情報工学実験3：進化計算

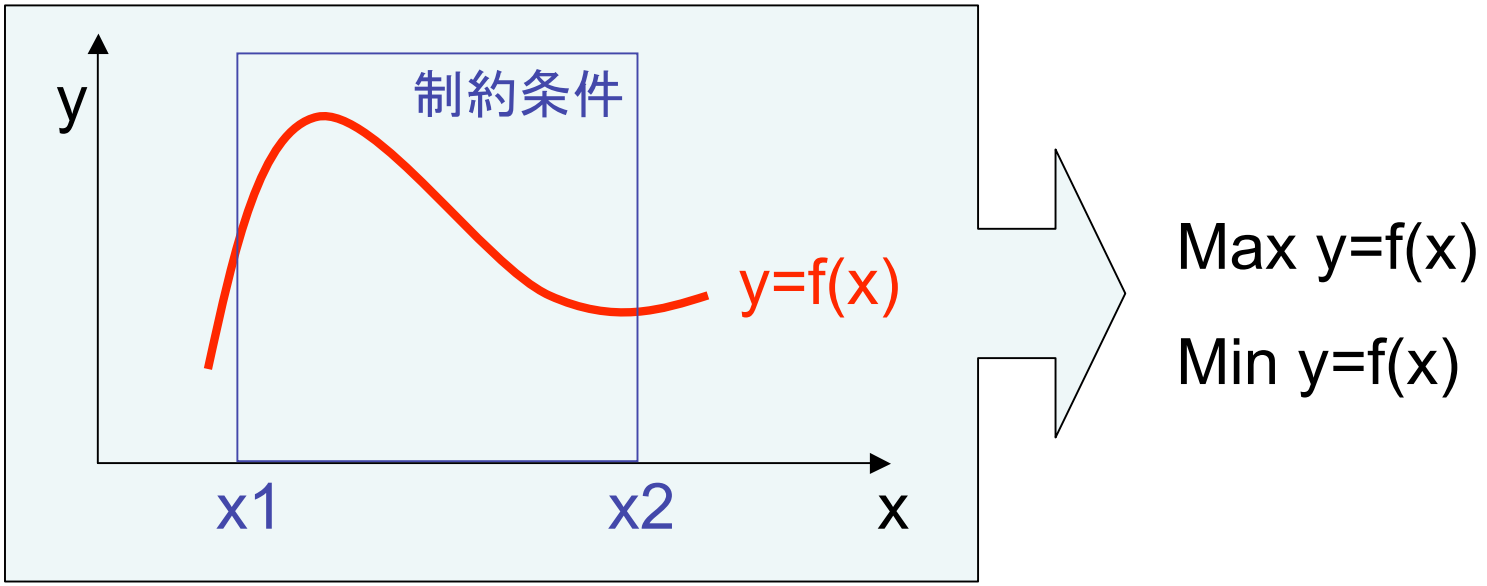
(week1) 背景・基礎知識

1. 探索（最適化）とは
2. 組み合わせ最適化問題と探索アルゴリズム
3. 完全解放と近似解法
4. 評価方法
5. 近似解法例とその特徴

http://www.eva.ie.u-ryukyu.ac.jp/~tnal/2005/info3/

探索（最適化）とは：連続値

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？**

問題空間の離散化

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．

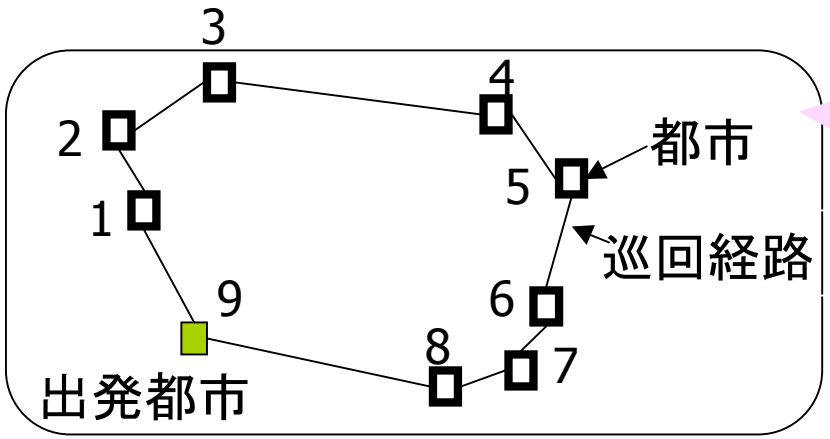
＋

制約条件

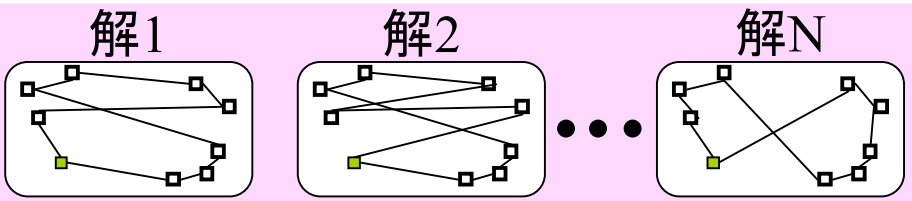
解空間 X は離散集合．

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？

例題：巡回セールスマン問題（TSP）



実行可能解



（目的）巡回に要するコスト（巡回経路長）を最小化せよ．
（制約条件）ただし，与えられた全ての都市はただ一度のみ巡回するものとする．

$$N = \frac{c(c-1)!}{2}$$

c=都市数

TSPの問題サイズ（＝実行可能解の数）

都市数	実行可能解数	1000/1sec
2	1	0
3	3	0
4	12	0
5	60	0
6	360	0
7	2520	2s
8	20160	20s
9	181440	3m
10	1814400	30m
11	19958400	5h
12	239500800	2d
13	3113510400	36d
14	43589145600	1.3y
15	6.53837E+11	20y
16	1.04614E+13	331y
17	1.77844E+14	3,639y
18	3.20119E+15	101,509y
19	6.08226E+16	1,928,670y
20	1.21645E+18	38,573,408y

実際に解きたいサイズは
数100都市〜数1000都市

- 最適解→順最適解
  - より早く
  - より質の高い解
  - 複数代替案

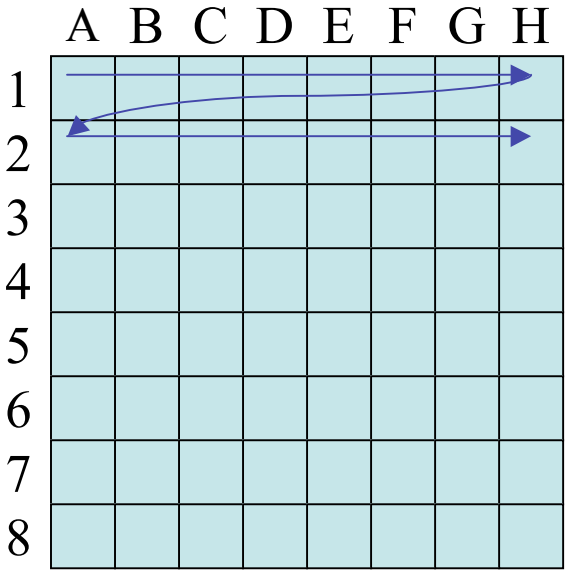
⇩

近似解法

Q: どんな探索手法が考えられるか？
また，その特徴は？

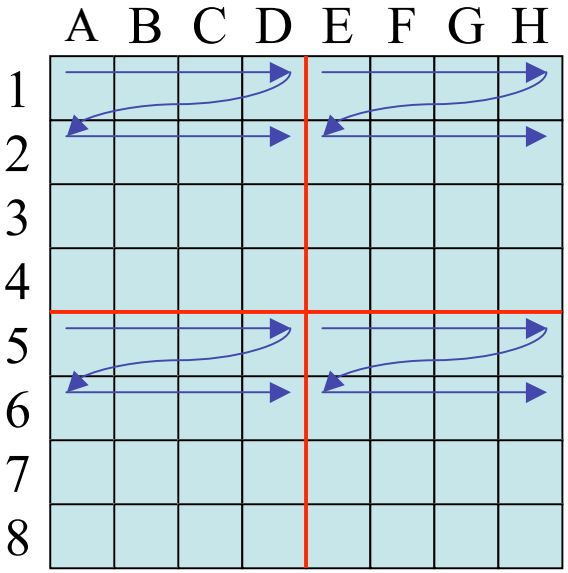
大規模問題を力技で解く！（付録）
並列処理による性能向上

- 逐次処理



8×8 = 64箇所

- 並列処理（e.g., 4分割）



8×8 = 64箇所
64箇所÷4PC = 16箇所/1PC

並列化するには！

1. 依存関係の無い処理を並列化（楽）
2. **依存関係のある処理**を並列化（難）

Q: 依存関係をどう解決するか？

並列処理例(付録)

- 依存関係無し/少
  - SETI@home
    - 空間的に完全に独立して処理可能(グリッド・コンピューティング)
  - バイオインフォマティクス:アミノ酸配列解析
    - 配列単位で処理可能(グリッド, PCクラスタ)
  - 画像処理(の一例)
    - ある画素を中心とした周辺画素を考慮して処理(SIMD, マルチプロセッサ)

- (依存関係あり)
  - 並列処理不可能ではないが, 特別な処理が必要
    - TSP
    - ナップサック問題
    - 車両配送問題
    - ゲーム(チェス・将棋・・・)
    - ロボティクス(行動獲得・姿勢制御・・・)

Q: 世の中にはどのような大規模問題があり, どのように処理しているか?

完全解法

- 全探索
  - 文字通り全ての解について評価し，目的関数を最小/最大とする解を探す.
  - 利点
    - 探索空間をきれいに分割できれば，並列処理によりある程度の規模までは処理可能.
  - 欠点
    - 探索空間が大きすぎると，現実的な解法ではない(＝実行出来ない).

- 線形計画法
  - 線形性を利用した解法
  - 利点
    - 全探索せずとも最適解が獲得出来る
  - 欠点
    - 目的関数・制約条件が線形式で定義される必要がある.

近似解法：立場の違い

- 解の精度を保証する
  - ある特定の問題を対象とした「特化アルゴリズム」
- 精度保証は無いが，（短時間で）なるべく良い解を求める
  - 不特定の問題に対してだいたい良い解を見つける
    - 用語：近似解，順最適解，局所的最適解，大域的最適解

Q: どのように性能評価すると，客観的なデータになりうるか？

- 指定された時間内に良質の解を求める
- 複数の代案を求める
- 多目的最適化

Q: 作ろうとしているシステムにおいて最重要項目（要件）は何か？

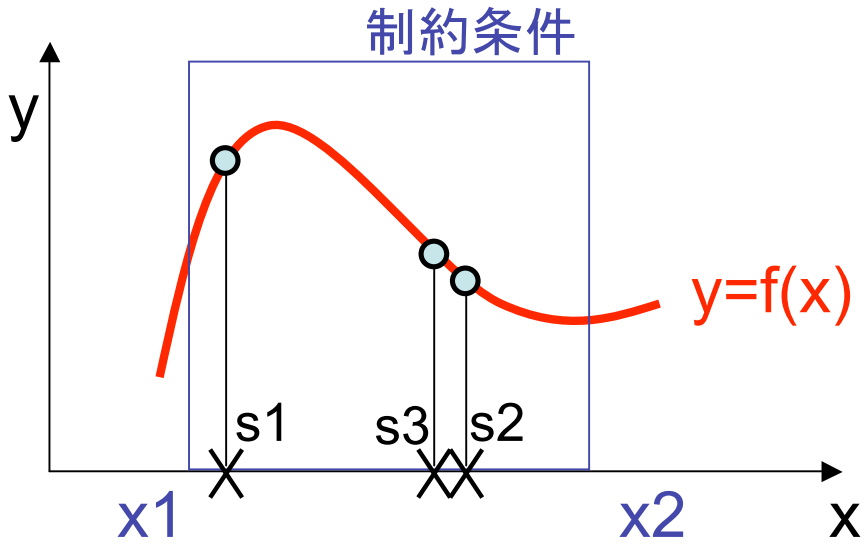
客観的な評価をするために

- 実験方法の明記
  - 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
- 平均/分散/etc.
  - (十分に)複数回実行し,「たまたま良い結果が得られた」例だけを持って良い手法だと論じてはならない.
  - 対象問題そのものもタイプ別に分類可能なら, それぞれについて同様に実験するとなお良い.

Q: 実験を効率よく実行するには, どのような準備をすると良いか?

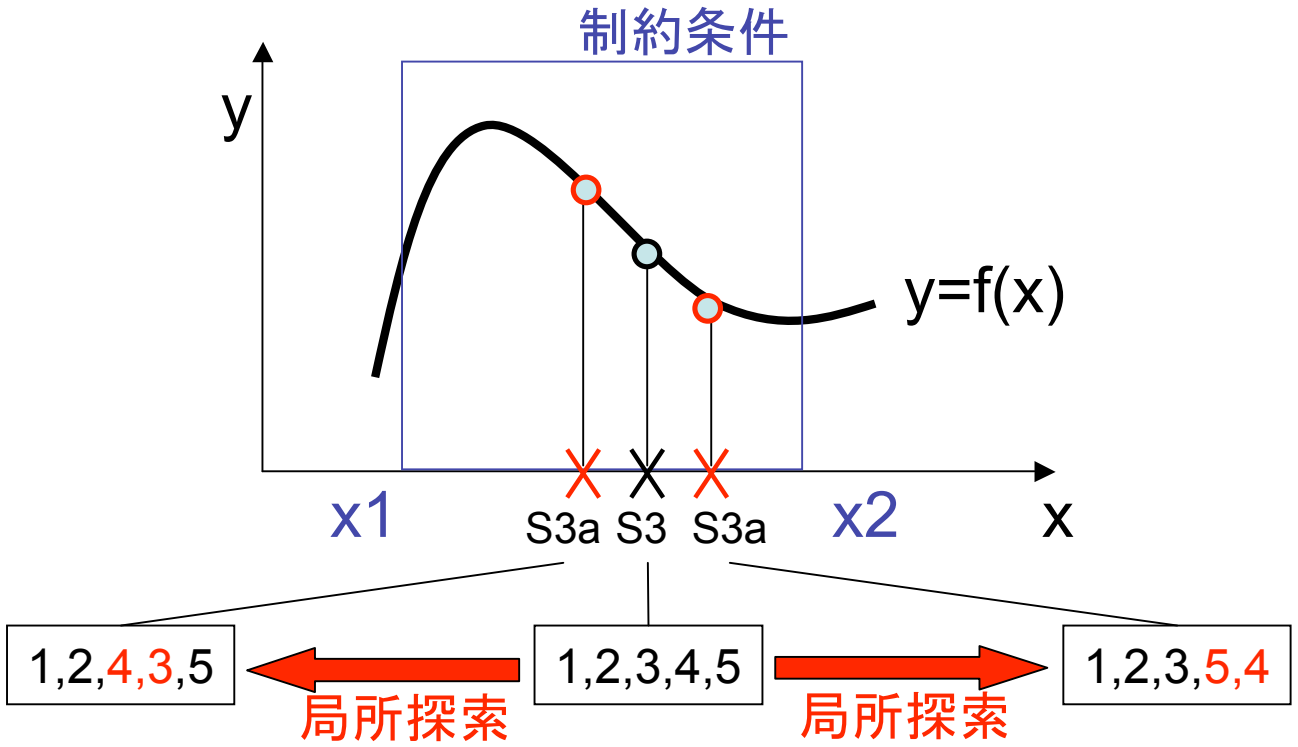
近似解法の種類1(ランダムサーチ)

- 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)

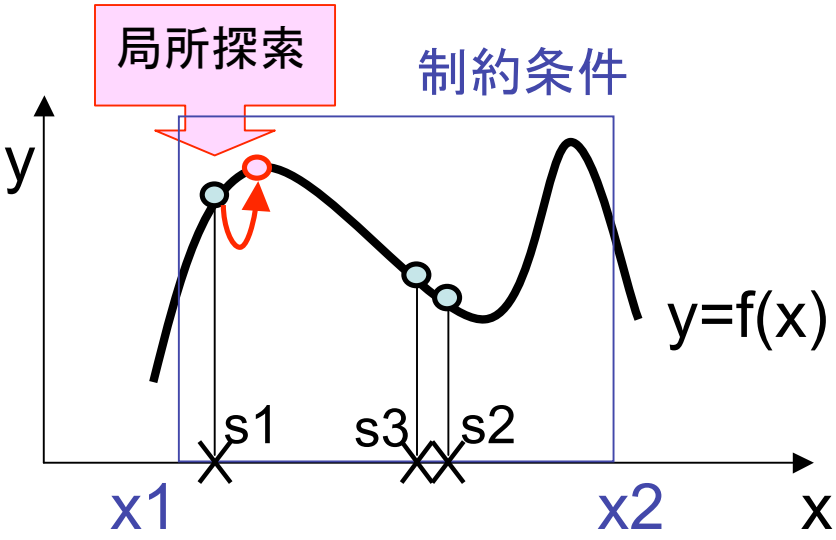
- 山登り法(Hill Climbing method)
  - 近傍のより良い解のある方向を探索
- 焼き鈍し法(Annealing method)
  - 確率的に悪い解へ探索を進める事もある

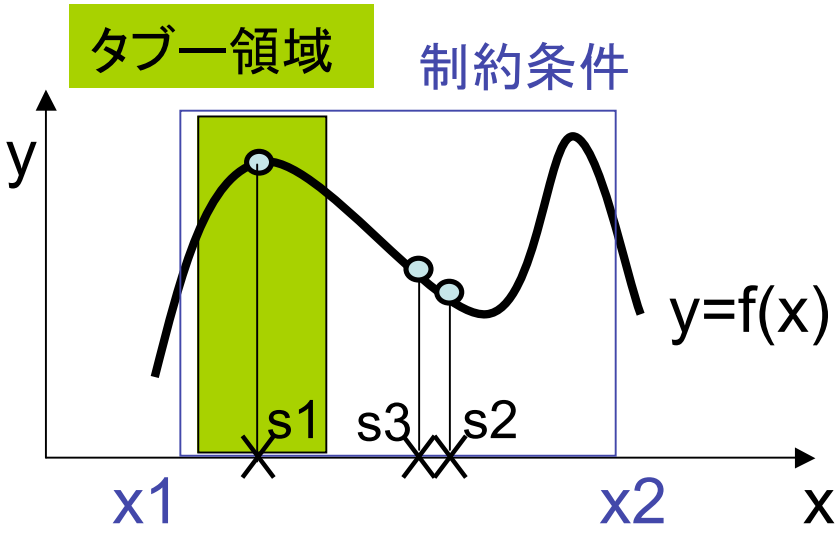


近似解法の種類3(多点探索・タブー)

- 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.





近似解法の特徴(まとめ)

手法	プログラム	速度	精度
ランダム	単純	早い	悪い
山登り法	比較的単純	早い	それなり (良い解が出る事もあるが, 基本的に悪い)
アニーリング	比較的単純	早い	それなり (良い解が出る事もあるが, 基本的に悪い)
多点局所探索	比較的単純	普通	それなりに良い
タブー	やや複雑	遅い	良い (チューニング命)
GA	?	?	?

Q: 他にどのような近似解法があるか？思いつくか？

まとめ

- 組み合わせ最適化問題
- 並列化
- 完全解法と近似解法
- 代表的な近似解法
  - 異なる特徴：速度・精度
  - ハイブリッド手法：特徴の異なる手法を組み合わせる事で，弱点を補う
- 課題
  - http://www.eva.ie.u-ryukyu.ac.jp/~tnal/2005/info3/index.html
  - Level1
    - コンピュータを使って計算（シミュレーション）をするとはどういうことか．何のためにコンピュータを使用するのかを考え，それを実現する方法について検討せよ．また，それらの目的達成のために便利なツールがあればそれを示せ．

次回

- GAイントロダクション
  - 用語
  - コーディング
  - 適応度計算
  - 選択
  - 交叉
  - 突然変異
- 適用例