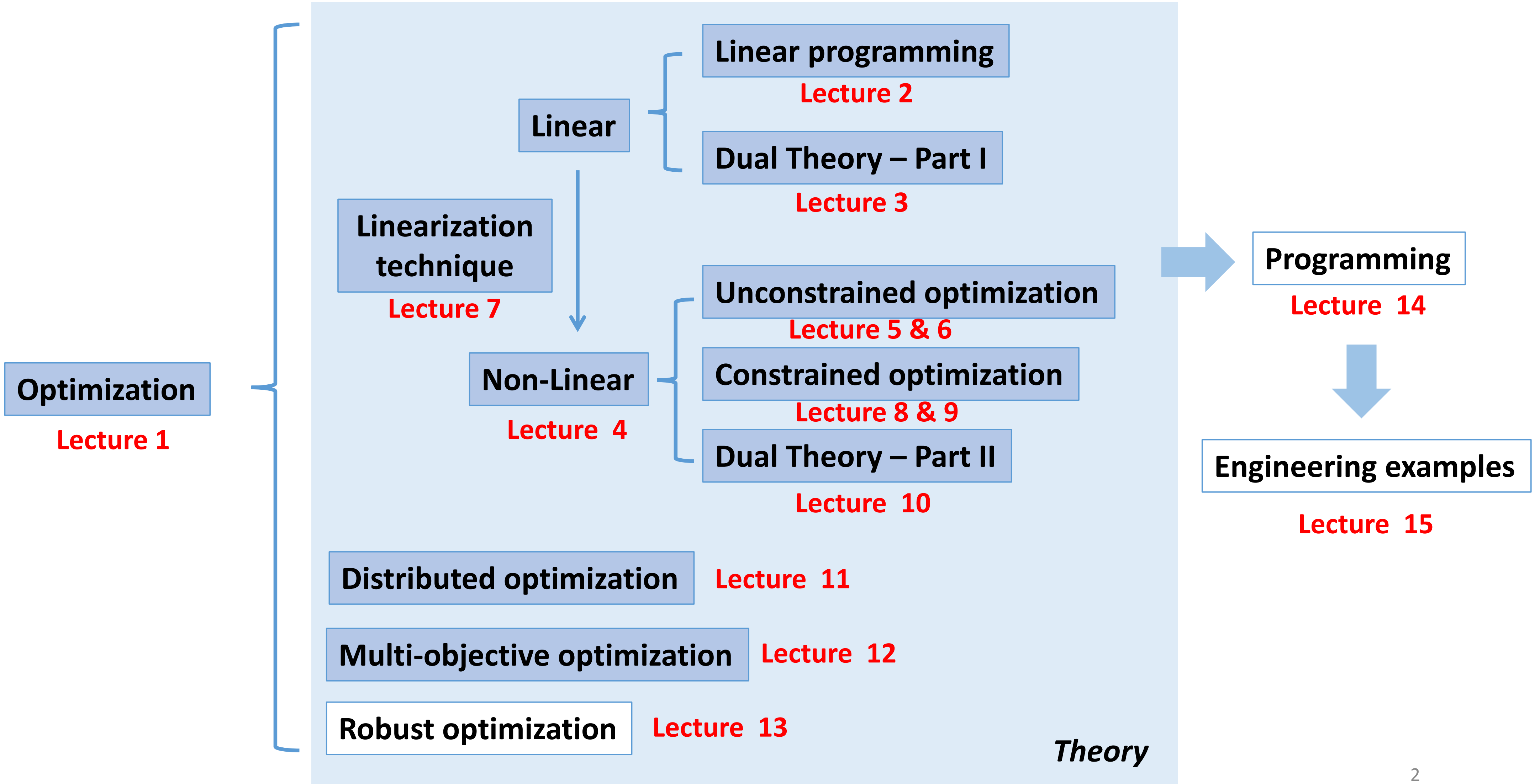
MAEG4070 Engineering Optimization

Lecture 12 Multi-objective Optimization

Yue Chen
MAE, CUHK
email: yuechen@mae.cuhk.edu.hk
Nov 16, 2022

Content of this course (tentative)



Overview

In previous lectures, the optimization problems aim to minimize or maximize a *single objective*. In practice, sometimes we care about *more than one objectives*.

These objectives are usually **competing**. Therefore, multi-objective analysis is used to reveal the **tradeoff** among different objectives.

Multi-objective optimization (MOO) try to find the set of solutions that define the **best tradeoff** between competing objectives

Example

Production Planning:
- max {total net revenue}
- min {overtime}
- min {finished goods inventory}
- …

Energy system operation:
- min {total operation cost}
- min {total pollution}
- max {system reliability}
- …

Aircraft Design:
- max {passenger volume}
- min {fuel consumption}
- min {lifecycle cost}
- …

History

F. Y. Edgeworth:

- King’s College (London) and later Oxford
- is the first to define an optimum for multicriteria economic decision-making
- application in the multiutility problem of two consumers.

Vilfredo Pareto:

- While working in Florence as a civil engineer from 1870-1893, Pareto takes up the study of philosophy and politics and is one of the first to analyze economic problems with mathematical tools
- In 1893, Pareto became the Chair of Political Economy at the University of Lausanne in Switzerland, where he created his two famous theory: circulation of the elites, **the Pareto Optimum**.

History

Wolfram Stadler:

- Began to apply the notion of Pareto Optimality to the fields of engineering and science in middle 1970's.

- Applications of multi-objective optimization in engineering design grew over the following decades.

Reference:

Marler R T, Arora J S. Survey of multi-objective optimization methods for engineering[J]. Structural and multidisciplinary optimization, 2004, 26(6): 369-395.

Table 1 Summary of primary methods

		Survey Section	Scalar Method	Possible Pareto Opt.	Necessary for Pareto Opt.	Sufficient for Pareto Opt.	Utopia Point	PC (0 to 5)	SUC (0 to 5)	CC (0 to 5)	PAP (0 to 5)
A Priori Articulation of Preferences	Wtd. Global Critn. I	3.1	x		F.W.	x	x	0	1	1	2
	Wtd. Global Critn. II	3.1	x		x*	x	x	0	1	1	2
	Weighted Sum	3.2	x			x		0	1	0	1
	Lexicographic	3.3				x		2	1	2	1
	Weighted Min-Max	3.4			x	x-weak	x	1	1	2	1
	Exponential Weighted	3.5	x		x	x		0	1	0	1
	Weighted Product	3.6	x		F.W.	x		0	1	1	1
	Goal Programming	3.7		x				1	1	2	1
	Bounded Obj. Fnc.	3.8		x				1	1	1	1
	Physical Programming	3.9			x	x	x	3	3	1	4
A Posteriori Articulation	Physical Programming	4.1			x	x	x	4	1	–	5*
	NBI	4.2				x	x	2	1	–	5*
	Normal Constraint	4.3			x	x	x	2	1	–	5*
	Genetic	6.2		x				5	1	–	5*
No Articulation of Preferences	Global Criterion	5.1									
	Achievement Fnc.	5.1	x		x	x		0	1	0	0
	Compromise Fnc.	5.1	x			x	x	0	1	1	0
	Objective Sum	5.1	x			x		0	0	0	0
	Min-Max	5.1		x		x-weak	x	1	0	2	0
	Nash Arbitration	5.2	x		F.W.	x		0	1	1	0
	Objective Product	5.2	x			x	x	0	0	1	0

Basic model

Single-objective optimization

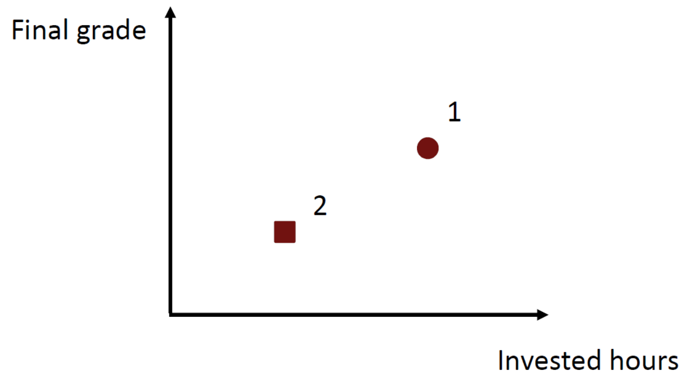
$$\min_x \; f(x)$$
$$\text{s.t. } g_i(x) \leq 0, \forall i = 1, ..., I$$
$$h_j(x) = 0, \forall j = 1, ..., J$$

where

- x is the decision variables
- $g_i(x), \forall i$ are inequality constraints
- $h_j(x), \forall j$ are equality constraints

Multi-objective optimization

$$\min_x \; f_1(x)$$
$$\max_x \; \vdots$$
$$\min_x \; f_n(x)$$
$$\text{s.t. } g_i(x) \leq 0, \forall i = 1, ..., I$$
$$h_j(x) = 0, \forall j = 1, ..., J$$



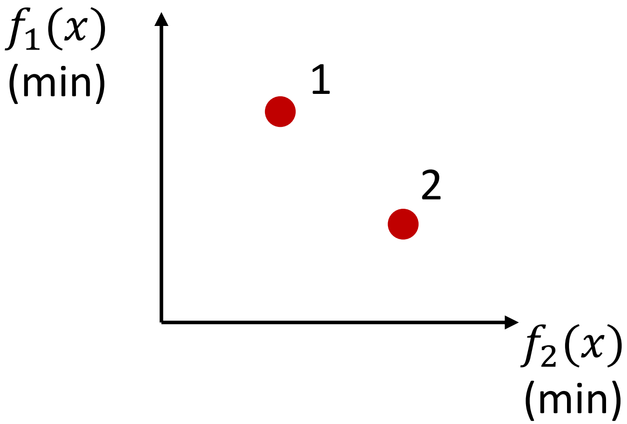
Dominance

In *single-objective optimization*, we can easily determine whether a solution is better than the other by comparing their *objective function* values. But how can we do that in multi-objective optimization?

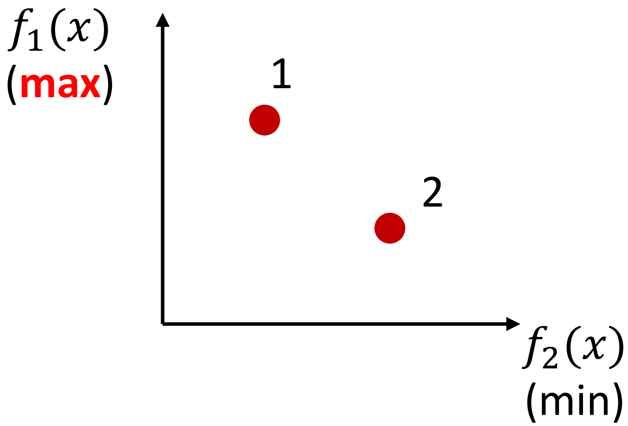
Dominance

- Solution x_1 **dominates** x_2 if:
  - ✓ Solution x_1 is **no worse** than x_2 in **all objectives**
  - ✓ Solution x_1 is **strictly better** than x_2 in **at least one objective**
- If x_1 **dominates** x_2, then x_2 **is dominated by** x_1
- If x_1 does not dominate x_2 and x_2 does not dominate x_1, then we say x_1 and x_2 are **non-dominated** solutions

Dominance

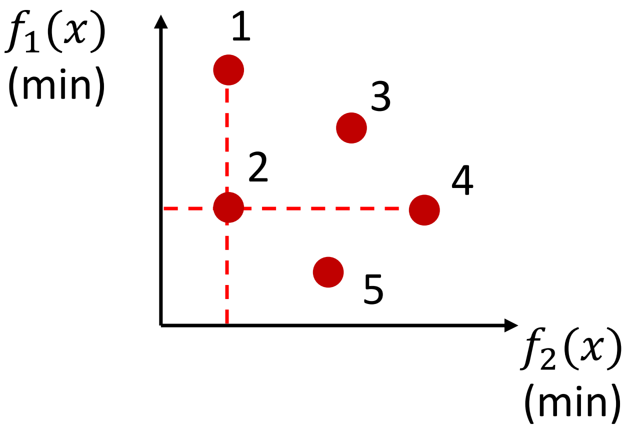


Non-dominated solutions



2 is dominated by 1; 1 dominates 2

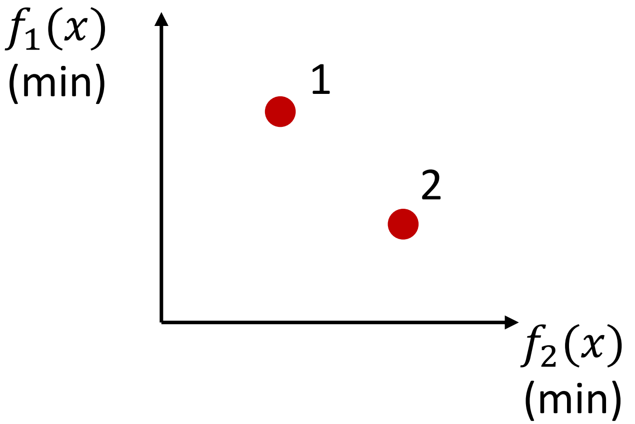
Try it yourself:



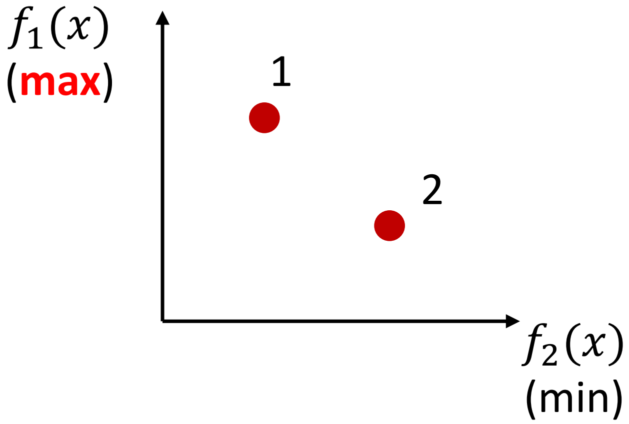
Try yourself:

- 1 & 2
- 1 & 3
- 2 & 3
- 2 & 4
- 3 & 4

Dominance

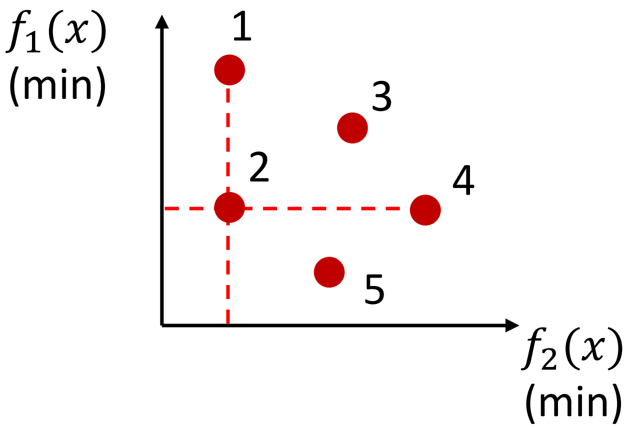


Non-dominated solutions



2 is dominated by 1; 1 dominates 2

Try it yourself:



Answer:

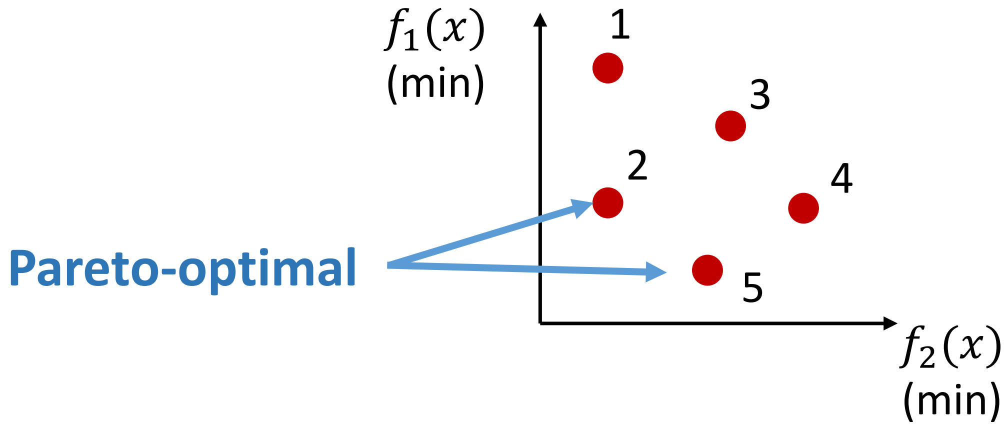
- 1 & 2: 2 dominates 1
- 1 & 3: non-dominated solutions
- 2 & 3: 2 dominates 3
- 2 & 4: 2 dominates 4
- 3 & 4: non-dominated solutions

Pareto optimal

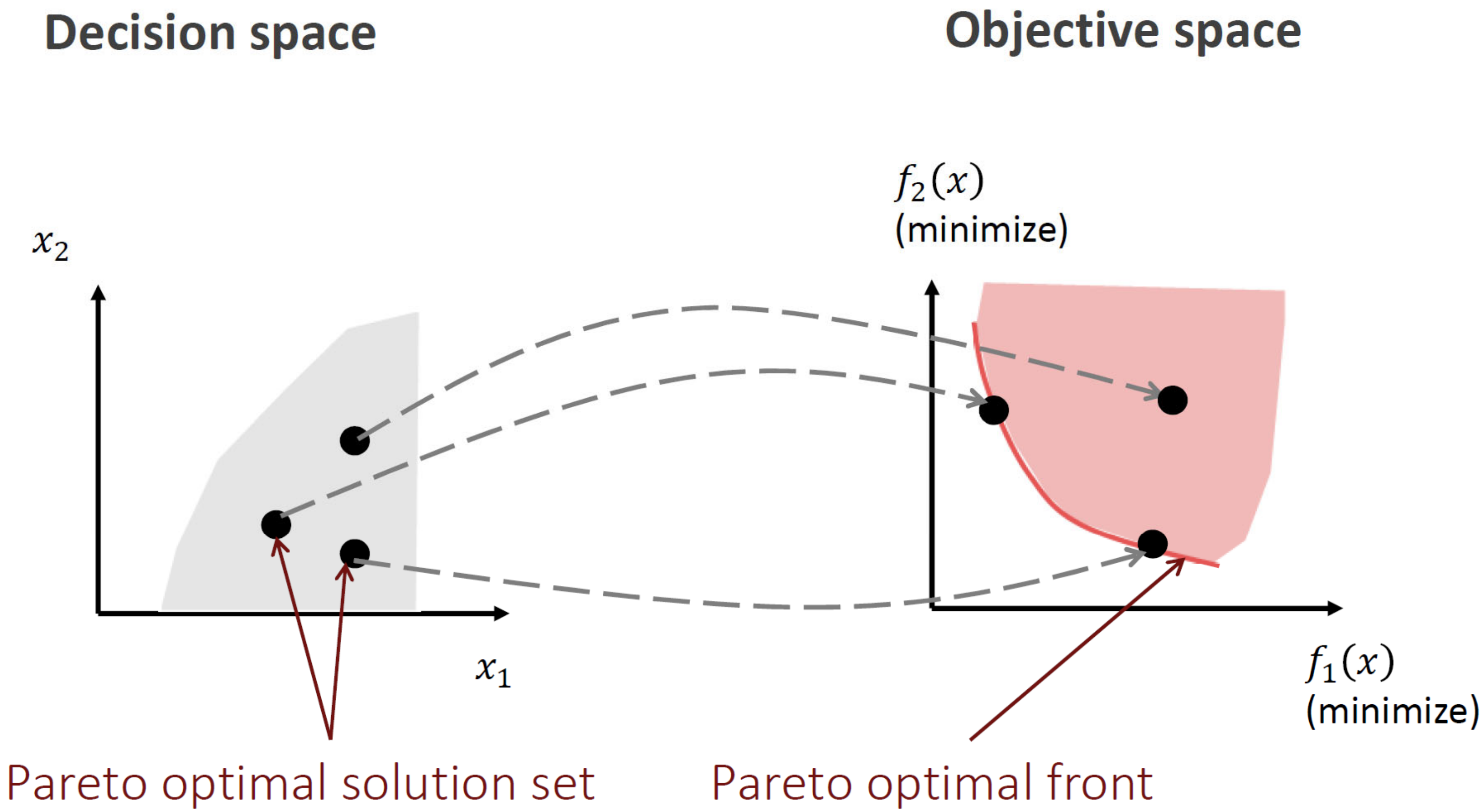
Given a set of solutions, the **non-dominated solution set** is a set of all the solutions that are **not dominated** by any member of the solution set.

The non-dominated set of the entire feasible decision space is called the **Pareto-optimal set**.

The boundary defined by the set of all points mapped from the Pareto optimal set is called the **Pareto-optimal front**.

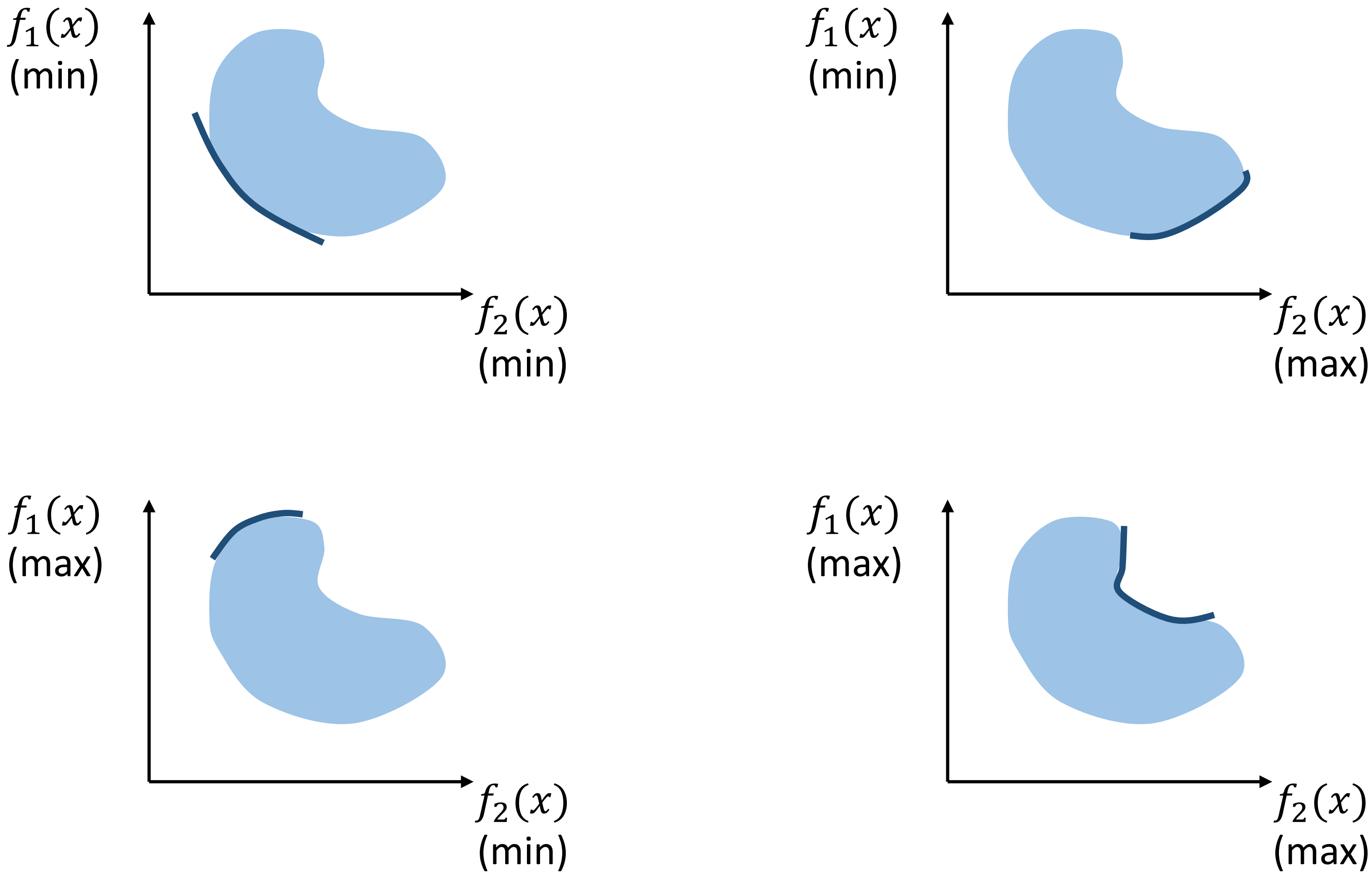


Pareto optimal



Pictures from Google image

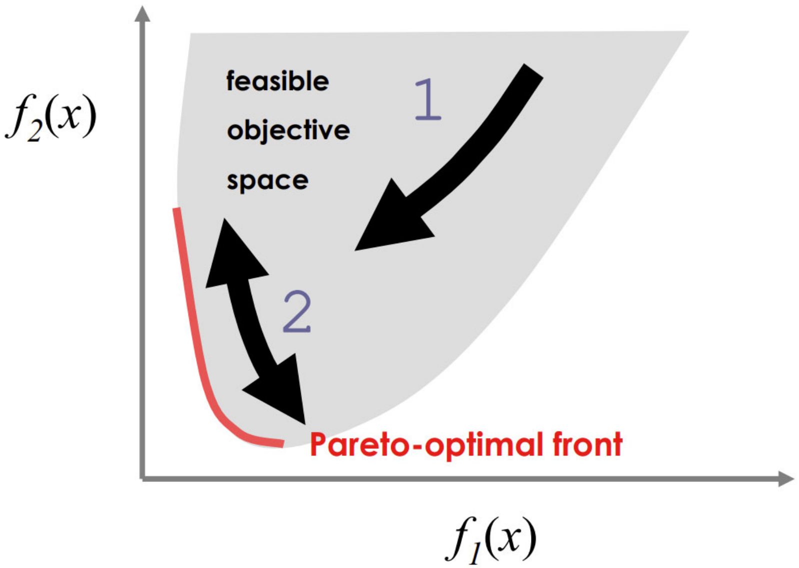
Pareto optimal



Goal of multi-objective optimization

Two goals:

- Find set of solutions as close as possible to the Pareto-optimal front
- Find a set of solutions as diverse as possible



Pictures from Google image

Solution algorithms

Multi-objective optimization aims to find a set of solutions whose objective values are close to the Pareto-optimal front, and these solutions can be as diverse as possible.

Classic solution algorithms include:

- **Weighted sum method**
- **ε-Constraint method**
- Nash bargaining
- Benson's method
- …

Weighted sum method

Basic idea: Turn a set of objectives into a single objective by adding each objective times a user-specified weight w_n.

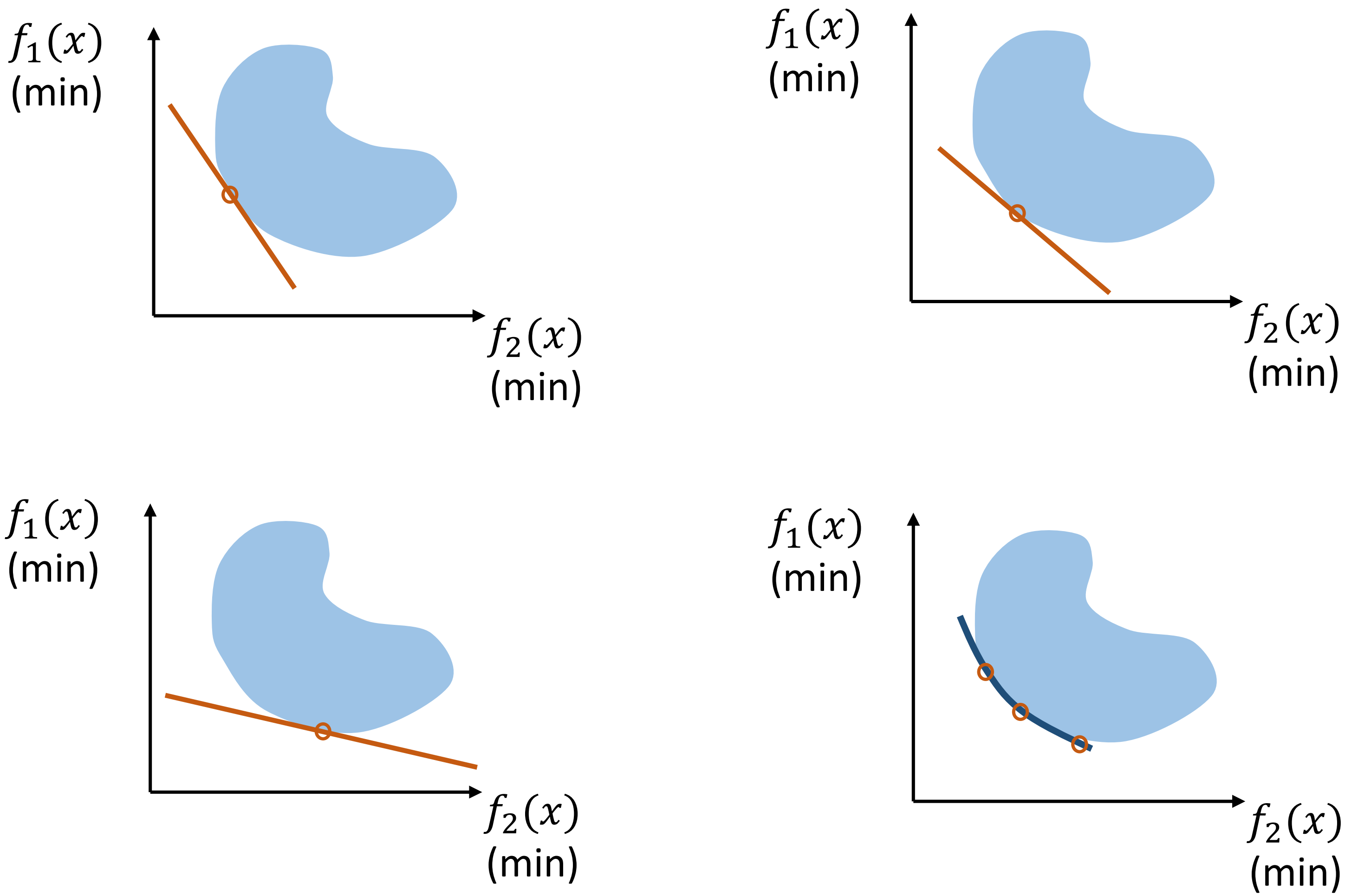
- **Step 1**: turn all objectives into "minimization"
- **Step 2**: Normalize the objectives (Nonlinear problems are sensitive to scale)
- **Step 3**: calculate the weighted sum

$$\min_x w_1 f_1(x) + \cdots + w_n f_n(x)$$

These weights can be chosen according to the relative importance of the objectives.

$$\begin{aligned} &\min_x \quad f_1(x) \\ &\max_x \quad \vdots \\ &\min_x \quad f_n(x) \\ &\text{s.t.} \quad g_i(x) \le 0, \forall i = 1, ..., I \\ &\qquad h_j(x) = 0, \forall j = 1, ..., J \end{aligned}$$

Weighted sum method

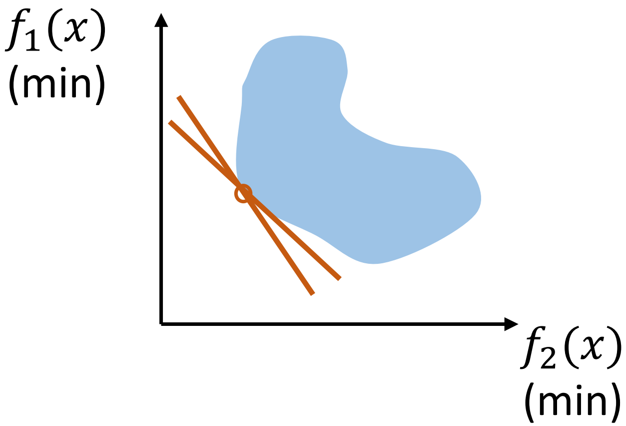


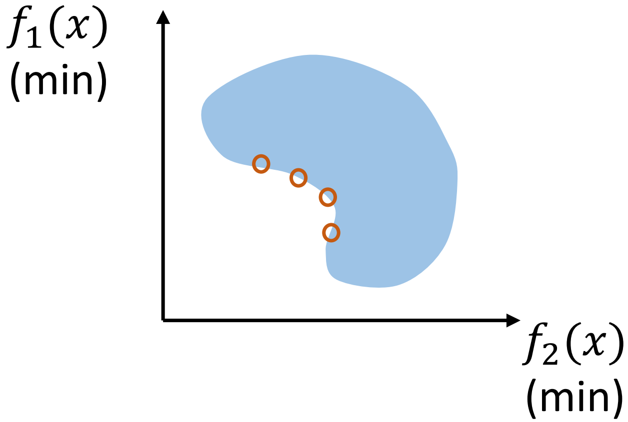
Change w_1, w_2 we have different solutions, can we get the whole Pareto front?

Weighted sum method

Two undesired cases:

- Two different set of weights does not necessarily lead to two different Pareto-optimal solutions.





- Cannot find certain Pareto-optimal solution if the objective space is nonconvex.

Weighted sum method

Advantages:

- Easy to implement, straightforward
- For convex problems, it guarantees to find solutions on the entire Pareto-optimal set

Disadvantages:

- Uniformly distributed set of weights does not guarantee a uniformly distributed set of Pareto-optimal solution
- Two different set of weights does not necessarily lead to two different Pareto-optimal solutions
- Cannot find certain Pareto-optimal solution if the objective space is nonconvex

ε-Constrained Method

Basic idea: Keep just one of the objectives, and treat the rest as constraints (set expectation for other objectives)

$$
\begin{aligned}
\min_x \quad & f_l(x) \\
\text{s.t.} \quad & f_k(x) \le \epsilon_k, \forall k = 1, ..., n, k \ne l \\
& g_i(x) \le 0, \forall i = 1, ..., I \\
& h_j(x) = 0, \forall j = 1, ..., J
\end{aligned}
$$

$$
\begin{aligned}
\min_x \quad & f_1(x) \\
\max_x \quad & \vdots \\
\min_x \quad & f_n(x) \\
\text{s.t.} \quad & g_i(x) \le 0, \forall i = 1, ..., I \\
& h_j(x) = 0, \forall j = 1, ..., J
\end{aligned}
$$

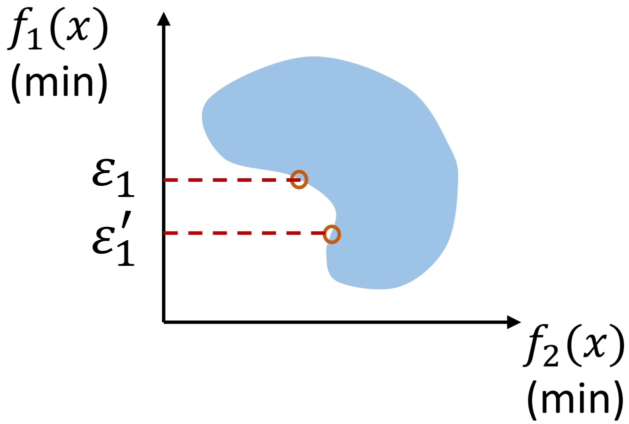
If we aims to maximize the objective, then the related constraint can be replaced by

$$f_k(x) \ge \varepsilon_k$$

ε-Constrained Method

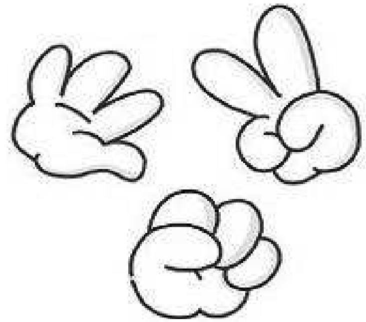
Advantages:
- Different Pareto-optimal solutions can be found using different ε values
- Appliable to either convex or nonconvex problems

Disadvantages:
- The value of ε should be carefully chosen so that it is within the minimum and maximum values of the individual objective function; otherwise, the problem may become infeasible.



What is a game?

n-person ***Normal Form*** game $< \mathcal{N}, \mathcal{A}, u >$



- **Players**: who makes the decisions?
  - $\mathcal{N} = \{1, \dots, n\}$ is a finite set, indexed by i

- **Action sets**: what can agents do?
  - a_i is the action of i, $\mathcal{A}_i$ is the action set, $a_i \in \mathcal{A}_i$
  - $a = (a_1, \dots, a_n) \in \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ action profile

- **Utility function**: How do agents value the outcome?
  - For agent $i, u_i: \mathcal{A} \to \mathbb{R}$
  - $u = (u_1, \dots, u_n)$ the profile of utility functions

Extensive Form: include timing of moves (outside the scope of this course)

Pictures from Google image

Strategic Reasoning

If you knew what other players was going to do, it would be easy to pick your own action. Let $a_{-i} =< a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n >$ and $a = (a_{-i}, a_i)$

Definition (Best response)
$a_i^* \in BR(a_{-i})$ iff $\forall a_i \in A_i, u_i(a_i^*, a_{-i}) \geq u_i(a_i, a_{-i})$

Definition (Nash equilibrium)
$a =< a_1, \dots, a_n >$ is a ("pure strategy") Nash equilibrium iff $\forall i, a_i \in BR(a_{-i})$

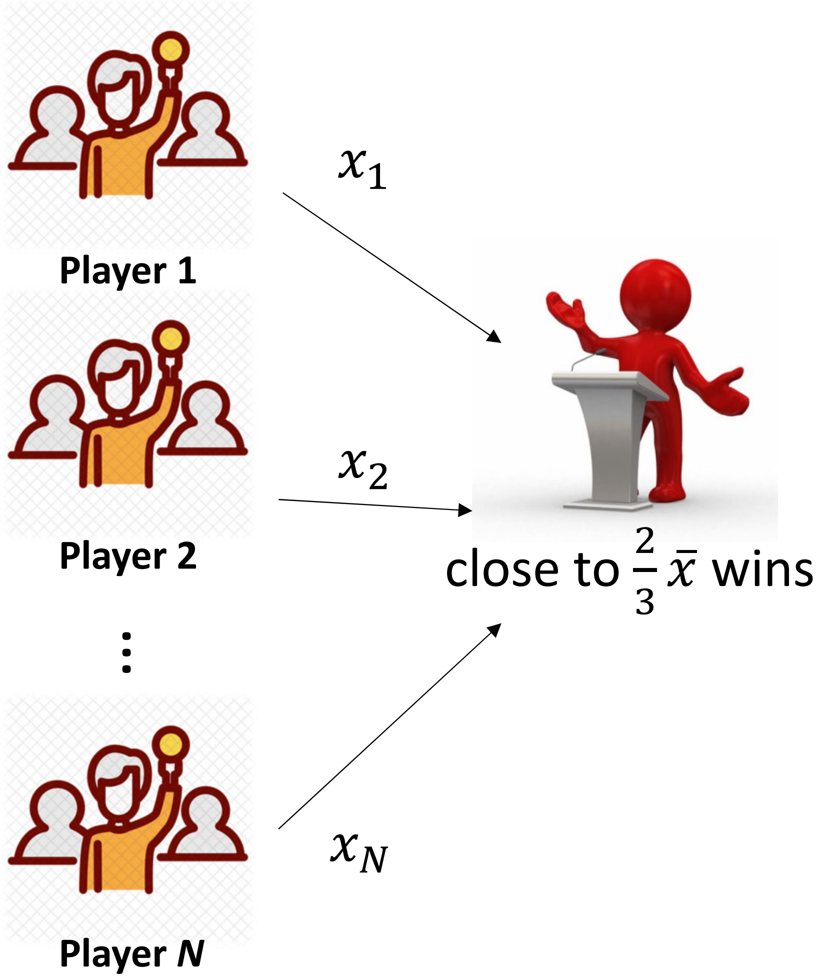
Nobody has incentive to deviate from their action if an equilibrium profile is played.

You may read:
Osborne M J. An introduction to game theory[M]. New York: Oxford university press, 2004.

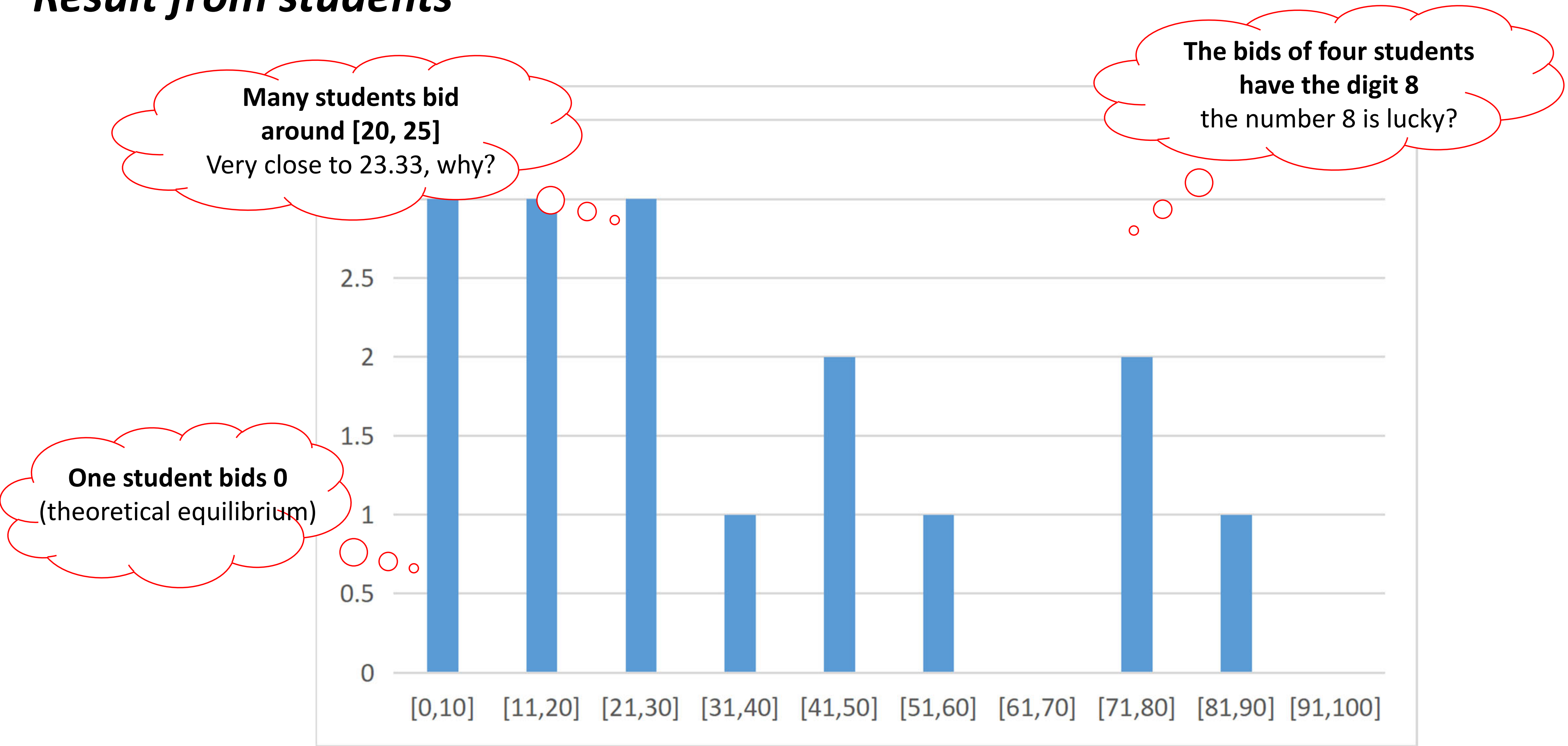
What if the objectives are pursued by different agents? (game theory)

The game in homework-3:

- Each player names an integer between 0 and 100
- The player who names the integer closest to 2/3 of the average integer wins!
- Three questions:
  - What will other players do?
  - What should I do in response?
  - Each player best responds to the others: *equilibrium*



Result from students



The average integer is 35, and $\frac{2}{3} \times 35 = 23.33$, the student names 24 wins!

Strategic Reasoning

- Suppose a player believes the average will be $\bar{x}$
- That player's optimal strategy is to say the closest integer to $\frac{2}{3}\bar{x}$
- $\bar{x}$ has to be less than 100, so the optimal strategy is no more than 67
- If $\bar{x}$ is no more than 67, then the optimal strategy is no more than $\frac{2}{3}67$
- If $\bar{x}$ is no more than $\frac{2}{3}67$, then the optimal strategy is no more than $\left(\frac{2}{3}\right)^2 67$
- Iterating, the unique Nash equilibrium of this game is everyone bids 0.

Theoretically, everyone will bid 0, but…. 24 wins, why? Bounded rationality

Thanks！