数理計画論

第2回目

- 数理計画問題法の種類と概要
- 線形計画問題の導入

謝孟春

xie.mengchun@yamato-u.ac.jp

授業の内容

- 線形計画法については、単体法において基底解法を説明し，ピボット変換によるシンプレックス法を詳述する
- 有界と非有界を具体的な解法の中で説明する
- ラグランジュ緩和による双対問題の導出を講じる
- 非線形計画法では、ラグランジュ法による解法とクーン・タッカーの定理による最適化の方法について論じる
- 組合せ最適化問題も説明する

到達目標

- 計画問題として、線形計画法とネットワーク計画法、および非線形計画を学ぶ
- 自ら計画問題を定式化し、最適化手法によって最適解を導出する
- 線形計画法では、問題の定式化を行い、シンプレックス法によって最適解を求める
- 双対性とラグランジュ緩和の意味を学ぶ
- 非線形計画問題では、ラグランジュ法とクーン・タッカーの定理を理解し、具体的に最適化問題を解法できる能力を身につける

到達目標

評価基準（ルーブリック）

	項目内容	評価4（秀相当）	評価3（優相当）	評価2（良相当）	評価1（可相当）
評価基準1	問題の定式化	現実の問題を適切に抽象化し，目的関数・制約条件を正確に定式化できる	基本的な問題について，概ね正しく定式化できる	教員の指示があれば定式化できる	定式化を部分的にできる
評価基準2	線形計画法の解法	シンプレックス法を正確に適用し，過程と結果を論理的に説明できる	手順に従い最適解を求めることができる	手順に従いある程度最適解を求め、概ね解法を理解している	一部誤りはあるが概ね解法を理解している
評価基準3	非線形計画法の解法	ラグランジュ法・KKT条件を適切に用い，問題を解ける	手順に従い解法を適用できる	教員の指示があれば手順に従い解法を適用できる	部分的に理解している

授業計画表

第1回	オリエンテーション・関数の最適化
第2回	計画問題の種類と概要
第3回	線形計画法の考え方
第4回	シンプレックス法
第5回	有界と非有界
第6回	２段階単体法
第7回	双対問題
第8回	相補性定理
第9回	中間試験
第10回	非線形最適化の基礎
第11回	最急降下法
第12回	緩和問題
第13回	Karush-Kuhn-Tucker 条件
第14回	組合せ最適化
第15回	まとめ・復習・総合演習問題

評価方法

- 演習課題　30%
- 中間試験　　40%
- 期末総合課題　30%

により成績を評価する

参考書

数理計画法-最適化の手法

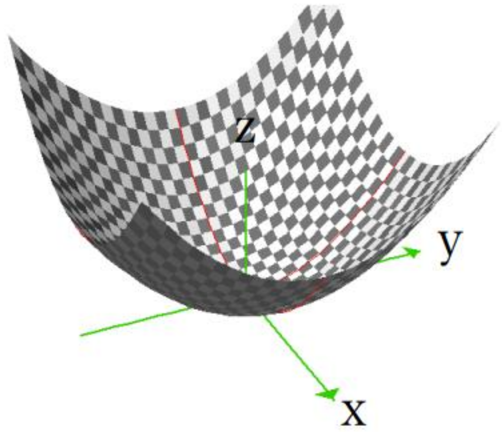
一森哲男　著

共立出版株式会社

関数の最適化

- ２変数関数とは
  - 関数：$f(x, y)$
  - 平面上の点 (x, y)に対して値をとる
  - ３次元空間の曲面として表される

【例】$f(x, y) = x^2 + y^2$



- 極値とは何か
  - 関数において局所的な最大値（極大値）と最小値（極小値）の総称
  - 山の頂上 → 極大値
  - 谷の底 → 極小値
  - 鞍点（サドルポイント）→ 極値ではない

関数の最適化

- **極値を持つための必要条件（1階偏微分）**
  - **関数が微分可能なとき：**
  - $\frac{\partial f}{\partial x} = 0, \frac{\partial f}{\partial y} = 0$

  ➡停留点（臨界点）と呼ぶ

- **関数$z = f(x, y)$ が点 (a, b) において極値をもつならば、**

  $$f_x(a, b) = 0 \text{ かつ } f_y(a, b) = 0$$

  が成り立つ

- **$f_x(a, b) = 0$ かつ$f_y(a, b) = 0$ を満たす点 (a, b) を$f(x, y)$の停留点という**

関数の最適化

【例1】関数$f(x,y) = x^3 + y^3 - 3xy$ の停留点をすべて求めよ。

【解答】

$$f_x(x,y) = 3x^2 - 3y$$
$$f_y(x,y) = 3y^2 - 3x$$

連立方程式$f_x(x,y) = f_y(x,y) = 0$

$$3x^2 - 3y = 0$$
$$3y^2 - 3x = 0$$

より

$$(x,y) = (0,0)$$
$$(x,y) = (1,1)$$

よって、 $f(x,y)$ の停留点は(0,0)と(1,1)である

関数の最適化

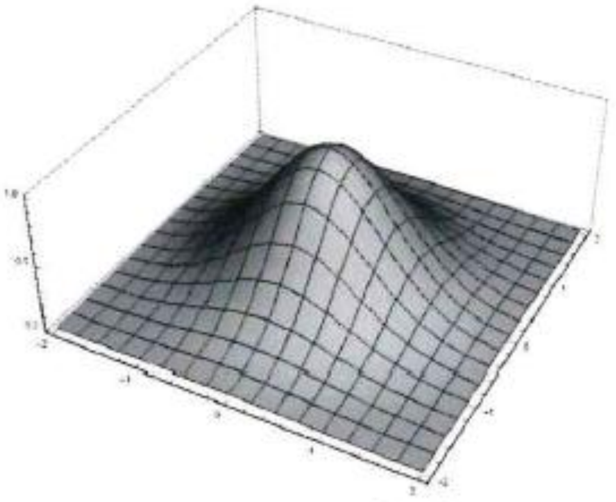
- 停留点 ＝ 極値とは限らない

【例】 $f(x,y)=x^2-y^2$ は点(0,0)を停留点にもつ

しかし、 $f(x,y)$は(0,0)で極値をもたない ➡ 鞍点

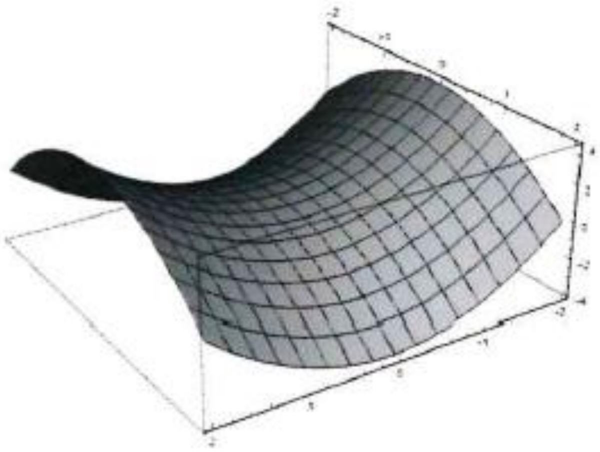
【停留点の例】

極大



(a) $z=e^{-x^2-y^2}$

鞍点



(b) $z=x^2-y^2$

関数の最適化

- 極値判定（2階偏微分による判定）
  停留点で極値を持つかどうかを判定為の条件：

ヘッセ行列（Hessian）

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \Rightarrow H(a,b) = \begin{pmatrix} f_{xx}(a,b) & f_{xy}(a,b) \\ f_{xy}(a,b) & f_{yy}(a,b) \end{pmatrix}$$

- 判定式：

$$detH(a,b) = f_{xx}(a,b)f_{yy}(a,b) - \left(f_{xy}(a,b)\right)^2$$

関数の最適化

- 極値判定条件（判定ルール）

$(1)\ \det \mathrm{H}(\mathrm{a},\mathrm{b}) > 0$　かつ　$f_{xx}(a,b) > 0$　➡　$f(x,y)$は点(a,b)で極小値

$(2)\ \det \mathrm{H}(\mathrm{a},\mathrm{b}) > 0$　かつ　$f_{xx}(a,b) < 0$　➡　$f(x,y)$は点(a,b)で極大値

$(3)\ \det \mathrm{H}(\mathrm{a},\mathrm{b}) < 0$　かつ　$f(x,y)$は点(a,b)で極値を持たない　➡　点(a,b)を鞍点という

- $det\ H(a,b) = 0$　➡ 判定不可

- 行列 $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$　に対する行列式det Aは det A=ad-bcで与えられる

関数の最適化

【例2】関数$\boldsymbol{f(x,y)=x^3+y^3-3xy}$ の停留点について、各停留点におけるヘッセ行列を求め、極値をとるかどうか判定せよ。

【解答】

$\boldsymbol{f_{xx}(x,y)=6x,\ \ f_{xy}(x,y)=-3,\quad f_{yy}(x,y)=6y}$

$$H(x,y)=\begin{pmatrix} f_{xx}(a,b) & f_{xy}(a,b) \\ f_{xy}(a,b) & f_{yy}(a,b) \end{pmatrix}=\begin{pmatrix} 6x & -3 \\ -3 & 6y \end{pmatrix}$$

【例１】より停留点は(0,0)と(1,1)

各停留点でのヘッセ行列:

$H(0,0)=\begin{pmatrix} 0 & -3 \\ -3 & 0 \end{pmatrix}$=-9<0　➡　(0,0)は鞍点

$H(1,1)=\begin{pmatrix} 6 & -3 \\ -3 & 6 \end{pmatrix}$=27>0 かつ$\boldsymbol{f_{xx}(1,1)=6>0}$ ➡　(1,1)極小値

数理計画法

- 数理計画法は数式で与えられた制約条件の下で、目的とする関数の値を最大又は最小にする問題を扱っている
- コンピュータの力を十分に活用するための数理的手法と考えられる
- 数理計画法の応用範囲は広い
  - 計算機科学
  - 情報工学
  - 電気工学
  - 経営学
  - システム工学
  - 電子工学
  - 政策科学
  - 経済学
- 特に、オペレーションズ・リサーチや経営工学では最も頻繁に利用されている

数理計画法の種類

- **線形計画法**
  - 線形つまり「まっすぐ」モデルを扱う
  - 折れ曲がりの「まっすぐ」も許す
  - 実用上、最も役に立つもの
- **ネットワーク計画法**
  - ネットワーク上のフロー問題が主
  - 特殊な線形計画法の問題
  - 整数解が容易に得られる
- **非線形計画法**
  - 「曲がり」モデルを扱う
  - 万能の解法はない
  - 実用上満足のいく解法

線形計画問題の例(P.2)

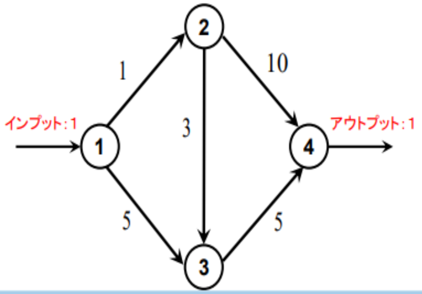
目的関数 $z = 2x_1 + x_2 \quad \rightarrow \quad \min$

制約条件 $x_1 + 2x_2 \geq 8$

$x_1 \geq 2$

$x_2 \geq 5$

ネットワーク計画問題の例(P.52)



非線形計画問題の例(P.83)

目的関数 $z = 3(1 - e^{-2x_1}) + 4(1 - e^{-2x_2}) \quad \rightarrow \quad \max$

制約条件 $x_1 + x_2 \leq 10$

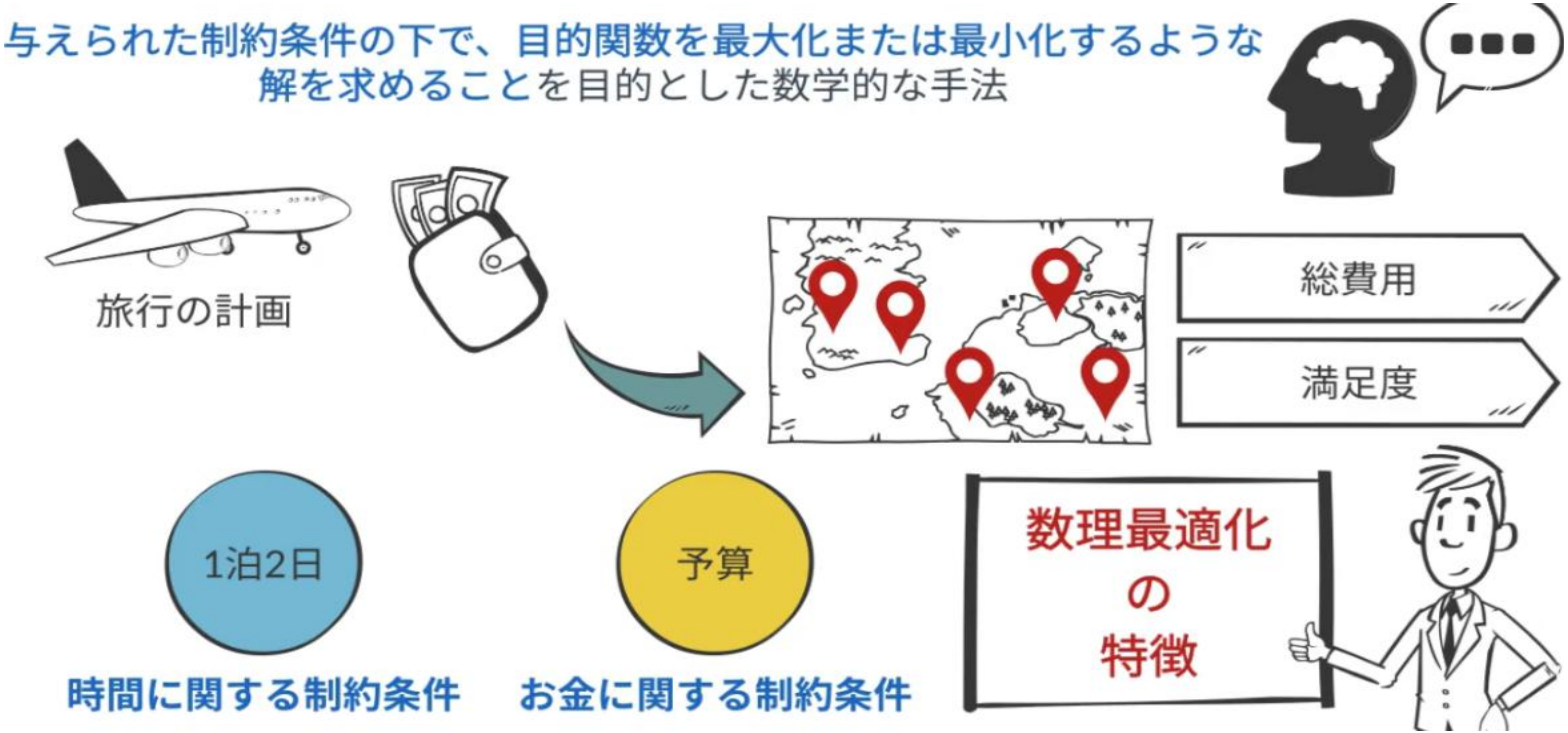
$x_1, x_2 \geq 0$

数理計画法による最適化(数理最適化)

- 最適とは、ある特定の目的、条件、状況において、最も適している、または最高の状態を指す
  - ビジネスでは制約条件の中で成果を最大化する「最適解」や、組織全体での効率化を指す「全体最適」の文脈でよく使われる

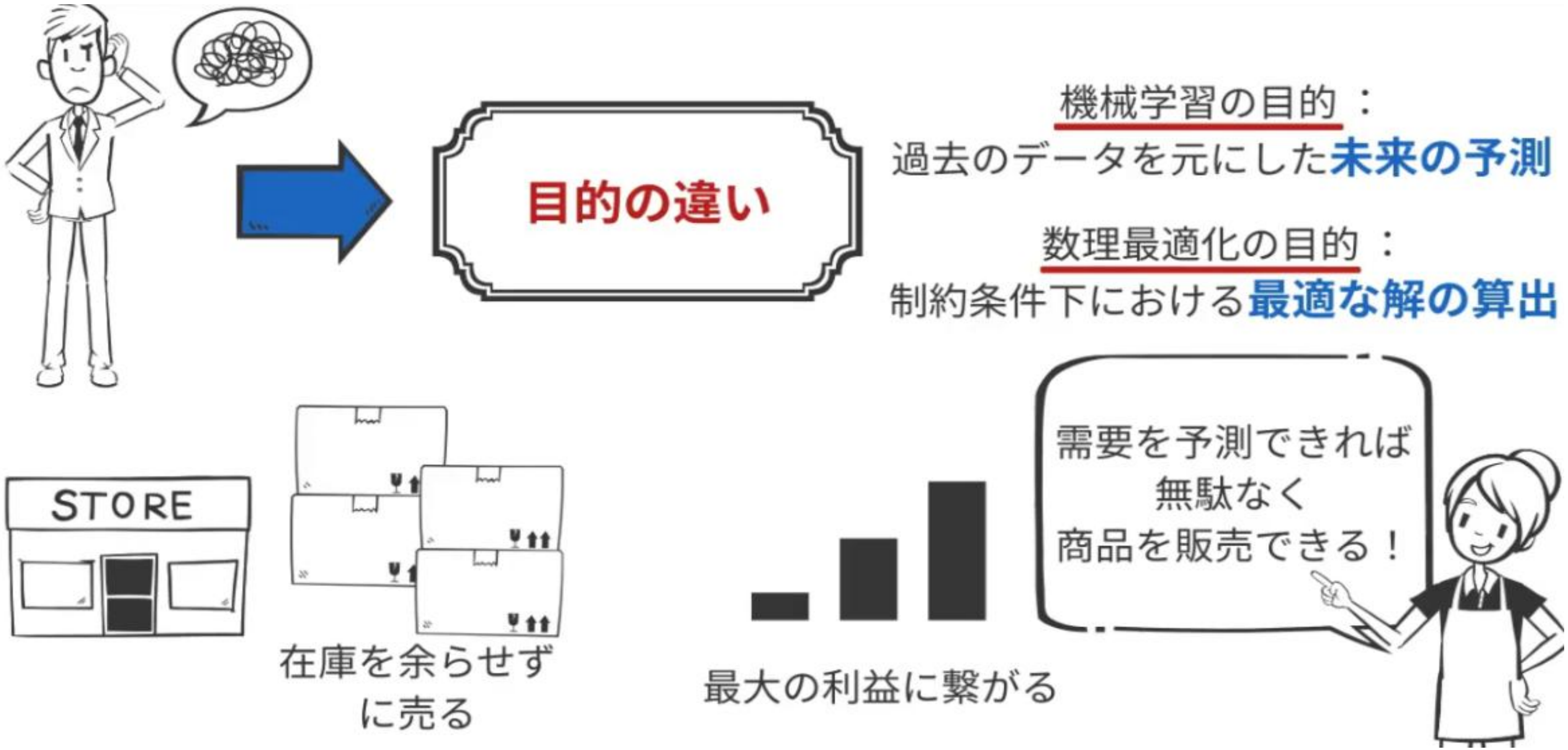
- 最適化（Optimization）とは、与えられた制約条件（予算、時間、資源など）の下で、特定の目的（コスト最小化、利益最大化、効率向上など）を最大限に達成する「最良の解」を見つけ出すプロセスや手法
  - 数学、工学、ビジネスなど様々な分野で、無駄を減らし効率を最大化するために用いられる

数理計画法による最適化(数理最適化)



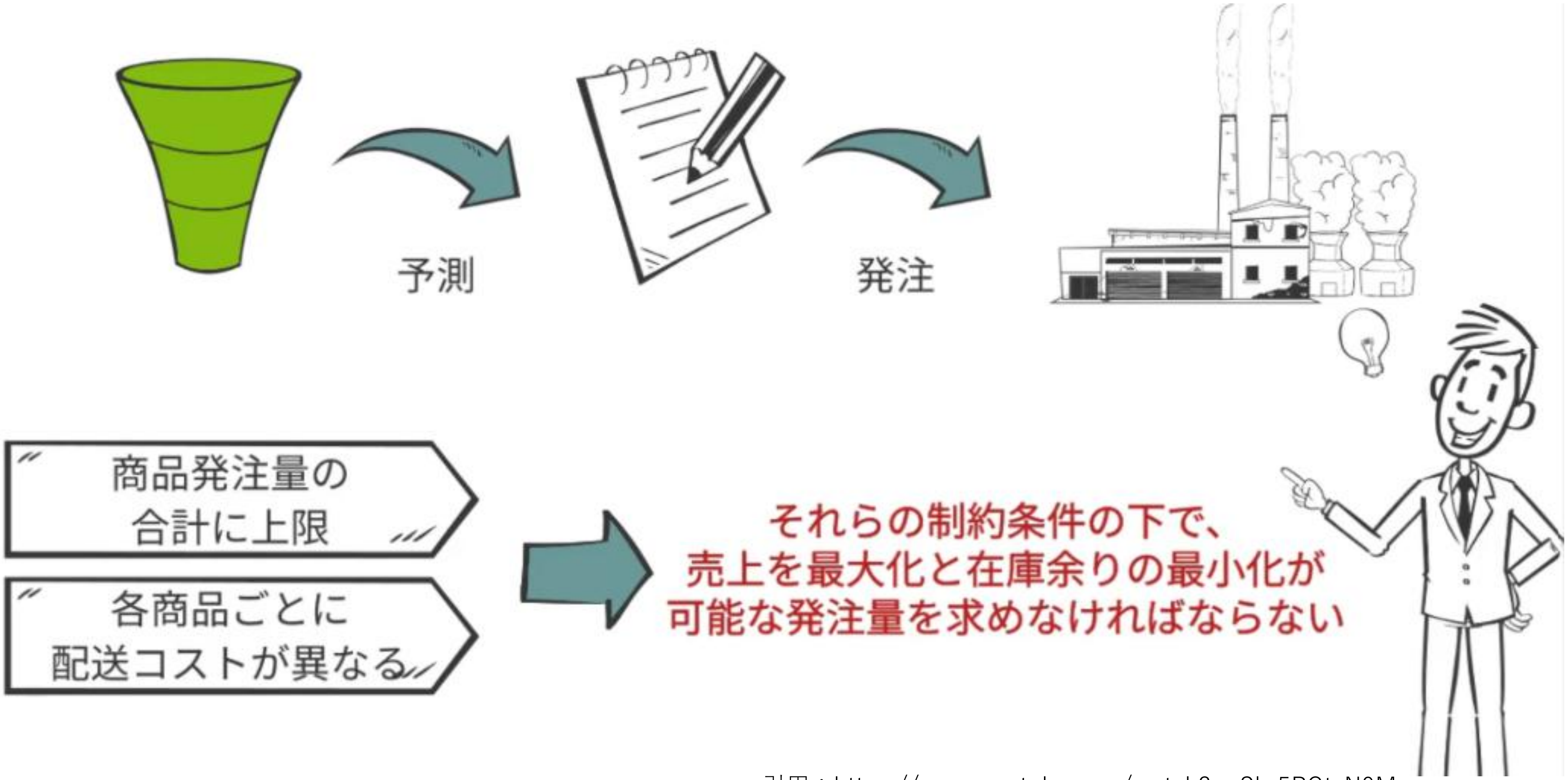
引用：https://www.youtube.com/watch?v=Sk-5PGtxN9M

数理最適化と機械学習との違い



引用：https://www.youtube.com/watch?v=Sk-5PGtxN9M

機械学習を用いて需要を予測



引用：https://www.youtube.com/watch?v=Sk-5PGtxN9M

数理最適化

目的関数

- 売上の最大化
- 在庫余りの最小化

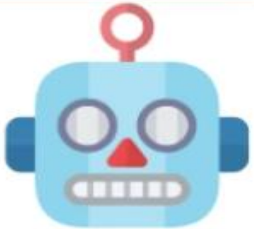
制約条件

- 商品発注量の合計
- 各商品ごとの発送コスト

制約条件を満たす発注量

日付	発注量 (商品A)	発注量 (商品B)
11/1	63	29
11/2	54	39
11/3	68	31

機械学習



未来を**予測**

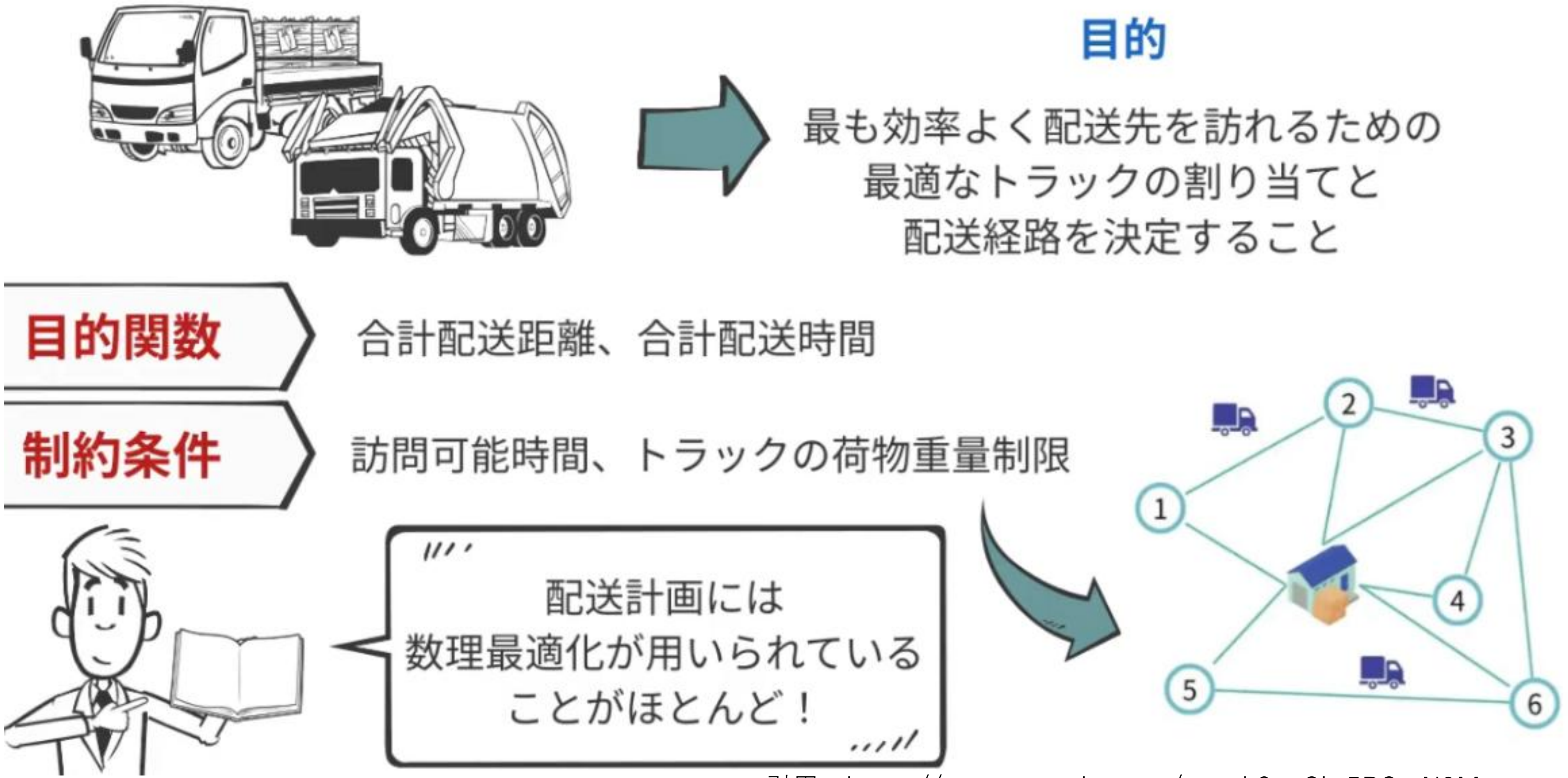
数理最適化

最適な**意思決定**

引用：https://www.youtube.com/watch?v=Sk-5PGtxN9M

数理最適化の応用例

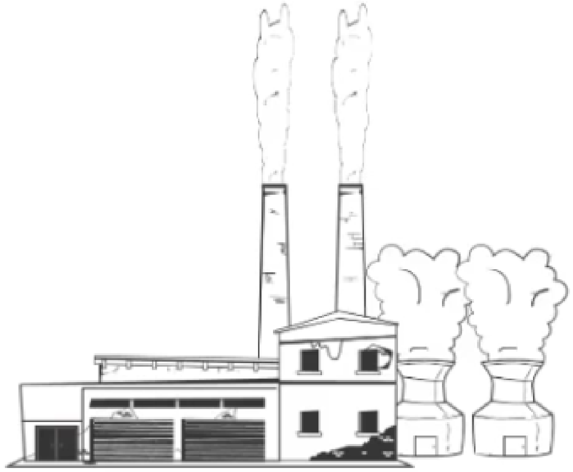
（１）配送計画の最適化



引用：https://www.youtube.com/watch?v=Sk-5PGtxN9M

数理最適化の応用例

（2）生産計画の最適化



目的

利益を最大化すること

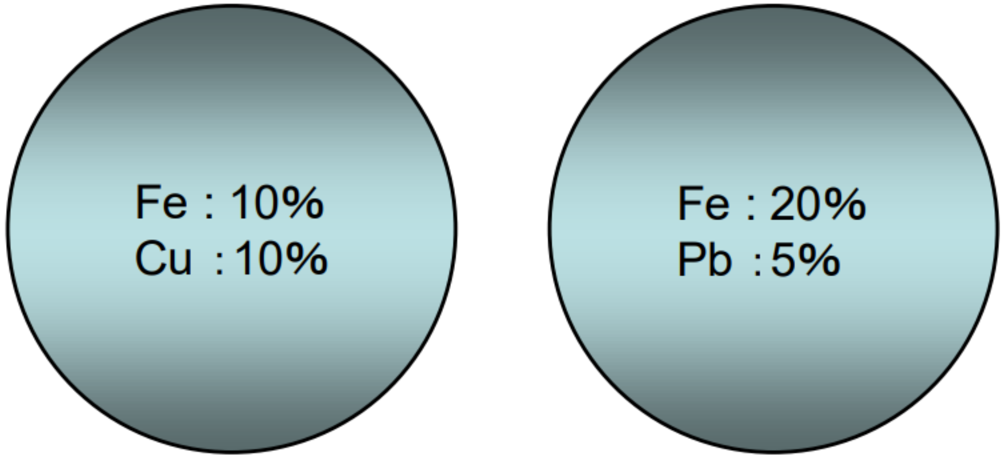
目的関数	利益の最大化
制約条件	各原材料の利用制限、 各生産ラインの稼働可能時間

引用：https://www.youtube.com/watch?v=Sk-5PGtxN9M

線形計画問題（モデル化）

- ２種類の鉱石が手に入る

鉱石１　　鉱石２



２万円/１単位　　1万円/１単位

- 1日当たり
  - Feが0.8単位
  - Cuが0.2単位
  - Pbが0.05単位

必要

- それぞれの鉱石を何単位購入？

線形計画法（定式化・モデリング）

- 変数の定義　鉱石１：x_1 単位　鉱石２： x_2 単位

- 目的関数　$z = 2x_1 + x_2$ ➡　最小化　　　（全体の購入費）

- 制約条件

$$0.1x_1 + 0.2x_2 \geq 0.8$$ （Feの必要量）

$$0.1x_1 \geq 0.2$$ （Cuの必要量）

$$0.05x_2 \geq 0.05$$ （Pbの必要量）

- 制約条件

$$x_1 + 2x_2 \geq 8$$

$$x_1 \geq 2$$

$$x_2 \geq 1$$

目的関数（有効性関数）：最小化すべき式
制約条件（制約）：制約式の群
線形計画問題：１次式で表された目的関数を
最大化もしくは最小化する問題

線形計画法（定式化・モデリング）【例】

- **変数の定義**　鉱石１：x_1 単位　鉱石２： x_2 単位
- **目的関数**　$z = 2x_1 + x_2$ ➡ 最小化　（全体の購入費）
- **制約条件**　$x_1 + 2x_2 \geq 8$　（Feの必要量）
  $x_1 \geq 2$　（Cuの必要量）
  $x_2 \geq 1$　（Pbの必要量）

【例】

$(x_1, x_2) = (6, 1)$　$z = 13$万円　$(x_1 + 2x_2 = 8 \geq 8)$　可能解
$(x_1, x_2) = (2, 3)$　$z = 7$万円　$(x_1 + 2x_2 = 8 \geq 8)$　可能解
$(x_1, x_2) = (2, 1)$　$z = 5$万円　$(x_1 + 2x_2 = 5 < 8)$　不能解

購入の仕方により購入費用が大幅にとこなる

- 実行可能解（可能解）：制約条件をすべて満足している解
- 実行不可能解（不能解）：制約条件の中に満足していないものがある
- 最適解：可能解の中で目的関数の値を最小にする解
  （最大化問題では最大にする解）
- 最適目的関数値（最適値）：最適解により定まる目的関数の値