CS 580
ALGORITHM DESIGN AND ANALYSIS

Linear programming 1

Vassilis Zikas

LINEAR PROGRAMMING

- Broad class of optimization problems where we have :
  - Linear constraints (tree must touch all the nodes, path must go from s to t, subsequence must be increasing, etc)
  - Linear objective (value of a flow, length of a path, etc)

LINEAR PROGRAMMING

- Input
  - A set of n variables
  - A set of m linear constraints (wrt to the variables)
  - A linear objective (and whether we want to maximize of minimize)
- Output
  - Decide on real values for the variables that satisfy the constraints, and make the objective as large/small as possible

EXAMPLE (AVRIM BLUM)

- There are 24*7=168 hours in a week
- Some of it you'll study
  - Encode as variable x_S
- Some of it you'll party
  - Encode as variable x_P
- Some of it you'll do something else
  - Encode as variable x_E

EXAMPLE (AVRIM BLUM)

- Constraints:
  - $x_S + x_P + x_E = 168$
  - (Maintain sanity) $x_P + x_E \geq 70$
  - (Pass courses) $x_S \geq 60$
  - $2x_S + x_E \geq 150 + 2x_P$ (too little sleep with too much partying makes it difficult to study)
- Question 1: Feasibility
  - Are there choices for x_S, x_P, x_E that satisfy all the constraints?
  - Yes! $x_S = 80, x_P = 20, x_E = 68$

EXAMPLE (AVRIM BLUM)

- Constraints:
  - $x_S + x_P + x_E = 168$
  - (Maintain sanity) $x_P + x_E \geq 70$
  - (Pass courses) $x_S \geq 60$
  - $2x_S + x_E \geq 150 + 2x_P$ (too little sleep with too much partying makes it difficult to study)
- Optimization:
  - maximize happiness: $x_P + 2x_E$
- Question 2: Can we find a feasible solution which maximizes the objective?

LINEAR PROGRAMMING

- Generally, three possibilities:
  - The program has a feasible and optimal solution
  - The program is infeasible
    - E.g. $x_1 \geq 100, x_1 \leq 10$
  - The program is unbounded
    - E.g., max x , subject to $x \geq 0$

LINEAR PROGRAMMING

- Generally, the objective is optional
- In fact, you can incorporate the objective as a new constraint
  - Instead of $\max x_1 + 3x_2$
  - $x_1 + 3x_2 \geq C$, for some constant C
  - Solve again and again until you find the largest C
  - Question: Is this efficient??
- But, mandatory to have **linear** expressions

LINEAR PROGRAMMING

- Examples of non-linear constraints:
  - $x_1 \cdot x_2 \geq 3$
  - $x^2 + 3y \leq 10$
  - $\max(x, y) \geq 1$
  - $x \in \{0,1\}$

VISUALIZING A LINEAR PROGRAM

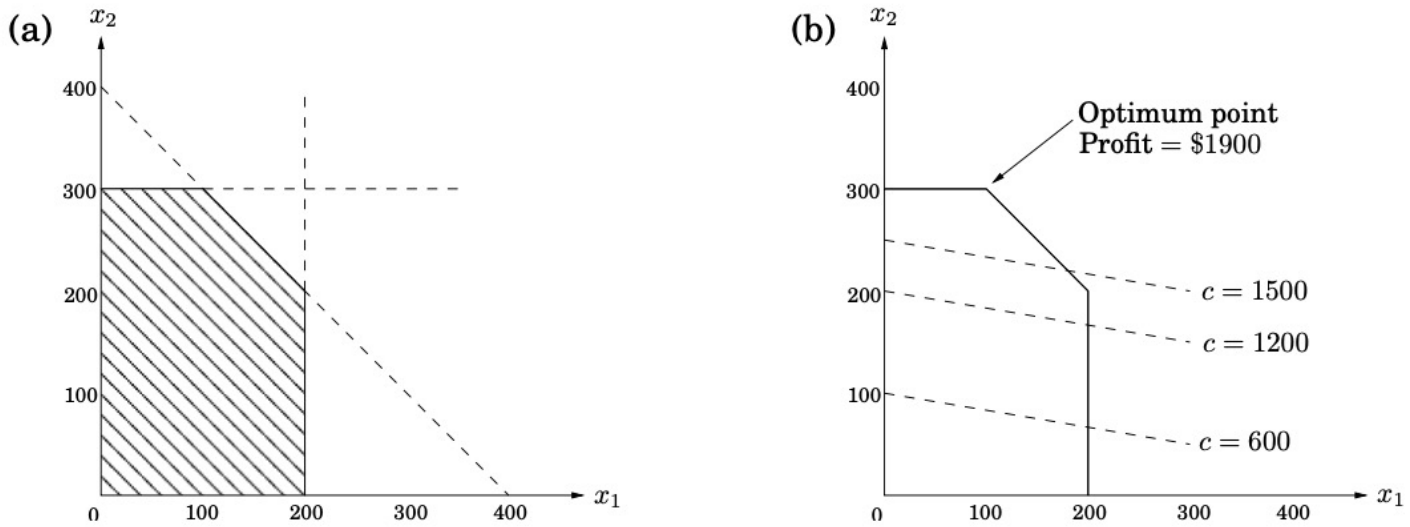
- We want to sell pizzas to maximize profit
- There are two types of pizzas: deep dish and Detroit style
  - A Detroit style slice sells for 1$
  - A deep dish slice sells for 6$
  - Our pizza oven can make at most 200 Detroit style pizza slices and 300 slices of deep dish
  - If we mix, we can make at most 400 slices in total

VISUALIZING A LINEAR PROGRAM

- Variables x_1 and x_2 for number of slices of Detroit style and deep dish
- $\max x_1 + 6x_2$
- Subject to
  - $x_1 \leq 200$
  - $x_2 \leq 300$
  - $x_1 + x_2 \leq 400$
  - $x_1, x_2 \geq 0$

VISUALIZING A LINEAR PROGRAM

Figure 7.1 (a) The feasible region for a linear program. (b) Contour lines of the objective function: $x_1 + 6x_2 = c$ for different values of the profit c.



MAX FLOW AS AN LP

- Let's try to write an LP for something useful
- Max flow problem:
  - Input: Directed graph $G = (V, E)$, with a capacity $c(e)$ on each edge $e \in E$ and two vertices s, t
  - Output: The maximum (valid) $s - t$ flow
- Can "valid flow" be encoded using linear constraints?
- Natural choice of variables: x_e for the amount of flow that goes through edge e

MAX FLOW AS AN LP

- What does valid mean?
  - Flow at most capacity: $x_e \leq c(e)$
  - Flow should be non-negative: $x_e \geq 0$
  - Flow conservation: for each vertex $v \in V \setminus \{s, t\}$ total incoming flow equals total outgoing flow
    - $\forall v \in V \setminus \{s, t\}$: $\sum_{e=(u,v), e \in E} x_e - \sum_{e=(v,u), e \in E} x_e = 0$
  - Total into t same as total out of s
  - Total number of constraints: $2|E|$ inequality plus $|V|$ equality constraints
- What's the objective?
  - Maximize total flow out of s
  - $\max \sum_{e=(s,u), e \in E} x_e$
- Done!
  - Minor caveat: This will only give us a fractional solution (for now)

MAX FLOW AS AN LP

- $\max \sum_{u:e=(s,u)\in E} x_e$
- Subject to
  - $\forall e \in E\colon\ x_e \leq c(e)$
  - $\forall v \in V \setminus \{s,t\}\colon \sum_{u:e=(u,v)\in E} x_e - \sum_{u:e=(v,u)\in E} x_e = 0$
  - $\forall e \in E\colon\ x_e \geq 0$

KNAPSACK AS AN LP

- Variable x_i for amount of item i in the knapsack
- $\max \sum_i x_i v_i$
- Subject to
  - $0 \leq x_i \leq 1$, for all items i
  - $\sum_i x_i w_i \leq W$

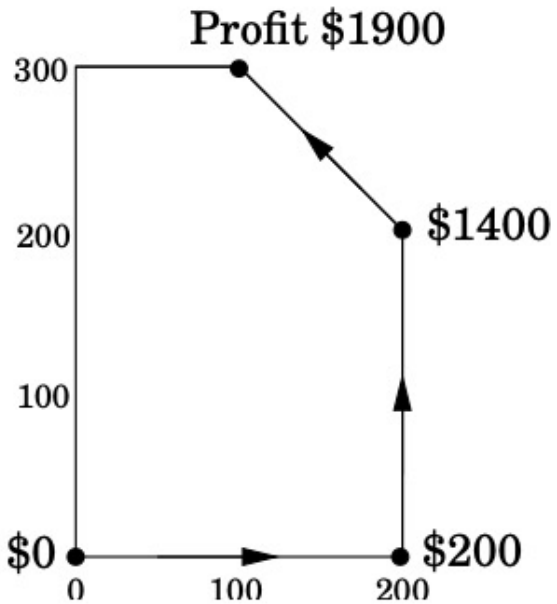
INTERVAL SCHEDULING

- Input: n jobs with start/end times s_i, t_i
- Objective: maximum number of compatible jobs
- Variable for fraction of each job scheduled
- $\max \sum_i x_i$
- Subject to
  - $\sum_{i:t\in[s_i,t_i]} x_i \leq 1$, for all times t (no overlaps)
  - $x_i \geq 0$
- Issue: does not give integral solution (for now)

SOLVING LINEAR PROGRAMS

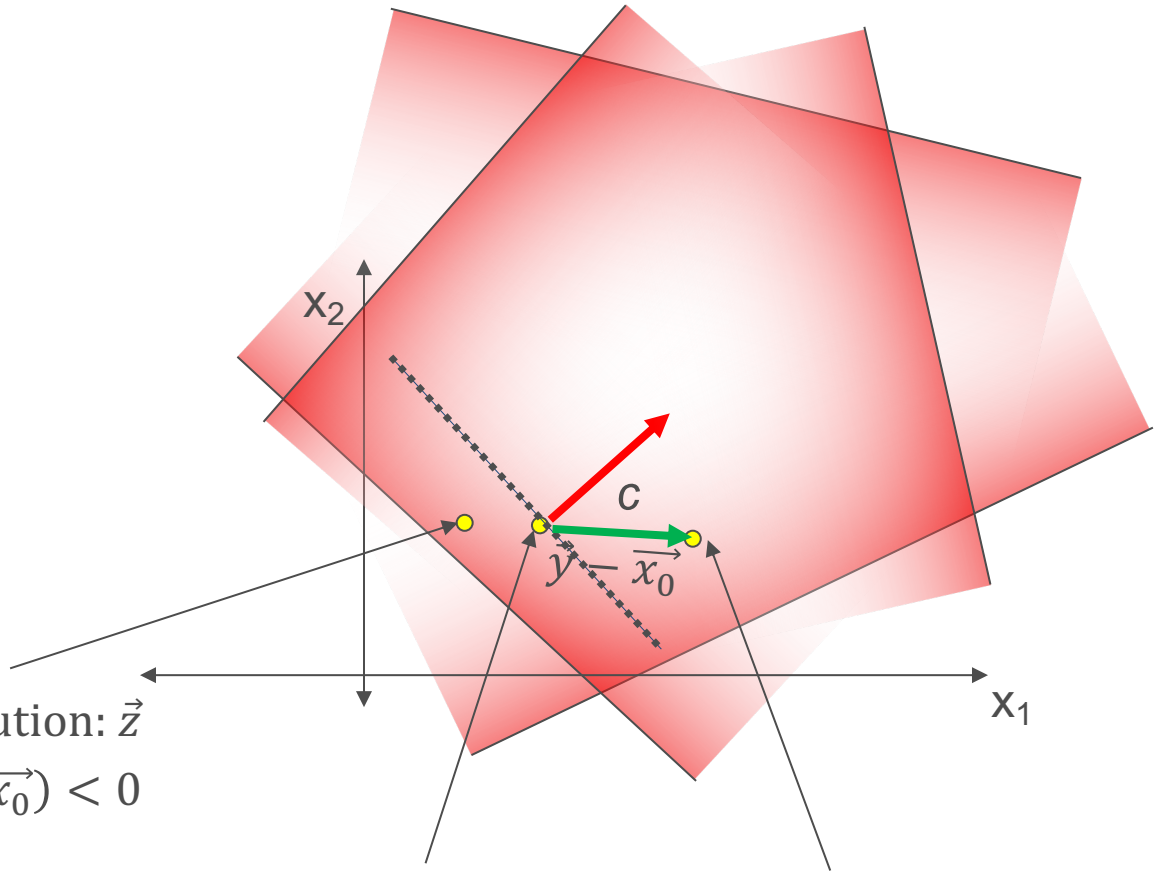
- Linear programs can be solved by the simplex method, devised by George Dantzig in 1947
- This algorithm runs in exponential time in the worst-case
- However, it has been observed to run really fast "in practice"
  - What does that even mean though?
  - Maybe some other time we'll talk about smooth analysis

SIMPLEX METHOD



SOLVING LINEAR PROGRAMS

- Goal: Maximize $2x_1+3x_2$ $c=(2,3)$



Worse Solution: $\vec{z}$

$\overrightarrow{C^T} \cdot (\vec{z} - \overrightarrow{x_0}) < 0$

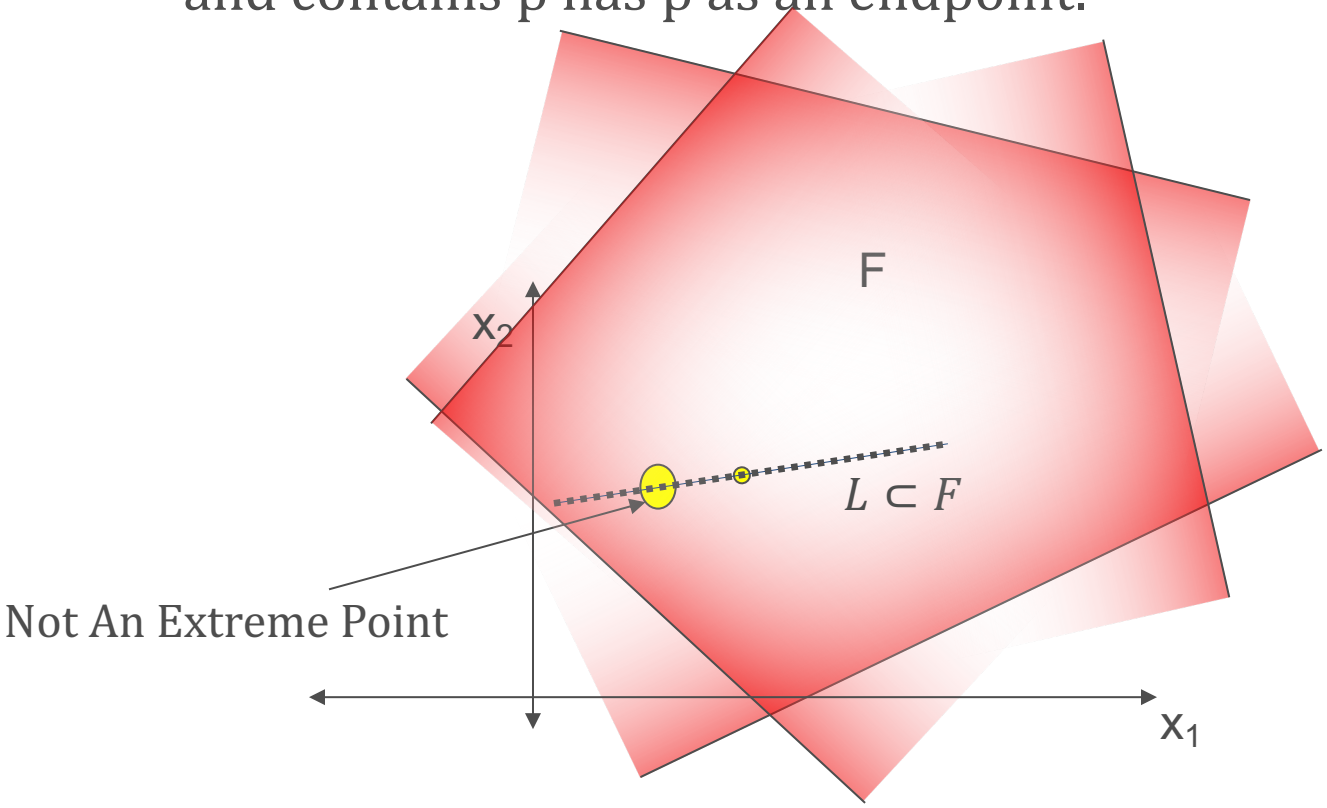
Initial Feasible Point: $\overrightarrow{x_0}$

Improved Solution: $\vec{y}$

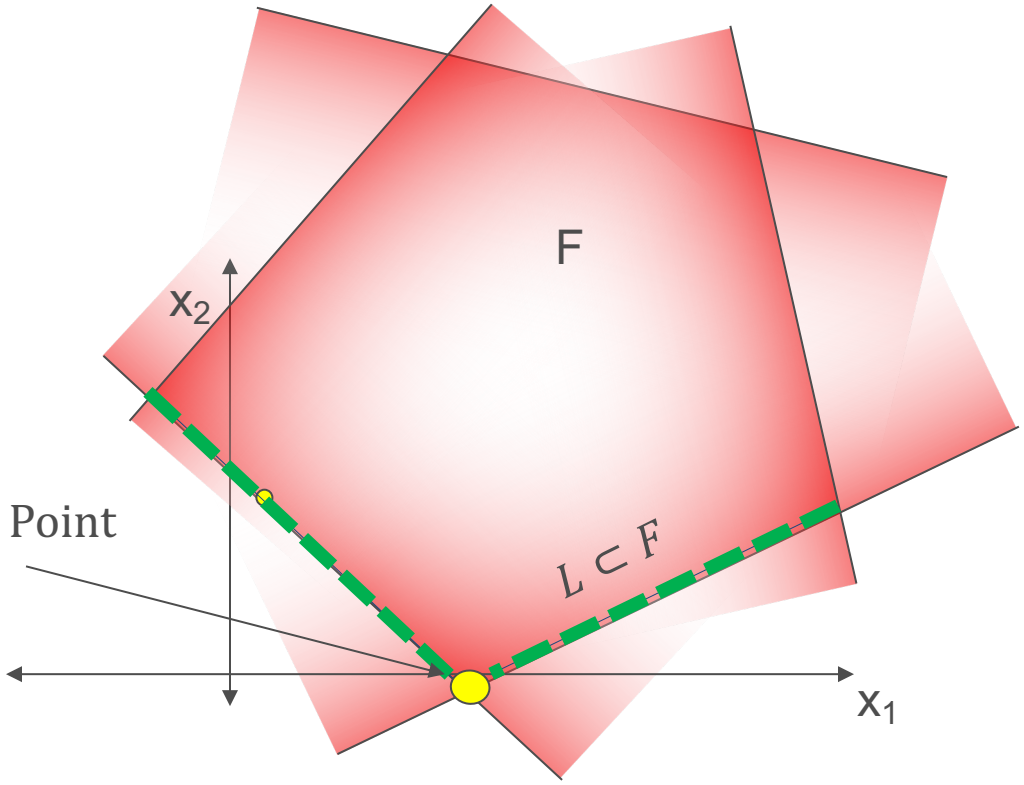
$\overrightarrow{C^T} \cdot (\vec{y} - \overrightarrow{x_0}) > 0$

LINEAR PROGRAMMING

- **Theorem:** Maximum value achieved at vertex (extreme point)
- **Definition:** Let F be the set of all feasible points in a linear program. We say that a point $p \in F$ is an extreme point (vertex) if *every* line segment $L \subset F$ that lies completely in F and contains p has p as an endpoint.



LINEAR PROGRAMMING



Is an Extreme Point

LINEAR PROGRAMMING

- **Theorem:** Maximum value achieved at vertex (extreme point)

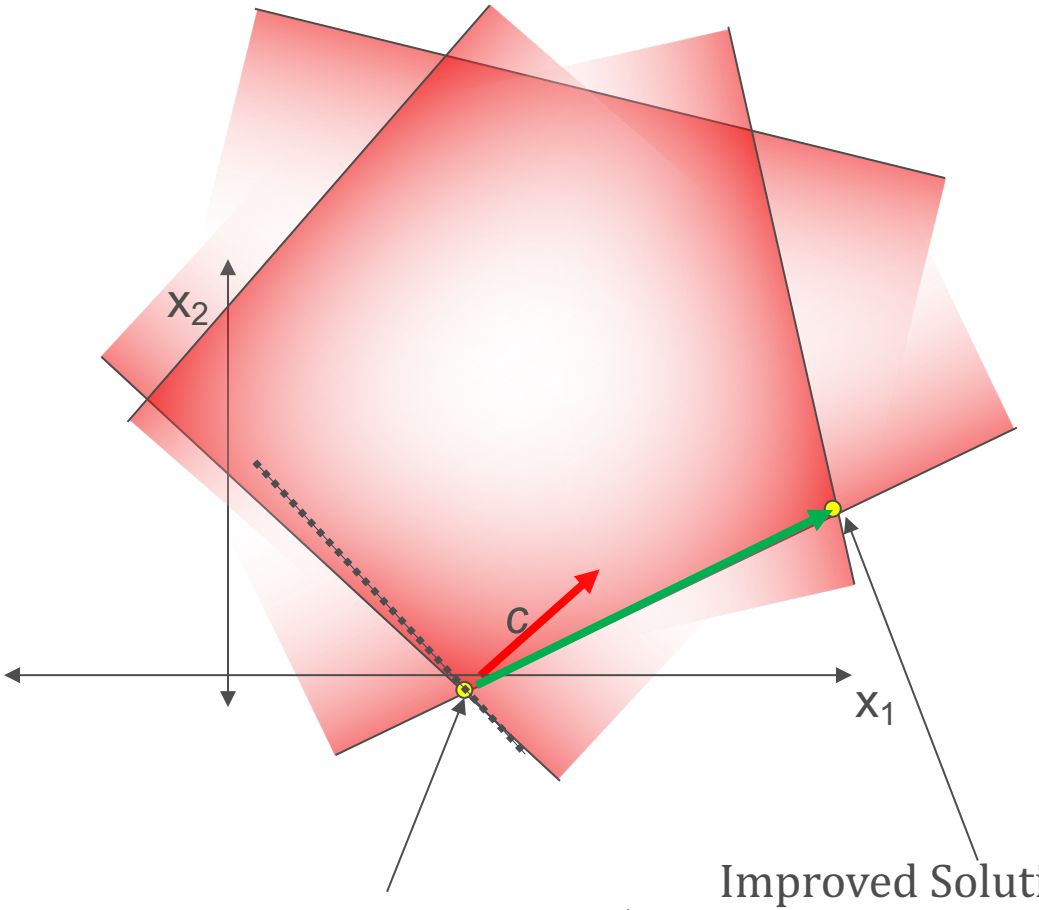
- **Definition:** Let F be the set of all feasible points in a linear program. We say that a point $p \in F$ is an extreme point (vertex) if *every* line segment $L \subset F$ that lies completely in F and contains p has p as an endpoint.

- **Observation**: Each extreme point lies at the intersection of n linearly independent constraints (where n is the number of variables)

- **Theorem:** a vertex is an optimal solution if there is no better *neighboring vertex*.
  - Main idea of simplex!

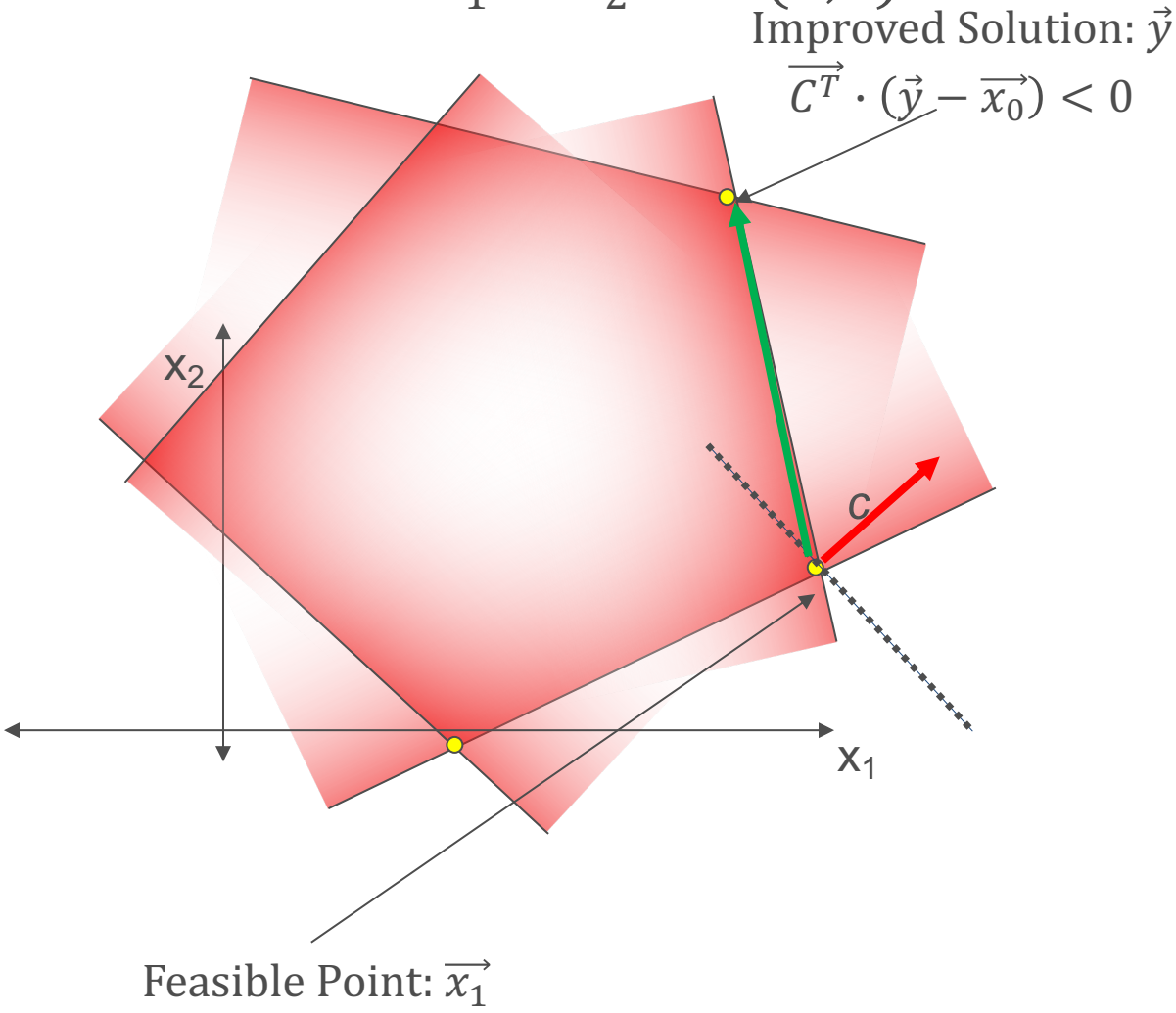
VERTEX WALKING

- Goal: Maximize $2x_1+3x_2$ $c=(2,3)$



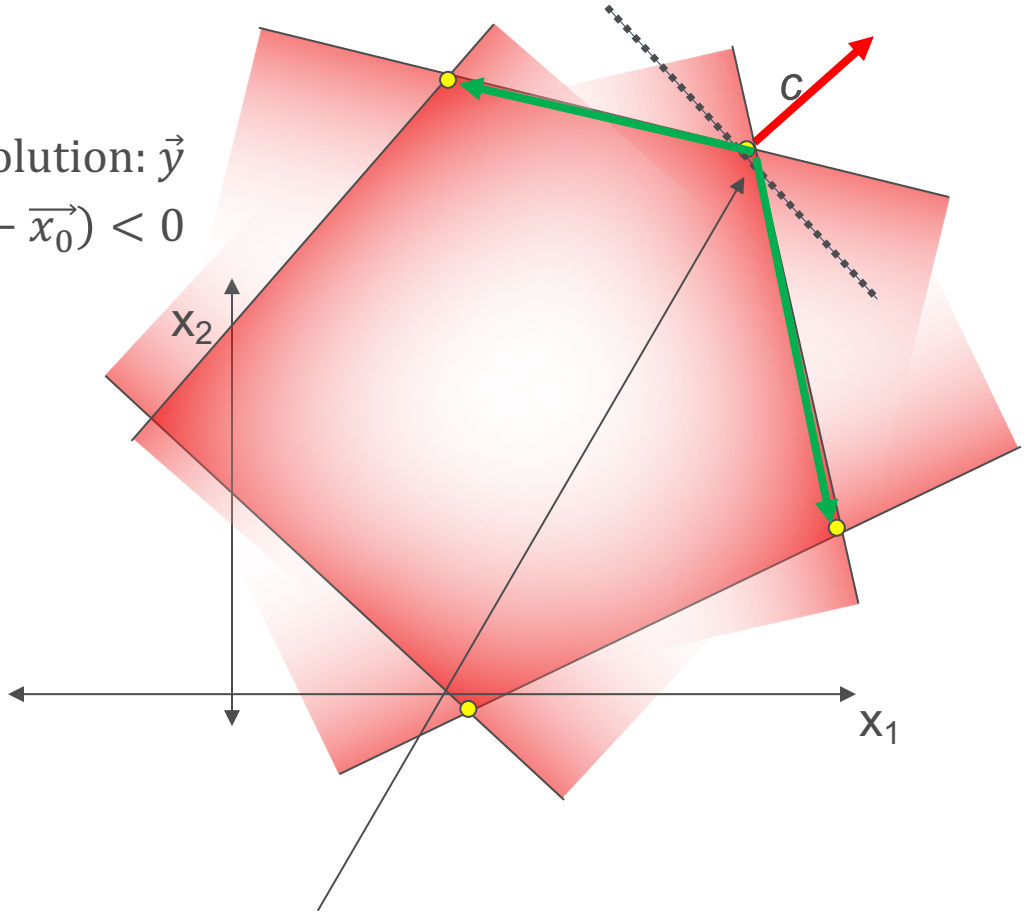
VERTEX WALKING

- Goal: Maximize $2x_1+3x_2$ c=(2,3)



ALGORITHMIC IDEA: VERTEX WALKING

- Goal: Maximize $2x_1+3x_2$ $c=(2,3)$



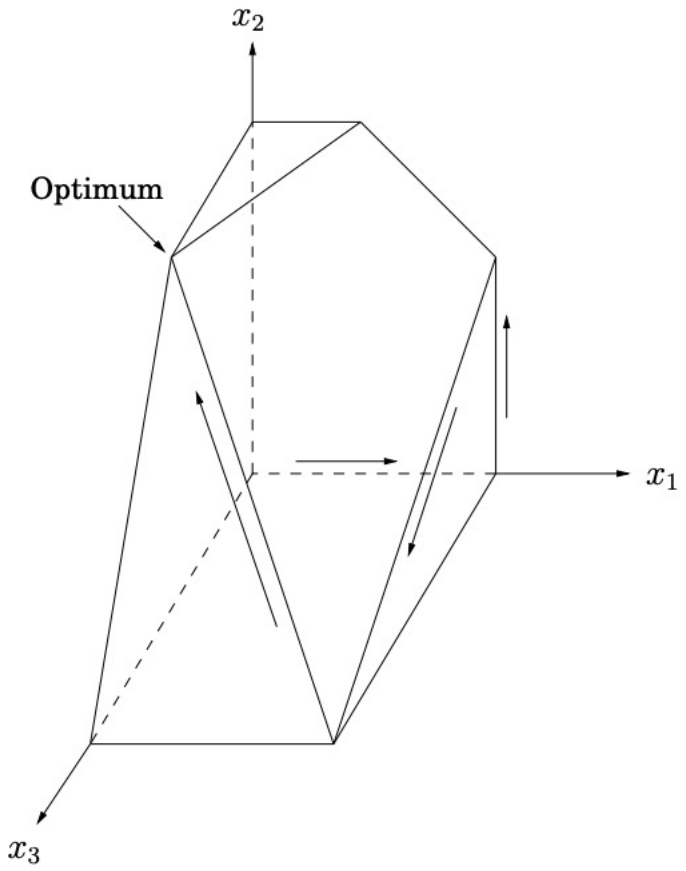
Worse Solution: $\vec{y}$

$\overrightarrow{C^T} \cdot (\vec{y} - \overrightarrow{x_0}) < 0$

Optimal Point: $\overrightarrow{x_2}$

SOLVING LINEAR PROGRAMS

Figure 7.2 The feasible polyhedron for a three-variable linear program.



SOLVING LINEAR PROGRAMS

- The ellipsoid algorithm (1980s) can solve a linear program in polynomial time
  - Huge theoretical breakthrough, but slow in practice
  - So, we have an exponential time algorithm fast in practice and a polynomial time algorithm that is slow in practice?? How fun…
- A few years later, Narendra Karmarkar developed a different polynomial time algorithm, that is also fast in practice, the interior point method.

SUMMARY

- Linear programming (29 in CLRS)
  - Basic definitions
  - Formulating problems
  - Simplex
- Next time:
  - Duality!