

アルゴリズム論

平成22年11月29日

Genetic Algorithm [2]



課題 last

- ◆ [課題 last] **NP 困難な最適化問題を一つ選び、その近似解を遺伝アルゴリズムを用いて求めるプログラムを実現せよ。**

染色体表現は、二進数列・順列のいずれのモデルでもよい。

プログラムを示すだけでなく、問題の記述と適応度計算関数 fitness の解説は必須である。

probX, probM, Npop, Times などのパラメータを適切に調整する過程の実験データが提示されることを望む。

どんな NP 困難問題を予定していますか？

遺伝(的)アルゴリズム

◆ Genetic Algorithm (GA)

- 最適化問題の汎用的近似解法 meta-heuristics
- 複数の実行可能解を保持
- 解を生物の進化のアナロジーを用いて改善

◆ 近似解法 \Leftrightarrow 生物進化 [対比の例]

- 実行可能解 \Leftrightarrow 染色体(遺伝子の列)
- 第 t 反復の解の集合 \Leftrightarrow 第 t 世代の個体集団
- 新たな解の生成 \Leftrightarrow 交叉／突然変異
- 実行不可能な解 \Leftrightarrow 致死遺伝子(をもつ染色体)
- 目的関数(最大化) \Leftrightarrow 適応度

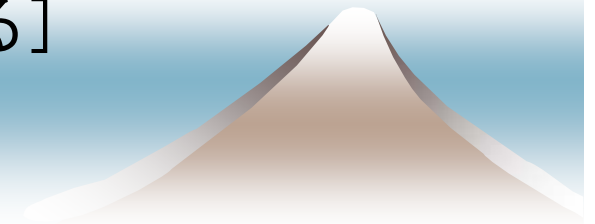
染色体は遺伝子の並び

◆ 染色体表現

- 二進数列：0 と 1 の並び
 - … 遺伝アルゴリズム研究は二進数列から始まった
- 三進数列：例えばGとCとP[ジャンケンゲーム]
- 順列：重複しない整数値の並び
 - … 古典的な GA と区別して，進化アルゴリズム
EA: evolutionary algorithm と呼ばれることもある

◆ 染色体の長さは，通常は，一定

[可変長の染色体を扱った研究もある]

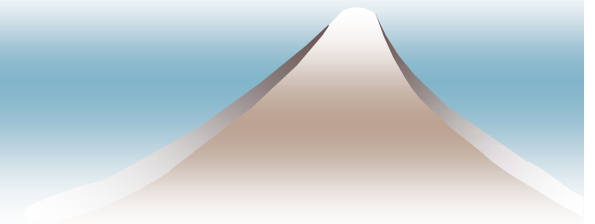


染色体の例

◆ 長さが5の場合の例：

- 二進数列 01100, 11001, 00000, 10111
- 三進数列 GGCPG, CCCCC, CGGPC, GCPGC
- 順列 23514, 12345, 54213, 32541

- 長さが n の二進数列は： 2^n 個ある
- 長さが n の三進数列は： 3^n 個ある
- 長さが n の順列は： $n!$ 個ある



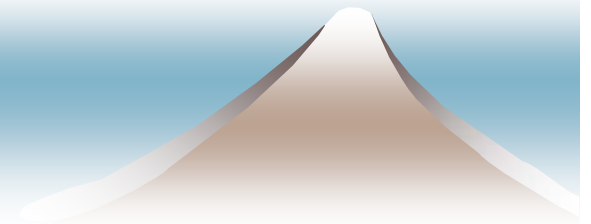
二進数列の染色体

□ 最大充足可能性問題 MAXSAT

[入力] n 変数よりなる m 個の和項 C_1, C_2, \dots, C_m
および各和項の重み w_1, w_2, \dots, w_m

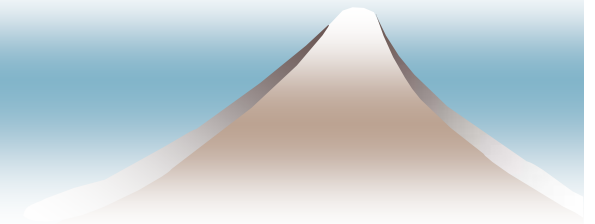
[出力] $f_{\text{MAXSAT}}(v) = \sum_{k=1}^m w_k C_k(v)$ を最大にする
各変数への 0-1 の割り当て $v \in \{0,1\}^n$

長さ n の二進数ベクトル $v \in \{0,1\}^n$ が一つの可能解
これをそのまま染色体の二進数列表現とすることが可能



GA の一般的な流れ

```
def GeneticAlgorithm(...): # 3箇所を二進数列バージョンに
    ... 個別問題の設定 ... # Length: 染色体の長さ
    ... GA パラメータの設定 ...
                                # Npop: 個体数, Times: 終了世代数
                                # probX: 交叉確率, probM: 突然変異確率
    group = Binititalize(Length, Npop) # 初期集団生成
    for k in range(Times):
        evaluate(group, ... 個別問題 ...) # 適応度の評価
        eliteChr = elite(group) # エリートを記憶
        prntAll(group, k)
        group = roulette(group) # 次世代をルーレット選択
        Bcrossover(group, probX) # 交叉
        Bmutation(group, probM) # 突然変異
        elitePrs(group, eliteChr, 2) # エリートを2個保存する
    ... 結果の出力 ...
    return
```



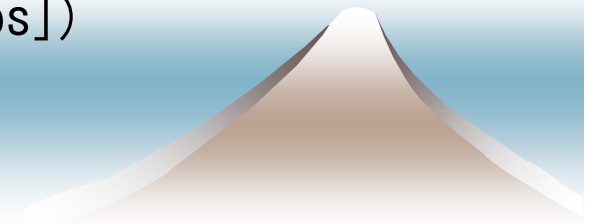
Binitialize

```
def Binitialize(Length, Npop):  
    """ 二進数列用初期集団生成 """  
    group = []  
    for k in range(Npop):  
        chromo = [random.choice([0, 1]) for j in range(Length)]  
        chromo.append(0)  
        group.append(chromo)  
    return group
```



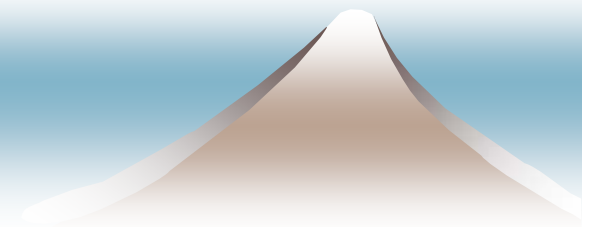
Bcrossover

```
def Bcrossover(group, probX):  
    """ 二進数列用 2 点交叉 """  
    posS = range(len(group[0]) - 1)  
    for k in range(0, len(group), 2):  
        if random.random() > probX:  
            continue  
        pos2 = random.sample(posS, 2)  
        posL = range(min(pos2), max(pos2))  
        BXbasic(group[k], group[k + 1], posL)  
    return  
  
def BXbasic(papa, mama, posL): # 2 点交叉  
    for pos in posL:  
        papa[pos], mama[pos] = (mama[pos], papa[pos])  
    return
```



Bmutation

```
def Bmutation(group, probM):  
    """ 二進数列用突然変異 """  
    posS = range(len(group[0]) - 1)  
    for chromo in group:  
        if random.random() > probM:  
            continue  
        pos = random.choice(posS)  
        chromo[pos] = 1 - chromo[pos]  
    return
```



課題 last

- ◆ [課題 last] **NP** 困難な最適化問題を一つ選び、その近似解を遺伝アルゴリズムを用いて求めるプログラムを実現せよ。

染色体表現は、二進数列・順列のいずれのモデルでもよい。

プログラムを示すだけでなく、問題の記述と適応度計算関数 `fitness` の解説は必須である。

`probX`, `probM`, `Npop`, `Times` などのパラメータを適切に調整する過程の実験データが提示されることを望む。

提出締切 12月5日(日)午後11時59分