Linear Optimization

Yinyu Ye

Department of Management Science and Engineering

Stanford University

Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye

Mathematical Programming (MP)

The class of mathematical programming problems considered in this course can all be expressed in the form

$$\begin{array}{lll} \text{(P)} & \text{minimize} & f(\mathbf{x}) \\ & \text{subject to} & \mathbf{x} \in \mathcal{X} \end{array}$$

where $\mathcal{X}$ usually specified by constraints:

$$\begin{array}{rcll} c_i(\mathbf{x}) & = & 0 & i \in \mathcal{E} \\ c_i(\mathbf{x}) & \leq & 0 & i \in \mathcal{I}. \end{array}$$

Global and Local Optimizers

A global minimizer for (P) is a vector $\mathbf{x}^*$ such that

$$\mathbf{x}^* \in \mathcal{X} \quad \text{and} \quad f(\mathbf{x}^*) \le f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X}.$$

Sometimes one has to settle for a local minimizer, that is, a vector $\bar{\mathbf{x}}$ such that

$$\bar{\mathbf{x}} \in \mathcal{X} \quad \text{and} \quad f(\bar{\mathbf{x}}) \le f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X} \cap N(\bar{x})$$

where $N(\bar{\mathbf{x}})$ is a neighborhood of $\bar{\mathbf{x}}$. Typically, $N(\bar{\mathbf{x}}) = B_\delta(\bar{\mathbf{x}})$, an open ball centered at $\bar{\mathbf{x}}$ having suitably small radius $\delta > 0$.

The value of the objective function f at a global minimizer or a local minimizer is also of interest. We call it the global minimum value or a local minimum value, respectively.

Linea Conic Optimization

The class of mathematical programming problems considered in this course can all be expressed in the form

$$
\begin{array}{lll}
\text{(P)} & \text{minimize} & \mathbf{c}^T\mathbf{x} \\
 & \text{subject to} & \mathbf{x} \in \mathcal{X}
\end{array}
$$

where $\mathcal{X}$ usually specified by linear and conic constraints:

$$
\begin{array}{ccc}
A\mathbf{x} & \{\leq, =, \geq\} & \mathbf{b} \\
\mathbf{x} & \in & \text{A Convex Cone.}
\end{array}
$$

Special Case: Linear Programming

•

$$
\begin{array}{rl}
\text{min(or max)imize} & c_1x_1 + c_2x_2 + ... + c_nx_n \\
\text{subject to} & a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \; \{\leq, =, \geq\} \; b_1, \\
& a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \; \{\leq, =, \geq\} \; b_2, \\
& \quad ..., \\
& a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \; \{\leq, =, \geq\} \; b_m, \\
& x_j \; \{\geq, \leq\} \; u_j, \quad j = 1, ..., n,
\end{array}
$$

- $$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

- $$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}.$$

-
$$\begin{array}{rl} \text{min(or max)imize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x}\ \{\leq, =, \geq\}\ \mathbf{b}, \\ & \mathbf{x}\ \{\geq, \leq\}\ \mathbf{0}. \end{array}$$

Important Terms

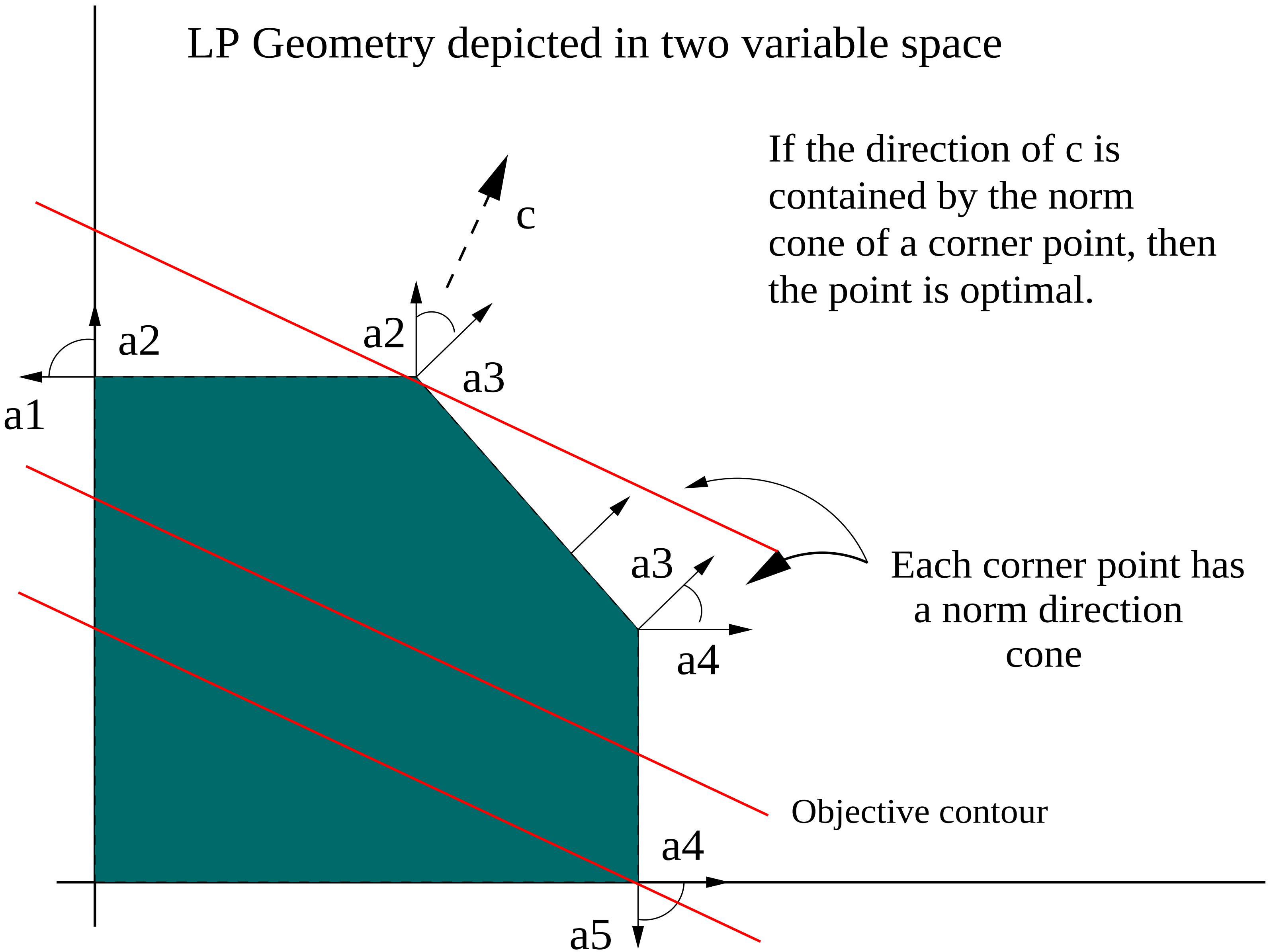
- decision variable/activity, data/parameter
- objective/goal/target
- constraint/limitation/requirement
- equality/inequality constraint
- constraint function/the right-hand side
- direction of inequality
- coefficient vector/coefficient matrix
- nonnegativity constraint
- integrality constraint
- satisfied/violated
- slack/surplus

Graphical Representation of LP

Consider

$$
\begin{array}{llll}
\text{maximize} & x_1 & +2x_2 & \\
\text{subject to} & x_1 & & \leq 1 \\
 & & x_2 & \leq 1 \\
 & x_1 & +x_2 & \leq 1.5 \\
 & x_1, & x_2 & \geq 0.
\end{array}
$$

LP Geometry depicted in two variable space



If the direction of c is contained by the norm cone of a corner point, then the point is optimal.

Linear Programming in Standard Form

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{array}$$

$\{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$ is the non-negative authant cone.

Reduction to the Standard Form

- Eliminating ”free” variable: use the difference of two nonnegative variables

$$x = x^+ - x^-, \quad x^+, x^- \geq 0.$$

- Eliminating inequality: add slack variable

$$\mathbf{a}^T\mathbf{x} \leq b \Longrightarrow \mathbf{a}^T\mathbf{x} + s = b, \quad s \geq 0$$

$$\mathbf{a}^T\mathbf{x} \geq b \Longrightarrow \mathbf{a}^T\mathbf{x} - s = b, \quad s \geq 0$$

- Eliminating upper bound: move them to constraints

$$x \leq 3 \Longrightarrow x + s = 3, \quad s \geq 0$$

- Eliminating nonzezro lower bound: shift the decision variables

$$x \geq 3 \Longrightarrow x := x - 3$$

Linear Conic Programming in Standard Form

Conic Linear Programming

$$\begin{array}{rl} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \in K, \end{array}$$

where K is a closed convex cone.

Math Programmming Terminology

- solution (decision, point): any specification of values for all decision variables, regardless of whether it is a desirable or even allowable choice
- feasible solution: a solution for which all the constraints are satisfied.
- feasible region (constraint set, feasible set): the collection of all feasible solution
- interior, boundary
- extreme point (corner)
- objective function contour (iso-profit, iso-cost line)
- optimal solution (optimum): a feasible solution that has the most favorable value of the objective function
- optimal (objective) value: the value of the objective function evaluated at an optimal solution

- active constraint (binding constraint)
- inactive constraint
- redundant constraint

Formulation 1: Air Traffic Control

Air plane $j, \quad j = 1, ..., n$ arrives at the airport within the time interval $[a_j, b_j]$ in the order of $1, 2, ..., n$. The airport wants to find the arrival time for each air plane such that the minimal metering time (inter-arrival time between two consecutive airplanes) is the greatest.

Let t_j be the arrival time of the jth plane. Then, the problem is

$$\begin{array}{ll} \text{maximize} & \min_{j=1,...,n-1}\{t_{j+1} - t_j\} \\ \text{subject to} & a_j \le t_j \le b_j, \quad j = 1, 2, ..., n. \end{array}$$

Do we need the constraint $t_{j+1} - t_j \ge 0$ for all j?

Air Traffic Control continued

Rewrite the problem as an LP:

$$
\begin{array}{ll}
\text{maximize} & \Delta \\
\text{subject to} & t_2 - t_1 - \Delta \geq 0, \\
& t_3 - t_2 - \Delta \geq 0, \\
& ..., \\
& t_n - t_{n-1} - \Delta \geq 0, \\
& a_j \leq t_j \leq b_j, \quad j = 1, 2, ..., n.
\end{array}
$$

Formulation: Four-Step Rule

- Sort out data and parameters from the verbal description
- Define the set of decision variables
- Formulate the objective function of data and decision variables
- Set up equality and/or inequality constraints

Formulation 2: Data Fitting I

Given data points $\mathbf{a}_j$, $j = 1, ..., n$, and the observation value c_j at data point $\mathbf{a}_j$, the least squares problem is to find $\mathbf{y}$ such that

$$\sum_j (\mathbf{a}_j^T \mathbf{y} - c_j)^2 = \|A^T \mathbf{y} - \mathbf{c}\|_2^2$$

is minimized.

Sometime, it is desired to minimize the p norm, where $p = 1$ or $p = \infty$,

$$\sum_j |\mathbf{a}_j^T \mathbf{y} - c_j| = \|A^T \mathbf{y} - \mathbf{c}\|_1 \quad \text{or} \quad \max_j |\mathbf{a}_j^T \mathbf{y} - c_j| = \|A^T \mathbf{y} - \mathbf{c}\|_\infty$$

Rewrite the problem as a linear program.

Data Fitting II

Suppose we want to minimize

$$\sum_i \|A_i^T \mathbf{y} - \mathbf{c}_i\|_2$$

This is equivalent to

$$\begin{array}{rl} \text{minimize} & \sum_i \delta_i \\ \text{subject to} & \|A_i^T \mathbf{y} - \mathbf{c}_i\|_2 \le \delta_i, \ \forall i \end{array}$$

It is a conic linear program.

Data Fitting III

Constrained data fitting–Fingerprint Matching: c_j is the measured signal strength from base-station j at a location, and $\mathbf{a}_j$ contains base-station j's signal strengths for all known individual locations.

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} |\mathbf{a}_j^T \mathbf{y} - c_j| \\ \text{subject to} & e^T \mathbf{y} = 1, \; y_i \in \{0, 1\}. \end{array}$$

LP relaxation:

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} |\mathbf{a}_j^T \mathbf{y} - c_j| \\ \text{subject to} & e^T \mathbf{y} = 1, \; \mathbf{y} \geq \mathbf{0}. \end{array}$$

Formulation 3: Transportation/Supply Chain Problem

Quantities s_i are to be shipped from m supply locations and received in amounts d_j in n demand locations, respectively.

$$
\begin{array}{lrl}
\min & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} & \\
\text{s.t.} & \sum_{j=1}^{n} x_{ij} & = s_i,\ \forall i = 1, ..., m \\
 & \sum_{i=1}^{m} x_{ij} & = d_j,\ \forall j = 1, ..., n \\
 & x_{ij} & \geq 0,\ \forall i, j.
\end{array}
$$

Assume that the total supply equal the total demand. Thus, exactly one equality constraint is redundant.

The problem has mn variables and $m + n$ equations.

Formulation 4: Supporting Vector Machine

Suppose we have two-class discrimination data. We assign the first class with 1 and the second with -1 for a binary varible. A powerful discrimination method is the Supporting Vector Machine (SVM).

Let the first class data points i be given by $\mathbf{a}_i \in R^d$, $i = 1, ..., n_1$ and the second class data points j be given by $\mathbf{b}_j \in R^d$, $j = 1, ..., n_2$.

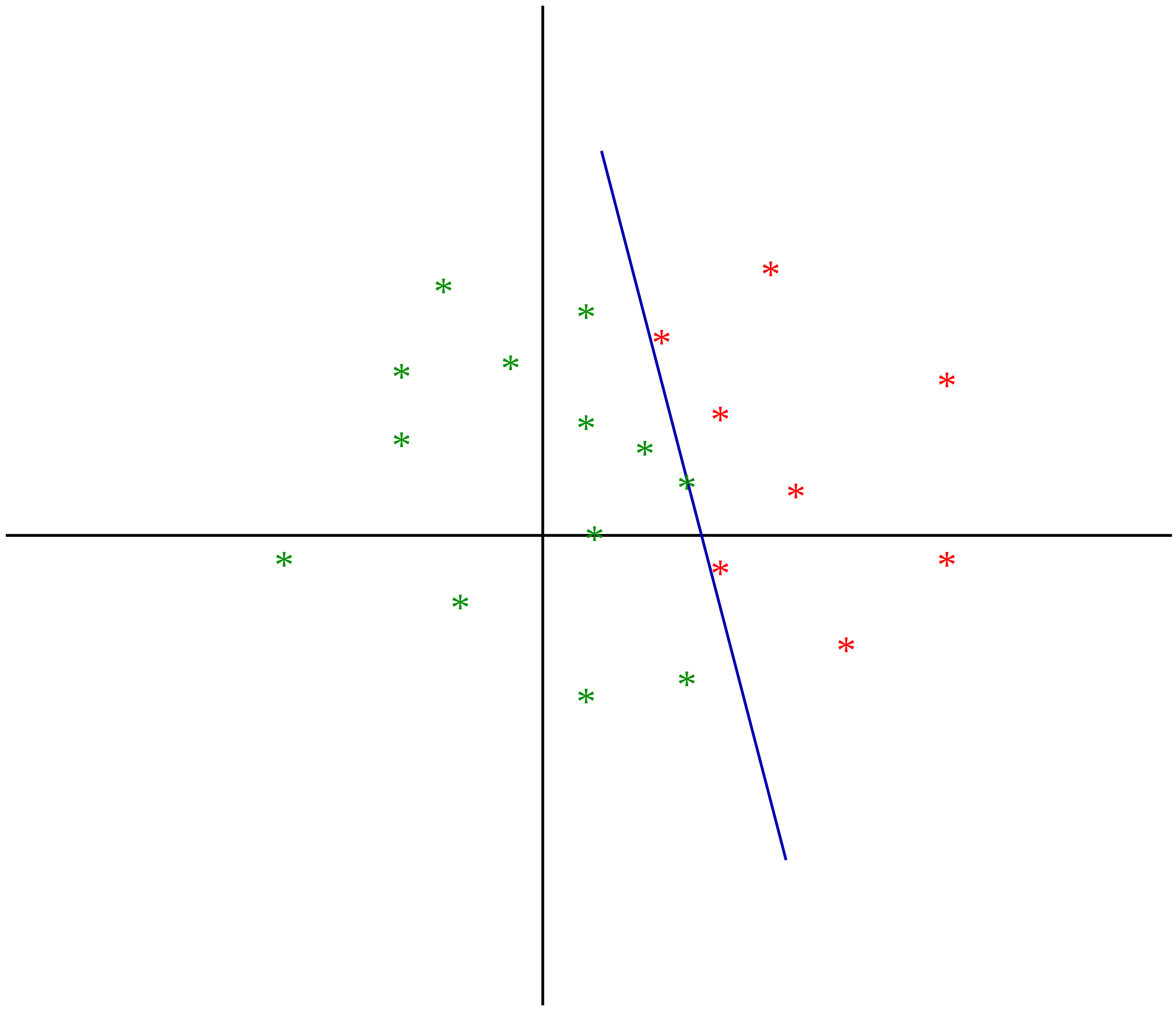


Figure 1: Linear Support Vector Machine

Supporting Vector Machine continued

We wish to find a hyper-plane in R^d to separate $\mathbf{a}_i$s (in red) from $\mathbf{b}_j$s (in green). Mathematically, we wish to find a slope vector $\mathbf{y} \in R^d$ and an intercept $\beta \in R$ such that

$$\mathbf{a}_i^T\mathbf{y} + \beta \geq 1 \; \forall i = 1, ..., n_1$$

and

$$\mathbf{a}_j^T\mathbf{y} + \beta \leq -1 \; \forall j = 1, ..., n_2.$$

This is an LP problem.

Once the slope vector $\mathbf{y} \in R^d$ and intercept $\beta \in R$ is fixed, the hyperp-lane would be

$$\{\mathbf{x} : \; \mathbf{y}^T\mathbf{x} + \beta = 0\}.$$

Supporting Vector Machine continued

If a clean separation is impossible, one can formulate the problem as an error minimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_i (\mathbf{a}_i^T \mathbf{y} + \beta - 1)^- + \sum_j (\mathbf{b}_j^T \mathbf{y} + \beta + 1)^+ \\ \text{subject to} & \mathbf{y} \in R^d, \ \beta \in R, \end{array}$$

which can be written as an LP problem:

$$\begin{array}{ll} \text{minimize} & \sum_i \delta_i + \sum_j \delta_j \\ \text{subject to} & \mathbf{a}_i^T \mathbf{y} + \beta + \delta_i \geq 1, \ \forall i, \\ & \mathbf{b}_j^T \mathbf{y} + \beta - \delta_j \leq -1, \ \forall j, \\ & \delta_i \geq 0, \ \delta_j \geq 0, \ \forall i, j. \end{array}$$

Here, $\delta_i > 0$ or $\delta_j > 0$ represents the possible error for a point on the wrong side.

Formulation 5: Combinatorial Auction I

Given m potential states that are mutually exclusive and exactly one of them will be realized at the maturity.

An order is a bet on one or a combination of states, with a price limit (the maximum price the participant is willing to pay for one unit of the order) and a quantity limit (the maximum number of units the participant is willing to accept).

A contract on an order is a paper agreement so that on maturity it is worth a notional \$$w$ dollar if the order includes the winning state and worth \$$0$ otherwise.

There are n orders submitted now.

Combinatorial Auction II: order data

The jth order is given as $(\mathbf{a}_j \in R^m_+,\ \pi_j \in R_+,\ q_j \in R_+)$: $\mathbf{a}_j$ is the betting indication vector where each entry is either 1 or 0

$$\mathbf{a}_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \ldots \\ a_{mj} \end{pmatrix},$$

where 1 is winning and 0 is non-winning; π_j is the price limit for one such a contract, and q_j is the maximum number of contracts the better like to buy.

Combinatorial Auction III: order fills

Let x_j be the number of units awarded to the jth order. Then, the jth bidder will pay the amount $\pi_j \cdot x_j$ and the total amount paid would be $\pi^T \mathbf{x} = \sum_j \pi_j \cdot x_j$.

If the ith state is the winning state, then the auction organizer need to pay the winning bidders

$$w \cdot \left(\sum_{j=1}^{n} a_{ij} x_j \right) = w \cdot \mathbf{a}_{i.} \mathbf{x}$$

The question is, how to decide $\mathbf{x} \in R^n$, that is, how to fill the orders.

Combinatorial Auction Pricing IV: worst-case profit maximization

$$
\begin{array}{lrl}
\max & \pi^T \mathbf{x} - w \cdot \max_i \{\mathbf{a}_i.\mathbf{x}\} & \\
\text{s.t.} & \mathbf{x} & \leq \mathbf{q}, \\
 & \mathbf{x} & \geq \mathbf{0}.
\end{array}
$$

$$
\begin{array}{lrl}
\max & \pi^T \mathbf{x} - w \cdot \max(A\mathbf{x}) & \\
\text{s.t.} & \mathbf{x} & \leq \mathbf{q}, \\
 & \mathbf{x} & \geq \mathbf{0}.
\end{array}
$$

Combinatorial Auction Pricing V: linear program

$$
\begin{array}{rrl}
\max & \pi^T \mathbf{x} - w \cdot s & \\
\text{s.t.} & A\mathbf{x} - \mathbf{e} \cdot s & \leq \mathbf{0}, \\
 & \mathbf{x} & \leq \mathbf{q}, \\
 & \mathbf{x} & \geq \mathbf{0}.
\end{array}
$$

$\pi^T \mathbf{x}$: the revenue amount can be collected.

$w \cdot s$: the worst-case cost (amount need to pay to the winners).