

ナップザック問題 for GA

学籍番号 045713C:大城和也

提出日:平成 18 年 07 月 25 日 (火)

1 課題内容

GA でナップザック問題を解く．その際 (1) 各遺伝子操作オペレータ (選択, 交叉, 突然変異) を実装した場合と (2) 実装しない場合, について実験を行い, 得られた結果の違いから各オペレータの役割を考察せよ．

2 ナップザック問題とは

ナップザック問題とはいくつかの荷物と一定容量の荷物を入れる袋がある時, 容量を超えないで価値の和が最大になる組み合わせを決める問題のことである．ナップザック問題は荷物が n 個の場合それぞれの荷物を入れる入れないを考えるので, 解の候補は 2^n となり, この問題の完璧な最適化を行うには全探索をする必要がある．

3 実行結果と考察

プログラムは以下からダウンロードしたものを使用する．

<http://www.eva.ie.u-ryukyu.ac.jp/%7Eetnal/Job/GA/Readme.html>

今回のプログラムの実行方法は

```
./run_ga.sh 1 1 KP_List.data 100
```

と統一し, このとき出力される 1-1-KP_List.data-100.max という適応度の最適値を出力したものをプロットしている．

3.1 全てを実装した場合

通常通りに make をしたものである．これは以下の様になる．問題が簡単であるため十回程度でちゃんと進化していることが見て取れる．

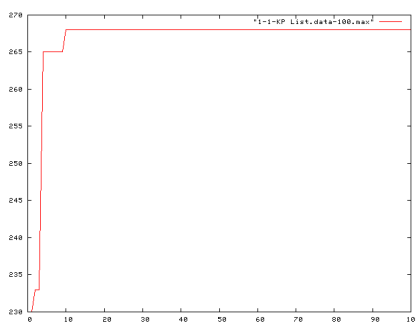


図 1: 全てを実装した場合

3.2 選択だけの実装

図2を見ればわかるように、選択だけを実装したとき、その値に変化は全くない。これは、選択によって初期の遺伝子のなかで値の良いものは増殖するが、新しく遺伝子を作ることが無いためである。

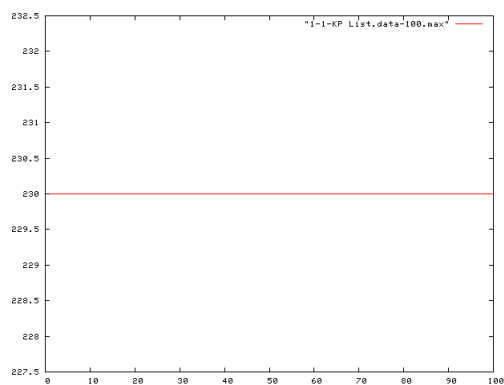


図 2: 選択だけを実装

3.3 交叉だけの実装

図3は急激に上昇したり、その後すぐに下降したりする。これはうまく交叉されて評価が高くなったが、選択が無いために値が増えること無く、次の交叉では値が低くなったりと繰り返すのだろう。

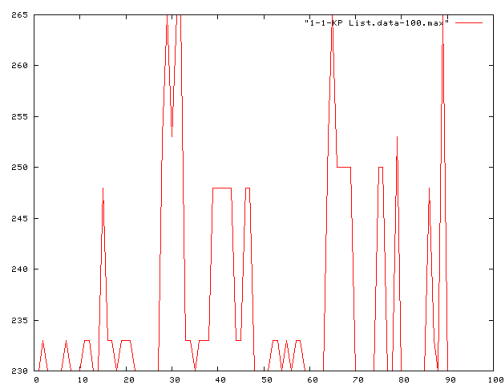


図 3: 交叉だけを実装

3.4 突然変異だけの実装

図4の突然変異の時は上下はする．これは交叉の時と同じ理由が挙げられる．一方，交叉より変化が激しくないのは，突然変異事態の頻度が低いためであろう．

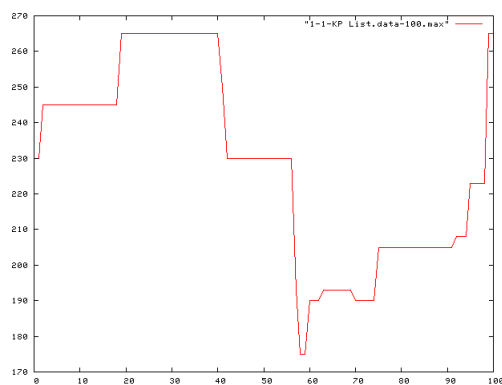


図 4: 突然変異だけを実装

3.5 交叉と突然変異の実装

図5は交叉と突然変異をやることによって同様に激しく振動するようになり，進化はうまく出来ていないことがわかる．

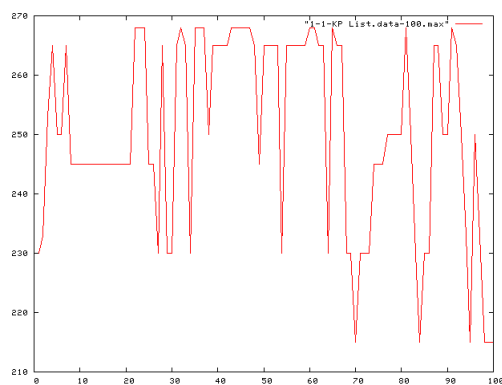


図 5: 交叉と突然変異を実装

3.6 選択と交叉の実装

図6を見ると、一見うまく進化しているように見える。これは選択によって評価が高いものが選ばれるようになった結果である。しかし、図1と比べるとわかるように、低い値で収束している。これは突然変異がないために局所解へと収束したのだろう。

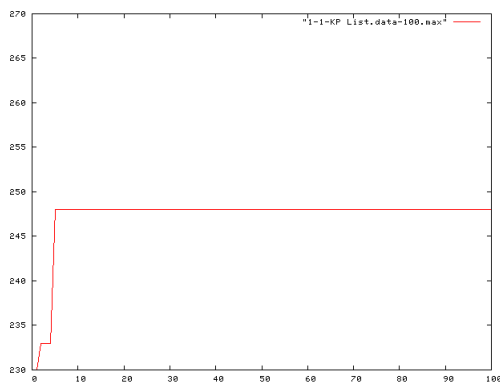


図 6: 選択と交叉を実装

3.7 選択と突然変異の実装

図7を見れば結果はうまく進化している。しかし、図1と比較すると収束までに時間がかかっている。これは交叉がないため、効率よく進化できないのが起因している。また、進化しているとは言っても、突然変異での変化であるのでこれは偶然だと考えるのが妥当だ。

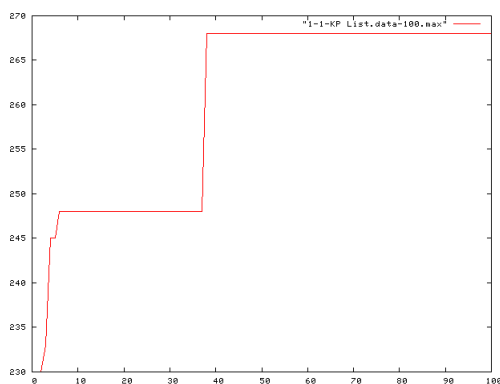


図 7: 選択と突然変異を実装

4 全体的な考察

全体的に見ればわかるように、突然変異、交叉、選択のオペレータはそれぞれが重要だということがわかる。特に遺伝子を淘汰を行なう選択を無くせば収束を行なうことがなくなる。また、選択だけでは変化もしない。つまり、選択に必要な評価計算が重要なことも示している。交叉は遺伝子を効率よく進化させるが、局所的になりやすく、そこから突然変異の必要性が見いだせる。