Aalto University
School of Business

Nonlinear Programming (NLP)

- *Geometric illustration and solution*
- *Computer Solution (gradient search)*
- *Lagrange multiplier*
- *Global and local optima*
- *Convex and concave NLPs (Convex sets, convex/concave functions)*
- *Computer solution (evolutionary algorithms)*

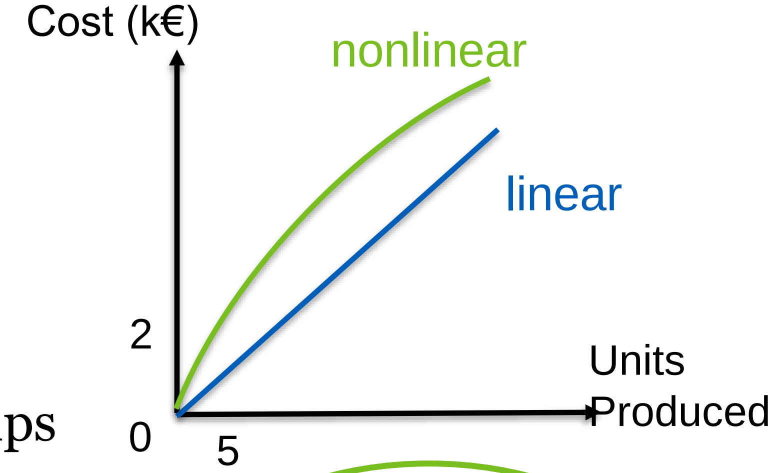
Next Monday’s guest lecture:

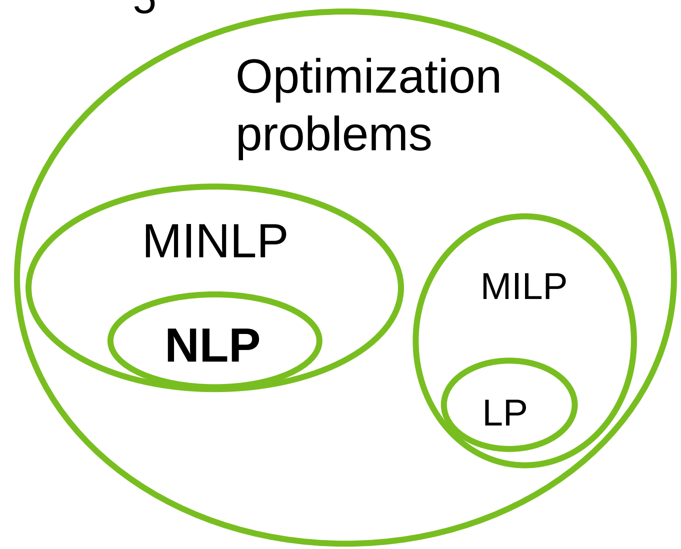
“Optimisation in Energy Transition”

Matti Vuorinen, Director, Digital Solutions in UPM Energy

Non-linear Programming (NLP)

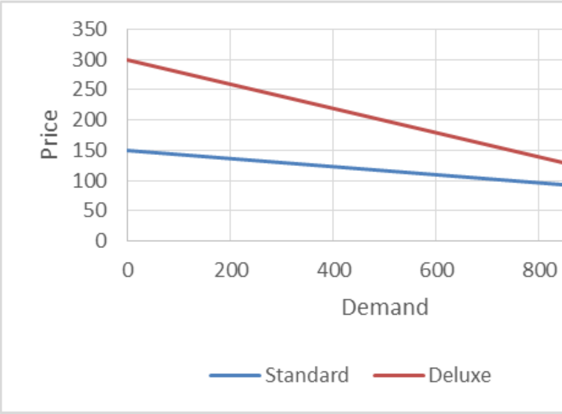
- LP models *proportional* relationships
  - Linear constraints and objective function
- NLP models *non-proportional* relationships
  - Nonlinear objective function and constraints
- Terminology
  - Even if only one of the constraints or the objective function is nonlinear $\rightarrow$ NLP model
  - NLPs can have integer variables $\rightarrow$ Mixed Integer Non-linear Programming (MINLP)





NLP Example: Production planning

- Par inc. manufactures standard and deluxe golf bags
  - Productions costs: \$70 and \$150
  - Production constraints:
    - Cutting & dying, sewing, finishing, inspection & packing
  - Demand (d) and price (p) have an inverse relationship



$$d_S = 2250 - 15p_S \Leftrightarrow p_s = (150 - \frac{1}{15}d_S)$$

$$d_D = 1500 - 5p_D \Leftrightarrow p_D = \left(300 - \frac{1}{5}d_D\right)$$

  - Total profit from producing x_S and x_D is thus:

$$\left(150 - \frac{1}{15}x_s\right)x_S - 70x_s + \left(300 - \frac{1}{5}x_D\right)x_D - 150x_D$$

$$= 80x_s - \frac{1}{15}x_s^2 + 150x_D - \frac{1}{5}x_D^2$$

NLP Example: Production planning (Cont'd)

$$\max 80x_s - \frac{1}{15}x_s^2 + 150x_D - \frac{1}{5}x_D^2$$

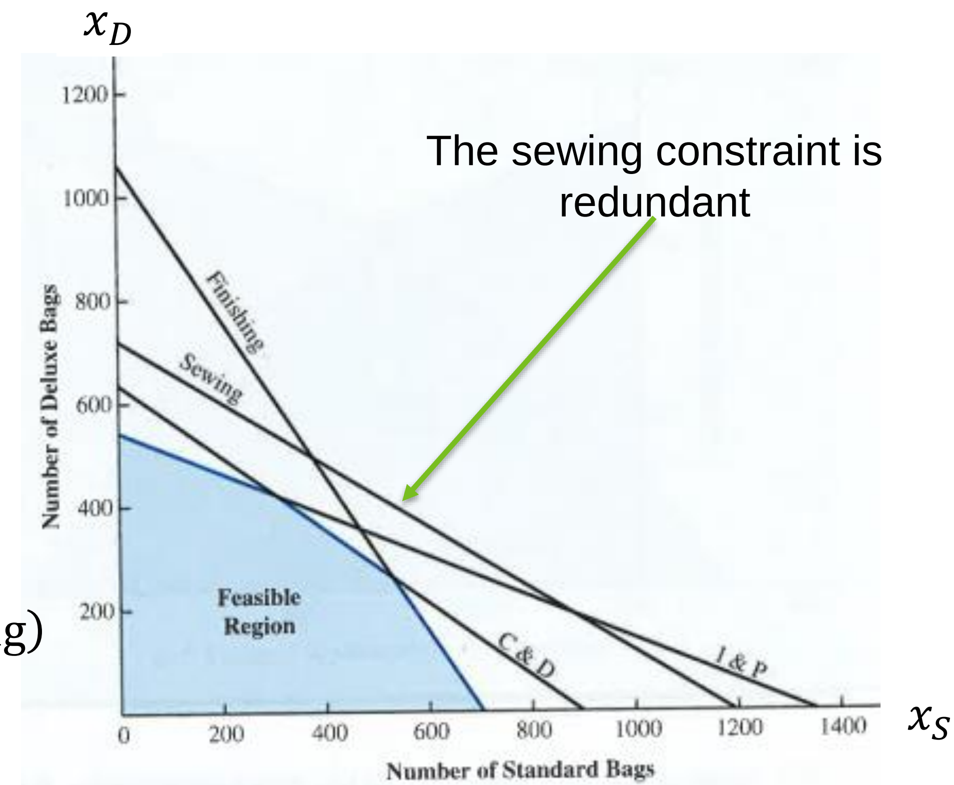
$$\frac{7}{10}x_S + x_D \leq 630 \text{ (cutting \& dying)}$$

$$\frac{1}{2}x_S + 5/6x_D \leq 600 \text{ (sewing)}$$

$$x_S + \frac{2}{3}x_D \leq 708 \text{ (finishing)}$$

$$\frac{1}{10}x_S + \frac{1}{4}x_D \leq 135 \text{ (inspection \& packing)}$$

$$x_S, x_D \geq 0$$



NLP Example: Production planning (Cont'd)

$$\max 80x_s - \frac{1}{15}x_s^2 + 150x_D - \frac{1}{5}x_D^2$$

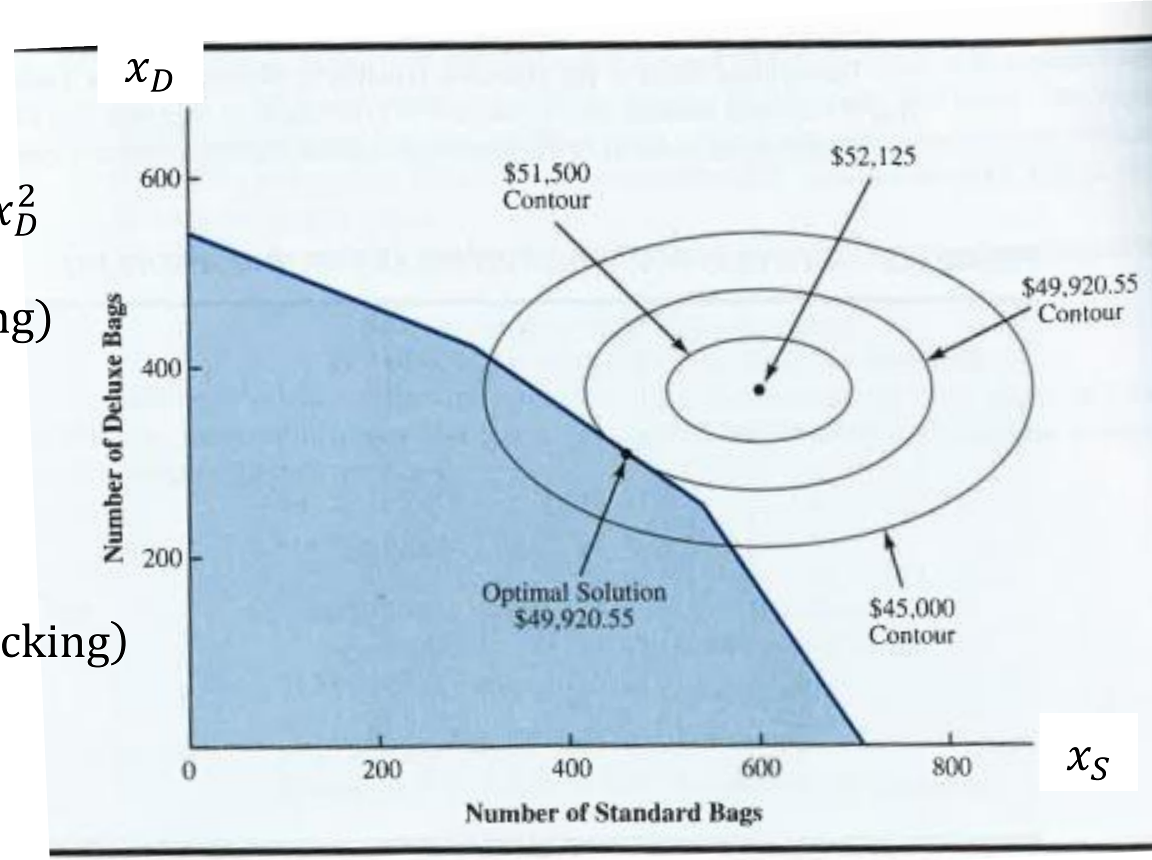
$$\frac{7}{10}x_S + x_D \leq 630 \text{ (cutting \& dying)}$$

$$\frac{1}{2}x_S + 5/6x_D \leq 600 \text{ (sewing)}$$

$$x_S + \frac{2}{3}x_D \leq 708 \text{ (finishing)}$$

$$\frac{1}{10}x_S + \frac{1}{4}x_D \leq 135 \text{ (inspection \& packing)}$$

$$x_S, x_D \geq 0$$



NLP Example: Production planning (Cont'd)

$$\max 80x_S - \frac{1}{15}x_S^2 + 150x_D - \frac{1}{5}x_D^2$$

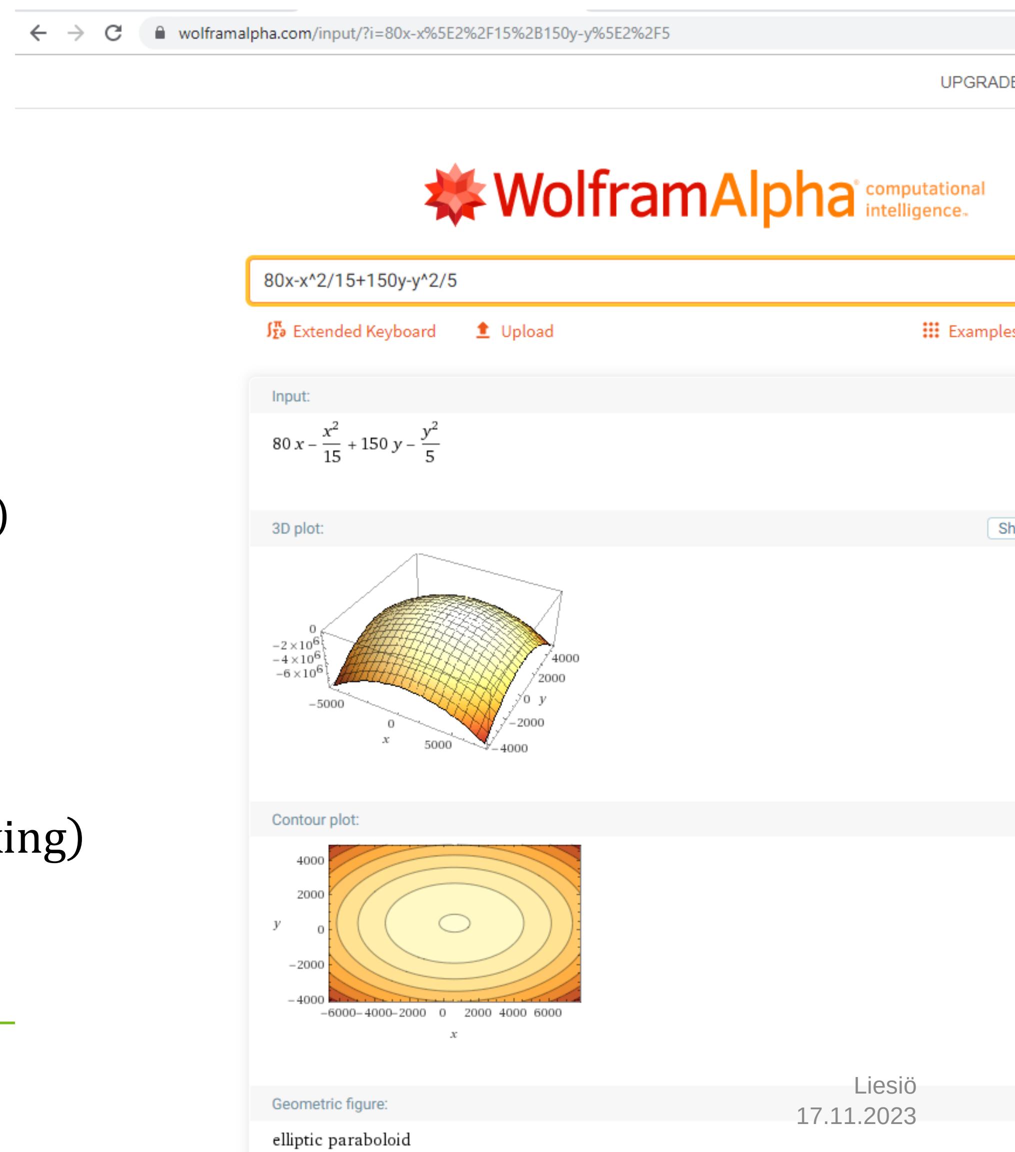
$$\frac{7}{10}x_S + x_D \leq 630 \text{ (cutting \& dying)}$$

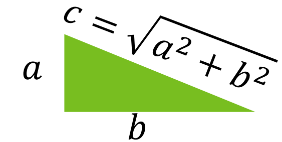
$$\frac{1}{2}x_S + 5/6x_D \leq 600 \text{ (sewing)}$$

$$x_S + \frac{2}{3}x_D \leq 708 \text{ (finishing)}$$

$$\frac{1}{10}x_S + \frac{1}{4}x_D \leq 135 \text{ (inspection \& packing)}$$

$$x_S, x_D \geq 0$$

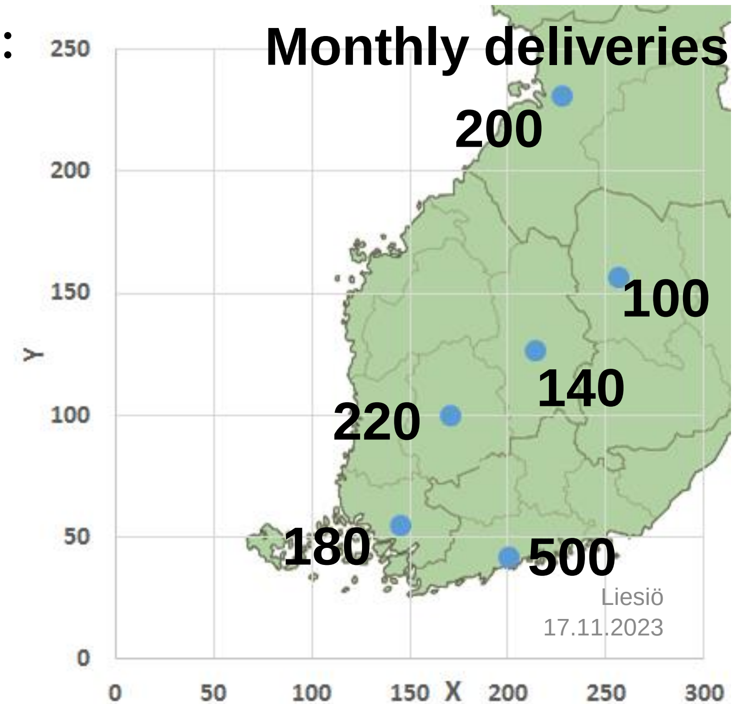




NLP Example: Location problem

- T-group is planning to use unmanned helicopters to deliver groceries across Finland
  - Market research has identified five areas with most demand (see map)
    - Helsinki (inc. Vantaa and Espoo), Turku, Tampere, Oulu, Jyväskylä, and Kuopio
- T-group has constructed the following NLP to choose a location for the new distribution center serving these areas:

$$
\begin{aligned}
\min\ & 500\sqrt{(x-200)^2+(y-42)^2} \\
& +220\sqrt{(x-170)^2+(y-100)^2} \\
& +200\sqrt{(x-227)^2+(y-231)^2} \\
& +180\sqrt{(x-145)^2+(y-55)^2} \\
& +140\sqrt{(x-214)^2+(y-127)^2} \\
& +100\sqrt{(x-256)^2+(y-157)^2} \\
& x, y \geq 0
\end{aligned}
$$



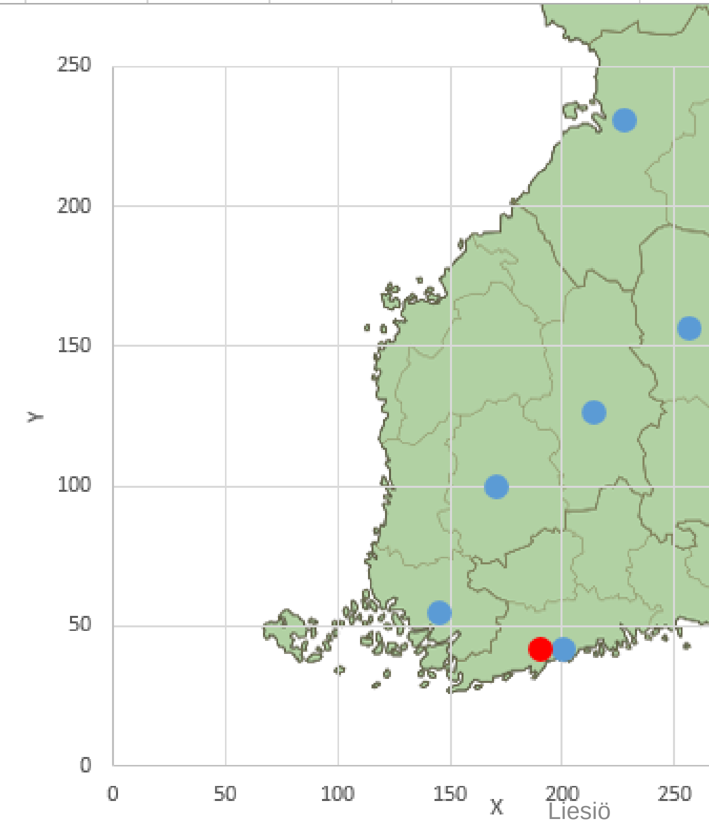
Question:

Interpret the NLP

NLP Example: Location problem – The spreadsheet

G15 =SQRT(($E$21-E15)^2+($F$21-F15)^2)

i	Area	Deliveries per month	x	y	Distance	Total dist. Per month
1	Helsinki	500	200	42	10	5000
2	Tampere	220	170	100	61	13497
3	Oulu	200	227	231	193	38518
4	Turku	180	145	55	47	8431
5	Jyväskylä	140	214	127	88	12365
6	Kuopio	100	256	157	133	13259
	DISTRIBUTION CENTER		190	42		91071



NLP Example: Location problem – The Solver

G15 =SQRT(($E$21-E15)^2+($F$21-F15)^2)

i	Area	Deliveries per month	x	y	Distance	Total dist. Per month
1	Helsinki	500	200	42	10	5000
2	Tampere	220	170	100	61	13497
3	Oulu	200	227	231	193	38518
4	Turku	180	145	55	47	8431
5	Jyväskylä	140	214	127	88	12365
6	Kuopio	100	256	157	133	13259
	DISTRIBUTION CENTER		190	42		91071

Solver Parameters

Set Objective: $H$21

To: Max Min Value Of: 0

By Changing Variable Cells:

$E$21:$F$21

Subject to the Constraints:

Make Unconstrained Variables Non-Negative

Select a Solving Method: GRG Nonlinear

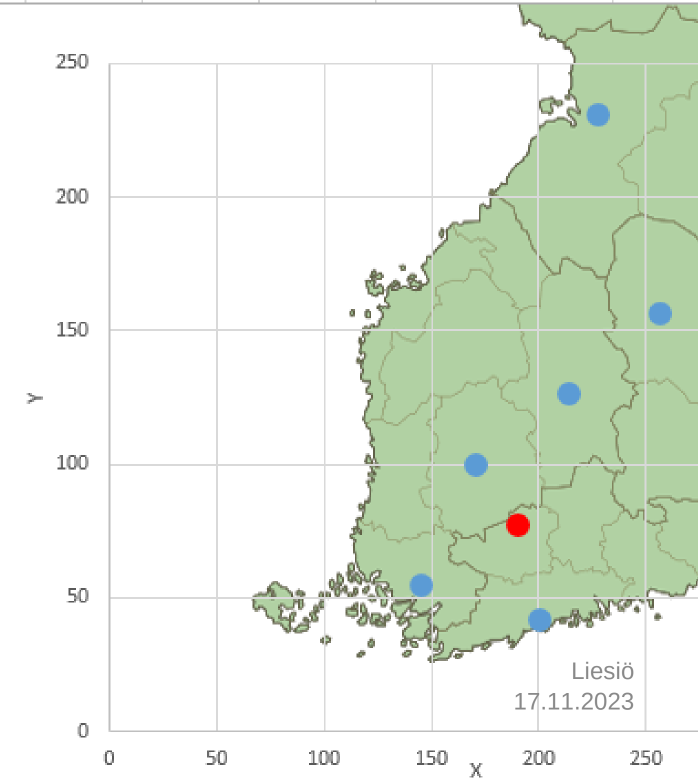
NLP Example: Location problem – Optimal solution

Question:

- Near which city should the distribution center be located?

H21 =SUM(H15:H20)

i	Area	Deliveries per month	x	y	Distance	Total dist. Per month
1	Helsinki	500	200	42	37	18437
2	Tampere	220	170	100	30	6625
3	Oulu	200	227	231	158	31580
4	Turku	180	145	55	50	9058
5	Jyväskylä	140	214	127	55	7701
6	Kuopio	100	256	157	103	10332
	DISTRIBUTION CENTER		190	77		83733



Formulas in Excel that result in a NLP model

Problem parameters (data) located in cells D1:D6

Decision variables located in cells C1:C6.

SUMPRODUCT(D4:D6, C4:C6)
[(D1 + D2) / D3] * C4
SUM(D4:D6)
2*C1 + 3*C4 + C6
C1 + C2 + C3
IF(D1>D2; 1; 0)

LP model

SUMPRODUCT(C4:C6, C1:C3)
[(C1 + C2) / C3] * D4
ABS(C1)
SQRT(C1)
C1 * C2
C1 / C2
C1 ^2
IF(C1>D2; 1; 0)

NLP model

NLP Example: Markowitz Portfolio optimization

- Hauck Financial Services allocates capital to 6 funds
  - Historical fund returns are used to construct 5 samples of possible returns for 2022 (scenarios)
  - Hauck aims at a 10% expected return with minimal risk

Question: Interpret the NLP model:

$$\min 0.2 \sum_{s=1}^{5} (r_s - \bar{r})^2$$

$$r_1 = 10.06x_A + 17.64x_B + \cdots + 24.56x_F$$
$$r_2 = 13.12x_A + 3.25x_B + \cdots + 25.32x_F$$
$$\ldots$$
$$r_5 = -21.93x_A + 7.36x_B + \cdots + 17.31x_F$$
$$\bar{r} = 0.2r_1 + 0.2r_2 + 0.2r_3 + 0.2r_4 + 0.2\, r_5$$
$$x_A + x_B + x_C + x_D + x_E + x_F = 1$$
$$\bar{r} \geq 10$$
$$x_A, \ldots, x_F \geq 0$$

	Historical returns				
Fund	**2017**	**2018**	**2019**	**2020**	**2021**
A	10.06	13.12	13.47	45.42	-21.93
B	17.64	3.25	7.51	-1.33	7.36
C	32.41	18.71	33.28	41.46	-23.26
D	32.36	20.61	12.93	7.06	-5.37
E	33.44	19.4	3.85	58.68	-9.02
F	24.56	25.32	-6.7	5.43	17.31

Statistics recap: For a random variable R that receives value r_i with probability p_i the expected value is $\mathrm{E}[R] = \sum_i p_i r_i$ and the variance is $\mathrm{Var}[R] = \sum_i p_i (r_i - E[R])^2$

NLP Example: Markowitz Portfolio optimization

L13 | =0.2*SUM(F13:J13)

	Scenario						
Fund	1	2	3	4	5		1
A	10.06	13.12	13.47	45.42	-21.93		0.158407
B	17.64	3.25	7.51	-1.33	7.36		0.52548
C	32.41	18.71	33.28	41.46	-23.26		0.042065
D	32.36	20.61	12.93	7.06	-5.37		0
E	33.44	19.4	3.85	58.68	-9.02		0
F	24.56	25.32	-6.7	5.43	17.31		0.274048
Scenario-specific return	18.96	11.51	5.64	9.73	4.16	Mean return	10.00
Squared deviation from mean mean return	80.23	2.29	18.98	0.07	34.12	Variance of return	27.14
						St. Dev. of return	5.21

Return target
10

Solver Parameters

Set Objective: $L$13

To: Max / Min (selected)

By Changing Variable Cells: $L$4:$L$9

Subject to the Constraints:
$L$2 = 1
$L$11 >= $P$11

Make Unconstrained Variables Non-Negative (checked)

Select a Solving Method: GRG Nonlinear

Constraints

Cell	Name	Final Value	Lagrange Multiplier
$L$2	Scenario	1	-7.523319825
$L$11	Scenario-specific return	10.00000002	6.179574311

NLP Example: Markowitz Portfolio optimization

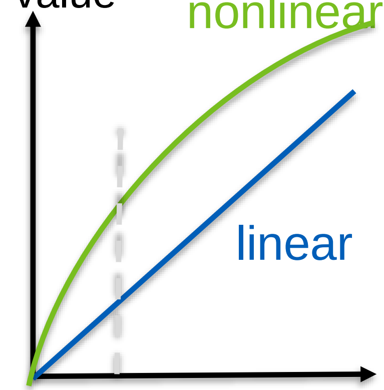
- Increasing return target by 0.1 units increases variance by about 0.62 (=27.76-27.14).
  - Consistent with “Lagrange multiplier 6.179” on the previous slide since 0.1*6.179=0.62

	Scenario					
Fund	**1**	**2**	**3**	**4**	**5**	1
A	10.06	13.12	13.47	45.42	-21.93	0.152692
B	17.64	3.25	7.51	-1.33	7.36	0.518809
C	32.41	18.71	33.28	41.46	-23.26	0.049069
D	32.36	20.61	12.93	7.06	-5.37	0
E	33.44	19.4	3.85	58.68	-9.02	0
F	24.56	25.32	-6.7	5.43	17.31	0.27943
Scenario-specific return	19.14	11.68	5.71	9.80	4.17	10.10 Mean return
Squared deviation from mean mean return	81.74	2.50	19.24	0.09	35.22	27.76 Variance of return
						5.27 St. Dev. of return

Return target
10.1

Sensitivity analysis in NLP

- Shadow price in **LP** is the rate of change in the objective function as the RHS of a constraint increases (all other data unchanged)
  - This rate is constant for a range of RHS values ("range of feasibility")
- In **NLP**s this rate is called "Lagrange multiplier"
  - However, in NLP the rate does not generally remain constant
  - It can be guaranteed to hold only for the current RHS value
    - Cf. A range of feasibility where both range upper and lower bound are equal to current RHS value

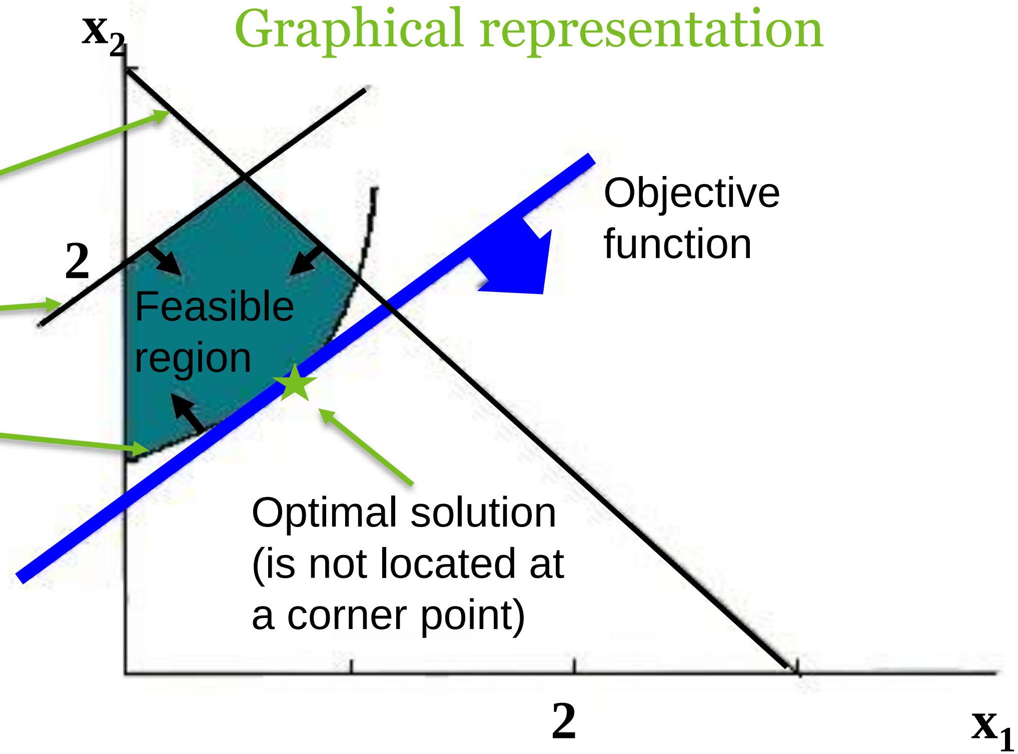


NLP Example: Linear objective function

Mathematical formulation

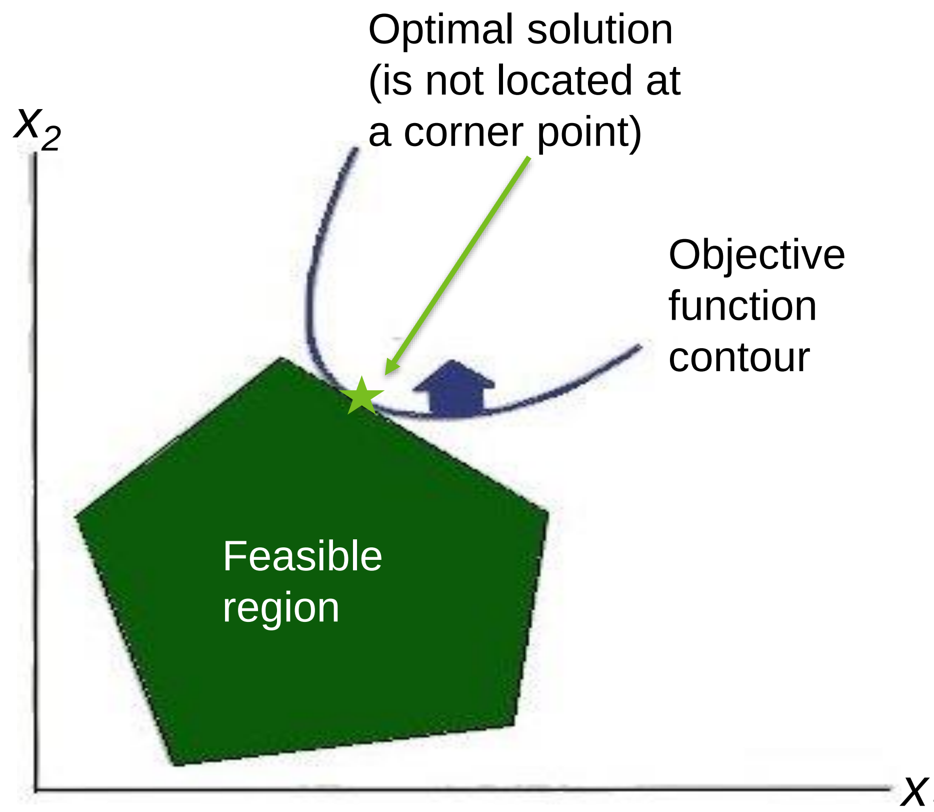
$$\max\ x_1 - x_2$$
$$s.t.\ \ x_1 + x_2 \le 3$$
$$-x_1 + x_2 \le 2$$
$$-x_1^2 + x_2 \ge 1$$
$$x_1, x_2 \ge 0$$

Graphical representation



NLP example: All constraints linear

- This NLP problem has
  - 2 decision variables
  - 5 linear constraints
  - a nonlinear objective function

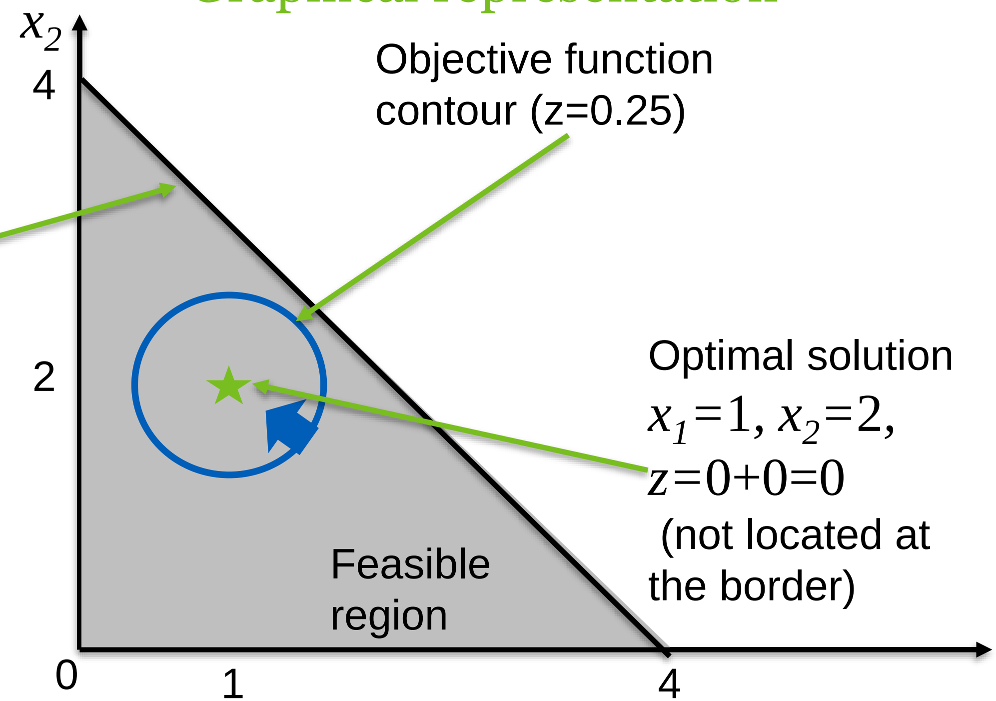


NLP Example: Optimum not on the border of the feasible region

Mathematical formulation

$$\min \; z=(x_1-1)^2 + (x_2-2)^2$$
$$s.t. \;\; x_1 + x_2 \le 4$$
$$x_1, x_2 \ge 0$$

Graphical representation



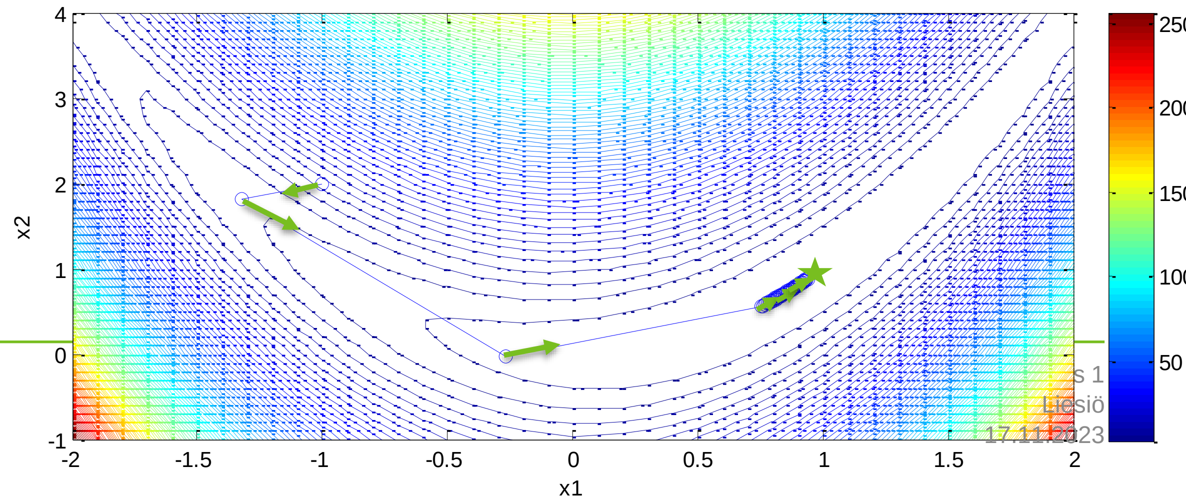
Computer solution to NLP problems

- The GRG algorithm in Solver is based on gradient search (“hill-climbing”)
  - With the initial starting solution, a direction is computed that most rapidly improves the objective function value
  - Solution is moved (values of decision variables changed) to this direction until
    - a constraint boundary is encountered OR
    - the objective function value no longer improves
  - A new direction is computed with the new solution and the process is repeated until no further improvement in any direction is possible
- Example:

  $\min\ (x_2 - x_1^2)^2 + (1 - x_1)^2$

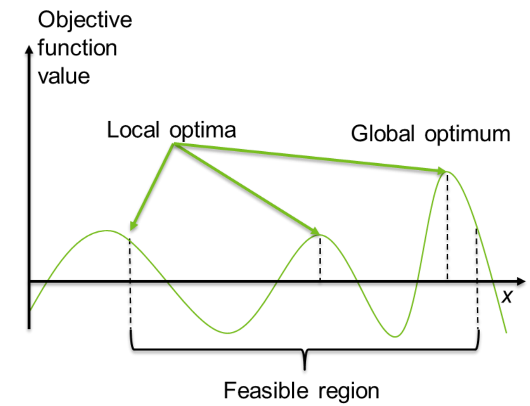
  s.t. $x_1, x_2 \geq -2$

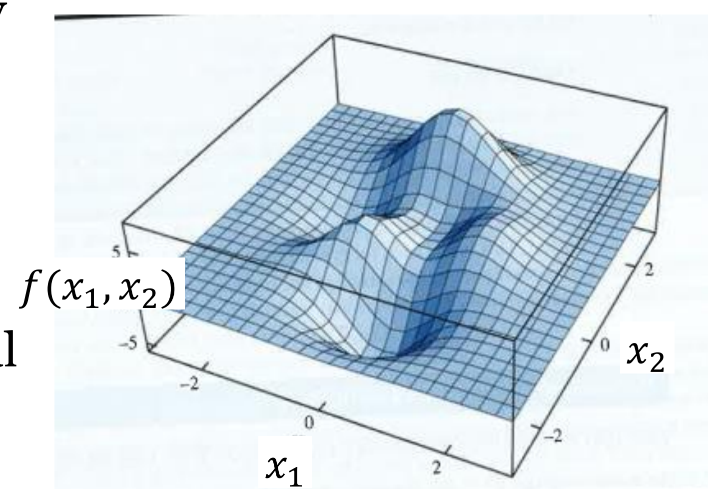
  Initial solution (-1,2)



Global and local optimal solutions

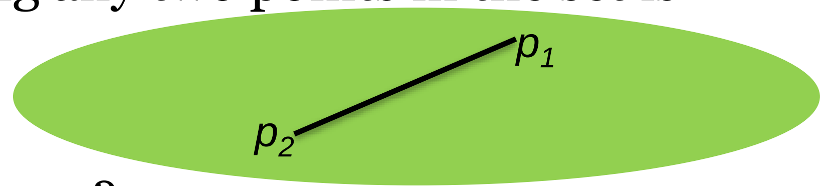
- For NLP problems we often do **not** have a guarantee that the optimal solution is a true **global** optimal solution
  - I.e. no other feasible solution provides a better objective function value
- Most NLP algorithms terminate when they have found a **local** optimal solution
  - I.e. a feasible solution such that all neighboring feasible solutions are worse
- Special case:
  - If NLP is "convex" or "concave" then any local optimal solution is a global optimal solution.



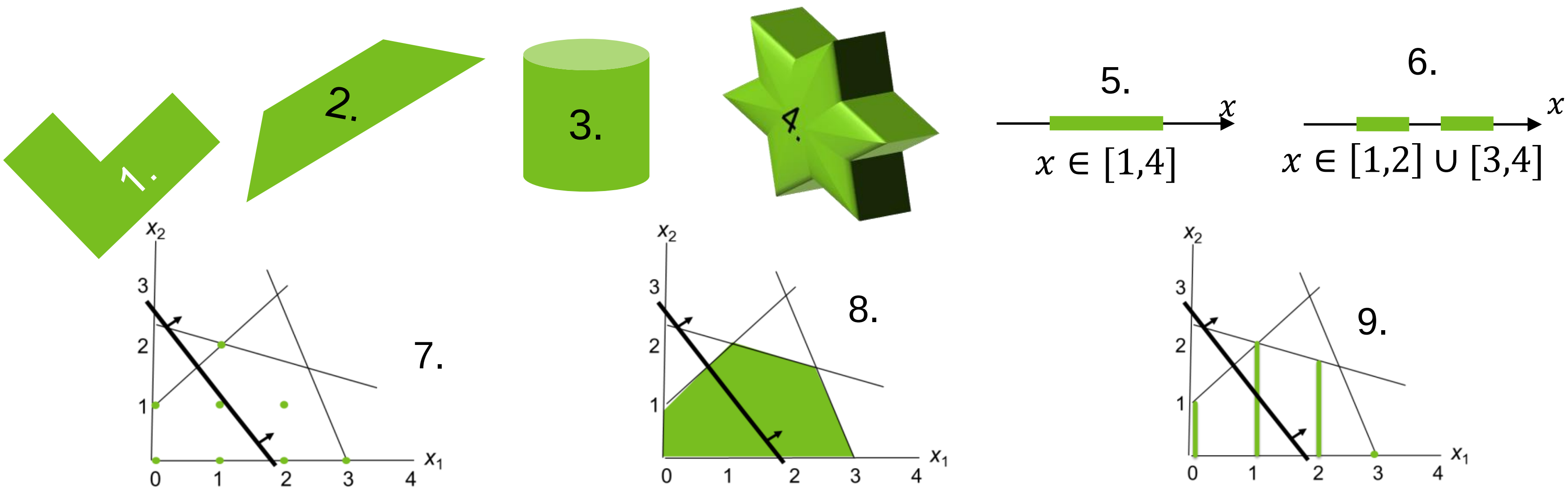


Convex sets

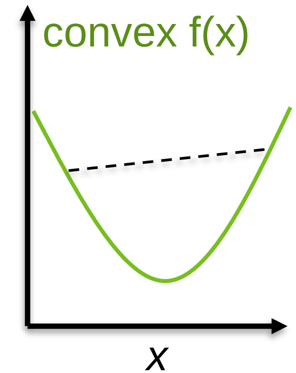
- **A set is convex** if a line connecting any two points in the set is contained entirely in the set

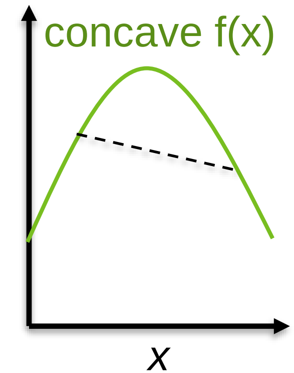


Question: Which of these sets are convex?



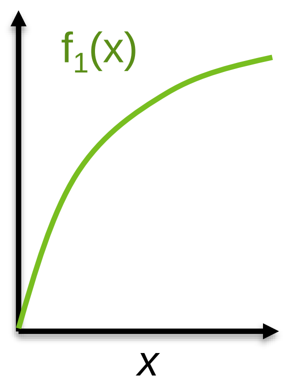
Convex and Concave functions

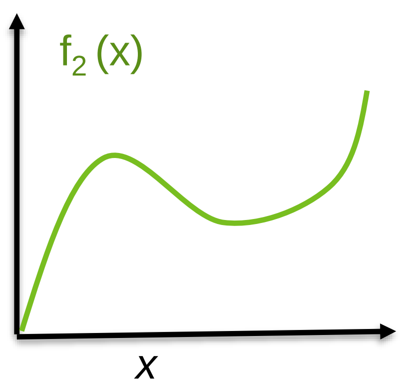




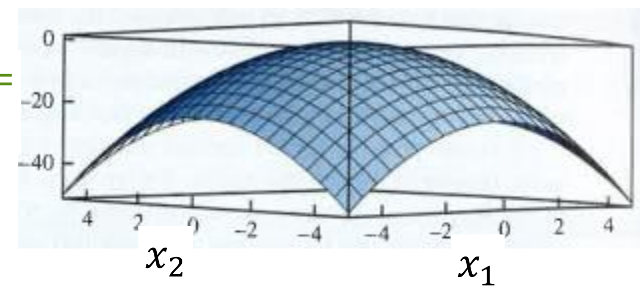
- **A function is convex** if a line connecting any two points lies above the function
- **A function is concave** if a line connecting any two points lies below the function

Question: Which of these 4 functions are concave and which are convex?

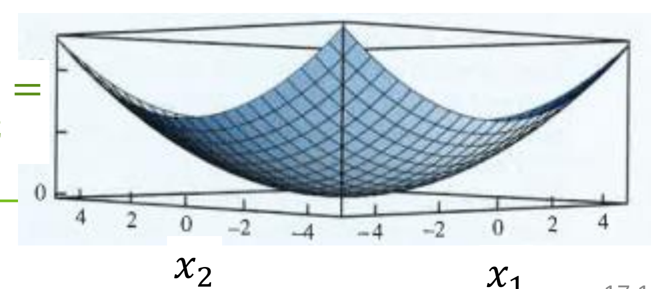




$f_3(x_1, x_2) = -x_1^2 - x_2^2$

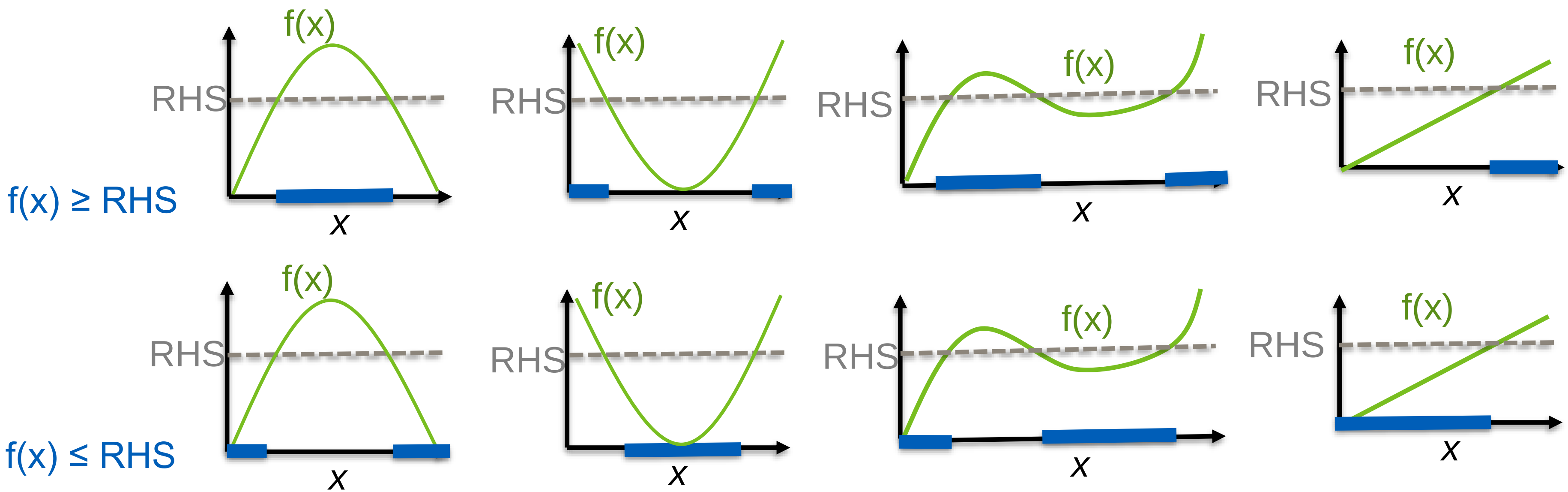


$f_4(x_1, x_2) = x_1^2 + x_2^2$



Convexity of sets defined by inequalities

- Constraint types in NLP: $f(x) \leq$ RHS or $f(x) \geq$ RHS (or both: $f(x)$=RHS)
- Question: When is the feasible region (= the set of values for x that satisfy the constraint) convex?



Convex and Concave NLPs

The result from the previous slide can be generalized:

- If in an NLP the LHS function of each ≤ (≥) constraint is convex (concave) then the feasible region is convex

Convex NLP:

- A convex objective function is minimized
- The feasible region is convex

Concave NLP:

- A concave objective function is maximized
- The feasible region is convex

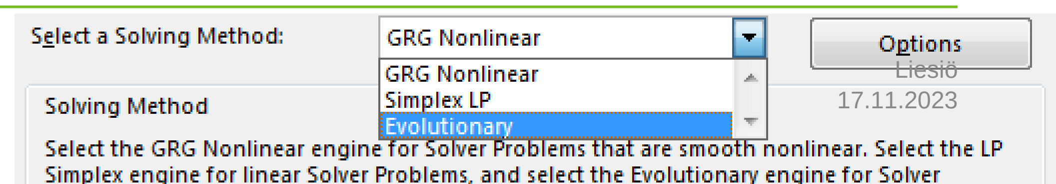
Same condition!

Property of concave and convex NLPs:

- Any local optimal solution is necessarily a global optimal solution.

Evolutionary optimization algorithms

- If an optimization problem is non-linear but its not a convex/concave NLP, how to solve it?
  - One possibility: Evolutionary Algorithms.
- Evolutionary algorithms are **heuristic**, i.e., provide a feasible solution with a “good” objective value, but no guarantees that it is optimal
  - Idea: A large set of solutions (”population”) simulated through multiple iterations (”generations”)
  - On each iteration :
    - Solutions with best objective function value (”fitness”) are combined to produce new solutions (”reproduction”)
    - Random changes to some solutions (”mutation”)
    - Infeasible solution and those with poor objective function value (”unfit”) are deleted
- Excel solver includes an evolutionary algorithm



Non-linear programming (NLP) - Summary

- Can contain a nonlinear objective function or one or more nonlinear constraints
  - Some constrains or the objective function can be linear
- Relaxation of constraints has the same effect as in LP
  - Cannot make objective function value worse
- Lagrange multiplier captures the effect of changes in the RHS of constraints
  - Holds only locally, not for a range of RHS values
- Most NLP algorithms do not ensure a global optimal solution
- For concave/convex NLPs a local optimal solution is also global
  - I.e., "max concave function"/"min convex function" over a convex feasible region

Extra slides

Equivalence between the two formulations of the Markowitz model

Classical formulation

- Return of the ith asset is a random variable R_i such that:
  - $\mathrm{P}(R_i = r_{si}) = \frac{1}{n}, \; s = 1, \dots, n$
  - $\mathrm{E}[R_i] = \bar{r}_i$
- Portfolio return:
  - In scenario s: $r_s = \sum_i x_i r_{si}$
  - Expected:

$\bar{r} = \mathrm{E}[\sum_i x_i R_i] = \sum_i x_i \mathrm{E}[R_i] = \sum_i x_i \bar{r}_i$

$$
\begin{aligned}
\boldsymbol{Var}\left(\sum_i x_i R_i\right) &= \sum_i \sum_j x_i x_j \mathrm{Cov}(R_j, R_i) \\
&= \sum_i \sum_j x_i x_j \left[\frac{1}{n} \sum_{s=1}^{n} (r_{si} - \bar{r}_i)(r_{sj} - \bar{r}_j)\right] \\
&= \frac{1}{n} \sum_s \sum_i x_i (r_{si} - \bar{r}_i) \sum_j x_j (r_{sj} - \bar{r}_j) \\
&= \frac{1}{n} \sum_s \sum_i (x_i r_{si} - x_i \bar{r}_i) \sum_j (x_j r_{sj} - x_j \bar{r}_j) \\
&= \frac{1}{n} \sum_s (r_s - \bar{r})(r_s - \bar{r}) = \frac{1}{n} \sum_s (r_s - \bar{r})^2
\end{aligned}
$$

Scenario based formulation