

水産大学の授業時間割の自動編成に関する基礎研究

メタデータ	言語: Japanese 出版者: 水産研究・教育機構 公開日: 2024-07-25 キーワード (Ja): キーワード (En): timetabling problem; simulated annealing; automatically construction; optimization problem 作成者: 徳永, 憲洋 メールアドレス: 所属:
URL	https://doi.org/10.57348/0002010424

水産大学の授業時間割の 自動編成に関する基礎研究

徳永憲洋^{1†}

Basic research for automatic construction of course timetables in National Fisheries University

Kazuhiro Tokunaga^{1†}

Abstract: Since the construction of class timetables in the university requires a great deal of effort and time, many studies have been conducted in Japan and abroad on the automatic construction of class timetables. Similarly, at the National Fisheries University, labor and time are devoted to the creation of class timetables. Therefore, author aims to conduct basic research on automatic timetable construction with the simulated annealing method in this study. This paper describes the proposed method and algorithm for generating timetables using the simulated annealing method. The generated timetables by the proposed method are also reported.

Key words: timetabling problem, simulated annealing, automatically construction, optimization problem

序 論

高校や大学などの授業時間割の編成は、クラス、教員、施設（教室など）などについて各々の条件やスケジュールを全て考慮し、なおかつ矛盾なく組み立てる必要がある。また、授業間の教室の移動距離や必修科目と選択科目の配置バランスなどの配慮や学校特有のルールなども考慮して編成をする必要も出てくる。このように授業の時間割編成では上記のような制約を全て考慮し矛盾なく時間割を組み立てる必要があり、その編成においては多くの時間と労力が費やされている。このため、編成の労力と時間を短縮するために国内外において授業時間割の自動編成についての研究が多くなされている。一般的に授業時間割の編成に関わる課題は、時間割編成問題（Timetabling problem）と呼ばれており、組み合わせ最適化問題として捉えて研究がなされている¹⁾。時間割編成の自動化方法のひとつとしては、時間割編成問題を目的関数と制約条件に定式化し整数計画問題として解を求める方法があり、多数の研究がなされて

いる^{2,3,4,5)}。他方、シミュレーテッドアニーリングや遺伝的アルゴリズムなどの最適解探索法を用いた時間割編成も提案されている^{6,7)}。シミュレーテッドアニーリング（SA: Simulated Annealing）法や遺伝的アルゴリズムは、どちらも代表的な確率的最適化法として知られており、どちらの方法も局所解に陥りにくく大域的最適解を求めやすい特徴を持つ。

水産大学校における授業時間割編成においても教職員の時間と労力が割かれている。水産大学校の教員は、教員や教室の数などのリソースに相当の制約がある中で授業を担当・実施せざるを得ない状況にある。例えば、教員は他学科あるいは所属学科も含めた複数学科で開講される授業を担当する場合がある。さらにこの場合は多数の受講人数を収容できる教室が必要だが、そのような大規模な教室は少ない。さらに足りない講師については非常勤講師を雇うことで対応するが、非常勤講師は曜日や時間などの制約がある。このように、水産大学校の授業編成では、制約条件の縛りが厳く編成の自由度が低いため、すべての条件をクリ

2023年10月31日受付、2024年1月26日受理

1 水産大学校海洋機械工学科（Department of Marine Mechanical Engineering, National Fisheries University）

† 責任著者（Corresponding author）: tokunaga@fish-u.ac.jp

考える。以上を踏まえると、時間割の編成とは 25×20 のマス目にそれぞれ授業を配置していくことに等しい。ここで、実際の水産大学校の時間割は、1 時限を 45 分で分けているが、ほとんどの授業が上記のように 90 分を 1 コマとする単位で授業配置が行われているため、本研究についても 90 分を 1 コマとして、時間割編成を考慮する。

時間割は前学期と後学期の 2 学期それぞれで別々に作

[illegible]

Figure 2 An example of a course class of timetable

成する。前、後学期ともに全クラスで約 100 の授業が受講される。また教員の数法非常勤講師の数も含めると約 100 名である。ここで、水産大学校では、コース科目と呼ばれる授業がある。これは 1, 2 年生で開講される共通科目をまとめた科目である。共通科目とは英語や体育など学科毎の専門科目ではない一般科目に相当する授業であり、特に 1, 2 年生で多く開講される。学科毎それぞれで一般科目を開講するとなると多くの授業担当教員が必要となるが、水産大学校においては一般科目を担当できる教員の人数が限られていることから、1 学年の学生を 4 つのコース（グループ）に分け、授業単位ではなくコース単位で時間割に組み入れている。例として、令和 4 年度の前学期の 1 年生の授業時間割の一部を図 2 に示し、図を使いながら以下説明をする。図 2 の赤太枠で囲った表がコース科目である。表の行方向に A ～ D の 4 コース、列方向に 6 個のコース科目 (I) ～ (VI) がある。各コースには共通科目である外国語や体育などの授業が配置されている。そして、時間割にはそれぞれのコース科目を 1 コマとして当てはめている（図 2 青太枠で囲ったコマ）。このコース科目の時間割への配置が特殊であるため、本研究における授業時間割作成は単純に 25×20 のマス目に授業を割り当てることができず、自動化を困難とさせている要因の一つである。また教室については、授業ごとに受講する人数が異なるうえに、講義棟の教室の数やキャパシティが限られていることも、授業配置における制約のひとつとなる。例えば、水産大学校の 1 学年当たりの学生数は約 200 名である。よってコース科目内の一部の科目や全学科で開講される科目は約 200 名の学生が入る教室が必要となるが、そのような教室は 1 室しか存在しない。また新型コロナウイルス感染症などの感染症への対応では室内が密とならないような配慮も必要となり、さらに教室のキャパシティによる制限は厳しくなる。また非常勤講師においては出勤できる曜日や時間帯が限られることから、教員毎の授業可能日時の制約も考慮して時間割を編成する必要がある。

時間割編成の評価関数の定式化について

先に述べたように、本研究で取り扱う時間割編成は 25×20 のマス目に授業を割り当てる作業である。しかし各マス目には、授業科目、教員、教室の情報も入ることになる。これを数学的に考えると、時間割とは、クラス、コマ(時間)、授業科目、教員、教室の 5 つのモードからなる

テンソルとして考えることができる。具体的にはクラス、コマ（時間）、授業科目、教員、教室のそれぞれの概数が、20, 25, 120, 100, 15 からなる巨大なテンソルである。ここで、クラスは各学年の学科を表しており図1の行に相当する。すなわち本テンソルは約 $20 \times 25 \times 120 \times 100 \times 15$ の要素があり、時間割が配置された要素に1が入り、他はすべて0となる、0と1からなるスパースなテンソル（5次元の配列）だと考えることができる（図3参照）。

すなわち、時間割編成とは、ある制約条件のもと、エネルギー（ペナルティ）が最小となるように0と1の組み合わせを考慮する0-1組み合わせ最適化問題であり、その最適化の計算と更新プロセスは全てテンソルをもとに行うことが可能である。このような最適化問題では以下で示すラグランジュ関数を最小化する問題として考えることが多い。

$$L(X, \lambda) = f(X) + \lambda g(X). \quad (1)$$

ここで、時間割編成問題で考えると、 X は時間割を表すテンソルであり、 $f(X)$ は目的関数、 $g(X)$ は制約関数である。また λ はラグランジュ乗数と呼ばれる。制約関数

の元、目的関数が最小化、あるいは最大化するような X を求める、ということになる。最適な X を求める方法は、今日におけるまで様々に提案されている。 X が連続的な値を持ち、なめらかな関数（微分可能）の場合は、最急降下法や山登り法などの勾配法に準じた方法を用いればよい。時間割編成の場合は、 X が0-1からなる変数であるため、SA法や遺伝的アルゴリズムなどの最適化法が適していると考えられる。本研究では、最も考え方が単純であり、プログラミング開発も簡単と思われるSA法を時間割編成の方法として導入した。次に、時間割編成問題における式1の目的関数と制約変数について説明をする。時間割編成問題を単純に考えると、ペナルティが最小となるような X を導出できれば良い。ここでペナルティとは、例えば、同時帯のコマで一人の教員が複数のコマの授業を担当している、同一週内で同じ授業が複数回開講されている、など、時間割を編成するうえであってはならないような状況（罰則）を数値として定義したものである。すなわち時間割編成における様々な制約条件を関数化したものが制約関数であり、その制約関数は制約に従っていなければペナルティの大きさを出力する関数だと考えれば良い。

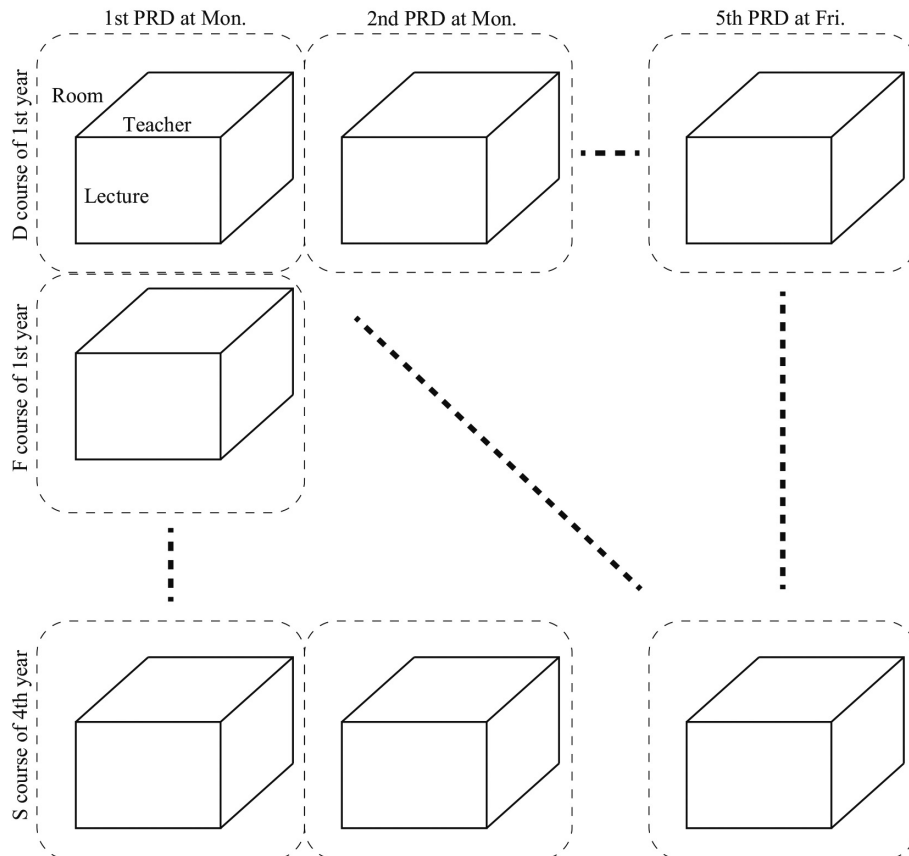


Figure 3 Diagram of timetable tensor

よって単純に式 1 における制約関数をペナルティの総数とするだけでも良い。ペナルティの数値の求め方は定式化することが好ましいが、難しい場合は開発者が独自で考えた方法でもかまわない。本研究ではこの考えに従い、いくつかのペナルティ項を定式化し、ペナルティの総数をSA法で最小化する方法を使用する。具体的には、以下の式を評価関数とし、評価関数が最小化するような X を求めることとする。

$$L(X) = \sum_{i=1}^I w_i P_i(X). \quad (2)$$

ここで、 I はペナルティ項の総数を表し、 $P_i(X)$ は i 番目のペナルティ項から算出されたペナルティとする。また w_i は i 番目のペナルティ項に対する重みを表す。重みの値が大きければそのペナルティを重要視するようになる。本研究では次の4つのペナルティ項を定義した。

- (1) どの授業も必ずどこかのコマに配置される。
- (2) 同じ教員が同じ時間帯において複数のクラスで授業を受け持たない。
- (3) 同じ部屋で複数の授業は開講できない。
- (4) 開講制約のかかっているコマには授業は配置できない（あるいはできる限り配置しない）。

(4) の制約では、授業によっては開講できない曜日、時間帯があったり、教員によっては授業が困難となる曜日、時間帯があるので、それらをまとめて、本稿では開講制約と呼ぶことにする。

ほかにも、週内に同じ授業が複数回開講できない、同じコマに複数の授業は配置できない、といった強い制約を一般的には定義する必要があるが、本研究では、これらの制約を後述するコマ配置のアルゴリズムで回避するようにしているため、ペナルティとして定義する必要がない。

また本研究では基礎的な研究が目的であるため、授業時間割を生成する上で必要不可欠である制約条件だけを取り扱う。しかし実際の運用を考えると、教室の移動距離や教員が連続して授業を行わないなどの人間的配慮もペナルティ項として含めたほうがより人間が編成する時間割に近くなると期待できる。

以下にそれぞれのペナルティ項の定義を示す。なお本研究において、以下のペナルティ項については定義した式のままプログラミングで実装をした。しかし X は多くの要

素が 0 であるスパースなテンソルであるため、定義した式のまま実装をすると計算時間が無駄である。このため、計算時間を向上させるのであれば、計算アルゴリズムを工夫したほうが良い。

(1) どの授業もどこかのコマに必ず配置される。

このペナルティ項は、用意されている全ての授業科目は週内のコマのどこかに配置されていなければならない、どこにも配置されていない授業が存在する場合は、その授業数をペナルティと定義する。今、 X の要素を $x_{l,p,r,c,t}$ とする。 l, p, r, c, t はそれぞれ、授業科目、教員、教室、クラス（学科やコース）、コマ（時間帯）のインデックスを表す。本ペナルティ $P_1(X)$ は以下のように定義できる。

$$P_1(X) = \sum_l \left(1 - \text{sign} \left(\sum_p \sum_r \sum_c \sum_t x_{l,p,r,c,t} \right) \right) \quad (3)$$

ここで、 $\text{sign}(\cdot)$ は符号関数（signum 関数）である。符号関数を用いる理由は、授業科目によっては複数の教員が担当したり、複数の教室を利用する場合があるためである。

(2) 同じ教員が同じ時間帯において複数のクラスで授業を受け持たない。

同じ教員が同じ時間帯で複数のクラスで授業を受け持っている数をペナルティとして定義する。本ペナルティ $P_2(X)$ は以下のように定義できる。

$$P_2(X) = \sum_p \sum_t (y_{p,t}) \quad (4)$$

$$y_{p,t} = \text{Re} \left(\sum_c \sum_p \sum_t x_{l,p,r,c,t} - 1 \right) \quad (5)$$

$\text{Re}(\cdot)$ は正規化線形関数（ReLU 関数）を表す。この式から導出される値は、同一時間帯のクラスにおいて複数の授業の担当が割り当てられた教員の数に等しい。

ここで、コース科目や複数クラスで開講される授業科目については、別途対策が必要となる。これらの授業科目は同時時間帯に複数クラスで開講されるため、テンソル X には同時時間帯において複数のクラスで同一教員の要素が1と表現することとなる。このまま式 4 で計算をすると、コース科目はペナルティとしてカウントされてしまうので別途対応が必要となる。対策の一例としては、このような複数のクラスで開講される授業科目については、 X の要素とし

て 1 を入れるのではなく、 $1/N_{cl}$ (N_{cl} は 1 番目の授業において開講されるクラス数) とすることで、上式を利用した対応が可能となる。

(3) 同じ部屋で複数の授業は開講できない。

同じ部屋で複数の授業が開講されている部屋数をペナルティとして定義する。本ペナルティ $P_3(X)$ は以下のように定義できる。

$$P_3(X) = \sum_r \sum_t (y_{r,t}) \quad (6)$$

$$y_{r,t} = \text{Re} \left(\sum_c \sum_r \sum_t x_{l,p,r,c,t} - 1 \right) \quad (7)$$

本ペナルティについても、複数のクラスで開講される授業科目についてはペナルティ項 (2) と同様の対策が必要である。

(4) 開講制約のかかっているコマには授業は配置できない (あるいはできる限り配置しない)。

授業、あるいは教員の都合により、開講が困難となるコマも生じる。例えば、教員によっては、会議が火曜日の午前中に毎週入るため火曜日の午前中は授業を開講したくない、などの開講に関わる制約が生じる。このため、開講制約を考慮した授業配置を考えなければならない。そこで開講制約のあるコマに授業が配置された場合はペナルティを算出するように定義をする。

まず、 S を開講制約テンソルとして定義する。 $S = \{s_{l,c,t}\}$ は授業科目 l 、コース c 、コマ t からなる $l \times c \times t$ テンソルであり、 l 番目の授業においてクラス c および t 番目のコマに制約がかかる場合は 1、かからない場合は 0 が入る。また、ゆるい制約をかけたい場合は (できるだけ開講したくない、などの場合)、0~1 の間の数値を入れてもよい。

開講制約テンソルを利用し、ペナルティ項 (4) のペナルティ $P_4(X)$ 計算は以下のように定義する。

$$P_4(X) = \sum_l \sum_t s_{l,c,t} \left(\text{sign} \left(\sum_l \sum_c \sum_t x_{l,p,r,c,t} \right) \right) \quad (8)$$

シミュレーテッドアニーリング法

SA 法は、S. Kirkpatrick, C. D. Gelatt, M.P. Vecchi ら、および V. Cerny が考案した方法^{8,9)}で、大域的最適化問題の解を見つけるための汎用的な確率的アルゴリズムであり、大域的最適解に対して、良い近似解を得ることが分かっている。SA 法は、金属の錬成における“焼きなまし”から名前が付けられており、解探索の初期では確率的に状態遷移する確率 (温度) を高めに設定し、徐々に温度を下げていくことで、大域的最適解を見つけやすくする、という方法である。一般的な勾配を利用した大域的最適解を探索する方法に比べ局所解に陥ることが少なく、理論的には大域的最適解を見つけることが可能であることがわかっている。

具体的な方法論について説明する。SA 法の説明については様々あるが、本稿では¹⁰⁾の解説を参考にした。まず、求めるべき評価関数を $E(x(k))$ とする。 $x(k)$ は更新ステップ k における状態を表す。本研究の時間割編成を例とすれば、式 2 の $L(X)$ が $E(x(k))$ 、テンソル X が $x(k)$ に相当する。SA 法の目的は、 $x(k)$ の状態を初期状態から次々と確率的に推移させ、最終的に評価関数 E が最小 (あるいは最大) となる $x(k)$ を発見することである。SA 法のアルゴリズムでは、生成関数、受理関数、徐冷関数の 3 つの関数が重要な役割を果たす。

まず生成関数は現在の状態 $x(k)$ を与えられて、次に推移する状態 $x'(k)$ を、確率分布 $G(x|x')$ に従って返す関数である。組み合わせ最適化問題の場合は、 $x'(k)$ は $x(k)$ の近傍となるように推移させるのが好ましい。もし状態推移が等確率で推移する場合は、確率分布は以下の式となる。

$$G(x, x') = \frac{1}{n(x(k))} \quad (9)$$

ここで $n(x(k))$ は状態 $x(k)$ の近傍にある状態の数を表す。等確率で推移する場合は、上式を用いて新しい状態を生成するのが一般的であるが、本研究における時間割編成の場合は、組まれた時間割の状態の近傍を把握することが困難であることと、近傍の状態が通常はあり得ない時間割の状態になっていることも多いため、上式を利用した生成はしない。詳しくは後述する。

次に受理関数は、現在の状態 $x(k)$ の評価関数 E と、状態 $x'(k)$ の評価関数 E' を計算し、その差をもとに次の状態 $x'(k)$ を受け入れるかどうか (真偽値) を返す関数である。

通常は以下の Metropolis の方法が受理関数として用いられる¹¹⁾。

$$A(E, E', T) = \begin{cases} 1 & \text{if } E' \leq E \\ p = e^{-(E-E')/T} & \text{otherwise} \end{cases} \quad (10)$$

T は温度であり、評価値が増大する方向への推移確率に影響を与えるパラメータである。すなわち温度が高い場合は状態 $x'(k)$ が状態 $x(k)$ よりも評価値が悪くても、 $x'(k)$ に推移する確率が高くなり、反対に温度が低い場合は、評価値が改善される状態へと推移する。上式は、評価値の差分が 0 より小さい場合は新しい状態を採択することになる。一方で、評価値の差分が 0 以上の場合、式中の下段の確率 p で新しい状態が採択されることになる。これをプログラム上で実現する場合は、区間 $[0,1]$ の一様乱数 ξ を発生させ、 $p > \xi$ の場合に新しい状態を採択するように設計すればよい。

次に温度 T は徐冷関数によってスケジューリングされる。徐冷関数として利用される関数は、最適解の漸近収束性を保証するためには、以下の式よりも急速に冷やしてはならない (T_0 は初期温度を表す)。

$$T_{k+1} = \frac{T_0}{\log k} \quad (11)$$

しかし、上式でスケジューリングすると、長い時間計算をする必要があるため、運用上は、以下の式を用いることが多い。

$$T_{k+1} = \gamma T_k \quad (0.8 \leq \gamma < 1.0) \quad (12)$$

なるべく γ は 1.0 に近い値にし、ゆっくりと温度を下げるように設定したほうが最適解にたどりつきやすい。

SA 法の基本アルゴリズムの手順を以下に示す。

手順 0 $k = 0$, 初期温度 T_0 の設定, 初期状態 $x(0)$ の設定, 初期評価値 $E(x(0))$ の計算。

手順 1 生成関数で次に推移する状態 $x'(k)$ を生成する。

手順 2 受理関数に従い、新しい状態 $x'(k)$ を採択するかどうかを決める。採択の場合は新しい状態へと推移させる。

手順 3 現温度で平衡状態となったかどうかを判断する。平衡状態と判断されたら、手順 4へ、そうでな

れば手順 1 に戻る。

手順 4 徐冷関数によって T_k を更新。

手順 5 基底状態であるかどうか判定し、基底状態であれば終了、そうでなければ $k = k+1$ として手順 1 に戻る。

アルゴリズムにおいて手順 1～手順 3 は、現在の温度で平衡状態が実現されるまで十分な探索を行う。各温度で十分に探索を行うことで大域的最適解に至ることが保証される。一般的に、十分に多くの解が生成されたと考えられる状態になれば平衡状態と呼び、さらに平衡状態となった場合に収束された解が得られている状態を基底状態と呼ぶ。通常は、一定回数手順 1～手順 3 を繰り返すことで平衡状態とし、各温度で平衡状態になった際に受理された解の個数が一定個数となったときに基底状態とする。本研究では手順 1～手順 3 を 30 回程度繰り返すことで平衡状態とした。

時間割の自動編成のアルゴリズムについて

本研究では、時間割の自動編成法として SA 法をベースとしたアルゴリズムを提案する。本手法は SA 法に準じた手法ではあるが、生成関数における状態の生成は授業時間割編成に即した本研究独自の生成方法である。簡単に述べると、人間が時間割編成をするならば、時間割に配置された授業科目を制約に矛盾していないかを考慮し、矛盾しているようであれば、授業科目を別の授業科目と入れ替えをする。それを繰り返すことで、最終的に最適解が導き出される。この人間が行う授業配置のプロセスを本生成関数では取り入れているところが、本研究独自だと言える。またコース科目については、コース科目の中まで考慮した生成はアルゴリズムが複雑になるため、本研究では複数教員、複数教室、複数学科で開講されるひとつの科目として扱うこととした。時間割の自動編成をする前に予めコース科目の編成を手動で行う必要があるが、本自動編成アルゴリズムの枠組みでもコース科目を問題なく時間割編成に組み込むことが可能となった。

以下に、時間割自動編成のアルゴリズムを示す。

手順 0 $k = 0$, 初期温度 T_0 の設定, 初期時間割テンソル $X(0)$ の設定, 初期評価値 $E(X(0))$ の計算。

手順 1 生成関数で次に推移する時間割 $X'(k)$ を生成する。

- 手順 2 受理関数に従い、新しい時間割 $X'(k)$ を採択するかどうかを決める。採択の場合は新しい時間割へと推移させる。
- 手順 3 現温度で平衡状態となったかどうかを判断する。平衡状態と判断されたら、手順 4へ、そうでなければ手順 1に戻る。
- 手順 4 徐冷関数によって T_k を更新。
- 手順 5 基底状態であるかどうか判定し、基底状態であれば終了、そうでなければ $k = k+1$ として手順 1に戻る。

通常、初期状態の設定は乱数を用いてランダムに設定される。しかし、時間割の場合、完全に乱数で設定すると、基底状態になるまでかなり計算に時間がかかる。これは生成関数での新しい時間割 $X'(k)$ の生成においても同様である。そのため、本研究の時間割編成においては次に挙げる制限の範囲でランダムに状態を設定するようにすることで、計算時間を短縮するようにした。

- (1) 授業科目の配置は開講されるクラス内に限定する。
- (2) 週内に同じ授業が配置されないよう制限する。
- (3) 同じコマで複数の授業科目が重ならないように配置する。
- (4) 担当教員は授業科目に紐付けて配置する。
- (5) 教室が固定されている授業科目は科目に紐づけて配置する。もし固定されていなければランダムに教室を配置する。

(1) では、科目毎に開講されるクラスはすでに決まっているので、決められたクラスの中でランダムに配置するよう限定をする。(2) では、授業科目が週内で複数開講されることはないので、一度配置をした授業科目がほかのコマで配置されることがないように限定をする。ここで (2) の限定をすることで、週内で複数の授業が開講されることがないようにペナルティ項として設定をする必要がなくなる。言い換えるとわざわざペナルティ項の計算をする必要もなくなり計算時間の短縮となる。(3) では、同じコマ（同一の時間帯とクラス）において、複数の授業を配置しないよう制限をする。状態の配置においてこのような制限をかけておけば、(2) と同様、ペナルティ項として制約条件を設ける必要もなく、計算時間の短縮となる。(4) では、授業科目の担当教員はすでに決まっており、尚且つ、固定された項

目なので、授業科目の配置とともに科目に紐付けられた教員の状態も 1 とする。(5) では、授業科目によって開講できる教室が限られている場合については、(4) と同様に、科目に紐付けられた教室の状態を 1 とする。このような授業科目には、例えばマルチメディアセンターを利用する情報科学や実験室や実習工場などを利用する実験や実習、そして受講人数が 150 人を超えるような授業科目が挙げられる。他の授業科目は受講人数の規模に応じて提案できるいくつかの講義棟教室をランダムに配置する。

初期時間割状態の設定と生成関数での状態設定において、上記のような制限をかけることで、探索する空間が制限でき、基底状態を発見するまでの時間を短縮できる。

ただし、上記のような制限によって、生成関数のプログラム設計は工夫をする必要がある。通常では状態 (0 と 1 の配置) を乱数を利用してランダムに設定するだけでかまわないが（現状態の近傍でランダムに設定、という制限はあるが）、時間割の場合は上記のような制限をかけるため、そのような簡便な状態設定はできない。以下に生成関数の処理を処理 1、処理 2、処理 3 の 3 つに分割し、それぞれの具体的な処理内容を説明する。

生成関数：処理 1

どこのコマにも配置されていない授業科目があるかどうかで分岐をする。コマに配置されていない授業科目がある場合は処理 2 へ、ない場合は処理 3 へ進む。

生成関数：処理 2

時間割のコマに配置されていない授業科目がある場合は、その授業科目を開講クラス内でランダムに配置する。具体的には時間帯をランダムに選択し、開講されるクラスと選んだ時間帯のコマにその授業を配置する。もし、配置するコマにすでに他の授業科目が配置されている場合は置き換える作業をする。具体的にはコマにもともと配置されていた授業科目の情報（授業科目、担当教員、教室）を全て 0 にし、新しい授業科目の情報を 1 にする。そして、もともと配置されたい授業科目は、開講されるクラス内の空コマ（授業が配置されていないコマ）がある場所に再配置をする。もし空コマがない場合は何もしない。その後、処理 3 へ進む。

生成関数：処理 3

ここでもさらに処理の分岐がある。開講制約があるコマに配置されている授業科目が存在するならば、その授業科

目を同開講クラス内においてランダムに配置させる。配置処理の具体的な内容は処理 2 と同様である。また開講制約があるコマに配置されている授業科目が存在しないなら、時間割に配置されている全ての授業科目からランダムに選択し、選択された授業科目を同開講クラス内においてランダムに配置させる。配置処理の具体的な内容は処理 2 と同様である。この工夫は開講制約があるコマに配置された授業科目を優先的に移動配置させることで最適解を導きやすくさせている。

以上が生成関数の処理の内容である。生成処理関数以外は SA 法と同じ処理となる。

提案したアルゴリズムを用いた 時間割の自動生成の検証実験

提案した自動時間割編成アルゴリズムを開発し時間割を生成させた。開発は Mathworks 社製の Matlab で行った。開発および実行をした計算機は、OS が windows 10 64bit, プロセッサが Intel Core i9-9920X (3.50GHz), メモリ 128GB である。

また提案する自動時間割編成アルゴリズムで、時間割を編成する際には、予め授業科目や担当教員、開講制約等の情報をデータベースとして作っておく必要があるため、Microsoft Excel を用いてデータベースを作成した。本検証では令和 4 年度の授業時間割を参考にデータベースを構築した。また非常勤講師などの授業可能日についても令和 4 年度の情報を適用した。コース科目についても同様に令和 4 年度の時間割のコース科目編成を適用した。

一方、SA 法においては、 $T_0 = 100$, $\gamma = 0.99$ と設定してシミュレーションを行った。ステップ数 k が 2000 になるまでシミュレーションを繰り返すようにし、もし $E(X(k))$ が 0 となれば、その時点で基底状態になったとみなしシミュレーションを終了した。

生成した結果の一例を図 4 と図 5 に示す。図 4 は、本提案手法で自動的に生成された前学期の授業時間割、図 5 は後学期の授業時間割を示す。どちらにおいても本研究で設定した制約条件の総ペナルティが 0 となる時間割が生成された。また 10 試行の生成を行ったうち 1 試行が 2000 回のシミュレーションを行ってもペナルティが 0 になることがなく、最適な授業時間割を生成できなかった。一方で 9 試行については約 300 回の繰り返しシミュレーションでペナルティが 0 になり最適解に収束することがあ

できた。このことから、本提案手法は約 10% で時間割生成を失敗するが、約 90% 程度で最適解に収束することができ、十分利用できることが示唆された。

計算時間については、本研究で利用した計算機では 2000 回のシミュレーションで 30 分程度の時間を要した。そのため 300 回程度で最適解が生成された場合は約 10 分程度で時間割が生成できた。

ま と め

本研究では、水産大学校の授業時間割の自動生成をする方法としてシミュレーテッドアニーリング法を採用し、その授業時間割自動生成のアルゴリズム構築を試みた。その結果、提案した方法によって時間割が自動的に生成できることが確認できた。また本研究ではシミュレーテッドアニーリング法における生成関数部のアルゴリズムに工夫を加えることで、最適な授業時間割の解が生成しやすくなった。また計算時間も抑えることができた。ただし解の生成においては、局所解に陥る場合もあった。局所解に陥った場合は、再度、シミュレーションをし直すことで対応が可能であるが、局所解になるべく陥らないようなアルゴリズムの工夫も今後必要だと考える。本研究の今後の展開としては、人間的な配慮を制約条件に含めること、実用化に向けたユーザーインターフェースの開発をすることが挙げられる。特に実際の運用を考えると、本手法の利用を前提としたデータベース構築が必要となる。データベースの構築は一般的には難しく入力ミスも多くなるので、ユーザーインターフェースなどを利用することで運用がしやすくなると考えられる。

謝 辞

伊澤瑞夫氏（当時、水産大学校特命准教授）には授業時間割編成について多数の助言を頂いた。また水産大学校教務課にも情報提供に関して協力を頂いた。この場でお礼を申し上げる。

参考文献

- 1) Chen M C, Sze S N, Goh S L, Sabar N R, Kendall G: A survey of university course timetabling problem: perspectives, trends and opportunities. IEEE Access, 9,

- 106515-106529 (2021)
- 2) 大坪正和, 倉重賢治, 亀山嘉正, 久保貴広: 中学校時間割編成問題における制約条件の重み付けの自動調整. 日本経営工学会論文誌, **61(5)**, 293-301 (2010)
 - 3) 柿本陽平, 高橋弘毅, 島川陽一: 制約充足問題を線形計画法で解く場合の制約条件の緩和とその評価-時間割編成を例に. 日本経営工学会論文誌, **66(4)**, 348-354 (2016)
 - 4) 堀尾正典: 汎用プロジェクトスケジューラの大学時間割問題への適用. 名古屋学芸大学教養・学際編・研究紀要, **4**, 61-74 (2008)
 - 5) 伊藤美登, 佐々木美裕, 鈴木敦夫, 伏見正則: 大学時間割編成モデルの研究. アカデミア. 情報理工学編: 南山大学紀要, **12**, 87-98 (2012)
 - 6) Aycan E, Ayav T: Solving the course scheduling problem using simulated annealing. In: Proceedings of IEEE International Advance Computing Conference, IEEE, 462-466 (2009)
 - 7) Beligiannis G N, Moschopoulos C, Likothanassis S D: A genetic algorithm approach to school timetabling. *Journal of the Operational Research Society*, **60(1)**, 23-42 (2009)
 - 8) Kirkpatrick S, Gelatt C D Jr., Vecchi M P: Optimization by simulated annealing. *Science*, **220(4598)**, 671-680 (1983)
 - 9) Cerny V: Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, **45**, 41-51 (1985).
 - 10) Rosen B E, 中野良平: シミュレーテッドアニーリング: 基礎と最新技術. 人工知能学会誌, **9(3)**, 365-372 (1994)
 - 11) Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H, Teller E: Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087-1092 (1953)

		月					火					水				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1年生	D															
	F															
	M		コース科目 (II-3)	水産数理科学 No.13	基礎生物学 No.41 No.43		コース科目 (II-4)	水産学概論 No.41No.43	コース科目 (I-2)	コース科目 (I-1)		コース科目 (II-2)	英語セミナー No.13	海と漁業生 産 No.41 No.43	水産数理科学 基礎セミナー (「数学編」) No.43	
	S															
	A															
2年生	D			水産政策史 No.43	食料経済論 No.36								水産経営学 No.41No.43			
	F			水産政策史 No.43	地文航海学 No.42					東シナ海・ 日本海資源 論 No.41		基礎漁具力学 No.41	水産経営学 No.41No.43			
	M		応用生物学 No.41 No.43	製図 製図室			環境倫理 No.41No.43	コース科目 (I)	コース科目 (II)			流体力学 No.36	材料力学 No.36	電気工学 No.42		
	S			生物化学 No.35	食料経済論 No.36								基礎微生物学 No.42	分析化学実験 食品実験室		
	A			水産政策史 No.43						浮遊生物学 No.42		水族病原微生物学 No.35				
3年生	D			英語(英会話) (2組) No.23 No.31	英語(英会話) (1組) No.31 No.34		水産地域振 興論 No.21					水産行政論 No.43		食品保蔵学 No.36	水産フイー ルド調査演 習 No.34	
	F		水産音響学 No.21	漁業計測学実験 No.1			英語(読解) F3 No.35	国際漁業管 理学 No.32	電子航海学 No.36	漁船運動力学 No.34		水産行政論 No.43	資源管理漁 具設計論 No.23	天文航海学 No.13	海事法規 No.21	
	M		環境計測学 No.35	英語(英会話) (2組) No.23 No.31	英語(英会話) (1組) No.31 No.34		水産環境学 No.36		船用機械実験 蒸気実験室			電気電子機 器 No.23			船用補機 No.25	
	S		食品加工学 No.36		遺伝子工学 No.25		水産環境学 No.36			英語(読解) S3 No.13			水産経営学 No.41No.43	食品保蔵学 No.36	食品衛生学 No.42	
	A		魚類増殖学 No.42	魚病学実験 生物実験室			水産遺伝資 源学 No.42			魚類行動学 No.35		水産行政論 No.43	水産経営学 No.41No.43			
4年生	D															
	F			航海英会話 No.32										航海情報計測学実験 No.1		
	M															
	S															
	A															

Figure 4(a) Timetable for the first semester automatically generated by the proposed method

木					金				
基礎化学 No.36 No.43	水産数理解 科学基礎セ ミナー No.13	食品安全利 用学 No.41No.43	コース科目 (II-1)			基礎工学演 習1 No.21 No.22 No.23No.31 No.32	水産経済学I No.41 No.43	水産と生物 No.41 No.43	コース科目 (I-3)
		漁業構造論I No.13					水産物市場 構造論 No.13		
			海洋物理学 No.35				水産動物学 No.36		
船用機関学 概論 No.41		機械工作実習 工作実習室			水産化学 No.41No.43		コース科目 (IV)	コース科目 (III)	
物理化学工 学 No.13									
	水産植物学 No.43	浮遊生物学実験 生物実験室				水産動物学 No.36			
		漁業構造論 II No.23		水産特論 No.41No.43			水産人物育 成論 No.22	水産資源経 営管理論 No.22	
漁業情報解 析学 No.42	漁船安全学 No.1	資源管理解 析学演習 No.22		水産特論 No.41No.43	水産資源環 境学 No.35		漁船運用学実習 海生実験室		
		水産冷凍工 学 No.36		水産特論 No.41No.43		海洋機械設 計 No.25	内燃機関 No.42	制御工学 No.25	
	機器分析実験・食品分析実験I 食品実験室			水産特論 No.41No.43	栄養生理学 No.25	機器分析実験・食品分析実験II 食品実験室			
水族栄養学 No.35		英語（読 解）A3 No.25		水産特論 No.41No.43	水族生理学 No.42	水族組織学 No.31	水族病原微生物学実験 生物実験室		
		漁船運動力学実験 No.1			船舶衛生管 理論 No.36	セミナー科 セミナー室			

1 年生コース科目

	コース科目 (I-1)	コース科目 (I-2)	コース科目 (I-3)
A コース	基礎解析学 No. 43	スペイン語 No. 13	国際社会と 法 No. 43 No. 42
B コース	中国語 No. 13	海洋文化論 No. 43 No. 42	基礎解析学 No. 36
C コース	基礎解析学 No. 43	フランス語 No. 25	国際社会と 法 No. 43 No. 42
D コース	フランス語 No. 35	海洋文化論 No. 43 No. 42	基礎解析学 No. 36

1 年生コース科目

	コース科目 (II-1)	コース科目 (II-2)	コース科目 (II-3)	コース科目 (II-4)
E コース	英語（読 解） No. 13	英語（文 法） No. 13	コンピュ ータ基礎 LL, 情報処 理演習室	
F コース	英語（文 法） No. 13	英語（読 解） No. 25		コンピュ ータ基礎 LL, 情報処 理演習室
G コース		コンピュ ータ基礎 LL, 情報処 理演習室	英語（文 法） No. 13	英語（読 解） No. 25
H コース	コンピュ ータ基礎 LL, 情報処 理演習室		英語（読 解） No. 25	英語（文 法） No. 13

2 年生コース科目

	コース科目 (I)	コース科目 (II)	コース科目 (III)	コース科目 (IV)
A コース	スペイン語 No. 13	海洋文化論 No. 43 No. 42		英語 (TOEIC) No. 13
B コース	英語（読 解） No. 25	中国語 No. 41	国際社会と 法 No. 43 No. 42	
C コース	英語 (TOEIC) No. 35	海洋文化論 No. 43 No. 42	フランス語 No. 13	確率統計学 No. 36, LL, 情報処理演 習室
D コース	確率統計学 No. 36, LL, 情報処理演 習室	英語（読 解） No. 35	国際社会と 法 No. 43 No. 42	フランス語 No. 25

Figure 4(b) Timetable for the first semester automatically generated by the proposed method

		月				火				水			
1年生	D									線形代数 No.21	漁具漁法 No.36	基礎物理学A No.36	
	F									情報科学 F LL情報処理演習室			
	M	海洋水産機械概論 No.41 No.43	海洋環境と機械 No.41 No.43	水産資源論 No.41No.43		体育理論 No.41No.43	コース科目VI	哲学 No.41No.43	コース科目IV	工業数学 No.34	情報科学M LL情報処理演習室	工業力学 No.42	増養殖技術論 No.41 No.43
	S									情報科学S No.31	分析化学 No.31	基礎物理学A No.36	
	A									生物学基礎実験 生物実験室		基礎物理学A No.36	
2年生	D	魚類学 No.36		水産流通経営セミナー1 No.31			水産制度論 No.22				水産労働論 No.34	水産流通経営セミナー2 No.32	水産遺伝育種学 No.36
	F	海洋気象学 No.42	資源解析学 No.36			地文航海学II No.25							
	M		機械工作実習 工作実習室			材料と機械工作法 No.36		コース科目V	応用物理学 No.41 No.43	技術者倫理 No.23 No.35 No.36No.41 No.43	機械力学 No.42	伝熱工学 No.35	
	S	魚類学 No.36		食品化学 No.32								食品分析 No.22	水産遺伝育種学 No.36
	A	水産動物増殖学 No.21	水産植物学実験 生物実験室			水産動物学実験 生物実験室					魚病診断治療学 No.22	増殖生態学 No.25	水産遺伝育種学 No.36
3年生	D					HACCP No.36	水産流通加工ビジネス論 No.21	水産統計データ解析 情報処理演習室	英語(読解)D No.23		水産経営分析論 No.31		
	F		英語(英会話)(1組) No.21 No.22No.23	英語(英会話)(2組) No.21 No.22No.23		水産資源環境学演習 情報処理演習室		漁具学実験 No.31		漁船運用学 No.42	天文航海学II No.13		
	M	蒸気工学 No.25		海洋エネルギー工学 No.35		機械制御学 No.35			工学系の力学 No.42				機関システム学 No.13
	S	海洋天然物化学 No.23	英語(英会話)(1組) No.21 No.22No.23	英語(英会話)(2組) No.21 No.22No.23		HACCP No.36	食品機能学 No.42	水産伝統食品科学 No.32	セミナーS No.34		生物化学・微生物学・食品衛生学実験1 食品実験室		
	A	藤場・干潟保全生態学 No.22	英語(英会話)(1組) No.21 No.22No.23	英語(英会話)(2組) No.21 No.22No.23		水族防疫学 No.22		水族生理学実験 生物実験室					水産植物増殖学 No.34
4年生	D												
	F												
	M												
	S												
	A												

Figure 5(a) Timetable for the second semester automatically generated by the proposed method

木					金				
コース科目I	コース科目II	情報科学D 情報処理演習室	コース科目V		水産食品科学 No.41 No.43	水産経済学II No.41No.43	コース科目III	心理学 No.41No.43	
		基礎物理学B No.36							
		基礎物理学B No.36							
		情報科学A LL							
コース科目I	コース科目II		漁業協同組合論 No.21			解析学 No.25	コース科目III	コース科目IV	
		漁業計測学 No.13	漁具力学 No.36		航海情報計測学 No.25	数値解析 LL			
		海洋機械実験 蒸気実験室 内燃実験室 工作実習室				エレクトロニクス No.36			
					応用微生物学 No.42	酵素化学 No.42			
		魚類学実験 生物実験室				沿岸環境生態学 No.21			
水産国際関係論 No.31					水産特論 No.41No.43	セミナーD No.23	水産物ロジスティックシステム論 No.21	水産資源経営管理論 No.21	
資源管理論 No.35	漁船システム論 No.35	国際漁業管理学演習 No.25							
		英語（読解）M No.35					海洋環境実験 蒸気実験室 内燃実験室		
生物化学・微生物学・食品衛生学実験2 食品実験室						食品生命科学 No.31	生物化学・微生物学・食品衛生学実験3 食品実験室		
水族栄養学実験 生物実験室	水族栄養学実験 生物実験室					水族遺伝学実験 生物実験室	水産増殖セミナー No.23		

1 年生コース科目

	コース科目I	コース科目II	コース科目III	コース科目IV	コース科目V	コース科目VI
A コース	社会学 No. 41, No. 43	法学 No. 41, No. 43	英語（作文） No. 13		英語（読解） No. 35	スペイン語 No. 13
B コース	歴史学オンライン	英語（作文） No. 13	文学大木 No. 41, 43	中国語 No. 13		英語（読解） No. 35
C コース	社会学 No. 41, No. 43	法学 No. 41, No. 43	英語（読解） 猪熊 No. 25	英語（作文） 猪熊 No. 25	フランス語 No. 25	
D コース	歴史学オンライン		文学大木 No. 41, 43	英語（読解） 納富 No. 35	英語（作文） No. 13	フランス語 No. 25

2 年生コース科目

	コース科目I	コース科目II	コース科目III	コース科目IV	コース科目V
A コース	歴史学オンライン	確率 No. 36, LL, 情報処理演習室	文学 No. 41, 43	英語（読解） No. 13	スペイン語 No. 13
B コース	社会学 No. 41, No. 43	法学 No. 41, No. 43	確率 No. 36, LL, 情報処理演習室	英語（TOEIC） No. 35	中国語 No. 25
C コース	歴史学オンライン	フランス語 No. 25	文学 No. 41, 43		英語（読解） No. 35
D コース	社会学 No. 41, No. 43	法学 No. 41, No. 43		フランス語 No. 25	英語（TOEIC） No. 42

Figure 5(b) Timetable for the second semester automatically generated by the proposed method