CS240 Algorithm Design and Analysis

Lecture 24

Randomized Algorithms

Quan Li
Fall 2024
2024.12.19

Linear Programming

Linear Programming

- A farmer has the following problem.
  - He has 100 acres of land, on which he can plant wheat or barley or both.
  - He has 420 kg of fertilizer, and 160 kg of pesticide.
  - Each acre of barley requires 5 kg of fertilizer and 1 kg of pesticide.
  - Each acre of wheat requires 3 kg of fertilizer and 2 kg of pesticide.
  - Wheat sells for \$3 per acre, barley sells for \$4 per acre.
    - Actually, he could probably make \$300 for wheat and \$400 for barley. Choose \$3 and \$4 for simplicity.
- How many acres of wheat and barley should the farmer plant his field to maximize his income?
- Let w, b be acres of wheat and barley farmer plants.
- He wants to maximize 3w+4b, subject to the land, pesticide and fertilizer constraints.

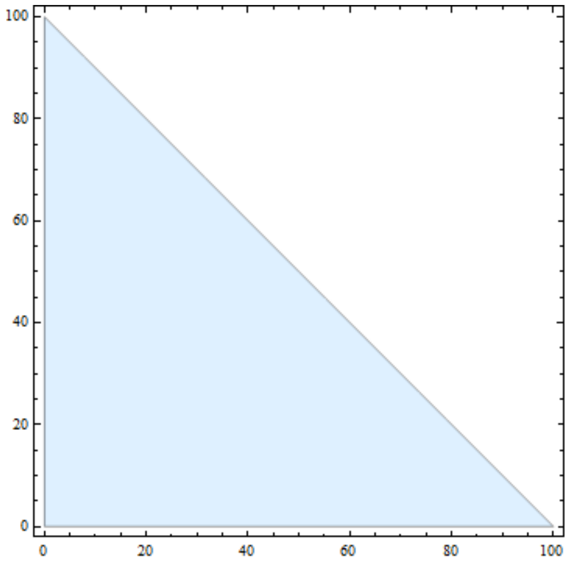
Linear Programming

- **maximize** 3w+4b **subject to**

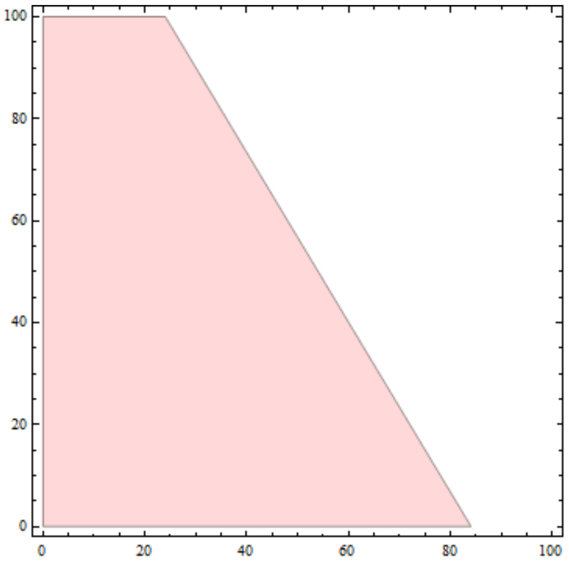
  w+b ≤ 100 (land)

  3w+5b ≤ 420 (fertilizer)

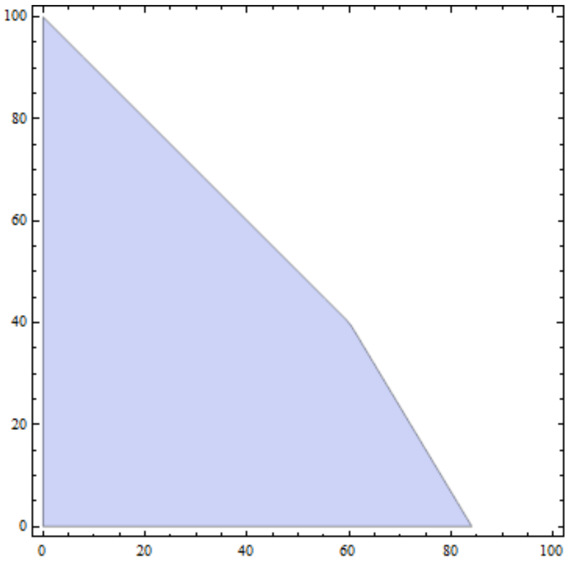
  2w+b ≤ 160 (pesticide)



land constraint



fertilizer constraint



land + fertilizer constraints

立志成才报国裕民

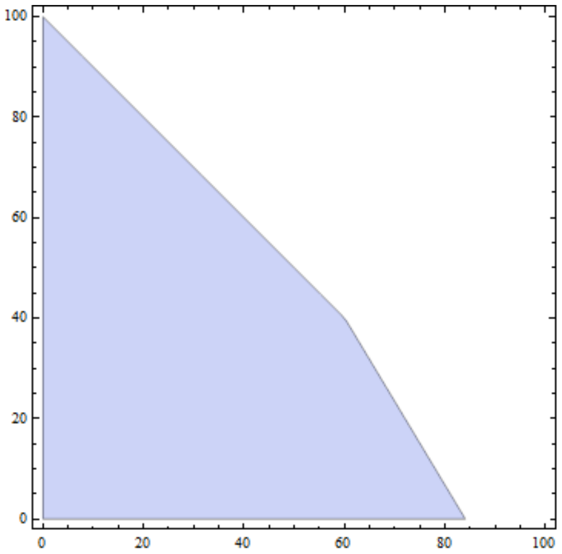
Linear Programming

- **maximize** 3w+4b **subject to**

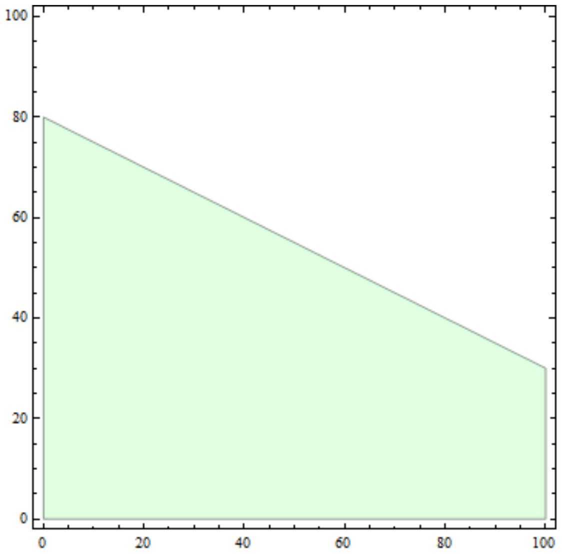
  w+b ≤ 100 (land)

  3w+5b ≤ 420 (fertilizer)

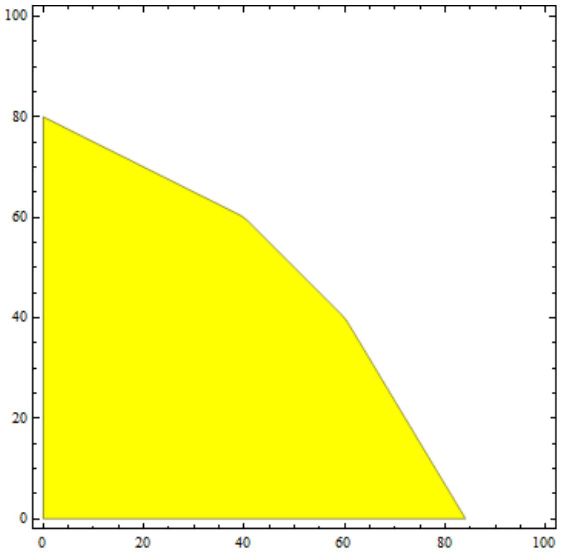
  2w+b ≤ 160 (pesticide)



land + fertilizer constraints



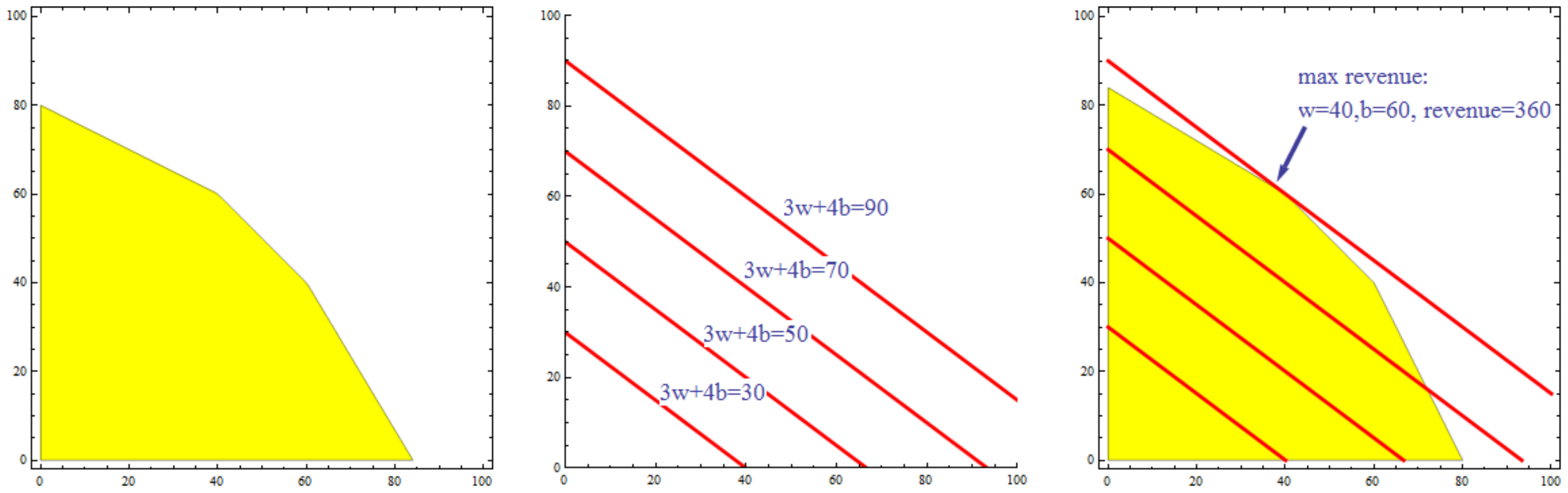
pesticide constraints



all three constraints

Linear Programming

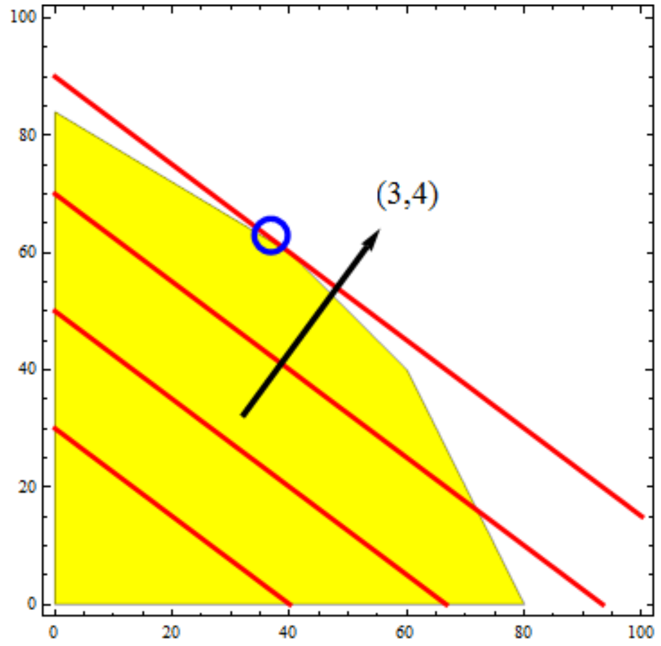
- **maximize** 3w+4b **subject to**
  w+b $\leq$ 100 (land)
  3w+5b $\leq$ 420 (fertilizer)
  2w+b $\leq$ 160 (pesticide)



上海科技大学
ShanghaiTech University
立志成才 报国裕民

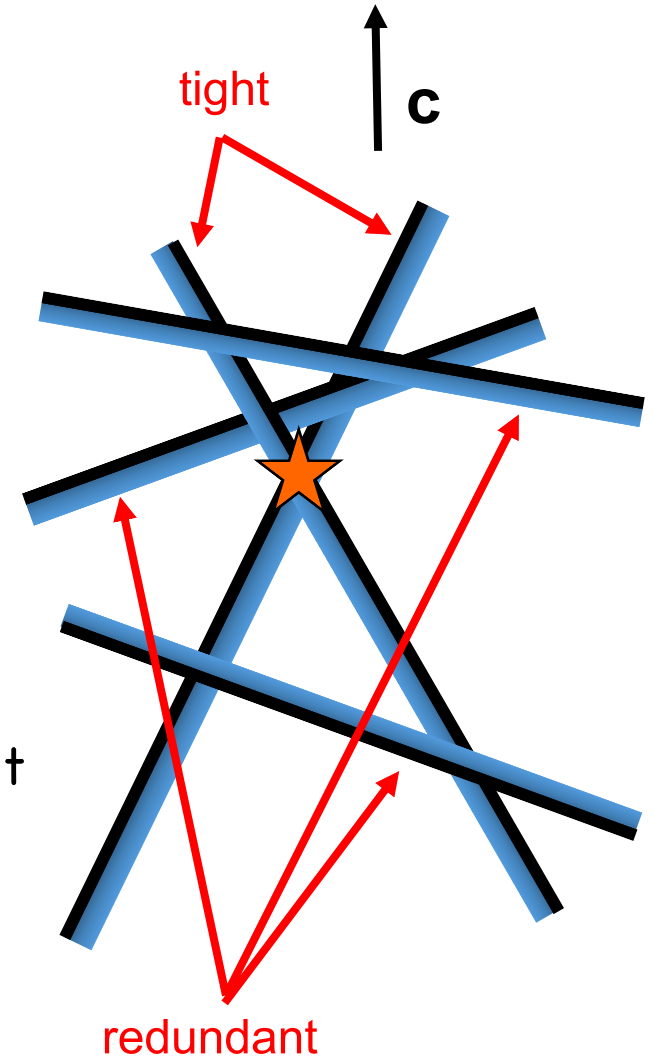
Linear Programming

- The feasible region is the area corresponding in which all the constraints are satisfied.
- Key Fact The optimum lies at an extreme point (corner).
- Find optimum by taking a line perpendicular to the direction pointed by the objective function, and shifting the line till when it will stop touching the feasible region.
- The optimum lies at the intersection of two constraints.
  - Call these the basis of the optimum.
  - For simplicity, assume constraints are general position, i.e., no 3 intersect at a point.



Randomized LP in 2D

- Since the optimum is defined by two constraints, the other constraints are redundant!
- A constraint is tight if the optimum lies on its defining line.
- Let H be set of n constraints. If pick random constraint, there's only 2/n probability it's tight.
- If constraint's not tight, we can discard it without changing optimum.
- How do we tell if it's tight?
  - For any constraint set G, let B(G) denote optimum.
- $h \in H$ is redundant iff $B(H)=B(H-\{h\})$.
  - i.e., the optimum is the same with or without h. So opt doesn't lie on h.

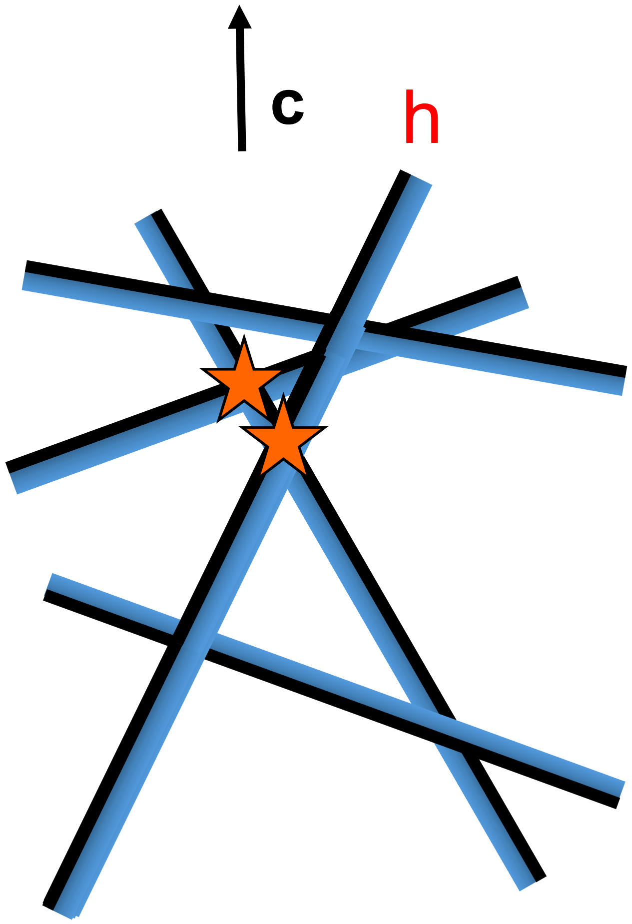


2D LP Algorithm

- If $|H|=2$, output intersection of the 2 halfplanes.
- Pick random constraint $h \in H$.
- Recursively find opt=B(H-{h}).
- If opt doesn't violate h, output opt.
  - opt violates h if opt lies outside h.
- Else project H-{h} onto h's boundary to obtain a 1D LP.
- Output the opt of the 1D LP.

Projection

- Given constraint h, let $\partial(h)$ be its boundary, i.e., the line defining h.
- Suppose B(H−{h}) violates h.
  - Then B(H) must lie on the boundary of h.
- Project a halfplane onto $\partial(h)$, reducing it to a line segment bounded on one or two sides.
- After projecting all H−{h} onto $\partial(h)$, we're left with a segment representing feasible region to 1D LP.
- Optimizing this is easy. The opt is one of the endpoints.

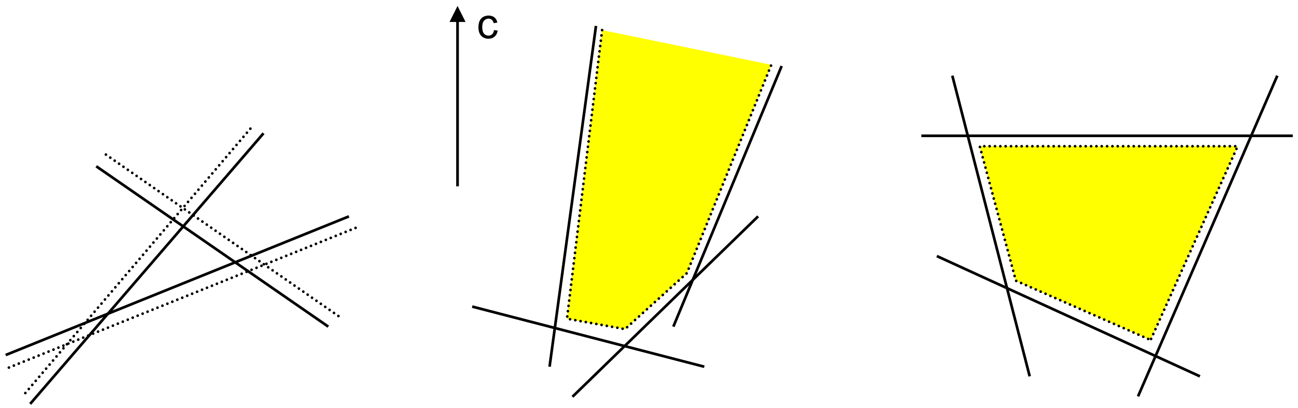


Analysis

- Let T(n) be expected time to solve 2D LP with n constraints.
- $T(n) \leq T(n-1)+O(1)+2/n(O(n)+O(1))$.
- T(n-1) time recursively find opt=B(H-{h}).
- First O(1) is time to check whether opt violates h.
- There's 2/n probability opt violates h, in which case we project all constraints onto $\partial(h)$.
  - O(n) to project H-{h} onto $\partial(h)$.
  - Final O(1) to solve 1D LP.
- T(n) solves to O(n).
  - So we can solve 2D LP with n constraints in expected linear time.

Corner Cases

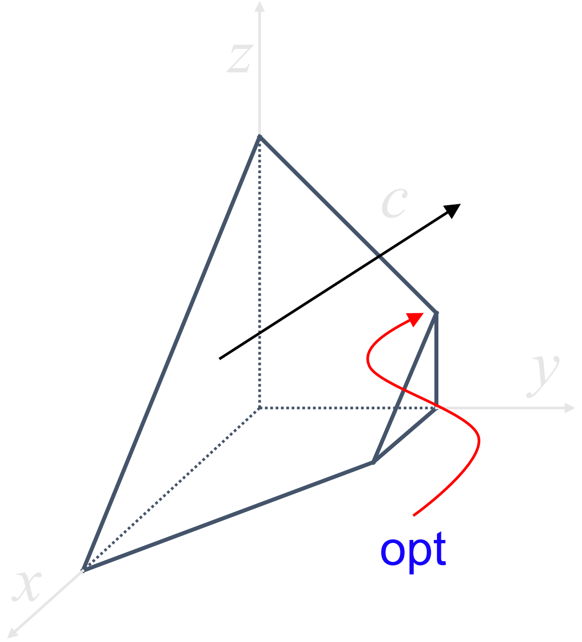
- For simplicity, we ignored several corner cases.
  - Infeasible means no points satisfy all the constraints.
    - Ex Constraints $x_1>1$ and $x_1<0$.
  - Unbounded means the optimum is infinite.
    - Ex Maximize x_2 s.t. $x_1+x_2>0$.
  - Non-unique optimum means an infinite number of points maximize the objective.
    - Ex Maximize x_2 s.t. $x_2 \leq 0$.
- Preprocess input to check for corner cases.

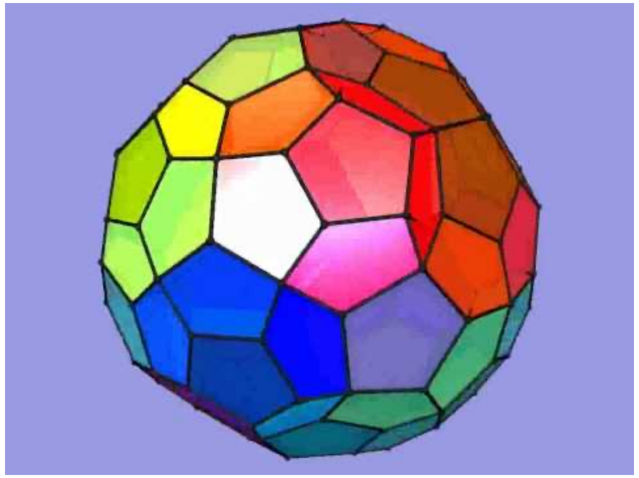


Infeasible Unbounded Non-unique optimum

Higher Dimensions

- In d>2 dimensions, lines become planes and each constraint corresponds to the space to one side of a plane, called a halfspace.
- The intersection of the halfspaces defines the feasible region.
  - This is a convex region called a polytope.
- Each extreme point (corner) of the polytope is the intersection of d halfspaces.
- The objective function defines a direction. Take a plane perpendicular to this direction and shift it till it stops touching feasible region.
- Hence optimum again lies at intersection of d halfspaces.
- Polytopes can be very complicated.





d-Dimensional LP Algorithm

- If |H|=d, output their intersection.
- Pick random constraint h∈H.
- Recursively find opt=B(H-{h}).
- If opt doesn't violate h, output opt.
- Else project H-{h} onto h's boundary to obtain a d-1 dimensional LP.
- Recursively solve the d-1 dim LP.

Analysis

- Let T(n,d) be expected time to solve d-dim LP with n constraints.
- T(n,d) $\leq$ T(n-1,d)+O(d)+d/n(O(dn)+T(n-1,d-1)).
- T(n-1,d) time recursively find opt=B(H-{h}).
- O(d) time to check whether opt violates h.
- There's d/n probability opt violates h.
  - Because opt is defined by d of the n halfspaces.
  - In this case we project all constraints onto $\partial(h)$.
  - O(dn) to project H-{h} onto $\partial(h)$.
  - We obtain a d-1 dim LP with n-1 constraints.
  - T(n-1,d-1) time to solve this.
- T(n,d) solves to O(d! n)
  - Linear in number of constraints.
  - Exponential in dimensions.

Matrix Formulation

- **maximize** 3w+4b **subject to**

  w+b ≤ 100

  3w+5b ≤ 420

  2w+b ≤ 160

- $$\text{maximize}\,[3,4]\cdot\begin{bmatrix} w \\ b \end{bmatrix}\text{s.t.}$$

- $$\begin{bmatrix} 1 & 1 \\ 3 & 5 \\ 2 & 1 \end{bmatrix}\cdot\begin{bmatrix} w \\ b \end{bmatrix}\leq\begin{bmatrix} 100 \\ 420 \\ 160 \end{bmatrix}$$

  Let $x \in \Re^{n\times 1}, c \in \Re^{1\times n}, A \in \Re^{m\times n}, b \in \Re^{m\times 1}$.

  maximize $c \cdot x$ s.t.

  $A \cdot x \leq b$