MATHEMATICAL MODELING

Methods and Application

Man VM. Nguyen and Khuong A. Nguyen

MATHEMATICAL MODELING

Methods and Application

Man Van Minh Nguyen(†)

Khuong An Nguyen(‡)

(†) Faculty of Science - Mahidol University

(‡) Faculty of Computer Science & Engineering - HCMC University of Technology

Contents

Index 93

Chapter 4

STOCHASTIC PROGRAMMING

Modeling with Uncertainty in Practice

AN OVERVIEW

Goals : To introduce to learners concepts, ideas and methods to

a) obtain practical experiences in **modeling** linear constraints/objectives with *uncertainty*,

b) study Two-Stage Model and its generalization

c) use a broader class of linear stochastic programming models in **various contexts**,

d) finally propose studying few popular models in R & D

$\star$ Stochastic Linear Programming (SLP) for Evacuation Planning in Disaster Responses

$\star\star$ Polyhedron Theory from ILP to SLP

$\star \star \bullet$ Facility Location problem with Disruptions and Shortages

$\star \star \bullet$ Stochastic Fleet Size and Mix Problem (SFSMP).

Practical and basic expectation: Learners should

1. Know Probabilistic Constraints and Standard form of a SP

2. Formulate stochastic linear program

3. Describe Two-Stage Modesl for many decision variables

4. Comprehend few trendy and key problems in `Smart Modern Urban Management` and employ basic methods requiring stochastic optimization.

Professional Learning Outcomes : We expect that learners should be able to

1. **Define Stages and Decisions** in SP, Wait-and-see vs. Here-and-now [EXAMPLE 4.1]
2. Explain **One-Stage Stochastic linear programming - No recourse** [Section 4.2] by Guess (Risk aversion) * Chance constraints * Recourse actions
3. Describe **Generic Stochastic Programming with RECOURSE** in Section 4.3
4. Formulate Two-stage Stochastic linear programming in Section 4.4
5. Build up a Stochastic LP for Evacuation Planning in Disaster Responses, see Section 4.5
6. (*) To research trainees, you would get the Mathematical Polyhedron Theory in Optimization, from Integer Proramming moving up to Stochastic Programming, starting in Section 4.7.

4.1 Stochastic Programming- Concepts and Objectives

4.1.1 *Programming? What is Stochastic Programming? Uncertainty?*

Programming in mathematics is generally viewed as a collection of computational and mathematical methods used by engineers, investigators, managers, scientists ... to

solve and make decision of **optimization problems**.

- **Mathematical Optimization** is about decision making using mathematical methods.
- **Stochastic Programming** (SP) is about decision making under *uncertainty*.

  View it as 'Mathematical Programming (Optimization) with *random* parameters'.
- *Stochastic linear programs* are linear programs (i.e. its objective function is linear) in which some problem data may be considered uncertain.

♣ **OBSERVATION 1** (Courtesy of Jeff Linderoth, Lehigh Univ.).

1. Randomness is unavoidable in problem formulation (from observed data to proposed model).

Where does randomness (or uncertainty) come from?

Where can we observe it? `Weather Related, Financial Uncertainty`...

Typically, randomness is ignored, or it is dealt with by *Sensitivity analysis*, but for large-scale problems, sensitivity analysis is useless! SP is the way to deal with randomness, **both in objective function and constraints**.

2. **Stochastic or Deterministic Model for Optimization**? Practical problems in S, E & T are not modeled as stochastic or deterministic. Engineers and scientists **determine whether** to model the phenomenon as either stochastic or deterministic based on the problem to be solved.

   Briefly, in *deterministic models*, the model's output is entirely determined by the parameter values and the initial conditions. On the other hand,

   a *stochastic model* is a tool allowing for random variation in one or more inputs over time.

3. **Recourse (Stochastic) programs** are those in which some decisions or recourse (remake, modify) actions can be taken after uncertainty is disclosed.

4.1.2 *Basic concepts, assumptions - Motivation*

Definition 4.1 (Linear program (LP) with random parameters - SLP)

SLP is an interesting, useful, and nice theory! ♣

A Stochastic linear program (SLP) is

$$\text{Minimize } Z = g(\boldsymbol{x}) = f(\boldsymbol{x}) = \boldsymbol{c}^T \cdot \boldsymbol{x}, \quad \text{s.t. } A\,\boldsymbol{x} = \boldsymbol{b}, \text{ and } \quad T\,\boldsymbol{x} \geq h$$

with $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ (decision variables),

certain real matrix A and vector $\boldsymbol{b}$ (for deterministic constraints),

and with random parameters T, h in $T\,\boldsymbol{x} \geq h$ define chance or probabilistic constraints. ■

♦ **EXAMPLE 4.1** (First simple motivation)**.**

We consider the optimization

$$\text{Minimize } z = x_1 + x_2, \quad \text{subject to } x_1 \geq 0, x_2 \geq 0.$$

$$\begin{cases} \omega_1\, x_1 + x_2 \geq 7; \\ \omega_2\, x_1 + x_2 \geq 4. \end{cases}$$

where parameters ω_1, ω_2 be uniform (random) variables following distributions

Uniform(a, b), precisely $\omega_1 \sim$ Uniform$(1, 4)$, $\quad \omega_2 \sim$ Uniform$(1/3, 1)$.

- When both $\omega_1 = \omega_2 = 1$ then the two conditons becomes

  $x_1 + x_2 = 4$ making the red line, and $x_1 + x_2 = 7$ making the dotted blue line,

  you obviously obtain the feasible region fully contaning the green arrow (Figure 4.1).

♣ **QUESTION.** How do we solve this problem if ω_1, ω_2 really are uniform (random) variables?

- **What do we mean by solving this problem**?

1. The **wait-and-see** approach: Suppose it is possible to decide about the decision variables $\mathbf{x} = [x_1, x_2]$ after the observation of the random vector $\boldsymbol{\omega} = [\omega_1, \omega_2]$? [partially representing for *data uncertainty* of the problem.] Can we solve the problem without waiting?

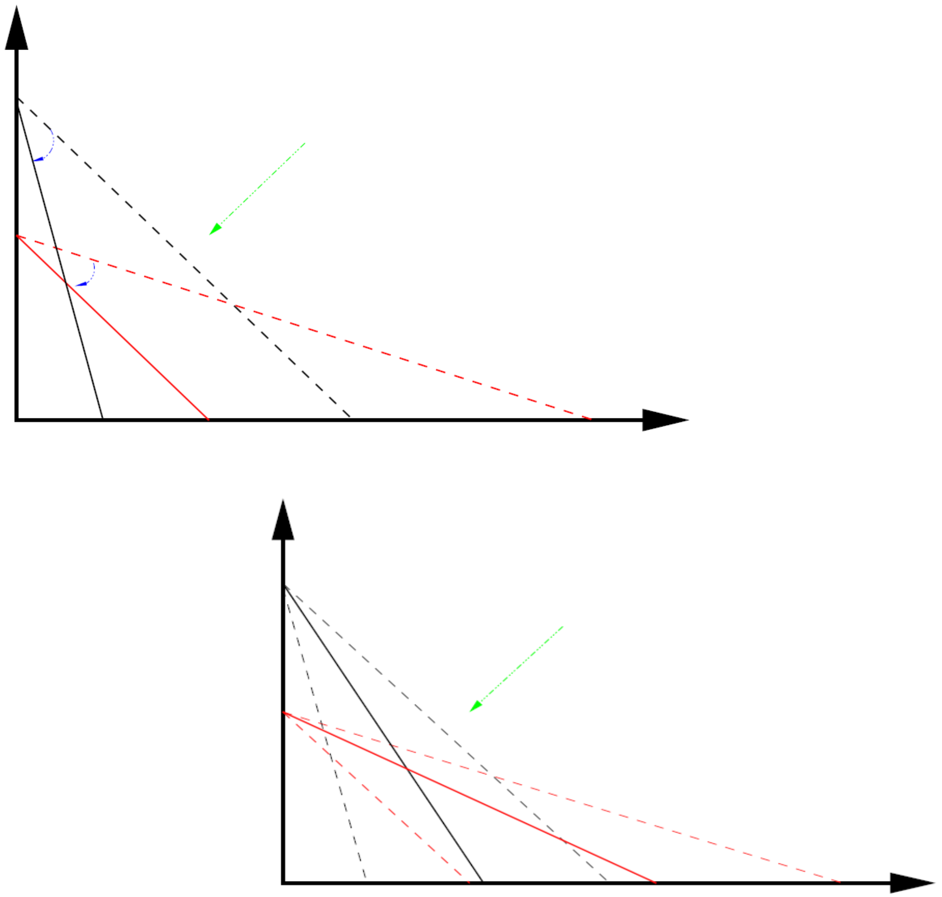


Figure 4.1: *Simple SP with two decision variables*

2. Yes, we can solve the problem but **no waiting**, i.e. we need to decide on $\mathbf{x} = [x_1, x_2]$ **before knowing** the values of $\boldsymbol{\omega} = [\omega_1, \omega_2]$?. We need to decide what to do about not knowing $\boldsymbol{\omega}$.

We suggest (a) Guess at uncertainty, and (b) Probabilistic Constraints (see Definition 4.1).

(a) Guess at uncertainty We will guess few reasonable values for $\boldsymbol{\omega}$. Three (reasonable) suggestions

– each of which tells us something about our level of 'risk'

- `Unbiased`: Choose mean values for each random parameters $\boldsymbol{\omega}$
- `Pessimistic`: Choose worst case values for $\boldsymbol{\omega}$
- `Optimistic`: Choose best case values for $\boldsymbol{\omega}$.

E.g, use **Unbiased method**, look at EXAMPLE 4.1, uniform distributions Uniform(a, b) clearly have mean $(a+b)/2$, so $\widehat{\boldsymbol{\omega}} = (\frac{5}{2}, \frac{2}{3})$. Our program

$$\text{Minimize } z = x_1 + x_2$$

has opt value $z_1 = \dfrac{50}{11}$ at point $\widehat{\mathbf{x}} = (\widehat{x}_1, \widehat{x}_2) = 18/11, 32/11$

if subject to

$$\begin{cases} \dfrac{5}{2}\, x_1 + x_2 \geq 7; \\ \dfrac{2}{3}\, x_1 + x_2 \geq 4; \\ x_1 \geq 0, \qquad\qquad x_2 \geq 0. \end{cases}$$

How about `Pessimistic` and `Optimistic` ways?

The Pessimistic and Optimistic solutions of the program

Minimize $z = x_1 + x_2$ respectively are

- the Pessimistic with $\widehat{\boldsymbol{\omega}} = [1,\ \dfrac{1}{3}]$

$$\begin{cases} \mathbf{1}\ x_1 + x_2 \geq 7;\ \ \mathbf{1/3}\ x_1 + x_2 \geq 4 \\ x_1 \geq 0,\ \ x_2 \geq 0 \end{cases}$$

$\Longrightarrow z_2 = 7$ at point $\widehat{\mathbf{x}} = (\widehat{x}_1, \widehat{x}_2) = (0, 7)$

- the Optimistic $\widehat{\boldsymbol{\omega}} = (4,\ 1)$

$$\begin{cases} \mathbf{4}\ x_1 + x_2 \geq 7; \\ \mathbf{1}\ x_1 + x_2 \geq 4... \end{cases}$$

$\Longrightarrow z_3 =?$ at point $\widehat{\mathbf{x}} = (\widehat{x}_1, \widehat{x}_2) =?$

PRACTICE 4.1. *Use soft* `MATLAB` *or suitable solvers (see Section 4.6) or direct logical argument to check the Pessimistic and Optimistic solutions.*

4.1.3 *Stochastic Programming (SP) Claims and Facts*

- Lots of application areas (Finance, Energy, Telecommunication) and it is truly Mature field (since Dantzig G. work in 1955)
- Variety of SP problem classes with specialized solution algorithms (e.g. Bender's Decomposition)
- Compared to deterministic mathematical programming (MP) it takes small fraction!

4.2 One-Stage Stochastic linear programming - No recourse

We now start with One-Stage SLP or 1-SLP, shortly means stochastic LP with one-stage, no recourse actions or specifically without penalize corrective actions.

Definition 4.2 (SLP with one-stage (No recourse) : 1-SLP)

Consider the following program $LP(\boldsymbol{\alpha})$ that is parameterized by the random vector $\boldsymbol{\alpha}$:

$$\text{Minimize } Z = g(\boldsymbol{x}) = f(\boldsymbol{x}) = \boldsymbol{c}^T \cdot \boldsymbol{x} = \sum_{j=1}^{n} c_j\, x_j$$

$$\text{s. t. } \; A\,\boldsymbol{x} = \boldsymbol{b}, \quad \text{(certain constraints)}$$

$$\text{and } \quad T\,\boldsymbol{x} \geq h \quad \text{(stochastic constraints)}$$

♣

with assumptions that

1. matrix $T = T(\boldsymbol{\alpha})$ and (vector) $h = h(\boldsymbol{\alpha})$ express uncertainty via stochastic constraints

$$T(\boldsymbol{\alpha})\,\boldsymbol{x} \geq h(\boldsymbol{\alpha}) \iff \alpha_1\, x_1 + \cdots + \alpha_n\, x_n \geq h(\boldsymbol{\alpha})$$

2. values (T, h) not known: they are unknown before an instance of model occurs, $h(\boldsymbol{\alpha})$ depends only on random α_j;

3. **uncertainty** is expressed by probability distribution of random parameters $(\alpha_j) = \boldsymbol{\alpha}$ so deterministic LP is the degenerate case of Stochastic LP when α_j are constant,

- We deal with decision problems where the vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in \mathcal{X}$ of decision variables must be made before the realization of parameter vector $\boldsymbol{\alpha} \in \Omega$ is known.
- Often we set lower and upper bounds for $\boldsymbol{x}$ via a domain $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$.

APPROACHES: In Stochastic Programming we utilize some assumption and facts.

Fundamental assumption- We know a (joint) probability distribution of data. Hence the first approach gives Probabilistic (Chance) Constraint LP.

The Scenario Analysis- not perfect, but useful, is the second approach. The scenario approach assumes that there are a finite number of decisions that nature can make (outcomes of randomness).

Each of these possible decisions is called a **scenario**.

4.2.1 APPROACH 1 : use Chance constraint and Acceptable risk

- We can replace $T\,\boldsymbol{x} \geq h$ by probabilistic constraints $\mathbb{P}[T\,\boldsymbol{x} \geq h] \geq p$ [1]

  for some prescribed reliability level $p \in (.5, 1)$, (to be determined by problem owner.)

  The LP in Definition (4.2) above with random parameters $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots]$ then is called **Probabilistic Constraint LP**, or just **1-SLP**.

- Risk then is taken care of explicitly, if define an

$$\text{acceptable risk} \quad r_x := \mathbb{P}[Not\ (T\,\boldsymbol{x} \geq h)] = \mathbb{P}[T\,\boldsymbol{x} \leq h] \leq 1 - p$$

  then $(1-p)$ is `maximal acceptable risk`.

  The chance constraint $T\,\boldsymbol{x} \leq h$ implies that

  the acceptable risk r_x is less than a specified maximal $1 - p \in (0, 1)$.

[1] or also replace $T\,\boldsymbol{x} \leq h$ by with $\mathbb{P}[T\,\boldsymbol{x} \leq h] \leq 1 - p$

Definition 4.3

*Stochastic LP or **1-SLP with Probabilistic Constraints** is defined by*

a random coefficients $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ *in chance constraints,*

and a linear objective $f(\boldsymbol{x})$*:*

$$SP: \quad \min_{\boldsymbol{x}} Z, \qquad Z = f(\boldsymbol{x}) = \boldsymbol{c}^T \cdot \boldsymbol{x} = \sum_{j=1}^{n} c_j\, x_j, \quad c_j \in \mathbb{R} \tag{4.1}$$

s. t.
$$\begin{cases} A\,\boldsymbol{x} = \boldsymbol{b} \qquad (\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \text{ makes decision variables}) \\ \mathbb{P}[T\,\boldsymbol{x} = \boldsymbol{\alpha} \cdot \boldsymbol{x} \leq h] \leq (1-p) \quad (0 < p < 1). \end{cases}$$

♣

NOTE: we use parameter vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots]$ in general, and

denote $\boldsymbol{\omega} = [\omega_1, \omega_2, \cdots, \omega_S]$ specifically for states s called *scenarios*. We treat each scenario $\omega \in \boldsymbol{\omega}$ possibly by a combination of many random parameters α_i at once in a SP.

4.2.2 APPROACH 2: for stochastic constraints $T(\boldsymbol{\alpha})\,\boldsymbol{x} \leq h(\boldsymbol{\alpha})$

Use Scenario analysis of $T(\boldsymbol{\alpha})\,\boldsymbol{x} \leq h(\boldsymbol{\alpha})$

For every scenario $(T^s; h^s),\ s = 1, \ldots, S$, solve

$$\text{Minimize } \{f(\boldsymbol{x}) = \boldsymbol{c}^T \cdot \boldsymbol{x}; \quad \text{s.t.} \quad A\,\boldsymbol{x} = \boldsymbol{b},\ T^s\,\boldsymbol{x} \leq h^s\ \}$$

This kind of program targets a specific linear objective while accounting for a probability function associated with **various scenarios**. Hence, we find an overall solution by looking at the scenario solutions $\boldsymbol{x}^s$ $(s = 1, \ldots, S)$.

Advantage: Each scenario problem is an LP. / vs / **Disadvantage**: discrete distribution $\longrightarrow$ mixed-integer LP model. (In general: possibly non-convex model).

♦ **EXAMPLE 4.2.** Consider the following LP with $n = 4$ decision variables $\boldsymbol{x} = (x_1, x_2, x_3, x_4)'$,

$$P_0 : \text{Min } Z = g(\boldsymbol{x}) = -x_2$$

and stochastic constraints in form $T(\boldsymbol{\alpha})\, \boldsymbol{x} \leq h(\boldsymbol{\alpha}) \longleftrightarrow T(\boldsymbol{\alpha})\, \boldsymbol{x} + W = h(\boldsymbol{\alpha})$.

We may start with a `discrete distribution` of parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$, satisfying

$\alpha_1 = 1,\ \alpha_2 = 3/4$ with density 0.25; and $\alpha_1 = -3,\ \alpha_2 = 5/4$ with density 0.75.

Here with random $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ we has only $S = 2$ scenarios

$\omega_1 = (\alpha_1 = 1,\ \alpha_2 = 3/4)$ with density 0.25; $\omega_2 = (\alpha_1 = -3,\ \alpha_2 = 5/4)$ with density 0.75.

Put $T(\boldsymbol{\alpha}) = [\alpha_1,\ \alpha_2, 0, 1],\ h(\boldsymbol{\alpha}) = 2 + \alpha_1$, and let $x_1 \geq 2,$ and $x_j \geq 0$ for, $j \in \{1, 2, 3, 4\}$

our program specifically is

$$\begin{cases} x_1 + x_2 + x_3 = 3 & \text{(as certain constraints), and} \\ \alpha_1\, x_1 + \ \alpha_2\, x_2 + \ x_4 = 2 + \alpha_1 & \text{(stochastic constraints)} \end{cases}$$

This means that there are two possible equations (or scenarios ω_1, ω_2):

either the second constraint is $x_1 + \dfrac{3}{4}x_2 + x_4 = 3$ (scenario occurs with probability 0.25),

or it is $-3x_1 + \dfrac{5}{4}x_2 + x_4 = -1$ (scenario occurs with probability 0.75).

4.3 Generic Stochastic Programming (GSP) with RECOURSE

PREVIEW of Generic Stochastic Programming (GSP) with RECOURSE

The above examples motivate the generic stochastic problem (GSP).

- We have a set of decisions taken without full information on some random events. These decisions are called **first-stage decisions** and are usually represented by a vector $\boldsymbol{x}$. Later, full information is received on the realization of some random vector $\boldsymbol{\alpha}$.
- Then, the second-stage or **corrective actions** $\boldsymbol{y}$ are taken, so we represent the 2nd-stage decisions by vector $\boldsymbol{y}$. Write **2-SP problem** for two-stage stochastic program which allow infeasibilities w.r.t. random constraints $T \cdot \boldsymbol{x} \leq h$, $T \cdot \boldsymbol{x} + W \cdot \boldsymbol{y} = h$ or equivalently $W \cdot \boldsymbol{y} = h - T \cdot \boldsymbol{x}$. We correct infeasibilities later, and must pay for corrections via $\boldsymbol{y}$.
- Both $\boldsymbol{\alpha}$ and $\boldsymbol{y}$ may use functional forms, as $\boldsymbol{\alpha}(\omega)$, $\boldsymbol{y}(\omega)$, or $\boldsymbol{y}(s)$, to show explicit dependence on a scenario ω or an outcome s of the `random experiment` **e** coupled with the 2-SP modeling.

We focus on modeling and leave out details if not essential for understanding concepts.

Definition 4.4 (Stochastic program in two stages (generic 2-SP problem))

The two-stage stochastic program (2-SP) extended from Definition 4.2 has the form

$$2 - SP: \quad \min_{\boldsymbol{x}} \; g(\boldsymbol{x}) \quad \text{with } g(\boldsymbol{x}) = f(\boldsymbol{x}) \; + \; \mathbf{E}_{\omega}[v(\boldsymbol{x}, \boldsymbol{\omega})] \tag{4.2}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ *is the first stage decision variables,*

$f(\boldsymbol{x})$ *can be linear or not, a part of the* ***grand objective*** *function* $g(\boldsymbol{x})$.

* *The mean* $Q(\boldsymbol{x}) := \mathbf{E}_{\omega}[v(\boldsymbol{x}, \boldsymbol{\omega})]$ *of a function*

$$v : \mathbb{R}^n \times \mathbb{R}^S \to \mathbb{R}$$

upon influences of scenarios $\boldsymbol{\omega}$. $Q(\boldsymbol{x})$ *is the optimal value of a certain second-stage problem*

$$\min_{\boldsymbol{y} \in \mathbb{R}^p} \boldsymbol{q} \; \boldsymbol{y} \; \Big| \; \text{subject to} \quad T \cdot \boldsymbol{x} + W \cdot \boldsymbol{y} = h. \tag{4.3}$$

♣

Vectors $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\omega)$ and $\boldsymbol{y} = \boldsymbol{y}(\omega)$ are named `correction, tuning or recourse` decision variables, only known after the experiment **e**.

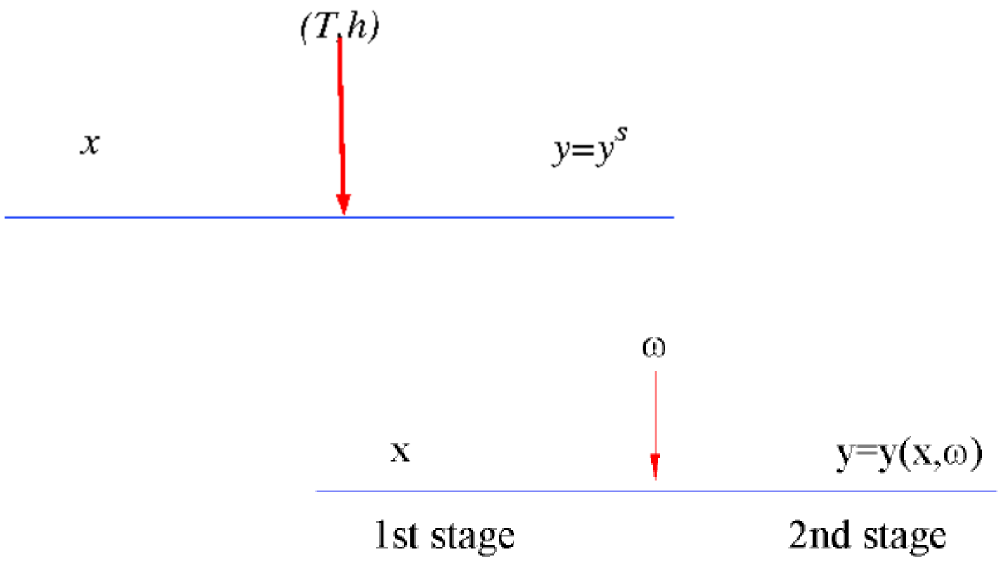


Figure 4.2: *Standard view of two-stage stochastic program*
Courtesy of Maarten van der Vlerk, Univ. of Groningen, NL

Briefly we **Minimize total expected costs** $g(\boldsymbol{x}) = f(\boldsymbol{x}) \; + \; Q(\mathbf{x})$ while satisfying

$$W \cdot \boldsymbol{y}(\boldsymbol{\omega}) = h(\boldsymbol{\omega}) - T(\boldsymbol{\omega}) \cdot \; \boldsymbol{x}$$

Here W is called $m \times p$ recourse matrix, and we begin with simple case of $m = 1$,

$\boldsymbol{q}$ is the unit recourse cost vector, having the same dimension as $\boldsymbol{y}$, and $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{\omega}) \in \mathbb{R}^p$. ■

ELUCIDATION (On **Recourse modeling issues**)

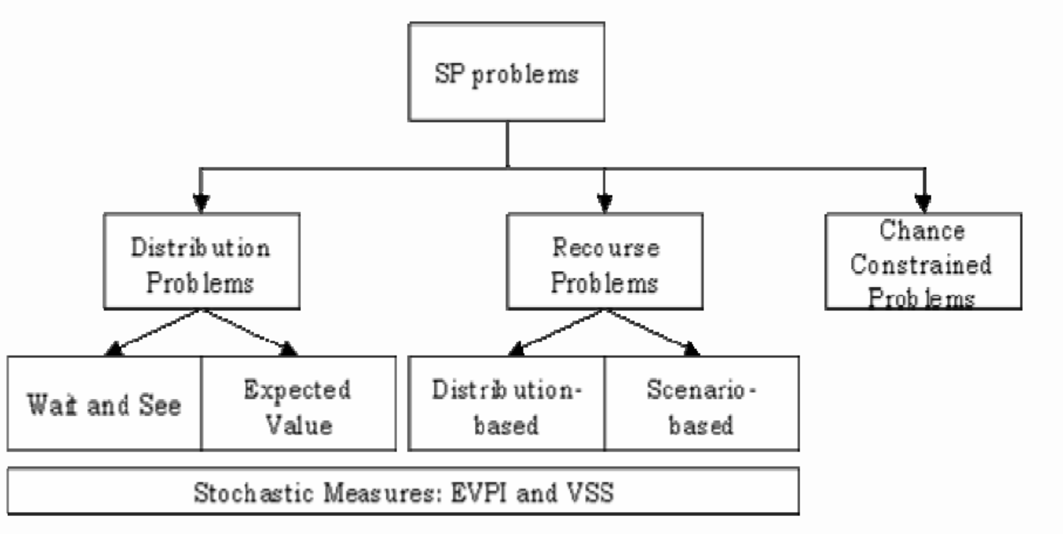
- Our grand objective $g(\boldsymbol{x})$ is built up by $f(\boldsymbol{x})$ and $Q(\mathbf{x})$. Here $\boldsymbol{y}$ is the decision vector of a second-stage LP problem, value $\boldsymbol{y}$ depends on the realization of $(T, h) := (T(\boldsymbol{\omega}), h(\boldsymbol{\omega}))$. Recourse variables $\boldsymbol{y}(\boldsymbol{\omega}) \sim$ corrective actions e.g. use of alternative production resources (overtime...)
- Quantitative risk measure: size of deviations $h(\boldsymbol{\omega}) - T(\boldsymbol{\omega}) \cdot \boldsymbol{x}$ is relevant.
- Here RISK is described by *expected recourse costs* $Q(\mathbf{x})$ of the decision $\mathbf{x}$.
- Model reformulation in fact is needed: Where do $\boldsymbol{q}$ and W come from?

♦ **EXAMPLE 4.3. [Industry- Manufacturing.]**: Production planning where production levels: in **1st stage**, inventories, backlogs: in **2nd stage**.

[Agriculture.]: The **1st stage** in Agriculture and Argo-industry corresponds to planting and occurs during the whole spring (in North Semi-sphere).

The **2nd-stage** decisions consist of sales and purchases. The `random experiment` **e** includes selling and purchasing. Selling extra *corn* would probably occur very soon after the harvest while buying missing corn will take place as late as possible. ∎

Key Stochastic Programming classes are briefed in Fig 4.3



Some Stochastic Programming Classes

(Courtesy of Michael R. Bussieck, GAMS Software GmbH)

Figure 4.3: *Stochastic Programming classes*

Most popular SLP is the simple **one-stage** (No Recourse) and two-stage (with Recourse). One-stage stochastic LP (1-SLP) does **not involve** with the random part $v(\mathbf{x}, \boldsymbol{\omega})$ in Section 4.2.

4.4 Two-stage Stochastic linear program: 2-SLP

We now treat the **two-stage** stochastic LP with recourse action.

4.4.1 *Two-stage SLP Recourse model - (simple form)*

Definition 4.5 (Two-stage Stochastic LP With Recourse : 2-SLPWR)

The Two-stage Stochastic linear program With Recourse (2-SLPWR) or precisely with penalize corrective action generally described as

$$2 - SLP : \quad \min_{\boldsymbol{x} \in \mathbf{X}} \boldsymbol{c}^T \cdot \boldsymbol{x} + \min_{\boldsymbol{y}(\boldsymbol{\omega}) \in \mathbf{Y}} \mathbf{E}_\omega[\boldsymbol{q} \ \cdot \boldsymbol{y}]$$

or in general

$$2 - SLP : \quad \min_{\boldsymbol{x} \in \mathbf{X},\ \boldsymbol{y}(\boldsymbol{\omega}) \in \mathbf{Y}} \mathbf{E}_\omega[\boldsymbol{c}^T \cdot \boldsymbol{x} + v(\boldsymbol{x}, \boldsymbol{\omega})] \tag{4.4}$$

with $v(\boldsymbol{x}, \boldsymbol{\omega}) := \boldsymbol{q} \ \cdot \boldsymbol{y}$ ♣

subject to

$$A\,\boldsymbol{x} = \boldsymbol{b} \qquad \text{First Stage Constraints ,} \tag{4.5}$$

$$T(\boldsymbol{\omega}) \cdot \ \boldsymbol{x} + \ W \cdot \boldsymbol{y}(\boldsymbol{\omega}) = h(\boldsymbol{\omega}) \ \ \text{Second Stage Constraints} \tag{4.6}$$

$$\text{or shortly} \quad W \cdot \boldsymbol{y} = h(\boldsymbol{\omega}) - T(\boldsymbol{\omega}) \cdot \ \boldsymbol{x} \tag{4.7}$$

♦ This SLP program specify the above 2-SP (4.2) to the target - a specific *random grand objective* (function) $g(\boldsymbol{x})$ having

(1) the *deterministic* $f(\boldsymbol{x})$- being linear function, while accounting

(2) for a probability function $v(\mathbf{x}, \boldsymbol{\omega})$ associated with various scenarios $\boldsymbol{\omega}$.

♦ $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x}, \omega) \in \mathbb{R}^p_+$ is named recourse action variable for decision $\boldsymbol{x}$ and realization of $\boldsymbol{\omega}$.

Recourse actions are viewed as **Penalize corrective** actions in SLP.

The Penalize correction is expressed via the mean $Q(\mathbf{x}) = \mathbf{E}_\omega[v(\mathbf{x}, \boldsymbol{\omega})]$. HOW to FIND IT?

Major Approaches- APPROACH 2: Scenarios analysis again

To solve system (4.4-4.6) numerically, approaches are based on a random vector $\boldsymbol{\alpha}$ having a finite number of possible realizations, called *scenarios*.

Expected value $Q(\mathbf{x})$ obviously for a discrete distribution of $\boldsymbol{\omega}$!

So we take $\Omega = \{\omega_k\}$ be a finite set of size S (there are a finite number of scenarios $\omega_1, \ldots, \omega_S \in \Omega$, with respective probability masses p_k).

Since $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x}, \omega)$ so the expectation of $v(\boldsymbol{y}) = v(\mathbf{x}, \boldsymbol{\omega}) := q \cdot \boldsymbol{y}$ (one cost q for all y_k) is

$$Q(\mathbf{x}) = \mathbf{E}_{\omega}[v(\mathbf{x}, \boldsymbol{\omega})] = \sum_{k=1}^{S} p_k \, q \, y_k = \sum_{k=1}^{S} p_k \, v(\mathbf{x}, \omega_k) \tag{4.8}$$

where

- p_k is the density of scenario ω_k, q is single unit penalty cost,
- and $q \; y_k = v(\mathbf{x}, \omega_k)$ - the penalty cost of using y_k units in correction phase, depends on both the first-stage decision $\mathbf{x}$ and random scenarios ω_k.

4.4.2 *Two-stage SLP Recourse model - (canonical form)*

We now fully characterize the system (4.4-4.6) in the linear case.

Definition 4.6 (Stochastic linear program With Recourse action (2-SLPWR))

The canonical 2-stage **stochastic linear** program with Recourse can be formulated as ♣

$$2 - SLP: \quad \min_{\boldsymbol{x}} \; g(\boldsymbol{x}) \text{ with}$$

$$g(\boldsymbol{x}) := \; \boldsymbol{c}^T \cdot \boldsymbol{x} + v(\boldsymbol{y}) \tag{4.9}$$

$$\text{subject to (s. t.)} \quad A\,\boldsymbol{x} = \boldsymbol{b} \;\text{ where }\; \boldsymbol{x} \in \mathbf{X} \subset \mathbb{R}^n, \boldsymbol{x} \geq \mathbf{0} \tag{4.10}$$

$$v(\boldsymbol{z}) := \min_{\boldsymbol{y} \in \mathbb{R}^p_+} \quad \boldsymbol{q}\;\boldsymbol{y} \quad \text{subject to} \quad W \cdot \boldsymbol{y} = h(\boldsymbol{\omega}) - T(\boldsymbol{\omega}) \cdot \; \boldsymbol{x} =: \boldsymbol{z} \tag{4.11}$$

where $v(\boldsymbol{y}) := v(\mathbf{x}, \boldsymbol{\omega})$ is the second-stage value function, and

$\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x}, \omega) \in \mathbb{R}^p_+$ is a recourse action for decision $\boldsymbol{x}$ and realization of $\boldsymbol{\omega}$.

1. The expected recourse costs of the decision $\mathbf{x}$ is $Q(\mathbf{x}) \; := \; \mathbf{E}_\omega[v(\mathbf{x}, \boldsymbol{\omega})]$ by Equation (4.8).

[precisely expected costs of the recourse $\boldsymbol{y}(\boldsymbol{\alpha})$, for any policy $\mathbf{x} \in \mathbb{R}^n$.] Hence overally we minimize total expected costs $\min_{\boldsymbol{x}\in\mathbb{R}^n,\ \boldsymbol{y}\in\mathbb{R}^p_+} \quad \boldsymbol{c}^T \cdot \boldsymbol{x} + Q(\mathbf{x})$.

2. We design the 2nd decision variables $\boldsymbol{y}(\boldsymbol{\omega})$ so that we can (tune, modify, or) react to our original constraints (4.10) in an intelligent (or optimal) way: we call it **recourse** action!

$$\mathbf{x} - - - - - - - - - -T, h, \boldsymbol{\omega} - - - - - - - - > \boldsymbol{y}.$$

3. The optimal value of the 2nd-stage LP is $v_* = v(\boldsymbol{y}^*)$, with $\boldsymbol{y}^* = \boldsymbol{y}^*(\boldsymbol{x}, \omega)$ is its optimal solution, here $\boldsymbol{y}^* \in \mathbb{R}^p_+$. The total optimal value is $\boldsymbol{c}^T \cdot \boldsymbol{x}^* + v(\boldsymbol{y}^*)$. ■

SUMMARY (Canonical 2-SLPWR problem)

(I) In Definition 4.6, at the first stage we have to make a "here-and-now" decision $\mathbf{x}$ **before** the realization of the uncertain data, viewed as a random vector $\boldsymbol{\omega}$ with values ω_k, is known.

The considered 2-SLPWR is **linear** because $\boldsymbol{c}^T \cdot \boldsymbol{x}$, $Q(\mathbf{x})$ and the constraints in the 2nd-stage LP problem (4.11) are linear.

(II) In the 2nd stage, W is a $m \times p$ matrix, called the **recourse matrix**. Right now, (and in nearly all problems we will see), we set $m = 1$ (simple recourse)- only one recourse cost W for every scenario u^s in constraints $T^s x + W u^s = h^s$ for all $s = 1.2.\ldots.S$. The canonical model is

$$\min \left\{ cx + Q(x) : Ax = b,\ x \in \mathbb{R}^n_+ \right\} \quad \text{with}$$

$$Q(x) = \mathbb{E}_\omega \left[v(h(\omega) - T(\omega)x) \right], \qquad v(z) = \min \left\{ qy : Wy = z,\ y \in \mathbb{R}^p_+ \right\}$$

Spec. case: $\Omega = \{\omega^1, \ldots, \omega^S\}$, prob. p_s; $\quad (T(\omega^s), h(\omega^s)) = (T^s, h^s)$

$$\begin{array}{llcccccccc}
\min & cx & + & p_1 \cdot qy^1 & + & \ldots & + & p_S \cdot qy^S & & \\
\text{s.t.} & Ax & & & & & & & = & b \\
 & T^1 x & + & Wy^1 & & & & & = & h^1 \\
 & \vdots & & & & \ddots & & & & \vdots \\
 & T^S x & & & & & + & Wy^S & = & h^S \\
 & x \geq 0 & & y^1 \geq 0 & & & & y^S \geq 0 & &
\end{array}$$

Large-scale LP, size $(m_1 + mS) \times (n + pS)$, with special structure

Figure 4.4: *Courtesy of Maarten van der Vlerk, Univ. of Groningen, NL*

E.g. demand $h := \boldsymbol{D}(\boldsymbol{\omega})$ of raw material in **[Industry- Manufacturing.]**: after a realization of $\boldsymbol{\omega}$

becomes available, we optimize our behavior by solving the above canonical optimization. ■

4.4.3 EXAMPLES

♦ **EXAMPLE 4.4.** **[Industry- Manufacturing.]**- *One decision variable and one scenario*

Consider now the **case of manufacturing** when the decision $\boldsymbol{x} = x$ (ordering raw material)

should be made before a realization of the demand D becomes known.

One possible way to proceed in such a situation is to

(i) view **the demand** as a random variable $\boldsymbol{\omega} = \omega =: D$,

and (ii) assume that **the probability distribution** of D is known.

The mean $\mathbf{E}_\omega\big[v(\boldsymbol{x}, \boldsymbol{\omega})\big]$ becomes $\mathbf{E}_\omega[v(\boldsymbol{x}, d)] = Q(\boldsymbol{x})$: viewed as a function $Q(\boldsymbol{x})$ of the quantity $\boldsymbol{x}$ only, after a scalar realization d of D is known. Here $n = 1 = S$. ■

PROBLEM 4.1. Consider an industrial firm where a manufacturer produces n products.

There are in total m different parts (sub-assemblies) to be ordered from 3rd-party suppliers.

A unit of product i requires $a_{ij} \geq 0$ units of part j , where $i = 1, \ldots, n$ and $j = 1, \ldots, m$. The demand for the products is modeled as a random vector $\boldsymbol{\omega} = \mathbf{D} = (D_1, D_2, \ldots, D_n)$.

The second-stage problem:

For an observed value (a realization) $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$ of the above random demand vector **D**, we can find the best production plan by solving the following stochastic linear program (SLP) with decision variables $\boldsymbol{z} = (z_1, z_2, \cdots, z_n)$ - the number of units produced,

and **other** decision variables $\boldsymbol{y} = (y_1, y_2, \cdots, y_m)$ - the number of parts left in inventory

$$LSP: \quad \min_{\boldsymbol{z},\boldsymbol{y}} Z = \sum_{i=1}^{n} (l_i - q_i)\, z_i - \sum_{j=1}^{m} s_j y_j, \tag{4.12}$$

where $s_j < b_j$ (defined as pre-order cost per unit of part j), and

$x_j,\ j = 1, \ldots, m$ are the numbers of parts to be ordered before production.

$$\text{subject to} \begin{cases} y_j = x_j - \sum_{i=1}^{n} a_{ij}\, z_i, \quad j = 1, \ldots, m \\ 0 \leq z_i \leq d_i, \quad i = 1, \ldots, n; \quad y_j \geq 0,\ j = 1, \ldots, m. \end{cases}$$

The whole model (of the second-stage) can be equivalently expressed as

$$\begin{cases} \min_{\boldsymbol{z},\boldsymbol{y}}\ Z = \ \boldsymbol{c}^T \cdot \boldsymbol{z} - \boldsymbol{s}^T \cdot \boldsymbol{y} \\ \text{with} \quad \boldsymbol{c} = (c_i := l_i - q_i) \text{ are cost coefficients} \\ \boldsymbol{y} = \boldsymbol{x} - A^T\, \boldsymbol{z}, \ \text{ where } A = [a_{ij}] \ \text{ is matrix of dimension } n \times m, \\ 0 \leq \boldsymbol{z} \leq \boldsymbol{d}, \quad \boldsymbol{y} \geq 0. \end{cases} \tag{4.13}$$

Observe that the solution of this problem, that is, the vectors $\boldsymbol{z}, \boldsymbol{y}$ depend on realization $\boldsymbol{d}$ of the random demand $\boldsymbol{\omega} = \boldsymbol{D}$ as well as on the 1st-stage decision $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)$.

The first-stage problem: The whole 2-SLPWR model is based on a popular rule that

production $\geq$ **demand**.

Now follow distribution-based approach, we let $Q(\boldsymbol{x}) := \mathbf{E}[Z(\boldsymbol{z}, \boldsymbol{y})] = \mathbf{E}_\omega[\boldsymbol{x}, \boldsymbol{\omega}]$ denote the optimal value of problem (4.12). Denote

$\boldsymbol{b} = (b_1, b_2, \cdots, b_m)$ built by preorder cost b_j per unit of part j (before the demand is known). The quantities x_j are determined from the following optimization problem

$$\min g(\mathbf{x}, \boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{b}^T \cdot \boldsymbol{x} + Q(\boldsymbol{x}) = \boldsymbol{b}^T \cdot \boldsymbol{x} + \mathbf{E}[Z(\boldsymbol{z})] \tag{4.14}$$

where $Q(\boldsymbol{x}) = \mathbf{E}_\omega[Z] = \sum_{i=1}^{n} p_i \, c_i \, z_i$ is taken w. r. t. the probability distribution of $\boldsymbol{\omega} = \boldsymbol{D}$.

The first part of the objective function represents the **pre-ordering cost** and $\boldsymbol{x}$. In contrast, the second part represents the **expected cost** of the optimal production plan (4.13), given by the updated ordered quantities $\boldsymbol{z}$, already employing random demand $\boldsymbol{D} = \boldsymbol{d}$ with their densities.

ELUCIDATION

- Decision variables include vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$, and also $\boldsymbol{z} \in \mathbb{R}^n$.
- After the demand D is observed, the manufacturer may decide which portion of the demand is to be satisfied so that the available numbers of parts are **not** exceeded. It costs additionally l_i to satisfy a unit of demand for product i, and the unit selling price of this product is q_i.
- After the demand D becomes known, we determine how much of each product to make. The parts not used are assessed salvage values s_j, giving vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$. ▷▷▷

SUMMARY

1. Problem (4.12)–(4.14) is an example of a **two-stage stochastic programming** problem, where (4.12) is called the *second-stage* problem and (4.14) is called the *first-stage* problem. As (4.12) contains random demand $\boldsymbol{D}$, its optimal value $Q(\boldsymbol{x}, \boldsymbol{d})$ is a random variable.

2. The 1st-stage decisions x should be made before a realization of the random data D becomes available and hence should **be independent** of the random data. The x variables are often referred to as `here-and-now` decisions.
3. The second-stage decision variables z and y in (4.12) are made after observing the random data and are functions of the data d. They are referred to as `wait-and-see` decisions (solution).
4. The problem (4.12) is feasible for every possible realization of the random data d; for example, take $z = 0$ and $y = x$.

4.5 Stochastic Linear Program for Evacuation Planning in Disaster Responses (SLP-EPDR)

AIM - MOTIVATION

We want to utilize two-stage SLP models to evacuate the affected people to safe areas during disaster response. The case study is based on researches of Li Wang [2] and Esra Koca [3].

- **Extreme natural disasters** (earthquakes, storms, fire, hurricanes, etc.) and unnatural ones (terrorist attacks, political issues, war, etc.) around the globe may strike a community with little warning and leave much damage and many casualties.
- The main goal of **emergency response** is to provide shelter and assistance to affected people as soon as possible. The optimal evacuation plan for affected people is one of the dominant

[2] A two-stage stochastic programming framework for evacuation planning in disaster responses, **Journal of Computers & Industrial Engineering**, vol 145, 2020 Elsevier;

[3] Two-stage stochastic facility location problem with disruptions and restricted shortages **Journal of Computers & Industrial Engineering**, vol 183, 2023 Elsevier.

components in emergency response after a disaster, and lots of scholars have denote their efforts into this interesting problem.

4.5.1 *BACKGROUND - OPEN ISSUSES*

A two-stage stochastic scenario-based programming model should be proposed to evacuate affected people in disaster areas. The first-stage decisions are the robust and reliable evacuation plan for all levels of disaster. The second-stage decisions involve the evacuation plan for affected people in response to specific scenario-based road conditions

We use a set of discrete scenarios to represents potential magnitudes of the disaster, which tries to formulate a model that combines **pre-event emergency evacuation plan**

with scenario-based evacuation plan for affected people after an event.

Specifically, a part of transportation roads may be destroyed during the event, causing stochastic travel times and capacities when traveling on the road. In other words, non-anticipative first-stage decisions are made in the advance of realization of uncertainty.

The 2nd-stage decisions (recourse), which are conditional on the 1st-stage decisions, are made after the realization of stochastic travel times and capacities. Therefore, the objective is to make the optimal pre-event evacuation plan in the first stage, which is under uncertainty conditions to be faced in the 2nd stage.

APPROACHES

The stochastic programming with recourse (**Dantzig**, [4]) is very popular method for dealing with randomness of factors, and this method is to find non-anticipative decisions that have to be made before knowing the realizations of random variables. According to the number of stages, the stochastic programming with recourse problem is generally referred to

as two-stage/multi-stage stochastic programming.

[4] Dantzig, G. (1955), Linear Programming under Uncertainty. Management Science, 1, 197–206.

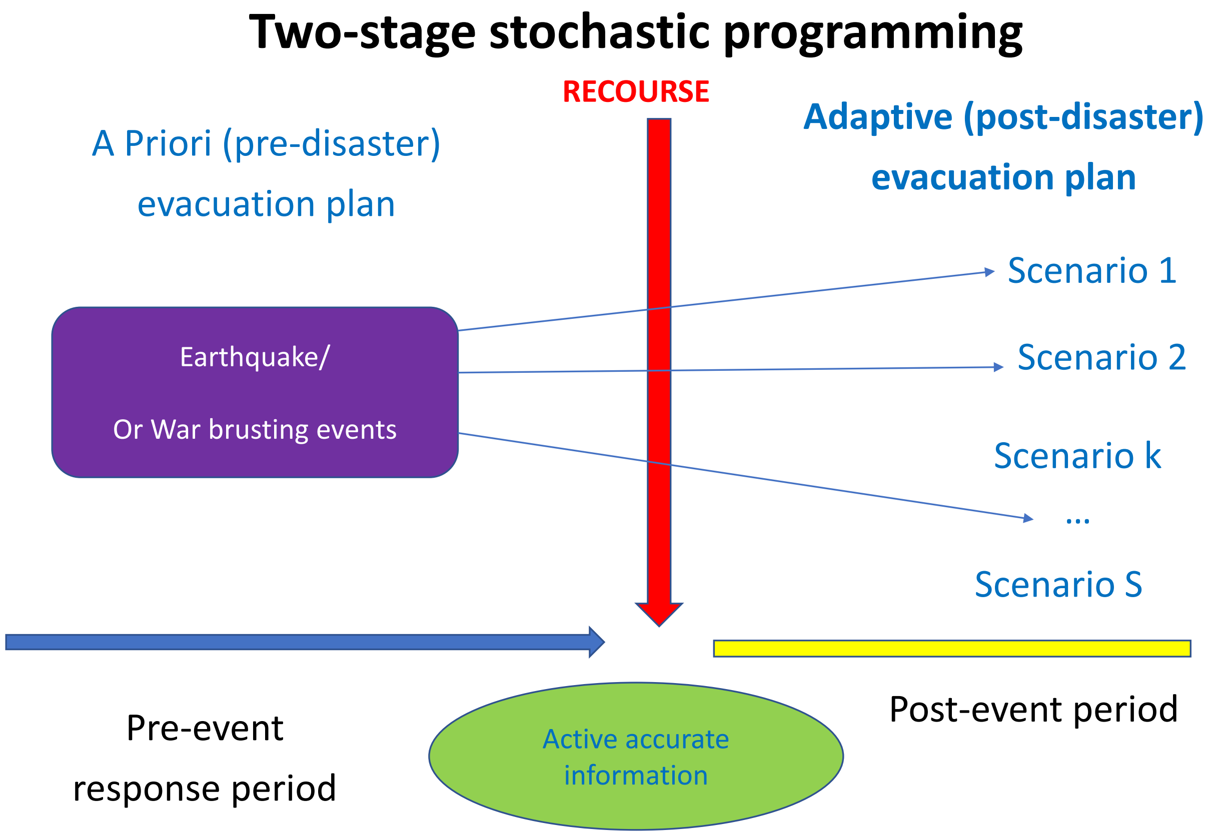


Figure 4.5: *An Illustratiion for Occurrence of Earthquake or War brusting events*

4.5.2 *ANALYSIS- Problem statements*

Representation of the evacuation problem

As a 2-SLP (two-stage stochastic programming)

We need to characterize the **evacuation process** under certain assumptions, and to make sure that **the evacuation phase** should be divided into two stages according to the acquisition time of accurate information. Briefly, the objective is to make

the optimal evacuation planning in the 1st stage under uncertainty to be faced in the 2nd stage.

Conservation laws should be defined for graph with Multiple sources and sinks.

4.5.3 *MODEL FORMULATION To ALGORITHMIC SOLUTION*

We have to properly define and discuss decision variables, system constraints and the objective function with relevant notations used in the mathematical formulation.

HINT: use Table 1, 2 of

Ref. 1 = Li Wang, *A two-stage stochastic programming framework for evacuation planning in disaster responses*, Journal of Computers & Industrial Engineering, vol 145, 2020 **Elsevier**

Two-stage stochastic evacuation planning model

The evacuation's objective is to obtain

(1) a **robust** evacuation plan in the first stage by

(2) the evaluation of **adaptive** evacuation plans in the second stage.

We should evaluate the evacuation plan of the first stage with

the **expected overall evacuation time** of each scenario's adaptive evacuation path, and

the probability of occurrence of each scenario s is assumed as $p_s = \mu_s, \; s = 1, 2, \ldots, S$.

- The teams might employ model (9) and its equivalent models [in **Ref. 1**]-

  they are called time-dependent and stochastic two-stage evacuation planning models.
- Make sure that you fully explain system constraints and the objective function of models.

MODELING APPROACHES for MM-HCMUT-2023 Assignment

In this assignment we focus on the strategy of combining

a priori (pre-disaster) and adaptive (post-disaster) path selection, which can be achieved by the `two-stage stochastic programming`, to determine the evacuation plan for affected people, either upon earthquake event or war brusting events.

The teams of at most 5 students of HCMUT should

1. Understand PROBLEM 4.1 [produce n products satisfying **production** $\geq$ **demand**] via a numerical instance.
2. Employ the two-stage stochastic programming model that considers both a **priori** (pre-disaster) and **adaptive** (post-disaster) path selection to provide a priori evacuation plan for the affected people from dangerous areas to safe areas.
3. Formulate the explicit movement process of affected people when a disaster occurs, this paper proposed a min-cost flow model based on two-stage stochastic programming

4.5.4 QUESTIONS for ASSIGNMENT 2023

STANDARD WORK- two tasks for HCMUT MM intake 2023:

1. **To PROBLEM 4.1** [produce n products satisfying **production** $\geq$ **demand**]. (4 points)

   Use the 2-SLPWR model given in Equations 4.13 and 4.14 when $n = 8$ products, the number of scenarios $S = 2$ with density $p_s = 1/2$, the number of parts to be ordered before production $m = 5$, we randomly simulate data vector $\boldsymbol{b}$, $\boldsymbol{l}$, $\boldsymbol{q}$, $\boldsymbol{s}$ and matrix A of size $n \times m$.

   We also assume that the random demand vector $\boldsymbol{\omega} = \mathbf{D} = (D_1, D_2, \ldots, D_n)$ where each ω_i with density p_i follows the binomial distribution $\mathbf{Bin}(10, 1/2)$.

   REQUEST: build up the nummerical models of Equations 4.13 and 4.14 with simulated data. Find the optimal solution $\mathbf{x}, \boldsymbol{y} \in \mathbb{R}^m$, and $\boldsymbol{z} \in \mathbb{R}^n$ by suitable soft (as GAMSPy)

2. **To the SLP-EPDR: Algorithmic Solutions** (6 points)

Few Solution algorithms are given in Section 4 of **Ref. 1**, and this year 20023 CSE - HCMUT students may try only the first approach (Algorithm 1) based on Min-cost Flow Problem.

Learn, implement and verify the effectiveness of the studied Algorithm or solving the two-stage stochastic evacuation planning model on a small grid network only, with Experiment Design approach. Precisely the simulated would have max of 50 nodes and 100 links ([in Section 5.1 of **Ref. 1**]).

ADVANCED WORK for MASTER level: Extend the SLP-EPDR (from 2024).

A) Carry out STANDARD WORK above.

B) Make your own comments and suggestion in the DISCUSSION part of your report.

Though the proposed model was an NP-hard problem, even without considering the behavior of affected people yet, few developments should be thought of for future senior projects with open issues. If time allows, propose some ideas if you want to combine the viewpoint of decision-makers, with the choice behavior of affected people in the evacuation process.

C) Extend the method of 2-SLPWR to

Stochastic Fleet Size and Mix Problem, see Section 4.8, or

Facility Location problem with Disruptions and Shortages, see REF. 2= Koca [5].

[5] Two-stage stochastic facility location problem with disruptions and restricted shortages **Journal of Computers & Industrial Engineering**, vol 183, 2023 Elsevier.

4.6 *Software For Stochastic Optimization*

Reference links

`riverlogic.com/?blog=software-requirements-for-stochastic-optimization`

`www.stoprog.org/resources`

With MaTLAB you may try `https://github.com/kul-optec/risk-averse`.

`softwaresim.com/blog/the-best-programming-languages-for-stochastic-modeling-and-analysis/`

Reference text

Stochastic Programming Codes, Chapter 1, 2 and 6 of the text

Applications of Stochastic Programming / edited by Stein W. Wallace and William T. Ziemba (SIAM-MPS 2005).

4.6.1 *Soft and programming languages For Stochastic Optimization*

Few well-known programming languages for stochastic modeling and analysis are

1. **GAMS/DECIS**: GAMS stands for General Algebraic Modeling Language, and is one of the most widely used modeling languages.

   DECIS is a system for solving *large-scale stochastic programs*, i.e. programs that include parameters (coefficients and right-hand sides) that are **not known** with certainty, but are assumed to be known by their probability distribution. It employs Benders decomposition and advanced *Monte Carlo sampling* techniques.

   DECIS includes a variety of solution strategies, such as solving the universe problem (all scenarios), the expected value problem, Monte Carlo sampling within the Benders decomposition algorithm.

   see `https://www.gams.com/latest/docs/S-DECIS.html`

   and `https://infanger.com/software/decis.html`

   * **GAMSPy =** GAMS + Python, `https://gamspy.readthedocs.io/en/latest/user/index.html`

2. **Stochastic Modeling Interface** (SMI),

   get from link `https://github.com/coin-or/Smi`, works on Linux OS

3. `R` : We can use `ARIMA` command in `R` to develop stochastic models.

   The background of `ARIMA` is complex, but you can try **? arima** in command line section to learn practical usages via manual.

   Also we must combine 3 soft WEKA, Rapid Miner and `R` in once to deal with SLP practically.

   Rapid Miner for academic use is available at `https://rapidminer.com/platform/educational/`

   A good research text in SHM is *Using R, WEKA and RapidMiner in Time Series Analysis of Sensor Data for Structural Health Monitoring*

   H. Kosorus, Jürgen Hönigl, J. Küng 22nd International Workshop... 29 August 2011 Computer Science International Workshop on Database and Expert Systems Applications, link

   `https://ieeexplore.ieee.org/document/6059835`

4.6.2 *Software Requirements For Stochastic Optimization*

Here are some key requirements for Stochastic Optimization, and SLP particularly.

1. Deterministic and stochastic together in the same model
2. End user selects distribution function:

   Stochastic definitions, including the specification of the probability distribution function and parameters, should be entirely data driven. The end user should be free to select the required distribution amongst a list of choices.
3. Coefficients and constraints can be made stochastic

   End users eventually ask for stochastic definition support for something not currently in their model, called **out-of-the-box support**. Stochastic variability to all types of data should be sought out that typically include:
   - Costs, including materials, labor
   - Input distributions and Output yields

- Process rates and downtime factors
- Minimum and maximum limits ...

4. Distribution functions and parameters defined at variable level.

   Stochastic definitions should be flexible enough to allow definition for each variable explicitly. A data-driven approach lets the end user assign a different distribution function to each and every coefficient, if necessary.

5. Individual stochastic definitions can be temporarily enabled/disabled. [Compare with Item 1.]
6. Visualizations to identify stochastic definitions should be taken into account.
7. Data checks identify conflicting data:

   Look for an optimization platform with built-in data checking. Generating stochastic values, like those used for variable constraints, can easily cause data errors. The best optimization modeling platforms include a large library of built-in data checks, which execute between the time the random values are generated and when the model is solved.

4.7 Mathematical Polyhedron Theory for ILP

Two parts

B/ Polyhedral Stochastic LP

Aimed for graduate level from 2024.

A/ Polyhedron Theory for ILP

Aimed for undergraduate students.

We firstly set a background of Mathematical Polyhedron Theory for ILP.

We study a few subjects here:

- The Warehouse or Facility Location problem with solution
- How to write a **good formulation** of an optimization problem?
  Illustration via a simplified version of the Facility Location problem
- Brief **Theory of Polyhedron** in Section 4.7.3
- **Linear algebra conditions** for solution's existence
- On **extreme points** of a polyhedron

- **Representation** of bounded polyhedral sets, in Section 4.7.6
- **Mixed** Integer Optimization
- Modeling of real world problems with LP and ILP

4.7.1 The Warehouse or Facility Location problem

To model Warehouse (factory) Location, decisions must be made about trade-offs between the costs for operating distribution centers, and the transportation costs [6].

A manager needs to decide which of n warehouses to use for meeting the demands of m customers for a specific good. The decisions to be made are

a/ **which warehouses to operate**, [related to the **cost of operation** there]

b/ **how much to ship** (*shipping cost*) from any warehouse to any customer (or client).

We define a simplified version of the **Facility Location** as follows.

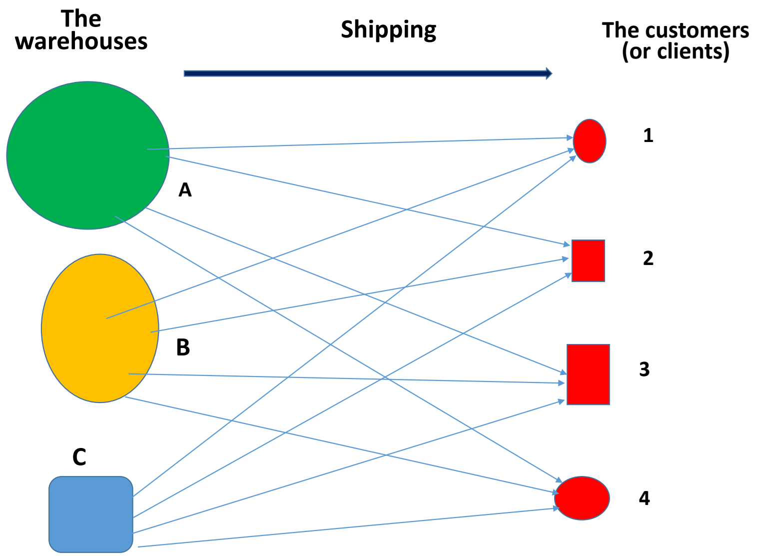
O) Data: Suppose

- $M = \{1, 2, 3, \ldots, m\}$: warehouse (facility) locations
- $I = \{1, 2, 3, \ldots, n\}$: set of clients

The cost f_i of facility placed at i,

and c_{ij} the transportation cost for shipping goods from facility i to customer j.

[6] see Section H. on **Transportation models** with integer variables



Here $m = 3$ warehouses, and $n = 4$ customers.

I) Decision variables: Decision variables into two groups:

Integer variables

$$y_i = \begin{cases} 1, & \text{if a facility or warehouse is placed at location } i \\ & \text{(if warehouse } i \text{ is opened)} \\ 0, & \text{otherwise;} \end{cases}$$

Real variables: For each pair of facility - client (i, j) define

$x_{ij} =$ the amount of demand of client j satisfied by facility i (equivalently an amount of goods to be sent from warehouse i to customer j).

II/ The Costs: Two type of costs are

- **Operating cost at warehouses**: The Type A variable y_i associates with the cost f_i - the fixed operating cost for facility i.

- **Cost related to goods**: For each pair of warehouse - customer (i, j)
  - the variable x_{ij} associates with
  - the cost c_{ij}- it is per unit operating cost at warehouse i plus (but we put out) the transportation cost for shipping goods from warehouse i to customer j.

III/ Constraints: There are two types

i) the demand d_j of each customer j must be filled from the warehouses; and

ii) goods can be shipped from a warehouse i only if it is opened (show by y_i).

IV) The model is

$$\textbf{IZ} = \min_{\boldsymbol{y},\boldsymbol{x}} \quad \sum_{i}^{m} f_i\, y_i + \sum_{i}^{m} \sum_{j}^{n} c_{ij} x_{ij}$$

subject to:

$$\sum_{i=1}^{m} x_{ij} = d_j \quad (\forall\, j = 1, 2, \ldots, n) \tag{4.15}$$

$$\sum_{j=1}^{n} x_{ij} \le y_i \Big(\sum_{j=1}^{n} d_j \Big) \quad (\forall\, i = 1, 2, \ldots, m)$$

$$y_i \in \{0, 1\}, \ (\forall\, i = 1, 2, \ldots, m) \tag{4.16}$$

$$x_{ij} \ge 0 \qquad (\forall\, i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n).$$

NOTATION: We denote the point-wise multiplication

$$\boldsymbol{c} \bullet \boldsymbol{x} = c^T \cdot \boldsymbol{x} = \sum_{i}^{m} \sum_{j}^{n} c_{ij} x_{ij},$$

and the inner product

$$\boldsymbol{f}^T \cdot \boldsymbol{y} = \sum_{i}^{m} f_i\, y_i.$$

Then our Facility Location problem finally can be described by

Mixed Integer Linear Program- MILP

The objective function becomes

IZ $= \quad \min_{(\boldsymbol{y},\boldsymbol{x})\in P} \quad \boldsymbol{f}^T \cdot \boldsymbol{y} + \boldsymbol{c} \bullet \boldsymbol{x}$ where $P = \Big\{ (\boldsymbol{y}, \boldsymbol{x}) : \boldsymbol{y} \in \{0,1\}^m, \boldsymbol{x} \in \mathbb{R}_+^{m\times n},$

($\mathbb{R}_+^{m\times n} = \mathbb{R}_+^m \times \mathbb{R}_+^n$ the Cartesian product of two sets),

$$\text{meaning} \quad y_i \in \{0,1\},\ x_{ij} \geq 0;$$

subject to: $$\sum_{i=1}^{m} x_{ij} = d_j,$$

$$\sum_{j=1}^{n} x_{ij} - y_i \left(\sum_{j=1}^{n} d_j \right) \leq 0$$

$\forall\, i = 1, 2, \ldots, m,$ (the warehouses),

$\forall\, j = 1, 2, \ldots, n$ (the customers) $\Big\}$.

4.7.2 What is a good formulation? And How to?

If we scale the client's demand to 1 (100%) then $d_j = 1 (\forall\, j = 1, 2, \ldots, n)$, and x_{ij} now is called the fraction of demand of client j satisfied by facility i.

The above model becomes

$$\mathbf{IZ}_2 = \min_{\boldsymbol{y},\boldsymbol{x}} \quad \sum_i^m f_i\, y_i + \sum_i^m \sum_j^n c_{ij} x_{ij}$$

subject to new constraints $\sum_{i=1}^m x_{ij} = 1 \quad (\forall\, j = 1, 2, \dots, n)$ and the well-known

$$\begin{cases} x_{ij} \le \ y_i \quad (\forall\, i = 1, 2, \dots, m) \\ y_i \in \{0, 1\}, \ (\forall\, i = 1, 2, \dots, m) \quad \text{binary?} \\ 0 \le x_{ij} \le 1 \qquad (\forall\, i = 1, 2, \dots, m,\ j = 1, 2, \dots, n). \end{cases} \tag{4.17}$$

Question: Are the models pure Integer LP? Which one is preferable?

- **IZ** $=$ **IZ**$_2$, since the integer points both formulations define are the same. Also computing them is hard. But their constraints give feasible sets, with relaxation:

$$P_1 = \{(\boldsymbol{y}, \boldsymbol{x}) : \sum_i^m x_{ij} = 1, \quad \text{with } x_{ij} \le y_i, \quad \underline{0 \le y_i,\ x_{ij} \le 1}, \}$$

$$\subseteq \ P_2 = \{(\boldsymbol{y}, \boldsymbol{x}) : \sum_i^m x_{ij} = 1, \sum_j^n x_{ij} \le n.y_i, \ \text{ with } \underline{0 \le y_i,\ x_{ij} \le 1}.\}$$

- However, let rewrite

$$Z_1 = \min_{(\boldsymbol{y},\boldsymbol{x})\in P_1} \quad \boldsymbol{f}^T \cdot \boldsymbol{y} + \boldsymbol{c} \bullet \boldsymbol{x}$$

$$Z_2 = \min_{(\boldsymbol{y},\boldsymbol{x})\in P_2} \quad \boldsymbol{f}^T \cdot \boldsymbol{y} + \boldsymbol{c} \bullet \boldsymbol{x}$$

then we see that $Z_2 \leq Z_1 \leq \mathbf{IZ}_1 = \mathbf{IZ}_2$.

♣ OBSERVATION 2.

It follows that

1. Finding $\mathbf{IZ}_1 (= \mathbf{IZ}_2)$ is not easy, but solving to find Z_2, Z_1 is Linear OP.
   Since Z_1 is closer to $\mathbf{IZ}_1$ several methods (like branch and bound) would work better.
2. One question is: suppose that if we solve $\min \quad \boldsymbol{f}^T \cdot \boldsymbol{y} + \boldsymbol{c} \bullet \boldsymbol{x}, \ (\boldsymbol{y}, \boldsymbol{x}) \in P_1$ we find an integral solution.
   Have we solved the facility location problem?
3. Formulation 1 is better than Formulation 2.

Question 1.

How to write a good formulation?

Denote by P a linear relaxation for a problem, then $P \subset \mathbb{R}^{m+mn} = \mathbb{R}^m \times \mathbb{R}^{mn}$. Let

$$H = \left\{(\boldsymbol{y}, \boldsymbol{x}) : \boldsymbol{y} \in \{0,1\}^m\right\} \bigcap P.$$

We define the Convex Hull of H (written $\mathbf{conv}(H)$ or $\mathbf{CH}(H)$) to be

$$\mathbf{CH}(H) = \left\{\boldsymbol{h} = (\boldsymbol{y}, \boldsymbol{x}) : \boldsymbol{h} = \sum_i \lambda_i \, \boldsymbol{h}^i, \text{ where } \sum_i \lambda_i = 1,\ \lambda_i \geq 0,\ \boldsymbol{h}^i \in H\right\}. \tag{4.18}$$

We see that $\mathbf{CH}(H) \subset \mathbb{R}^m \times \mathbb{R}^{mn}$ and that the extreme points of $\mathbf{CH}(H)$ have $\{0,1\}$ coordinates for the $\boldsymbol{y}$ part.

So, if we know $\mathbf{CH}(H)$ explicitly, then by solving

$$\min \quad \boldsymbol{f}^T \cdot \boldsymbol{y} + \boldsymbol{c} \bullet \boldsymbol{x}, \quad \boldsymbol{h} = (\boldsymbol{y}, \boldsymbol{x}) \in \mathbf{CH}(H)$$

we solve the problem. Besides, quality of formulation is judged by closeness to $CH(H)$:

$$CH(H) \subseteq P_1 \subseteq P_2.$$

SUMMARY 1.

- An IO formulation is better than another one if the **polyhedra of their linear relaxations** are closer to the convex hull of the IO (than that of the original IO).
- Modeling with binary variables allows a lot of modeling power. However, a good formulation may have an exponential number of constraints.
- *Conjecture* (Dimitris Bertsimas, MIT):

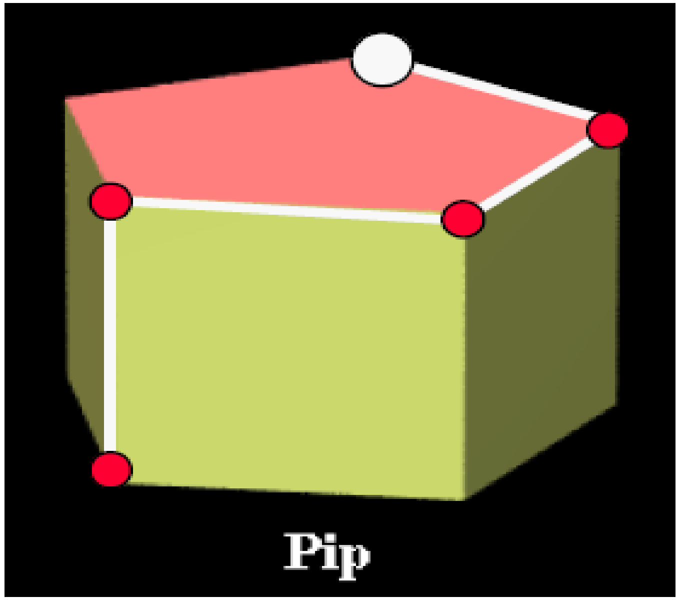
Formulations characterize the complexity of problems. If a problem is solvable in polynomial time, then the convex hull of solutions is known.

4.7.3 Basic Polyhedral concepts in High-dimension Geometry

- In 3D, any **plane** has the form $ax + by + cz + d = 0$ where $a, b, c \in \mathbb{R}$, and $a^2 + b^2 + c^2 \neq 0$.
- In general n-**D** space, planes are called **hyper-planes**, that are defined by linear equality constraints of the form $\boldsymbol{a}'\boldsymbol{x} = b = -a_0$, with $\boldsymbol{a}' = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^n$,, explicitly

$$a_1x_1 + a_2x_2 + \ldots + a_{n-1}x_{n-1} + a_nx_n + a_0 = 0, \ \text{s.t.} \ \sum_{i=1}^{n} a_i^2 \neq 0.$$

- A **polyhedron** P in n-D space (generalized from polygon in 2D, 3D) is a *bounded set* formed by at least $n + 1$ un-parallel hyper-planes [meaning linearly independent].
- An **edge** of a polyhedron $P \subset \mathbb{R}^n$ is the set of solutions of a *linear equations* formed by the intersection of $(n - 1)$ linearly independent defining hyperplanes.



In 3D space, a linear constraint with equality gives a **plane**, and two un-parallel planes cutting each other give us a **line**.

A **polygon** is a bounded solid body made by at least 4 planes. An **edge** is a bounded line, and an extreme point (at the corner of the polygon) is the intersection of 2 lines. In other words, each **extreme point** of a convex set is the intersection of 3 planes.

Figure 4.6: *Geometric view of the feasible region*

Definition 4.7 (Valid inequalities and faces) ♣

We say

1. $\boldsymbol{a}'\boldsymbol{x} \geq b$ is called a valid inequality for a set P if it is satisfied by all points in P.
2. Let $\boldsymbol{f}'\boldsymbol{x} \geq g$ be a valid inequality for a polyhedron P, and let

$$F = \{\boldsymbol{x} \in P : \boldsymbol{f}'\boldsymbol{x} = g\}.$$

   Then F is called a face of P and we say that $\boldsymbol{f}'\boldsymbol{x} \geq g$ represents F.
   A face is called **proper** if $F \neq \emptyset$, and $F \neq P$.
3. A face F of P represented by the inequality $\boldsymbol{f}'\boldsymbol{x} \geq g$ is called a **facet** of P if $\dim(F) = \dim(P) - 1$. The inequality $\boldsymbol{f}'\boldsymbol{x} \geq g$ is named **facet defining**.

Fact 4.1.

For each **facet** F of P, at least one of the inequalities representing F is necessary in any description of P. Besides, every inequality representing a face of P of dimension less than $\dim P - 1$ is not necessary in the description of P, and can be dropped.

■ **CONCEPT 1.**

Vectors $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^k \in \mathbb{R}^n$ are *affinely independent* if the unique solution of the linear system

$$\sum_i^k a_i \boldsymbol{x}^i = 0, \text{ and } \sum_i^k a_i = 0,$$

is $a_i = 0$ for all $i = 1, \ldots, k$.

It is known that the vectors $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^k \in \mathbb{R}^n$ are affinely independent **if and only if** the vectors $\boldsymbol{x}^2 - \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k - \boldsymbol{x}^1$ are linearly independent.

Definition 4.8 (*Dimension of a polyhedron*)

*Let $P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\,\boldsymbol{x} \geq \boldsymbol{b}\}$. Then P has **dimension** k, denoted $\dim P = k$ if the maximum number of affinely independent points in P is $k+1$.* ♣

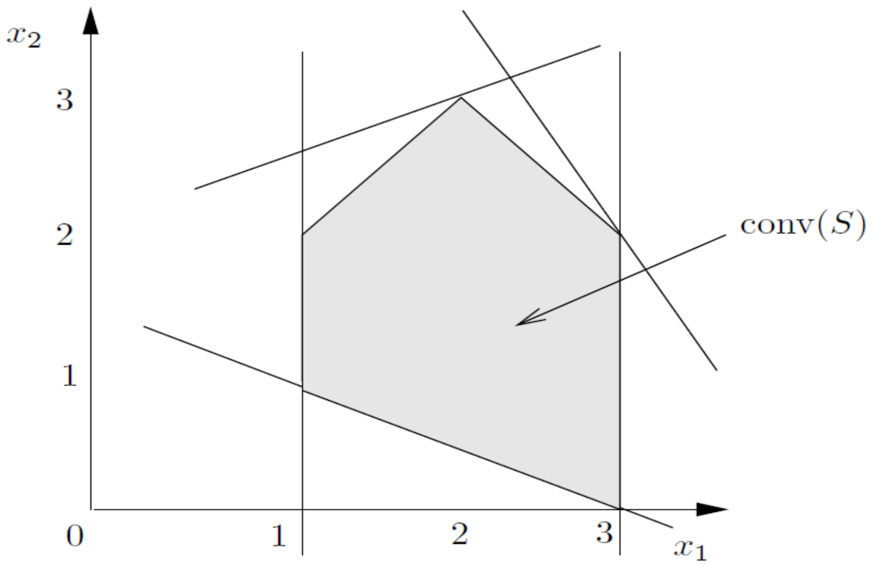
Example 4.1 Find $\dim P =?$ if $P = \{\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2 : x_1 - x_2 = 0, 0 \leq x_1, x_2 \leq 1\}$.

Example 4.2 Let

$$S = \{(x_1, x_2) \in \mathbb{Z}^2 : \ x_1 \leq 3, x_1 \geq 1, \ -x_1 + 2x_2 \leq 4, \ 2x_1 + x_2 \leq 8, \ x_1 + 2x_2 \geq 3\}$$

be a polygon. Draw it in 2D-plane, and write down the equations of faces of dimension one.

Faces of dimension one: $-x_1 + 2x_2 \leq 4$, and $2x_1 + x_2 \leq 8$.



A polyhedron in 2D with facets, and its convex hull

4.7.4 Linear algebra conditions for solution's existence

Let $P \subset \mathbb{R}^n$ be a polyhedron defined by a finite number m of linear inequality and equality constraints.

Definition 4.9

We can think of $m = |M|$ where M consists of possible constraints of types $i)$ $\boldsymbol{a}'\boldsymbol{x} \geq b$, $ii)$ $\boldsymbol{a}'\boldsymbol{x} = b$ and $iii)$ $\boldsymbol{a}'\boldsymbol{x} \leq b$.

1. *A vector $\boldsymbol{x} \in P$ is a **vertex** of P if there exists some c such that $\boldsymbol{c}'\boldsymbol{x} < \boldsymbol{c}'\boldsymbol{y}$ for all $\boldsymbol{y}$ satisfying $\boldsymbol{y} \in P$ and $\boldsymbol{x} \neq \boldsymbol{y}$.*
2. *If a vector $\boldsymbol{x}^*$ satisfies any type of above constraint $i), ii)$ $iii)$ we say that the corresponding constraint is **active** or **binding** at $\boldsymbol{x}^*$.* ♣

Recall the Basic Solutions of polyhedron $P \subset \mathbb{R}^n$

1. Solution $\boldsymbol{x} \in \mathbb{R}^n$ is a basic solution if
   (a) $\boldsymbol{x}$ satisfies all equality constraints of P [all equality constraints are active]
   and (b) at least n of the constraints of S are active at $\boldsymbol{x}$ and are linearly independent.
2. If $\boldsymbol{x}$ also satisfies all constraints of P, then $\boldsymbol{x}$ is called a **basic feasible solution**.

The convex combinations of two points $\boldsymbol{p}^1, \boldsymbol{p}^2 \in \mathbb{R}^n$ are points $\boldsymbol{p}$ on the line connecting $\boldsymbol{p}^1, \boldsymbol{p}^2$, with

$$\boldsymbol{p} = \lambda\, \boldsymbol{p}^1 + (1-\lambda)\, \boldsymbol{p}^2, \quad \text{where } \ 0 \leq \lambda \leq 1.$$

How about convex combinations of $k > 2$ points in $F = \{\boldsymbol{p}^1, \boldsymbol{p}^2, \ldots, \boldsymbol{p}^k\}$, every $\boldsymbol{p}^i \in \mathbb{R}^n$?

- The Convex Hull $\mathbf{CH}(F)$ of F is the set of all convex combinations of points in F, meaning:

$$\mathbf{conv}(F) = CH(F) := \{\boldsymbol{x} : \boldsymbol{x} = \sum_i \lambda_i \, \boldsymbol{p}^i, \} \tag{4.19}$$

where the convex condition hold: $\sum_i \lambda_i = 1, \; \lambda_i \geq 0.$
- The convex hull of a finite set of points in $\mathbb{R}^n$ is called a **polytope**.

In Figure 4.7 , if our LP has constraints of the form $A\,\boldsymbol{x} \leq \boldsymbol{b}$,

then its visualization is a tetraheron T with 4 extreme points say $F = \{O, A, B, C\}$ (at the vertices of the tetraheron). Then the convex hull $CH(F)$ is exactly the whole tetraheron T (inner and surface), consisting of all feasible solutions.

PRACTICE 4.2.

* If we know explicitly 4 points $O(0,0,0)$, $A(1,0,0)$, $B(0,1,0)$, and $C(0,0,1)$, can you describe T by formula?

* Does vector $\boldsymbol{x} = (1/3, 1/3, 1/4)$ belongs to T? See Theorem 4.6 for hint.

One linear constraint only Ax = b, gives us the plane P.

Non-negativity is x_1, x_2, x_3 >= 0, gives us three planes.

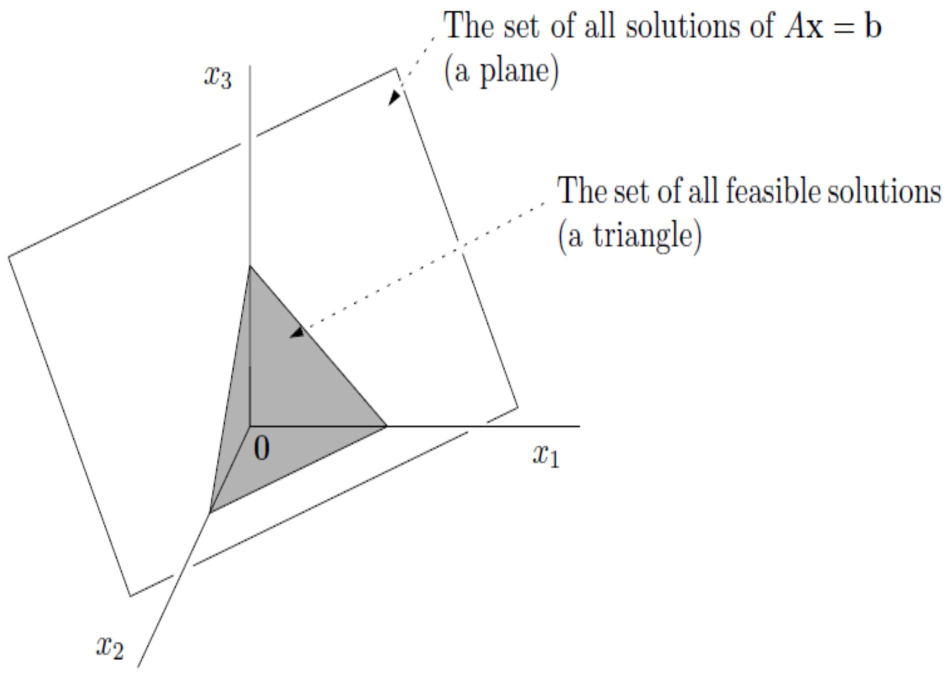


Figure 4.7: *Geometric view of the feasible region when LP has one constraint only*

Definition 4.10

Convex set and its extreme points ♣

1. A set $C \subset \mathbb{R}^n$ is called a convex set if for every finite set $F \subset C$, the convex hull $\textbf{conv}(F) \subset C$.

2. A point $\boldsymbol{x} \in C$ is an **extreme point** of C if it can not be expressed as a convex combination of other points in C.

We have given so far three different definitions that are meant to capture the same concept; two of them are geometric (extreme point, vertex) and the third is algebraic (basic feasible solution).

Fortunately, all three definitions are equivalent.

We use these terms interchangeably throughout the rest of this book.

Theorem 4.1 (Theorem 2.3, [Bertsimas]) ♡

Suppose that $\emptyset \neq P \subset \mathbb{R}^n$ be a polyhedron, and let $\boldsymbol{x}^* \in P$. The followings are equivalent.

- x^* is a vertex;
- x^* is an extreme point;
- x is a **basic feasible solution**.

4.7.5 On extreme points of a polyhedron

Few following observations are useful.

- Currently, we know we can describe a polyhedral set by its **defining hyperplanes**, and from this we can describe its *extreme points*. Can we describe the set knowing only its extreme points and directions? From our experience with two-variable linear programs, the feasible region appears as the convex hull of the extreme points, except when it was unbounded.
- Although the number of basic solutions could be infinite, basic *feasible* solutions are extreme points, therefore, their number is guaranteed to be finite, though it can be very large. We finally claim the following important fact: **Given a finite number of linear inequality constraints of a polyhedron, there can only be a finite number of basic feasible solutions.**

  Can we give an example? It is $\{0,1\}^n$, the hypercube.

Existence of extreme points

Now let $S \subset (\mathbb{R}_+)^n = \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{x} \geq \mathbf{0}\}$ be a nonempty polyhedron in the non-negative orthant. Then S has at least one extreme point. WHY? Indeed, consider the polyhedral set in 2D-plane

$$S = \{(x, y) : 1 \leq x + y \leq 3\}.$$

The set S is the region bounded by two parallel lines in $\mathbb{R}^2$ and has no extreme points. WHY? For some linear programs, such as those we will commonly see, we can say something more definitive.

Theorem 4.2 (Existence of extreme points) ♡

Let $S \subseteq \mathbb{R}^n$ be a nonempty polyhedral set in the non-negative orthant $\mathbb{R}^n_+ = \{\boldsymbol{x} : \boldsymbol{x} \geq 0\}$.

Then S has at least one extreme point. [i.e. $\emptyset \neq S \subseteq \{\boldsymbol{x} : \boldsymbol{x} \geq 0\}$ has a finite number of extreme points.]

How about the general cases?

We will obtain necessary and sufficient conditions for a polyhedron to have at least one extreme point. We first observe that not every polyhedron has this property. It turns out that the existence of an extreme point depends on whether a polyhedron contains an **infinite line** or not.

Definition 4.11 (Infinite line of a polyhedron)

*A polyhedron $P \subset \mathbb{R}^n$ **contains a line** if there exists a vector $\boldsymbol{x} \in P$ and a nonzero vector $\boldsymbol{d} \in \mathbb{R}^n$ that $\boldsymbol{x} + \lambda\, \boldsymbol{d} \in P$ for all scalars λ.* ♣

Theorem 4.3

We have the following result. Suppose that the polyhedron ♡

$$P = \{\boldsymbol{x} \in \mathbb{R}^n : A\, \boldsymbol{x} \geq \boldsymbol{b} \quad \text{or} \quad \boldsymbol{a}_i'\boldsymbol{x} \geq \ b_i, i = 1, \ldots, m\} \subset \mathbb{R}^n$$

is non empty. Then, the following are equivalent:

(a) The polyhedron P has at least one extreme point.

(b) The polyhedron P does not contain a line.

(c) There exist n vectors out of the family $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$, which are linearly independent.

Proof.

1. $(a) \Longrightarrow (c)$: Suppose P has at least one extreme point $\boldsymbol{x}$. We show the existence of n vectors out of the family $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$, which are linearly independent.
   By Theorem 4.1, $\boldsymbol{x}$ is also a basic feasible solution, then due to Definition **??**, at least n of the m constraints of P are active at $\boldsymbol{x}$ and the corresponding vectors $\boldsymbol{a}_i$ are linearly independent.

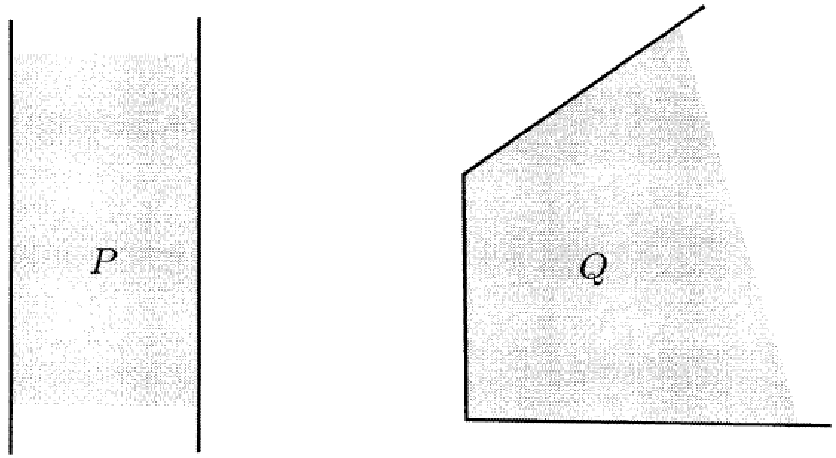
2. $(c) \implies (b)$: Suppose that $n \leq m$ vectors out of the family $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$, which are linearly independent. Without loss of generality, let us assume that $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ are linearly independent. We prove that the polyhedron P **does not** contain a line, by contradiction.

   Assume that P does contain a (an infinite) line, it means there exists a point $\boldsymbol{x} \in P$ such that $\boldsymbol{y} = \boldsymbol{x} + \lambda\, \boldsymbol{d} \in P$ where $\boldsymbol{0} \neq \boldsymbol{d} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$.
   Then,

$$\boldsymbol{a}_i'\boldsymbol{y} = \boldsymbol{a}_i'[\boldsymbol{x} + \lambda\, \boldsymbol{d}] = \boldsymbol{a}_i'\boldsymbol{x} + \lambda\, \boldsymbol{a}_i'\boldsymbol{d} \geq\ b_i \Leftrightarrow \lambda\, \boldsymbol{a}_i'\boldsymbol{d} \geq 0(*) \Rightarrow \boldsymbol{a}_i'\boldsymbol{d} = 0,$$

   for all $i = 1, \ldots, n$, and all λ. This is because if $\boldsymbol{a}_i'\boldsymbol{d} > 0$ then we can violate the constraint (*) by picking very large negative λ...
   We get coefficients $\boldsymbol{d} = 0$ since $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ are linearly independent: contradict with $\boldsymbol{0} \neq \boldsymbol{d}$.
3. $(b) \implies (a)$: Now suppose the polyhedron P does not contain a line, we prove P has at least one extreme point, or equivalently a basic feasible solution. See details in [Theorem 2.6, [**Bertsimas**]].

$\square$



The polyhedron P contains a line and does not have an extreme point,
while Q does not contain a line and has extreme points.

Notice that a bounded polyhedron does not contain a line. Similarly, the positive orthant does not contain a line. Since a polyhedron in standard form is contained in the positive orthant, it does not contain a line either. These observations establish the following important corollary.

Corollary 4.1

Every nonempty bounded polyhedron and every non empty polyhedron in standard form has at least one basic feasible solution. ♡

Optimality of extreme points

We found optimal solutions to linear programs at “corner points,” or **extreme points**, where two constraints intersect in 2D cases.

Question 2.

A key concern now is:

“Does every LP in nD case that has a finite optimal solution have, as one of its global optimum, a “corner points?”

In other words,

as long as a LP problem has an *optimal solution* and

as long as the feasible set has at least one *extreme point*,

can we always find an optimal solution within the set of extreme points of the feasible set?

Theorem 4.4 (Theorem 2.7, [Bertsimas]) ♡

Consider the linear programming of minimizing $\boldsymbol{c}'\boldsymbol{x}$ over a polyhedron $P \subseteq \mathbb{R}^n$.

Suppose that P has at least one extreme point and that there exists an optimal solution. Then, there exists an optimal solution which is an extreme point of P.

Proof. Prove that P does not contain infinite line.

Let Q be the set of of all optimal solutions our LP, which we have assumed to be nonempty.

Let P be of the form $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\ \boldsymbol{x} \geq \boldsymbol{b}\}$, and let v be the optimal value of the cost $\boldsymbol{c}' \geq \boldsymbol{x}$. Hence,

$$Q = \{\boldsymbol{x} \in P : \boldsymbol{c}' \geq \boldsymbol{x} = v\} \subset P.$$

By Theorem 4.3, P has at least one extreme point so P does not contain infinite line.

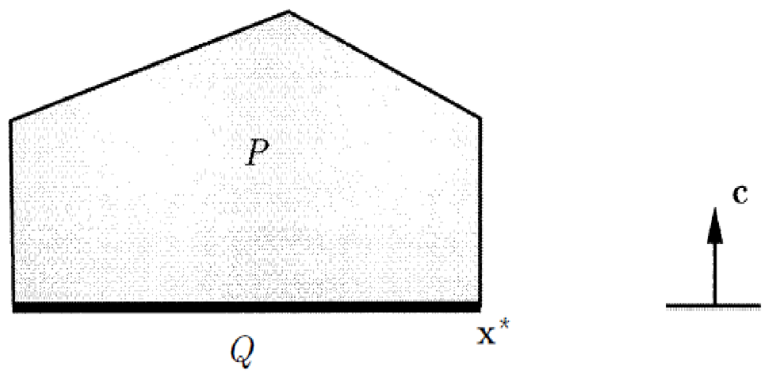


Illustration of the proof of Theorem 10.11

Set $Q \subset P$ hence Q does not contain infinite line either. Therefore, Q has an extreme point, name it $\boldsymbol{x}^*$. We just need to prove $\boldsymbol{x}^*$ is also an extreme point of P, by contradiction; and the fact that $\boldsymbol{x}^*$ is an optimal solution follows from $\boldsymbol{x}^* \in Q$.

□

The next result is stronger than the above. It shows that the existence of an optimal solution can be taken for granted, as long as the optimal cost is finite.

Theorem 4.5 (Basic result of linear programming (Dantzit, [Bertsimas])) ♡

Consider the linear programming (LP) problem of minimizing

$$z = \boldsymbol{c}^T \boldsymbol{x} = \sum_{j=1}^{n} c_j x_j$$

over a polyhedron

$$P = \{\boldsymbol{x} \in \mathbb{R}^n : A\,\boldsymbol{x} \geq \boldsymbol{b} \quad \text{or} \;\; \boldsymbol{a}_i'\boldsymbol{x} \geq \; b_i, i = 1, \ldots, m\} \subset \mathbb{R}^n$$

Suppose that P has at least one extreme point. Then

- either z attains its **optimal value** at some extreme point of P, [i.e. there exists an extreme point which is optimal];
- or the linear program is unbounded, it means $z = -\infty$.

Conclude, we need to consider only extreme points as potential optimal solutions. It is possible for an **optimal solution** of a linear program **not to be** an extreme point; this is **not** ruled out. However,

if an optimal solution exists, and there are extreme points,

then at least one of the optimal solutions must be at an extreme point.

4.7.6 Representation of bounded polyhedra

We have represented a polyhedron in terms of their defining in-equalities.

We now provide an alternative, by showing that a bounded polyhedron can also be represented as the convex hull of its extreme points. (There is a similar representation of unbounded polyhedra involving extreme points and "extreme rays" -edges that extend to infinity).

Lemma 4.1 ♡

Consider the canonical form of LP with decision variables $\boldsymbol{x} = (x_1, x_2, \cdots, x_{n-1}, x_n)$:

$$\text{Canonical LP}: \quad \min_x Z = c^T \cdot \boldsymbol{x} = \sum_{j=1}^{n} c_j x_j$$

$$\text{s. t. } A\boldsymbol{x} = \boldsymbol{b}, \qquad (A = [a_{ij}] \text{ is an } m \times n - \text{ matrix})$$

$$\text{meaning } \sum_{j=1}^{n} a_{i,j} x_j = b_i, \qquad \forall i \in \{1, 2, \ldots, m\}$$

$$\boldsymbol{x} \in \mathbb{R}_+^n \quad (\text{meaning } x_j \geq 0), \qquad \forall j \in \{1, 2, \ldots, n\},$$

Assume that you have already known two distinct feasible solution vectors $\boldsymbol{u} = (u_1, u_2, \cdots, u_{n-1}, u_n)$ and $\boldsymbol{v} = (v_1, v_2, \cdots, v_{n-1}, v_n)$. Prove that the vector $\boldsymbol{y}$ defined by

$\boldsymbol{y} = \alpha\, \boldsymbol{u} + \beta\, \boldsymbol{v},$ where real parameters $\alpha, \beta \geq 0$ and $\alpha + \beta = 1$ is also a feasible solution vector.

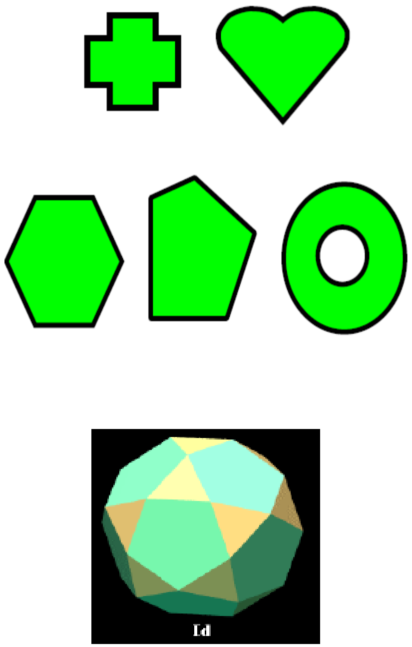


Figure 4.8: *Not all region are convex, and has extreme (corner) points*

Theorem 4.6

A nonempty and **bounded polyhedron** is the convex hull of its extreme points.

Proof. Every convex combination of extreme points is an element of the polyhedron, since polyhedra are convex sets (WHY?).

Thus, we only need to prove the converse result and show that every element of a bounded polyhedron can be represented as a convex combination of extreme points.

We defined the dimension of a polyhedron $P \subseteq \mathbb{R}^n$ [see Definition 4.8] as the smallest integer k such that P is contained in some k-dimensional affine subspace of $\mathbb{R}^n$. The proof proceeds by induction on the dimension of P.

If P is zero-dimensional, it consists of a single point. This point is an extreme point of P and the result is true. See details of inductive step in [Theorem 2.9, [**Bertsimas**]]. □

4.7.7 Mixed Integer Optimization

Mixed integer optimization (MIO) means *mixed integer linear program*. In integer programming, Lenstra found a polynomial-time algorithm when the number of variables is fixed. Although integer programming is NP-hard in general, the polyhedral approach has proved successful in practice.

For MIO we consider the objective function

$$\begin{aligned} \textbf{MILP} Z = \max_{(\boldsymbol{x},\boldsymbol{y})\in P} \quad & \boldsymbol{c}\bullet\boldsymbol{x} + \boldsymbol{h}\bullet\boldsymbol{y} \\ \text{subject to: } \; & A\boldsymbol{x} + G\boldsymbol{y} \leq \boldsymbol{b} \\ \text{the variables } \boldsymbol{x} \in \mathbb{Z}^n \text{ integral}, \; & \boldsymbol{y} \in \mathbb{R}^p_+, \text{ real} \end{aligned} \tag{4.20}$$

where $\boldsymbol{c}, \boldsymbol{h}, \boldsymbol{b}$ and A, G are rational vectors and matrices, respectively. It means

$$x_i \in \mathbb{Z}, \quad \forall\, i = 1, 2, \ldots, n; \text{ and } y_j \in \mathbb{R}_+, \ \forall\, j = 1, 2, \ldots, p;$$
$$\boldsymbol{c} \in \mathbb{Q}^n, \ \boldsymbol{h} \in \mathbb{Q}^p, \ \boldsymbol{b} \in \mathbb{Q}^m \text{ rational vectors}, \ A \in \mathbb{Q}^{m\times n}, G \in \mathbb{Q}^{m\times p}.$$

The set S of feasible solutions to (4.20) is called a *mixed integer linear set* when $p \geq 1$ and a pure integer linear set when $p = 0$. The polyhedral approach is a powerful tool for solving mixed integer linear programs (4.20).

REMINDER: A polyhedron in $\mathbb{R}^n$ is a set of the form

$$P = \{\boldsymbol{x} \in \mathbb{R}^n : A\,\boldsymbol{x} \leq \boldsymbol{b} \quad \text{or } \boldsymbol{a}_i'\,\boldsymbol{x} \leq \ b_i, \ i = 1, \ldots, m\} \subset \mathbb{R}^n$$

where $A \in \mathbb{R}^{m\times n}$ - a real matrix, and $\boldsymbol{b} \in \mathbb{R}^m$ a real vector.

■ CONCEPT 2.

1) A polyhedron of the form $\{\boldsymbol{x} \in \mathbb{R}^n : A\,\boldsymbol{x} \leq \boldsymbol{0}\}$ is called a **polyhedral cone**.

2) A finitely generated cone C is the *conic hull* of a nonempty finite set $S \subset \mathbb{R}^n$:

$$C := \mathbf{cone}(S) = \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{x} = \sum_{i}^{k} \lambda_i\, \boldsymbol{p}^i\} \tag{4.21}$$

where the cone condition $\lambda_i \geq 0$ hold, and $\boldsymbol{p}^i \in S$ for $k \geq i \geq 1.$

Minkowski-Weyl theorems

Theorem 4.7

(Minkowski-Weyl theorem for cones) ♡

A cone is polyhedral if and only if it is finitely generated.

Formally, for a set $C \subset \mathbb{R}^n$, the two following conditions are equivalent:

1. There is a matrix K such that $C = \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{x} = K\,\boldsymbol{d}, \ \boldsymbol{d} \geq 0\}$.
2. There is a matrix A such that $C = \{\boldsymbol{x} \in \mathbb{R}^n : \ A\,\boldsymbol{x} \geq 0\}$.

Theorem 4.8

(Minkowski-Weyl theorem for polyhedra) ♡

Every polyhedron P can be written as the sum of a polytope Q and a finitely generated cone C. Here

$$Q + C := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} = \boldsymbol{y} + \boldsymbol{z} \ \text{ for some } \boldsymbol{y} \in Q \text{ and } \boldsymbol{z} \in C\}. \tag{4.22}$$

Theorem 4.9 (The fundamental theorem for Mixed ILP - R.R. Meyer[7]) ♡

Given rational matrices $A \in \mathbb{Q}^{m\times n}$, $G \in \mathbb{Q}^{m\times p}$ and a rational vector $\boldsymbol{b} \in \mathbb{Q}^m$, let

$$P = \{(\boldsymbol{x}, \boldsymbol{y}) : \ A\boldsymbol{x} + G\boldsymbol{y} \leq \boldsymbol{b}\}, \quad \text{and} \ \ S = \{(\boldsymbol{x}, \boldsymbol{y}) \in P : \ \boldsymbol{x} \in \mathbb{Z}^n \ \text{ integral } \}.$$

- Then there exist rational matrices $A',\ G'$ and a rational vector $\boldsymbol{b}'$ such that

$$\mathbf{conv}(S) = \{(\boldsymbol{x}, \boldsymbol{y}) :\ A'\boldsymbol{x} + G'\boldsymbol{y} \leq \boldsymbol{b}'\}. \tag{4.23}$$

In other words, when P is a rational polyhedron, then the convex hull of S is also a rational polyhedron.

The mathematical theory of INTEGER OPTIMIZATION ends right here.

4.8 The Stochastic Fleet Size and Mix Problem (SFSMP)

OVERVIEW (Aimed for senior projects or graduate projects.)

The SFSMP will be applicable in particular urban logistics and

generally for Smart Modern Urban Management.

With MM and SP as key methodologies the SFSMP and similar trendy problems and perhaps theirs sub-optimal solutions in Smart Modern Urban Management,

with supportive knowledge from AI, Data Analytics, and Process Analytics.

CHAPTER AIM:

We investigate the smart adoption, integration, and efficient use of electro-mobility in urban logistics. In particular, we study the strategic problem of identifying the size of a fleet of mixed EVs and/or ICEVs for companies involved in urban logistics.

Key words: EVs (electric vehicles) , commercial vehicles, urban areas.

4.8.1 *SETTING and TERMS*

We consider the vehicles to be used for either cargo pickup and/or service actions (e.g. electric installations, medical visits etc.). The SFSMP is defined as a strategic decision problem. It aims at determining the optimal fleet size and mix for service operations repeating over multiple operational periods.

We assume a posteriori customer requests that are not known, owing to the lack of advance information about each day's requests at the strategic level. However, at the operational level, requests are deterministic. Since strategic decision-making here relies on the trade-off between the fleet acquisition cost and average operational cost, an accurate estimation of the operational costs is also needed.

ASSUMPTION

Customer requests and temperature are stochastic at the strategic planning stage and are only revealed prior to operations every day

The operational problem

a vehicle routing problem where demand is realized from the stochastic parameters of every single scenario.

The the strategic level- the optimization decisions concern the selection of which vehicles should compose the fleet. The decision is represented by the edges connecting each vehicle with the depot.

The network between the customers and the depot represents the operational periods where customer requests are stochastic and appear with a probability π_i

Problem 1- the **Urban freight transport logistics** includes metropolitan activities that involve vehicle movement on congested roadways in densely populated regions. Over many years, the widespread use of conventionally powered Internal Combustion Engine Vehicles (ICEVs) has negatively impacted the environment and public health in many ways. Greenhouse gas emissions from ICEVs not only aggravate climate change but also take a severe toll on the cardiovascular and respiratory health of the humans inhaling them.

Objective 1 for Problem 1: study the Informational processes, the System dynamics, secondly, set up (build) a simple car distribution model - perhaps a two-stage model, and thirdly, work out an algorithm for a two-stage or multi-stage problem.

REF: **Stochastic Fleet Size and Mix Problem**, ref. paper [**malladi2022stochastic**].

4.8.2 The Car-distribution Problem: Overview and Specification

Fleet management: Challenges and Problem Specification

- Customer's requests arrive randomly over time, often requiring service within a narrow interval. Since it can take from several days to more than a week to move transportation equipment over long distances, it is **not** possible to wait until a customer request is known before moving the equipment. As a result, it is necessary to move equipment to serve demands before they are known. In actual applications, there are other sources of randomness, such as *transit times* and *equipment failures*.

- In addition to the challenge of planning inventories, **deterministic models** have an annoying property that can produce practical severe problems. SP-MDP provides a framework for modeling the evolution of information much more accurately than is done with current technologies

The Problem Specification: The Car-distribution Problem includes two fundamental processes: the physical processes that govern car distribution and the information processes. For our modeling, we are going to represent them straightforwardly.

4.8.3 The physical process

Due to Warren B. Powell and Huseyin Topaloglu [**powell2005fleet**], There are four types of decision points:

(a) CAR: the location when the car first becomes empty, making a fleet

(b) DEPOT: the regional depot where the car is first placed when pulled from the customer,

(c) CUSTOMER intermediate location of customers and

(d) STATION: the regional depot at the destination.

NOTE 1: The last flexible decision point (d) could be replaced by a recharging station or newly designed structure/destination capturing new functionality to deal with uncertainty—

NOTE 2: Substitution is one of traffic or railroad engineers' most potent strategies to handle uncertainty. Substitution occurs across three primary dimensions:

(A) Geographic substitution Cars in different locations may be used to satisfy a particular order. The ability to choose among cars at different points in space is referred to as *geographical substitution*.

(B) Temporal substitution The engineers may provide a car that arrives on a different day.

(C) Car type substitution We may try to satisfy the order using a slightly different car type.

Cars in the logistics industry come in a variety of types. As a general rule, a customer will require a car from a particular group and may even require a car of a particular type within a group.

Our car distribution problem evolves due to flows of *exogenous information* processes and decisions. There are five classes of exogenous information processes:

1. *Car orders*. Customers call in car orders, typically the week before they are needed. The car order does not include the destination of the order.
2. *Order destination*. The destination of the order is not revealed until after the car is loaded.
3. *Empty cars*. Empty cars become available from four potential sources: cars being emptied by shippers, empty cars coming online from other railroads, cars that have just been cleaned or repaired, and new cars have just been purchased or leased.
4. *Transit times*. As a car progresses through the network, we learn the time required for specific steps (after they are completed).

5. *Updates to the status of a car*. Cars break down ("bad order" in the language of railroads) or are judged (typically by the customer) to be not clean enough.

List of Figures

Bibliography

[1] ALEXANDER HOLMES, Introductory Business Statistics, OpenStax, Rice University, 2017

[2] Annette J. Dobson and Adrian G. Barnett, *An Introduction to Generalized Linear Models*, Third Edition, CRC (2008)

[3] Antal Kozak, Robert A. Kozak, Christina L. Staudhammer, Susan B. Watts *Introductory Probability and Statistics Applications for Forestry and Natural Sciences*, CAB (2008)

[4] *Canvas paintings* by Australian artists of ethnic minorities, Australian National Museum

[5] John J. Borkowski's Home Page, *www.math.montana.edu/ jobo/courses.html/*

[6] David S. Moore, George P. McCabe and Bruce A. Craig, 2009. *Introduction to the Practice of Statistics*, 6th edition, W. Freeman Company, New York

[7] Man Nguyen, 2018. *Statistical Data Analysis I*, 1st edition, Mahidol University

[8] S.R. Dalai and al., *Factor-covering designs for Testing Software, Technometrics* 40(3), 234-243, American Statistical Association and the American Society for Quality, 1998.

[9] Douglas C. Montgomery, George C. Runger, *Applied Statistics and Probability for Engineers*, Sixth Edition, (2014) John Wiley & Sons

[10] Jay L. Devore and Kenneth N. Berk, *Modern Mathematical Statistics with Applications*, 2nd Edition, Springer (2012)

[11] M. F. Fecko and al., *Combinatorial designs in Multiple faults localization for Battlefield networks, IEEE Military Communications* Conf., Vienna, 2001.

[12] Glonek G.F.V. and Solomon P.J. *Factorial and time course designs for cDNA microarray experiments, Biostatistics* **5**, 89-111, 2004.

[13] Hedayat, A. S., Sloane, N. J. A. and Stufken, J. *Orthogonal Arrays, Springer-Verlag*, 1999.

[14] Robert V. Hogg, Joseph W. McKean, Allen T. Craig *Introduction to Mathematical Statistics,* Seventh Edition *Pearson*, 2013.

[15] Paul Mac Berthouex, Linfield C. Brown, *Statistics for Environmental Engineers*, 2nd Edition, LEWIS PUBLISHERS, CRC Press, 2002

[16] Michael Baron, *Probability and Statistics for Computer Scientists*, 2nd Edition (2014), CRC Press, Taylor & Francis Group

[17] R. H. Myers, Douglas C. Montgomery and Christine M. Anderson-Cook
Response Surface Methodology : Process and Product Optimization Using Designed Experiments, Wiley, 2009.

[18] Man Nguyen, Tran Vinh Tan and Phan Phuc Doan,
Statistical Clustering and Time Series Analysis for Bridge Monitoring Data, Recent Progress in Data Engineering and Internet Technology,
Lecture Notes in Electrical Engineering 156, (2013) pp. 61 - 72, Springer-Verlag

[19] Man Nguyen and Le Ba Trong Khang.
Maximum Likelihood For Some Stock Price Models, Journal of Science and Technology, Vol. 51, no. 4B, (2013) pp. 70- 81, VAST, Vietnam

[20] Nguyen Van Minh Man,
Computer-Algebraic Methods for the Construction of Designs of Experiments, Ph.D. thesis

[21] Nguyen, Man V. M. *Some New Constructions of strength 3 Orthogonal Arrays*,

the Memphis 2005 Design Conference Special Issue of the **Journal of Statistical Planning and Inference**, Vol 138, Issue 1 (Jan 2008) pp. 220-233.

[22] Nathabandu T. Kottegoda, Renzo Rosso. *Applied Statistics for Civil and Environmental Engineers*, 2nd edition (2008), Blackwell Publishing Ltd and The McGraw-Hill Inc

[23] Man Nguyen (2005) *Computer-algebraic Methods for the Construction of Design of Experiments*, Ph.D thesis, Eindhoven Univ. Press, 2005

[24] Paul Mac Berthouex. L. C. Brown. *Statistics for Environmental Engineers*; 2nd edition (2002), CRC Press

[25] Ron S. Kenett, Shelemyahu Zacks. *Modern Industrial Statistics with applications in R, MINITAB*, 2nd edition, (2014), Wiley

[26] Sheldon M. Ross. *Introduction to probability models*, 10th edition, (2010), Elsevier Inc.

[27] Sloane N.J.A., `http://neilsloane.com/hadamard/index.html/`

[28] Google Earth, Digital Globe, 2014- 2019

[29] Vo Ngoc Thien An, Design of Experiment for Statistical Quality Control, Master thesis, LHU, Vietnam (2011)

[30] Larry Wasserman, *All of Statistics- A Concise Course in Statistical Inference*, **Springer**, (2003)

[31] C.F. Jeff Wu, Michael Hamada *Experiments: Planning, Analysis and Parameter Design Optimization,* Wiley, 2000.