Aalto University
School of Business

Multi-Objective Programming (MOP)

- *Terminology*
- *Graphical representation of MOP problems*
- *Efficient solutions: Definition*
- *Examples*
- *The weighted sum approach for solving efficient solutions*
- *Goal programming*

Next Monday's guest lecture:

"Optimisation in Energy Transition"

Matti Vuorinen, Director, Digital Solutions in UPM Energy

Multi-objective programming problems

- Many problems have multiple objectives:
  - Planning the national budget
    - improve social security, reduce debt, cut taxes, build national defense
  - Admitting students to college
    - high SAT or GMAT, high GPA, diversity
  - Planning an advertising campaign
    - reach, expenses, target groups
  - Designing a distribution system:
    - minimize transportation costs, minimize CO_2emissions
  - Choosing taxation levels
    - raise money for government, incentives for work, minimize flight of business
  - Planning an investment portfolio
    - maximize expected returns, minimize risk

MOP and MOO terminology

- **O**ptimization/**P**rogramming problems with multiple objective functions are called Multi-objective (**MO**)
  - MOLP = Multi-Objective Linear Programming
  - MOILP =
  - MOZOLP =
  - MONLP =
  - MOINLP =

- The term “Bi-objective” is sometimes used to highlight that a problem has only two objective functions
- Both the terms “criteria” and “objectives” are used
  - E.g. Multi-criteria linear programming

MOLP Example

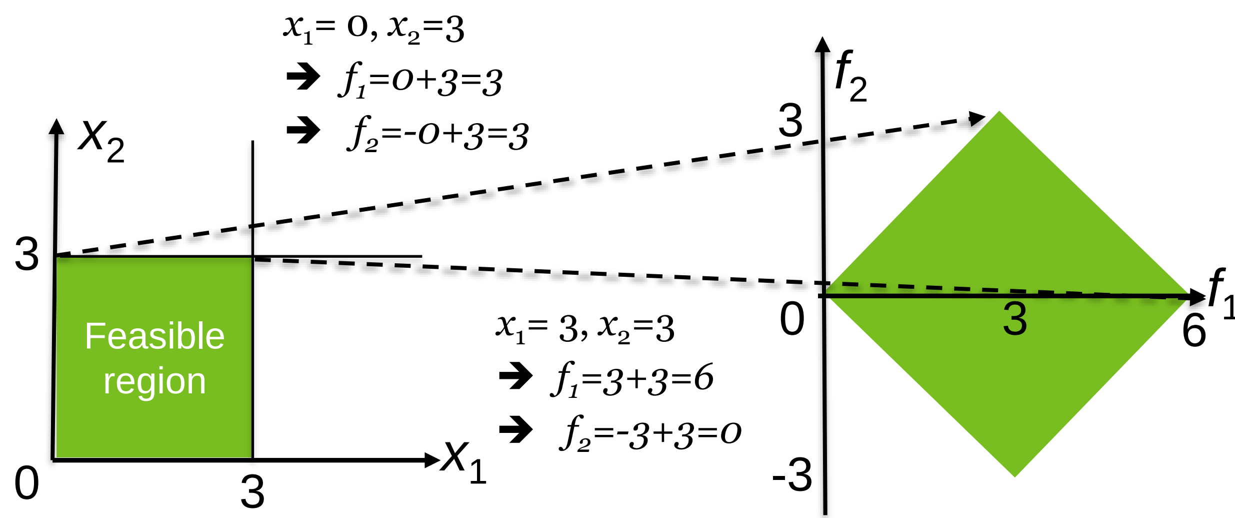
Graphical representation in the...

Math. formulation

$$\text{Max } f_1 = x_1 + x_2$$
$$\text{Max } f_2 = -x_1 + x_2$$
$$\text{s.t. } x_1 \leq 3$$
$$x_2 \leq 3$$
$$x_1, x_2 \geq 0$$

decision variable space

objective function space



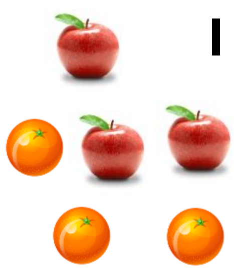
What is an “optimal solution” to a MOP problem?

- Generally, there does not exist a feasible solution that simultaneously optimizes all the objective functions

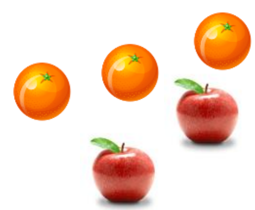
- Assume your objectives are to
  (i) maximize the number of oranges
  (ii) minimize the number of apples

  Which of the fruit baskets would you choose?

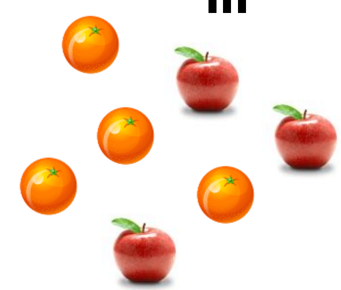
I



II



III



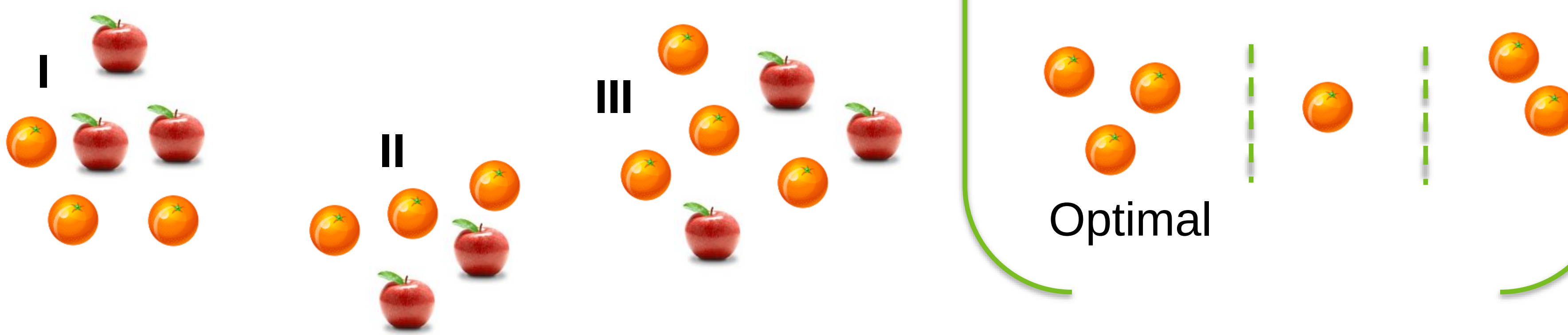
Efficient solutions: Definition

- **Definition**: A feasible solution to MOP problem is <u>efficient</u>, if there does not exist another feasible solution which yields

  (i) a better or equal value in each objective function AND

  (ii) a strictly better value in some objective function.

C.f. in single objective optimization problems a feasible solution is optimal if there does not exists another feasible solution which yields a strictly better objective function value



Efficient solutions: Alternative equivalent definition

- Consider a MOP problem with n objective functions $f_1(x), \dots, f_n(x)$ to be maximized

- **Definition:** Solution x <u>dominates</u> solution x' if
  $f_i(x) \geq f_i(x')$ for all $i \in \{1, \dots, n\}$, and
  $f_i(x) > f_i(x')$ for some $i \in \{1, \dots, n\}$.

- **Definition:** A feasible solution x is <u>efficient</u> if it is not dominated by any other feasible solution

- The term "non-dominated solution" is sometimes used instead of the term "efficient solution"

Efficient solutions example: Marketing Plan

- The Supersuds Corporation is developing its next year's marketing plan
  - Spots on five TV shows purchased under limited budget
    - $x_j \in \{0,1\}, j = 1, \dots, 5$: purchase spot in jth show
  - Objective is to maximize reach in three important consumer groups
    - $f_i, i = 1, \dots, 3$: reach in the ith consumer group

Question:

- Supersuds has identified four feasible solutions: Which of them are efficient solutions?

	$f_1(x)$	$f_2(x)$	$f_3(x)$
A: $x = (1,1,1,0,0)$	1000	700	200
B: $x = (1,0,1,1,0)$	700	500	300
C: $x = (1,0,1,0,1)$	100	800	400
D: $x = (0,0,1,1,1)$	900	600	150

Example: Efficient solutions in MOLP

Graphical representation in the…

Math. formulation

Max $f_1 = x_1 + x_2$

Max $f_2 = -x_1 + x_2$

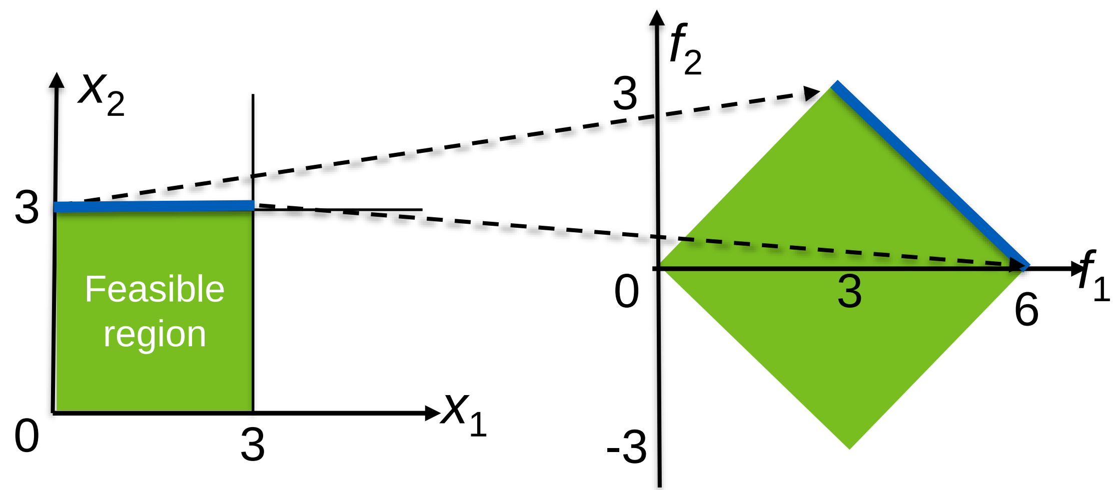
s.t. $x_1 \leq 3$

$x_2 \leq 3$

$x_1, x_2 \geq 0$

Decision variable space

Objective function space



Solving efficient solutions

- For some special classes of problems (e.g. MOLP, MOILP, MOMILP) there are algorithms that identify the entire set of efficient solutions

- Most available methods transform the MOP problem into a single objective problem and then solve it using standard algorithms
  - E.g., Simplex for LPs; B&B for MILPs; Gradient search for NLPs
  - These methods generate one efficient solution on each run
  - Approaches:
    - Set target levels (=constraints) for all but one of the objective functions
      - C.f. return at least 13% and minimize risk (=variance)
    - Maximize the **weighted sum** of the objectives functions (next slides)
      - E.g. max 1*(# of oranges)-2*(# of apples)

Weighted sum approach

General MOP formulation

$$\max \; f_1(x_1,\ldots,x_m)$$
$$\max \; f_2(x_1,\ldots,x_m)$$
$$\ldots\ldots$$
$$\max \; f_n(x_1,\ldots,x_m)$$

subject to constraints on decision variables $x_1,\ldots,x_m$

The general formulation can always be obtained by replacing "min $f_i(x_1,\ldots,x_m)$" with "max $-f_i(x_1,\ldots,x_m)$"

- Weighted sums approach
    1. Select (at random) positive weights $w_1,\ldots,w_n$ for the objective functions
    2. Solve the single objective optimization problem ➔ Solution is efficient
  - Repeat Steps 1 and 2 until enough efficient solutions have been found

Weighted sum formulation of MOP

$$\max \; \sum_{i=1}^{n} w_i f_i(x_1,\ldots,x_m)$$

subject to constraints on decision variables $x_1,\ldots,x_m$

Example: Weighed sum approach in MOLP

MOLP math. formulation

Max $f_1 = x_1 + x_2$

Max $f_2 = -x_1 + x_2$

s.t. $x_1 \leq 3$

$x_2 \leq 3$

$x_1, x_2 \geq 0$

Weighted sum formulation

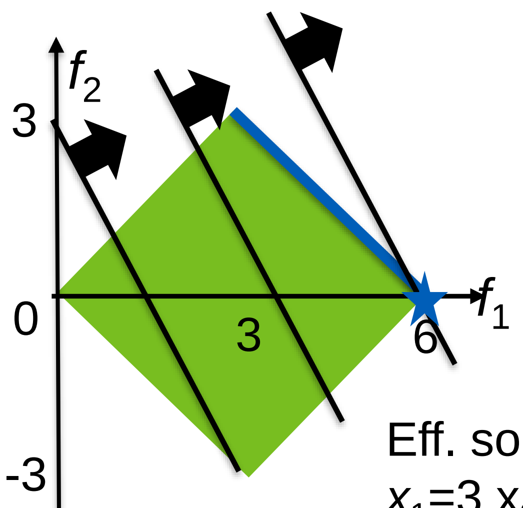
Max $w_1(x_1 + x_2) + w_2(-x_1 + x_2)$

s.t. $x_1 \leq 3$

$x_2 \leq 3$

$x_1, x_2 \geq 0$

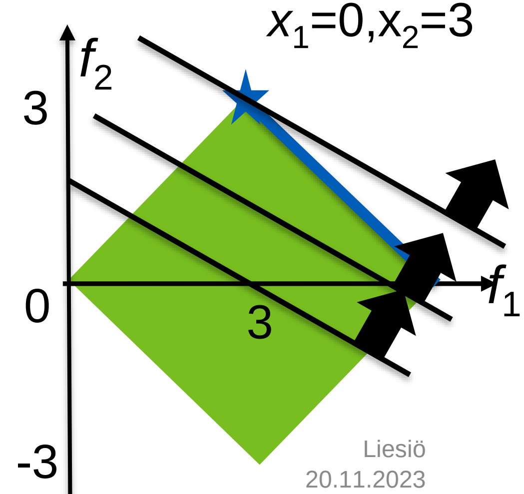
w_1=2, w_2=1: "Unit increase in objective 1 is equally important to increase of two units in objective 2"



Eff. solution x_1=3,x_2=3

w_1=1, w_2=2: "Unit increase in objective 2 is equally important to increase of two units in objective 1"

Eff. solution x_1=0,x_2=3



MONLP Example: The Markowitz Model revisited

- Hauck Financial Services allocates capital to 6 funds
  - Historical fund returns are used to construct 5 samples of possible returns for 2019 (scenarios)
  - Objective: max. expected return & min. standard deviation of return

$$\max 0.2 \sum_{s=1}^{5} r_s$$

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

$$r_1 = 10.06x_A + 17.64x_B + \cdots + 24.56x_F$$
$$r_2 = 13.12x_A + 3.25x_B + \cdots + 25.32x_F$$
$$\ldots$$
$$r_5 = -21.93x_A + 7.36x_B + \cdots + 17.31x_F$$
$$x_A + x_B + x_C + x_D + x_E + x_F = 1$$
$$x_A, \ldots, x_F \geq 0$$

	Historical returns				
Fund	**2013**	**2014**	**2015**	**2016**	**2017**
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 $\mathrm{E}[R] = \sum_i p_i r_i$, the variance is $\mathrm{Var}[R] = \sum_i p_i (r_i - E[R])^2$, and the standard deviation is $\sqrt{\mathrm{Var}[R]}$

Weighed sum approach for the Markowitz Model

- MONLP with all objective functions maximized:

$$\max 0.2 \sum_{s=1}^{5} r_s$$

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

$$r_1 = 10.06x_A + 17.64x_B + \cdots + 24.56x_F$$
$$r_2 = 13.12x_A + 3.25x_B + \cdots + 25.32x_F$$
$$\cdots$$
$$r_5 = -21.93x_A + 7.36x_B + \cdots + 17.31x_F$$
$$x_A + x_B + x_C + x_D + x_E + x_F = 1$$
$$x_A, \dots, x_F \geq 0$$

- Weighted sum formulation:

$$\max w_1 \left(0.2 \sum_{s=1}^{5} r_s \right) + w_2 \cdot -\sqrt{0.2 \sum_{s=1}^{5} (r_s - \bar{r})^2}$$
$$r_1 = 10.06x_A + 17.64x_B + \cdots + 24.56x_F$$
$$r_2 = 13.12x_A + 3.25x_B + \cdots + 25.32x_F$$
$$\cdots$$
$$r_5 = -21.93x_A + 7.36x_B + \cdots + 17.31x_F$$
$$x_A + x_B + x_C + x_D + x_E + x_F = 1$$
$$x_A, \dots, x_F \geq 0$$

Weighed sum approach for the Markowitz Model

EXPON.DIST | =P11*L11-P14*L14

Fund	Scenario 1	2	3	4	5	L
						1
A	10.06	13.12	13.47	45.42	-21.93	0
B	17.64	3.25	7.51	-1.33	7.36	0
C	32.41	18.71	33.28	41.46	-23.26	0.110229
D	32.36	20.61	12.93	7.06	-5.37	0
E	33.44	19.4	3.85	58.68	-9.02	0.889771
F	24.56	25.32	-6.7	5.43	17.31	0

	1	2	3	4	5	L		Weights
Scenario-specific return	33.33	19.32	7.09	56.78	-10.59	21.19	Mean return	0.87
ion from mean mean return	147.36	3.47	198.62	1266.97	1009.78	525.24	Variance of return	
						22.92	St. Dev. of return	0.13

R: Weighted Sum — ?14*L14

Solver Parameters

Set Objective: $R$13

To: (•) Max ( ) Min

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

Subject to the Constraints: $L$2 = 1

☑ Make Unconstrained Variables Non-Negative

Select a Solving Method: GRG Nonlinear

Weighed sum approach for the Markowitz Model

- Lets solve it for 50 different weights generated randomly…

	K	L	M	N	O	P	Q	R	S
1									
2		1							
3									
4		0							
5		0							
6		0.11023							
7		0							
8		0.88977							
9		0							
10						Weights			
11		21.19	Mean return			0.87		Weighted	
12								Sum	
13		525.24	Variance of return					15.55	
14		22.92	St. Dev. of return			0.13			

T	U	V	W	X	Y	Z	AA	AB	AC	AD
	Obj. Fun. Weights		Opt. Obj. Fun. Values		Optimal solution (decision var. values)					
	w_1	w_2	E[R]	Std. R	A	B	C	D	E	F
1	0.87	0.13								
2	0.22	0.78								
3	0.46	0.54								
4	1.00	0.00								
5	0.41	0.59								
6	0.61	0.39								
7	0.39	0.61								
8	0.70	0.30								
9	0.48	0.52								
10	0.21	0.79								
11	0.58	0.42								
12	0.24	0.76								

Weighed sum approach for the Markowitz Model

- A lot of work … lets use a macro!

Microsoft Visual Basic for Applications - 5-Markowitz.xlsm

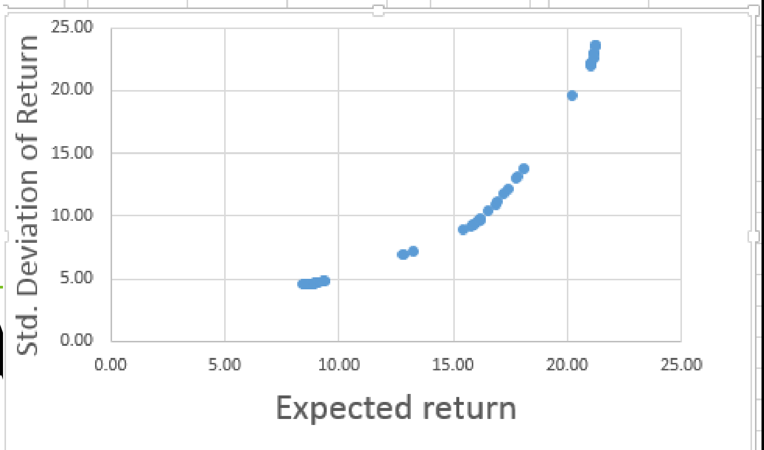
5-Markowitz.xlsm - Module1 (Code)

```
Sub RepeatedSolve()
RowCount = 52 ' Start from row 52

Range("W3:AD52").Value = ""

Do While RowCount > 2 'Repeat until row 3 has been processed
    ' Set weights for objective functions
    Cells(11, 16).Value = Cells(RowCount, 21).Value
    Cells(14, 16).Value = Cells(RowCount, 22).Value

    ' Solve NLP
    SolverSolve (True)
    SolverFinish keepFinal:=1

    ' Store solution: (i) objective function values
    Cells(RowCount, 23).Value = Cells(11, 12).Value
    Cells(RowCount, 24).Value = Cells(14, 12).Value

    ' Store solution: (i) decision variable values
    Cells(RowCount, 25).Value = Cells(4, 12).Value
    Cells(RowCount, 26).Value = Cells(5, 12).Value
    Cells(RowCount, 27).Value = Cells(6, 12).Value
    Cells(RowCount, 28).Value = Cells(7, 12).Value
    Cells(RowCount, 29).Value = Cells(8, 12).Value
    Cells(RowCount, 30).Value = Cells(9, 12).Value

    RowCount = RowCount - 1 'One row up

Loop
MsgBox "processing over"

End Sub
```

Weighed sum approach for the Markowitz Model

- Running the macro gives 50 efficient solutions

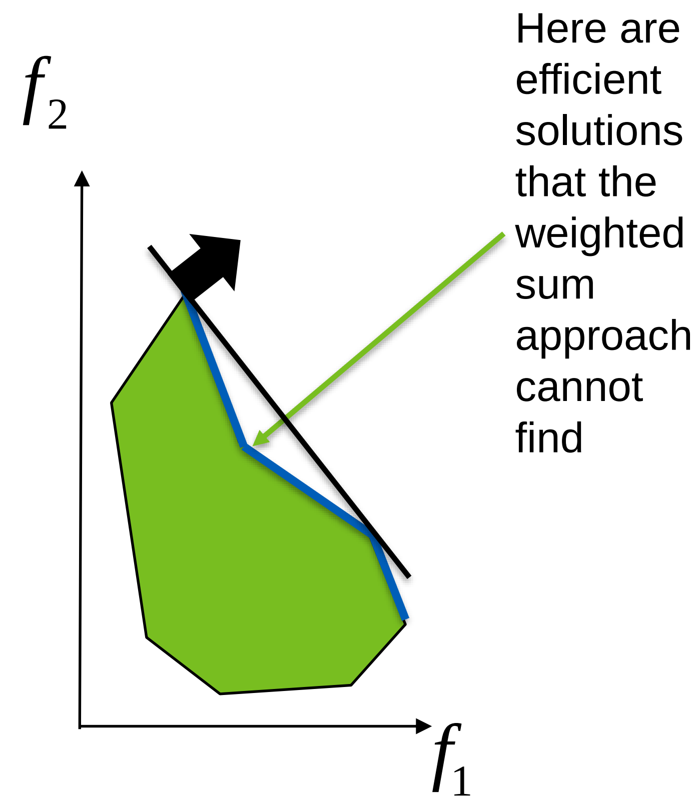
J	K	L	M	N	O	P	Q	R	S
		1							
5									
1.93		0							
.36		0							
3.26		0.110229							
.37		0							
.02		0.889771							
'.31		0							
						Weights			
).59		21.19	Mean return			0.87		Weighted	
								Sum	
9.78		525.24	Variance of return					15.55	
		22.92	St. Dev. of return			0.13			



T	U	V	W	X	Y	Z	AA	AB	AC	AD
	Obj. Fun. Weights		Opt. Obj. Fun. Values		Optimal solution (decision var. values)					
	w_1	w_2	E[R]	Std. R	A	B	C	D	E	F
1	0.87	0.13	21.19	22.92	0.00	0.00	0.11	0.00	0.89	0.00
2	0.22	0.78	8.87	4.63	0.18	0.65	0.00	0.00	0.00	0.17
3	0.46	0.54	15.46	8.94	0.00	0.03	0.34	0.00	0.00	0.63
4	1.00	0.00	21.27	23.55	0.00	0.00	0.00	0.00	1.00	0.00
5	0.41	0.59	13.24	7.22	0.00	0.29	0.25	0.00	0.00	0.46
6	0.61	0.39	16.21	9.78	0.00	0.00	0.41	0.00	0.00	0.59
7	0.39	0.61	12.83	6.94	0.00	0.33	0.24	0.00	0.00	0.43
8	0.70	0.30	17.43	12.20	0.00	0.00	0.40	0.00	0.17	0.44
9	0.48	0.52	15.78	9.22	0.00	0.00	0.35	0.00	0.00	0.65
10	0.21	0.79	8.86	4.63	0.18	0.65	0.00	0.00	0.00	0.17
11	0.58	0.42	16.05	9.53	0.00	0.00	0.39	0.00	0.00	0.61
12	0.24	0.76	8.94	4.65	0.18	0.64	0.00	0.00	0.00	0.18
[illegible]	0.27	0.73	9.04	4.68	0.18	0.62	0.00	0.00	0.00	0.19
[illegible]	0.36	0.64	9.39	4.85	0.19	0.57	0.00	0.00	0.00	0.24
[illegible]	0.90	0.10	21.27	23.55	0.00	0.00	0.00	0.00	1.00	0.00
[illegible]	0.02	0.98	8.46	4.57	0.17	0.72	0.00	0.00	0.00	0.11
[illegible]	0.02	0.98	8.47	4.57	0.17	0.72	0.00	0.00	0.00	0.11
[illegible]	0.18	0.82	8.77	4.60	0.18	0.67	0.00	0.00	0.00	0.16
19	0.67	0.33	16.85	10.93	0.00	0.00	0.40	0.00	0.09	0.51
20	0.72	0.28	18.07	13.77	0.00	0.00	0.39	0.00	0.25	0.36
21	0.13	0.87	8.66	4.58	0.17	0.69	0.00	0.00	0.00	0.14
22	0.24	0.76	8.94	4.65	0.18	0.64	0.00	0.00	0.00	0.18
23	0.68	0.32	16.97	11.18	0.00	0.00	0.40	0.00	0.10	0.49
24	0.69	0.31	17.24	11.76	0.00	0.00	0.40	0.00	0.14	0.46
25	0.10	0.90	8.61	4.58	0.17	0.69	0.00	0.00	0.00	0.13
26	0.71	0.29	17.82	13.14	0.00	0.00	0.39	0.00	0.22	0.39
27	0.10	0.90	8.61	4.58	0.17	0.69	0.00	0.00	0.00	0.13

Cautionary note about the weighted sum approach

- Every solution generated by the weighted sum approach is efficient
  - Assuming all weights are strictly positive
- However, if the feasible region is not convex, there can be efficient solutions that the weighted approach cannot find
  - These solution do not maximize the weighted sum for any weights
- For instance, MILP, ILP and BLP problems do not usually have a convex feasible region



Goal programming (GP)

- Idea: set goal for each objective function
  - The goals are listed in the order of their importance.
  - Begin by minimizing deviation from the most important goal
  - Do the same for the second most important goal, but require that the deviation from the from the first goal is not increased
  - Continue to the following goals, always requiring that the deviations from the previously optimized goals do not increase

Major drawback: May lead to a solution that is not efficient

- Different flavors exist: Preemptive GP (above), weighted GP,…

Multi-Objective Programming - Summary

- Optimization problems with multiple objective functions

- Instead of an optimal solution there is a (possibly infinite) set of efficient solutions
  - Definition: A feasible solution is efficient if no other feasible solution provides (i) an equal or better value in each objective function, and (ii) a strictly better value in at least one objective function
  - Terms “Pareto optimal solution” and “Non-dominated solution” widely used as synonyms for “Efficient solution”

- The are several methods for generating efficient solutions which make use of standard (i.e., single objective) solution algorithms
    - E.g. weighted sums approach