ETE351 Grundläggande optimeringslära

- Examinator:
  - Kaj Holmberg
  - kaj.holmberg@liu.se
- Kurshemsida: http://courses.mai.liu.se/GU/ETE351
- Lärare:
  - Föreläsningar: Kaj Holmberg
- Litteratur:
  - Kaj Holmberg: Optimering (Liber, 2018) upplaga 2
  - Kaj Holmberg: Grön optimering
- Undervisning:
  - Föreläsningar
  - Lektioner
  - Laborationer
- Examination:
  - Skriftlig tenta på distans
  - Laborationer, redovisas skriftligt

Optimering

När inte intuitionen räcker till...

- Matematisk bas.
- Löser verkliga problem.
- Använder speciellt utvalda algoritmer.
- Fanns ej utan datorer.
- Kräver kompetens/kunskaper från flera olika områden.
- Två specifika svårigheter:
  - Konstruera relevant modell.
    - ⋆ Ta med relevanta saker (t.ex. kostnader, fysik, miljö).
  - Lösa modellen.
    - ⋆ Välja/använda lämplig metod.
- Intressant blandning av teori (lätt) och verklighet (svårt).

Långsiktiga mål med kursen

- Känna igen optimeringsproblem.
- Kunna formulera problem matematiskt.
- Förstå principerna bakom olika metoder.
- Kunna välja lämplig lösningsmetod.
- Kunna använda tillgänglig programvara.
- Medverka vid utveckling av ny programvara.
- (Räkna för hand.)

När (inte om) ni stöter på ett optimeringsproblem, ska ni kunna angripa det med optimeringsteknik.

Vad är optimering?

Konsten att göra på bästa möjliga sätt.

(Men inte “konst”, utan “vetenskap”.)

“Bästa” definieras av vad jag vill (minimera eller maximera).
Kallas “målfunktion”. Exempel: Minimera kostnaden.

“Möjliga” definieras genom att förbjuda det som är omöjligt/otillåtet.
Kallas “bivillkor”. Exempel: Negativa värden ej tillåtna.

Målfunktion och bivillkor måste definieras exakt/matematiskt.

Obs: benämningen “optimalt” saknar betydelse om man inte har/känner till modellen, dvs. målfunktion och bivillkor!
Detta gäller även media, politiker m.fl.

Hur kan man misslyckas?

1. Lös problemet felaktigt, dvs. finn en lösning som inte är optimal, utan sämre och/eller otillåten.
   Dålig metod.
2. Lös fel problem, dvs. finn korrekt optimallösning till fel modell.
   Dålig modell.
3. Både 1 och 2, dvs. finn fel lösning till fel modell.
   Dålig modell och metod.

2 är vanligast.

3 kan vara bättre än 2.

Optimering är ett bra sätt att hitta fel i en modell.
Lösningen blir knäpp om man har glömt något viktigt bivillkor.

Optimering

- Optimus: "Bäst" (på latin).
- Optimeringsproblem: $\min f(x)$ då $g_i(x) \leq b_i$ för $i = 1, \dots m$
  - x kallas variabler.
  - $f(x)$ kallas målfunktion.
  - $g_i(x) \leq b_i$ kallas bivillkor.
- Delområden (beroende på strukturen hos problemet):
  - Linjärprogrammering, LP (Dantzig 1949)
  - Ickelinjärprogrammering, ILP (Kuhn & Tucker 1951)
  - Heltalsprogrammering, HP (Gomory 1958)
  - Dynamisk programmering, DynP (Bellman 1957)
  - Kombinatorisk optimering: Räkna upp kombinationer
- Programmering: (grekiska: pro + gramma = föreskrift, planering)

Hur man använder optimering: Operationanalys

1. **Formulering av problemet.** Finns det ett problem? Vad vill man optimera? Vilka begränsningar finns?
2. **Konstruktion av en matematisk modell.** Definiera variabler, målfunktion samt bivillkor. Är resultatet en LP-, ILP eller HP-modell?
3. **Insamling av data.**
4. **Lösning av det matematiska problemet.** Välj lämplig optimeringsmetod.
5. **Utvärdering av resultat (och modell).** Är resultatet realistiskt, lämpligt, vettigt, "bra"? Om inte, gå till 2.
6. **Använd resultatet.**

Mål när man gör modellen

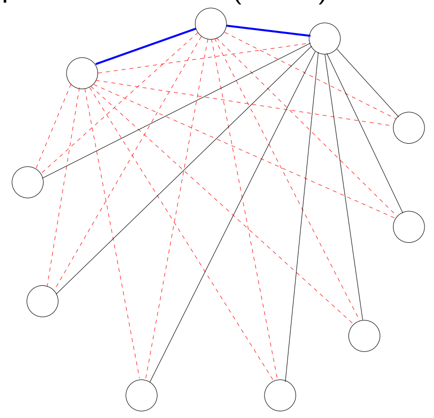
- Få med allt relevant, dvs. som påverkar vilken lösning som är optimal.
- Undvik det som är irrelevant, dvs. som inte påverkar vilken lösning som är optimal.
- Modellen ska vara korrekt, dvs. göra det man vill att den ska göra.
- Modellen ska vara lösbar, dvs. gå att lösa på rimlig/tillgänglig tid.
- Data (koefficienter) ska kunna tas fram.
- De förenklingar man kan tvingas göra ska vara medvetna och genomtänkta.
- Undvik onödiga komplikationer, såsom olinjäriteter.
- Välj målfunktion.

Mål när man väljer optimeringsmetod

- Lös problemet så effektivt som möjligt.
- Viktigt, ty verkliga problem är stora.
- En dålig metod kan ta lång tid.
- En smart implementering är ej tillräckligt.

Är handelsresandeproblemet svårt?

Möjlig metod (?): Jämför alla rundturer.
Exempel med 10 orter (noder):



Totalt 362 880 möjligheter.
Matematiskt: Det finns $(n-1)!$ olika sätt att besöka n platser.
För hand: 1 sekund per tur: 362 880 sekunder, dvs. ca 4 dagar.
Dator: 1 ms per tur: 362 sekunder, dvs. ca 6 minuter.

Men vi kanske vill lösa större problem

n	Antal turer	Tid för hand	Tid för dator	Snabb dator
6	120	2 min	120 ms	120 μs
7	720	12 min	0.7 s	0.7 ms
8	5040	1 timme	5 s	5 ms
9	40320	11 timmar	40 s	40 ms
10	362880	4 dagar	6 min	0.4 s
11	3628800	42 dagar	1 timme	4 s
12	39916800	1 år	11 timmar	40 s
13	479001600	15 år	5.5 dagar	8 min
14	6227020800	197 år	72 dagar	1.7 timmar
15	87178291200	2764 år	2.7 år	24 timmar
16	1307674368000	$4.15 * 10^4$ år	41.5 år	15 dagar
18	355687428096000	$1.13 * 10^7$ år	$1.13 * 10^4$ år	11 år
20	121645100408832000	$3.86 * 10^9$ år	$3.86 * 10^6$ år	3860 år
21	2432902008176640000	$7.71 * 10^{10}$ år	$7.71 * 10^7$ år	$7.71 * 10^4$ år
22	51090942171709440000	$1.62 * 10^{12}$ år	$1.62 * 10^9$ år	$1.62 * 10^6$ år
23	1124000727777607680000	$3.56 * 10^{13}$ år	$3.56 * 10^{10}$ år	$3.56 * 10^7$ år

Men vi kanske vill lösa större problem

Det sägs att det har gått 13.7 miljarder år sedan “the Big Bang”.

Om vi började räkna för hand då, hade vi hunnit med ett problem med 20 noder, men inte ett med 21 noder.

Med en dator hade vi hunnit med ett problem med 22 noder.

Med en snabbare dator hade vi hunnit med ett problem med 24 noder.

Så problemen är svåra om man använder en osmart metod.

Hur bra kan man göra med en smart metod?

De största som lösts till optimalitet

År	n	Forskare	Turer
-	10	-	$3.6 * 10^{6}$
-	20	-	$1.2 * 10^{17}$
1954	49	Dantzig, Fulkerson och Johnson	$1.2 * 10^{61}$
1971	64	Held och Karp	$2.0 * 10^{87}$
1975	100	Camerini, Fratta och Maffioli	$9.3 * 10^{155}$
1977	120	Grötschel	$5.6 * 10^{196}$
1980	318	Crowder och Padberg	$2.1 * 10^{659}$
1987	532	Padberg och Rinaldi	$1.5 * 10^{1218}$
1987	666	Grötschel och Holland	$1.5 * 10^{1590}$
1987	2392	Padberg och Rinaldi	$7.5 * 10^{7041}$
1994	7397	Applegate, Bixby, Chvatal och Cook	$2.5 * 10^{25405}$
1998	13509	Applegate, Bixby, Chvatal och Cook	$1.1 * 10^{49932}$
2001	15112	Applegate, Bixby, Chvatal och Cook	$1.4 * 10^{56593}$
2004	24978	Applegate, Bixby, Chvatal och Cook	$3.9 * 10^{98992}$
2006	85900	Applegate, Bixby, Chvatal, Cook, Espinoza, Goycoolea och Helsgaun	$1.1 * 10^{386522}$

Ett första exempel: Text

Företaget *Mickey AB* producerar olika sorters datormöss. Man ska nu under en period fokusera på två sorter, Optimus och Rullmus. Man har begränsat tillgång av vissa delar. Optimusen har två knappar medan Rullmusen har tre. Man kan få fram material till maximalt 30 knappar per timme. Varje Optimus har en optisk enhet, och man kan använda högst 6 optiska enheter per timme. Att montera en Optimus kräver 6 minuter i musmaskinen, medan en Rullmus bara kräver 4 minuter. Under en timme kan maskinen användas i medel 50 minuter. Resterande tid åtgår till rengöring och service. En Optimus ger vinsten 4 kr och en Rullmus ger 3 kr. Mickey vill inte ändra produktionen alltför ofta, utan föredrar att använda samma timplanering tills yttre förutsättningar ändras. Hur många möss av varje sort skall man göra varje timme för att maximera intäkterna?

Ett första exempel: Matematisk modell

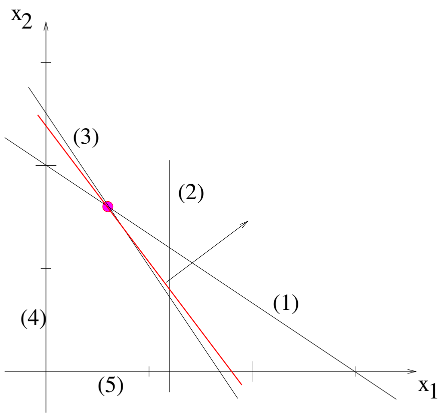
Variabeldefinition:

x_1 = antal enheter Optimus som görs varje timme.
x_2 = antal enheter Rullmus som görs varje timme.

Matematisk modell:

$$
\begin{array}{llrcrcrll}
\max & z = & 4x_1 & + & 3x_2 & & & & \\
\text{då} & & 2x_1 & + & 3x_2 & \leq & 30 & (1) & \text{(knappar)} \\
 & & x_1 & & & \leq & 6 & (2) & \text{(optik)} \\
 & & 6x_1 & + & 4x_2 & \leq & 50 & (3) & \text{(monteringstid)} \\
 & & x_1 & & & \geq & 0 & (4) & \\
 & & & & x_2 & \geq & 0 & (5) &
\end{array}
$$

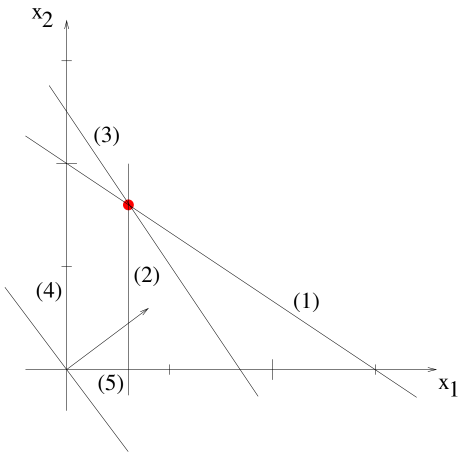
Ett första exempel: Grafisk lösning



Tillåtet område Målfunktion Optimallösning: $x_1 = 3, x_2 = 8, z = 36$.
Gör 3 st Optimus och 8 Rullmus varje timme vilket ger en vinst på 36 kr per timme. Alla knappar går åt och all monteringstid används, men det blir 3 optiska enheter över varje timme.

Ett första exempel: Variation

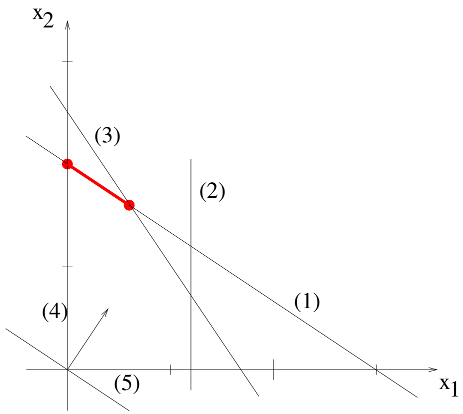
Man fattar ett principbeslut att halvera tillgången av optiska enheter. Bivillkor 2 blir då $x_1 \leq 3$.



Degenererad lösning. Tre bivillkor aktiva. Alla resurser tar slut.

Ett första exempel: Variation

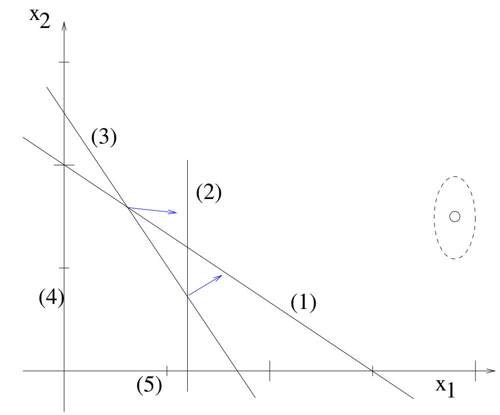
Istället för att ransonera tillgången på optiska enheter införs en straffavgift på 2 kr per enhet. Ny målfunktion: $\max z = 2x_1 + 3x_2$.



Icke-unik optimallösning: $x_1 = 3, x_2 = 8, z = 30$ och $x_1 = 0, x_2 = 10, z = 30$ samt alla som ligger mellan dem.

Ett första exempel: Olinjär variation

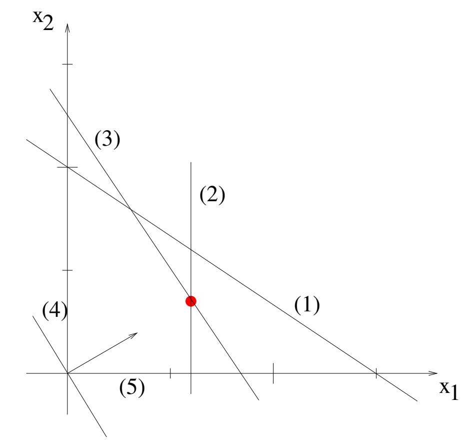
Vinsten avtar med mängden: $c_1(x_1) = 4 - 0.1x_1$ och $c_2(x_2) = 3 - 0.2x_2$, vilket ger målfunktionen $f(x) = c_1(x_1)x_1 + c_2(x_2)x_2 = (4 - 0.1x_1)x_1 + (3 - 0.2x_2)x_2 = 4x_1 - 0.1x_1^2 + 3x_2 - 0.2x_2^2$.



Kan ej lösas grafiskt.

Ett första exempel: Variation

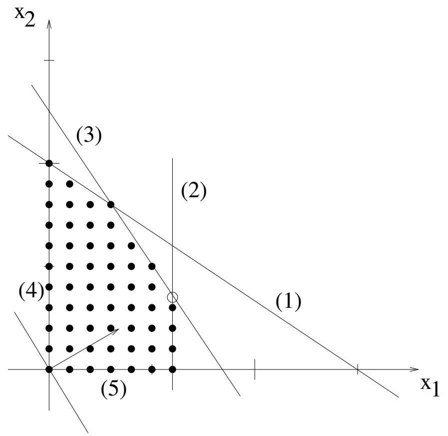
Optimus ger intäkt 5 kr per enhet. Målfunktion: $\max z = 5x_1 + 3x_2$



LP-lösning: $x_1 = 6, x_2 = 3.5, z_{LP} = 40.5$. Icke heltalig optimallösning.

Ett första exempel: Heltalsproblemet

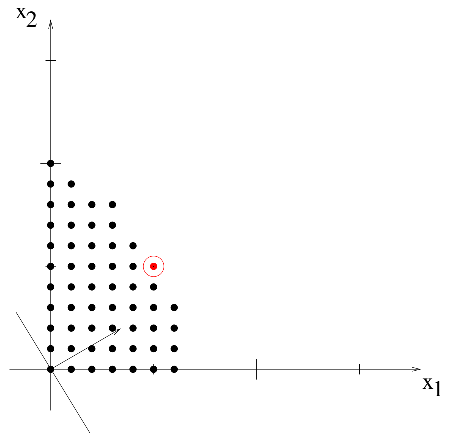
Antalet enheter måste vara heltal. Lägg till: x_1, x_2 heltal



Tillåtet område: Enbart de svarta prickarna.

Ett första exempel: Heltalsproblemet

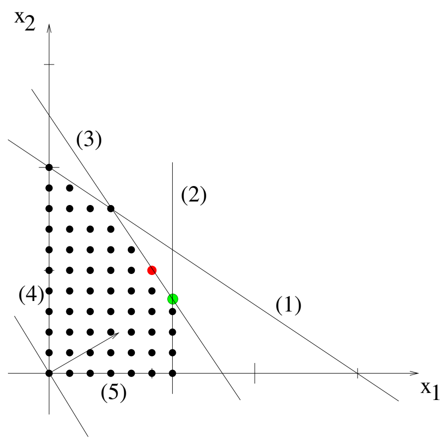
Antal enheter måste vara heltal. Lägg till: x_1, x_2 heltal



Tillåtet område: Enbart de svarta prickarna.
Heltalslösning: $x_1 = 5, x_2 = 5, z^* = 40$
Gör 5 enheter av båda sorterna.

Ett första exempel: Heltalsproblemet

Antalet enheter måste vara heltal. Lägg till: x_1, x_2 heltal



Skillnaden mellan LP-lösning och heltalslösning: $z^* - z_{LP} = 0.5$.
Heltalsoptimum kan inte fås genom avrundning av LP-optimum.

Introduktion till komplexitet

Teoretisk bas för frågorna:

- Är en viss metod bra eller dålig?
- Är ett visst problem lätt eller svårt?

Hur många operationer krävs, som funktion av indatas storlek, i värsta fall?

Vi skiljer på *polynomisk* komplexitet (lätt) och *exponentiell* (svår).

Ex: 2^n blir alltid större än n^4, om n blir stort nog.

Introduktion till komplexitet

Vilken tidskomplexitet har den bästa kända metoden för problemet?

P är den klass av problem som kan lösas av en polynomisk algoritm.

Svårare klasser: NP-fullständiga, NP-svåra

Tro: Det finns ingen polynomisk algoritm för något NP-fullständigt/-svårt problem.

Introduktion till komplexitet

Exempel 1:
Sortera n heltal i stigande ordning: $O(n\log(n))$. Polynomisk algoritm.

Exempel 2:
Genomlöpa alla hörn i en hyperkub i n dimensioner: $O(2^n)$. Exponentiell algoritm.

Heuristiker

"Smarta" metoder som ger "skapliga" lösningar.

Ger ej garanterat optimum.

Men kan göra det om man har tur.

Snabbare än optimerande metoder.

Enda möjligheten för riktigt stora svåra problem.

Ofta: NP-svårt problem, polynomisk heuristik.

Tillämpningsområden

- Optimal mandatfördelning.
- Optimal digital kartmatchning.
- Optimal formering av studentgrupper.
- Optimal snöröjning.
- Optimal design av kullager.
- Optimal placering av UAVer som kommunikationsreläer.
- Optimal planering av militära attackmönster.
- Optimal design, styrning och kontroll av IP-nät.
- Optimal omruttning för symmetriska trestegs-Closnätverk
- Optimal placering och förflyttning av tomvagnar på järnväg.
- Optimal packning av pappersrullar i järnvägsvagnar.

Tillämpningsområden

- Optimal stråldosering vid cancerbehandling.
- Optimal planering av sjuksköterskor.
- Optimal planering av skogsavverkning, transporter mm.
- Optimalt vägunderhåll.
- Trafikplanering (nya vägar, vägtullar).
- Optimerade dagbrott.
- Optimal design av filter.
- Frekvensplanering i GSM.
- Optimalt utnyttjande av kraftverk.
- Optimal ruttplanering för gas-/oljeleverantörer.
- Finansiell riskhantering.

Trend

Miljöaspekter blir allt viktigare i optimeringsmodellerna.

Lagkrav.

PR-värden (certifiering etc).

Samvete.