

(組み合わせ)最適化

概要

最適化とは、いくつかの条件を満たすものの中で一番いいものを、見つける事をいいます。その見つける場所や方法を眺めつつ、どのような問題があり、それをどのように解決していくのを一緒に考えてみましょう。

當間愛晃 @ 琉球大学工学部情報工学科
tnal@ie.u-ryukyu.ac.jp
<http://www.eva.ie.u-ryukyu.ac.jp/~tnal/>



目次

> 話題の中心

1. 問題空間（考慮すべき組み合わせ数）の広さ
2. 定式化（モデル化）の意義
3. 最適化の考え方

< 定式化例として >

1. ナップサック問題
2. （巡回セールスマン問題（TSP））

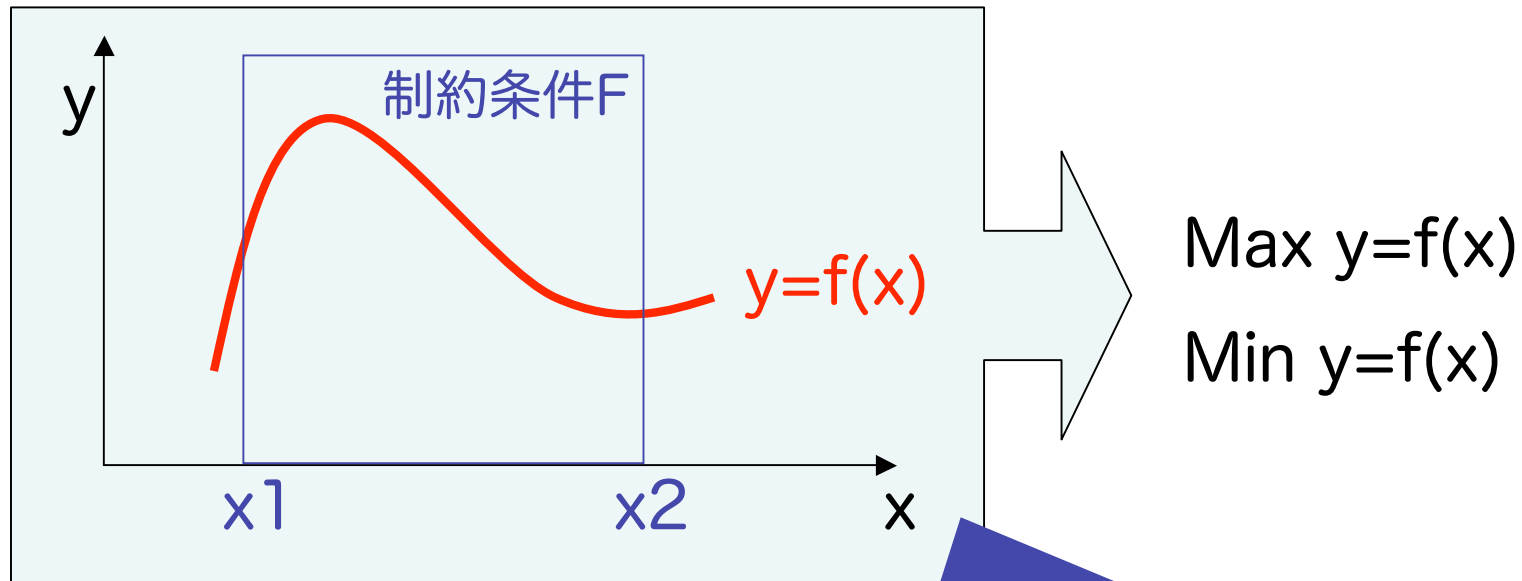
> 最適化手法の例

1. 完全解法と近似解法

今日は具体事例を交えつつ考えてみましょう！

最適化のイメージ1

ある解空間 X において制約条件 F を満足しつつ、目的関数 $f(x)$ を最大化 (or 最小化) する解 x を求める。



Q: どんな探索手法が考えられるか?
また、その特徴は?

最適化のイメージ2

組み合わせ最適化問題

最適化問題

ある解空間 X において制約条件 F を満足しつつ、目的関数 $f(x)$ を最大化 (or 最小化) する解 x を求める。

制約条件

+

解空間 X は離散集合。

Q: 連続空間を対象とした探索手法は、組み合わせ最適化問題にもその適用可能か？
利用上の観点からは十分現実的な解法か？

(例1) Tさんの小遣いの使い道

- 毎月2,000円の小遣いがある。
- 当然、買いたい物リストは山積みで、全てを買う事は出来ない。
- (例えば半年分の小遣いで、) **最も満足度が高くなる組み合わせはどれだろうか？**

	満足度	値段
1. 漫画	3	500
2. 服	5	5,000
3. 映画	6	1,600
4. ゲーム1	7	5,800
5. ゲーム2	4	1,500
...		

選択例

漫画＋服 → 満足度8, 値段5,500
服＋映画 → 満足度11, 値段6,600
服＋映画＋ゲーム1 → 満足度18, 値段12,400(NG)

ナップサック問題 (定式化)

入力データ : n 個の商品の満足度 $value_i$ と値段 $cost_i$, 予算上限 $limit$
制約条件 : 選んだ組み合わせの値段合計が予算上限を超えない事
目的 : 満足度を最大化する組み合わせを見つける事

目的関数 (満足度) : $\max \left(\sum_{i=1}^n x_i \times value_i \right)$

制約条件 : $limit \leq \sum_{i=1}^n x_i \times cost_i$

例えば, 商品1,3を購入
満足度= $value_1 + value_3$
 $limit \leq cost_1 + cost_3$

ただし, $x_i=1$, when 商品 i を買う
 $x_i=0$, otherwise (商品 i を買わない)

ナップサック問題（具体例）

> 基本

- 商品満足度, コスト制約 → 最大満足解

> 具体例

- 納入商品選び
 - 商品価値, コスト制約 → 売り上げ(価値)最大
- 人工衛星のパーツ選び
 - 部品価値, 重量制約 → 最大価値
- 残り時間の使い方
 - 授業単位, レポート作成時間 → 最大単位

なぜ定式化するのか？

> 物事を簡潔にとらえる

対象問題に対する認識／理解を深める(特徴明確化)
枝葉に囚われずに考えることができる

> 過去の財産（手法）を利用する

有用なアイデアを再利用できる（可能性がある）
過去の実績があれば評価コストも削減できる

> 手法を応用（拡張）できる

一度確立すれば類似の問題に適用できる

様々なコスト
を減らしたい



理論的なコ
スト削減



定式化のために基
礎（数学/物理/英
語）が重要

ナップサック問題（組合せ数）

仮に1,000個の組み合わせまでは1秒で処理できる計算機があるとすると…

> 商品数 $n=1$ の場合

どういう解が考えられるか？

商品1を買う

商品1を買わない(2通り)

> 商品数 $n=2$ の場合

どういう解が考えられるか？

商品1も2も買わない

商品1だけを買う

商品2だけを買う

商品1も2も買う

→ $2 \times 2 = 4$ 通り

> 商品数 $n=3$ の場合

$2 \times 2 \times 2 = 8$ 通り

> 商品数 $n=N$ の場合

組み合わせ数 = 2^N

商品数	組み合わせ数	計算時間(目安)
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	
8	256	
9	512	
10	1,024	約1秒
20	1,048,576	約17分
30	1,073,741,824	約12日
40	1,099,511,627,776	約34年
50	1,125,899,906,842,620	約35,702年

ナップサック問題（アプローチ）

> 前提

すべての組み合わせをしらみつぶしに調べることは出来ないほど大規模な問題空間!

Q: どのように最適解を見つければ良いのか?

完全解法

確実に最適解
を見つける

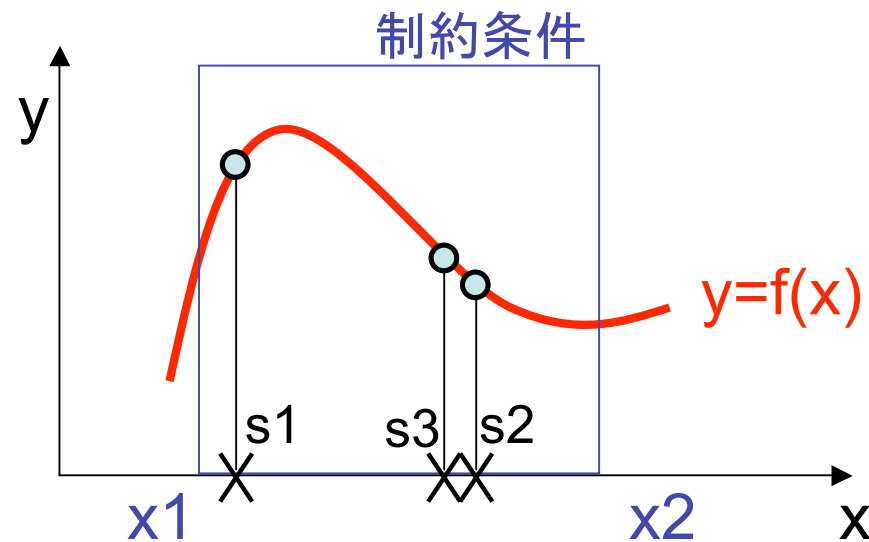


近似解法

(短時間で) なるべく
良い解を求める

近似解法の種類1 (ランダムサーチ)

ランダムサーチ
(Random Search)
ランダムに解を生成



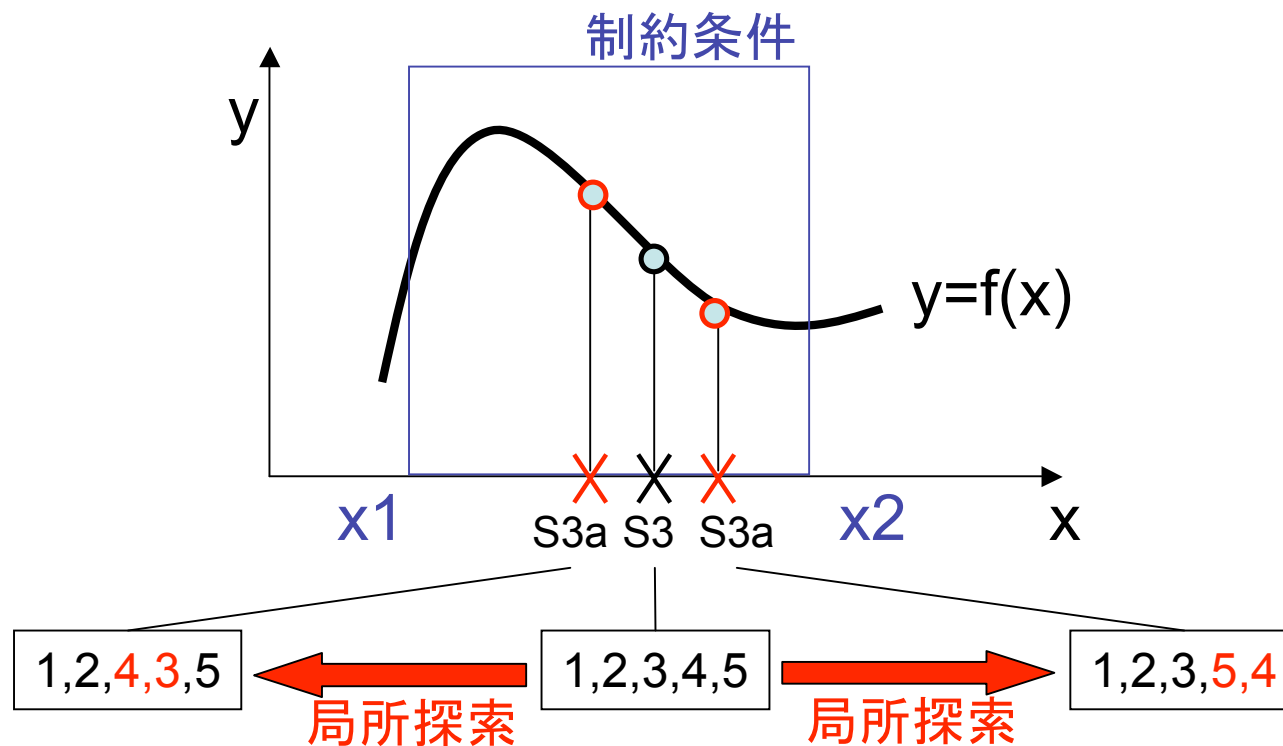
近似解法の種類2 (局所探索)

山登り法 (Hill Climbing method)

近傍のより良い解のある方向を探索

焼き鈍し法 (Annealing method)

確率的に悪い解へ探索を進める事もある



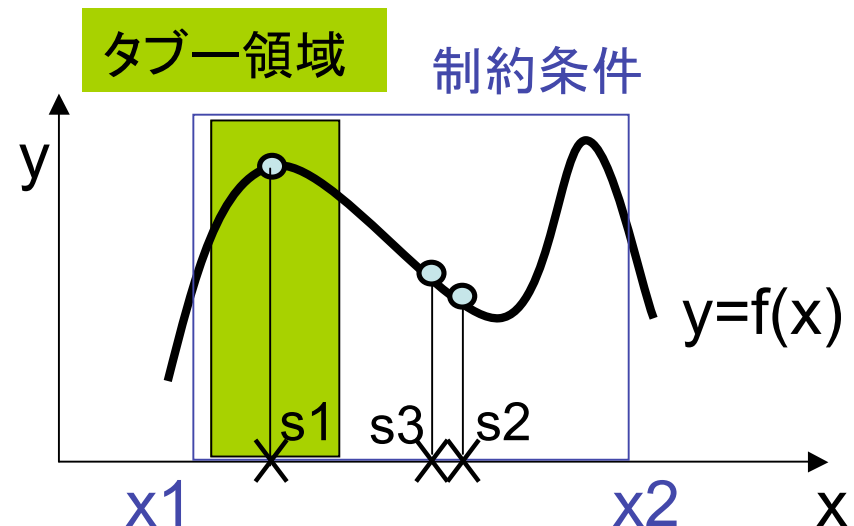
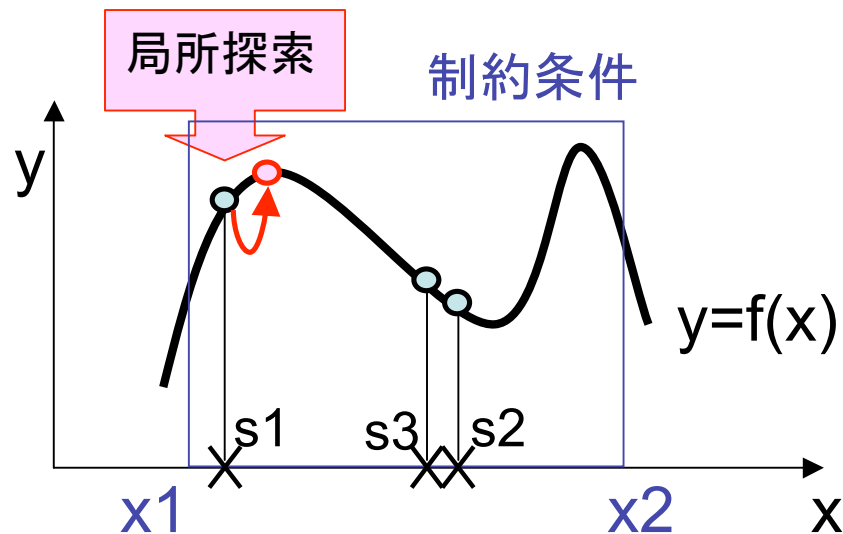
近似解法の種類3 (多点探索・タブー)

多点探索

同時に複数の解を探索。
他の手法に組み込んで利用。

タブー探索

タブーリストを作成し、該当する解への探索を禁止。



最適化手法を考えるにあたって

> 最適化の方法

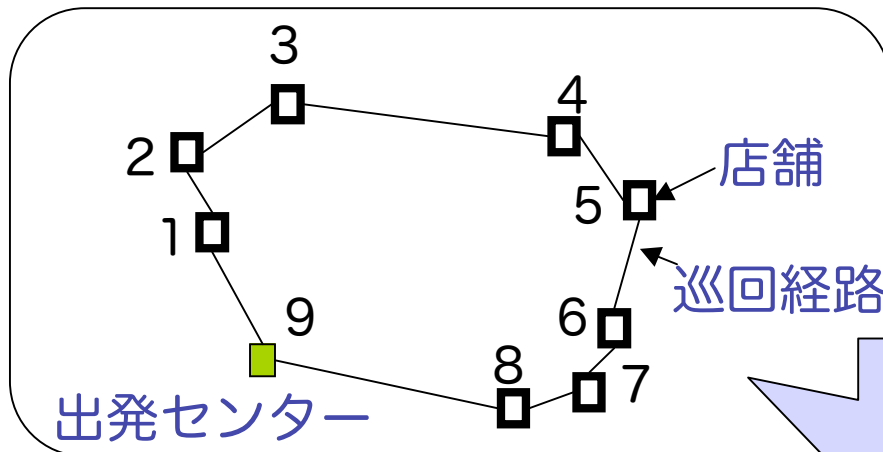
- 何とかして「無駄を省く」
- 確率的な方法 or 確定的な方法

> 考慮すべき点

- 優先項目は？
 - 時間？精度？再現性？代替案？
- 客観的な評価は？
 - 第三者が追実験可能であること
 - 数値化
 - 良く知られた手法との比較

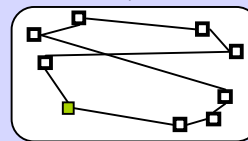
(例2) N社の配送計画

- コンビニ各店舗に商品を配送する必要がある。
- 配送時間を短くできると、それだけ配送コストを節約でき、環境にも良い。
- **最も短時間で配送し終える順序はどれだろうか？**

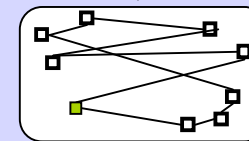


巡回例 (実行可能解)

解1

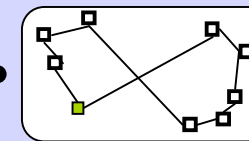


解2



...

解N



巡回セールスマン問題 (定式化)

入力データ : n 個の拠点と拠点間の距離 $dist$

制約条件 : 全ての拠点を一度ずつ巡回すること

目的 : 総距離が最小となる巡回経路を見つける事

目的関数 : $min \left(\sum_{i=1}^{n-1} path_i \right)$
(総距離)

$path_i$ は2拠点経路 (パス)
例えば,
「都市2から出発→都市3へ」
は「 $path_1=\{2,3\}$ 」と記述.

制約条件 : $set(AllCities) = set \left(\frac{\sum_{i=1}^{n-1} path_i}{2} \right)$

巡回セールスマン問題（具体例）

> 基本

- 拠点座標→総距離最小化

> 具体例

- ドリル穴空け
 - 電子基板上の穴→総移動距離最小化
- 配送計画全般
 - N人セールスマン
 - N人セールスマン+Mステーション
 - 時間制約付き

巡回セールスマン問題 (組合せ数)

> 都市数 $n=2$ の場合

どういふ解が考えられるか？

都市1→都市2

都市2→都市1(距離は上と同じなので1通り)

> 都市数 $n=3$ の場合

どういふ解が考えられるか？

1-→2-→3

1-→3-→2

2-→1-→3

2-→3-→1

3-→1-→2

3-→2-→1

→ $(3 \times 2) / 2 = 3$ 通り

> 都市数 $n=4$ の場合

どういふ解が考えられるか？

(1) 最初に4通り: 1,2,3,4のどれかを選択

(2) 2番目に3通り: (1)から除いた3都市のどれか

(3) 3番目に2通り: (2)から除いた2都市のどれか

(4) 4番目に1通り: (3)から除いた1都市

→ $(4 \times 3 \times 2 \times 1) / 2 = 12$ 通り

> 都市数 $n=N$ の場合

組み合わせ数 = $(n-1)! / 2$

巡回セールスマン問題（組合せ数2）

ナップサック問題（再掲）

巡回セールスマン問題

商品数	組み合わせ数	1000/1sec
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	
8	256	
9	512	
10	1,024	約1秒
20	1,048,576	約17分
30	1,073,741,824	約12日
40	1,099,511,627,776	約34年
50	1,125,899,906,842,620	約35,702年

都市数	組み合わせ数	1000/1sec
2	1	
3	3	
4	12	
5	60	
6	360	
7	2,520	約2秒
8	20,160	約20秒
9	181,440	約3分
10	1,814,400	約30分
11	19,958,400	約5.5時間
12	239,500,800	約2.7日
13	3,113,510,400	約36日
14	43,589,145,600	約1.4年
15	653,837,184,000	約20.7年
16	10,461,394,944,000	約331.7年

巡回セールスマン問題（アプローチ）

> 前提

すべての組み合わせをしらみつぶしに調べることは出来ないほど大規模な問題空間!

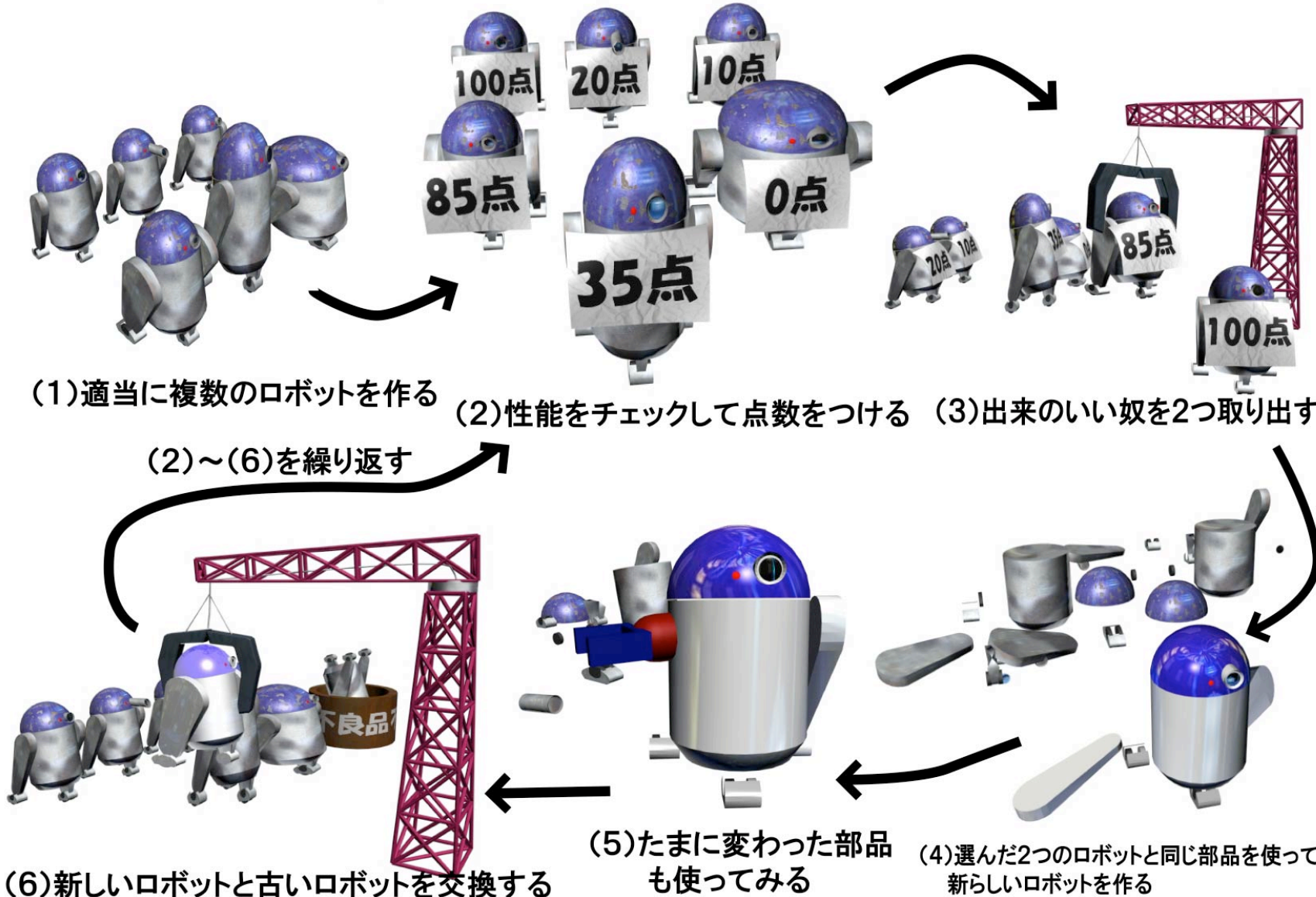
Q: どのように最適解を見つければ良いのか?

> 考慮すべき項目 <

(例) 時間, 解の精度, 代替案の有無,
確率的/確定的, 評価方法, , ,

汎用的なアプローチ例

遺伝的アルゴリズム 生命の遺伝を模倣したモデル



進化計算班

これまでに作られたパッケージ

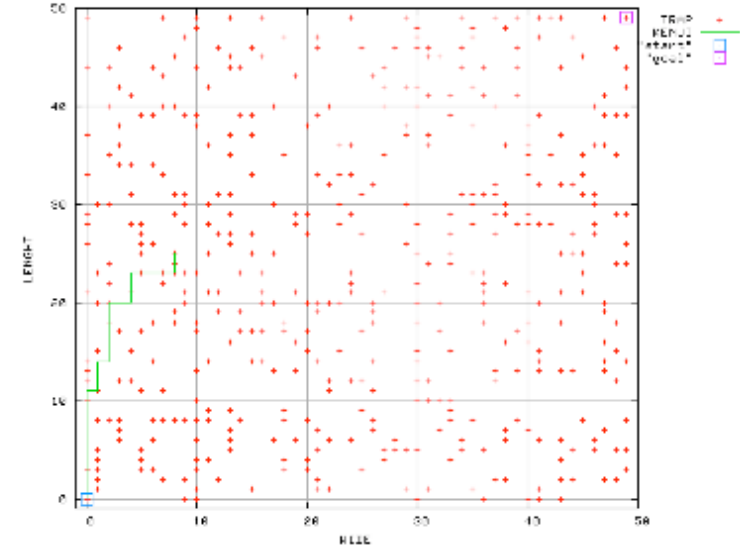
<http://www.nal.ie.u-ryukyu.ac.jp/~info3/wiki/>

けんじくえすと2 KENJI QUEST

■ ケンジクエストとは

- 昨年先輩が作成したGAによるケンジの冒険
- 旅の途中にトラップ（モンスター）に出くわすと戦闘
- ケンジはゴールを目指して進化
- マップはケンジの妨害をするように進化

マップ最大評価時のゲーム状況



Kenji Quest v1 (2005年度後期)

Kenji Quest v2 (2006年度前期)

Chelnov(ダンジョンvs探検家s)

IPD(繰り返し囚人のジレンマ)

Labos(研究室配属問題)

Diet_plan(ダイエットメニュー提案システム)

Obatchan(ヤクルトおばちゃん配送経路計画)

} 公開

などなど

まとめ

> 組み合わせ最適化問題とは

- ・出来るだけ無駄を省きつつ良質の解を見つけるためのベンチマーク問題

> 定式化する事で

- ・再利用可能な共有の財産となる

> 最適化の方法は

- ・アプローチの仕方は適材適所だが、客観的な方法で評価する必要がある

> (最近の流行として)

- ・可能な限り短時間 (リアルタイム処理)

最後に・・・

> あなたの夢にはどのような「解決すべき問題点」がありますか？

> それは技術的な問題ですか？金銭的な問題ですか？政治的な問題ですか？

> それはどうすれば解決出来ると思いますか？

夢を実現できるようにお互い頑張りましょう！

参考/関連サイト

Nagamochi Laboratory (Kyoto University)

<http://www-or.amp.i.kyoto-u.ac.jp/>

<http://www-or.amp.i.kyoto-u.ac.jp/research/introduction.pdf>

村上・泉田研究室 (愛媛大学)

<http://ipr20.cs.ehime-u.ac.jp/>

<http://ipr20.cs.ehime-u.ac.jp/column/ga/chapter4.html>

数理計画入門 (東京大学)

<http://www.misojiro.t.u-tokyo.ac.jp/~tomomi/comb-opt/>

<http://www.misojiro.t.u-tokyo.ac.jp/~tomomi/comb-opt/intro/ppframe.htm>

人工知能のやさしい説明 (人工知能学会)

<http://www.ai-gakkai.or.jp/jsai/whatsai/>

マッチ箱の脳 (AI) — 使える人工知能のお話

森川 幸人 (著), 新紀元社 ; ISBN: 4883170802 ; (2000/12)