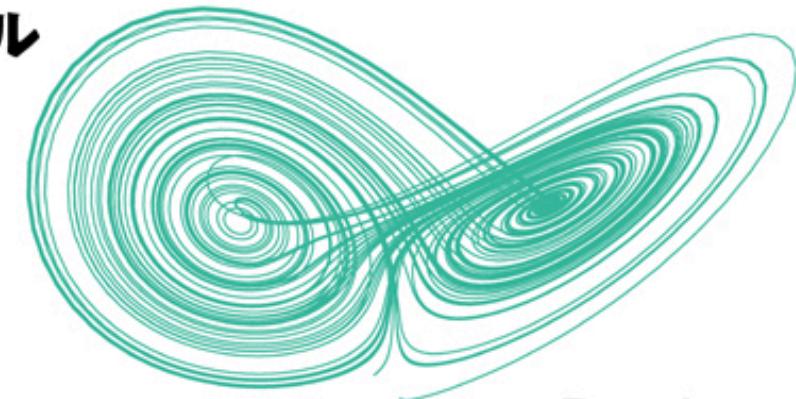


Python

コンピュータシミュレーション入門

人文・自然・社会科学の
数理モデル



Pythonコンピュータシミュ レーション入門

Python Computer Simulation

マルコフ連鎖
確率微分方程式
感染症モデル
フラクタル
在庫管理
ベイズ推定
等の拡散
遺伝的アルゴリズム
ライフゲーム
囚人のジレンマ
強化学習
意思決定

*Python
Computer
Simulation*

橋本洋志+牧野浩二 [共著]
Hashimoto Hiroshi+Makino Koji

第9章 遺伝的アルゴリズムに基づくモデル

Chapter 9 Models based on genetic algorithms

9.1 遺伝的アルゴリズムの概要

9.2 組合せに関係した問題（ナップザック問題）

9.3 順序に関係した問題（巡回セールスマン問題）

9.4 順序と組合せに関係した問題（トラック配車計画の最適化）

9.5 実数値を用いた問題（数値最適化）

9.1 遺伝的アルゴリズムの概要 / Outline of Genetic Algorithms

- 遺伝的アルゴリズム (model) は生物の進化に倣った最適化のためのアルゴリズムの一つ
 - 問題を生物の遺伝子に置き換え、淘汰と交配を繰り返して優秀な個体（望ましい結果）を残しながら最適に近い答えを見つける
 - 最適性は保証されないが、最適に近い答えを高速に見つけることが可能
 - すべての組合せや順序を探索することが難しい問題に対して有用

--

- **Genetic algorithm** (model) is one of the algorithms for optimization following the evolution of living organisms
 - It replaces the problem with the genes of an organism and finds a near-optimal answer through repeated selection and mating, leaving superior individuals (the desired outcome)
 - No guarantee of optimality, but can find near-optimal answers at high speed
 - Useful for problems where it is difficult to search for all combinations and sequences

遺伝的アルゴリズムの手順 / Procedure

- 遺伝的アルゴリズムでは、対象の問題を遺伝子で表す (=コーディングを行う) 必要がある

--

- Genetic algorithms require a genetic representation (i.e., **Coding**) of the target problem
- [バイナリコーディング / Binary Coding]

0	1	0	1	1	0
---	---	---	---	---	---

- [順序コーディング / Permutation Coding]

0	5	3	4	2	1
---	---	---	---	---	---

- [実数値コーディング / Value Coding]

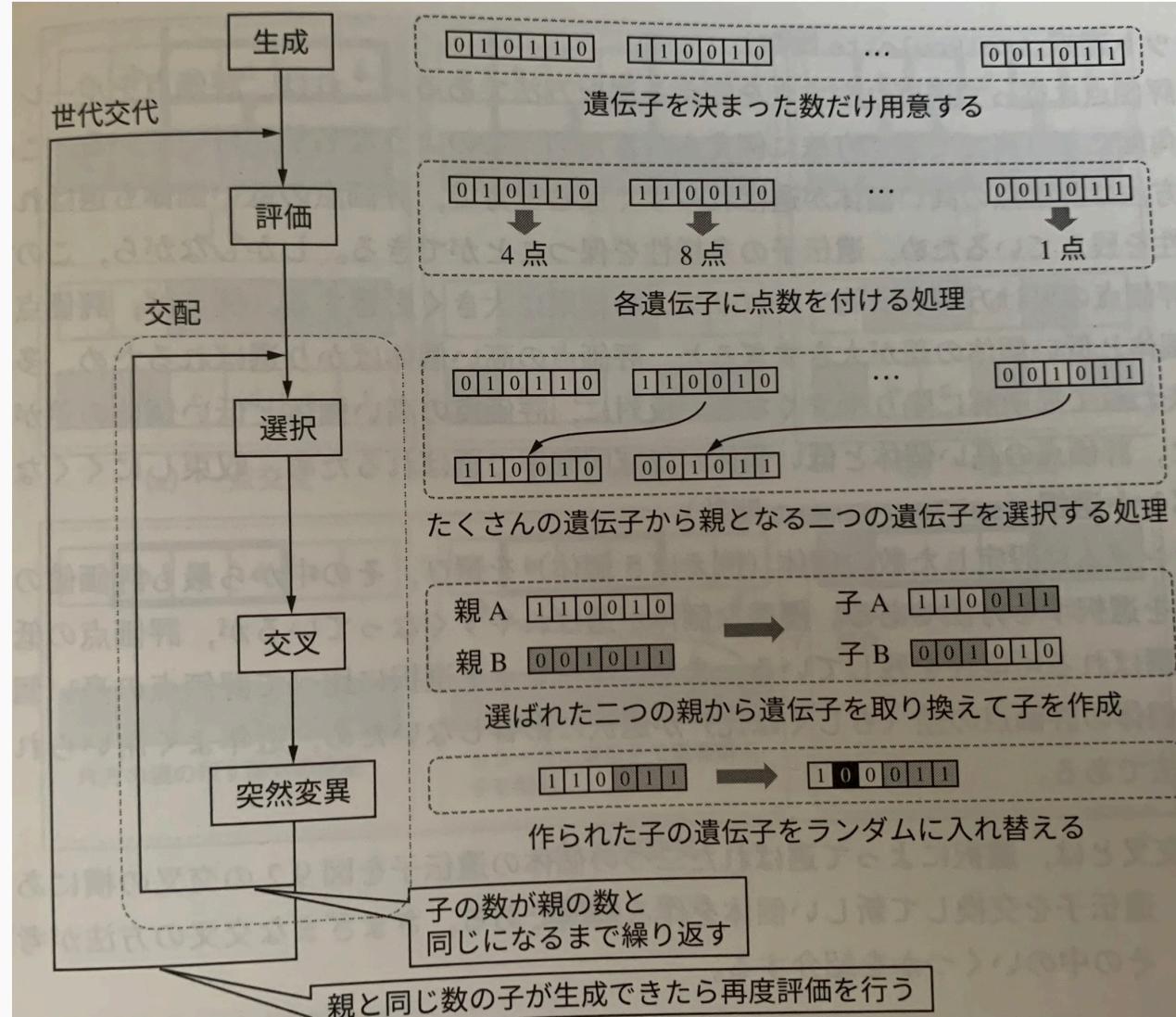
1.3	2.2	0.1	1.8	0.6	2.5
-----	-----	-----	-----	-----	-----

遺伝的アルゴリズムの手順 / Procedure

- 遺伝子を生成し、評価と交配を行って世代交代を繰り返す
- 交配は 選択 -> 交叉 -> 突然変異 の3ステップから成る

--

- Generate genes, evaluate and mating, and repeat the generational change
- Mating consists of 3 steps: Selection -> Crossover -> Mutation



遺伝的アルゴリズムの手順 / Procedure

評価 / Evaluation

- 各遺伝子に点数を付ける行為
 - 評価の方法は問題によって異なる
-
- Score each gene
 - Evaluation methods depends on the problems

遺伝的アルゴリズムの手順 / Procedure

選択 / Selection

- その後の交配に用いる遺伝子を選ぶ処理
 - 多様性を保つために、評価点の高い個体だけでなく低い個体も残すことが望ましい
-
- The process of selecting genes for subsequent mating
 - To maintain diversity, it is desirable to retain not only highly rated individuals but also those with low ratings

エリート選択 / Elite Selection	評価が高い順に個体を選択する。多様性が失われるため、局所解に陥りやすい	Selected in order of evaluation. Prone to local solutions due to loss of diversity
ランダム選択 / Random Selection	評価に関係なく選択する。多様性は保たれるが、解の収束性は悪い	Selected regardless of the evaluation. Diversity is preserved, but solution convergence is poor
ルーレット選択 / Roulette Selection	評価点に従って選択確率を変える。評価点の付け方によって多様性・収束性が左右される	Selection probability is changed according to the evaluation score. Diversity and convergence depend on how the evaluation points are assigned
トーナメント選択 / Tournament Selection	ランダムに設定した数の中から最も評価が高い個体を選択する。近年よく用いられる	Selected with the highest evaluation from a randomly set number. Often used in recent years

遺伝的アルゴリズムの手順 / Procedure

交叉 / Mutation

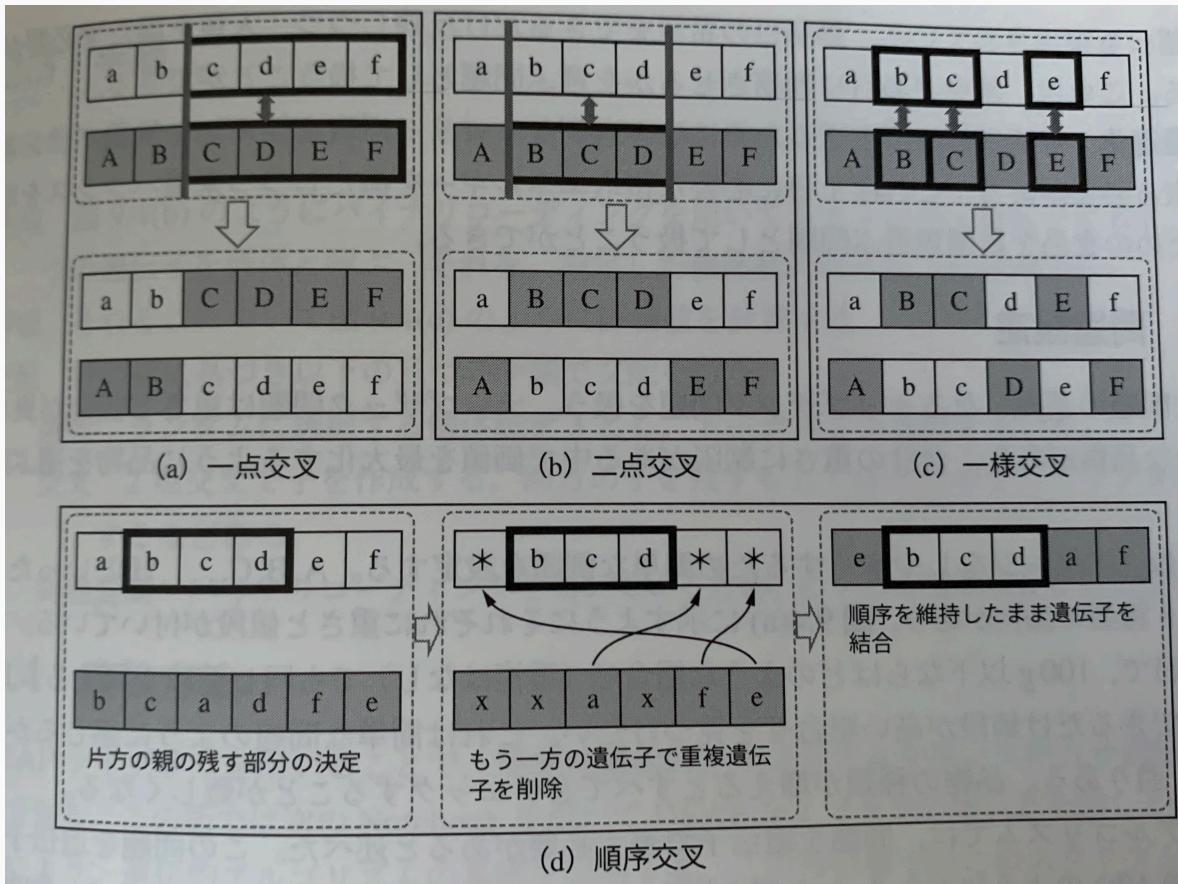
- 選択によって選ばれた2つの個体の遺伝子を交換し、新しい個体を作る
-
- Exchange the genes of two individuals selected by selection to create a new individual

一点交叉 / One-point Crossover	1ヶ所で遺伝子を分割し入れ替える	Split and replace genes in one location
二点交叉 / Two-points Crossover	2ヶ所で遺伝子を分割し入れ替える	Split and replace genes in two locations
多点交叉 / Multi-points Crossover	3ヶ所以上で遺伝子を分割し入れ替える	Split and replace genes in many locations
一様交叉 / Uniform Crossover	それぞれの遺伝子のどちらかを1つずつ選ぶ	Choose one of each gene
順序交叉 / Order Crossover	(順序コーディングにおいて) 片方の親からは一部をそのまま受け継ぎ、残りの部分はもう片方の親から相対的な順序を受け継ぐ	(In permutation coding) inherits part of it verbatim from one parent, and the rest inherits the relative order from the other parent
ブレンド交叉 / Blend Crossover	(実数値コーディングにおいて) それぞれの親の遺伝子に対しランダムに選ばれた重みを用いて重みづけ平均した値を受け継ぐ	(In value coding) inherit weighted average values using randomly chosen weights for each parent's genes

遺伝的アルゴリズムの手順 / Procedure

交叉 / Crossover

- 選択によって選ばれた2つの個体の遺伝子を交換し、新しい個体を作る
-
- Exchange the genes of two individuals selected by selection to create a new individual



遺伝的アルゴリズムの手順 / Procedure

突然変異 / Crossover

- ある確率で値を入れ替えることで、交配では得られない個体を生み出す処理
 - 局所解に陥るのを防ぐことができる
-
- A process that produces individuals that cannot be obtained by mating by swapping values with a certain probability
 - Prevents falling into a local solution

ビットの反転 / Bit Flipping	(バイナリコーディングにおいて) ビットを反転させる	(In binary coding) flip bits
順序の入れ替え / Re-ordering	(順序コーディングにおいて) 遺伝子を入れ替えて順序を変える	(In permutation coding) replacing and reordering genes
整数の入れ替え / Integer Replacement	(実数値コーディングにおいて) 遺伝子をランダムな値に入れ替える	(In value coding) Replace genes with random values

9.2 組合せに関係した問題 - ナップザック問題 / Combination-related problems - Knapsack problem

- 組合せに関係した問題 は「ある条件のもとで評価値を最大（または最小）にする問題」を解くことに相当する

--

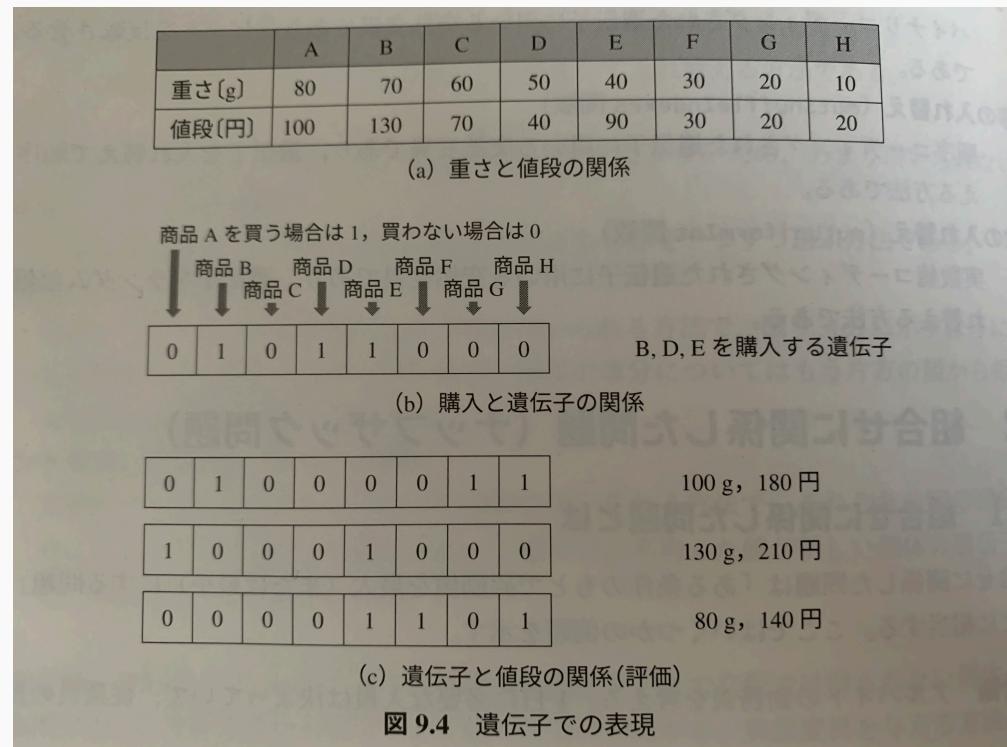
- Combination-related problems are equivalent to solving “the problem of maximizing (or minimizing) the evaluated value under certain conditions”

ナップザック問題 / Knapsack problem

- ナップザック問題 は、重さと価値が異なる様々な品物があり、合計の重さに制限がある中で価値を最大化するように品物を選ぶ問題

--

- Knapsack Problem is the problem of selecting items to maximize value with a variety of items of different weights and values, with a limit on total weight



ナップザック問題のシミュレーション手順 / Simulation Procedure

1. 生成：バイナリコーディングで遺伝子を生成する
 2. 評価：それぞれについて値段を計算する
 3. 交配
 - i. 選択：ランダムに選択する個体を3つとし、トーナメント選択を行う
 - ii. 交叉：2点交叉で子を作成する
 - iii. 突然変異：ビット反転を行う
-

1. Generation: generate genes by binary coding
2. Evaluation: calculate a price for each
3. Mating
 - i. Selection: tournament selection with 3 to be selected at random
 - ii. Crossover: create offspring by 2-point crossover
 - iii. Mutation: Perform a bit flipping

ナップザック問題のシミュレーション手順 / [Simulation Procedure](#)

- Let's check `GA_Knapsack.ipynb` and `GA_Knapsack2.ipynb`

9.3 順序に関係した問題 - 巡回セールスマン問題 / Order-related problems - Traveling salesman problem

- 順序に関係した問題は「効率のよい順番」を解くことに相当する

--

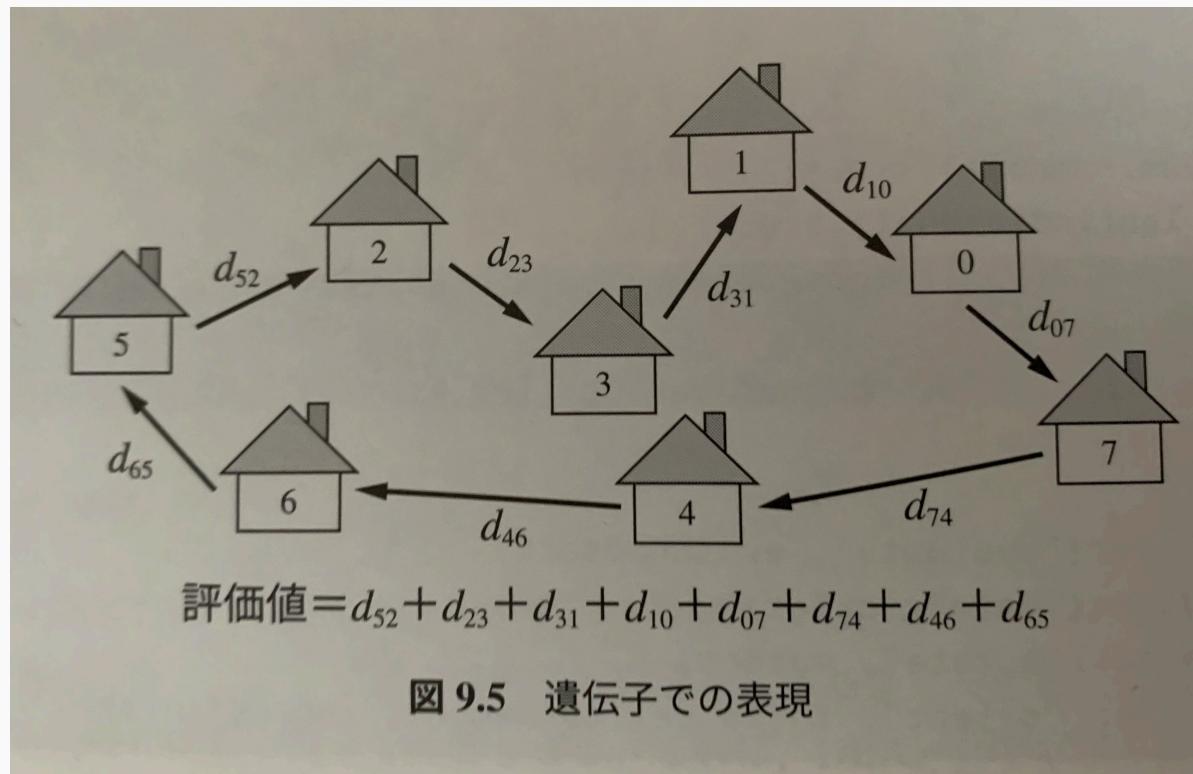
- Order-related problems are equivalent to solving "efficient order"

巡回セールスマン問題 / Traveling salesman problem

- 巡回セールスマン問題 は、すべての訪問先を最短経路でめぐる問題

--

- Traveling salesman problem is the problem of the shortest route around all the destinations



巡回セールスマン問題のシミュレーション手順 / Simulation Procedure

1. 生成：順序コーディングで遺伝子を生成する
 2. 評価：各訪問先の距離の合計を計算する
 3. 交配
 - i. 選択：ランダムに選択する個体を3つとし、トーナメント選択を行う
 - ii. 交叉：順序交叉で子を作成する
 - iii. 突然変異：順序の入れ替えを行う
-

1. Generation: generate genes by permutation coding
2. Evaluation: calculate the total distance of each visit
3. Mating
 - i. Selection: tournament selection with 3 to be selected at random
 - ii. Crossover: create offspring by ordinal crossover
 - iii. Mutation: Perform a re-ordering

巡回セールスマン問題のシミュレーション手順 / [Simulation Procedure](#)

- Let's check `GA_Salseman.ipynb`

9.4 順序と組合せに関係した問題 - トラック配車計画の最適化 / Order and combination related problems - Truck dispatch planning optimization

- 順序と組合せに関係した問題は「ある条件のもとで評価値を最大（または最小）にする順序付きの組合せ」を解くことに相当する

--

- Order-related problems are equivalent to solving for “the ordered combination that maximizes (or minimizes) the evaluated value under a certain condition

トラック配車計画の最適化 / Truck dispatch planning optimization

- **トラック配車計画の最適化問題** は、2台のトラックが複数店舗を手分けして巡回する問題
- 以下の2つを同時に満たす解を探す
 - うまく2つにわける（組合せ）
 - 最短経路で回る（順序）

--

- **Truck dispatch planning optimization problem** is the problem of two trucks making hand-to-hand patrols of multiple stores
- Find a solution that satisfies both of the following at the same time
 - Divide well into two (combination)
 - The shortest route around (order)

トラック配車計画の最適化 / Truck dispatch planning optimization

1. 生成：順序コーディングで遺伝子を生成する
 2. 評価：2台のトラックそれぞれの巡回距離の合計を求める
 3. 交配（巡回セールスマントロードと同じ）
 - i. 選択：ランダムに選択する個体を3つとし、トーナメント選択を行う
 - ii. 交叉：順序交叉で子を作成する
 - iii. 突然変異：順序の入れ替えを行う
-

1. Generation: generate genes by permutation coding
2. Evaluation: calculate the total distance traveled by each of the two trucks
3. Mating (Same as Travel salesman problem)
 - i. Selection: tournament selection with 3 to be selected at random
 - ii. Crossover: create offspring by ordinal crossover
 - iii. Mutation: Perform a re-ordering

トラック配車計画の最適化 / Truck dispatch planning optimization

- Let's check `GA_SalsemanKnapsack.ipynb`

9.5 実数値を用いた問題 - 数値最適化 / Problems with real numbers - Numerical optimization

- 実数値を用いた問題 は「ある条件のもとで評価値を最大（または最小）にする問題」を解くことに相当する
 - 実数値を用いる、という点で組合せ問題とは異なる

--

- Problems with real numbers are equivalent to solving “the problem of maximizing (or minimizing) the evaluated value under certain conditions”
 - It differs from combination-related problems in that it uses real numbers

数値計画問題 / Mathematical Programming Problem

- 数理計画問題 は、制約条件のもとで目的関数を最大（または最小）にする問題

--

- Mathematical Programming Problem is the problem of maximizing (or minimizing) the objective function under constraints

Objective function

$$4x_1 + 5x_2$$

Constraints

$$2x_1 + 2x_2 \leq 7$$

$$3x_1 + 5x_2 \leq 14$$

数値計画問題 / Mathematical Programming Problem

1. 生成：実数値コーディングで2つの値 (x_1, x_2) を生成する
 2. 評価：目的関数を計算する
 3. 交配
 - i. 選択：ランダムに選択する個体を3つとし、トーナメント選択を行う
 - ii. 交叉：ブレンド交叉で子を作成する
 - iii. 突然変異：整数の入れ替えを行う
-

1. Generation: generate two values (x_1, x_2) with real-valued coding
2. Evaluation: calculate the objective function
3. Mating (Same as Travel salesman problem)
 - i. Selection: tournament selection with 3 to be selected at random
 - ii. Crossover: create offspring by blend crossover
 - iii. Mutation: Perform an integer replacement

數值計画問題 / Mathematical Programming Problem

- Let's check `GA_Value.ipynb`