

## 進捗報告

### 1 今週やったこと

- TDGA と SGA を用いた FastAutoAugment 実験

### 2 TDGA と SGA を用いた FastAutoAugment 実験

先週の話にあがっていた諸パラメータを以下で設定して実験を回した.

- 遺伝子表現:ビットエンコーディング
- 個体数と世代数の調整:個体数 16(SGA では 32), 50 世代
- 突然変異の選択: ビット反転. 突然変異率は 0.02(SGA では 0.05). ナップサックの値をそのまま使っているだけなので調整の余地あり
- 初期温度の選択, 温度スケジューラの導入: 初期温度 0.1, 終了温度 0.02, 温度スケジューラ:指数スケジューラ
- loss の最小化か acc の最大化か: acc の最大化
- 最終的にどの拡張を採用するか: SGA : 各探索において最良個体を保存し, あとで結合. TDGA : 各探索において選択操作により何個体か選択し, あとで結合

その他, 適用強度については RandAugment の拡張範囲 (各操作に対して上限と下限を設定し,  $[0, 30]$  でスケール) を採用した操作集合に対し, 全操作の強度を統一した. また, 操作集合は [ AutoContrast, Equalize, Invert, Rotate, Posterize, Solarize, SolarizeAdd, Color, Contrast, Brightness, Sharpness, ShearX, ShearY, CutoutAbs, TranslateXabs, TranslateYabs ] である. 各操作を 0, 15, 30 の強度で適用した例を下に示す. ただし, AutoContrast, Equalize, Invert は強度によらない.



図 1: 元画像

## 2.1 SGA

SGA においては交叉に 1 点交叉, 選択にトーナメント戦略 (サイズ 2) を用いた. 適用強度を変化させながら実験した結果, 全ての探索においてどの拡張も採用しない個体が最も上位の適応度を持っていた. これに関しては元々学習している子モデルが拡張なしで学習しているためだと考えられる. 下に探索結果の一例を示す.

また, このとき Validation Accuracy は 91.240 % となった.

## 2.2 TDGA

TDGA においては交叉に一様交叉を用いた. TDGA では多様性を維持しつつ大域解を探索することができる特徴があるので, ベイズアプローチにおいて 8 回探索している (子モデルの学習 4 回, 各子モデルに対し 2 回のベイズ探索) とところを 1 回に減らして (子モデルの学習 1 回, 子モデルに対し 1 回の GA 探索) 探索において上位 8 個体を採用するという方針で実験をした. ちなみにこの時点で FastAutoAugment とは何の関連性もなくなった. また, 強度は 10 とし, 初期温度は 1(やや高い気もする), 終了温度は 0.02 と設定した. 温度スケジューラについては森先生の論文中の静的環境の対するスケジューリング関数と動的環境に対する Feedback TDGA(FTDGA) を読んだ結果, 今回の問題は静的環境で指数スケジューリングを採用しているという点で現実装のままで問題なさそうだったのでひとまずこのままにしている. FTDGA の静的環境における有効性も示されていたが, 実装コストの面で一旦保留.

結果, この条件において FastAutoAugment と同程度の実行時間で終了した (3 時間ちょっと). 下に探索結果の一例を示す.

探索が進んでいること, および温度のスケジューリングにより個体群がある遺伝子座において収束をしていることがわかる. また, このとき Validation Accuracy は 91.370 % となった. また, 元のレポジトリの FastAutoAugment では 91.810 % の精度が出ていた. この精度がどれくらい伸びるかは適用強度, 突然変異率, 温度パラメータ等を変化させながら観察したい. また, 強度を 15, エポックを倍にすると 91.740 % まで精度が上昇することを確認した.

また, このとき選択された操作は下のようになった.

Listing 1: subpolicies\_TDGA

```
1 Search Phase 0
2 (Compose(
3     Pad(padding=4, fill=0, padding_mode=constant)
4     RandomCrop(size=(32, 32), padding=None)
5     RandomHorizontalFlip(p=0.5)
6     ToTensor()
7 ), array([0.8486], dtype=float32))
8 (Compose(
9     Pad(padding=4, fill=0, padding_mode=constant)
10    RandomCrop(size=(32, 32), padding=None)
11    RandomHorizontalFlip(p=0.5)
12    AutoContrast
13    Color(magnitude=0.70)
14    Contrast(magnitude=0.70)
15    Sharpness(magnitude=0.70)
16    ToTensor()
17 ), array([0.825], dtype=float32))
18 (Compose(
19     Pad(padding=4, fill=0, padding_mode=constant)
20     RandomCrop(size=(32, 32), padding=None)
21     RandomHorizontalFlip(p=0.5)
22     AutoContrast
23     Brightness(magnitude=0.70)
24     ToTensor()
25 ), array([0.8402], dtype=float32))
26 (Compose(
27     Pad(padding=4, fill=0, padding_mode=constant)
28     RandomCrop(size=(32, 32), padding=None)
29     RandomHorizontalFlip(p=0.5)
30     Color(magnitude=0.70)
31     ToTensor()
32 ), array([0.8454], dtype=float32))
33 (Compose(
```

```

34     Pad(padding=4, fill=0, padding_mode=constant)
35     RandomCrop(size=(32, 32), padding=None)
36     RandomHorizontalFlip(p=0.5)
37     AutoContrast
38     Brightness(magnitude=0.70)
39     Sharpness(magnitude=0.70)
40     ShearX(magnitude=0.10)
41     ToTensor()
42 ), array([0.79819995], dtype=float32))
43 (Compose(
44     Pad(padding=4, fill=0, padding_mode=constant)
45     RandomCrop(size=(32, 32), padding=None)
46     RandomHorizontalFlip(p=0.5)
47     AutoContrast
48     Color(magnitude=0.70)
49     Contrast(magnitude=0.70)
50     ToTensor()
51 ), array([0.83779997], dtype=float32))
52 (Compose(
53     Pad(padding=4, fill=0, padding_mode=constant)
54     RandomCrop(size=(32, 32), padding=None)
55     RandomHorizontalFlip(p=0.5)
56     Sharpness(magnitude=0.70)
57     ToTensor()
58 ), array([0.843], dtype=float32))
59 (Compose(
60     Pad(padding=4, fill=0, padding_mode=constant)
61     RandomCrop(size=(32, 32), padding=None)
62     RandomHorizontalFlip(p=0.5)
63     ShearY(magnitude=0.10)
64     ToTensor()
65 ), array([0.797], dtype=float32))

```

拡張なしが最も高い適応度を示すのは SGA と同じだが、しっかりと多様性を維持しつつ最終的なポリシーが得られていることがわかる。しかしながら、選ばれている操作はかなり限定的であり、おおよそ元の画像に位置的な変化を加えないような操作が選ばれることがわかった。これをどう改善していくかは今後の課題になりそう。ただ、元の FastAutoAugment ではしっかり変形等の拡張が採用されていた。これはおそらくベイズ探索の強度を下げてあまりロスの下がりすぎない操作を大域的に探しているからだと考えられる。実際に探索結果の Validation Accuracy は 0.4 - 0.6 とそこまで良くないものが採用されていることがわかった (ただし、ベイズでやってるのはロスの最小化なので Accuracy が目的関数ではない)。TDGA においても遺伝子空間の多様性に加えてこのあたりの表現空間の多様性を維持する方法を取り入れたいという感じがした。

### 3 来週のタスク

TDGA の適用強度と初期温度、終了温度を変化させたときの影響を観察する。

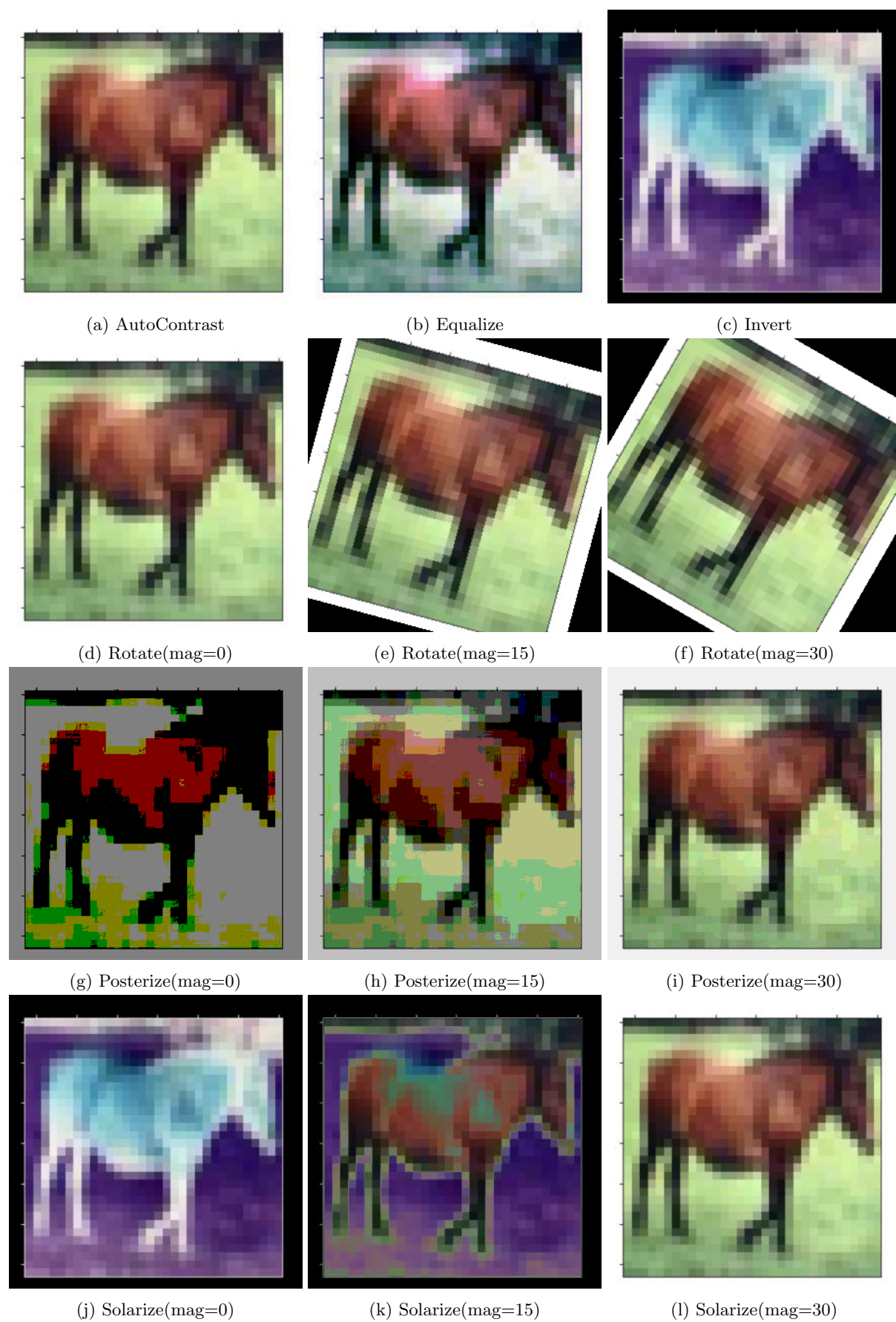


図 2: 強度を変えた各操作の適用例 (1)

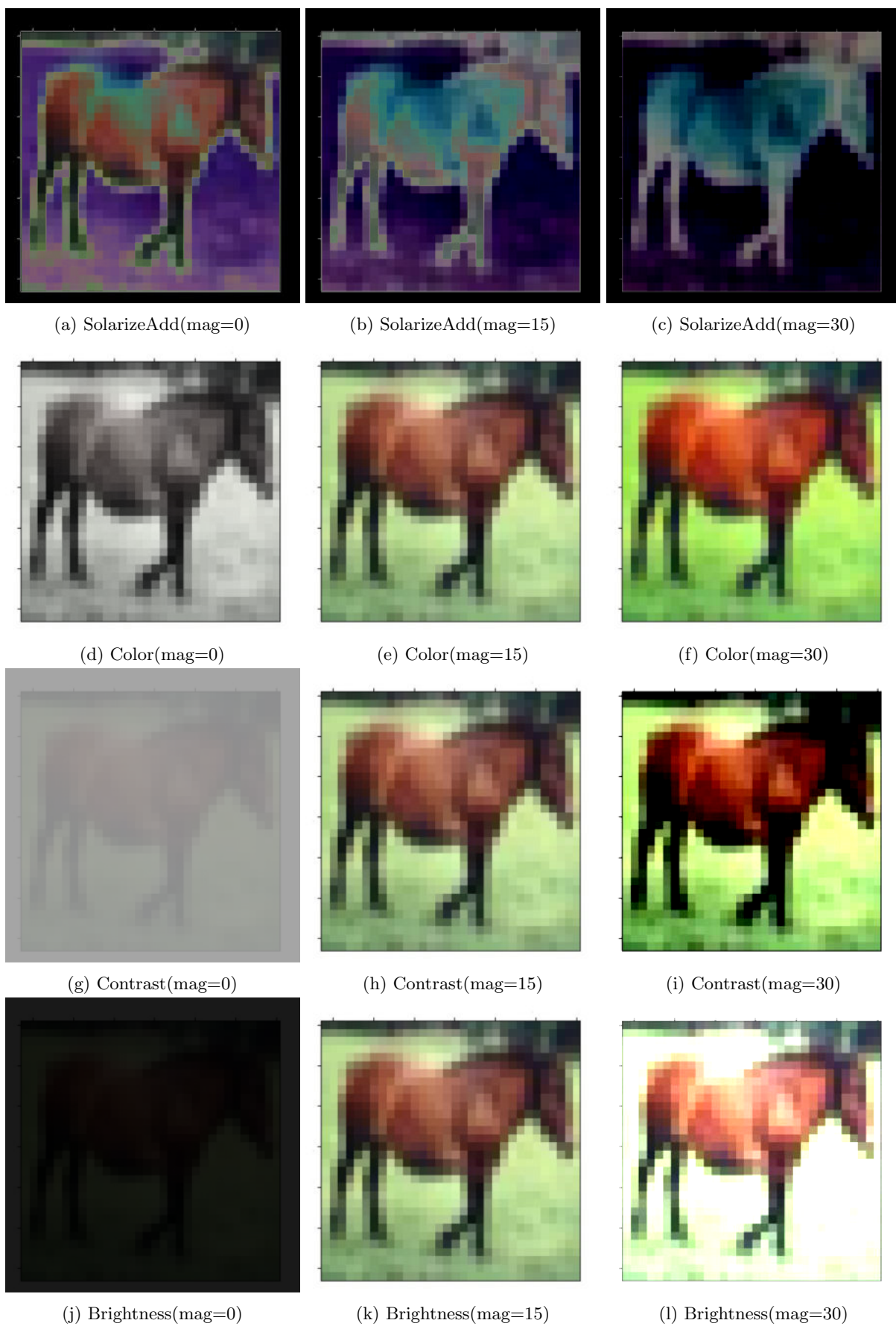


図 3: 強度を変えた各操作の適用例 (2)





(a) Sharpness(mag=0)



(b) Sharpness(mag=15)



(c) Sharpness(mag=30)



(d) ShearX(mag=0)



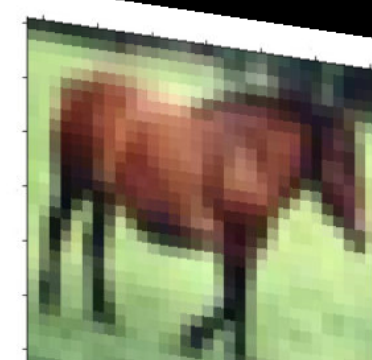
(e) ShearX(mag=15)



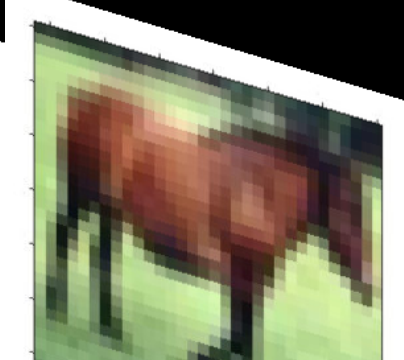
(f) ShearX(mag=30)



(g) ShearY(mag=0)



(h) ShearY(mag=15)



(i) ShearY(mag=30)

図 4: 強度を変えた各操作の適用例 (3)



(a) CutoutAbs(mag=0)



(b) CutoutAbs(mag=15)



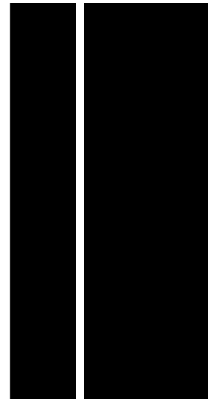
(c) CutoutAbs(mag=30)



(d) TranslateXabs(mag=0)



(e) TranslateXabs(mag=15)



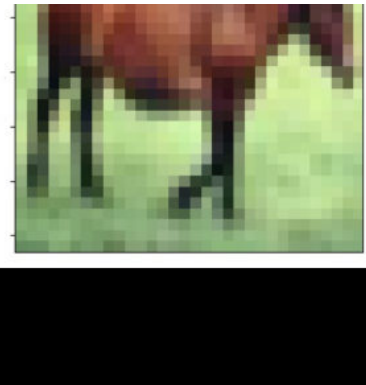
(f) TranslateXabs(mag=30)



(g) TranslateYabs(mag=0)



(h) TranslateYabs(mag=15)



(i) TranslateYabs(mag=30)

図 5: 強度を変えた各操作の適用例 (4)

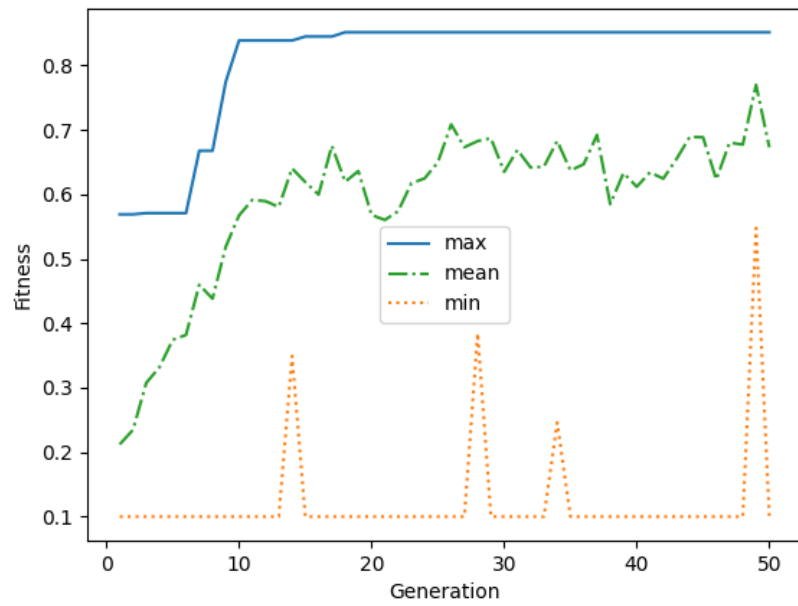


図 6: SGA における探索の過程

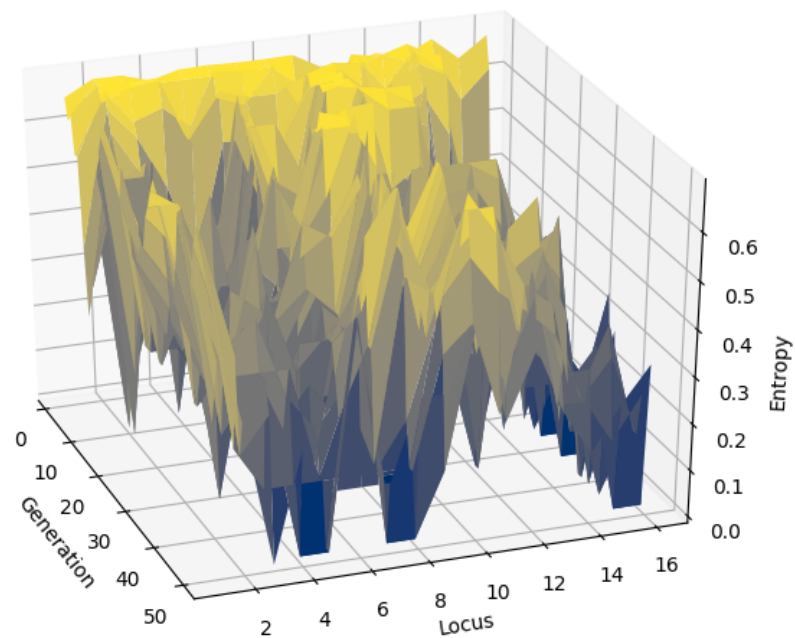


図 7: SGA における遺伝子座毎のエントロピー



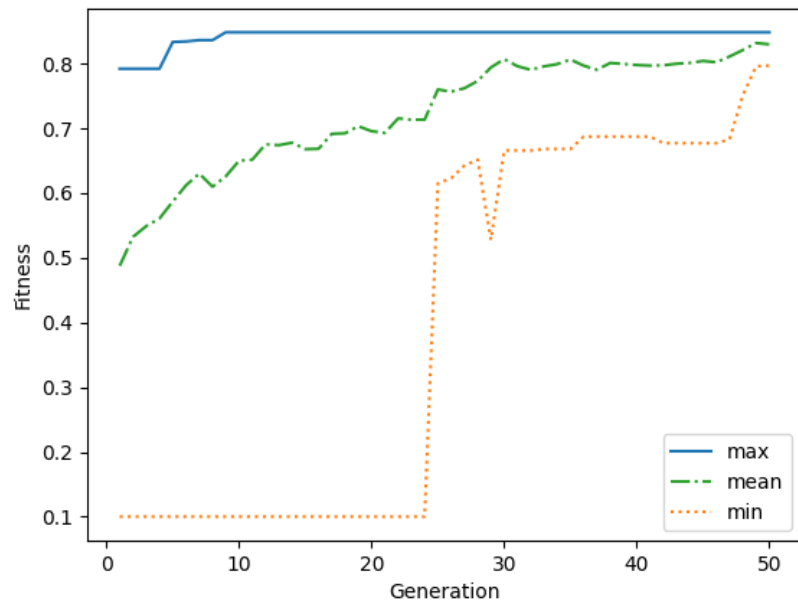


図 8: TDGA における探索の過程

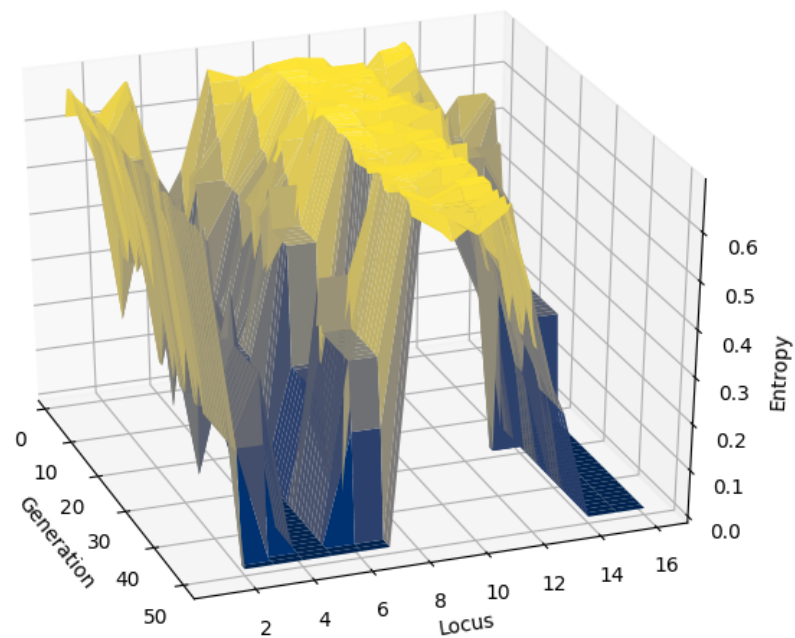


図 9: TDGA における遺伝子座毎のエントロピー