

制約に基づく対話型時間割編成システム

西 森 雄 一[†] 狩 野 均[†] 西 原 清 一[†]

時間割編成問題は、制約条件を満たすように、科目を時間帯に割り当てる探索問題である。従来のシステムでは、カリキュラムが単純な高校の時間割編成が対象であったが、本論文では、より複雑な大学の時間割を制約充足パラダイムにより編成する方法を提案する。大学の時間割編成問題は、課せられる制約条件が多様であり、また探索空間が膨大である、という問題点がある。そこで本論文では、制約を、個別の科目への制約、2科目間の関係に関する制約、時間割表全体の整合性に関する制約に分類整理することにより、前者の問題点を解決する。また、後者に対しては、制約に違反点数を与え、その違反点数の最小化を規範とする山登り法を導入し、さらに対話によってユーザと協調して探索を進めることにした。本システムは、編成の基になる時間割を作成する初期割当作成と、その時間割から不具合のある割当を反復改善する割当改善の2段階より成る。本システムを筆者らの大学の実際の時間割編成に適用した結果、人手による編成では数時間かかるところを、数十分で編成ができることを確認した。

Constraint-based Interactive Timetabling System

YUUCHI NISHIMORI,[†] HITOSHI KANO[†] and SEIICHI NISHIHARA[†]

Timetabling is a search problem to assign the subjects to the time-slots so that various constraints are satisfied. Conventional timetabling systems are applicable to comparatively simple problems such as high school timetabling. In this paper, we propose a method for scheduling university timetables, which have two major problems: first, it is hard to represent concretely all various kinds of constraints, and, second, the search space is extremely large. To resolve the first problem, we classify all the constraints into three groups: the constraints claimed per each subject, the binary constraints to be satisfied between two subjects, and the constraints concerning the total consistency of the timetable. As for the second problem, we introduce penalty per constraint to make the min-conflicts hill-climbing strategy try to minimize the total penalty. Further, our system interacts with the user to exchange suggestions each other. Timetabling is performed in two steps: generation of an initial assignment and iterative improvement of the total penalty to optimize the timetable. Applying our interactive method, we actually constructed timetables for our university, where the planning time is considerably reduced.

1. はじめに

時間割編成問題とは、各科目が実施される時間帯をまとめた時間割を編成する問題のことである。時間割編成は、制約充足問題¹⁾または最適化問題として多くの研究がなされている^{2)~6)}。しかし、時間割編成問題を解くにはカリキュラムに関する専門的な知識が必要で、その多くが計算機内で表現しにくく、表現できても膨大な計算時間と記憶領域を要する。また、それぞれの事例ごとに制約条件が異なるために有効な解法が見つけられておらず、編成は人手に頼らざるを得ないのが現状である。以上のような背景から、実際の時間

割編成に対応できる問題の表現方法や、それを効率良く解く手法の開発が望まれている。

従来、研究開発されてきた時間割編成システムは、高校と大学の時間割編成に分けられる。表1に高校と大学の時間割編成の違いについて示した。前者の高校の時間割編成は、クラスごとに英語・数学などの10程度の科目を実施する時間帯に割り振る問題である。高校では学年・クラスによって実施される科目が決まっており、クラス間の相互の関連性が低いことから、制約条件は比較的単純である。それに対して、大学では、学生が科目を選択する自由度が大きく、さらに科目どうしの関連性も高いことから、制約条件は複雑になる。

従来の時間割編成システムには以下のものがある。高校の時間割編成では、Cooperら²⁾が、編成の要求事項を簡単な言語で表現し、ヒューリスティクスを組

[†] 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba

表1 高校と大学の時間割編成の違い

Table 1 The difference between high school and university of timetabling.

項目	高校 ³⁾	大学(本問題)
カリキュラム	単純	複雑
編成	学年・クラス単位	学年全体
教官	学年ごとに独立	学年全体
学年間の関連性	低い	高い
科目間の関連性	低い	高い
科目数	約 700 [☆]	236
制約数	約 24000	約数百
制約密度 ^{☆☆}	約 30 (大)	5 未満 (小)

み込んだ木探索を行って編成を行うシステムを開発した。また、Yoshikawaら³⁾は、Constraint Relaxation Problem Solver (COASTOOL) によって時間割編成システムを開発した。Yoshikawaらが提案した手法は、まず辺整合を用いた Really-Full-Lookahead Greedy アルゴリズムで良質な初期割当を生成し、その後 Min-Conflict Hill-Climbing (MCHC) を用いて割当改善を行い、最適解を得るものである。一方、大学の時間割編成問題では、Burkeらによって、グラフカラーリング問題におけるヒューリスティクスを用いた解法⁵⁾、また、このヒューリスティクスを GA に組み入れた解法⁶⁾が提案されているが、実際規模の問題に対する有効性は示されていない。また、以上述べた方式では、編成中に人間の知識を活用することを前提としていないため、編成者の立場におけるきめ細かい編成が困難であると考えられる。さらに、大学の時間割編成問題のように、制約密度(制約の個数/変数の個数)の小さい問題は解くのが一般に難しく、上記の MCHC で探索しても、その効果は期待できない⁷⁾。

そこで、本システムでは、より難解で現実的な時間割の編成を目的とするために、ユーザとの対話によって、システムとユーザが協調して編成を行うことにした。大学の時間割のように複雑なカリキュラムにおける要求事項を、それぞれの科目に対する制約、2科目の関係に関する制約、時間割表の整合性に関する制約に分類した。このようにまとめることで、要求事項が簡潔かつ少数の制約で表現することができ、それぞれの制約は各対象データの参照とそれに対する簡単な手続きによって実現できる。さらに、表現の困難である

ような要求事項に対しては、対話操作で編成に介入することにより解決を図る。これにより、時間割編成にユーザの多種多様な要求を反映させることができる。また、MCHCの局所最適解に陥りやすいという欠点を回避するため、探索の過程において、適宜、人間に別の可能な候補を色別表示などで示し、選択の余地を残すことにした。

以下2章では、まず時間割編成問題を筑波大学第三学群情報学類の実例より定式化し、時間割編成における課題や問題点を整理する。さらに、その対策として、本システムでとった方針を示す。次に3章で、提案する手法について、制約の表現および探索アルゴリズムなどを詳しく述べた後、システムの処理手順を示す。最後に4章で本システムを上記実例に適用した実行結果について述べる。

2. 問題点と対策

2.1 時間割編成問題

時間割編成問題とは、基本的に科目を実施する学年、および時間帯を決定することである。ただし、「科目には担当する教官がおり、その教官が担当可能なように時間を決定すること」などの制約が存在しており、それらを満たすように時間割を編成しなければならない。

本論文では実例として、筑波大学第三学群情報学類の時間割編成問題について取り上げる。本問題において、時間割表に割り当てる科目データの一部を、表2に示した。表2のように、科目には、科目名・担当教官・履修学年・専攻グループ・分野グループなどの属性情報を持つ。さらに相互に関係がある科目も多く、制約が複雑になっている。次に、本システムで取り上げた制約を表3に示す。これらは時間割編成上の経験則や大学における教育方針を踏まえ、現場の専門家(カリキュラム編成担当教官)の協力を得て決定したものである。具体的には、通常のエキスパートシステムにおける知識ベースの構築手順に従った⁸⁾。まず、時間割編成上考慮すべき制約条件をアンケートの形で専門家から獲得した。次にシステム開発者が、これらを現在の時間割に当てはめ、疑問の点は専門家に対するインタビューによって制約を追加・改良した。最後に、システム構築後、過去の時間割に対して本システムを適用して、最終的な制約ならびに違反点数を決定した。

時間割編成問題を解くということは、すべての科目のある時間帯に、制約を満たすように割り当てることである。本問題では、制約に違反点数を導入し、それらの合計点数を最小化する最適化問題としてとらえることにした。

[☆] 高校の科目数が約 700 と多いが、実際には 60 クラスの合計である。それに対して、大学の方は複雑であるために分けることができにくい。

^{☆☆} 制約密度は、制約の個数/変数の個数で定義される。制約の個数は制約(定義)を実際のデータに適用した個数であり、変数の個数は科目の個数である。

表2 科目データの一部(筑波大学情報学類)
Table 2 Subject table (a part).

name	teacher	grade	course	group
代数学	池辺	1	必修	数学
代数学演習		1	必修	数学
幾何学	伊藤	1	必修	数学
情報科学 A	井田	2	必修	情報科学
情報科学 B	五十嵐	2	必修	情報科学
情報科学 C	中田	2	必修	情報科学
非数値処理	西原	2	通常 2	情報処理
数値計画法	稲垣	3	3 年共	情報数学
情報理論 1	海老原	3	情報科	情報理論
最適化工学	寅市	3	情報工	情報数学

表3 時間割編成における制約
Table 3 The constraints for timetabling.

分類	制約番号	時間割編成に含まれる制約	違反点数
科目単位	c1	科目は割当可能な時間帯に割り当てられなければならない。	10
	c2	担当教官の都合の悪い時間帯に割り当ててはならない。	10
	c3	担当教官の意向を考慮する。	5
2科目間	c4	2科目間に関係があれば、それを満足しなければならない。	10
	c5	2科目が学期集中のとき、なるべく時限連続にする。	2
	c6	c5が満たされなときは異なる曜日に割り当てる。	3
	c7	2科目が学期連続のとき、なるべく同曜日同時限に割り当てる。	2
	c8	2科目が時限連続のとき、なるべく昼休みを挟まない。	3
全体	c9	教官は同時に2つ以上の科目を担当できない。	10
	c10	科目には course の指定があり、異なる course の科目どうしであれば同じ時間帯に割当可能である。	10
	c11	3つ以上の科目を同じ時間帯に割り当てることはできない。	10
	c12	2つの科目を同じ時間帯になるべく割り当てない。	1
	c13	低学年および高学年の必修科目を同時に実施しない。	3
	c14	同曜日に実施される同分野の科目の時限数を制限する。	3

2.2 時間割編成における問題点

時間割編成システムには2つの重要な課題がある。

- (1) 制約の表現方法の開発
- (2) 探索アルゴリズムの開発

(1)の問題点は、制約が多様であり、計算機では実際に表現しきれないことである。

時間割編成問題では、すべての制約を充足することは現実的ではなく、なるべく満たされることが望ましいという制約が存在する。このような弱い性質を持つ制約を特別に扱うことは、制約の扱いを繁雑にするこ

とになる。また、時間割編成問題における制約は、その制約を適用する対象が様々である。これは、たとえば科目別に考慮しなければならない制約(科目単位制約)、2科目の関係に対する制約(2科目間制約)、割り当てられた時間割表全体の整合性をとるための制約(全体制約)などがあり、これらを計算機で扱う場合には、表現方法がそれぞれ異なることになる。

さらに時間割編成システムでは、複雑なカリキュラムをすべて制約で表現しきれないことが多い。たとえば、科目「情報工学基礎実験」では、実際には計算機台数の制限により学生を複数のグループに分け、講義と演習を交互に実施しているが、これを制約として表現することは難しい。たとえ表現できても、問題の複雑化を招き、探索コストを増大させることになる。逆に、表現のできる制約のみで構成された時間割編成問題では、多数の解の存在が予想され、この中から自動的に探索して、表現されなかったユーザの要求を満たす解を得ることは困難である。

(2)の問題点は、探索空間が膨大であることと、制約密度が小さいことがあげられる。時間割編成問題は問題の規模が大きく、制約の種類も多いため、探索空間が膨大である。たとえば科目数が75、時間帯数が75で重複のない比較的単純な大学1学年の時間割表の編成では、探索空間のサイズは 10^{109} (=75!)になる。このため、通常の本探索法などでは対処しきれない。これに対し、高校では学年・クラス単位で編成するため、全体として大規模な問題でも、探索空間は大学ほど大きくないと考えられる。

探索手法を検討する際には、その問題の制約密度を考慮する必要がある。制約密度は、制約数を変数(=科目)の個数で割ったものとして定義される。Mintonら⁷⁾によると、MCHCは、制約密度が大きい場合に効果的な性能を示すが、制約密度が小さい場合は、局所最適解に陥る割合が多くなる。表1より、高校を対象とした時間割は、制約数が膨大であるため、制約密度が約30と大きい。そのため、MCHCで効率的に探索を行うことができる。しかしながら、大学の時間割は、制約密度が5未満と小さい。したがって、高校の時間割編成を解くためのアプローチを大学の時間割編成に用いても、同じような結果は望めないと考えられる。

2.3 基本方針

2.2節の問題点の対策として、本システムでは次の4つの方針をとることにした。

- (1) 重要度の高い制約に違反したとき、より多くの罰則を与えられるように、制約の重要度を違反

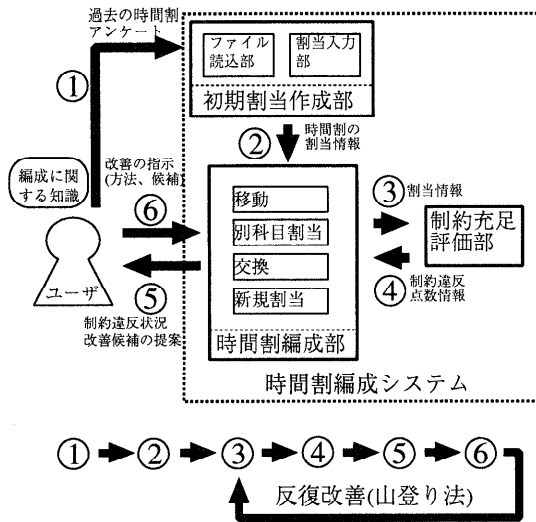


図1 本システムの構成
Fig.1 The framework of the system.

点数で表現する

- (2) 多種多様な制約を科目単位制約, 2科目間制約, 全体制約に分類整理する
- (3) 制約違反点数を最小化する戦略を用いた山登り法 (MCHC) を導入し, さらに対話によってユーザの判断で探索ができるように改良する
- (4) システムで表現が困難な制約を時間割編成に活用させるために, ユーザと協調して探索を進める対話型システムとする

以上を実現するために, 本システムは図1で示す構成とすることにした. 本システムは, 初期割当作成と割当改善の2段で構成され, 後者が対話によって探索を進めることで, 不具合のある割当を反復改善するものである. 本システムでの編成の流れを, 図1の矢印で示した.

初期割当作成は, 編成したい時間割の基となるような過去の時間割や, 教官からの要望をまとめたアンケートから, 初期割当を作成する. また, ユーザが適当な初期割当を入力する.

割当改善は, 時間割編成部と制約充足評価部で実施する. 時間割編成部は, ユーザとの対話により時間割を編成するものである. 制約充足評価部から受け取った違反点数情報を基に, ユーザに制約違反の状況を視覚的に示し, かつどのように変更すればよいかを提案する. 制約充足評価部は, 時間割編成部で作成した時間割の候補が制約を満たしているかどうか判定し, 違反点数を計算する.

3. 制約に基づく時間割編成

3.1 問題の定式化

2.1節で示した時間割編成問題を計算機で解くために, 以下のように定式化する. 本問題を次の4つ組で定義する.

$$\text{時間割編成問題} = (U, L, C, W)$$

$$U = \{ \text{科目 } 1, \dots, \text{科目 } M \} \quad : \quad \text{科目 } i \text{ の集合}$$

科目 i には以下の属性が定義されている (表2).

- $name(i)$: 科目名
- $teacher(i)$: 担当教官
- $grade(i)$: 履修学年
- $course(i)$: 専攻グループ
- $group(i)$: 分野グループ
- $slot(i)$: 割当先時間帯

$$L = \{ l_1, \dots, l_N \} \quad : \quad \text{時間帯 } l_j \text{ の集合}$$

$$l_j = (\text{学期, 曜日, 時限})$$

$$C = \{ c_1, \dots, c_P \} \quad : \quad \text{制約 } c_k \text{ の集合}$$

$$W = \{ w_1, \dots, w_P \} \quad : \quad \text{制約 } c_k \text{ の重み } w_k \text{ の集合}$$

表2で示した科目表は U , 時間割表は L , 表3で示した制約は C , 表3の右端の欄の違反点数は W である. それぞれの科目には, 表2の各欄に対応した属性が存在する. ここで, 時間割編成問題を解くということは, 制約を満たすように, すべての科目 i の $slot(i)$ を決定するということである.

3.2 制約の表現方法

本システムでは, 制約は手続きで表現されている. 本システムで導入した制約は2.2節の表3で示した14個であり, それらは次のように3つに分類できる.

$$C = \{ C_U, C_R, C_E \}$$

これらの制約の定義を以下に示す.

(1) 科目単位制約 C_U

科目単位制約は, 制約の対象が科目単位である制約のことである. 科目単位制約には以下の制約がある.

$$C_U = \{ c_1, c_2, c_3 \}$$

科目単位制約は, 次の表4の形式の表を参照する (c_1 は表4, c_2, c_3 は付録の表6, 表7). 表の左欄は, 科目番号または教官名である. 表の右欄には, (学期, 曜日, 時限)の形式で表記された時間帯が入る.

時間帯の3つの要素には, 1つの値もしくは, [値, 値, ...]の形式である値のリスト, すべての値を示す*が入る (たとえば, 科目番号29: 「新入生オリエンテーション」は, 1学期実施のため, 2, 3学期の

表4 割当不可能時間帯表

Table 4 The table of rejected time-slots per subject.

科目	割当不可能時間帯
29	([2,3], *, *)
56	(*, *, [1,2])
⋮	⋮

表5 2科目間関係表

Table 5 The table of binary relation's of subjects.

(科目 k_1 , 科目 k_2)	関係
(1, 2)	時限連続
(1, 3)	学期集中
(30, 31)	学期連続
(31, 32)	学期連続
⋮	⋮

どの時間帯も割当不可である)。

科目単位制約 c_1 の制約充足判定の内容は次のとおりである (c_2, c_3 についても同様。付録参照)。

c_1 : 科目 i の割当時間帯 $slot(i)$ が表4の割当不可能時間帯と一致していれば、制約違反。

(2) 2科目間制約 C_R

2科目間制約は、2つの科目間の制約条件のことである。

2科目間関係は次の表5のように与えられている。

表5の関係の定義は以下のとおりである。

- 時限連続 ... 科目 k_1 と科目 k_2 の割当時間帯の時限が連続である。
- 学期連続 ... 科目 k_1 と科目 k_2 の割当時間帯の学期が連続である。
- 学期集中 ... 科目 k_1 と科目 k_2 の割当時間帯の学期が等しい。

2科目間制約には次のものがある。

$$C_R = \{c_4, c_5, c_6, c_7, c_8\}$$

2科目間制約 c_4, c_5 のそれぞれの制約充足判定の内容は次のとおりである (c_6, c_7, c_8 については付録に示した)。

- c_4 : 科目 k_1, k_2 の割当が、表5の右欄の関係を満足していなければ、制約違反。
- c_5 : 科目 k_1, k_2 の関係が学期集中であるとき、これらの関係が時限連続でないならば、制約違反。

(3) 全体制約 C_E

これは、時間割表全体についての制約である。全体制約の手続きは、制約によって異なる。

$$C_E = \{c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}\}$$

全体制約 c_9, c_{11} のそれぞれの制約充足判定の内容は次のとおりである ($c_{10}, c_{12}, c_{13}, c_{14}$ については付録に示した)。

c_9 : 任意の科目 i_1, i_2 ($i_1 \neq i_2$) に対して、 $teacher(i_1) = teacher(i_2)$, かつ、 $slot(i_1) = slot(i_2)$ のとき制約違反。

c_{11} : すべての科目 i ($i = 1, \dots, M$) に対して、 $slot(i)$ が等しいものの個数が3個以上ならば制約違反。

3.3 制約違反点数

2.2節で示した制約の重要度を実現するために、それぞれの制約に制約違反点数を導入する。

ある科目割当が制約に違反しているとき、その制約の違反点数をその科目に加算する。これを次のように定義する。

$$conf(i, c_k) = \begin{cases} w_k \dots & \text{科目 } i \text{ の割当が} \\ & \text{制約 } c_k \text{ に違反} \\ 0 \dots & \text{その他} \end{cases}$$

科目 i の違反点数:

科目 i の違反点数は、その科目が違反しているすべての制約の違反点数の合計である。

$$W_i = \sum_{k=1}^P conf(i, c_k)$$

総合違反点数:

総合違反点数は、編成中の時間割の制約違反の程度を表す。この総合違反点数を最小化することが、本システムの目的である。

$$W_{total} = \sum_{i=1}^M W_i$$

制約の違反点数 w_k の決定方針:

- 1) 編成上必ず満たさなければならない制約と満たすことが望ましい制約に分類する。
- 2) 制約の違反点数は1から10までの範囲の点数とし、必ず満たさなければならない制約の違反点数は10点を与える。
- 3) 満たすことが望ましい制約の違反点数は下記のよう定める。
 - 3-1) 表3の分類ごとに、それらの中で重要度による優先順位を作成する。ここで、優先順位が一意に決定できない場合は、その制約自体があいまいであると判断し、表3から除外する(この制約条件は対話によって時間割に反映させるものとす

る)

具体例)

2科目間制約 $\{c_5, c_6, c_7, c_8\}$:

$$w_5 < w_6$$

全体制約 $\{c_{12}, c_{13}, c_{14}\}$:

$$w_{12} < others$$

3-2) 表3の各分類間の優先順位を比較して、全体の優先順位を定め、具体的な違反点数を決定する。

具体例)

まず、各分類間の優先順位を決定すると、 $w_5 < w_{13}$, $w_5 < w_{14}$, $w_7 < w_{13}$, $w_7 < w_{14}$ となる。次に、 w_{12} が最小なので $w_{12} = 1$ とする。以上より、 $w_5 = 2$, $w_{13} = 3$, $w_{14} = 3$, $w_7 = 2$, ... を得る。

4) 3-1) と 3-2) で考慮する重要度は、基本的にカリキュラム委員会の編成方針を反映させたものである。さらに、編成者の経験による判断も用いられる。

3.4 制約違反点数最小化戦略

本システムでは、探索手法として制約違反最少化戦略による山登り法⁷⁾を次のように適用した。

制約違反点数最小化戦略:

評価値を総合違反点数の逆数として、総合違反点数が小さくなる方向に探索を進める。

制約違反最少化戦略は次の2ステップを行う⁷⁾。

Step1 制約違反をしている変数を1つ選択する。

Step2 選択された変数に、制約違反が最少になるような値に変更する。

これを本手法では、次のように対応させた。

Step1':

W_i が最大となる科目 i ($i = 1, \dots, M$) を R とする。複数ある場合はユーザが選択する。新規割当の場合は、本Stepは必要なし。

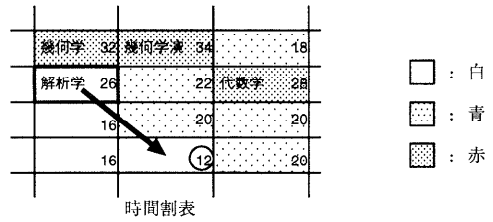
Step2':

時間割編成問題では、割当の変更は次の4通りが考えられる。変更方法として、この4つの方法から選択し、違反点数が最小になるように変更することで、制約違反を改善する。

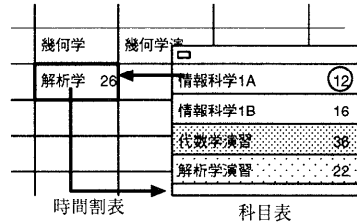
(a) 移動

科目 R を移動させることで、 W_{total} が最小となるような時間帯に科目 R を移動する。

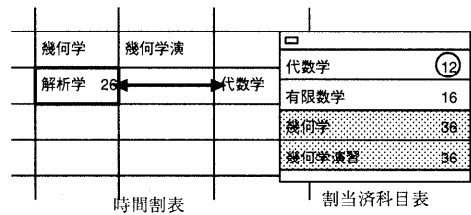
移動は、時間割表の科目を他の時間帯に移動させることである (図2(a) 参照)。



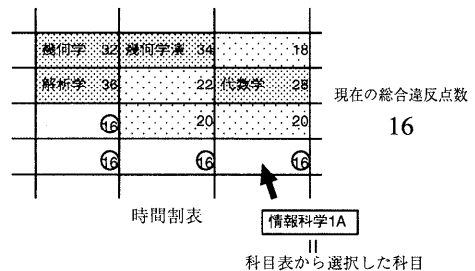
(a) 変更方法が移動の場合



(b) 変更方法が別科目割当の場合



(c) 変更方法が交換の場合



(d) 変更方法が新規割当の場合

図2 時間割の変更方法

Fig. 2 Operations of refining the timetable.

(b) 別科目割当

科目 R を割当解除し、その時間帯に W_{total} が最小となるような科目を割り当てる。

割当解除とは、時間割表から科目をはずし、科目表に戻すことである。すなわち、これは科目 R の割当時間帯 $slot(R)$ を未定義に戻す (図2(b) 参照)。

(c) 交換

科目 R の割当との交換により、 W_{total} が最

小となるような割当と交換する。

交換は、時間割表にある科目どうして行う (図 2(c) 参照)。

(d) 新規割当

まだ割り当てられていない科目を、 W_{total} が最小となるような時間帯に割り当てる。新しく割り当てることから、通常、 W_{total} が現状のままである、すなわち制約違反しない割当の候補が最小となる。

新規割当は、科目表から時間割表に科目を割り当てる動作である。(図 2(d) 参照)。

図 2(a)~(c) では、太枠で囲まれた解析学が変更する割当の科目 R である。また、図 2(a)~(d) では、各候補の右側にその候補に変更したときの総合違反点数が表示されている。図 2(a), (d) では候補は時間帯、図 2(b), (c) では科目である。ここで、総合違反点数が最小となる時間帯または科目の候補に丸印が付けられており、この候補をシステムがユーザに提案する。画面上では、各候補は改善の程度によって、10 点以上改善のとき白、1 点以上 10 点未満改善のとき青、改善されないとき赤で色分け表示されている。

3.5 処理手順

本システムの処理手順を図 3 に PAD で示す。以下図 3 に従って説明する。

- 1) 編成したい時間割の基となる過去の時間割などをファイルから読み込み、割当を行う。
- 2) 1) で割り当てられなかった科目を、ユーザが自由に割り当てる。
- 3) システムは、まず時間割表から制約違反の程度が最も大きい割当の科目を変更することを提案する。ユーザは、その提案された科目を選択

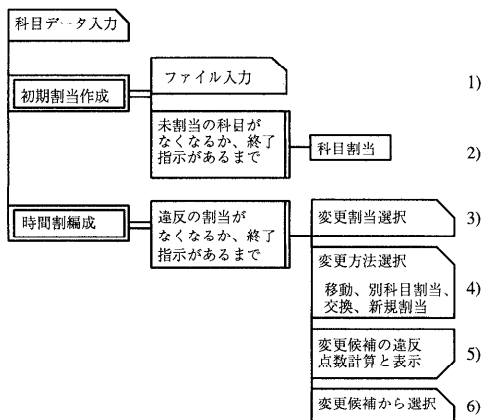


図 3 処理手順 (PAD)

Fig. 3 The general flow of the system.

するか、またはユーザが変更したい科目を指示する。

- 4) ユーザが 3) で選択した科目をどのように変更するのかを、3.4 節で示した (a) 移動, (b) 別科目割当, (c) 交換, (d) 新規割当から選択する。
- 5) システムは、4) で選択した方法によって変更してできるそれぞれの候補の総合違反点数を計算し、点数とともに科目や時間帯を色分けして表示する。同時に最小の総合違反点数の候補を提案する。
- 6) ユーザは、5) でシステムが提案した候補を選択するか、5) で得られる候補の違反点数の変化情報を基に、候補を選択する。

4. 実行例

本システムの実行例を図 4, 図 5, 図 6 に示す。

図 4 は、3.5 節の処理手順 4) で変更方法「移動」を選んだときの実行例である。変更する割当は、3) で違反点数が最大 ($\max(W_i)$) の「確率統計」が選ばれている。ここで、システムは、「確率統計」をそれぞれの時間帯に変更したときの総合違反点数を計算し、各候補 (ここでは時間帯) の点数として右側に表示し、同時に、時間帯を改善の程度 (10 点以上改善が白、1 点以上 10 点未満改善が青、改善されないが赤) によって色分け表示をする。さらに、これらの候補の中から、最小の総合違反点数の候補を強調表示し (図 4 の下の木曜 2 限の時間帯)、ユーザにそこへ移動することを提案する。

図 5 は、3.5 節の処理手順 4) で変更方法を「交換」とした場合の実行例である。このときの諸条件は上述の図 4 の場合と同様である。変更方法が交換の場合は、すでに時間割表に割り当てられている科目と変更したい「確率統計」との割当を交換したときの時間割

月	1	総合科目 2 127	総合科目 2 127	非数値処理 130	非数値処理	
	2	総合科目 2 127	総合科目 2 127	確率統計 160	数値計算法	
	3	総合科目 2 127	総合科目 2 127	確率統計 129	数値計算法	
火	1	計算機原論 127	計算機原論 127	情工基礎実験 130	情工基礎実験	
	2	情科 2 B 127	情科 2 B 127	情工基礎実験 130	情工基礎実験	
	3	情科 2 C 127	情科 2 C 127	情工基礎実験 130	情工基礎実験	
水	移動改善: 現在の総合違反点数は 129 変更科目を最少の総合違反点数となる時間帯に移動させますか? <input type="button" value="はい"/> <input type="button" value="いいえ"/>					
木	2	116	114	情工基礎実験 130	情工基礎実験	
	3	電子工学 127	116	情工基礎実験 130	情工基礎実験	

図 4 移動改善の表示画面 (部分) の例

Fig. 4 A displayed timetable being refined by 'move' operations.

1	総合科目 2	127	総合科目 2	127	非数値処理	130	非数値処理
2	総合科目 2	127	総合科目 2	127	確率統計	160	数値計算法
3	総合科目 2	127	総合科目 2	127	確率統計	129	数値計算法
1	計算機原論	127	計算機原論	127	情工基礎実験	130	情工基礎実験

交換改善: 現在の総合違反点数は 129
以下の割当済み科目表から交換したい科目を選択して下さい

科目ID	科目名	担当教官	学年	グループ	分野	単	点数
301201	極限数学	池田	2	専門基礎	数学	不可	144
301496	応用数学	城名	2	専門基礎	数学	不可	148
302401	電気回路	斎藤	2	通常2年	物理	可	126
302601	電磁気学	工藤	2	通常2年	物理	可	110
302701	電子工学		2	通常2年	物理	可	108

図5 交換改善の表示画面(部分)の例

Fig. 5 A displayed timetable being refined by 'exchange' operations.

1	総合科目 2	0	総合科目 2	0	非数値処理	0	非数値処理
2	総合科目 2	0	総合科目 2	0	電気回路	0	数値計算法
3	総合科目 2	0	総合科目 2	0		0	数値計算法
1	計算機原論	0	計算機原論	0	情工基礎実験	0	情工基礎実験
2	情科2B	0	情科2B	0	情工基礎実験	0	情工基礎実験
3	情科2C	0	情科2C	0	情工基礎実験	0	情工基礎実験
1	有限数学	0	有限数学	0	体育2	0	
2	電磁気学	0	電磁気学	0	体育2	0	
3	システム制御	9		0	体育2	0	
1	情科2A	4		0	情工基礎実験	0	情工基礎実験

図6 本システムで得られた時間割の例(部分)

Fig. 6 An example of optimized results.

表の総合違反点数を計算する。この場合は、候補が割当済みの科目であるため、交換改善のための科目表を表示する。この科目表の最後の列には、総合違反点数を表示する。さらに、改善の程度による色分けも行う。また、移動の場合と同様に、最小の総合違反点数になる科目(電子工学)を強調表示し、その科目と交換すれば違反が改善されることを示す。

図6は、過去の時間割を初期値としてシステムに入力し、本システムで得られた時間割である。入力した過去の時間割は、担当教官の変更や科目の統廃合などにより、所々現在のカリキュラムに違反している。当初、総合違反点数は129点であったが、本システムにより編成後は28点にまで減少した。このように、制約違反は少なくなり、現実に即した時間割が編成された。このとき、編成は反復改善7ステップで終了し、1ステップに3回問合せ(変更科目選択, 変更方法選択, 候補選択)があるために合計21回の問合せがあった。なお、図6の科目の右側の数値は、その科目の違反点数である。

上記の結果に対し、実際に人手によって編成され用いられている時間割では、本システムで計算した総合違反点数は36点であった。結果として、本システムが若干点数が減少したが、現実に必要とされる時間割

としては、ほぼ同じものである。また、人手の編成には数時間を要したが、本システムでは、専門の知識を持たない人でも、数十分で同様の時間割の編成が可能である。

5. おわりに

本論文では、従来の自動探索では対処しきれない実際の大学の時間割編成問題を効率良く解くために、対話操作と探索を組み合わせたシステムを開発し、その手法について述べた。

本システムは次の特徴がある。

- (1) 制約の定式化が困難である時間割編成問題において、システムとユーザが協調して探索することによって、システムが扱うことの難しい制約を編成に反映させることができる。
- (2) 制約違反点数を最小化する戦略による山登り法を用いることによって、時間割編成を効率的に行うことができる。

今後は、制約の表現をより一般化して、レイアウト問題などの他の制約充足問題に本手法を適用する予定である。

参考文献

- 1) 西原清一: 整合ラベリング問題と応用, 情報処理, Vol.31, No.4, pp.500-507 (1990).
- 2) Cooper, T. and Kingston, J.: The Solution of Real Instances of the Timetabling Problem, *The Computer Journal*, Vol.36, No.7, pp.645-653 (1993).
- 3) Yoshikawa, M., Kaneko, K., Nomura, Y. and Watanabe, M.: A Constraint-based Approach to High-school Timetabling Problems: A Case Study, *AAAI-94*, pp.1111-1116 (1994).
- 4) 西森雄一, 内野寛治, 狩野均, 西原清一: 制約違反最小化戦略による対話型時間割編成システム, 情報処理学会研究報告, 95-AI-101, Vol.95, No.86, pp.25-32 (1995).
- 5) Burke, E., Elliman, D. and Weare, R.: A University Timetabling System Based on Graph Colouring and Constraint Manipulation, *Journal of Research on Computing in Education*, Vol.27, No.1, pp.1-18 (1994).
- 6) Burke, E., Elliman, D. and Weare, R.: A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems, *ICGA-95*, pp.605-610 (1995).
- 7) Minton, S., Johnston, M., Philips, A. and Laird, P.: Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems., *Artificial Intelligence*,

Vol.58, pp.161-205 (1992).

- 8) Frederick Hayes-Roth ほか：エキスパート・システム，産業図書 (1985).

付 録

(1) 科目単位制約 C_U

c_2 : 科目 i の担当教官 $teacher(i)$ が表 6 に入っているとき，その担当不可能時間帯に科目 i の割当時間帯が入っていれば，制約違反。

c_3 : 科目 i の担当教官 $teacher(i)$ が表 7 に入っているとき，その優先担当時間帯に科目 i の割当時間帯が入っていなければ，制約違反。

(2) 2 科目間制約 C_R

c_6 : 科目 k_1, k_2 の関係が学期集中であるとき，かつこれらの関係が時限連続でないとき，割当時間帯が同じ曜日ならば，制約違反。

c_7 : 科目 k_1, k_2 の関係が学期連続であるとき，これらの割当時間帯が同曜日同時限でないならば，制約違反。

c_8 : 科目 k_1, k_2 の関係が時限連続であるとき，これらの割当時間帯が昼休みを挟むならば，制約違反。

(3) 全体制約 C_E

c_{10} : 任意の科目 $i_1, i_2 (i_1 \neq i_2)$ に対して， $course(i_1) = course(i_2)$ ，かつ $slot(i_1) = slot(i_2)$ ，かつ $grade(i_1) = grade(i_2)$ のとき制約違反。

c_{12} : すべての科目 $i (i = 1, \dots, M)$ に対して， $slot(i)$ が等しいものの個数が 2 個ならば

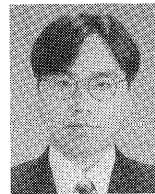
制約違反。

c_{13} : 任意の科目 $i_1, i_2 (i_1 \neq i_2)$ に対して， $course(i_1) = course(i_2) = \text{必修}$ ，かつ， $grade(i_1) \neq grade(i_2)$ ，かつ， $slot(i_1) = slot(i_2)$ のとき制約違反。

c_{14} : すべての科目 $i (i = 1, \dots, M)$ に対して， $group(i)$ が等しく， $slot(i)$ の学期・曜日が等しいものの個数が 4 個以上ならば制約違反。

(平成 8 年 8 月 5 日受付)

(平成 9 年 4 月 3 日採録)



西森 雄一 (正会員)

1972 年生。1995 年筑波大学第三学群情報学類卒業。1997 年同大学大学院理工学研究科修士課程修了。現在，ソニー（株）勤務。在学中は，制約に基づく対話型問題解決方式の

研究に従事。



狩野 均 (正会員)

1954 年生。1978 年筑波大学第一学群自然科学類卒業。1980 年同大学大学院理工学研究科修士課程修了。同年，日立電線（株）入社。同社オプトロシステム研究所において人工知

能・神経回路の応用に関する研究に従事。1993 年より筑波大学電子情報工学系。現在，同助教授。制約に基づく問題解決，遺伝的アルゴリズムの研究に従事。工学博士。1992 年電気学会論文賞受賞。電気学会，人工知能学会，計測自動制御学会等各会員。



西原 清一 (正会員)

1968 年京都大学工学部数理工学科卒業。同年，同大学大型計算機センター助手。1975 年より筑波大学電子・情報工学系。現在，同教授。工学博士。1982~1983 年文部省在外

研究員として米国ヴァージニア工科大学。図形画像処理，グラフィックスと CAD，組合せ探索アルゴリズム，知識処理，制約充足問題，GA の研究に従事。著書に「データ構造」（オーム社）など。電子情報通信学会，人工知能学会，ACM，IEEE 各会員。

表 6 教官の担当不可能時間帯表

Table 6 The table of time-slots to be avoided per teacher.

教官名	担当不可能時間帯
井田	(1, *, *)
大出	(*, [月, 水], *)
斉藤	(*, *, [3,4,5])
⋮	⋮

表 7 教官の優先担当時間帯表

Table 7 The table of priority time-slots per teacher.

教官名	優先担当時間帯
池辺	(*, *, [3,4,5])
斉藤	(*, [月, 火, 水], *)
田中	([1,2], *, *)
⋮	⋮