

情報工学実験4 ~進化計算班~

Boy meets Girl ~ GPで解く ~



メンバー

095703J 岩瀬翔

095733A 砂川 昌平

要求仕様

スーパーマリオのような、2Dのフィールドを用意し、その中にプレイヤー、ゴール、障害物を配置する。
フィールドのイメージは、暗い洞窟である。
プレイヤー（探検家）は、ヘッドライトを付けており、ヘッドライトで見える範囲のものを認識することができる。
ヘッドライトの電池は限られているので、それを制限時間とする。認識したものを手がかりとして、プレイヤーは自らの行動を決定し、ゴール（お宝）を目指す。

プレイヤー

ー行動

前進／後退する

停止する

ジャンプする（角度,力）→ジャンプの力は上限がある。

ーヘッドライト

ヘッドライトの向き（角度）

ヘッドライトで見える範囲は限られていて、そこにあるものしか認識できない

ープレイヤーが知り得ること

ヘッドライトの電池の残量

ライトで照らした物体の認知、距離

障害物

ー穴

落下すると終了する。

ートゲ

接触すると、終了する。

ーブロック

動かすことはできない。乗ることができる。

ークリーチャー

様々な方向に移動し、探検家が当たると終了する。

ゴール

ーライトを当てない限り、場所はわからない。

特に譲れない部分

- ・ ヘッドライト

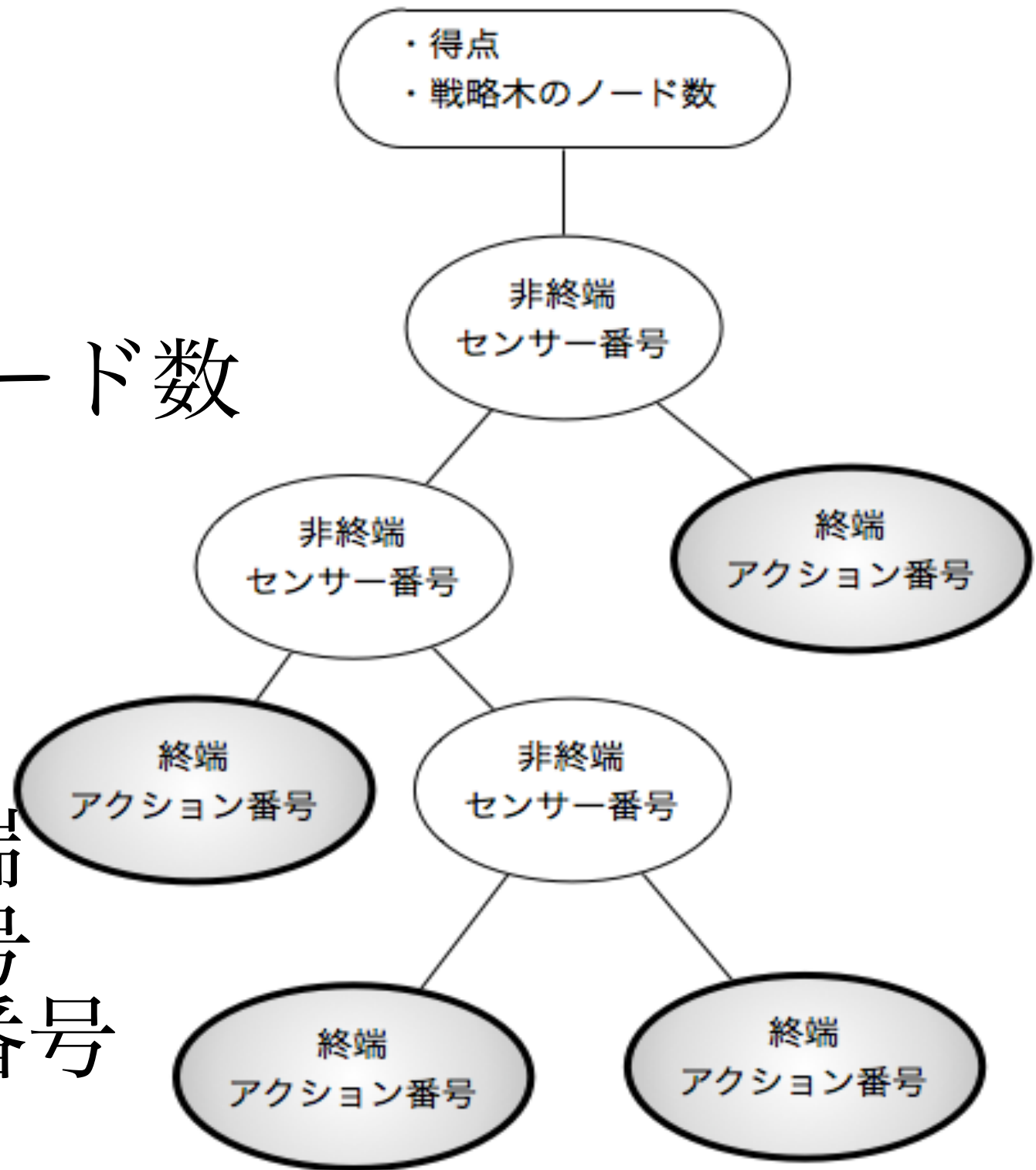
GP設計

タグ

- ・ 適応度
- ・ 戦略木のノード数

ノード

- ・ 終端or非終端
- ・ センサー番号
- ・ アクション番号



GP設計

- ・ 非終端ではセンサー
センサーがヘッドライトの代わり

各種センサー

計11個

| | | |
|-----------|------------|-----------|
| センサー A | センサー B | センサー C |
| センサー D | BOY | センサー E |
| センサー F | センサー G | センサー H |

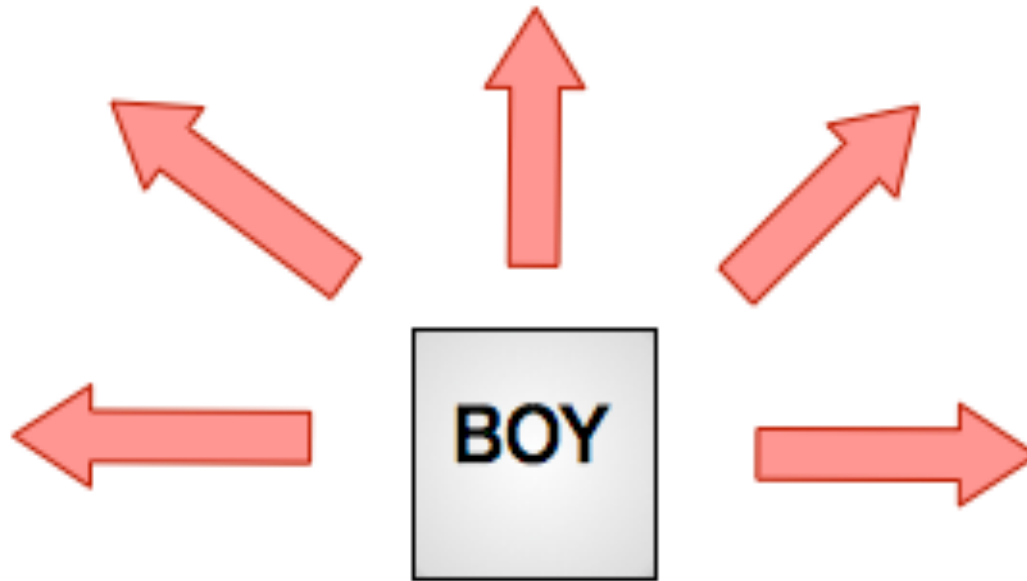
| |
|-------------------|
| ジャンプ センサー I |
| タイム センサー J |

| | | | |
|------------|-------------------|-------------------|-------------------|
| BOY | エネミー センサー K | エネミー センサー K | エネミー センサー K |
|------------|-------------------|-------------------|-------------------|

GP設計

- ・ 終端ではアクション

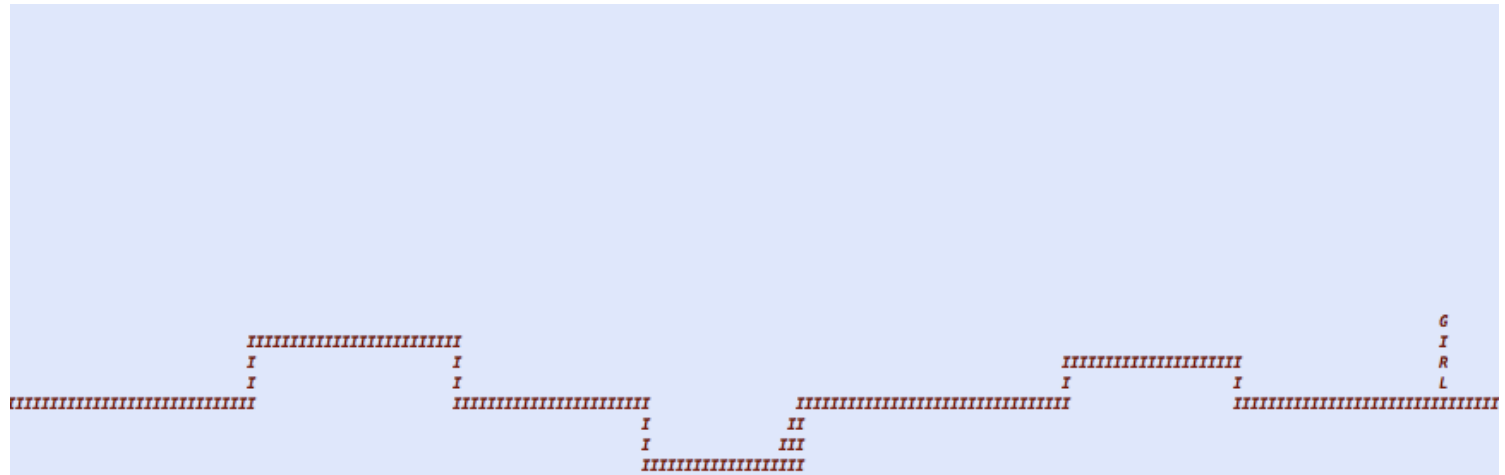
(左) (左上) (上) (右上) (右)



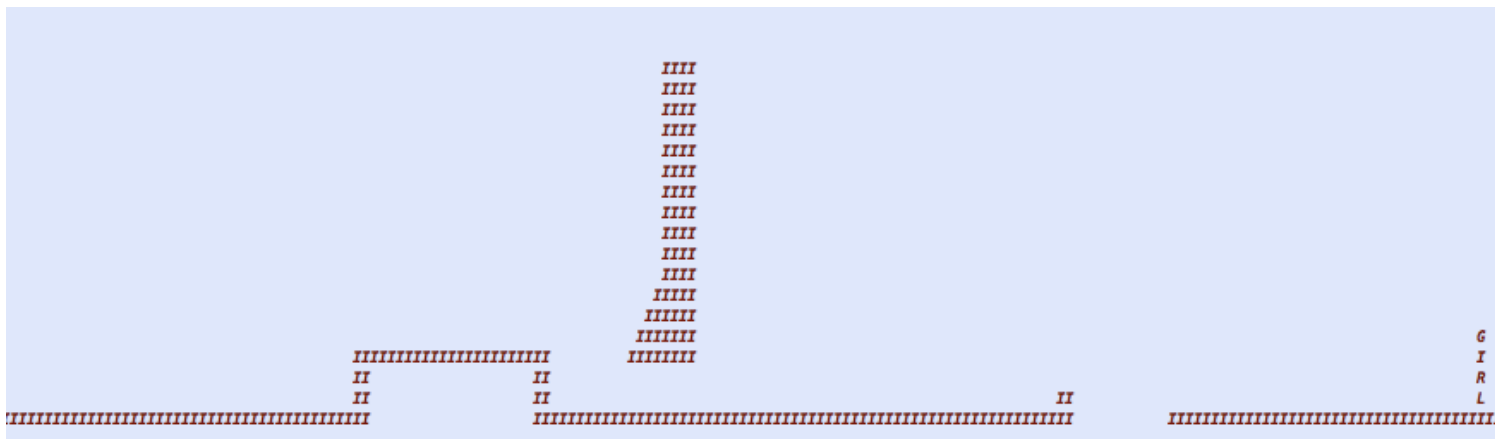
GP設計

適応度計算

静的に用意したステージを戦略木を用いて Girl に会えたかどうか判定
会えたら、残りの Timelimit が得点(適応度)となる



(上)ステージ4



(左)ステージ6

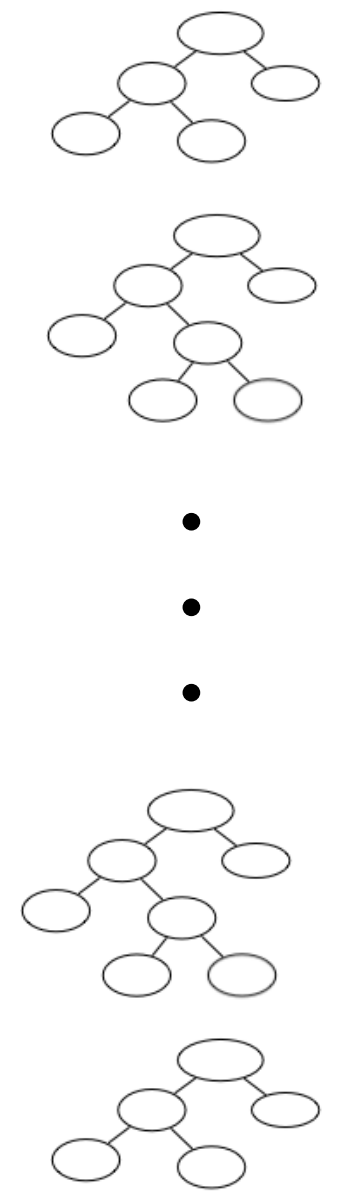
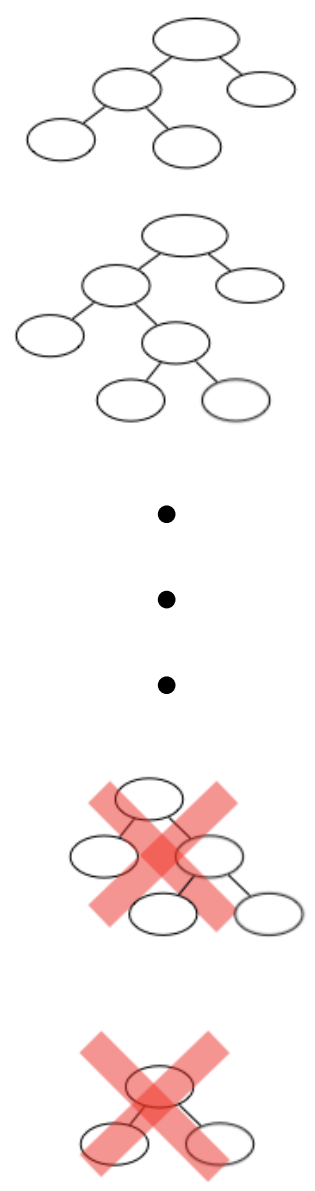
GP設計

選 択

適応度(高)



適応度(低)

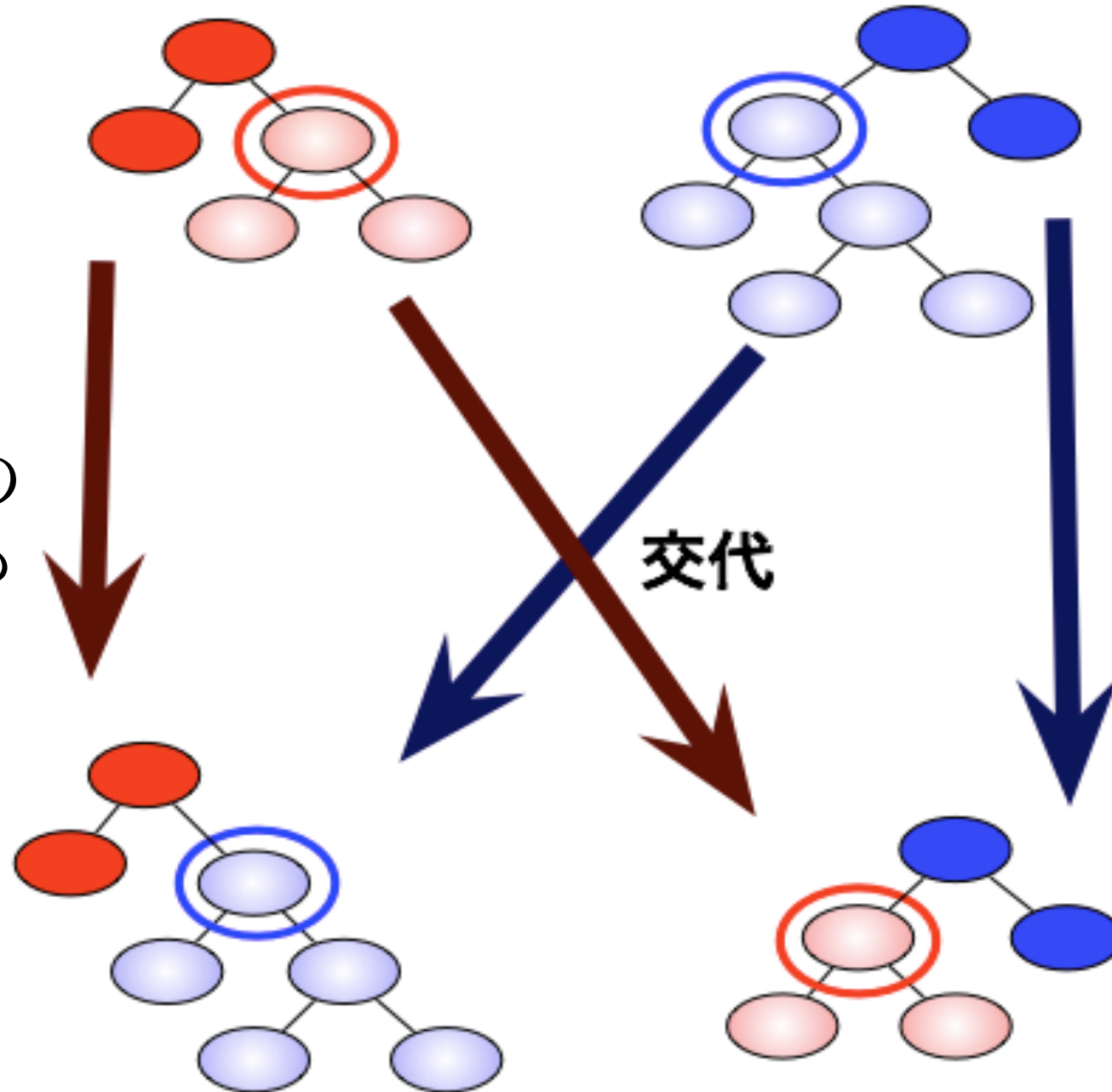


GP設計

交叉

交叉対象木
はランダム

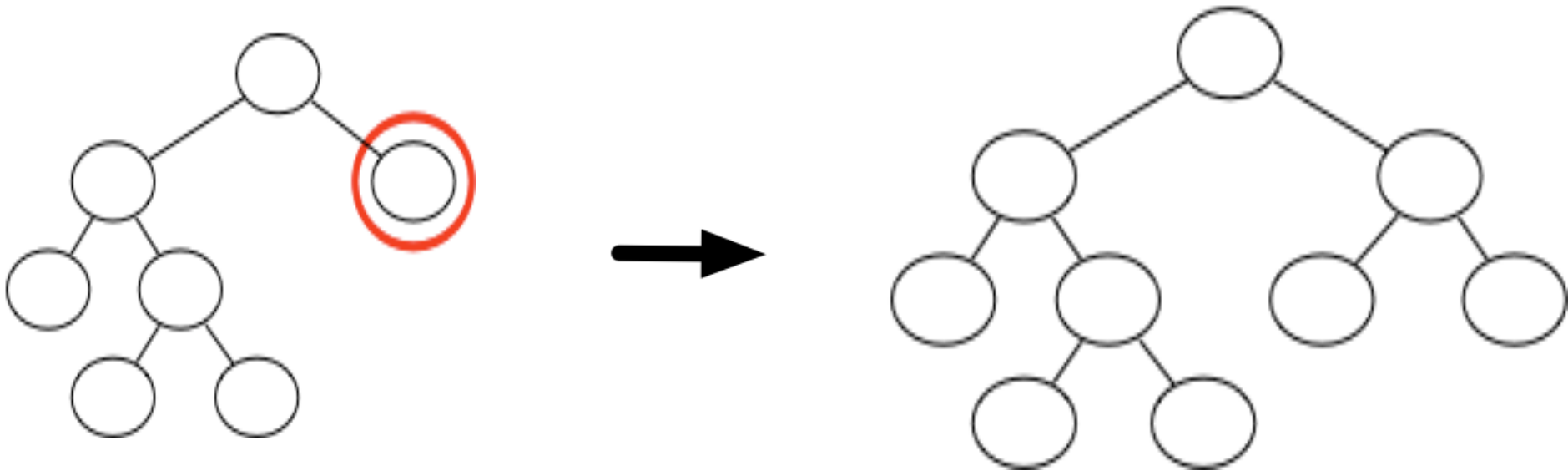
交叉対象木の
ノード選択も
ランダム



GP設計

突然変異

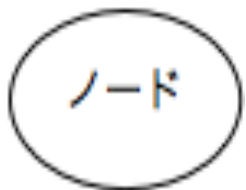
各個体が低い確率で、突然変異を起こす。
選択ノードはランダム(ルートノードも選択
される可能性有り)
深さは指定できない(0~5)



GP設計

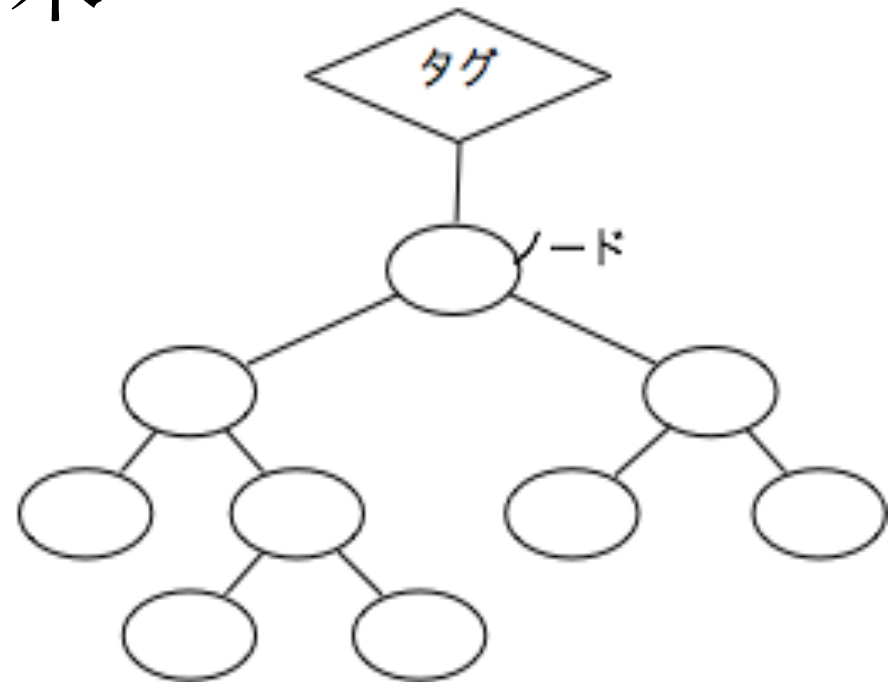


```
typedef struct _tag {  
    int inodeNUM;  
    int iPOINT;  
    node_t *p_strTOP;  
} tag_t;
```



```
typedef struct node_t {  
    bool bEND;  
    int iACTION;  
    int iSENSOR;  
    struct node_t *p_strRIGHT_CHILD;  
    struct node_t *p_strLEFT_CHILD;  
} node_t;
```

パラメータ
～ 戦略木 ～



GP設計

パラメータ ~ Boy meet Girl ~

```
typedef struct Agent_status {  
    int X;  
    int Y;  
    int INTERTIA;  
    int TIMES;  
    int END_FLAG;  
    int TOTAL_POINTS;  
    int ON_OFF;  
    int Jumpflag;  
    int ENEMY_TOUCH_FLAG;  
} Agent_status;
```

GP設計

実験で用いた確率

- ・選択

上位10%を下位10%にコピー

- ・交叉

交叉対象は全体の20%とする

- ・突然変異

各個体の突然変異率は2%

実行方法

- 第一引数 : ステージの最大個数
- 第二引数 : Manual->1、Auto->2
- 第三引数 : 個体数
- 第四引数 : 世代数
- 第五引数 : 低速->0、高速->1

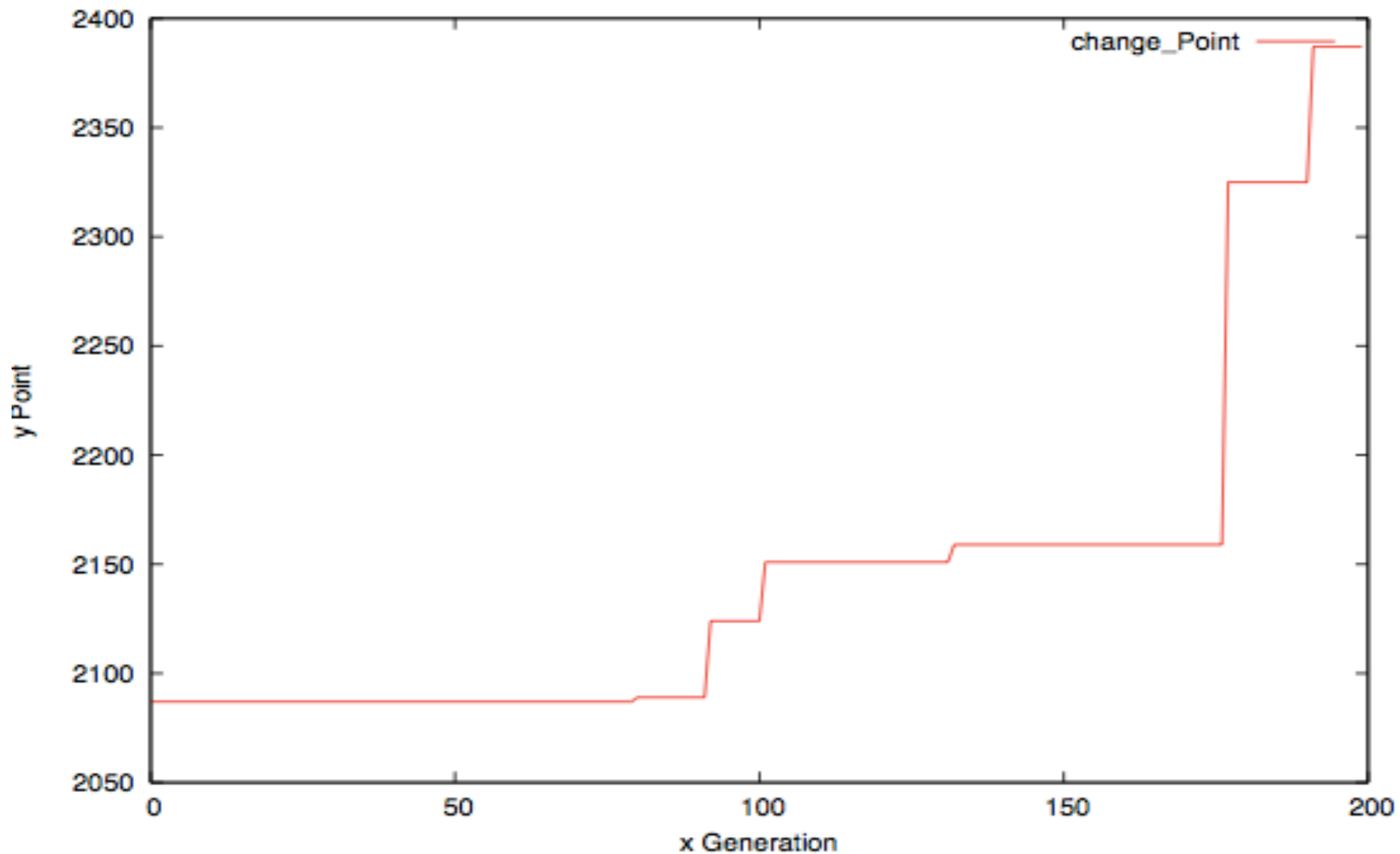
今回

```
./simulator 10 2 80 200 1
```

実行結果



1位の得点の推移図



時間の都合上、省いた点

- ユーザ側から見たらゴール地点がまるわかり
- クリーチャー
- GUI