Aalto University
School of Science

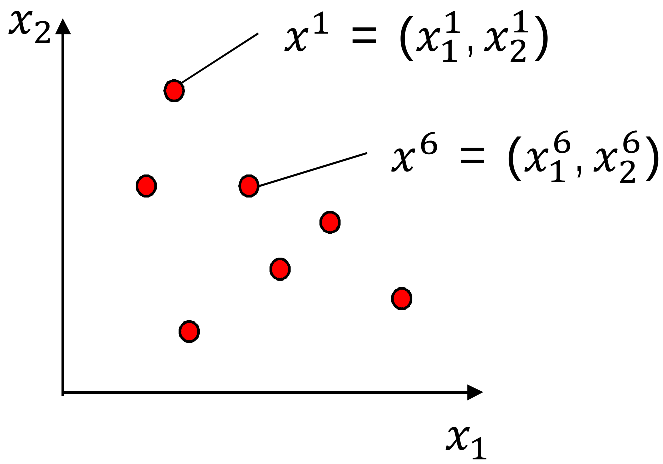
Decision making and problem solving – Lecture 8

- Multiple objective optimization (MOO)
- Pareto optimality (PO)
- Approaches to solving PO-solutions: weighted sum, weighted max-norm, and value function methods

Liesiö, Punkka, Salo, Vilkkumaa

Until this lecture

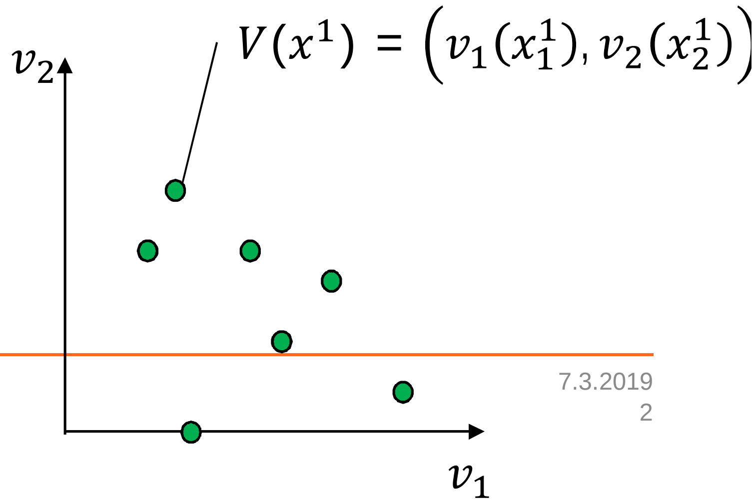
- Explicit set of alternatives $X = \{x^1, \dots, xm\}$, which are evaluated with regard to n criteria



- Evaluations $x_i^j : X \to \mathbb{R}^n$

- Preference modeling
  - Value functions

    $\max_{x^j \in X} V\left(x^j\right) = V(x_1^j, \dots, x_n^j)$



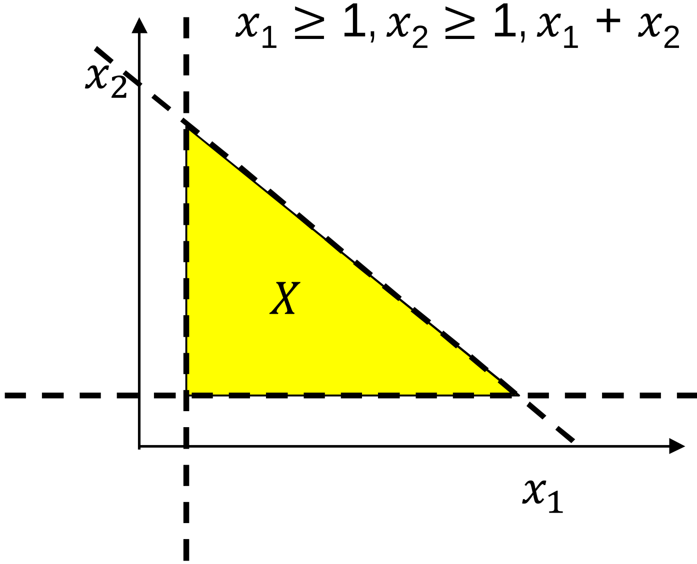
Need for other kind of approaches

- The decision alternatives cannot necessarily be listed
- Preference modeling can be time-consuming and difficult at the early stages of the analysis
- Conditions required for the additive value function to represent preferences do not necessarily hold or are difficult to validate
- We might want to see some results quickly to get a better understanding of the problem at hand

Multi-objective optimization: concepts

- Set of feasible solutions
  $$X = \{x \in \mathbb{R}^n | g(x) \leq 0\}$$
- Objective functions
  $$f = (f_1, \dots, f_n): X \to \mathbb{R}^n$$
- Preference modeling on trade-offs between objectives
  - Value functions
    $$\max_{x \in X} V(f(x)) = V(f_1(x), \dots, f_n(x))$$
  - Pareto approaches
    $$\text{v}-\max_{x \in X} V(f(x)) = (f_1(x), \dots, f_n(x))$$
  - Interactive approaches (not covered)

$$X = \{(x_1, x_2) \in \mathbb{R}^2 | x_1 \geq 1, x_2 \geq 1, x_1 + x_2 \leq 7\}$$



$$f = (f_1, f_2) = (x_1 + 2x_2, -x_2)$$

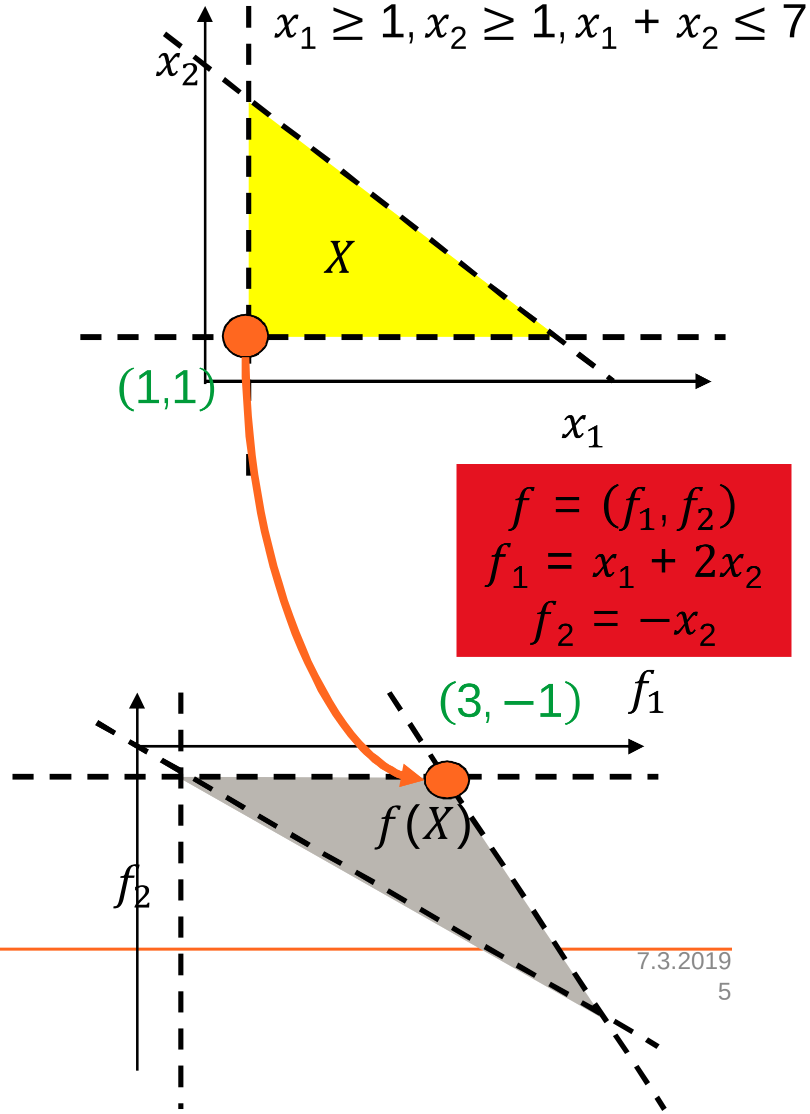
Multi-objective optimization: concepts

- Objective functions f map the feasible solutions X to $f(X)$ in the solution space:

$$f(X) = \{y \in \mathbb{R}^n \mid \exists x \in X \ so\ that\ y = f(x)\}$$

$$f(X) = \{(f_1, f_2) \in \mathbb{R}^2 \mid f_2 \leq -1, f_2 \leq 7 - f_1, 2f_2 \geq 1 - f_1\}$$

$$X = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 \geq 1, x_2 \geq 1, x_1 + x_2 \leq 7\}$$



Preferential independence

- In multi-objective optimization (MOO), each objective is assumed preferentially independent of the others

- **Definition** (cf. Lecture 5): Preference between two values of objective function *i* does not depend on the values of the other objective functions

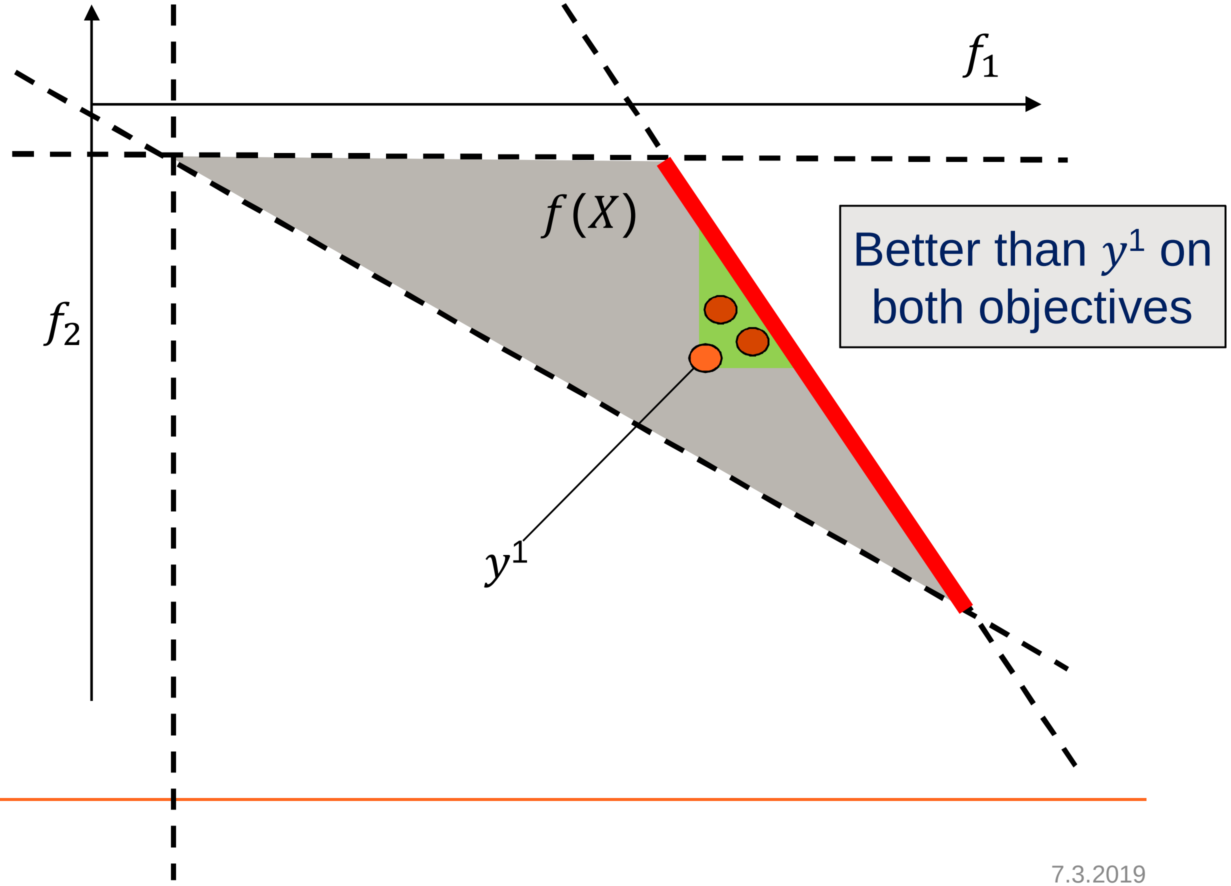
→ Without loss of generality, we can assume all objectives to be maximized

– MIN can be transformed to MAX: $\min_{x \in X} f_i(x) = -\max_{x \in X}[-f_i(x)]$

Which feasible solution(s) to prefer?

- Selection of y^1 cannot be supported because other solutions have higher f_1 and f_2

  $\rightarrow$ Focus on Pareto-optimal solutions

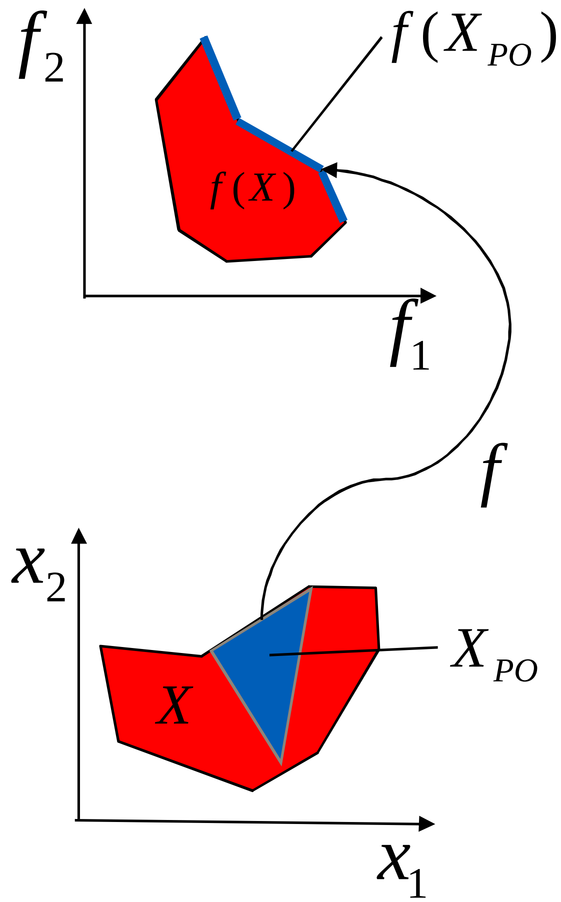


Pareto-optimality

Definition. $x^* \in X$ is Pareto-optimal if there does not exist $x \in X$ such that

$$\begin{cases} f_i(x) \geq f_i(x^*) & \text{for all } i \in \{1, \dots, n\} \\ f_i(x) > f_i(x^*) & \text{for some } i \in \{1, \dots, n\} \end{cases}$$

Set of all Pareto-optimal solutions: X_{PO}

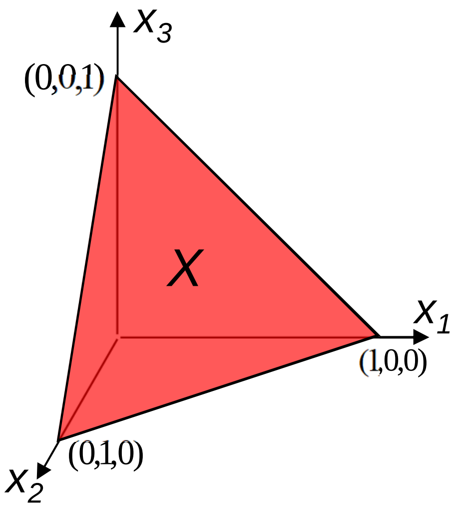


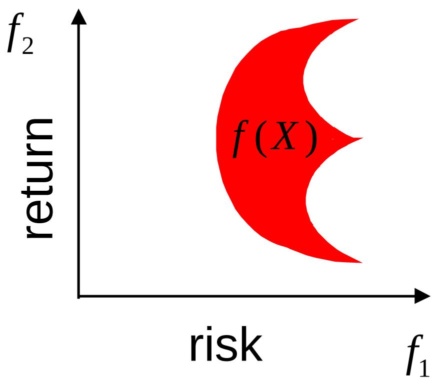
Definition. Objective vector $y \in f(X)$ is Pareto-optimal, if there exists a Pareto-optimal $x^* \in X$ s.t. $f(x^*)=y$

- Set of Pareto-optimal objective vectors: $f(X_{PO})$
- Notation $f(X_{PO}) = \text{v}-\max_{x \in X} f(x)$

Example: Markowitz model

- Optimal asset portfolio selection
  - How to allocate funds to m assets based on
    - Expected returns $\bar{r}_i$, i=1,...,m
    - Covariances of returns σ_{ij}, i,j=1,...,m
- Set of feasible solutions
  - Decision variables $x_1,\ldots,x_m$
    - Allocate x_j *100% of funds to j-th asset
  - Portfolio $x \in X = \{x \in \mathbb{R}^m | x_i \geq 0, \sum_{i=1}^m x_i = 1\}$
- Objective functions
  1. Maximize expected return of portfolio $f_2(x) = \sum_{i=1}^n \bar{r}_i x_i$
  2. Minimize variance (risk) of portfolio $f_1(x) = \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^m \sigma_{ij} x_i x_j$



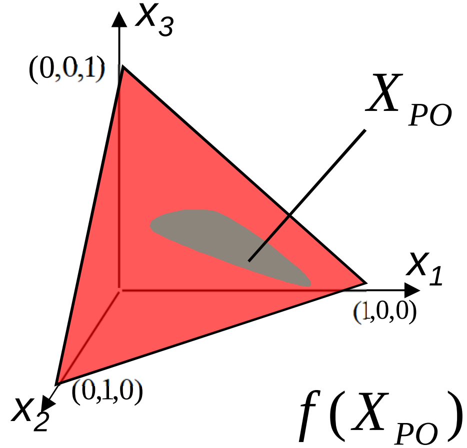


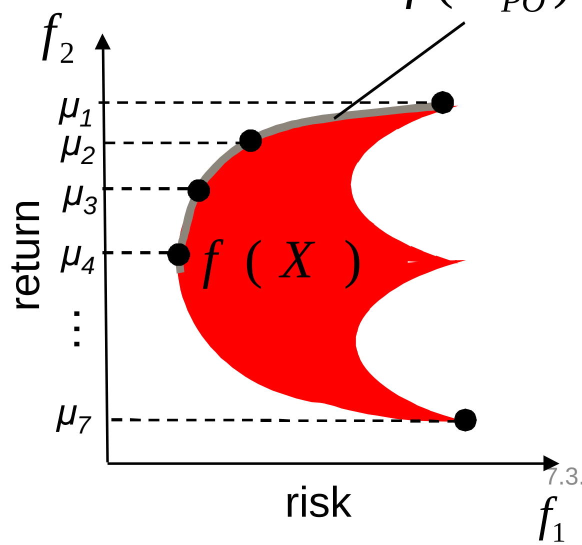
Pareto-optimality in Markowitz model

- Portfolio *x* is Pareto-optimal, if no other portfolio yields greater or equal expected return with less risk
- One possibility for computation:
  - Choose d = max number of solutions computed
  - Solve $\mu_1 = \max f_2$, $\mu_d = \min f_2$
  - For all k=2,...,d-1 set μ_k s.t. $\mu_{k-1} > \mu_k > \mu_d$ and solve (1-dimensional) quadratic programming problem

$$\min_{x \in X} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} \sigma_{ij} x_i x_j \text{ such that } \sum_{i=1}^{n} \bar{r}_i x_i = \mu_k$$

  - Discard solutions which are not PO
  - Not attractive when n>2





Algorithms for solving Pareto-optimal solutions (1/2)

❑ Exact algorithms

- Guaranteed to find all PO-solutions X_{PO}
- Only for certain problem types, e.g., Multi-Objective Mixed Integer Linear Programming (MOMILP)

❑ Use of single-objective optimization algorithms

- Sequentially solve ordinary (i.e. 1-dimensional) optimization problems to obtain a subset of all PO-solutions, X_{POS}
- Performance guarantee: $X_{POS} \subseteq X_{PO}$
  - o Solutions may not be "evenly" distributed in the sense that majority of the obtained solutions can be very "close" to each other
- Methods:
  - o Weighted sum approach, weighted max-norm approach, ε-constraint approach

Algorithms for solving Pareto-optimal solutions (2/2)

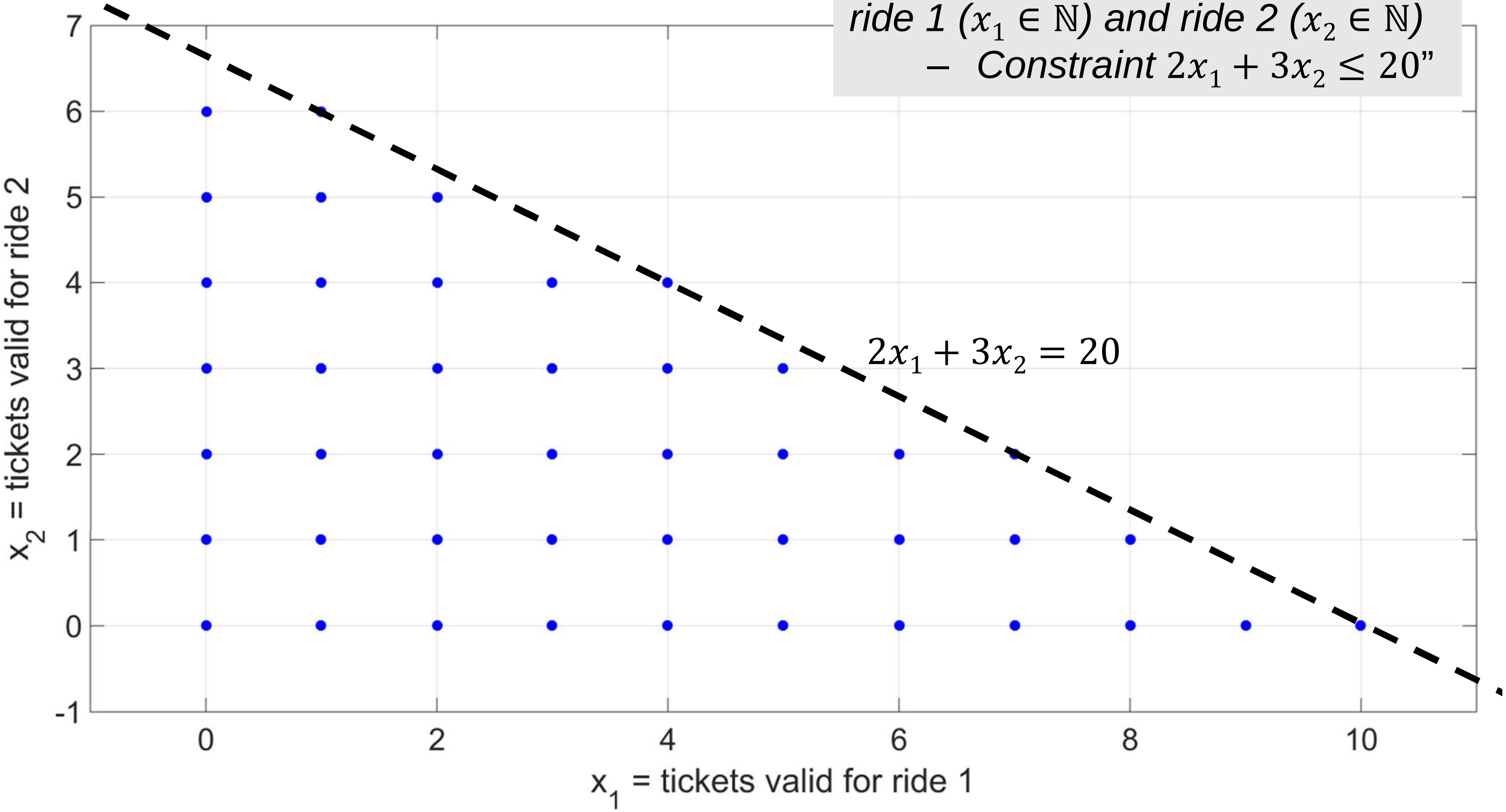
- Approximation algorithms
  - Obtain an approximation X_{POA} of X_{PO} in polynomial time
  - Performance guarantee: For every $x \in X_{PO}$ exists $y \in X_{POA}$ such that $||f(x)-f(y)||< \varepsilon$
  - Only for very few problem types, e.g., MO knapsack problems
- Metaheuristics
  - No performance guarantees
  - Can handle problems with
    - A large number of variables and constraints
    - Non-linear or non-continuous objective functions/constraints
  - Evolutionary algorithms (e.g., SPEA, NSGA)
  - Stochastic search algorithms (simulated annealing)

Example: Multiobjective integer linear programming (MOILP)

- Ben is at an amusement park that offers 2 different rides:
  - Tickets to ride 1 cost 2 €. Each ticket lets you take the ride twice
  - Tickets to ride 2 are for one ride and cost 3 €
- Ben has 20 euros to spend on tickets to ride 1 ($x_1 \in \mathbb{N}$) and ride 2 ($x_2 \in \mathbb{N}$) → constraint $2x_1 + 3x_2 \leq 20$
- Each time Ben takes ride 2, his grandfather cheers for him
- Ben maximizes the number of (i) rides taken and (ii) cheers → objective functions $f = (f_1, f_2) = (2x_1 + x_2, x_2)$

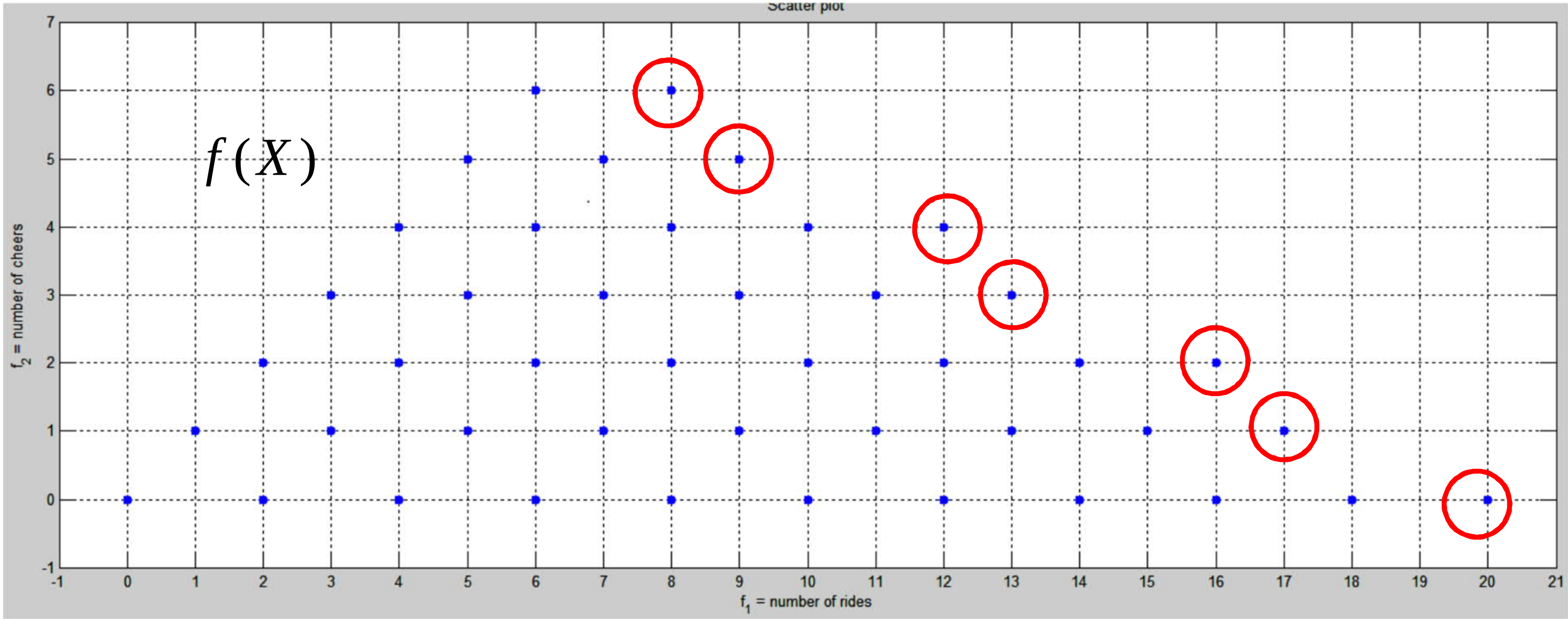
Feasible solutions X

"Ben has 20 euros. He is choosing the number of tickets to ride 1 ($x_1 \in \mathbb{N}$) and ride 2 ($x_2 \in \mathbb{N}$)
– Constraint $2x_1 + 3x_2 \leq 20$"

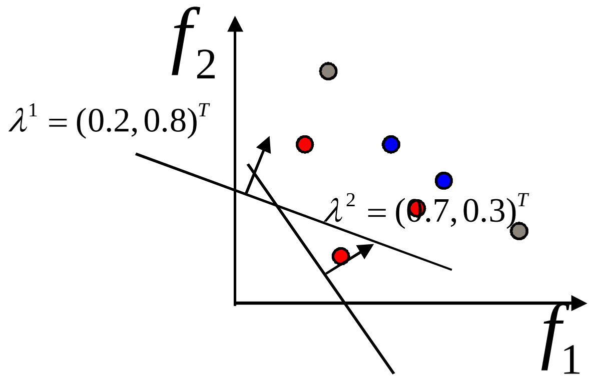


Example: MOILP (cont'd)

- Blue points are feasible solutions; the 7 PO solutions are circled



Weighted sum approach



❑ Algorithm

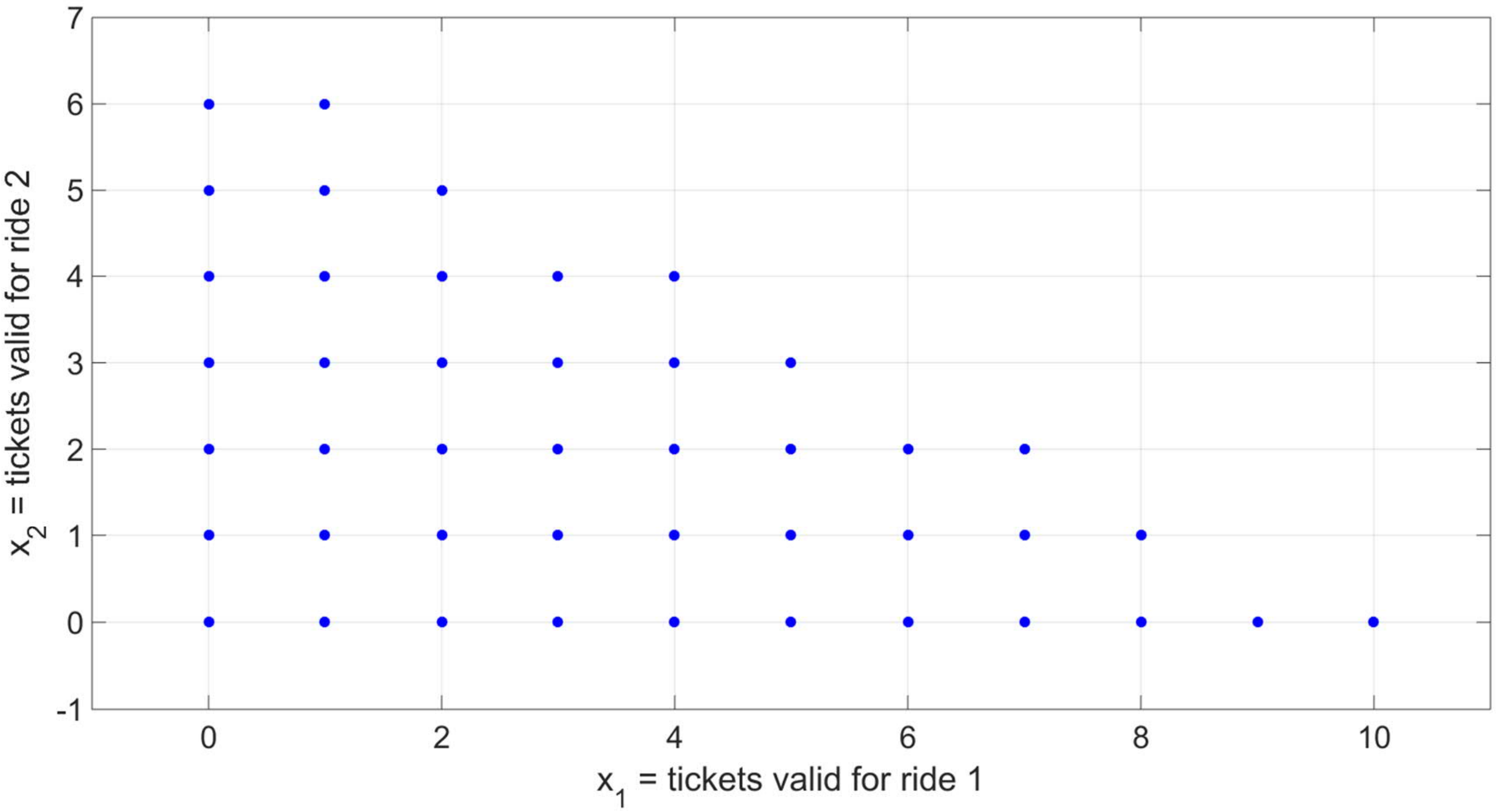
1. Generate $\lambda \sim UNI(\{\lambda \in [0,1]^n \mid \sum_{i=1}^{n} \lambda_i = 1\})$
2. Solve $\max_{x \in X} \sum_{i=1}^{n} \lambda_i f_i(x)$
3. Solution is Pareto-optimal

Repeat 1-3 until enough PO-solutions have been found

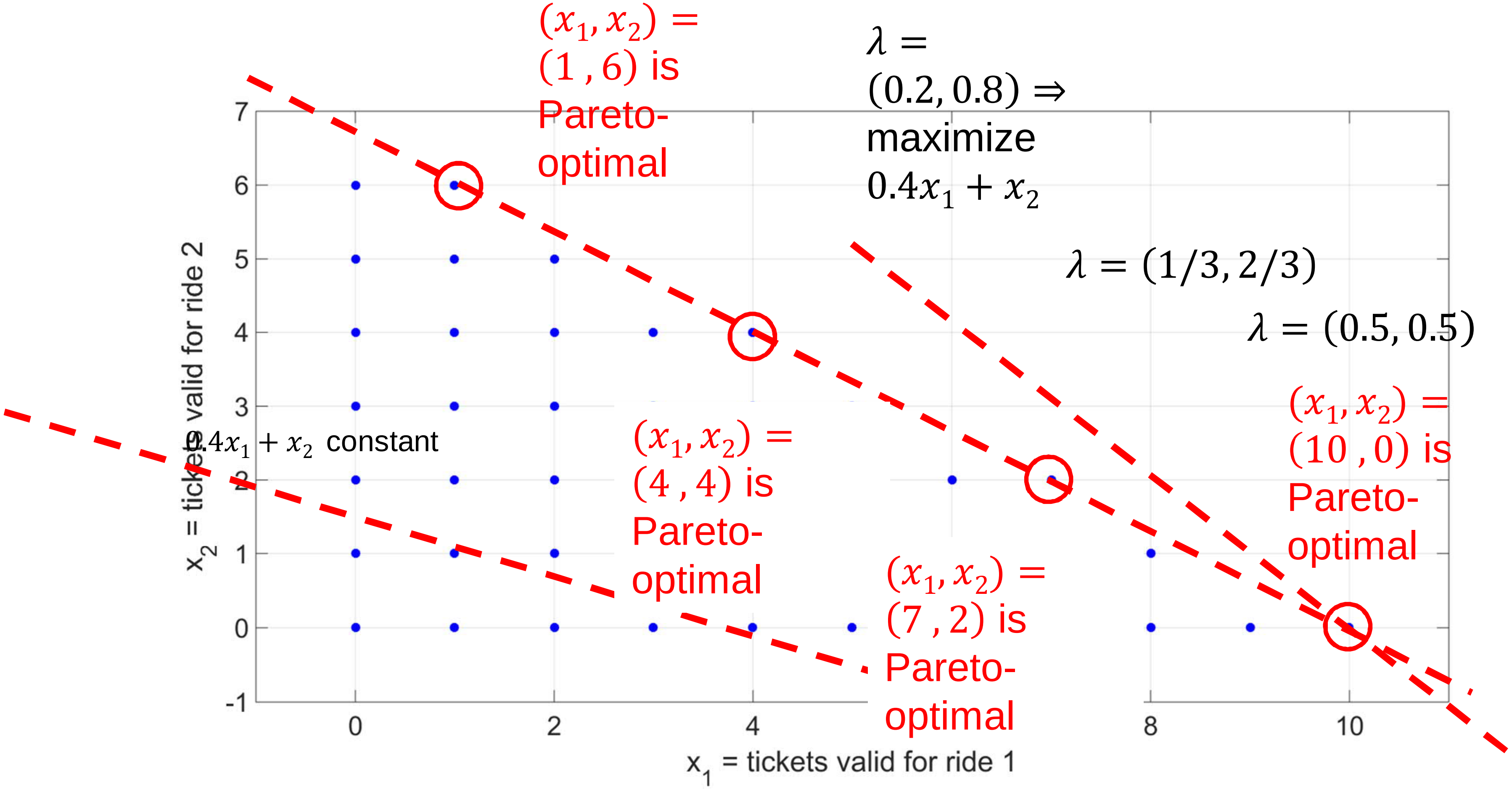
\+ Easy to implement

– Cannot find all PO solutions if the problem is non-convex (if PO solutions are not in the border of the convex hull of f(X))

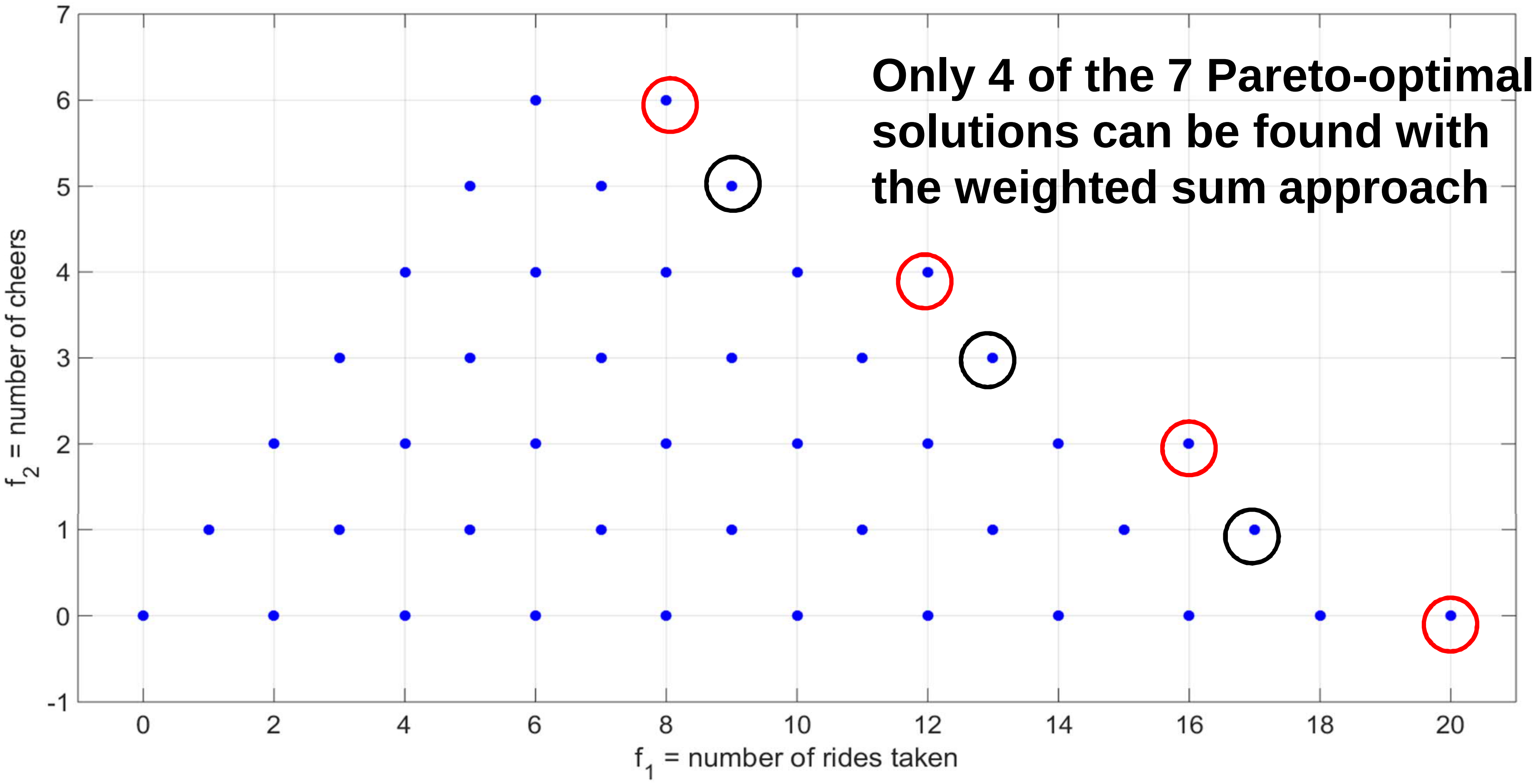
$$\max_{\substack{x_1, x_2 \in \mathbb{N} \\ 2x_1 + 3x_2 \leq 20}} [2\lambda_1 x_1 + (\lambda_1 + \lambda_2) x_2]$$



$$\max_{\substack{x_1,x_2\in\mathbb{N}\\ 2x_1+3x_2\le 20}} \; [2\lambda_1 x_1 + (\lambda_1+\lambda_2)x_2]$$



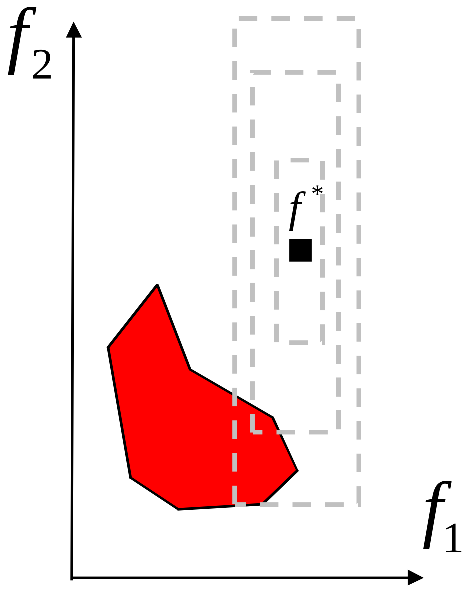
$f(X)$ and Pareto-optimal solutions



Weighted max-norm approach

- Idea: define a utopian vector of objective function values and find a solution for which the distance from this utopian vector is minimized
- Utopian vector: $f^* = [f_1^*, \dots, f_n^*], f_i^* > f_i(x)\ \forall x \in X, i = 1, \dots, n$
- Distance is measured with weighted max-norm $\max_{i=1,\dots,n} \lambda_i d_i$, where d_i is the between f_i^* and $f_i(x)$, and $\lambda_i > 0$ is the weight of objective *i* such that $\sum_{i=1}^n \lambda_i = 1$.
- The solutions that minimize the distance of $f(x)$ from f^* are found by solving:

$$\min_{x \in X} \|f^* - f(x)\|_{max}^{\lambda} = \min_{x \in X} \max_{i=1,\dots,n} \lambda_i \left(f_i^* - f_i(x)\right)$$
$$= \min_{x \in X, \Delta \in \mathbb{R}} \Delta \quad s.t.\ \lambda_i\left(f_i^* - f_i(x)\right) \le \Delta\ \ \forall i = 1, \dots, n$$

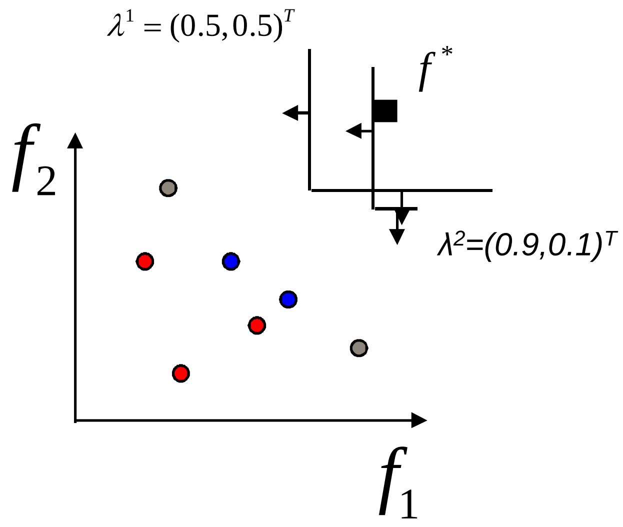


Contours of $\left\| f^* - f(x) \right\|_{\max}^{\lambda}$
when $\lambda = (0.9, 0.1)^T$

Weighted max-norm approach (2/2)

- Algorithm
  1. Generate $\lambda \sim UNI(\{\lambda \in [0,1]^n \mid \sum_{i=1}^{n} \lambda_i = 1\})$
  2. Solve $\min_{x \in X} \|f^* - f(x)\|_{max}^{\lambda}$
  3. At least one of the solutions of Step 2 is PO

  Repeat 1-3 until enough PO solutions have been found



\+ Easy to implement

\+ Can find all PO-solutions

– n additional constraints, one additional variable

Example: MOILP (cont'd)

- Find a utopian vector *f**
  - max $f_1 = 2x_1+x_2$ s.t. $2x_1+3x_2 \leq 20$, $x_1,x_2 \geq 0$
    - $x=(10,0)$; $f_1=20$
  - max $f_2 = x_2$ s.t. $2x_1+3x_2 \leq 20$, $x_1,x_2 \geq 0$
    - $x=(0, 20/3)$; $f_2=20/3$
  - Let $f^*=(21,7)$
- Minimize the distance from the utopian vector:

$$\min_{\Delta\in\mathbb{R}} \Delta \text{ s.t.}$$
$$\lambda_1\big(21-(2x_1+x_2)\big) \leq \Delta$$
$$\lambda_2(7-x_2) \leq \Delta$$
$$2x_1+3x_2 \leq 20, x_1, x_2 \in \mathbb{N}$$

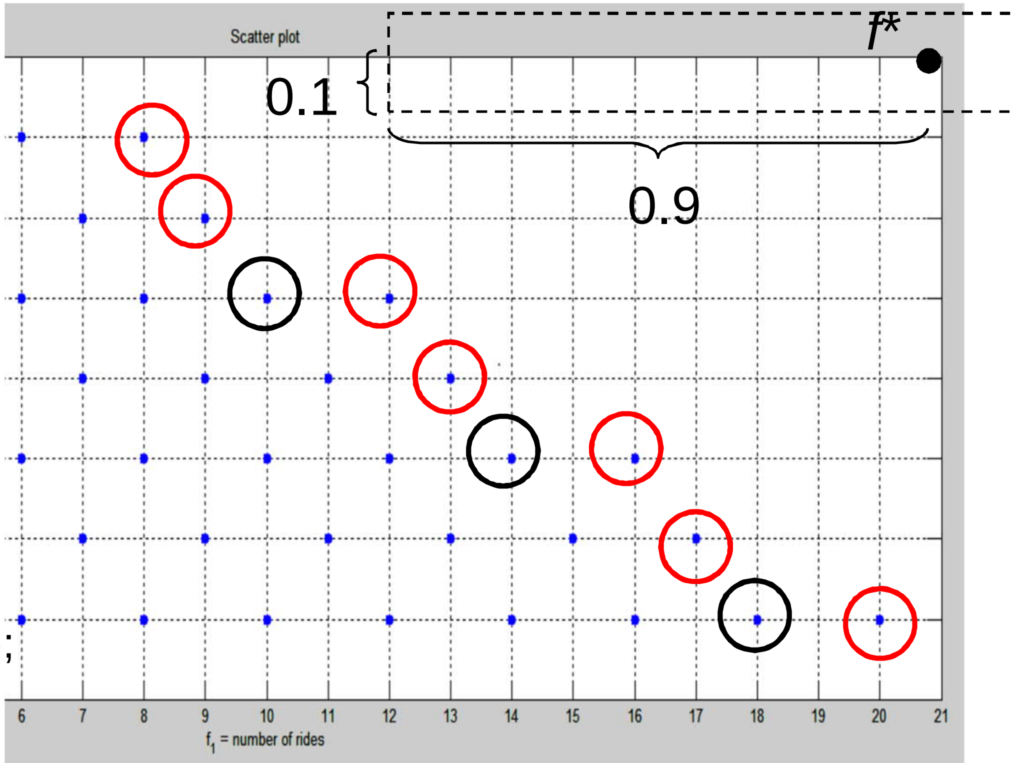
$\lambda_1=0.1$, $\lambda_2=0.9$:

$$\min_{\Delta\in\mathbb{R}} \Delta \text{ s.t.}$$
$$2.1-0.2x_1-0.1x_2 \leq \Delta$$
$$6.3-0.9x_2 \leq \Delta$$
$$2x_1+3x_2 \leq 20$$
$$x_1, x_2 \in \mathbb{N}$$

Solution: $\Delta=1.3$, $x=(1,6) \Rightarrow$
$x=(1,6)$, $f=(8,6)$ is PO

Example: MOILP revisited

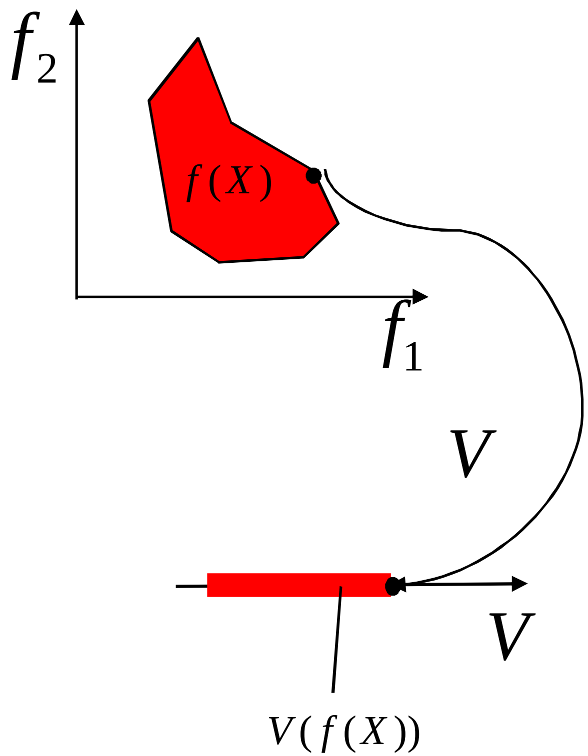
1. λ_1=0.1; solution: {Δ=1.3, *x*=(1,6)} ⇒ *x*=(1,6), *f*=(8,6) is PO
2. λ_1=0.2; 3 solutions *x*=(2,5), *x*=(3,4), *x*=(4,4). Only *x*=(2,5), *f*=(9,5) and *x*=(4,4), *f*=(12,4) are PO
3. λ_1=0.35; *x*=(5,3); *f*=(13,3) is PO
4. λ_1=0.4; 2 solutions *x*=(6,2) and *x*=(7,2); *x*=(7,2), *f*=(16,2) is PO
5. λ_1=0.55; *x*=(8,1); *f*=(17,1) is PO
6. λ_1=0.70; 2 solutions *x*=(9,0) and *x*=(10,0); *x*=(10,0), *f*=(20,0) is PO



Value function methods (1/2)

- Use value function $V: \mathbb{R}^n \to \mathbb{R}$ to transform the MOO problem into a single-objective problem
  - E.g., the additive value function $V\big(f(x)\big) = \sum_{i=1}^{n} w_i v_i(f_i(x))$

- **Theorem:** Feasible solution *x** with the highest value *V(x*)* is Pareto-optimal



Value function methods (2/2)

- Consider the additive value function $V\big(f(x)\big) = \sum_{i=1}^{n} w_i v_i(f_i(x))$ with incomplete weight information $w \in S \subseteq S^0$

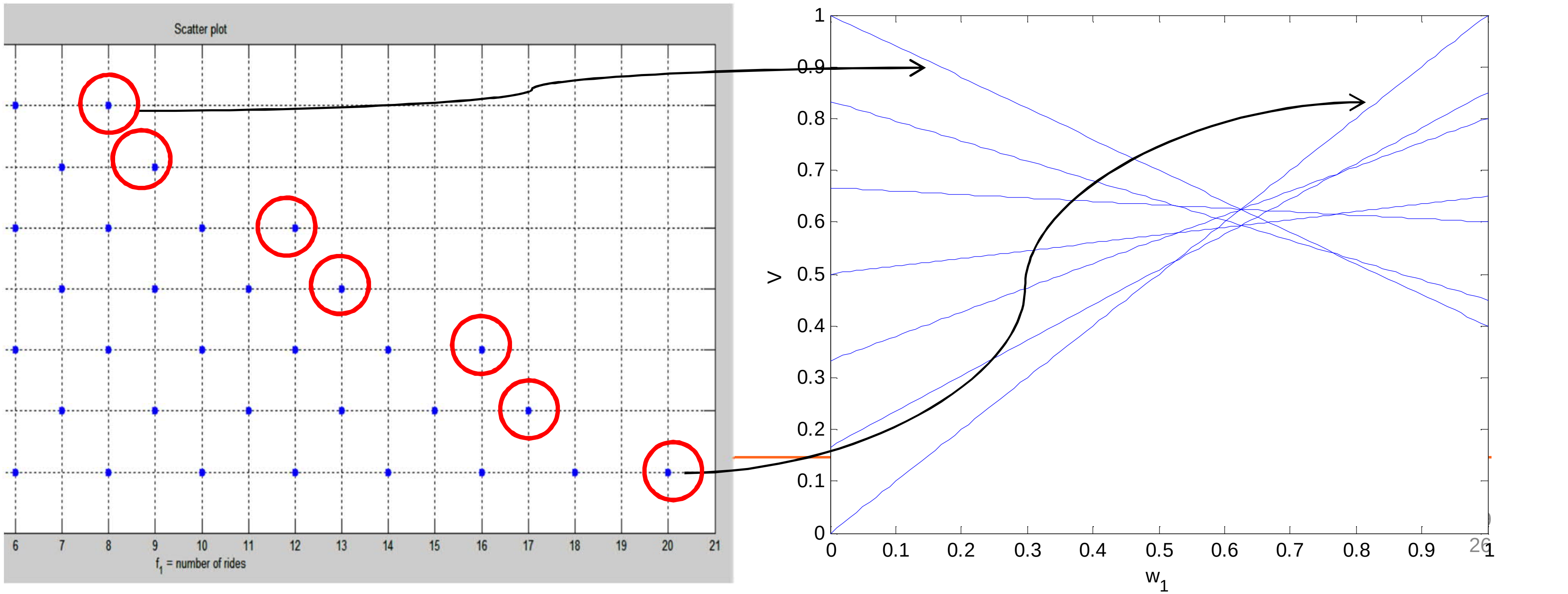
- Set of Pareto-optimal solutions X_{PO}= set of non-dominated solutions with no weight information $X_{ND}(S^0)$

- Preference statements on weights decrease the set of feasible weights to $S \subseteq S^0 \rightarrow$ focus on preferred PO-solutions $X_{ND}(S) \subseteq X_{ND}(S^0) = X_{PO}$

Example: MOILP revisited

- Choose $v_i(f_i(x))=f_i(x)/C_i^*$, normalization constants $C_1^*=20$, $C_2^*=6$

$$V(f(x),w)=\sum_{i=1}^{n} w_i v_i(f(x)) = w_1 v_1\big(f_1(x)\big)+(1-w_1)v_2\big(f_2(x)\big)=\frac{w_1(2x_1+x_2)}{20}+(1-w_1)(x_2/6)$$



Example: Bridge repair program (1/7)

- Total of 313 bridges calling for repair
- Which bridges should be included in the repair program under the next three years?
- Budget of 9,000,000€
- Program can contain *maximum* of 90 bridges
  - Proxy for limited availability of equipment and personnel etc.
- Program must repair the total sum of damages by *at least* 15,000 units

P. Mild, J. Liesiö and A. Salo (2015): Selecting Infrastructure Maintenance Projects with Robust Portfolio Modeling, *Decision Support Systems*

Example: Bridge repair program (2/7)

- Set of feasible solutions *X* defined by linear constraints and binary decision variables:

$$X = \{x \in \{0,1\}^{313} | g(x) \leq 0\}, \quad g(x) = \begin{bmatrix} \sum_{j=1}^{313} c_j x_j - 9000000 \\ \sum_{j=1}^{313} x_j - 90 \\ 15000 - \sum_{j=1}^{313} d_j x_j \end{bmatrix}$$

- x_j = a decision variable: $x_j = 1$ repair bridge j
- $x=[x_1,\ldots,x_{313}]$ is a repair program
- c_j = repair cost of bridge j
- d_j = sum of damages of bridge j

Example: Bridge repair program (3/7)

- Six objective indexes measuring urgency for repair
  1. Sum of Damages ("SumDam")
  2. Repair Index ("RepInd")
  3. Functional Deficiencies ("FunDef")
  4. Average Daily Traffic ("ADTraf")
  5. Road Salt usage ("RSalt")
  6. Outward Appearance ("OutwApp")
- All objectives additive over bridges: $f_i(x) = \sum_{j=1}^{313} v_i^j x_j$,

  where v_i^j is the score of bridge *j* with regard to objective *i:*

Example: Bridge repair program (4/7)

- A multi-objective zero-one linear programming (MOZOLP) problem

$$\text{v}-\max_{x\in X}\left(\sum_{j=1}^{313} v_1^j x_j, \ldots, \sum_{j=1}^{313} v_6^j x_j\right)$$

- Pareto-optimal repair programs X_{PO} generated using the weighted max-norm approach

$$\min_{x\in X, \Delta\in\mathbb{R}} \Delta$$

$$\Delta \geq \lambda_i\left(f_i^* - \sum_{j=1}^{313} x_j v_i^j\right) \; \forall i = 1, \ldots, 6$$

Aalto University
School of Science

Example: Bridge repair program (5/7)

- Additive value function applied for modeling preferences between the objectives: $V(x,w) = \sum_{i=1}^{6} w_i f_i(x) = \sum_{i=1}^{6} w_i \sum_{j=1}^{313} v_i^j x_j$
- Incomplete ordinal information about objective weights: {SumDam,RepInd} ≥{FunDef, ADTraf} ≥ {RSalt,OutwApp}

$$S = \left\{w \in S^0 \middle| w_i \geq w_j \geq w_k, \forall i = 1,2; j = 3,4; k = 5,6\right\}$$

- Non-dominated repair programs

$$X_{ND}(S) = \left\{x \in X | \nexists x' \in X \text{ s.t. } \begin{cases} V(x',w) \geq V(x,w) \text{ for all } w \in S \\ V(x',w) > V(x,w) \text{ for some } w \in S \end{cases}\right\}$$

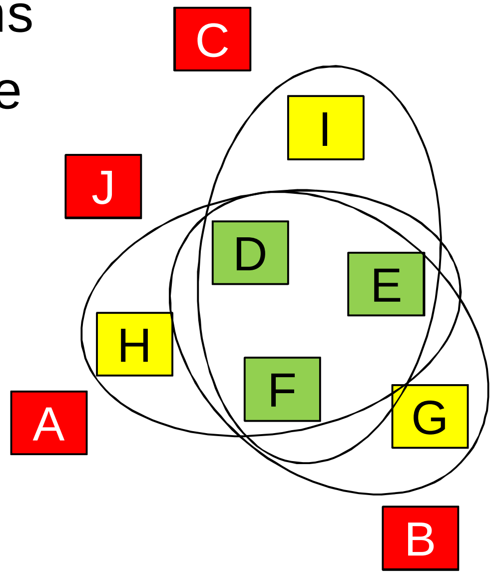
$$X_{PO} = X_{ND}(S^0) \supseteq X_{ND}(S)$$

Example: Bridge repair program (6/7)

- Ca. 10,000 non-dominated bridge repair programs
- Bridge-specific decision recommendations can be obtained through a concept of *core index*:

$$CI_j = \frac{|\{x \in X_{ND}(S) | x_j = 1\}|}{|X_{ND}(S)|}$$

- Of the 313 bridges:
  - 39 were included in all non-dominated repair programs (CI=1)
  - 112 were included in some but not all non-dominated programs (0<CI<1)
  - 162 were included in none of the non-dominated programs (CI=0)



Example: Bridge repair program (7/7)

- Bridges listed in decreasing order of core indices
  - Tentative but not binding priority list
  - Costs and other characteristics displayed

- The list was found useful by the program managers

		BRIDEGES' SCORES						
Bridge number and name	Core Index	DamSum	RepInd	FunDef	ADTraf	Rsalt	OutwApp	Cost
2109 Lavusjoen silta	1.00	5.00	1.65	4	2.6	1	2.6	50000
2218 Joroisvirran silta	1.00	5.00	5.00	2	5	5	2.6	180000
2217 Rautatieylikulkusilta	1.00	3.49	5.00	1.5	5	5	1.8	130000
763 Hurukselantien risteyssilta	1.00	2.27	2.33	1	3.4	5	1	280000
80 Suolammenojan silta	1.00	1.36	1.53	2	4.2	5	1.8	10000
257 Villikkalan silta	0.81	1.97	1.96	5	1	1	1.8	20000
1743 Huuman silta II	0.76	1.64	1.53	1	5	5	1.8	140000
730 Mälkiän itäinen risteyssilta	0.63	1.33	1.58	1.5	5	5	1	120000
2804 Raikuun kanavan silta	0.60	3.93	1.12	2.5	1	1	1	20000
856 Ojaraitin alikulkukäytävä I	0.54	1.46	1.46	1	5	5	1	20000
2703 Grahnin alikulkukäytävä	0.43	1.70	1.23	1	5	5	1	60000
817 Petäjäsuon risteyssilta	0.39	1.52	1.37	1	5	5	1	50000
725 Mustolan silta	0.29	1.98	1.93	2	1.8	1	4.2	190000
2189 Reitunjoen silta	0.24	1.90	1.63	3	1.8	1	1.8	10000
2606 Haukivuoren pohjoinen ylikulkusilta	0.15	1.84	2.09	1.5	2.6	1	1	70000
125 Telataipaleen silta	0.14	1.38	1.12	1	5	5	1.8	40000
608 Jalkosalmen silta	0.03	1.54	1.50	3	1.8	1	2.6	10000
556 Luotolan silta	0.00	1.74	1.26	3	1	1	1.8	10000
661 Raikan silta	0.00	1.95	1.58	2	1	1	1.8	10000
2613 Pitkänpohjanlahden silta	0.00	1.27	1.16	1	4.2	5	2.6	20000
738 Hyypiälän ylikulkusilta	0.00	1.72	1.79	1	3.4	1	1.8	90000
2549 Uitonsalmen silta	0.00	1.71	1.37	3	1	1	1	30000
703 Tokkolan silta	0.00	1.82	1.70	2	1.8	1	1	10000
870 Tiviän alikulkukäytävä	0.00	1.10	1.07	1	5	5	1	20000
377 Sudensalmen silta	0.00	1.88	1.66	1	2.6	1	1.8	20000
953 Sydänkylän silta	0.00	1.23	1.33	3.5	1	1	1.8	10000
700 Kirjavalan ylikulkusilta	0.00	1.42	1.98	1.5	1	1	1	60000
2142 Latikkojoen silta	0.00	1.43	1.58	2.5	2.6	1	1.8	20000
464 Jokisilta	0.00	1.19	1.25	3.5	1.8	1	1	20000
1025 Hartunsalmen silta	0.00	1.18	1.09	3.5	1.8	1	2.6	20000
95 Touksuon silta	0.00	1.83	1.18	2	2.6	1	2.6	20000
418 Laukassalmen silta	0.00	1.54	1.35	1.5	2.6	1	1.8	10000
420 Sillanmäenojan silta	0.00	1.20	1.07	1.5	2.6	1	1.8	10000

Summary

- MOO differs from MAVT in that
  - Alternatives are not explicit but defined implicitly through constraints
  - MOO problems are computationally much harder
- MOO problems are solved by
  - Computing the set of all Pareto-optimal solutions – or at least a subset or an approximation
  - Introducing preference information about trade-offs between objectives to support the selection of one of the PO-solutions