ALGORITHMEN ZUR OPTIMALEN LÖSUNG VON LP-s

(aus The Nature of Computation)

(oBdA. A und b haben Integer Koordinaten)

Wichtige Beobachtung: Wenn eine LP-Instanz mit n Bits repräsentiert wird, dann sind alle Koordinaten von Ecken, und alle Zielwerte von ~~Lösungen~~ Ecken in Absolutwert höchstens $2^{O(n)}$. Weiterhin, $2^{-O(n)}$ Exaktheit in den Berechnungen reicht für die Eindeutigkeit einer Lösung (Ecke). (minimale Distanz zwischen Ecken ist größer)

Sei ein LP in kanonischer Form gegeben: minimiere $c^T \cdot x$ s.d. $A \cdot x \geq b$

Die Ellipsoid-Methode

(war der erste Polynomialzeit-Algorithmus für LP)
1979

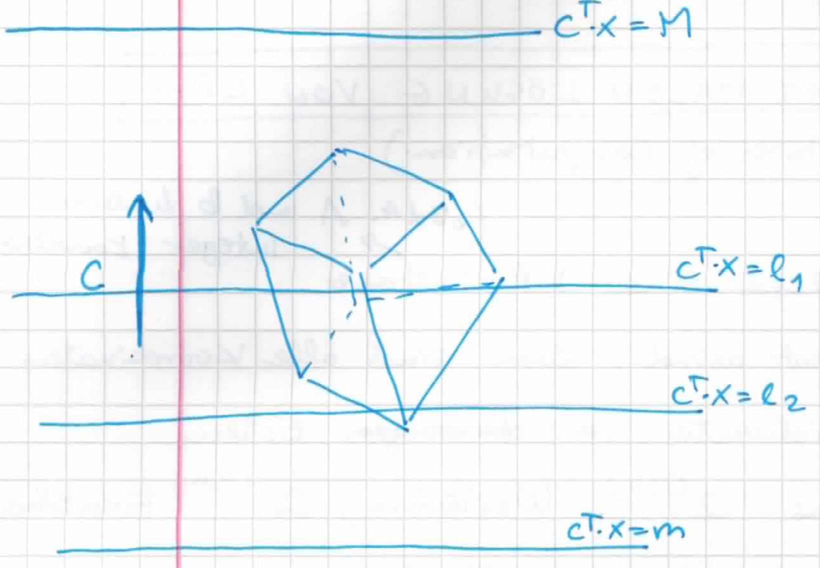
Betrachte die Entscheidungsversion von LP:

LP-FEASIBILITY (Version mit Schwelle ℓ)

Eingabe: A, b, c, ℓ

Aufgabe: Entscheide ob eine Lösung x mit $Ax \geq b$ und Zielwert $c^T \cdot x \leq \ell$ existiert
(auf Anfrage, falls JA, dann gib eine solche Lösung aus)

→ Wenn LP-FEASIBILITY polynomiell lösbar, dann können wir LP optimieren — mit binärer Suche nach dem kleinsten ℓ mit Antwort JA
(mit der nötigen Exaktheit $\frac{1}{2^{O(n)}}$ in Poly(n) Laufzeit)



Wie?
Seien M und m triviale obere und untere Schranken für $c^T \cdot x$ über dem Polyeder $Ax \geq b$ (genauer: über seinen Ecken)

Aber: Wie löst man LP-FEASIBILITY effizient?

Beachte: es wird nur irgendeine Lösung x gesucht für ein (neues) LP-Problem mit Bedingungen

$$Ax \geq b$$
$$c^T \cdot x \leq \ell$$

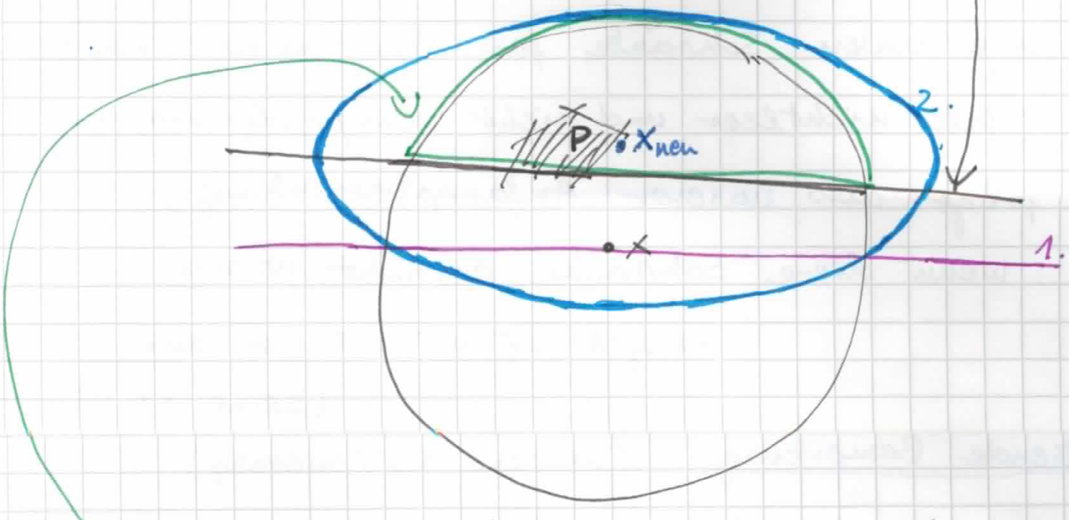
d.h. ein Punkt im Lösungspolyeder P von diesem Problem (da $c^T \cdot x \leq \ell$ nur eine weitere lineare Bedingung ist)

Um eine Lösung x in P zu finden, werden wir sowas wie eine n-dimensionale binäre Suche in $\mathbb{R}^n$ durchführen!
("fange einen Löwen in der Sahara in dem immer kleinere Bereiche umzäunt werden")

- Starte mit einer hinreichend großen Kugel, die das Lösungspolyeder P (falls nichtleer) enthalten soll (oder zumindest alle Ecken von P)
- sei x das Zentrum der Kugel
  - falls $x \in P$ → fertig
  - sonst gibt es eine lineare Bedingung, die x nicht erfüllt

(die nicht-erfüllte Bedingung findet der Alg. ~~[illegible]~~ durch Prüfen der m Bedingungen)

- diese entspricht einer Hyperebene, die x von P trennt



- wir sollten die Suche nach einem $x \in P$ im abgetrennten Bereich der Kugel weitermachen....

dies würde aber zu immer komplizierter Bereichen führen, und letztendlich (evtl.) zum Ursprungsproblem (Lösungspolyeder) P und wir hätten nichts gewonnen

in Komplexität. Wie kann man den aktuell betrachteten Bereich einfach halten, - wo man leicht ein x findet (im Bereich)
- ~~der~~ mit wenigen Parametern definierbar ist
- flexibel ist um in jeder Runde um einen Faktor kleiner zu werden?

- um weiterhin mit einfachen Objekten → effizient zu rechnen,

REPEAT
1. man nimmt die <u>parallele Hyperebene</u> zur trennenden Hyperebene, <u>durch x</u> → ein Halbkreis entsteht. (Halb-Ellipsoid)
2. dann nimmt man das Ellipsoid, das diesen Halbkreis (Halb Ellipsoid) enthält, ~~mit~~ dem kleinsten Volumen
3. sei das neue x das Zentrum des neuen Ellipsoids

– das Volumen des Ellipsoids wird in jeder Runde um einen gewissen konstanten Faktor kleiner

– eine untere Schranke für das Volumen von P (falls nichtleer und nicht weniger-dimensional) folgt aus unserer Anfangsbemerkung; wenn diese Schranke erreicht wurde

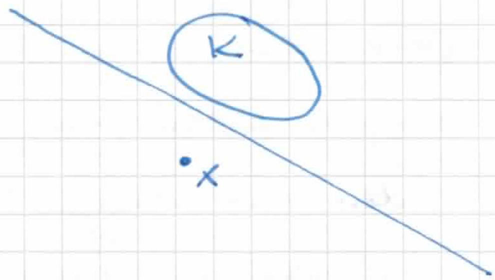
→ gib KEINE LÖSUNG aus (keine $x \in P$)

Abschließende Bemerkung: (konvexe Optimierung)

Diese Methode kann man für beliebige konvexe Lösungsmenge K verwenden

FALLS für jede K und $x \notin K$ eine trennende (separierende) Hyperebene effizient berechnet werden kann

d.h. wenn ein effizientes sog. Trennungs-Orakel (Algorithmus) bekannt ist.



Ein Trennungs-Orakel (separation oracle) für die konvexe Menge K nimmt ein Punkt x als Input.

Falls $x \in K$, sagt es JA

falls $x \notin K$ gibt es eine Hyperebene $a^T \cdot y = b$ aus, so dass $a^T \cdot x < b$, aber $a^T \cdot y' \geq b$ für jede $y' \in K$.

(trennende Hyperebene – separating hyperplane)

(Beachte: Für nicht-konvexe Lösungsmengen gibt es eine trennende Hyperebene im Allgemeinen nicht

)

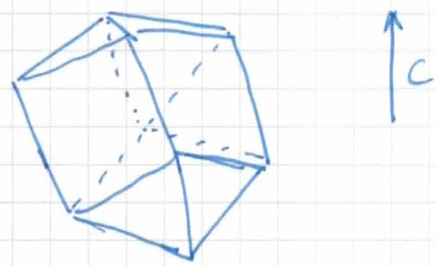
Interior - Point Verfahren

1984 effizienter (schneller) als die Ellipsoid-Methode

$$\text{minimiere } c^T \cdot x$$
$$\text{s.d.} \quad Ax \geq b$$

Sei c vertikal, und das Lösungspolyeder P beschränkt (Polytop)

denke an eine (weiche) Kugel, die man (vertikal) in P fallen lässt, und an ihren Weg in die tiefste Ecke x^* (in slow-motion) am Rand von P



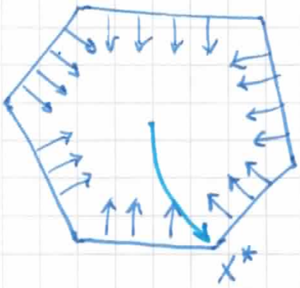
((alternativ: Luftballon mit Helium wird in einer Cathedrale losgelassen))

Ein Algorithmus, der den Weg dieser Kugel bis zur tiefsten Ecke (optimalen Lösung x^*) berechnet, wäre nach den ersten paar Schritten äquivalent mit Simplex Alg.:

- die Kugel fällt auf eine $n-1$ dim. Seitenfläche, rollt auf eine $n-2$ dim. Schnitt von zwei Seitenflächen, und so weiter, bis in eine 1 dimensionale Kante, und dann rollt von Kante zur Kante weiter bis x^*

→ wir wissen vom Simplex-Algorithmus, dass sein Weg über exponentiell viele Kanten nach unten führen kann, also hätte ein solcher Algorithmus exponentiell viele Schritte im worst-case.

Idee: → halten wir die Kugel fern von den Seitenwänden, um sie nicht in das Kanten-Labyrinth zu führen.



führen wir Abstoßkräfte von jeder Wand ein, die unendlich stark werden, als die Kugel die Wand annähert: im Punkt x, die Kraft von Wand i ist proportional zur $\frac{1}{r_i}$ Inverse der Distanz r_i von x zu dieser Wand.

(r_i ist proportional zur $b_i - a_i^T x$)

Diese halten die Kugel fern vom Rand des Lösungspolyeders.

→ Um die Kugel doch in die tiefste Ecke x^* zu schieben, muss sie in jeder Runde schwerer werden. Deshalb wird man die Gravitationskraft $\lambda \cdot (-c)$ Runde für Runde von $\lambda = 0$ auf immer verstärken, und in jeder Runde die Gleichgewichtsposition der Kugel berechnen, (d.h. wo sich alle Kräfte ausgleichen).

Im Karmarkar-Algorithmus, startet die Kugel irgendwo in der Mitte mit $\lambda = 0$ (0 Gravitation), dann wird λ Schritt für Schritt erhöht, so dass die Position der Kugel die Distanz zur x^* jeweils etwa halbiert.

⇒ Wir finden x^* mit $2^{-O(n)}$ Exaktheit in $O(n)$ Runden!

Bemerkung: Wenn λ kontinuierlich erhöht wird, wird der Weg der Kugel eine glatte Kurve zur x^*. Die verschiedenen Interior-Point Verfahren (wie auch der Karmarkar-Alg.) sind numerische Approximationen dieser Kurve

— Polygonzüge mit $O(n)$ geraden Strecken.