:

$$\xi - c^T x_j \leq e^{-N/5n^2} \frac{R^2}{r} \|c\| \left(\frac{nV_n}{V_{n-1}}\right)^{\frac{1}{n}} \tag{4.27}$$

Bewering 2. $\left(\frac{nV_n}{V_{n-1}}\right)^{\frac{1}{n}} < 2$ *voor alle* $n \in \mathbb{N}_{\geq 2}$.

Voor we deze bewering zullen bewijzen bekijken we eerst wat dit betekent. Herinner $N = 5n^2 \lceil \log \frac{2R^2\|c\|}{r\epsilon} \rceil$. Invullen in vergelijking (4.27) geeft

$$\begin{aligned} \xi - c^T x_j &\leq e^{-N/5n^2} \frac{R^2}{r} \|c\| \left(\frac{nV_n}{V_{n-1}}\right)^{\frac{1}{n}} \\ &< 2e^{-N/5n^2} \frac{R^2}{r} \|c\| \\ &\leq 2e^{-\lceil \log \frac{2R^2\|c\|}{r\epsilon} \rceil} \frac{R^2}{r} \|c\| \\ &< 2e^{-\log \frac{2R^2\|c\|}{r\epsilon}} \frac{R^2}{r} \|c\| \\ &< \left(\frac{2R^2\|c\|}{r\epsilon}\right)^{-1} 2\frac{R^2}{r} \|c\| \\ &= \epsilon \end{aligned} \tag{4.28}$$

Dit besluit het bewijs, we zien immers $c^T x_j \geq \xi - \epsilon$. □

Ons rest nu dus nog de bewering te bewijzen.

Bewijs bewering. Laat

$$a_n = \left(\frac{nV_n}{V_{n-1}}\right)^{\frac{1}{n}}, \qquad n = 2, 3, \ldots$$

Om de bewering te bewijzen willen we een belangrijke eigenschap van deze rij gebruiken. Namelijk dat deze rij twee dalende deelrijen $(a_{2k})_{k=1}^{\infty}$ en $(a_{2k+1})_{k=1}^{\infty}$ heeft die samen de hele rij vormen. We zullen eerst laten zien dat deze deelrijen dalend zijn. Met andere woorden: we zullen laten zien dat $a_{n+2} < a_n$ voor alle $n \in \mathbb{N}$. Oftewel $\frac{a_{n+2}}{a_n} < 1$ voor alle $n \in \mathbb{N}$.
Merk allereerst op dat $(n+2)^{1/(n+2)} < n^{1/n}$ voor alle $n > 2$, zoals eenvoudig met differentiëren aan te tonen is (het maximum van de functie $f : \mathbb{R} \to \mathbb{R}$ gegeven door $f(x) = x^{1/x}$ ligt bij $x = e$, voor $x > e$ geldt $f'(x) < 0$). Dus $\frac{(n+2)^{1/(n+2)}}{n^{1/n}} < 1$.
Ons rest nu dus nog aan te tonen dat

$$\left(\frac{V_{n+2}}{V_{n+1}}\right)^{1/(n+2)} < \left(\frac{V_n}{V_{n-1}}\right)^{1/n}.$$

Hiertoe zullen we laten zien dat

$$\frac{\frac{V_{n+2}}{V_{n+1}}}{\frac{V_n}{V_{n-1}}} < 1.$$

Aldus, we weten $V_n = \pi^{n/2}/\Gamma(\frac{n}{2}+1)$. Zo zien we

$$\frac{\frac{V_{n+2}}{V_{n+1}}}{\frac{V_n}{V_{n-1}}} = \frac{V_{n+2}}{V_n}\frac{V_{n-1}}{V_{n+1}} \tag{4.29}$$

$$= \frac{\pi^{n/2+1}/\Gamma(n/2+2)}{\pi^{n/2}/\Gamma(n/2+1)}\frac{\pi^{(n-1)/2}/\Gamma((n-1)/2+1)}{\pi^{(n+1)/2}/\Gamma((n+1)/2+1)} \tag{4.30}$$

$$= \frac{\pi}{\pi}\frac{\Gamma(n/2+1)}{(n/2+1)\Gamma(n/2+1)}\frac{(n+1)/2\Gamma((n-1)/2+1)}{\Gamma((n-1)/2+1)} \tag{4.31}$$

$$= \frac{n+1}{n+2} < 1 \tag{4.32}$$

Waarbij we in (4.31) de volgende eigenschap van de gammafunctie gebruiken: $\Gamma(z+1) = z\Gamma(z)$.
Maar dan volgt dus

$$\left(\frac{V_{n+2}}{V_{n+1}}\right)^{\frac{1}{n+2}} = \left(\frac{n+1}{n+2}\frac{V_n}{V_{n-1}}\right)^{\frac{1}{n+2}} \tag{4.33}$$

$$< \left(\frac{V_n}{V_{n-1}}\right)^{\frac{1}{n+2}} \tag{4.34}$$

$$< \left(\frac{V_n}{V_{n-1}}\right)^{\frac{1}{n}} \tag{4.35}$$

Waarbij we in (4.35) gebruikt hebben dat $\frac{V_n}{V_{n-1}} \geq 1$.
We vinden dus dat $a_{n+2} < a_n$ voor alle $n \in \mathbb{N}$. Dus de deelrijen (a_{2k}) en (a_{2k+1}) zijn dalend.
Maar dat wil zeggen dat we het supremum van de rij (a_n) over alle n kunnen bepalen door het maximum van de eerste twee termen van de rij te nemen. Dus $\sup_n a_n = \max\{a_2, a_3\}$. Een eenvoudige berekening geeft $a_2 = \left(\frac{\pi}{2}\right)^{1/2}$ en $a_3 =$

$\left(\frac{4}{3}\right)^{1/3}$, beide zijn kleiner dan 2. Dus $\sup_n a_n < 2$. Daarmee is dus aangetoond dat $\left(\frac{nV_n}{V_{n-1}}\right)^{\frac{1}{n}} < 2$ voor alle $n \in \mathbb{N}_{\geq 2}$. □

We hebben dus nu bewezen dat het algoritme correct is, dat wil zeggen het geeft daadwerkelijk een oplossing voor het gegeven probleem in het aantal stappen dat vooraf bepaald is. We zullen in het volgende hoofdstuk het belang van het algoritme toelichten, dit doen we aan de hand van een eenvoudig voorbeeld. De klasse van polytopen.

5. Het belang van het ellipsoïdealgoritme

In dit hoofdstuk zullen we het belang van het ellipsoïdealgoritme toelichten. We zullen eerst kort wat zeggen over de complexiteit van het algoritme. Vervolgens zullen we als voorbeeld de klasse van polytopen bekijken. We zullen voor dat probleem de vergelijking maken met een ander bekend algoritme, het SIMPLEX algoritme. Dat zal de voordelen van dit algoritme illustreren.

5.1. **De complexiteit.** Met de complexiteit van een algoritme wordt de rekentijd bedoeld in een *worst-case* scenario. Deze rekentijd is meestal afhankelijk van de input. Bijvoorbeeld van de dimensie van het probleem en/of van de grootte van de getallen in de input. Algoritmen kunnen dan in twee klassen worden ingedeeld:

(1) Algoritmen waarbij de rekentijd een polynoom is in de grootte van de invoer, de grootte van de invoer wordt gemeten in de lengte van een codering van de invoer (in ons geval binair),
(2) Algoritmen waarbij dat niet zo is.

Voor heel grote invoer zijn algoritmen van de eerste klasse dus sneller dan die van de tweede. Het is dan ook gebruikelijk om je af te vragen of een algoritme tot de eerste of tot de tweede klasse behoort. Evenzo is het een interessante vraag of er voor een bepaald probleem een algoritme van de eerste klasse bestaat. Als dat zo is kan je namelijk concluderen dat het probleem niet al te moeilijk is. Laten we nu kijken naar het ellipsoïdealgoritme.
In paragraaf 3.2 hebben we in de vergelijkingen (3.1) tot en met (3.6) het ellipsoïdealgoritme gezien. Laten we eerst de laatste drie stappen bekijken, de bepaling van b_k, x_k^* en A_k^*. Aangezien deze stappen slechts optelling en vermenigvuldiging van vectoren en/of matrices bevatten is het duidelijk dat deze stappen uitgevoerd kunnen worden in een aantal berekeningen wat begrensd wordt door de dimensie n. Verder zien we dat in vergelijking (3.1) dat het benodigde aantal stappen $N = 5n^2 \lceil \log \frac{2R^2\|c\|}{r\epsilon} \rceil$ is. Het logaritme geeft hier de grootte van (een deel van) de invoer weer. N is dus een polynoom in de grootte van de invoer.
We zien dus dat we elke stap de rekentijd kunnen begrenzen door een polynoom, bovendien kunnen we het aantal stappen begrenzen door een polynoom, dus kunnen we ook de benodigde rekentijd begrenzen door een polynoom. Hierbij zijn we wel aan één belangrijk punt voorbij gegaan. Voor we de laatste drie stappen kunnen uitvoeren moeten we eerst een scheidend hypervlak vinden. Dit doen we met het separatiealgoritme. We zien dan ook dat het ellipsoïdealgoritme een gegeven probleem polynomiaal kan oplossen dan en slechts dan als het separatiealgoritme polynomiaal is. Dit kunnen we als stelling formuleren:

Stelling 3. *Laat $\mathcal{K}$ een klasse van convexe, compacte verzamelingen zijn. Dan geldt dat het optimaliseringsprobleem voor de instanties van $\mathcal{K}$ polynomiaal oplosbaar is als het separatieprobleem polynomiaal oplosbaar is voor instanties van $\mathcal{K}$.*

Bewijs. Het bewijs volgt onmiddelijk uit het ellipsoïdealgoritme. □

Deze stelling zou zelfs als een "dan en slechts dan als" bewering geformuleerd kunnen worden, het bewijs daarvan kan worden gevonden in Hoofdstuk 3 van [4]. Het is voor dit verhaal echter niet zo van belang.

5.2. **Polytopen.** Een belangrijke klasse van convexe, compacte verzamelingen zijn de polytopen. Hierbij zien we een polytoop als een (eindige) doorsnede van gesloten halfruimten die begrensd is. Merk op dat een gesloten halfruimte een convexe verzameling is, dus ook een polytoop is convex. Tevens is een

doorsnede van gesloten verzamelingen ook gesloten, dus een polytoop is gesloten. Aangezien we eisen dat een polytoop begrensd is volgt er dus dat een polytoop gesloten en begrensd is. Maar dat wil zeggen dat een polytoop compact is, volgens de bekende stelling van *Heine-Borel*: een verzameling $V \subseteq \mathbb{R}^n$ is compact dan en slechts dan als V gesloten en begrensd is. (Stelling 1.4 van [3])
We kunnen een gesloten halfruimte $C \subseteq \mathbb{R}^n$ beschrijven door middel van een vector $0 \neq a \in \mathbb{R}^n$ en een getal b en wel op de volgende manier: $C = \{x \in \mathbb{R}^n : a^T x \leq b\}$. Een doorsnede van k halfvlakken kunnen we dan ook beschrijven met een $k \times n$ matrix A en een k-vector b als $\{x \in \mathbb{R}^n : Ax \leq b\}$ waarbij de rijen van A de vectoren $a_1, a_2, \ldots, a_k$ zijn. Een polytoop zullen we dan ook vaak schrijven als $Ax \leq b$ of (A, b), hiermee bedoelen we $\{x \in \mathbb{R}^n : Ax \leq b\}$.

Nemen we nu een vector $c \in \mathbb{R}^n$ dan kunnen we daarmee een lineaire kostenfunctie $c^T x$ definiëren. Gegeven een polytoop $P = (A, b)$ en een inwendig punt $y \in \mathbb{R}^n$ kunnen we, zoals in definitie 5, dan het volgende optimaliseringsprobleem op deze polytoop definiëren:

$$\begin{array}{ll} \max & c^T x \\ \text{o.d.v.} & Ax \leq b \end{array} \tag{5.1}$$

Een bekend algoritme om dit probleem op te lossen is het simplexalgoritme, zie Figuur 3, we zullen hier niet verder op de werking van het algoritme in gaan, voor meer informatie over het simplexalgoritme kan gevonden worden in Hoofdstuk 2 van [1].

begin
opt:='nee', onbegrensd:= 'nee'
(Opmerking: zodra een van beide 'ja' wordt, stopt het algoritme)
while opt = 'nee' & onbegrensd = 'nee' **do**
if $\bar{c_j} \geq 0$ voor alle j **then** opt:='ja'
else begin
Kies een j zo dat $\bar{c_j} < 0$;
if $x_{ij} \leq 0$ voor alle i **then** onbegrensd:='ja'
else vind

$$\frac{x_{k0}}{x_{kj}} = \min_i \{\frac{x_{i0}}{x_{ij}} : x_{ij} > 0\}$$

en pivot op x_{kj}.
end
end
end
end

FIGUUR 3. Het simplexalgoritme in pseudocode.

Het belangrijkste om op te merken is dat dit algoritme ook echt geschikt is voor dit soort problemen. Tevens is ook bekend dat dit algoritme niet polynomiaal is. We zullen nu laten zien dat het separatiealgoritme voor dit probleem polynomiaal is, dan volgt uit Stelling 3 dat het probleem wel polynomiaal oplosbaar is met behulp van het ellipsoïdealgoritme.
Neem een polytoop (A, b) waarbij A een $k \times n$ matrix is, laat a_i de i-de rij van A zijn. We moeten dus laten zien dat we gegeven een $x \in \mathbb{R}^n$ in polynomiale tijd kunnen bepalen of $Ax \leq b$ en als dat niet zo is moeten we een hypervlak geven zó dat x aan de ene kant van het hypervlak ligt en de polytoop aan de andere

kant. Welnu we kunnen voor $i = 1, 2, \ldots, k$ eenvoudig in polynomiale tijd nagaan of $a_i x \leq b_i$. Als dat voor elke i zo blijkt te zijn kunnen we concluderen dat x in de polytoop zit. Stel dat er een ongelijkheid geschonden wordt, dus $a_j x > b_j$ voor een j, dan geeft die ongelijkheid een scheidend hypervlak: het vlak gegeven door de vergelijking $a_j z = a_j x$. Immers geldt $a_j x > b_j \geq a_j y$ voor alle y in de polytoop, dus het vlak $a_j z = a_j x$ is inderdaad een scheidend hypervlak. Merk op dat het vinden van een scheidend hypervlak duidelijk ook in polynomiale tijd kan. Dit laat zien dat voor polytopen het separatiealgoritme polynomiaal is en dus kunnen we dit soort problemen polynomiaal oplossen met behulp van het ellipsoïdealgoritme.
Merk op dat we hier een aanname hebben gemaakt die bij simplex niet vereist is namelijk dat we een inwendig punt y van de polytoop kennen. Dit is hier 'nodig' omdat het ellipsoïdealgoritme dit als initialisatie gebruikt. Er staan hier aanhalingstekens om 'nodig' omdat je ook zou kunnen laten zien dat het ellipsoïdealgoritme zonder deze kennis vooraf een goede oplossing kan geven. Een schets voor deze methode is gegeven in Hoofdstuk 7. Voor veel problemen waarop je simplex zou toepassen is er echter wel een toegelaten oplossing, oftewel een inwendig punt van de polytoop, beschikbaar dus het is ook geen al te sterke aanname.
We zien dus dat het ellipsoïdealgoritme het optimaliseringsprobleem voor polytopen kan oplossen als de polytoop expliciet gegeven is. Dat is op zich natuurlijk al een mooie prestatie, maar het algoritme kan nog meer. Als de polytoop niet expliciet gegeven is maar er wel een separatiealgoritme beschikbaar is dan kan het ellipsoïdealgoritme het probleem nog steeds oplossen. Dit is natuurlijk interessant als de polytoop heel veel facetten heeft. Als voorbeeld zullen we de *Spanning-Tree* polytoop van een graaf beschouwen.

Voorbeeld 6 (Spanning-Tree polytoop)**.** *Een opspannende boom is een verzameling lijnen T in een graaf zo dat elk punt verbonden is met een lijn en er geen cykels bestaan in de (V, T). Merk op dat voor T geldt $|T| = n - 1$ met $n = |V|$. Laat $G = (V, E)$ een graaf zijn. Voor iedere $e \in E$ definiëren we een variabele $x(e)$ en voor een verzameling $U \subseteq V$ noteren we met $E(U)$ de verzameling lijnen in G waarvoor beide eindpunten in U liggen. We gebruiken de volgende notatie $x(E(U)) = \sum_{e \in E(U)} x(e)$. De polytoop bestaat dan uit alle x waarvoor:*

$$\begin{aligned} x(E) &= n - 1 \\ x(E(U)) &\leq |U| - 1, \qquad U \subseteq V \\ 0 \leq x(e) &\leq 1, \qquad e \in E \end{aligned} \tag{5.2}$$

Dit is het convexe opspansel van alle opspannende bomen in G. De tweede voorwaarde wil hierbij precies zeggen dat de verzameling lijnen in de graaf gegeven door x geen cykels bevat. Gezien deze voorwaarde is het duidelijk dat deze polytoop gegeven wordt door exponentieel veel voorwaarden in het aantal punten van de graaf. Hebben we echter een x gegeven dan is het eenvoudig na te gaan of deze x in de polytoop zit doormiddel van een max-flow/min-cut algoritme. We kunnen immers de graaf opstellen die als gewichten op de lijnen de waarden $x(e)$ heeft. Vervolgens kunnen we voor elk tweetal punten i, j de maximale stroom van i naar j bepalen en de bijbehorende minimale snede. Als die maximale stroom groter dan 1 is wil dat zeggen dat x een convexe combinatie van vectoren y, z is waarvan ten minste een van de twee een cykel bevat (die i, j bevat). De minimale snede geeft nu een vector d zo dat $d^T x > 1$ en $1 \geq d^T y$ voor alle y in de polytoop. Als voor elk tweetal punten de maximale stroom kleiner of gelijk aan 1 is kan je concluderen dat x in de polytoop zit. Er bestaan polynomiale max-flow/min-cut algoritmen en bovendien is het aantal tweetallen punten wat je moet onderzoeken gelijk aan $\binom{n}{2} = \frac{n(n-1)}{2}$. In

het voorgaande is aan de eerste en derde eis voorbijgegaan, beide zijn echter eenvoudig te controleren. In het geval dat de eerste eis geschonden wordt vormt de vector met op elke positie een 1 *een oplossing voor het separatieprobleem. In het geval dat de derde ongelijkheid voor een e geschonden wordt voldoet de vector met op de e-de positie een* 1 *en op alle andere posities een* 0*.*
Dit laat zien dat het separatieprobleem ook op te lossen is zonder de expliciete polytoop te gebruiken.

6. Een benadering van de Lovász ϑ functie

Zoals al eerder genoemd is het ellipsoïdealgoritme op een heel groot aantal problemen toepasbaar, in de praktijk wordt het echter vrijwel nooit geïmplementeerd. In dit hoofdstuk zullen we een toepassing van dit algoritme bekijken: het benaderen van de Lovász ϑ functie. We zullen allereerst de Lovász ϑ functie definiëren en kort uitleggen waarom dit een interessante functie is. Vervolgens tonen we aan dat dit probleem ook daadwerkelijk op te lossen is met het ellipsoïdealgoritme. Dit doen we door twee dingen te laten zien:

(1) de verzameling waarover geoptimaliseerd wordt is compact en convex, tevens zullen we een inwendig punt van deze verzameling geven,
(2) we geven het separatiealgoritme.

Daarna zullen we de implementatie in Matlab bespreken, hierbij zullen we vooral ingaan op de problemen die ontstonden vanwege de benodigde precisie en de oplossing daarvan. Tot slot zullen we van een aantal grafen de Lovász ϑ functie met de schatting vergelijken.

6.1. De Lovász ϑ functie.

De Lovász ϑ functie is een functie die aan een graaf $G = (V, E)$ een getal $\vartheta(G)$ toekent. Hierbij bedoelen we met $G = (V, E)$ de graaf G met puntenverzameling V en lijnenverzameling E. De elementen van E zijn van de vorm (i, j) waarbij $i, j \in V$. Met een graaf bedoelen we een ongerichte graaf, dus (i, j) vormt hier een ongeordend paar. Laat nu het aantal punten gelijk zijn aan n, dus $|V| = n$. Dan kunnen we de Lovász ϑ functie als volgt opschrijven:

$$
\begin{aligned}
\vartheta(G) = \max \quad & \langle J, X \rangle \\
\text{o.d.v.} \quad & \mathrm{Tr}(X) \leq 1 \\
& X_{ij} = 0 \qquad \text{als } (i, j) \in E \\
& X \succeq 0,
\end{aligned}
\tag{6.1}
$$

waarbij J de $n \times n$ matrix met op elke positie een 1 is. Dus het inproduct $\langle J, X \rangle$ is niets anders dan de som van alle coördinaten van X. Waarbij we als inproduct tussen twee $n \times n$ matrices het standaardinproduct in $\mathbb{R}^{n^2}$ gebruiken (door de matrices te beschouwen als vectoren in $\mathbb{R}^{n^2}$). $X_{ij} = 0$ als $(i, j) \in E$ wil zeggen dat we eisen dat de posities van X die corresponderen met een lijn in de graaf gelijk zijn aan 0. Verder eisen we nog dat X positief semidefiniet is. Het is dan ook een semidefiniet programma (SDP). In de volgende paragraaf zullen we aantonen dat dit programma op te lossen is met het ellipsoïdealgoritme, eerst zullen we echter nog wat achtergrondinformatie geven over deze functie.

Het is vanuit de definitie niet gelijk duidelijk waarom deze functie interessant is. Maar het blijkt dat $\vartheta(G)$ een belangrijk getal is voor een graaf aangezien het een sterke relatie heeft met een aantal andere belangrijke eigenschappen van een graaf: $\alpha(G)$, $\Theta(G)$ en $\omega(G)$. Deze getallen zijn allen zelf lastig te bepalen voor algemene grafen, de Lovász ϑ functie echter niet. We zullen hieronder kort behandelen wat deze getallen betekenen, hiervoor hebben we een aantal definities nodig.

Definitie 9 (Onafhankelijke verzameling)**.** *Laat $G = (V, E)$, een onafhankelijke verzameling in G is een verzameling $S \subseteq V$ zó dat voor alle $i, j \in S$ geldt $(i, j) \notin E$.*

We kunnen nu het onafhankelijkheidsgetal, $\alpha(G)$, van een graaf G definiëren:

Definitie 10 ($\alpha(G)$)**.** *Laat $G = (V, E)$ een graaf zijn. We definiëren het onafhankelijkheidsgetal van G, $\alpha(G)$, als de maximale grootte van een onafhankelijke verzameling in*

G. Dus

$$\alpha(G) := \max_{S \subseteq V}\{|S| : S \text{ is een onafhankelijke verzameling in } G\}$$

Hieraan gerelateerd zijn de volgende twee definities:

Definitie 11 (Kliek)**.** *Laat $G = (V, E)$ een graaf zijn. Een kliek in G is een verzameling $S \subseteq V$ zó dat voor alle $i, j \in S$ met $i \neq j$ geldt $(i, j) \in E$.*

Definitie 12 ($\omega(G)$)**.** *Laat $G = (V, E)$ een graaf zijn. We definiëren $\omega(G)$, als de maximale grootte van een kliek in G. Dus*

$$\omega(G) := \max_{S \subseteq V}\{|S| : S \text{ is een kliek in } G\}$$

Het derde getal $\Theta(G)$ wordt ook wel de Shannon-capaciteit van een graaf genoemd. Dit wordt als volgt gedefinieerd:

Definitie 13 (Shannon-capaciteit)**.** *Laat $G = (V, E)$ een graaf zijn. Dan is de Shannon-capaciteit van G gelijk aan*

$$\Theta(G) := \sup_{k \in \mathbb{N}} \left(\alpha(G^k)\right)^{\frac{1}{k}}.$$

Hierin is G^k gelijk aan het sterke graafproduct van G met zichzelf, k maal. Het sterke product van twee grafen G en H wordt hierbij gegeven door

- als puntenverzameling het carthesisch product van de puntenverzamelingen van G en H,
- $((u, u'), (v, v')) \in E(G \times H)$ dan en slechts dan als ($(u', v') \in E(H)$ of $u' = v'$) en ($(u, v) \in E(G)$ of $u = v$) .

De Shannon-capaciteit is een belangrijke eigenschap binnen de informatietheorie. Het geeft namelijk de effectieve grootte van een alfabet in een communicatiemodel weer wanneer G de graaf is die bij dat alfabet 'hoort'. De graaf G 'hoort' bij dat alfabet indien de punten van G precies de letters in het alfabet zijn en er een lijn tussen twee punten loopt indien deze letters tijdens de communicatie met elkaar te verwarren zijn. Letters zijn met elkaar te verwarren indien ze veel op elkaar lijken. Zo kan het in geschreven tekst bijvoorbeeld lastig zijn om 1 en l van elkaar te onderscheiden, in de graaf behorende bij dat alfabet zou er dus tussen 1 en l een lijn lopen. Voor meer informatie over de Shannon-capaciteit van een graaf wil ik verwijzen naar het bachelorproject van Eline Rietberg.
Het is niet moeilijk te zien dat $\alpha(G) \leq \Theta(G)$ voor elke graaf G. Lovász heeft ook aangetoond dat $\alpha(G) \leq \Theta(G) \leq \vartheta(G)$. Dus $\vartheta(G)$ is een bovengrens op de Shannon-capaciteit. Aangezien de Shannon-capaciteit van een graaf over het algemeen erg lastig te berekenen is is het heel handig om $\vartheta(G)$ te kennen. Natuurlijk zou het erg fijn zijn als we op de een of andere manier zouden kunnen aantonen dat $\vartheta(G) = \Theta(G)$, dit blijkt echter in het algemeen niet het geval te zijn. Voor een belangrijke klasse van grafen echter wel, de perfecte grafen, en dit is ook bewezen. We zullen nu definiëren wat een perfecte graaf is, hiervoor hebben we echter nog wel een andere definitie nodig.

Definitie 14 (Chromatisch getal)**.** *Het chromatisch getal van een graaf G, $\chi(G)$, is gelijk aan de kleinste k waarvoor de graaf k-kleurbaar is.*

Een graaf heet k-kleurbaar als er een kleuring van de graaf met k kleuren bestaat. Een k-kleuring van een graaf is een partitie van de puntenverzameling $(A_1 \cup A_2 \cup \ldots \cup A_k)$ zo dat $A_i \cap A_j = \emptyset$ voor $i \neq j$ en voor elke i is A_i een onafhankelijke verzameling. Met andere woorden: in een kleuring van een graaf hebben twee punten waartussen een lijn loopt niet dezelfde kleur.
We kunnen nu de definitie van een perfecte graaf geven:

Definitie 15 (Perfecte graaf)**.** *Een graaf heet perfect als $\chi(H) = \omega(H)$ voor alle geïnduceerde deelgrafen H van G.*

Een geïnduceerde deelgraaf van een graaf G is een graaf verkregen door een deelverzameling van de punten van G te pakken en alle lijnen tussen deze punten. Merk op dat $\omega(G) \leq \chi(G)$ voor elke G, het is immers niet moeilijk in te zien dat je minstens $\omega(G)$ kleuren nodig hebt om de graaf te kleuren. Alle punten in een kleurklasse zijn immers onafhankelijk en er bestaat een kliek van grootte $\omega(G)$. Een graaf is dus perfect indien voor elke geïnduceerde deelgraaf H geldt dat de triviale ondergrens van $\omega(H)$ kleuren ook echt voldoende is om de deelgraaf mee te kleuren.

Voorbeeld 7 (Een perfecte graaf)**.** *Laat $G = \{\{1,2,3,4\}, \{(1,2),(2,3),(3,4),(1,4)\}\}$, de viercykel. Laten we voor het gemak deze graaf even voorstellen als een vierkant. Het is duidelijk dat de hoekpunten van het vierkant gekleurd kunnen worden met twee kleuren, de punten op een diagonaal kunnen dezelfde kleur krijgen. Bovendien is de maximale grootte van een kliek gelijk aan 2, elke lijn vormt met zijn eindpunten immers een kliek van grootte 2 (groter is niet mogelijk: de diagonalen vormen geen lijnen). Verwijderen we één punt uit deze graaf dan veranderd dit duidelijk niets aan het kliekgetal. Bovendien kan de kleuring van de viercykel met 2 kleuren ook worden gebruikt om deze deelgraaf te kleuren met 2 kleuren. Indien we twee punten verwijderen kunnen er twee gevallen ontstaan: de overgebleven punten liggen op een diagonaal van het oorspronkelijke vierkant of de overgebleven punten liggen op één zijde van het oorspronkelijke vierkant. In het eerste geval is het kliekgetal 1 en het is niet moeilijk te zien dat de twee punten met 1 kleur te kleuren zijn, ze zijn immers onafhankelijk. In het tweede geval zijn er duidelijk 2 kleuren nodig en zoals al eerder opgemerkt vormen de twee punten een kliek van grootte 2. Verwijderen we drie punten dan blijft er slechts één punt over. Één punt is duidelijk met 1 kleur te kleuren en het vormt ook een kliek van grootte 1. Verwijderen we alle vier de punten dan blijft de lege graaf over, die is per definitie perfect. We kunnen dus concluderen dat voor elke geïnduceerde deelgraaf H van de viercykel geldt $\chi(H) = \omega(H)$, dus de viercykel is perfect.*

We zullen nu laten zien dat voor een perfecte graaf geldt $\vartheta(G) = \Theta(G)$. Dit volgt direct uit een bekende stelling van Lovász, voor we deze kunnen formuleren hebben we echter nog een andere definitie nodig. We moeten weten wat het complement van een graaf is.

Definitie 16 (Complement)**.** *Het complement van een graaf $G = (V, E)$ is gedefinieerd als $\overline{G} = (V, E^c)$ waarbij $(i,j) \in E^c$ desda $(i,j) \notin E$.*

Dus het complement van een graaf bestaat uit dezelfde punten en precies alle lijnen die niet in de oorspronkelijke graaf zitten.
Lovász heeft nu de volgende stelling geformuleerd:

Stelling 4 (Lovász "sandwich"stelling)**.** *Voor elke graaf G geldt $\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G})$.*

Alles bij elkaar geeft ons dat $\alpha(G) \leq \Theta(G) \leq \vartheta(G) \leq \chi(\overline{G})$. Indien we kunnen laten zien dat $\alpha(G) = \chi(\overline{G})$ voor perfecte grafen dan zijn we klaar, dan volgt immers direct $\alpha(G) = \Theta(G) = \vartheta(G) = \chi(\overline{G})$.
Berge heeft in 1963 het *Strong perfect graph conjecture* uitgesproken, dit werd pas in 2006 bewezen door Chudnovsky, Robertson, Seymour en Thomas (Stelling 5.5.3 in [6]).

Stelling 5 (Sterke perfecte graaf stelling, Chudnovsky, Robertson, Seymour en Thomas 2006)**.** *Een graaf G is perfect dan en slechts dan als zowel G als $\overline{G}$ geen oneven cykel van lengte ten minste 5 bevat als geïnduceerde deelgraaf.*

Merk op dat we met deze stelling eenvoudig kunnen zien dat de graaf in Voorbeeld 7 perfect is, de viercykel heeft immers maar vier punten.
Een direct gevolg van de sterke perfecte graaf stelling staat bekend als de perfecte graaf stelling. Deze stelling werd ook door Berge als een vermoeden uitgesproken, maar hij werd al in 1972 door Lovász bewezen (Stelling 5.5.4 in [6]):

Stelling 6 (Perfecte graaf stelling, Lovász 1972)**.** *Een graaf is perfect dan en slechts dan als zijn complement perfect is.*

Het is niet moeilijk in te zien dat voor elke graaf geldt $\alpha(G) = \omega(\overline{G})$ en $\alpha(\overline{G}) = \omega(G)$. Nemen we nu aan dat G perfect is dan geldt wegens de perfecte graaf stelling dat ook $\overline{G}$ perfect is en dus $\omega(\overline{G}) = \chi(\overline{G})$. Hieruit volgt $\alpha(G) = \chi(\overline{G})$.
Oftewel voor perfecte grafen geldt $\alpha(G) = \Theta(G) = \vartheta(G)$. Dus voor perfecte grafen is de Shannon-capaciteit te berekenen door $\vartheta(G)$ te bepalen. Merk op: in het bijzonder geldt dat $\vartheta(G)$ een geheel getal is!

6.2. **Is de Lovász ϑ functie te benaderen met het ellipsoïdealgoritme?** We zullen nu laten zien dat de Lovász ϑ functie te benaderen is met het ellipsoïdealgoritme. Merk allereerst op dat de doelfunctie van het SDP inderdaad lineair is. Neem nu een graaf $G = (V, E)$ en laat

$$K := \{X \in \mathbb{R}^{n\times n} : \mathrm{Tr}(X) \leq 1, X_{ij} = 0 \text{ als } (i,j) \in E, X \succeq 0\}. \tag{6.2}$$

De matrix X leeft in $\mathbb{R}^{n\times n}$, dus het zou voor de hand liggen om als dimensie voor het probleem n^2 te nemen. Dit is echter niet toegestaan, het ellipsoïdealgoritme werkt met een volledigdimensionale compacte en convexe verzameling. De verzameling K is echter niet volledig dimensionaal. We weten namelijk dat $X \succeq 0$, dus X is symmetrisch. Dat wil zeggen dat er hooguit $n + \binom{n}{2}$ variabelen in X zitten. Tevens hebben we op de posities X_{ij} waarvoor $(i,j) \in E$ de waarde van X vast gelegd. Dat betekent dat er nog eens $|E|$ variabelen niet langer variabel zijn. Door de affiene deelruimte te bekijken waar K in ligt kunnen we dan ook zeggen dat de dimensie van het probleem $n + \binom{n}{2} - |E|$ is. Deze affiene deelruimte zullen we noteren met $\mathbb{R}^{(n\times n)^*}$.
We moeten nu nog een aantal dingen laten zien om aan te tonen dat we de Lovász ϑ functie kunnen benaderen met het algoritme:

(1) K is een convexe en compacte verzameling.
(2) Er bestaat een $X_0 \in K$ en $r, R \in \mathbb{R}$ zo dat $S(X_0, r) \cap \mathbb{R}^{(n\times n)^*} \subseteq K \subseteq S(X_0, R)$.
(3) Er bestaat een separatiealgoritme voor K.

We zullen nu laten zien dat K convex en compact is.

Lemma 7. *De verzameling K is convex.*

Bewijs. Laat $A, B \in K$ en $\lambda \in \mathbb{R}$ met $0 \leq \lambda \leq 1$. We zullen laten zien dat $\lambda A + (1-\lambda)B \in K$. We weten dat $A, B \succeq 0$ en dus volgens Gevolg 1 is ook $\lambda A + (1-\lambda)B$ positief semidefiniet. Verder geldt $\mathrm{Tr}(A) \leq 1$ en $\mathrm{Tr}(B) \leq 1$ dus ook $\mathrm{Tr}(\lambda A + (1-\lambda)B) = \lambda \mathrm{Tr}(A) + (1-\lambda)\mathrm{Tr}(B) \leq \lambda + (1-\lambda) = 1$
Laat nu i, j zo dat $(i,j) \in E$. Dan geldt $A_{ij} = 0 = B_{ij}$ en dus ook $(\lambda A + (1-\lambda)B)_{ij} = \lambda A_{ij} + (1-\lambda)B_{ij} = 0$. Dit laat zien dat $\lambda A + (1-\lambda)B \in K$. Dus K is convex. □

Om te laten zien dat K compact is maken we wederom gebruik van de stelling van *Heine-Borel*, Stelling 1.4 van [3], die zegt ons dat een deelverzameling van $\mathbb{R}^k$ compact is dan en slechts dan als hij gesloten en begrensd is. Dit laatste zullen we van de verzameling K aantonen.

Lemma 8. *De verzameling K is gesloten.*

Bewijs. Aldus neem een convergent rijtje $X_1, X_2, \ldots$ in K. Merk op dat dit convergentie ten op zichte van de de matrixnorm is, maar dat impliceert ook dat we coördinaatsgewijze convergentie hebben. We zullen laten zien dat $X := \lim_{n\to\infty} X_n$ in K ligt. Merk op dat aangezien $\mathrm{Tr}(X_n) \leq 1$ voor alle n eenvoudig volgt dat ook $\mathrm{Tr}(X) \leq 1$. Evenzo volgt eenvoudig dat voor de posities X_{ij} waarvoor $(i,j) \in E$ geldt dat $X_{ij} = \lim_{n\to\infty}(X_n)_{ij} = 0$. Aangezien we coördinaatgewijze convergentie hebben volgt eenvoudig dat X symmetrisch is. Immers geldt $X_{ij} = \lim_{n\to\infty}(X_n)_{ij} = \lim_{n\to\infty}(X_n)_{ji} = X_{ji}$ waarbij we de symmetrie van ieder van de X_n hebben gebruikt. Omdat $\|X_n - X\| \to 0$ en elk van de eigenwaarden van X_n groter of gelijk aan 0 zijn (voor iedere n) volgt dat ook de eigenwaarden van X

allen groter of gelijk aan 0 zijn. Dus $X \succeq 0$. Dit alles wil dus zeggen dat $X \in K$. Maar dan is K dus gesloten. □

We zullen nu laten zien dat K ook begrensd is.

Lemma 9. *De verzameling K is begrensd.*

Bewijs. Laat $X \in K$. We weten $\mathrm{Tr}(X) = \lambda_1 + \lambda_2 + \ldots + \lambda_n \leq 1$ waarbij $\lambda_1, \lambda_2, \ldots, \lambda_n$ de eigenwaarden van X zijn. Aangezien $\lambda_i \geq 0$ voor alle i, X is immers positief semidefiniet, wil dit zeggen dat $\lambda_i \leq 1$ voor alle i. Maar dat wil dus zeggen dat $\|X\| \leq 1$. Dit laat zien dat K een deelverzameling van de eenheidsbol in $\mathbb{R}^{n\times n}$ is, en dus is K begrensd. □

De stelling van *Heine-Borel* geeft ons dus dat K compact is.
Gezien het vorige lemma is het wel duidelijk dat er voor elke $X_0 \in K$ een $R \in \mathbb{R}$ bestaat zo dat $K \subseteq S(X_0, R)$. We zullen nu laten zien dat we ook een X_0 kunnen vinden waarvoor er een $r \in \mathbb{R}$ bestaat zo dat $S(X_0, r) \cap \mathbb{R}^{(n\times n)^*} \subseteq K$.

Bewering 3. *Voor $X_0 := \frac{1}{2n} I$ bestaat er een $r \in \mathbb{R}$ zo dat $S(X_0, r) \cap \mathbb{R}^{(n\times n)^*} \subseteq K$.* [1]

Bewijs. Merk allereerst op dat X_0 inderdaad in K ligt. Immers is het eenvoudig te zien dat $\mathrm{Tr}(X_0) = \frac{1}{2} \leq 1$. $(X_0)_{ij} = 0$ voor $i \neq j$ en voor alle i, j waarvoor $(i, j) \in E$ geldt in ieder geval $i \neq j$ (lijnen van een punt naar zichzelf zijn niet toegestaan). X_0 is symmetrisch en alle eigenwaarden van X_0 zijn $\frac{1}{2n}$ en dus groter dan 0, dus $X_0 \succeq 0$.
We zullen nu laten zien dat er een $r \in \mathbb{R}$ bestaat zo dat $S(X_0, r) \cap \mathbb{R}^{(n\times n)^*} \subseteq K$. We kunnen hiervoor de stelling van *Gershgorin* gebruiken (zie bladzijde 106 van [7]). Die geeft ons dat alle eigenwaarden in de vereniging van cirkels met straal $\sum_{j=1, j\neq i}^{n} |X_{ij}|$ rond de punten X_{ii} liggen. Kiezen we nu r zo dat $\frac{1}{2n} - |X_{ii}| \geq \sum_{j=1, j\neq i}^{n} |X_{ij}|$ voor alle $X_0 + X \in S(X_0, r) \cap \mathbb{R}^{(n\times n)^*}$ dan zien we dat geldt dat alle eigenwaarden van $X_0 + X$ groter of gelijk aan 0 zijn en dus dat $X_0 + X \succeq 0$.
Omdat $X_0 + X \in S(X_0, r)$ weten we dat $\|X\| \leq r$ en dus $|X_{ij}| \leq r$ voor alle i, j. Kies nu $r = \frac{1}{2n^2}$. Dan zien we $\frac{1}{2n} - \sum_{j=1}^{n} |X_{ij}| \geq \frac{1}{2n} - n\frac{1}{2n^2} = 0$. En dus geldt de gewenste ongelijkheid. *Gershgorin* geeft ons dus dat $X_0 + X \succeq 0$.
We moeten nu nog laten zien dat $\mathrm{Tr}(X_0 + X) \leq 1$ voor deze r. Maar dit volgt eenvoudig. We zien immers op eenzelfde manier $\mathrm{Tr}(X) = \sum_{i=1}^{n} X_{ii} \leq \sum_{i=1}^{n} |X_{ii}| \leq nr = \frac{1}{2n}$. En dus, gebruikmakend van de lineariteit van het spoor, zien we $\mathrm{Tr}(X_0 + X) = \mathrm{Tr}(X_0) + \mathrm{Tr}(X) = \frac{1}{2} + \mathrm{Tr}(X) \leq \frac{1}{2} + \frac{1}{2n} \leq 1$.
Maar dan volgt dus dat $X_0 + X \in K$. □

We zullen nu laten zien dat er ook een separatiealgoritme voor K bestaat. We zullen er hierbij weer vanuit gaan dat alle matrices in de deelruimte $\mathbb{R}^{(n\times n)^*}$ liggen. We moeten dus een algoritme geven wat gegeven een $X \in \mathbb{R}^{(n\times n)^*}$ bepaalt of $X \in K$. Als $X \notin K$ dan moet het algoritme tevens een vector d geven met $\|d\| \geq 1$ en $\max\{\langle d, Y\rangle : Y \in K\} \leq \langle d, X\rangle$. Aangezien elke vector d te normaliseren is zonder de ongelijkheid te veranderen zullen we met de eis $\|d\| \geq 1$ geen rekening houden.
We zullen eerst het algoritme geven, vervolgens zullen we laten zien dat het voldoet.

(1) Bereken $\mathrm{Tr}(X)$. Als $\mathrm{Tr}(X) > 1$ laat dan $d = I$. Als $\mathrm{Tr}(X) \leq 1$ ga verder naar 2.

[1] Merk op dat we in deze bewering de euclidische afstand in $\mathbb{R}^{n+n(n-1)/2-|E|}$ gebruiken, zoals later ook in de implementatie in Matlab

(2) Veeg X symmetrisch. Dit geeft je een diagonaalmatrix D en een matrix P zo dat $D = PXP^T$. Als $D_{ii} < 0$ voor een i laat dan $x = Pe_i$ en $d = -xx^T$, waarbij e_i de i-de eenheidsvector in $\mathbb{R}^n$ is. Als $D_{ii} \geq 0$ voor alle i concludeer dan dat $X \in K$.

We zullen nu laten zien dat dit algoritme inderdaad een separatiealgoritme is voor K. Indien in stap 1 wordt gevonden dat $\mathrm{Tr}(X) > 1$ dan geldt duidelijk $X \notin K$. Nemen we nu $d = I$ dan zien we $\langle d, X\rangle > 1 \geq \max\{\langle d, Y\rangle : Y \in K\}$. Immers is $\langle I, X\rangle = \mathrm{Tr}(X)$ en voor alle $Y \in K$ geldt $\mathrm{Tr}(Y) \leq 1$. Nu stap 2. Als $D_{ii} \geq 0$ voor alle i dan is eenvoudig te zien dat $y^T Xy \geq 0$ voor alle $y \in \mathbb{R}^n$ (op eenzelfde manier als in het bewijs van Lemma 1) en dus is $X \succeq 0$ en daarom geldt $X \in K$. Als $D_{ii} < 0$ voor een i dan kunnen we een vector $y \in \mathbb{R}^n$ vinden waarvoor $y^T Xy < 0$. Nemen we de i-de kolom van de matrix P als vector y dan zien we $\langle yy^T, X\rangle = y^T Xy = D_{ii} < 0$. En voor alle positief definiete matrices Y geldt $\langle yy^T, Y\rangle \geq 0$. Dus in dit geval is $d = -yy^T$ een geschikte vector. Merk op dat $y = Pe_i$.
Dus het gegeven algoritme vormt inderdaad een separatiealgoritme voor K. We hebben nu dus laten zien dat K compact en convex is. We hebben stralen $r, R \in \mathbb{R}$ en een punt X_0 gegeven zo dat $S(X_0, r) \cap \mathbb{R}^{(n\times n)^*} \subseteq K \subseteq S(X_0, R)$. En we hebben een separatiealgoritme voor K gegeven. Maar dat betekent dat we aan alle voorwaarden van het ellipsoïdealgoritme voldoen. Dus dit probleem is benaderbaar met het ellipsoïdealgoritme, dat wil zeggen we kunnen $\vartheta(G)$ bepalen op een ϵ nauwkeurig!

6.3. **De implementatie.** In deze paragraaf zullen we bespreken hoe we het ellipsoïdealgoritme geïmplementeerd hebben om de Lovász ϑ functie tot op een epsilon nauwkeurig te bepalen. De implemenatie vindt plaats in het bekende rekenprogramma Matlab. We zullen allereerst uitleggen hoe we een gegeven matrix transformeren naar een vector in $\mathbb{R}^{n+n(n-1)/2-|E|}$ waarbij $G = (V, E)$, dit zal ook gelijk duidelijk maken wat de input van het programma is en hoe de doelfunctie omgezet wordt naar een vector. Vervolgens zullen we de initialisatie van het algoritme bespreken. Tot slot hebben we het nog over de problemen die ontstonden vanwege de precisie van Matlab en de oplossing daarvan. De Matlab code is te vinden in Hoofdstuk 8.

Laat een graaf $G = (V, E)$ gegeven zijn. Merk op dat een representatie van een graaf zijn adjacency matrix is (de adjacency matrix van een graaf is een $|V| \times |V|$ matrix A met $A_{ij} = 1$ als $(i, j) \in E$ en $A_{ij} = 0$ anders). We zullen er dan ook vanuit gaan dat een graaf op die manier gegeven is. Dus we krijgen een n en een $n \times n$ matrix. Aangezien elke positief semidefiniete matrix symmetrisch is beschouwen we alleen de posities boven de diagonaal. Maak nu van een matrix $X \in \mathbb{R}^{(n\times n)^*}$ de vector met op de eerste n posities de hoofddiagonaal van X, op de volgende $n-1$ de eerste diagonaal, op de volgende $n-2$ de tweede diagonaal, enzovoort. We verkrijgen zo een vector $X' \in \mathbb{R}^{n+n(n-1)/2}$. We zijn echter nog niet klaar, we willen namelijk een vector in $\mathbb{R}^{n+n(n-1)/2-|E|}$. Volg nu hetzelfde proces als hiervoor met de adjacency matrix[2] A, dit geeft een vector $A' \in \mathbb{R}^{n+n(n-1)/2}$. Merk op dat $\sum_i A'_i$ precies het aantal takken in de graaf G is, dus $\sum_i A'_i = |E|$. Maak nu van X' de nieuwe vector X'' door elke positie van X' waarop A' op dezelfde positie gelijk is aan 1 te verwijderen. Merk op dat $X'' \in \mathbb{R}^{n+n(n-1)/2-|E|}$, dit is dus een vector in de gewenste dimensie die alle relevante informatie van X weergeeft.

[2]Aangezien het eenvoudiger is om de vector A' en n te geven zal dit de daadwerkelijke input van het programma zijn.

Nu de doelfunctie: $\langle J, X\rangle$, met J de matrix met op elke positie een 1. We zien dat we die ook kunnen schrijven als $\langle C, X\rangle$ waarbij

$$C := \begin{pmatrix} 1 & 2 & \cdots & \cdots & 2 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & 2 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

Aangezien X symmetrisch is. Voeren we nu hetzelfde proces uit op C als eerder op X en A dan vinden we de kostenvector c in $\mathbb{R}^{n+n(n-1)-|E|}$ met op de eerste n posities een 1 en op de overige posities een 2. We zien nu dat $\langle J, X\rangle = \langle C, X\rangle = \langle c, X''\rangle$, waarbij het laatste inproduct het standaardinproduct in $\mathbb{R}^{n+n(n-1)-|E|}$ is. De laatste gelijkheid volgt uit het gegeven dat iedere $X \in \mathbb{R}^{(n\times n)^*}$ gelijk is aan 0 op de posities die verwijderd zijn.
Laten we nu naar de initialisatie van het algoritme kijken. De input is bekend, die hebben we hiervoor beschreven. Natuurlijk moet er ook nog een epsilon gekozen worden. Vervolgens kiezen we x_0 zoals in de vorige paragraaf: x_0 is de vector behorende bij de matrix $\frac{1}{2n}I$. De straal r kiezen we ook zoals in de vorige paragraaf, dus $r = \frac{1}{2n^2}$. Ons rest nu nog om de initiële ellips te geven. We zullen hiervoor een cirkel met straal R nemen rond het punt x_0. Waarbij $R = n + n(n-1)/2 - |E|$, oftewel de dimensie van het probleem. De reden dat we hiervoor niet dezelfde straal als in Lemma 9 kunnen nemen is dat we daar de matrixnorm hadden gebruikt en we nu met de euclidische lengte in $\mathbb{R}^{n+n(n-1)-|E|}$ werken. Waarom dan deze straal? Omdat we kunnen laten zien dat voor iedere belangrijke $X \in K$ geldt $0 \leq X_{ij} \leq 1$ voor alle i, j. En dus geldt voor de euclidische lengte van de vector x behorende bij X dat $\|x\| \leq \sqrt{n + n(n-1)/2 - |E|}$, om niet gelijk te beginnen met het berekenen van een wortel (nauwkeurigheid) nemen we als straal R het kwadraat hiervan, oftewel de dimensie. We zullen nu laten zien dat voor iedere interessante $X \in K$ geldt dat $0 \leq X_{ij} \leq 1$ voor alle i, j.

Bewering 4. *Het is voldoende om $X \in K$ te beschouwen met $X_{ij} \geq 0$ voor alle i, j, bovendien geldt $X_{ij} \leq 1$ voor alle i, j.*

Bewijs. Allereerst stel dat een $A \in K$ negatieve posities heeft, maak dan de matrix A' door op elke positie de absolute waarde te nemen. Merk op dat A' weer symmetrisch is. We weten $0 \leq x^T Ax$ en $x^T Ax \leq x^T A'x$ dus ook $0 \leq x^T A'x$ voor alle $x \in \mathbb{R}^n$. Dus ook $A' \succeq 0$. Het is duidelijk dat $\langle J, A\rangle < \langle J, A'\rangle$, dus door A buiten beschouwing te laten veranderen we niets aan het optimum. Neem nu een $A \in K$ die op elke positie groter of gelijk aan 0 is. We zullen nu laten zien dat $A_{ij} \leq 1$. Merk allereerst op dat $A_{ii} \leq 1$, aangezien $\mathrm{Tr}(A) \leq 1$.
Laat $x \in \mathbb{R}^n$ de vector zijn met op de i-de positie een 1 en op de j-de positie een -1, op alle andere posities een 0. Aangezien A positief semidefiniet is weten we $x^T Ax \geq 0$. Maar $x^T Ax = \sum_{i,j} A_{ij}x_i x_j = A_{ii} + A_{jj} - 2A_{ij}$ dus $A_{ii} + A_{jj} - 2A_{ij} \geq 0$. We weten dat $A_{ii} \leq 1$, dus er geldt ook $2 - 2A_{ij} \geq A_{ii} + A_{jj} - 2A_{ij} \geq 0$. Maar dan volgt $A_{ij} \leq 1$. □

We hebben dus nu laten zien dat we voor alle interessante $X \in K$ een straal R kunnen vinden zodat al die $X \in S(X_0, R)$. Strikt gesproken kunnen we dus niet zeggen dat we met deze initialisatie heel K beschouwen, we beschouwen in feite $K' = \{X \in K : X_{ij} \geq 0 \text{ voor alle } i, j\}$. Zoals zojuist opgemerkt veranderd dit echter niets aan het optimum! En dus kunnen we nog steeds zeggen dat we $\vartheta(G)$

berekenen. Merk overigens op dat K' evenwel compact en convex is, aangezien we K' uit K verkrijgen door te doorsnijden met een eindig aantal gesloten halfruimten.

De implementatie in Matlab zorgde voor een aantal problemen, met name vanwege de benodigde precisie. De benodigde precisie is namelijk nogal groot. In Tabel 1 staan de waarden voor N en p (het aantal stappen en de benodigde precisie in binaire cijfers) gegeven voor een aantal waarden van n. n is hier de dimensie van het probleem, oftewel het aantal vrije variabelen in X. Bijvoorbeeld voor de lege graaf op 4 punten heb je al $n = 4 + 4 \cdot 3/2 = 10$. Om de waarden in deze tabel te berekenen nemen we $R = n$, $r = \frac{1}{2n^2}$, $\|c\| = 1$ en $\epsilon = \frac{1}{100}$. Merk op dat deze r iets kleiner is dan de eerder genoemde r, de reden dat we de andere r niet gebruiken is omdat we dan nog een parameter nodig hebben (namelijk het aantal punten in de graaf). Herinner $N = 5n^2\lceil \log \frac{2R^2\|c\|}{r\epsilon} \rceil$, vullen we de gekozen waarden in dan vinden we $N = 5n^2\lceil \log(100 \cdot (2n^2)^2) \rceil$. Tevens geldt $p = 5N$. We kunnen zien dat de precisie enorm snel toeneemt, zelfs voor kleine waarden

n	N	p
1	24	120
2	144	720
5	1300	6500
10	6400	32000
20	28800	144000

TABEL 1. Benodigde aantal stappen en precisie (in binaire cijfers) gegeven $R = n$, $r = \frac{1}{2n^2}$, $\|c\| = 1$ en $\epsilon = \frac{1}{100}$

van n. Zo zien we dat we voor de lege graaf op 4 punten bijvoorbeeld al een precisie van 32000 binaire cijfers nodig hebben.
Matlab werkt standaard met een precisie van 10^{-16}. Dat betekent dat we met de standaard precisie van Matlab het probleem nog niet eens zouden kunnen oplossen voor $n = 1$, immers $2^{-120} \approx 10^{-36} < 10^{-16}$ (terwijl het probleem voor $n = 1$ vrij triviaal is). We zouden nu twee dingen kunnen doen: (1) een ander rekenprogramma gebruiken waarin de precisie makkelijk aan te passen is, of (2) onze eisen bijstellen voor het vinden van een geschikte oplossing. Na een aantal simpele tests in Matlab bleek dat het huidige programma al vrij lang over de uitvoering van het algoritme doet (meer hierover in het volgende hoofdstuk). Dit dus met een 'lage' precisie, indien we de precisie enorm vergroten dan zal dat de rekentijd natuurlijk niet ten goede komen. Optie (1) lijkt dus niet de meest verstandige keuze.
Wat de tests verder nog lieten zien was dat we vanwege de verminderde precisie vaak op een (fatale) foutmelding liepen in het berekenen van de vector b, hierdoor stopte het algoritme voortijdig. Herinner

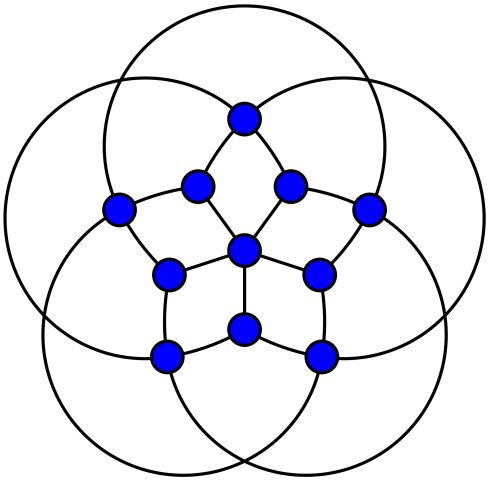
$$b_k = \frac{A_k a}{\sqrt{a^T A_k a}}.$$

De fout die we te zien krijgen was een 'NaN'-fout, oftewel Not-a-Number, de reden hiervoor was dat Matlab probeerde door 0 te delen. Maar door 0 delen wil zeggen dat $a^T A_k a = 0$, oftewel de matrix A_k heeft een eigenwaarde gelijk aan 0. Maar dat wil zeggen dat de ellips in stap k geen ellips in n dimensies is, maar een ellips in $n - 1$ dimensies! Plastisch gezegd: de ellips is platgedrukt. Maar in Lemma 4 hadden we juist laten zien dat elke eigenwaarde groter dan 0 zou

moeten zijn van iedere A_i met $0 \leq i \leq N$, kennelijk is een van de eigenwaarden van A_k zo klein geworden dat deze onder de precisie van Matlab valt en afgerond wordt naar 0. Dit gebeurde altijd in een stap waarin $x_k \in K$, oftewel $a = c$. De grote vraag is natuurlijk: kunnen we nog wat met de uitkomst die het algoritme dan geeft? Het antwoord is gelukkig ja. Als in de laatste stap inderdaad $x_k \in K$ hebt dan volgt dat $c^T A_k c = 0$, oftewel c is een eigenvector bij eigenwaarde '0'. Maar dat betekent dat c met de juiste precisie een eigenvector was geweest bij een eigenwaarde $\lambda_0 < 10^{-16}$. Herinner dat de 'dikte' van een ellips in de richting van een eigenvector gegeven wordt door de wortel van de eigenwaarde behorende bij die eigenvector, dus in de richting gegeven door de vector c is de 'dikte' maximaal 10^{-8}. Als we er van uit gaan dat we nog wel aan Lemma 5 voldoen, dus $K_i \subseteq E_i$ voor $0 \leq i \leq k$, dan volgt hieruit dat $c^T x_k > c^T x^* - \|c\| \cdot 10^{-8}$ waarbij x^* de optimale x in K is (wederom gebruikmakend van de ongelijkheid van *Cauchy-Schwarz*). Dat wil zeggen dat we het optimum tot op $\epsilon = \|c\| \cdot 10^{-8}$ kunnen benaderen met het algoritme! (bovendien blijkt het in de praktijk vaak nog wel mogelijk om met de uiteindelijke x_k door middel van slim afronden handmatig tot het echte optimum te komen)
Natuurlijk geldt dit alleen als we nog steeds aan Lemma 5 voldoen. Het ligt voor de hand dat we niet voor alle dimensies aan dit Lemma blijven voldoen. We zullen nu dan ook aangeven voor welke dimensies we nog wel aan dit Lemma voldoen. Kijken we terug naar het bewijs van Lemma 5 dan zien we dat het bewijs goed gaat zolang voor de restterm R_1 geldt $R_1 < \frac{1}{12n^2}$. Waarbij R_1 ongeveer gelijk was aan $n^2 2^{-p}$. In ons geval is de precisie niet 2^{-p} maar 10^{-16}. De vergelijking waar we dus aan moeten voldoen is $n^2 \cdot 10^{-16} < \frac{1}{12n^2}$. Gelijkheid geldt als $n^4 = \frac{10^{16}}{12}$, dus $n \approx 5372.9$ (waarbij we gebruiken dat $n > 0$). Het is eenvoudig te zien dat voor $n \leq 5372$ de ongelijkheid geldt. Dus dat wil zeggen dat de dimensie van ons probleem 5372 mag zijn willen we het probleem nog goed kunnen oplossen.

Samengevat: in de implementatie is er een regel toegevoegd die controleert of $b_k = \text{NaN}$ (regel 42 in de code), als dat zo is stopt het algoritme. Vervolgens kunnen we bekijken of $x_k \in K$, als dat zo is kunnen we concluderen dat $c^T x_k > c^T x^* - \|c\| \cdot 10^{-8}$. Als het algoritme niet voortijdig stopt kunnen we natuurlijk gewoon concluderen dat het gevonden optimum binnen ϵ van het echte optimum af ligt.

6.4. **Resultaten.** Met behulp van de implementatie heb ik voor een aantal grafen de Lovász ϑ functie bepaald. Allereerst voor de volledige graaf op 5 punten (K_5) en het complement daarvan, de lege graaf op 5 punten ($\overline{K_5}$). Deze grafen zijn interessant als 'test' voor het algoritme omdat het met behulp van Lovász' sandwich stelling (Stelling 4) eenvoudig is om de Lovász ϑ functie te bepalen. Het is immers eenvoudig te zien dat $\alpha(K_5) = 1 = \chi(\overline{K_5})$, dus ook $\vartheta(K_5) = 1$. En $\alpha(\overline{K_5}) = 5 = \chi(K_5)$ dus $\vartheta(\overline{K_5}) = 5$. Natuurlijk zijn dit niet de enige twee grafen waarvoor de Lovász ϑ functie eenvoudig te bepalen is. Voor elke perfecte graaf geldt immers $\alpha(G) = \vartheta(G)$, dus als $\alpha(G)$ bekend is is $\vartheta(G)$ dat ook. Dit zijn slechts twee voorbeelden. Het is natuurlijk ook interessant om te zien hoe het algoritme het doet als de Lovász ϑ functie niet geheeltallig is. Lovász heeft ooit laten zien dat voor de 5-cykel, C_5, geldt dat $\vartheta(C_5) = \sqrt{5}$. Dat zal dus de volgende graaf zijn die we bekijken. Vervolgens heeft Eline Rietberg een graaf voorgesteld op 11 punten, de *Grötzsch graph*, zie ook Figuur 4. Van deze graaf is het eenvoudig te zien dat α groter of gelijk aan 5 is, haar doel was om de Shannon-capaciteit te bepalen. Als dus de ϑ van deze graaf gelijk is aan 5 zou dat betekenen dat de Shannon-capaciteit ook gelijk is aan 5.



FIGUUR 4. De *Grötzsch graph*.

Graaf (G)	$\vartheta(G)$	Optimum	$\vartheta(G)$-Optimum	Rekentijd (sec.)
K_5	1	0.999999984595351	$1.5405 \cdot 10^{-8}$	0.649813
$\overline{K_5}$	5	4.999999995883318	$4.1167 \cdot 10^{-9}$	3.739331
C_5	$\sqrt{5}$	2.236067977478152	$2.1638 \cdot 10^{-11}$	1.573533
Grötzsch-graaf	'5'	4.999999998509743	$1.4902 \cdot 10^{-9}$	144.574803

TABEL 2. Voor een aantal grafen de Lovász ϑ functie (indien bekend), het optimum gevonden met het algoritme, het verschil hiertussen en de benodigde rekentijd (in seconden).

In Tabel 2 zijn de resultaten gevonden met Matlab te zien[3]. Merk op dat er bij de laatste graaf in de tweede kolom aanhalingstekens om de waarde van $\vartheta(G)$ staan, dit omdat dit slechts een vermoeden is (wat wel in zekere mate bevestigd wordt door het algoritme). Tevens hebben we in de vierde kolom het verschil tussen de echte waarde en het berekende optimum bepaald, dit hebben we echter zonder gedaan zonder de absolute waarde te nemen. Dat is verantwoord aangezien $\vartheta(G)$ het maximum over een verzameling is, het optimum van het algoritme wordt bepaald door een maximum over een eindig aantal punten uit deze verzameling te nemen. Het optimum kan dus niet groter zijn dan $\vartheta(G)$.
Een aantal dingen zijn interessant om op te merken. Wat in de tabel niet te zien is is het theoretisch benodigd aantal stappen en het uiteindelijke aantal stappen, deze zijn op zichzelf immers niet zo interessant. Wat hieraan natuurlijk wel interessant is is dat ze in geen van de onderzochte gevallen overeenkwamen. Oftewel het algoritme stopt in elk van deze gevallen voortijdig[4]. Een eenvoudige controle wees uit dat er in de laatste stap voordat het algoritme afgebroken werd telkens een toegelaten oplossing gevonden was. Zoals in de vorige paragraaf ook al vermeld kunnen we dan alsnog concluderen dat we een 'goede' oplossing hebben gevonden en wel bij $\epsilon = \|c\| \cdot 10^{-8}$. Dit wordt ook bevestigd door de

[3]De initialisatie is zoals eerder besproken. De gekozen epsilon is gelijk aan $\frac{1}{100}$, maar we zullen zo zien dat dit niet echt van belang is.

[4]In het licht van de vorige paragraaf is het nog even belangrijk om op te merken dat deze grafen allen minder dan 5372 variabelen geven, we voldoen dus nog steeds aan Lemma 5.

vierde kolom, hier zien we dat het verschil in elk van de gevallen kleiner is dan $\|c\| \cdot 10^{-8}$. Voor K_5 geldt immers $c = (1,1,1,1,1)$, dus $\|c\| = \sqrt{5}$ en $1.5405 < \sqrt{5}$. Voor de andere grafen geldt zeker $\|c\| > 1$ en we zien dat het verschil daar kleiner is dan 10^{-8}.
Vervolgens kunnen we nog even naar de benodigde rekentijd kijken. We zien dat de graaf met de minste variabelen ($n + n(n-1)/2 - |E|$, waarbij n het aantal punten in de graaf is) ook de minste rekentijd nodig had. K_5 geeft immers maar 5 variabelen. Wat vervolgens nog opvallend is is dat $\overline{K_5}$ een ruim twee keer zo lange rekentijd gaf als C_5, terwijl het verschil in variabelen maar 5 is (C_5 heeft immers 5 lijnen, $\overline{K_5}$ heeft er 0, beide hebben $n = 5$). 15 tegenover 10 variabelen. Dit 'grote' verschil in rekentijd is natuurlijk voor een groot deel te verklaren door het aantal variabelen, maar wat we niet moeten vergeten is dat in beide gevallen het algoritme voortijdig gestopt was. De fractie van het aantal uitgevoerde stappen en het aantal benodigde stappen speelt natuurlijk ook een rol. Tot slot lijkt het alsof de laatste graaf voor een enorme explosie in rekentijd heeft gezorgd. Dit is echter niet helemaal het geval. Waar de eerste drie grafen respectievelijk $5, 15$ en 10 variabelen gaven geeft de laatste graaf 46 variabelen. De benodigde rekentijd is overigens nog steeds redelijk klein, minder dan 3 minuten.
Al met al kunnen we dus concluderen dat we met deze implementatie over het algemeen een erg goede benadering van het optimum kunnen vinden binnen niet al te veel tijd. In sommige gevallen is het vervolgens niet lastig om met de output handmatig nog tot een betere oplossing te komen. Een goed voorbeeld hiervan is de eerste graaf K_5. We vonden hierbij de optimale vector

$$x = \begin{pmatrix} 0.199999978683650 \\ 0.199999978683650 \\ 0.200000030236868 \\ 0.200000012199195 \\ 0.199999984791989 \end{pmatrix}$$

De bijbehorende matrix lijkt heel sterk op de eenheidsmatrix maal $1/5$. We kunnen dus eens kijken of geldt dat $\frac{1}{5}I \in K$. Het is duidelijk dat $\mathrm{Tr}(\frac{1}{5}I) = 1$, verder geldt dat de eenheidsmatrix 0 heeft staan buiten de diagonaal en het is eenvoudig te zien dat $\frac{1}{5}I$ positief definiet is. Dus inderdaad $\frac{1}{5}I \in K$. Verder zien we $\langle J, \frac{1}{5}I \rangle = 1$. Dit geeft dus inderdaad een betere benadering van $\vartheta(K_5)$ (in feite geeft dit je zelfs $\vartheta(K_5)$, maar dat kan je in het algemeen natuurlijk niet weten).

Referenties

[1] Christos H. Papadimitiriou en Kenneth Steiglitz, 1998, *Combinatorial optimization*, Dover edition, Dover publications.

[2] Bryan P. Rynne en Martin A. Youngson, 2008, *Linear Functional Analysis*, Second edition, Springer.

[3] A.C. Zaanen, 1989, *Continuity, Integration and Fourier Theory*, Universitext, Springer-Verlag.

[4] M. Grötschel, L. Lovász en A. Schrijver, 1981, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica.

[5] Martin Henk, 2012, *Löwner-John Ellipsoids*, Documenta Mathematica Extra Volume: Optimization stories (2012).

[6] Reinhard Diestel, 2010, *Graph Theory*, Fourth edition, Springer.

[7] C. Vuik, P. van Beek, F. Vermolen en J. van Kan, 2006, *Numerieke methoden voor differentiaalvergelijkingen*, VSSD.

7. Appendix A

In deze appendix zullen we een schets geven van de methode waarmee het ellipsoïdealgoritme een inwendig punt zou vinden van de polytoop $Ax \leq b$. We nemen wel aan dat er een R bekend is zo dat de polytoop in $S(0, R)$ ligt, daarnaast moet de polytoop ook volledigdimensionaal zijn. We zullen hier niet alle bewijzen geven, deze zijn te vinden in [1] voor zover ze niet hier staan.
Het achterliggende idee bij het ellipsoïdealgoritme is dat de ellips in elke stap een kleiner volume krijgt, uiteindelijk heeft de ellips een zodanig klein volume dat men kan concluderen dat het gevonden optimum binnen epsilon van het echte optimum ligt. De reden dat het ellipsoïdealgoritme in dit geval ook zou werken is dat we wel een minimum volume kunnen geven dat omvat wordt door de polytoop. Aangezien de polytoop volledig-dimensionaal is bestaan er immers $n+1$ lineair onafhankelijke punten in $\mathbb{R}^n$ die allen binnen de polytoop liggen. Een verzameling punten heet lineair onafhankelijk als er geen drie punten op 1 lijn liggen.
Het volume van het convexe opspansel van $n+1$ punten wordt gegeven door

$$\frac{1}{n!}\left|\det\begin{pmatrix} v_1 - v_0 & v_2 - v_0 & \cdots & v_n - v_0 \end{pmatrix}\right|.$$

Dit kan als volgt ingezien worden: het volume van het convexe opspansel van $n+1$ punten is gelijk aan het volume van dezelfde $n+1$ punten getransleerd over de vector v_0. Op deze manier kunnen we dit convexe opspansel zien als het opspansel van de nulvector en een lineaire transformatie van de eenheidsvectoren in $\mathbb{R}^n$. Dat wil zeggen dat het volume van het gewenste convexe opspansel gelijk is aan het volume van het opspansel van de nulvector en de eenheidsvectoren maal de determinant van deze lineaire transformatie. De determinant van deze transformatie is gelijk aan $\det\begin{pmatrix} v_1 - v_0 & v_2 - v_0 & \cdots & v_n - v_0 \end{pmatrix}$. Het volume van het opspansel van de nulvector en de eenheidsvectoren in $\mathbb{R}^n$ is gelijk aan $\frac{1}{n!}$, zoals te zien is door te integreren:

$$\begin{aligned}
&\int_0^1 \int_0^{1-x_1} \int_0^{1-x_1-x_2} \cdots \int_0^{1-x_1-x_2-\ldots-x_{n-1}} 1 dx_n d_{n-1} \cdots dx_1 \\
&= \int_0^1 \int_0^{1-x_1} \cdots \int_0^{1-x_1-x_2-\ldots-x_{n-2}} (1 - x_1 - x_2 - \ldots - x_{n-2} - x_{n-1}) dx_{n-1} dx_{n-2} \cdots dx_1 \\
&= \int_0^1 \int_0^{1-x_1} \cdots \int_0^{1-x_1-x_2-\ldots-x_{n-3}} -\frac{1}{2}\left(1 - x_1 - \ldots - x_{n-2} - x_{n-1}\right)^2 \Big|_0^{1-x_1-\ldots-x_{n-2}} dx_{n-2} \cdots dx_1 \\
&= \int_0^1 \int_0^{1-x_1} \cdots \int_0^{1-x_1-x_2-\ldots-x_{n-3}} \frac{1}{2}(1 - x_1 - x_2 - \ldots - x_{n-2})^2 dx_{n-2} \cdots dx_1 \\
&\vdots \\
&= \int_0^1 \frac{1}{(n-1)!}(1 - x_1)^{n-1} dx_1 \\
&= -\frac{1}{n!}(1 - x_1)^n |_0^1 = \frac{1}{n!}
\end{aligned}$$

Gebruiken we nu Lemma 8.5 uit [1] dan weten we dat elk van deze punten v_i te schrijven is als een rationale vector $\frac{u_i}{D_i}$ waarbij zowel de absolute waarde als de noemer begrensd zijn door 2^L. Hierbij is L de grootte van de instantie $L = mn + \lceil \log|P| \rceil$ waarbij m het aantal vergelijkingen (rijen van A) is en P het product van alle niet-nul coëfficienten in A en b. u_i is hierbij een geheeltallige vector. Maar

dat wil zeggen dat

$$\begin{aligned}\left|\det\begin{pmatrix} v_1 - v_0 & v_2 - v_0 & \cdots & v_n - v_0 \end{pmatrix}\right| &= \frac{1}{\prod D_i}\left|\det\begin{pmatrix} u_1 - u_1 & u_2 - u_0 & \cdots & u_n - u_0 \end{pmatrix}\right| \\ &\geq \frac{1}{\prod D_i} \geq \frac{1}{2^{nL}}\end{aligned}$$

Waarbij we hebben gebruikt dat de punten lineair onafhankelijk zijn en dat dus $\left|\det\begin{pmatrix} u_1 - u_1 & u_2 - u_0 & \cdots & u_n - u_0 \end{pmatrix}\right| \geq 1$. Maar dat wil dus zeggen dat we het volume van het opspansel van de $n+1$ punten minstens $\frac{1}{n!}2^{-nL}$ is. We kunnen met het algoritme dus zo veel stappen uitvoeren dat het volume van de ellips in de laatste stap kleiner is dan $\frac{1}{n!}2^{-nL}$. Dit kan aangezien we Lemma 6 hebben. Die geeft ons dat het volume van de ellips in de laatste stap (zeg k) gelijk is aan $e^{-k/(4n)}$ maal het volume van de eerste ellips. Het volume van de eerste ellips is natuurlijk bekend, we kunnen k dus groot genoeg kiezen. Maar als het volume van de laatste ellips kleiner is dan het minimale volume in het inwendige van de polytoop dan moeten we dus een keer een inwendig punt van de polytoop hebben gevonden!

8. Appendix B: de Matlab code

In deze appendix is de code van de gebruikte Matlab programma's te vinden. In een aantal gevallen heb ik vanwege de breedte van a4-papier een extra regel moeten toevoegen, de onderstaande code kan dan ook niet letterlijk worden gekopieerd naar Matlab. We zullen beginnen met het programma waarin de invoer wordt geplaatst, vandaaruit zullen we 'de diepte in gaan'. Oftewel elke nieuwe code wordt gebruikt door de vorige.

De uitvoer van het ellipsoidealgoritme.

```
clear all;
clc;

tic; %start measuring the calculation time

%Input
n = 5;
G = [zeros(1,5) ones(1,4) zeros(1,3) zeros(1,2) ones(1,1)];

m = n + n*(n-1)/2 - sum(G); %The dimension of the problemspace, R^m
eps = 1/100; %epsilon

a1 = [1/(2*n)*ones(1,n) zeros(1,m-n)];
%The vector in the interior of the convex set
r = 1/100;
R = m;
c = [ones(1,n) 2*ones(1,m-n)];
%Cost vector, the main diagonal is counted once,
%the others because of symmetry twice.

N = ceil(abs(5*m^2*log(2*R^2*norm(c)/(r*eps))))

%Initialization
x = a1;
A = R*eye(m);
xval= 0 ;
xopt = a1;

%The ellipsoid algorithm
for k = 1:N
    tempa = SEP(x,n,G,m); %call to the separation algorithm
    if(tempa == zeros(m,1)) %The case x is in K
        a = c';
        if(c*x' > xval)
            xval = c*x';
            xopt = x;
        end
    else
```

```
        a = -tempa;
    end
    b = A*a/sqrt(a'*A*a);
    tf = isnan(b); %Checking whether or not we still have an ellips.
    %(if an element of b equals nan than A*a = 0, so A has an ev = 0)
    if(tf(1)==1)
        break;
    end
    x = x + 1/(m+1)*b';
    A = (2*m^2+3)/(2*m^2)*(A-2/(m+1)*(b*b'));
end

%printing the results
optimum = xval
xoptimum = xopt

toc %end measuring the calculation time
```

In het ellipsoidealgoritme wordt een functie SEP gebruikt, deze functie is het separatiealgoritme welke hieronder is weergegeven.

```
function d = SEP(x,n,G,m) %length(x) = m, n nodes, graph G)

y = transformdim(x,G);
%Adding the 'missing' coordinates (the ones in G)

B = diag(x(1:n));
%initializing construction of the symmetric matrix
%corresponding to x

teller = n+1;

if(sum(x(1:n)) > 1) %if trace > 1, separating hyperplane is z
    z = [ones(n,1); zeros(m-n,1)];
    d = z;
else
%Completing B by filling in the off-diagonal entries
    for l = 1:n-1
        B = B + diag(y(teller:teller+n-l-1),l)
        + diag(y(teller:teller+n-l-1),-l);
        teller = teller + n-l;
    end

    [P,D] = symvegen(B);
%Call to symvegen, the output: D diagonal,
%P is a transformation such that D = P*B*P'

    diagonaal = diag(D);

```

```
29      i = find(diagonaal < 0,1);
30  %Negative entry would mean B is not psd
31      if(isempty(i))      %The return indicating B is psd
32          d=zeros(m,1);%(note 0 can never be a separating hyperplane)
33
34      else
35          ev = P(i,:)'; %<B,ev*ev'> ≤ 0
36          zmat = ev*ev';
37          z = transformmatvec(zmat,G,m,n);
38  %transforming the separating matrix
39  %back to an m-dimensional vector
40
41          d = -z/norm(z);
42      end
43  end
44  end
```

Dit algoritme gebruikt de functies transformdim, transformmatvec en symvegen. De eerste twee zijn, zoals de naam al doet vermoeden, functies die van het ene object een equivalent object van een andere vorm maken. Symvegen is een functie die een matrix symmetrisch veegt.

```
1  function y = transformdim(x,G)
2  %Input: m-dimensional vector x, the graph G.
3  %Output: the (n+n(n-1)/2)-dimensional vector y with
4  %added zeros on the coordinates on which G=1
5
6  z = zeros(length(G),1);
7  teller = 1;
8
9  for h = 1:length(G)
10     if (G(h) == 0)
11         z(h) = x(teller);
12         teller = teller +1;
13     end
14 end
15 y = z;
16 end
```

```
1  function y = transformmatvec(A,G,m,n)
2  %transforms an n-by-n matrix A into the corresponding
3  %m-dimensional separating vector y which has the main
4  %diagonal on its first n entries and each kth diagonal
5  %(k>0) multiplied by 2 on the other coordinates
6
7  z = zeros(length(G),1);
8  z(1:n) = diag(A);
9  %The main diagonal is only counted once in <A,xx^T>
```

```
teller = n+1;

for i = 1:n-1 %The i-th diagonal is counted twice in <A,xx^T>
        z(teller:teller+n-i-1) = 2*diag(A,i);
        teller = teller + n-i;
    end

%removing the coordinates for which G=1
w = zeros(m,1);
teller = 1;

for h = 1:length(G)
    if (G(h) == 0)
        w(teller) = z(h);
        teller = teller +1;
    end
end
y = w;
```

```
function [P,D] = symvegen(A)
%Input: symmetric matrix A, output: D diagonal,
%P transformation such that D = P*B*P'

n = length(A(1,:));
B = eye(n);

for i = 1:n-1
    for k = i+1:n
        if(A(i,i) == 0) %diagonal entry zero
            if(¬isempty(find(A(i,:) ≠ 0,1)))
        %if there is a non-zero entry in this row/col,
        %add it to row/col i to make the diagonal entry non-zero
            j = find(A(i,:) ≠ 0,1);
%The overall change in coordinates
            B(i,:) = B(i,:) + B(j,:);
            T = eye(n);
%Coordinate transformation in this step
            T(i,:) = T(i,:) + A(j,i)*T(j,:);
            A = T*A*T';

            frac = (A(k,i)/A(i,i));
            B(k,:) = B(k,:) - B(i,:)*frac;
            T = eye(n);
            T(k,:) = T(k,:) - frac*T(i,:);
            A = T*A*T';
            end
        %Note that there is no else to this if,
        %if there is no non-zero entry to row
```

```
        %i we can proceed with i = i+1
        else
        frac = (A(k,i)/A(i,i));
        B(k,:) = B(k,:) - B(i,:)*frac;
        T = eye(n);
        T(k,:) = T(k,:) - frac*T(i,:);
        A = T*A*T';
        end

    end
end
P = B;
D = A;
end
```

Tot slot volgt nog een handige file waarin een aantal mogelijke inputs gegeven zijn.

```
%Some interesting graphs

%K_n, optimum = 1
n = 5;
G = [zeros(1,n) ones(1,n*(n-1)/2)];

%Empty graph on n points, optimum = n
n = 5;
G = zeros(1, n + n*(n-1)/2);

%pentagon graaf, C5, optimum = sqrt(5)
n = 5;
G = [zeros(1,5) ones(1,4) zeros(1,3) zeros(1,2) ones(1,1)];

%The graphs below were given to me by Eline Rietberg,
%they are interesting for her project.

%3-regular stargraph, optimum = 8.541019662384862
n=20;
G= [zeros(1,20) ones(1,4) zeros(1,11) ones(1,4) zeros(1,18)
    zeros(1,17) 1  zeros(1,5) ones(1,4) zeros(1,5) 1
    ones(1,15) zeros(1,39) zeros(1,5) 1 zeros(1,5) zeros(1,55)];

%Grotzsch graph, optimum = 5
n=11;
G= [zeros(1,11) 1 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1
    0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 1
    1 1 0 0 0 0 0 0 0 0 1 0 0];
```