Deterministic OR Models

CO 370

Bertrand Guenin

LaTeXed by *Sibelius Peng*

Preface

Disclaimer Much of the information on this set of notes is transcribed directly/indirectly from the lectures of CO 370 during Fall 2021 as well as other related resources. I do not make any warranties about the completeness, reliability and accuracy of this set of notes. Use at your own risk.

What is operations research (OR)? There's no standard definitions for it. One particular definition: use of mathematical models to make complex decisions for real life problems. The origin is British military in WW2. OR is actually everywhere today. Key milestone: Simplex algorithm (1947).

Recall optimization problem is of the form:

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & \text{a set of constraints} \end{aligned}$$

There are some applications: mail delivery, machine scheduling, inventory problem, network design, facility location, class scheduling, portfolio optimization, surgery planning, sensor location.

For any questions, send me an email via https://notes.sibeliusp.com/contact.

You can find my notes for other courses on https://notes.sibeliusp.com/.

Sibelius Peng

Contents

Part I:

Formulations

1

LP formulations

1.1 Production problem

Products $J = \{1, \ldots, n\}$
Resources $I = \{1, \ldots, m\}$

Data:

- $\forall j \in J : c_j =$ value of unit of product j
- $\forall i \in I : b_i =$ number of units of resource i available
- $\forall i \in I, \forall j \in J : a_{ij} =$ number of units of resource i going to product j

Goal: maximize values of product made subject to available resources

Var: $x_j =$ number of units of product j produced

Then problem is

$$\begin{array}{rll} \max & \sum_{j \in J} c_j x_j & \\ \text{s.t.} & \sum_{j \in J} a_{ij} x_j \leq b_i & (i \in I) \\ & x_j \geq 0 & (j \in J) \end{array}$$

Now let's generalize this problem to have more than one period.

Products $J = \{1, \ldots, n\}$
Resources $I = \{1, \ldots, m\}$
Periods $K = \{1, \ldots, p\}$

Then we have **data**

- $\forall j \in J, k \in K$: $c_{jk} =$ unit value of product j in period k
- $\forall i \in I, k \in K$: $w_{ik} =$ unit price for resource i in period k
- $\forall i \in I, j \in J$: $a_{ij} =$ number of units of resource i going into product j

and the **goal**: decide how much of each resource to buy & how much of each product to make during each period, to maximize total profit. Unused resources are available at next time period.

Var:

- p_{ik} = number of units of resource i, purchased at start of period k
- x_{jk} = number of units of product j made in period k
- z_{ik} = number of units of resource i at the end of period k

$$\text{Profit} = \sum_{k\in K}\left[\sum_{j\in J} c_{jk}x_{jk} - \sum_{i\in I} w_{ik}p_{ik}\right] \tag{1.1}$$

Then we keep track of resources: for $i \in I$, $k \in K$

$$z_{ik} = z_{i(k-1)} + p_{ik} - \sum_{j\in J} a_{ij}x_{jk} \tag{1.2}$$

and we define for $i \in I$,

$$z_{i0} = 0 \tag{1.3}$$

Thus the optimization problem is

$$\begin{aligned} \max \quad & (1.1) \\ \text{s.t.} \quad & (1.2), (1.3) \\ & p, x, z \geq 0 \end{aligned} \tag{P}$$

Remark:
If (P) has a feasible solution of value that is bigger than 0, then (P) is unbounded. So we are missing some assumptions, maybe? For example, b_{ik} = amount of resource i that can be bound during period k. Then we can add constraints: $p_{ik} \leq b_{ik}$.

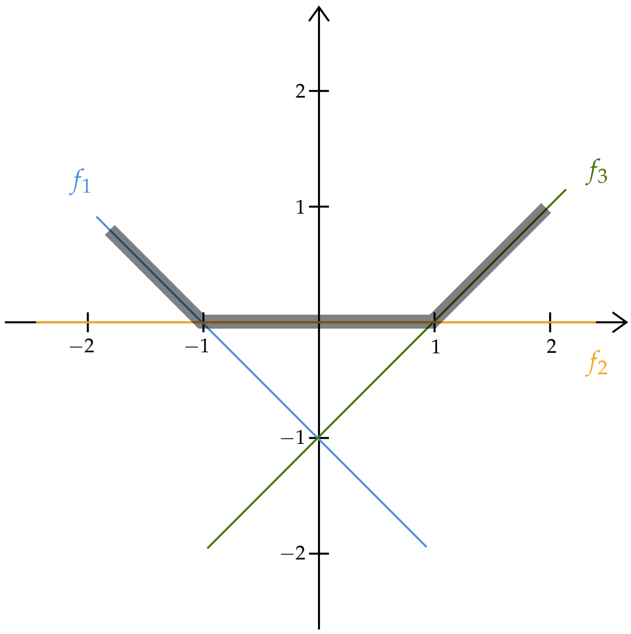
1.2 Minimax

Consider the problem of the form

$$\begin{aligned} &\min_x \max\{f_1(x), \ldots, f_k(x)\} := g(x) \\ &\text{s.t.} \ \cdots \end{aligned}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$.

Example:
$f_1(x) = -x - 1, f_2(x) = 0, f_3(x) = x - 1$. Then $\max\{f_1(x), f_2(x), f_3(x)\}$ is as follows



A motivation

- $\forall i \in [k]$, $f_i(x) =$ completion time for task i.
- Project consists of task $1, \ldots, k$.
- $g(x) =$ completion time of entire project

Note that minimax is not an optimization problem as we defined it. We can revise it as follows

$$\begin{array}{rl} \min & y \\ \text{s.t.} & y \geq f_1(x) \\ & y \geq f_2(x) \\ & \vdots \\ & y \geq f_k(x) \\ & \cdots \end{array}$$

An application minimize a piece-wise linear convex function using linear programming.

2

Flows

digraph

A directed graph (digraph) is a pair (V, E) where

- V is a set of vertices,
- E is a set of ordered pairs of vertices called arcs.

Notation Let $q \in V$, then

$$\delta^+(q) = \{e \in E \mid e \text{ leaves } q\}$$
$$\delta^-(q) = \{e \in E \mid e \text{ arrives at } q\}$$

2.1 Max st-flow model

Given

1. digraph $G = (V, E)$,
2. two vertices $s, t \in V$, and $s \neq t$,
3. $\forall e \in E$, arc e has capacity $u_e \geq 0$.

Now we construct an LP.

For every arc e, we will have a variable x_e, and x_e will be called the **flow** on arc e.

Notation Let $q \in V$:

$$f_x(q) := \sum_{e \in \delta^+(q)} x_e - \sum_{e \in \delta^-(q)} x_e$$

The maximization problem is then

$$\begin{array}{rll} \max & f_x(s) & \\ \text{s.t.} & f_x(q) = 0 & (q \in V, q \neq s, q \neq t) \\ & 0 \leq x_e \leq u_e & (e \in E) \end{array} \tag{P}$$

A feasible solution to (P) is a flow. An optimal solution to (P) is a maximum flow. The value of a flow x is $f_x(S)$.

Remark:
(P) is always feasible. It is not unbounded, so there always exists a maximum flow.

Application computer network. Suppose we have

- computers s and t ($s \neq t$)
- capacity u_e (gb/s) for every link e

The goal is to computer number gb that can be sent from s to t across network. Then x_e is the amount of information across e. $f_x(q) = 0$ means no information lost.

Magic property

If u is integer, then there exists an optimal solution to (P) that is integer.

Remark:
We need the condition "u is integer" so that the property is still true. Also, an optimal solution to (P) is not necessarily integers.

Generalize max st-flows

We can add lower bounds to arcs: ℓ_e $\forall e \in E$

$$\begin{array}{rll} \max & f_x(s) & \\ \text{s.t.} & f_x(q) = 0 & (q \in V, q \neq s, q \neq t) \\ & \ell_e \leq x_e \leq u_e & (e \in E) \end{array} \tag{Q}$$

Magic property - revised

If ℓ, u is integer, and there exists an optimal solution to (Q), then there exists an optimal solution to (Q) that is integer.

Example: Consistent rounding
The goal is to round all entries to nearest up/down integer, so that row sums & column sums still hold.

Any feasible solutions give consistent rounding.

2.2 Min cost flow model

Given

- Digraph $G = (V, E)$
- Capacities $u_e \geq 0$ ($e \in E$)
- Costs c_e ($e \in E$)
- Supply/demands b_q ($q \in V$)

Then the model is

$$\begin{array}{lll} \min & \sum_{e\in E} c_e x_e & \\ \text{s.t.} & f_x(q) = b_q & (q \in V) \\ & 0 \leq x_e \leq u_e & (e \in E) \end{array} \tag{P}$$

Similarly, feasible solution to (P) is a flow. An optimal solution to (P) is a min cost flow.

Magic property - min cost flows

Suppose u, b are integer, then if there exists a min cost flow, then there exists a min cost flow that is integer.

Similarly, we can add a lower bound:

$$\begin{array}{lll} \min & \sum_{e\in E} c_e x_e & \\ \text{s.t.} & f_x(q) = b_q & (q \in V) \\ & \ell_e \leq x_e \leq u_e & (e \in E) \end{array} \tag{P}$$

Magic property - min cost flows (revised)

Suppose u, b, ℓ are integer, then if there exists a min cost flow, then there exists a min cost flow that is integer.

What is a necessary condition for b_q so that there exists a flow? $\sum_{q\in V} b_q = 0$.

Example: Staffing problem

hours	# of employees required
1-2	2
2-3	3
3-4	3
4-5	2

and we have cost of hiring a single employee between

hours	cost
1-5	6
1-4	4
3-5	5
2-4	3

The goal is to minimize cost of hiring employees while meeting staff needs.

3

IP formulations

3.1 IP tricks

Imaging we are forcing variable to take some prescribed set of values: $x \in \{5, 9, 13, 36\}$. Then we can introduce variables $z_1, z_2, z_3, z_4 \in \{0, 1\}$ so that we have two constraints:

$$\begin{aligned} 1 &= z_1 + z_2 + z_3 + z_4 \\ x &= 5z_1 + 9z_2 + 13z_3 + 36z_4 \end{aligned}$$

3.2 Modeling piece-wise linear function

Let $f : \mathbb{R} \to \mathbb{R}$ be piecewise linear. Given $a_1, \ldots, a_k$ and $f_1, \ldots, f_k$ such that $f(a_i) = f_i$. The goal is to write IP constraints with variables x, y such that $y = f(x)$, $x \in \underbrace{[a_1, a_k]}_{\uparrow\ \mathrm{dom}(f)}$.

To generalize,

1. $\lambda_1, \ldots, \lambda_k \geq 0$, $\sum_{i=1}^{k} \lambda_i = 1$
2. $x = \sum_{i=1}^{k} \lambda_i a_i$
3. $y = \sum_{i=1}^{k} \lambda_i f_i$
4. $z_0, \ldots, z_k \in \{0, 1\}$, $z_0 = z_k = 0$
5. $\sum_{i=0}^{k} z_i = 1$
6. $\forall p \in [k]$, $\lambda_p \leq z_{p-1} + z_p$

By (4) and (5), we may assume that $z_p = 1$ and $z_j = 0\ \forall j \neq p$.

Claim λ_p, λ_{p+1} are the only non-zero λ variables.

Proof:
Pick $j \neq p, j \neq p + 1$. As

$$0 \leq \lambda_j \leq z_{j-1} + z_j = 0 + 0 = 0$$

□

With this claim, we can simplify (2) and (3).

3.3 Modeling union of polyhedra

Let

$$P_1 = \{x \mid A^1 x \leq b^1\}$$
$$P_2 = \{x \mid A^2 x \leq b^2\}$$

Goal: write condition: $x \in P_1 \cup P_2$ as part of IP.

Hypothesis: If $x \in P_1 \cup P_2$, then $0 \leq x \leq U$ for some U.

Constraints:

1. $y_1, y_2 \in \{0, 1\}$
2. $y_1 + y_2 = 1$
3. $x = x^1 + x^2$
4. $A^i x^i \leq y_i b^i$, $i = 1, 2$
5. $0 \leq x^i \leq y_i U$, $i = 1, 2$

To show that $x \in P_1 \cup P_2 \iff \exists x^1, x^2, y_1, y_2$ such that all 5 conditions hold.

Proof:
First let's assume (1) - (5) hold. We may assume $y_1 = 1, y_2 = 0$. Then (5) tells us $x^2 = 0$. (3) tells us $x = x^1$. (4) implies

$$A^1 x^1 \leq y_1 b^1 \implies A^1 x \leq b^1 \implies x \in P_1 \subseteq P_1 \cup P_2$$

Now suppose $x \in P_1 \cup P_2$. We may assume $x \in P_1$. Then we set $y_1 = 1, y_2 = 0, x^1 = x, x^2 = 0$. Then we can verify that all 5 conditions hold. $\square$

3.4 Perfect formulations

basis

Let A be a matrix with column indices $\{1, \ldots, n\}$. Then $B \subseteq \{1, \ldots, n\}$ is a basis if

1. A_B square,
2. A_B non-singular.

Remark:
A has a basis $\iff$ rows of A are independent.

basic solution

Let A be a matrix with column indices $1, \ldots, n$. Consider

$$Ax = b \tag{$*$}$$

Pick B as basis of A. Then x is a basic solution of $(*)$ if

1. $Ax = b$.
2. $x_j = 0$ if $j \notin B$.

x is a basic solution for $(*)$ if x is a basic solution for some basis B.

standard equality form

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

and rows of A are independent.

The correctness of simplex algorithm implies the following theorem:

Theorem 3.1

If an LP is SEF has an optimal solution, then it has an optimal solution that is basic.

Let A be matrix, b vector with same number of entries as rows of A. Then $A \leftarrow_j b$ denotes matrix obtained from A by replacing column j by b.

Theorem 3.2: Cramer's rule

Let M be a non-singular matrix and consider $Mx = b$.

$$\bar{x}_j = \frac{\det(M \leftarrow_j b)}{\det(M)} \quad \forall j$$

then $M\bar{x} = b$.

Proposition 3.3

Let M be square matrix with $\det(M) = \pm 1$, and M, b are integer. Then there exists a unique solution to $Mx = b$ is integer.

Proof:
Directly from Cramer's rule. □

Totally unimodular matrix

A matrix A is totally unimodular if every square submatrix N of A, $\det(N) \in \{0, +1, -1\}$.

Proposition 3.4

Let A be TU, b integer. Then every basic solution of $Ax = b$ is integer.

Proof:
Suppose $\bar{x}$ is a basic solution for basis B. If $j \notin B$, $\bar{x}_j = 0$, which is integer. The basic variables $\bar{x}_B$ are the unique solution to $A_B x_B = b$. Since A is TU, then $\det(A_B) \in \{0, -1, +1\}$. Since B is basis, $\det(A_B) \neq 0$. Then $\det(A_B) = \pm 1$. A_B is integer as it is TU. Then by Proposition 3.3, $\bar{x}_B$ is integer. □

Theorem 3.5

If an LP is SEF has an optimal solution, it has an optimal solution that is basic.

Theorem 3.6: main theorem

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax = b, x \geq 0 \end{aligned} \tag{P}$$

Suppose A is TU and b integer. Then if (P) has an optimal solution, it has an optimal solution that is integer.

Proof:
We may assume rows of A are linearly independent. Then (P) is SEF. Thus if (P) has an optimal solution, then by Theorem 3.5, it has an optimal basic solution $\bar{x}$. By Proposition 3.4, every basic solution is integer. In particular, so is $\bar{x}$. □

Constructing TU matrices

Proposition 3.7

Let A be a $0, -1, +1$ matrix such that for every column

1. we have at most one $+1$,
2. we have at most one -1.

Then A is TU.

Proof:
We need to show every $k \times k$ submatrix M of A has $\det(M) \in \{0, -1, +1\}$. We can prove it by induction on k. Discuss by cases.

- M has a column of zeros. Then $\det(M) = 0$.
- Every column has exactly one $+1$ and one -1, then row sum is zero, then rows are linearly dependent, thus $\det(M) = 0$.
- One column has a unique non-zero entry M_{ij}. Applying cofactor expansion,

$$\det(M) = M_{ij}(-1)^{i+j} \det\left(M'_{ij}\right)$$

Then by induction, the statement holds.

□

Constructing TU matrices from TU matrices

Proposition 3.8

Let A be TU. Then

1. permutation of A,
2. A^T,
3. any matrix obtained from A by multiplying rows/columns by -1,
4. any matrix obtained by adding a unit vector to thw rows/columns

are TU.

Theorem 3.9

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax = b, x \geq 0 \end{aligned} \tag{P}$$

Suppose A is TU, b integer. Then if (P) has an optimal solution, then it has one that is integer.

Theorem 3.10

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax = b, 0 \leq x \leq U \end{aligned} \tag{Q}$$

Suppose A is TU, b and U are integer. Then if (Q) has optimal solution, then it has one that is integer.

Proof:
We can rewrite (Q) in SEF:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & \begin{bmatrix} A & 0 \\ I & I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} b \\ U \end{bmatrix} \\ & x, s \geq 0 \end{aligned} \tag{P'}$$

We denote the first constraint by $A'x' = b'$. Now note that b' is integer as b, U are integer. It's clear that A' is TU. Then apply Theorem 3.9. □

3.5 Application to flows

vertex-arc incidence matrix

M is the vertex-arc incidence matrix of digraph $G = (V, E)$ if

1. rows of M correspond to V,
2. columns of M correspond to E,
3. for column uv we have $+1$ for every u, -1 for entry v and 0 otherwise.

Remark:
Let M be vertex-arc incidence matrix of $G = (V, E)$. Pick $v \in V$: for every entry e of row v, if $e \in \delta^+(v)$, then we have $+1$; if $e \in \delta^-(v)$, we have -1, otherwise we have 0.

Magic property - min cost flow

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & f_x(q) = b_q \quad (q \in V) \\ & 0 \leq x \leq U \end{aligned} \tag{P}$$

Suppose b, U integer, and there exists an optimal solution. Then there exists an integer optimal solution.

Proof:
Let A be the vertex-arc matrix.

Claim $f_x(q) = b_q \iff \text{row}_q(A)x = b_q$.

Proof of the claim:
Using the remark, we know that

$$\text{row}_q(A)x = \sum_{e \in \delta^+(q)} x_e - \sum_{e \in \delta^-(q)} x_e := f_x(q)$$

□

We can write (P) as

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & 0 \leq x \leq U \end{aligned}$$

We proved A is TU. Then it follows by Theorem 3.10. □

4

Cone programming

This is generalization of

1. linear programming
2. second-order cone programming
3. semi-definite programming

We have good algorithms for these problems.

cone

$C \subseteq \mathbb{R}^m$ is a cone if

1. C is convex,
2. $\forall \lambda \geq 0$ and $a \in C$, $\lambda a \in C$.

The second condition: C is closed under (non-negative scaling).

pointed cone

A cone is pointed if it does not contain an infinite line.

4.1 Examples of cones

$\mathbb{R}^m_+ = \{u \in \mathbb{R}^m : u \geq 0\}$

Second Order Cone (SOC):

$$L^m = \{u \in \mathbb{R}^m : \|(u_1, \ldots, u_{m-1})\| \leq u_m\}$$

semi-definite matrix

An $n \times n$ matrix M is semi-definite if $\forall x \in \mathbb{R}^n$: $x^T M x \geq 0$.

Semi-definite cone is

$$S^n_+ = \{n \times n \text{ semi-definite matrices } M\}$$

Direct sum

Let $A \subseteq \mathbb{R}^p$, $B \subseteq \mathbb{R}^q$. The direct sum and A and B is

$$A \oplus B := \{(a,b) : a \in A, b \in B\} \subseteq \mathbb{R}^{p+q}$$

Proposition 4.1

Let A, B be cones, then $A \oplus B$ is a cone.

4.2 Cone programming model

Let $c \in \mathbb{R}^n$, A $m \times n$ matrix, $b \in \mathbb{R}^m$, K pointed cone.

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax - b \in K \end{aligned} \tag{P}$$

Cases:

1. $K = \mathbb{R}^m_+$. Then $Ax - b \in \mathbb{R}^m_+ \iff Ax \geq b$. Then it's an LP.
2. $K = L^m$. $Ax - b \in L^m$ can be rewritten as
$$\begin{bmatrix} A' \\ d^T \end{bmatrix} x - \begin{bmatrix} b' \\ p \end{bmatrix} \in L^m$$
$$\|A'x - b\| \leq d^T x - p$$
and this is restricted SOC program.
3. $K = S^n_+$. This is semi-definite program.

Example: Least square problem

$$\min_x \|Ax - b\|$$

or

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & \|Ax - b\| \leq t \end{aligned}$$

which is restricted SOC program

4.3 (General) SOC program

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \|A_i x - b_i\| \leq d_i^T x - p_i \quad (i = 1, \dots, k) \end{aligned}$$

Is this still a cone program? In other words, is there A, B, K such that the above constraint is equivalent to $Ax - b \in K$? Yes. It's direct sum of pointed cones.

5

Robust optimization

$$\begin{aligned} \min \quad & \cdots \\ \text{s.t.} \quad & a^T x \leq \beta \\ & \cdots \end{aligned}$$

This particular constraint might be important. And a is part of data that is given, but a here might have some imprecision. This brings us the uncertainty.

5.1 Model uncertainty for a

Given: $\bar{a} \in \mathbb{R}^n$ estimate, $\epsilon \subseteq \mathbb{R}^n$ uncertainty. True a satisfies $a - \bar{a} \in \epsilon$.

Then we can write the condition

$$a^T x \leq \beta \quad (a - \bar{a} \in \epsilon)$$

We have different choices for ϵ, or think it for different ways.

For example, the ball:

$$a^T x \leq \beta \quad (a - \bar{a} \in \{u : \|u\| \leq r\} := \epsilon_1) \tag{5.1}$$

the cube:

$$a^T x \leq \beta \quad (a - \bar{a} \in \{u : -r \leq u_i \leq r, \forall i\} := \epsilon_2) \tag{5.2}$$

Let

$$\begin{aligned} S_1 &= \{x \mid x \text{ satisfies } (5.1)\} \\ S_2 &= \{x \mid x \text{ satisfies } (5.2)\} \end{aligned}$$

What's the relationship between S_1 and S_2? We know that $\epsilon_1 \subseteq \epsilon_2$. The constraints in (5.1) form a subset of constraints in (5.2). Thus $S_2 \subseteq S_1$.

We will see: (5.1) can be replaced by a single SOC constraint: $\|f(x)\| \leq g(x)$ where f, g affine. (5.2) can be replaced by a finite number of linear constraints (with additional variables).

5.2 Modeling uncertainty by a ball

Given data $r > 0$, $\bar{a} \in \mathbb{R}^n$, $\beta \in \mathbb{R}$. We then have $\epsilon = \{u \in \mathbb{R}^n \mid \|U\| \leq r\}$.

$$a^T x \leq \beta \qquad (a - \bar{a} \in \epsilon) \tag{$*$}$$

As $(*)$ has infinitely many of inequalities, the goal is to replace $(*)$ with a single inequality. The key idea is to fix x. Note that

$$(*) \iff \max\{a^T x : a - \bar{a} \in \epsilon\} \le \beta$$
$$\bar{a}^T x + \max\{a^T x - \bar{a}^T x : a - \bar{a} \in \epsilon\} \le \beta$$
$$\bar{a}^T x + \max\{(a - \bar{a})^T x : a - \bar{a} \in \epsilon\} \le \beta$$
$$\bar{a}^T x + \max\{u^T x : u \in \epsilon\} \le \beta$$

Note that $\max\{u^T x : u \in \epsilon\} = r\|x\|$ given ϵ is a radius r ball. Then

$$\bar{a}^T x + r\|x\| \le \beta$$

and this is exactly SOC constraint.

ellipsoid

An ellipsoid is a set of the form $\{Pu : \|u\| \le 1\}$ and P is a non-singular matrix.

For example, let $P = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$

This is generalization of the ball.

5.3 Modeling uncertainty by a hypercube

Recall the constraint

$$a^T x \le \beta \quad (a - \bar{a} \in \epsilon)$$

where $\epsilon = \{U \mid -r \le v_j \le r, \forall j\}$. We need to show

1. ( ) is equivalent to single constraint, involving absolute values.
2. ( ) is equivalent to finite number of linear constraints.

$$a^T x \ge \beta \quad (a - \bar{a} \in \epsilon)$$

Fix x. Then the above constraint is equivalent to

$$\min\{a^T x \mid a - \bar{a} \in \epsilon\} \ge \beta$$

Then similarly,

$$\bar{a}^T x + \min\{u^T x \mid u \in \epsilon\} \ge \beta$$

Note that assume u is optimal, then

$$\begin{aligned} u^T x &= \sum_j u_j x_j \\ &= \sum_{j:x_j \ge 0} u_j x_j + \sum_{j:x_j < 0} u_j x_j \\ &= \sum_{j:x_j \ge 0} -r x_j + \sum_{j:x_j < 0} r x_j \\ &= -r \sum_j |x_j| \end{aligned}$$

Proposition 5.1

$$a^T x \geq \beta \quad (a - \bar{a} \in \{u \mid -r \leq u_j \leq r, \forall j\})$$

is equivalent to

$$\bar{a}^T x - r \sum_j |x_j| \geq \beta$$

Corollary 5.2

$$a^T x \leq \beta \quad (a - \bar{a} \in \{u \mid -r \leq u_j \leq r, \forall j\})$$

is equivalent to

$$\bar{a}^T x + r \sum_j |x_j| \leq \beta$$

Proposition 5.3

1. $\exists x$ such that $\bar{a}^T x + r \sum_j |x_j| \leq \beta$
2. $\exists x, y$ such that $\bar{a}^T x + r \sum_j y_j \leq \beta$, $y_j \geq x_j$, $y_j \geq -x_j$.

are equivalent.

Proposition 5.4

$$a^T x \leq \beta \quad (a - \bar{a} \in \{u \mid -r \leq u_j \leq r, \forall j\})$$

is equivalent to

$$\bar{a}^T x + r \sum_j y_j \leq \beta$$

$$y_j \geq x_j, y_j \geq -x_j, \ \forall j$$

This can be generalized to be a rectangle. Given $\Delta \geq 0$. Then

$$\epsilon = \{i \mid -\delta_j \leq u_j \leq \delta_j, \ \forall j\}$$

For example, $\Delta = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

Part II:

Interpretations of optimal solutions

6

Duality review