

# 授業時間割の生成アルゴリズムの開発

後藤 拓巳<sup>†</sup> 平山 基<sup>‡</sup>

<sup>‡</sup> 阿南工業高等専門学校 〒774-0017 徳島県阿南市見能林町青木 265

E-mail: <sup>†</sup> 1124145@anan-nct.ac.jp, <sup>‡</sup> hmotoi@anan-nct.ac.jp

あらまし 学校の授業時間割生成アルゴリズムを、SA 法 (Simulated Annealing) を用いた最適化手法を開発した。4 クラスの時間割を用いて最適化を行い、その結果を踏まえて 5 学年 22 クラスの最適化を行い、手動による最適化に比べて大幅に時間と労力を省き、効率よく解を求めることができると分かった。

キーワード Simulated Annealing 法, 最適化問題

## Development of Generation Algorithm of Class Timetable

Takumi GOTO<sup>†</sup> and Motoi HIRAYAMA<sup>‡</sup>

<sup>‡</sup> National Institute of Technology, Anan College 265 Minobayashi-cho, Anan-shi, Tokushima, 774-0017 Japan

E-mail: <sup>†</sup> 1124145@anan-nct.ac.jp, <sup>‡</sup> hmotoi@anan-nct.ac.jp

**Abstract** We developed a generation algorithm of class timetables using simulated annealing method. First, we succeeded to optimize timetables of 4 classes. On the basis of those results, we easily optimized timetables for 22 classes. the algorithm we developed lead to a labor-saving and efficiency technology of schedule management without burdensome manual operations.

**Keywords** Simulated Annealing method, Optimization problem

### 1. はじめに

時間割作成は主に手作業で行われているものが多く、教員や学生数、講義数が増加すると複雑な条件が課され、全ての条件を満たす解を導くことが困難であり、多大な時間と労力がかかるため改善が必要となっている。優れた時間割作成の手法が発見されることに大きな期待が集まっており、様々な手法が開発されてきた。例えば、時間割作成アルゴリズムに遺伝的アルゴリズム[1,2]や SAT ソルバ[3-5]を用いた研究例が報告されている。比較的 condition 設定の少ない時間割であれば解を求めることができるが、高専の時間割のような過密で複雑な時間割が対象となると、解が求まるまでに時間を要したり、最適解が求まらなかったりするため実用化が厳しい。そのため、本研究では焼きなましという物理現象を起源にもつ最適化アルゴリズムである SA 法 (Simulated annealing method) を用いて授業時間割の最適化を行う。

### 2. 最適化手法

#### 2.1. Simulated Annealing 法

焼きなまし (Annealing) とは鉄などの金属を結晶化させる際に高温で加熱した金属材料を徐々に温度を下げ、冷却することで、元の金属より欠陥の少ない硬い結晶構造を作る手法がある。加熱と冷却を繰り返し行うことで最適な結晶構造を得ることができる。この手法を多次元最適化問題に適用することで、局所安定点多い系においても大域的安定点を求めることができる。SA 法では熱エネルギー (温度) に相当する評価値を定義し、状態を遷移させていく。

系の遷移とともに評価関数により系の適合度を評価し、遷移前後の比較をすることで遷移の取捨選択を行う。ここで、評価値の減少方向のみの遷移を採用した場合、初期状態近傍の局所安定点が解として得られるが評価値の変化量に対応した一定の割合で評価値の増加方向の遷移を採用することで局所安定点から大域的安定点へと解を導く。

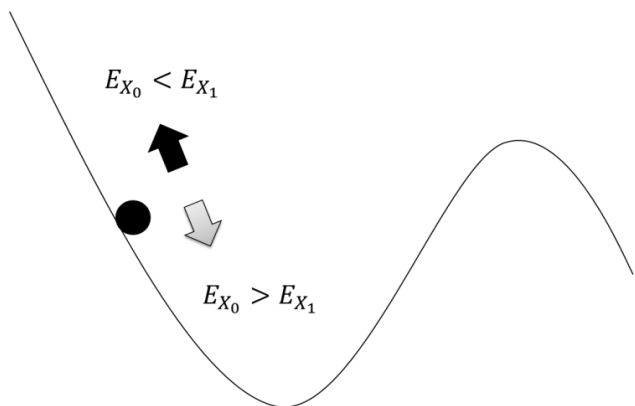


図 1 安定点と遷移確率

遷移確率  $a_t$  は、評価値  $E$  の関数として、

$$a_t = e^{-\frac{E_0}{E_1}}$$

と設定する。遷移確率  $a_t$  は 0 から 1 までしか必要ではないが、評価値が  $E_{X_0} > E_{X_1}$  の場合は遷移確率  $a_t$  が 1 であるため必ず遷移する。  $E_{X_0} < E_{X_1}$  の場合は  $a_t$  が 1 より小さくなるため、遷移確率  $a_t$  に従い遷移させる。

| Class 1 | Mon | Tue | Wed | Thu | Fri |
|---------|-----|-----|-----|-----|-----|
| 1       | 1   | 4   | 7   | 10  | 12  |
| 2       | 2   | 5   | 8   | 2   | 9   |
| 3       | 3   | 6   | 3   | 11  | 13  |
| 4       | 0   | 0   | 9   | 11  | 0   |
| Class 2 | Mon | Tue | Wed | Thu | Fri |
| 1       | 6   | 5   | 8   | 9   | 13  |
| 2       | 1   | 4   | 7   | 10  | 12  |
| 3       | 2   | 3   | 2   | 11  | 3   |
| 4       | 0   | 0   | 14  | 11  | 0   |
| Class 3 | Mon | Tue | Wed | Thu | Fri |
| 1       | 15  | 1   | 13  | 6   | 10  |
| 2       | 4   | 12  | 5   | 8   | 15  |
| 3       | 3   | 9   | 3   | 11  | 7   |
| 4       | 0   | 0   | 7   | 11  | 0   |
| Class 4 | Mon | Tue | Wed | Thu | Fri |
| 1       | 4   | 12  | 5   | 8   | 7   |
| 2       | 15  | 1   | 13  | 15  | 10  |
| 3       | 6   | 3   | 9   | 11  | 3   |
| 4       | 0   | 0   | 3   | 11  | 0   |

図 2 既存の時間割 (2016 年度前期)。表中の数字は講義の担当教員の ID を示す。

系の遷移は同一クラスのランダムな講義を 2 つ選択し入れ替えることで講義の種類やコマ数を保存したまま遷移を行うことができる。例えば、図 3 のようにクラス 1 の月曜日の 2 コマ目と火曜日の 1 コマ目の講義を入れ替える。クラス 1 のみに注目してみると矛盾なく入れ替わっているように見えるが、4 クラスの時間割を同時に確認するとクラス 1 とクラス 4 の月曜日の 2 コマ目がともに「4」の教員で重複しているため、解の条件を満たさなくなっていることがわかる。このと

き、系の評価値 (エネルギー) が上昇するように、評価関数を設定する必要がある。すべての条件を満たさなければ評価値が最低とならず、繰り返し入れ替えを行う。

| 入れ替え前   |     |     |     |     |     | 入れ替え後   |     |     |     |     |     |
|---------|-----|-----|-----|-----|-----|---------|-----|-----|-----|-----|-----|
| Class 1 | Mon | Tue | Wed | Thu | Fri | Class 1 | Mon | Tue | Wed | Thu | Fri |
| 1       | 1   | 4   | 7   | 10  | 12  | 1       | 1   | 2   | 7   | 10  | 12  |
| 2       | 2   | 5   | 8   | 2   | 9   | 2       | 4   | 5   | 8   | 2   | 9   |
| 3       | 3   | 6   | 3   | 11  | 13  | 3       | 3   | 6   | 3   | 11  | 13  |
| 4       | 0   | 0   | 9   | 11  | 0   | 4       | 0   | 0   | 9   | 11  | 0   |
| Class 2 | Mon | Tue | Wed | Thu | Fri | Class 2 | Mon | Tue | Wed | Thu | Fri |
| 1       | 6   | 5   | 8   | 9   | 13  | 1       | 6   | 5   | 8   | 9   | 13  |
| 2       | 1   | 4   | 7   | 10  | 12  | 2       | 1   | 4   | 7   | 10  | 12  |
| 3       | 2   | 3   | 2   | 11  | 3   | 3       | 2   | 3   | 2   | 11  | 3   |
| 4       | 0   | 0   | 14  | 11  | 0   | 4       | 0   | 0   | 14  | 11  | 0   |
| Class 3 | Mon | Tue | Wed | Thu | Fri | Class 3 | Mon | Tue | Wed | Thu | Fri |
| 1       | 15  | 1   | 13  | 6   | 10  | 1       | 15  | 1   | 13  | 6   | 10  |
| 2       | 4   | 12  | 5   | 8   | 15  | 2       | 4   | 12  | 5   | 8   | 15  |
| 3       | 3   | 9   | 3   | 11  | 7   | 3       | 3   | 9   | 3   | 11  | 7   |
| 4       | 0   | 0   | 7   | 11  | 0   | 4       | 0   | 0   | 7   | 11  | 0   |
| Class 4 | Mon | Tue | Wed | Thu | Fri | Class 4 | Mon | Tue | Wed | Thu | Fri |
| 1       | 4   | 12  | 5   | 8   | 7   | 1       | 4   | 12  | 5   | 8   | 7   |
| 2       | 15  | 1   | 13  | 15  | 10  | 2       | 15  | 1   | 13  | 15  | 10  |
| 3       | 6   | 3   | 9   | 11  | 3   | 3       | 6   | 3   | 9   | 11  | 3   |
| 4       | 0   | 0   | 3   | 11  | 0   | 4       | 0   | 0   | 3   | 11  | 0   |

図 3 入れ替え操作

## 2.2. 評価関数

阿南高専で時間割に求められる条件は以下の 5 つである。解の条件として、

- ① 教員の重複
- ② 連続した講義
- ③ クラスの空き時間
- ④ 教員の代休日
- ⑤ 非常勤教員の希望

などが考えられる。②および③については、遷移の過程で条件を適用するが、①および④については、評価関数を用いて評価する。⑤については最適化の範囲から除外する。評価値は整数とし、教員の重複が生じる場合に +1 する。また各教員に空き日 (1 日講義のない日) がない場合も評価値を +1 する。また、評価値の最小値は遷移確率  $a_t$  が発散しないよう 1 とした。

## 3. 結果と考察

### 3.1. 4 クラスの最適化①

初期状態として既存の時間割から最適化を行った (図 2)。手動で最適化された時間割のため、①~⑤のすべての条件を満たしている。図 6 は、評価値の時間進展を示す。最適化の初期状態では、評価値は最適解である 1 である。状態遷移を繰り返し、80step 付近で局所安定点に落ち着いているが、150step 付近で評価値が急激に上昇し、200 から 350step 付近で大域的安定点に最適化されている。SA 法による状態遷

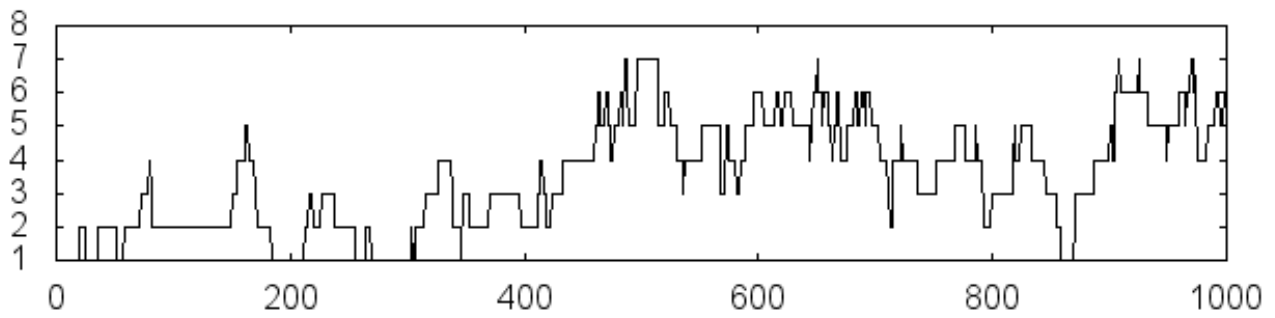


図 6 4クラスの最適化過程

移によって局所安定点から抜け出し、大域的安定点に落ち着いていることがわかる。さらに 860step で再び最適解が求まっている。図 7 は 860step の最適化された時間割を示す。全てのクラスで初期状態とは異なる時間割が得られており、教員の配置がすべての条件を満たしていることもわかる。

| Class 1 | Mon | Tue | Wed | Thu | Fri |
|---------|-----|-----|-----|-----|-----|
| 1       | 0   | 3   | 10  | 2   | 6   |
| 2       | 9   | 11  | 7   | 3   | 13  |
| 3       | 9   | 2   | 4   | 5   | 1   |
| 4       | 12  | 0   | 8   | 11  | 0   |
| Class 2 | Mon | Tue | Wed | Thu | Fri |
| 1       | 7   | 2   | 11  | 14  | 4   |
| 2       | 0   | 0   | 9   | 13  | 5   |
| 3       | 6   | 12  | 10  | 1   | 0   |
| 4       | 3   | 8   | 3   | 2   | 11  |
| Class 3 | Mon | Tue | Wed | Thu | Fri |
| 1       | 13  | 7   | 8   | 15  | 9   |
| 2       | 7   | 5   | 11  | 3   | 1   |
| 3       | 10  | 0   | 11  | 12  | 4   |
| 4       | 15  | 0   | 3   | 6   | 0   |
| Class 4 | Mon | Tue | Wed | Thu | Fri |
| 1       | 4   | 1   | 0   | 13  | 0   |
| 2       | 10  | 15  | 12  | 9   | 0   |
| 3       | 3   | 7   | 3   | 8   | 11  |
| 4       | 6   | 5   | 11  | 3   | 15  |

図 7 860step 目の状態

### 3.2. 4クラスの最適化②

最適化のテストケースとして、4クラス全ての初期状態が同一の状態から最適化を行った。図 8 は、評価値の時間発展を示す。初期状態では4クラスの時間割が重複しているため、1step の近傍で非常に大きい値であることがわかる。最適化を繰り返す度に徐々に解が落ち着き 400step 付近で大域的安定点に落ち着いている。さらに最適化を繰り返すと 1100step 付近で大域的安定点に再び落ち着いている。図 6 と比較すると、初期状態の評価値は高くなっているものの、評価値が 10 程度まで最適化が進行すると、クラス毎の時間割の重複が少なくなり、初期状態の影響がほぼ無視できると考えられる。これらのことから、初期状態の影響は 200step 程度で消失していることもわかる。

図 9 は 1100step での時間割を示す。同一の時間割から最適化を開始した場合においても、すべての条件を満たしており、図 7 の結果とも異なっている。

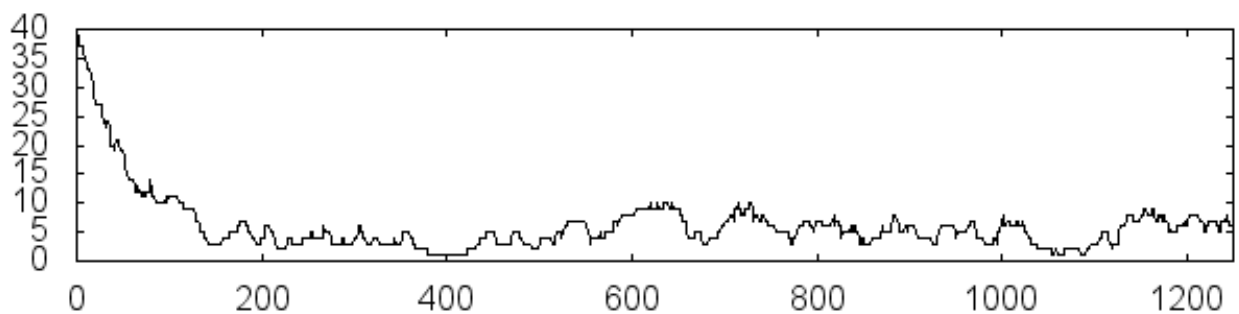


図 8 同一の時間割の最適化過程

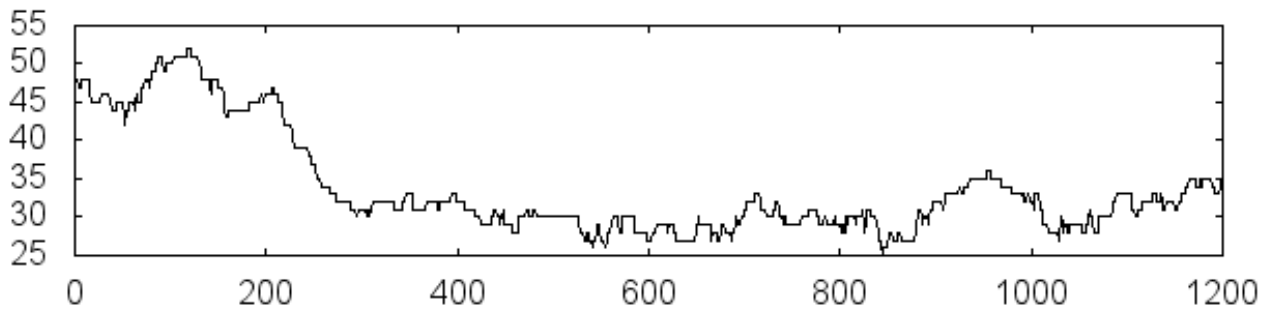


図 10 22クラスの最適化過程

| Class 1 | Mon | Tue | Wed | Thu | Fri |
|---------|-----|-----|-----|-----|-----|
| 1       | 9   | 5   | 11  | 4   | 0   |
| 2       | 2   | 6   | 3   | 9   | 2   |
| 3       | 7   | 12  | 0   | 3   | 0   |
| 4       | 11  | 13  | 10  | 8   | 1   |
| Class 2 | Mon | Tue | Wed | Thu | Fri |
| 1       | 3   | 0   | 0   | 2   | 2   |
| 2       | 6   | 9   | 11  | 12  | 11  |
| 3       | 1   | 0   | 3   | 9   | 13  |
| 4       | 4   | 5   | 7   | 10  | 8   |
| Class 3 | Mon | Tue | Wed | Thu | Fri |
| 1       | 2   | 0   | 11  | 3   | 6   |
| 2       | 11  | 13  | 5   | 8   | 4   |
| 3       | 2   | 10  | 0   | 9   | 3   |
| 4       | 9   | 7   | 12  | 0   | 1   |
| Class 4 | Mon | Tue | Wed | Thu | Fri |
| 1       | 11  | 0   | 4   | 11  | 13  |
| 2       | 7   | 9   | 2   | 3   | 2   |
| 3       | 12  | 3   | 5   | 6   | 0   |
| 4       | 10  | 9   | 8   | 0   | 1   |

図 9 1100step 目の状態

## 5. 参考文献

- [1] GA と SA 法を組み合わせた時間割編成問題の解法, 志村直子, 斎藤博昭, 中西正和, 情報処理学会第 59 回全国大会.
- [2] 遺伝的アルゴリズムと焼きなまし法による学校時間割作成について, 福島誠, 島根大学教育学部紀要 (自然科学) 第 36 巻 pp.71-77.
- [3] SAT を用いた定期試験時間割作成の設計, 岡田裕生, 岩佐健司, 児島雄志, 平成 27 年度電気関係学会四国支部連合大会講演要旨, 17-27 (2015)
- [4] SAT を用いた授業時間割作成の設計, 中野政利, 岩佐健司, 児島雄志, 平成 27 年度電気関係学会四国支部連合大会講演要旨, 17-28 (2015)
- [5] Minisat を用いた定期試験時間割作成の設計, 坪内智哉, 岩佐健司, 児島雄志, 平成 28 年度電気関係学会四国支部連合大会講演要旨, 17-4 (2016)

### 3.3. 22クラスの最適化

阿南高専で実施されているクラス数 (22クラス) および講義数での時間割の最適化を行った. 図 10 は, 評価値の時間進展を示す. 初期状態は既存の時間割を用いているため, 最適化初期の評価値は 1 となるはずであるが, 学年共通で開講する帯科目が複数あるため, 評価値が高くなっている. 帯科目の取り扱いについては, 状態遷移において全ての開講クラスを動かす必要があり, 今後実装予定である.

## 4. まとめと今後の課題

本研究では, SA 法を用いて学校の授業時間割の最適化を行った. 確率的に評価値の高い状態を選択することによって手動より短時間で多くの最適解を抽出することができるという結果が得られた. このようなことから今後は, より細かい条件設定を追加し, GUI などを開発することにより, ユーザビリティの向上に取り組みたい.