

シミュレーション論 I

第12回

様々なシミュレーション手法

中間レポートについて

- 以下の内容から**2つ**を選び、自分でシミュレーションを作成・実施してレポートを作成してください
- 全てについて、講義で作成したものに**オリジナルの追加要素を加える**こと
 - (1)モンテカルロ法による円周率の計算
 - (2)つり銭問題のシミュレーション
 - (3)待ち行列のシミュレーション
 - (4)講義で学んだことをもとにしたオリジナルのシミュレーション
- 内容や結果をまとめ、A4用紙5～10枚程度で作成して提出
- 詳細についてはプリントを参照

提出期限:2019年7月11日(木)
16:45までに教学センター レポートボックスへ

第11回のレポート回答例

(例)

- 講義に出席するかどうかのシミュレーション
 - ・セルオートマトン法を用いて、ある講義の出席人数をシミュレーションする
 - ・各セルを受講者とし、隣接するセルを各自の友人と考え、「自分の友人のうち半数がサボったら自分も講義を休む」とする
 - ・初期条件や欠席の条件を変化させ、講義の回数を重ねるごとに欠席者がどのように増減するかをシミュレーションする

今回の内容: 様々なシミュレーション手法

- シミュレーションを行う場合、数式として記述したモデルが解析的に解けなかったり、計算量が膨大なために解くことが困難な場合がある。
- 周囲の状況が動的に変化するような場合には、一度解を求めても随時周囲の状況に応じて修正していく必要がある。
- 上記のような困難な問題に対して、問題の最適化や環境への適応をおこなうシミュレーション手法が提案されている。

生物にヒントを得たシミュレーション手法

- 困難な問題に対するシミュレーション手法のうち、生物の仕組みや能力にヒントを得たものがある。
- 進化の仕組み：遺伝的アルゴリズム等
- 脳の仕組み：ニューラルネットワーク等
- 学習など：強化学習法等

遺伝的アルゴリズムとは

- 遺伝的アルゴリズム (Genetic Algorithm : GA)
 - 1960年代後半～70年代、John Hollandにより提案
 - 生物の進化のメカニズムを応用した問題の最適化手法
 - 問題の解を「遺伝子」としてコンピュータ内に作成し、形質の遺伝、淘汰、突然変異などのメカニズムを模して最適解を探索する



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

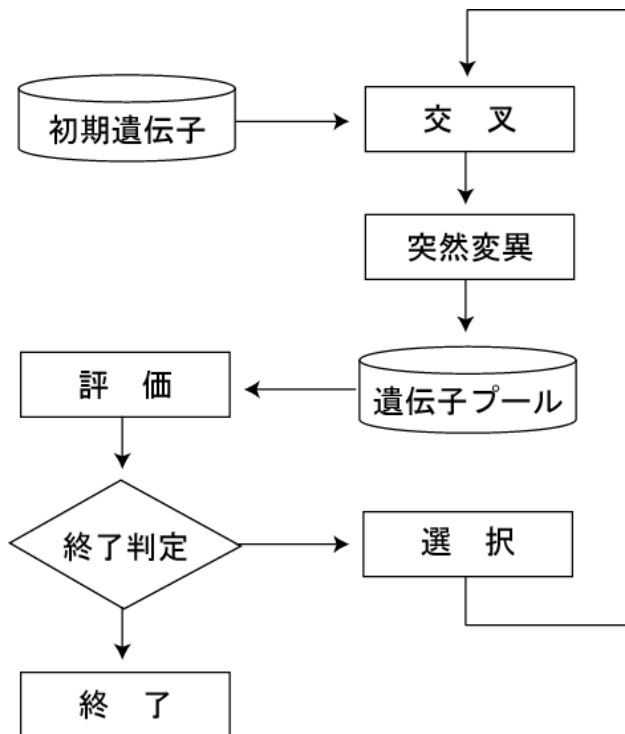
1	1	1	0	1
---	---	---	---	---

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

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

遺伝的アルゴリズム(1)

- 遺伝的アルゴリズム(GA)の代表的な手順は、以下のようになる



1. 問題の解の候補を「遺伝子」としてランダムに発生させる
2. 遺伝子の**交叉**によって遺伝子プールに新たな遺伝子を格納する
3. 交叉の際、一定の確率で**突然変異**を起こす
4. 対象とする問題から「適応度」の計算をおこない、評価の高いものを残す(**淘汰**)

遺伝的アルゴリズム(2)

- 遺伝子の表現: GAにおける遺伝子の表現方法は、対象とする問題に応じてビット列など、数値やアルファベットの列として表現されることが多い。
- ビット列のような遺伝子の表現を「遺伝子型 (genotype)」、遺伝子を対象とする問題にあわせて表現したものを「評価型 (phenotype)」という。

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

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

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

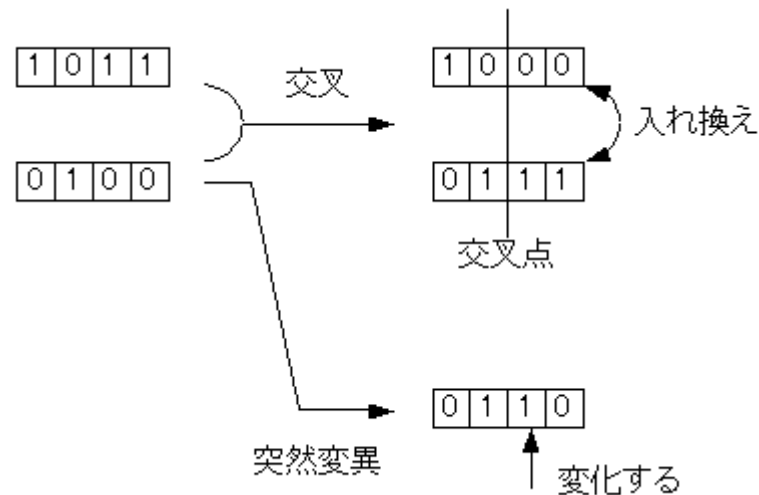
1	0	0	0	0
---	---	---	---	---



2, 545, 32, 8 ...

遺伝的アルゴリズム(3)

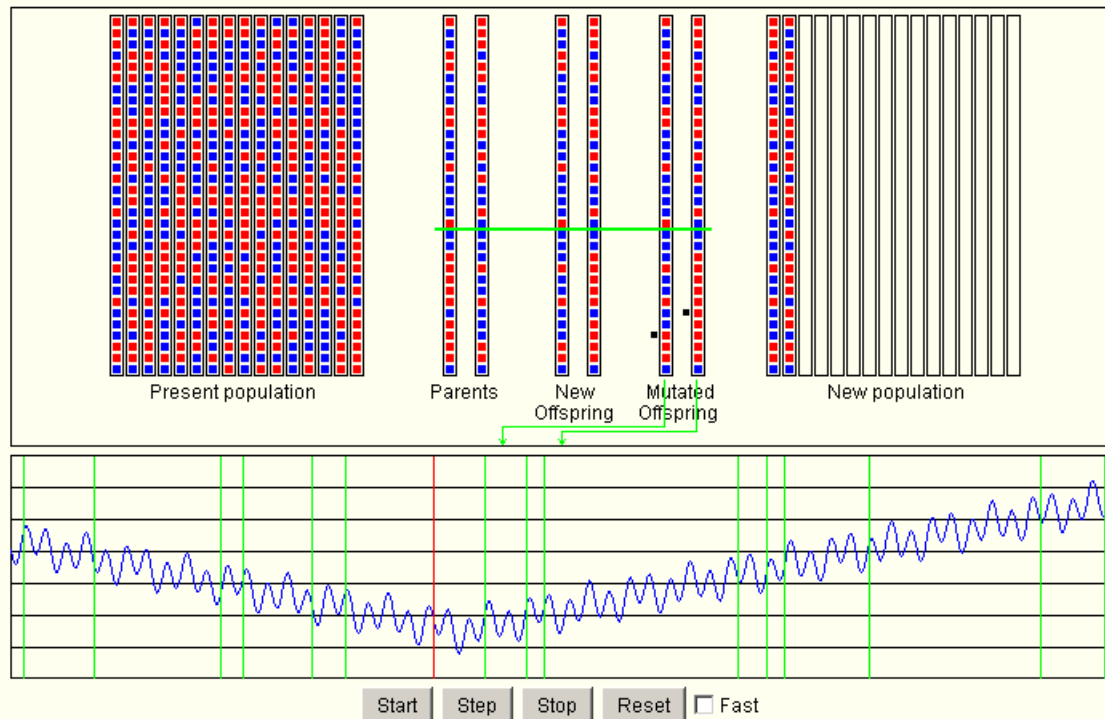
- 遺伝的アルゴリズムにおける基本操作:
 - 交叉: 一对の遺伝子をある部分で切り取ってつなぎ合わせ、新たに子となる遺伝子を作成する
 - 突然変異: 遺伝子座の一部を入れ替えたり、他の遺伝子座と取り替えたりする
 - 選択: 遺伝子の適応度(問題に対する評価値)をもとに、よりよいものを選んで残す



遺伝的アルゴリズムのサンプル

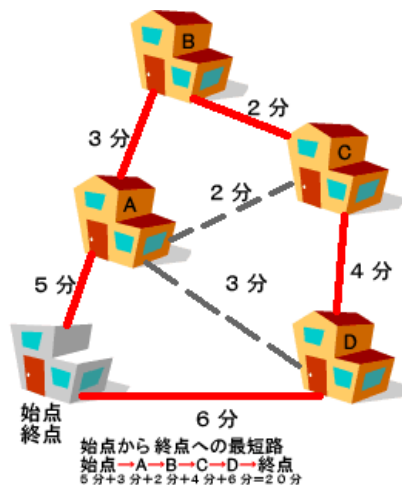
- 多峰性関数の最小値の探索

<http://www.obitko.com/tutorials/genetic-algorithms/index.php>



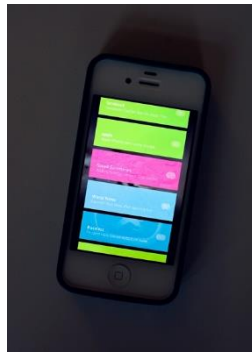
遺伝的アルゴリズムの応用

- 遺伝的アルゴリズムは解析が困難な最適化問題の解法として用いられることが多い
- 巡回セールスマン問題 (Traveling Salesman Problem: TSP): セールスマンが都市を巡回する際に、もっとも効率的な順序を求める問題
 - 6都市→720通りの経路、15都市→約1兆3076億通りの経路
 - 単純な総当り解法ではとても解けない



遺伝的アルゴリズムの応用(2)

- ブログ等の内容を自動で要約するニュースアプリ「Summly」
- 当時15歳のニック・ダロイジオが開発、米ヤフーが3000万ドルで買収



- ランダムな四角形を進化させて画像を描く

Original Best Evolving

93.18% Fitness
1710 Improvements
34462 Mutations

Start

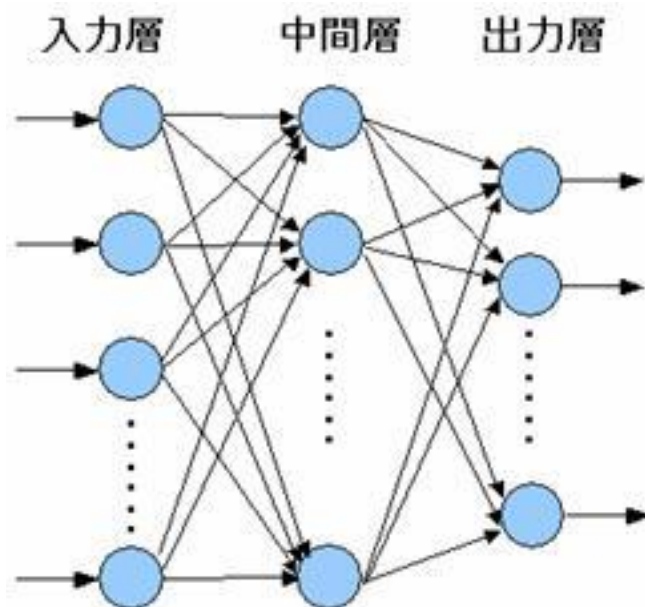
4m 59s Elapsed time
217.6 Mutations per second

Set new image URL

Image should be small (~200x200 pixels), otherwise fitness computation is very slow.

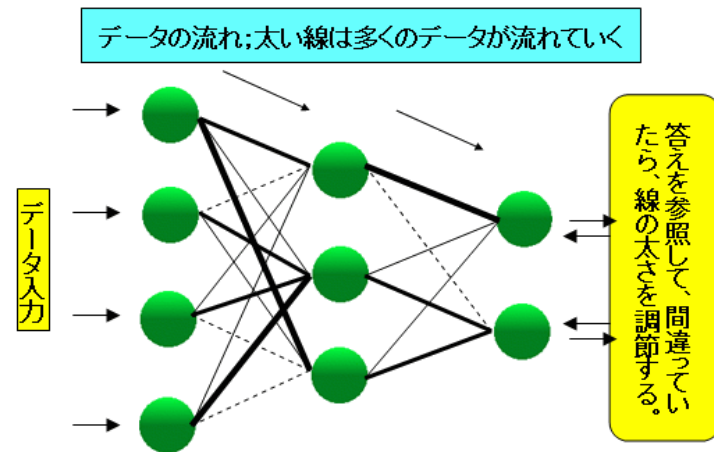
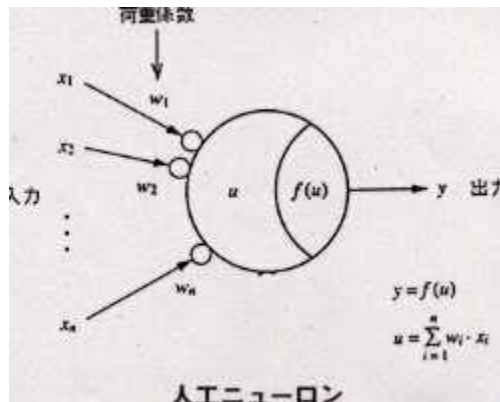
ニューラルネットワーク

- ニューラルネットワーク: 脳の機能をモデル化し、神経細胞(ニューロン)を模した「ユニット」の相互結合とそれぞれの結合荷重によって、目的とする「入力→出力」を生み出すネットワーク構造を作成する



ニューラルネットワーク(2)

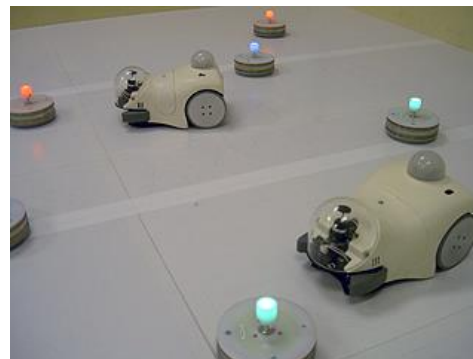
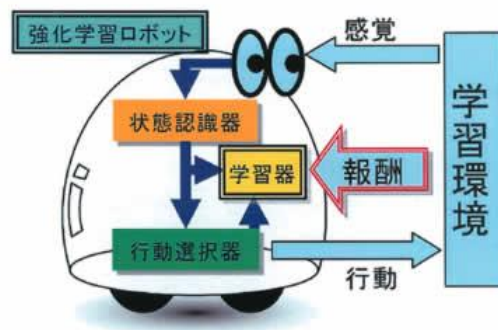
- ニューラルネットワークの各ニューロンは、入力値に「荷重」をかけて計算し、出力する機能をもつ
- 入出力の関係をあらかじめ記述した「教師データ」によりネットワークの結合や荷重を調整し、ネットワークを学習させていく



- <http://www.ritsumeai.ac.jp/se/~watanabe/HTML/SAMPLE/sampleplograms.html>

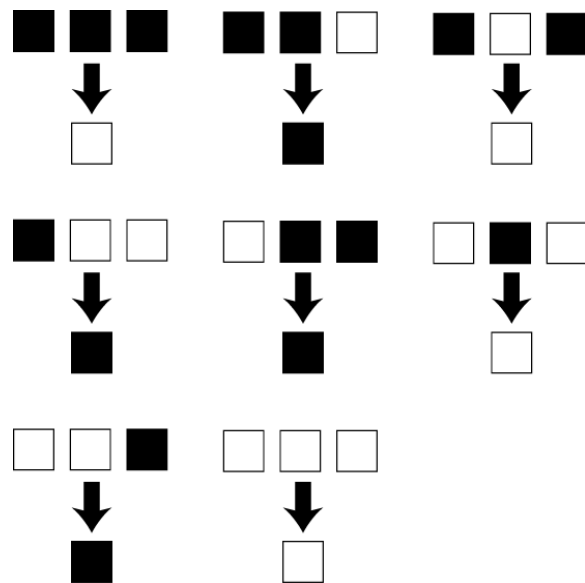
強化学習

- 強化学習：試行錯誤をくりかえして、よりよい行動方針を獲得する手法
- 状態と行動をセットにして記述し、うまくいった場合に「報酬」、失敗した場合に「罰」を与えることでよりよい行動を獲得するようになる
- 教師データが不要なため、未知の環境への応用が可能
- ロボットの行動獲得などによく利用される



Excelによる1次元セルオートマトン

- 以下の推移規則にしたがって、1次元セルオートマトンを作成してみましょう
- 初期状態では中央に黒が1つあるものとします



- ノートPCをお持ちでない方は説明を聞いたのち、別課題をやってください

Excelによる1次元セルオートマトン(2)

- Excelの if 文を使って推移規則を記述する
- 黒を1、白を0として数字で記述

(1) 一行目に0を入力(A~T列) 一番左と一番右は計算用セルとして使うので2行目以降の計算式を入力しない

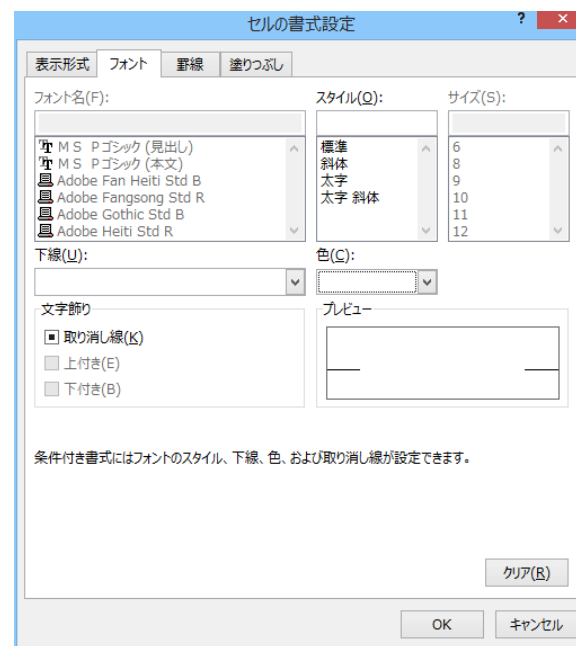
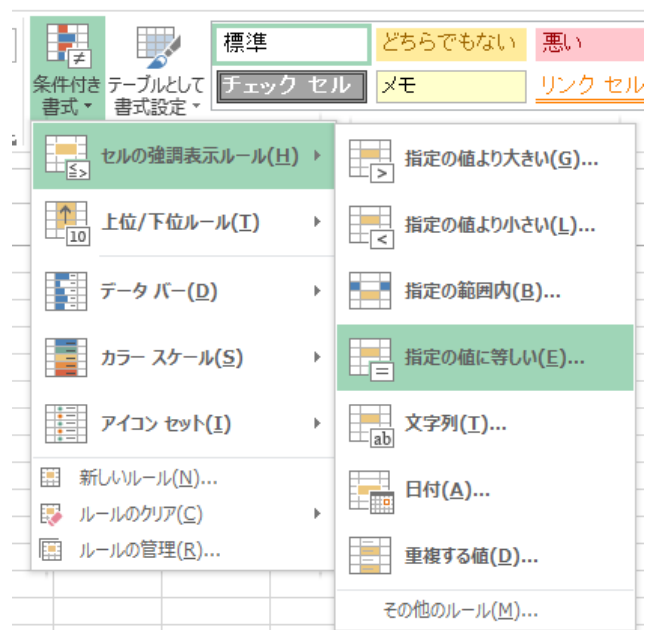
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2																						
3																						
4																						
5																						

(2) B2セルに以下の数式を入力

```
= IF(AND(A1=1,B1=1,C1=1),0, IF(AND(A1=1,B1=1,C1=0),1,  
    IF(AND(A1=1,B1=0,C1=1),0, IF(AND(A1=1,B1=0,C1=0),1,  
    IF(AND(A1=0,B1=1,C1=1),1, IF(AND(A1=0,B1=1,C1=0),0,  
    IF(AND(A1=0,B1=0,C1=1),1, 0)))))))))
```


Excelによる1次元セルオートマトン(4)

- (4)「条件付き書式」→「セルの強調表示ルール」→「指定の値に等しい」を選んでセルに色をつける
- 値を「0」、書式を「ユーザー設定の書式」とし、フォントの色とセルの背景色を白にする
 - 同様に、値が「1」のときフォントとセルの色を黒にする



Excelによる1次元セルオートマトン(5)

- (5) 一行目の好きなところに「1」を入力してみよう
その部分が黒になり、パターンが現れる

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1										1										
2									1	1	1									
3									1	1	1	1	1							
4									1	1	1	1	1	1						
5									1	1	1	1	1	1	1					
6									1	1	1	1	1	1	1	1				
7									1	1	1	1	1	1	1	1	1			
8									1	1	1	1	1	1	1	1	1	1		
9									1	1	1	1	1	1	1	1	1	1	1	
10									1	1	1	1	1	1	1	1	1	1	1	
11									1	1	1	1	1	1	1	1	1	1	1	
12									1	1	1	1	1	1	1	1	1	1	1	
13									1	1	1	1	1	1	1	1	1	1	1	
14									1	1	1	1	1	1	1	1	1	1	1	
15									1	1	1	1	1	1	1	1	1	1	1	

※ 余裕があれば右や下へ伸ばしてみよう

第12回のレポート

- ごく簡単な遺伝的アルゴリズムを用いて、関数の最小値を求めてみましょう
- 問題: 以下の関数の**最小値**とそのときの x を $0 \leq x \leq 15$ の範囲で求める

$$f(x) = (x-1)(x-7)(x-10)(x-13)$$

- 遺伝子は4ケタのビット列を5つとし、一点交叉と突然変異を使ってやってみよう
- 選択は「評価値の高い(関数の値が小さい)」ものを上から2つ残し、交叉させて新たに2つ子を作り、一番評価値の高かったものの一部を突然変異させて再度5つにします

レポートの手順

- 1. 4ケタのビット列(2進数)を適当に5つ作成
- 2. ビット列を10進数に変換し、 $f(x) = (x-1)(x-7)(x-10)(x-13)$ に代入
- 3. 最も $f(x)$ の値が小さくなるものから順に2つを残し、残りを削除(淘汰)
- 4. 残った2つを適当なところで交叉させ、新たに2つの子遺伝子を作成
- 5. 1つ遺伝子を選び、うち1~3箇所を0と1を入れ替える(突然変異)
- 6. 上記の手順を繰り返す、最も $f(x)$ の値が小さくなるものを抽出

