学校時間割り自動編成

吉川　昌澄

1. はじめに

各高校では毎年春休みに次年度の毎週の時間割りを編成している．大規模な学校では，この作業は極めて困難であり，150人日もの労力を掛けているのが実情である．また，大学や予備校など，各種の学校では時間割り編成のために2～3人程度の要員を配置している例もある．このように，時間割りの編成は極めて困難な作業であり，その自動化が望まれている．

筆者は，これまでに，学校時間割り編成問題を制約充足問題（CSP：Constraint Satisfaction Problems）として定式化し，汎用的なアプローチで自動解決技術を開発してきた．現在では，中学校・高等学校・大学向けのソフトウェアパッケージとして広く利用されている．

一般に制約充足問題はNP完全であり万能の解法は存在しない．そこで高速の近似解法やユーザ編集機能が重要となる．また，現実世界の問題をいかに定式化し，不完全な解法を用いてシステム化するかというモデリングの問題が極めて重要となる．

本稿では，筆者らが開発した学校時間割り自動編成技術を概説するとともに，汎用的制約問題解決技術の重要性について概観し，学校時間割り自動編成システムと大学時間割り自動編成システムについて述べる．最後に，今後の課題と展望について考察する．

2. 学校時間割り編成問題の定式化

ここでは，まずモデルとなる制約充足問題について，続いて学校時間割り編成問題について述べる．

2.1 制約充足問題（CSP）

CSPは，有限個の変数 $v_1, \cdots, v_N$ と，各変数の取りうる値の有限集合である候補集合 $D_i=\{x^i_1, \cdots, x^i_{di}\}$ と，有限個の制約 $C_1, \cdots, C_M$ からなる．制約 C_j は，その制約に関与する変数の組 $\{v_{k1}, \cdots, v_{kMj}\}$ と，関与変数の値の組が満足すべき条件（例えば論理式）$C_j^{k1,\cdots,kMj}(v_{k1}, \cdots, v_{kMj})$ を持つ．ここで，1変数に関与する制約をユナリ（unary）制約，2変数の場合をバイナリ（binary）制約，一般にN変数間（N-ary）制約と呼ぶ．問題は，すべての制約 C_j を同時に充足するような各変数 v_i の値 x^i を対応する候補集合 D_i 中から選択する組み合わせ探索問題である[15, 16]．

CSPの例として三色問題がある．これは，日本の都道府県区分図のように，平面上の線により区切られた各領域を，隣接する領域が同色にならないように三色で塗り分ける問題である．各隣接領域の組の間で同色とならないという条件を制約（$C_1^{東京,\ 神奈川}(x, y) \equiv x \neq y$；$C_2^{東京,\ 千葉}(x, y) \equiv x \neq y$；…）とすれば，制約充足問題として定式化可能である．その他のトイプロブレムの例として，N-Queen問題[7]，SAT問題，各種グラフ問題，パズルなどがある．

2.2 学校時間割り編成問題

高校の時間割り編成問題（図1）は，1週間の授業や会議の予定を決定する問題である．クラス数30・先生70人程度の大きな学校では，述べ100～150人日という工数を掛けて時間割りを編成している．各授業に出席する先生や対象クラス，利用する特殊教室などの属性はあらかじめ与えられる．ここでは，各授業を開催する曜日時限がCSPの変数となり，1週間の時限｛月1，…，土4｝が候補集合となる．主な制約は

	1-1	..	3-10
月1	数I	..	古文
…	…	..	…
土4	現文	..	幾何

(a) クラス時間割り表

	田中	..	佐藤
月1	数I	..	古文
…	…	..	…
土4	幾何	..	現文

(b) 先生時間割り表

図1　高校時間割り編成問題のイメージ

よしかわ　まさずみ
ソフトバンク・イーシーホールディングス株式会社
〒103-0015 東京都中央区日本橋箱崎町24-1
(本研究は著者がNEC在籍中の成果)

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

次の通り．

C1　同じクラス・先生の授業が同じ時限に重ならない．

C2　選択授業など同時開催の授業や，芸術など連続開催の授業がある．

C3　特殊教室での同時開催授業数の制限．

C4　先生などの予定により不都合な時限がある．

C5　各クラスの同じ教科・科目の授業は異なる曜日にする．

C6　先生の1日，または，連続の授業数の制限．

C7　週2コマの授業はなるべく間を1日以上あける．

ここで問題となるのは，例えば，制約C7に「なるべく」とあるように，絶対とは言わないが充足が望ましいという制約があることである．現に実際の高校では一部の制約を違反した時間割りが運用されている．このように絶対制約と考慮制約を持つ問題を定式化する枠組みとして，CSPを若干修正した制約緩和問題(CRP: Constraint Relaxation Problem)を定義する．CRPは，CSPのモデルに加えて各制約が違反時の罰則となる違反点数を持つ．問題は，違反している制約の違反点数の合計が最小となるような割り付け(各変数とその値の組)を探索する組み合わせ最適化問題である．絶対制約と考慮制約が混在する場合，絶対制約には十分大きい違反点数を，考慮制約にはその重要度に応じた違反点数を与えることにより，現実問題の要求に即した定式化が可能となる．

3. 学校時間割り編成問題のモデリング

筆者らは，制約ベース計画シェルCOASTOOL (Constraint-based Assignment and Scheduling TOOL)[13, 22]を開発し，各種時間割り編成問題や生産計画問題に適用してきた．ここでは，COASTOOLを中心に，現実問題のモデリングについて述べる．

3.1 COASTOOL/Lispによるモデリング

COASTOOLは汎用的なスケジューリング問題解決を目指すシェルである．前章で述べたように，制約問題は現実の時間割り問題のモデルとして有効である．しかし，その一方で，制約問題はNP完全問題であり，無二万能の解法は存在しないという課題がある．そこで，COASTOOLは制約問題モデルに基づく宣言的問題記述言語と，ある範囲の問題に有効な汎用的解法を集めた割付ライブラリとから構成される(図2)．ユーザは現実問題を制約問題に定式化し，宣言的問題記述言語を用いてこれを記述し(図3)，その問題にふさわしい解法を選択することにより問題を解決できる．ここで重要な点は，問題自身とその解決方法が独立であることである．したがって，1つの問題に異なる解法を適用すること，また逆に，異なる問題に1つの解法を再利用することが可能である．また，適当な解法がない場合には，割付ライブラリの制約伝播等の機能を用いて，個別の解法を開発する．

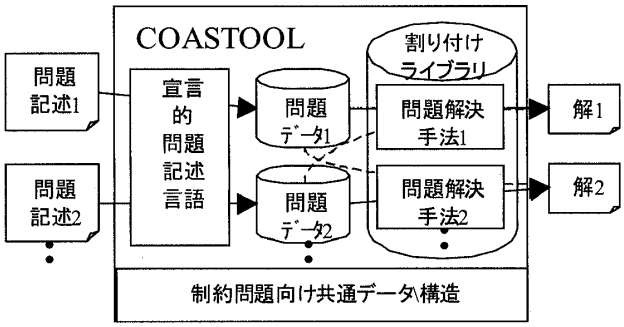

図2　COASTOOLアーキテクチャ

```
(def-set 授業 数I1-1a 数I1-1b ...)          ;;変数の集合
(def-set 時限 月1 月2... 土4)               ;;候補集合
(def-constraint 非常勤田中先生の都合        ;;制約
    :variable ((v 田中授業))                ;;関与変数
    :condition (not (is-a v 田中不在時間))  ;;条件
    :penalty 10)                            ;;違反点数
(def-problem 時間割り編成問題               ;;CRP
    :variables ((授業 時限))                ;;変数と候補集合
    :constraints (非常勤田中先生の都合 ...) ;;制約
    :minimize 制約違反点数の総和)           ;;目的関数
```

図3　COASTOOL/Lispによる問題記述例

筆者らは，COASTOOL/Lispを用いて久喜北陽高校の時間割り問題をモデル化した[22]．まず高校を訪問し3時間程度のインタビューでデータを入手，1.5人日の工数で問題(制約500行，データ1,000行)を記述した．COASTOOLが編成した時間割りを先生にファックスしてコメントをいただき，データの修正，制約の追加・削除・違反点数の調整などのフィードバックを10回程度繰り返した．これにより，わずか3人日程度で人手並に高品質の時間割りを自動編成することに成功，エキスパートシステムのエンジン部分の開発を完了した．この成功の要因としては，問題と解法が独立であり解法のヒアリングや実装が不要なこと，宣言的な制約問題のモデルによりモデル化が容易なこと，フィードバックが容易で緻密なプロトタイピング

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

が可能なことなどが考えられる．

3.2 COASTOOL/C++による制約の実装

スケジューリングシステムはデータベース（DB）上のデータをメモリ上の制約問題データ構造に展開し，解法アルゴリズムにより自動立案を行い，画面インタフェース（GUI）を通してユーザ編集を行う．一般に制約問題はNP完全のため，メモリ上での効率的なデータの実装方法，高速かつ高品質な解法アルゴリズムが重要となる．COASTOOL/Lispには，問題の記述能力，メモリ容量，速度性能，DBやGUIとの連携などの課題があり，COASTOOL/C++を開発した[23]．COASTOOL/C++は一種のオブジェクト指向フレームワークで，宣言的問題解決言語の代わりに，変数や制約など制約問題解決に必要なクラスライブラリを提供する．ユーザはクラスライブラリを用いて，メモリ上に制約問題データを構築し（図4），割付ライブラリの解法を適用する（または，ライブラリを用いて個別解法を開発する）．

COASTOOL/C++の最大の特長は，仮想制約と呼ぶ制約管理方法である．例えば，制約C1「同じ先生の授業が同じ時限に重ならない」を考える．今，ある先生が3つの授業x, y, zを持つとしよう．この制約には，2通りの実装方法が考えられる．1つは，この先生が受け持つ相異なる2つの授業の組み合わせすべてに対して，同じ時間にならないというバイナリ制約（$C_1^{x,y}(x, y) \equiv x \neq y$；$C_2^{y,z}(y, z) \equiv y \neq z$；$C_3^{z,x}(z, x) \equiv z \neq x$）を定義する方法である．もう1つは，この先生が受け持つすべての授業の組に対して，重なりがないという1つの制約（$C_0^{x,y,z}(x, y, z) \equiv (x \neq y) \wedge (y \neq z) \wedge (z \neq x)$）を定義する方法である．前者は制約の個数が組み合わせ的に爆発するという課題がある（一般に，先生がn個の授業を持つ場合は${}_nC_2$個）．後者は制約違反時にどの変数を直せばよいか不明瞭であるという課題がある．そこで，両者を組み合わせて，それぞれの利点を利用可能にしたのが仮想制約である．すなわち，常に後者の形の制約1個（C_0）を持ち，制約違反が発生した場合にはその部分に関して動的に前者の形の制約（C_1, C_2, C_3）を生成する．仮想制約により，制約の個数を制限し，かつ，制約違反時に変更すべき変数が明らかになる．

```
class TeacherAbsence :public Constraint {
    Teacher *_tea;
    TimeSlotVariable *_var;
  public:
    TeacherAbsence(Teacher *t, Lesson *l)
        :_tea(t), _var(l->timeslot()) {};
    virtual int doCheck() const {
      return(isTeacherAbsentAt(*_tea, _var->value()))
    };
    virtual float penalty() const { return 10.0; };
    …
}
```

図4 COASTOOL/C++による問題記述例

例として，制約C6「先生の1日授業数の制限」を考える（COASTOOL/Lispでは実装されていない）．ある先生の授業が15個あり，1日授業数の制限が4だとすると，違反する5個の変数の組み合わせは（先生1人につき）${}_{15}C_5 = 3003$通りある．この大量の制約も，1つの仮想制約と違反している制約の実体により置き換えることができる．COASTOOL/C++では，仮想制約を用いることにより，メモリ容量を1/10に削減，立案時間を1/20に短縮することに成功した[23]．仮想制約のメカニズムは複雑であり，その実現方法は決して簡易ではない．しかし，さまざまな制約をシステム上に実装し効率的に処理するためには，このような実現方法の工夫が必要である．

4. 学校時間割り編成問題の解法

主なCSPの解法には，無矛盾性（consistency）手法，バックトラック法，局所探索手法がある[15, 16]．

学校時間割り編成問題において，1週間の時限数34，クラス数30の高校では，授業の全駒数は約1,000弱となる（選択授業など複数クラスにまたがる授業がある）．したがって変数は約1,000個の授業，候補集合は34の時限となり，その組み合わせの数は$34^{1,000} \fallingdotseq 10^{1,531}$となる．このような膨大な探索空間において，無矛盾性手法，バックトラック法（あるいは分枝限定法）は計算量爆発の問題がある．また，制約緩和の可能性があるため，枝刈りによる探索空間の限定は困難である．

そこで近年注目されているのが局所探索法である．局所探索法は，まず，乱数などを用いて初期割付を作成する．通常，初期割付は何らかの制約を違反している．次に，制約違反が小さくなるように（または目的関数を改良するように）繰り返し改良操作を施し，解または近似解を求める．解発見の完備性がない反面，制限時間内に何らかの近似解を得られる点，制約緩和が可能な点などが時間割りへの応用に適している．

改良操作の内容と繰り返しの制御方法によりさまざ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

まな方法がある．山登り法は改悪となる改良操作を行わない手法である．MCHC(min-conflicts hill-climbing) 法[7]は，制約を違反する変数をランダムに選び，その変数の値を違反制約の個数（または違反点数）が最小になる値（同点の場合はランダム）に変更する．MCHC法は，最適性は低いが大規模問題の高速解決に適している．

SA(Simulated Annealing) 法は，金属の焼きなましを模倣したランダムに改悪する局所探索法である．制御変数に温度を設け，高温から徐々に冷やしていく．高温時ほど高い確率で改悪を受け入れる．ゆっくり冷やせば初期割付によらず最適解に到達するが，かなりの計算時間を要する．速度性能よりも最適性を追及する問題に適している．

筆者らは高校時間割り編成にバックトラック法を適用したところ，数日計算機を走らせ続けても最初の解には到達しなかった．MCHC法は高速だったが先生の手作業が2/3程度完了した時点と同等の品質であった．SA法は品質は少し良いが計算時間が長かった．

そこで，高品質の初期割付法であるRFLG(Really Full Lookahead Greedy) 法[22]を開発しMCHC法と組み合わせたところ，1～2分で先生の手作業の90%以上の品質の時間割りを自動編成することに成功した（図5）．

RFLG法は，制約伝播により候補集合の絞込みを行い，残った候補を順次変数に割り付ける．割付の結果は，再度，制約伝播で残りの変数へ伝播される．候補数の少ない変数と他の変数への影響の小さい値を優先している．途中，候補がなくなった変数は後回しにして，違反点数が最小となる値を順次割り付けるGreedy初期割付法により割付を行う．制約伝播を用いることにより，MCHC法の結果よりも優れた初期割付を作成できた（図5）．

ここで用いている制約伝播はアーク無矛盾性（AC: arc consistency）手法[6]である．これは，個々の制約を考えて，充足の可能性のない値を候補集合から取り除く処理である．例えば，制約 $x<y$ について，候補集合 $D_x=D_y=\{1,2,3\}$ を絞り込めば $D_x=\{1,2\}$, $D_y=\{2,3\}$ となる．いずれの制約伝播も，何らかの変化が生じた場合には，最終的に変化がなくなるまで伝播を繰り返す．

また，その後，RFLGを改良し，局所最適解脱出手法を開発した．RFLGの改良は候補がなくなった変数を後回しにせず，その時点で繰り返し改良を適用する段階的立案手法である[4]．局所最適解脱出手法は，同じクラスの授業が同じ時限に重ならない（制約C1）を利用して，同じクラスの2つの授業を玉突きのように動かすことにより，局所最適解から脱出する時間割り専用の方法である[2]．これらの改良により，違反点数をさらに30～97%削減することに成功し，より制約の厳しい高校に対しても，先生の人手の時間割りに遜色のない品質を得ることが可能となった[5]．

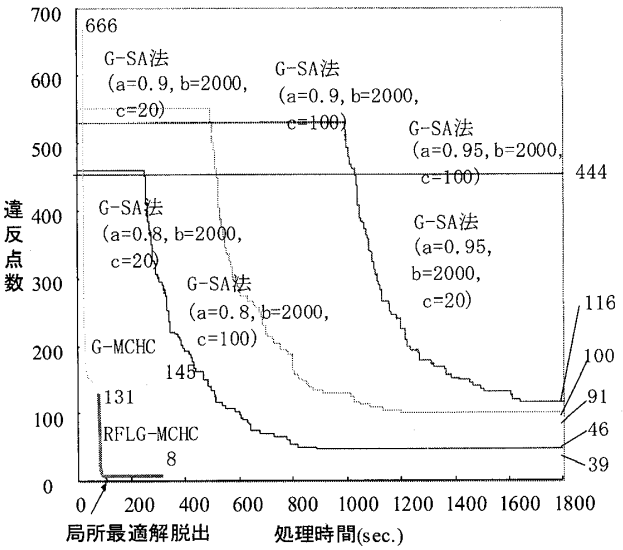

図5　高校時間割り編成の実験結果：GはGreedy初期割り付け．SAで α は冷却比率，β は同温での繰返し回数，γ は初期温度，プロットは初期割り付け修了以降のみ

5. AI学校時間割り自動編成システム「スクールマジック」

図6に，AI学校時間割り自動編成システム「スク

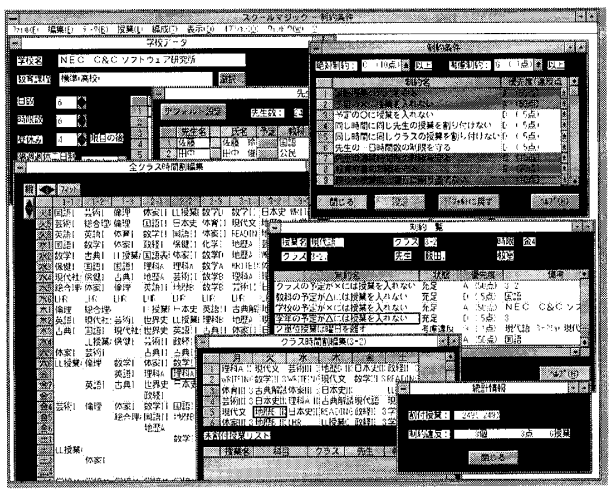

図6　スクールマジック画面イメージ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

ールマジック」の画面イメージを示す．スクールマジックは，各種データ編集，制約違反点数の設定，各種時間割り表示，制約違反状況や推薦移動先の提示など，約40種類の画面を備えるパッケージソフトウェアである．

スケジューリング問題における違反点数や目的関数を正確に定めることは難しい．また，厳密な定式化ができたとしても，解法の能力が不十分な場合も多い．そこで，GUI（図6）を通したユーザ編集機能が重要となる．ここでは主導権混在型（mixed-initiative）問題解決法[23]（図7）が有効である．例えば，自動割付の結果をユーザが確認し一部をフィックス，一部を修正し，その結果を初期割付とみなして繰り返し改良を適用する．このように，システムの自動機能とユーザの手動機能を自由に行き来する協調的な問題解決が可能である．局所探索法は，再計画同様，自動修正にも有効である．

情報提示機能としては，制約の違反状況の表示，修正を施した場合の影響を示すwhat-if解析，最適な修正方法を示す推薦機能，割付不可能な理由を解析する説明機能などがある．これら機能を実装する上で，システムが制約問題を宣言的なデータとして保持することが有効である．また，システム側とユーザ側が同一の問題データを対象とすることが重要となる．例えば，ユーザが制約や目的関数を修正した場合，システムの自動機能が機敏に対応可能なことが望ましい．

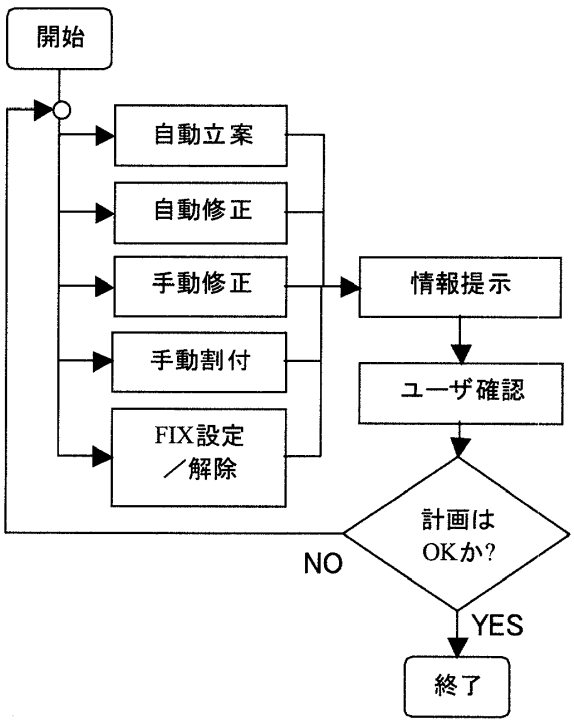

図7　主導権混在型問題解決のフロー

結局，スクールマジックにより，従来150人日を要して時間割りを編成していた高校でも，データ入力から開始して2人日程度で時間割りを完成することが可能となった．スクールマジックは，現在，100を越える中学・高校・大学で利用されている[11]．

6．大学時間割り自動編成システム

筆者らは，COASTOOL/C++を用いて，大学版スクールマジックを開発，早稲田大学の時間割り編成に適用し，実用化に成功した．大学時間割り編成問題は，教員があらかじめ決められた各講義や会議を，さまざまな制約を考慮しながら，より多くの制約を満足するように1週間のうちの1つの時限に割り付けると同時に，講義を開催する教室を割り付ける問題である．大学時間割り編成問題は高校時間割り編成問題と類似であるが，下記の点で異なっている．

① 高校時間割りに比べて問題規模が大きい．
② 時限だけでなく教室の割付が必要．
③ 前期のみ/後期のみ/通年の講義が混在し前期と後期の時間割りを同時に編成する．
④ 講義は選択性のため基本的にクラスはない．しかし，同一専修コースの必須科目など，重ねてはいけない講義がある．
⑤ 先生の希望教室・時限や昨年度の時間割りの割付情報を元に類似の割付を優先する．

筆者らは，COASOOL/C++上に大学時間割り編成問題を定義し，高校時間割りとまったく同一の割付ライブラリをそのまま適用し，実用化に成功した．すなわち，10分程度の計算時間で，人手並みの高品質な時間割りの自動立案に成功した．これは，COAS-

初期割付法	違反点数	時間(秒)	最良の回数
Greedy	399.2	193.7	0
RFLG	395.1	3349.2	0
段階RFLG	0.9	704.0	7

(a) 一部の初期割付情報（前年度時間割り）を利用

初期割付法	違反点数	時間(秒)	最良の回数
Greedy	96.9	178.1	0
RFLG	8.5	678.4	6
段階RFLG	26.5	837.6	6

(b) 初期割付情報なしで白紙から自動編成

図8　大学時間割り編成での実験結果

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

TOOL/C++アーキテクチャの汎用性が有効であることを示している．解法の点では類似の特性を示したが，昨年度割付情報利用時は，特に，段階的立案手法が有効であった[5]（図8）．これは，段階的立案手法が昨年度割付の問題点を早期に解消してから初期割付を進めるためである．

7. 今後の課題と展望

上記のように，筆者らはCOASTOOLによる汎用的制約問題解決により，逐次改良手法を適用することにより，中学・高校・大学向け時間割り自動編成システム「スクールマジック」を開発し，100以上の学校で利用されるに至っている．

まず重要な点は，大規模複雑な現実の時間割り問題に対し，制約伝播を用いた高品質初期割付と高速の繰り返し改良手法の組み合わせによる解法が，極めて高い有効性を示したことである．従来の汎用的解法技術に比べて，飛躍的な高速性と高品質性を同時に満足し，実用的にも人手とほぼ同等の時間割りの自動編成を可能とした．

次に重要な点は，COASTOOLにおいて問題のモデリングと解法とを分離したことである．これにより，微妙で複雑な現実問題の精緻な定式化が可能となった．また，同一の割付プログラムが中学・高校・大学向けそれぞれのスクールマジック製品において利用されるに至った．

しかし，特に実用に耐えるCOASTOOL/C++での制約や制約問題の記述には，研究者レベルの専門的能力が要求される．このため，個々の問題に対して，膨大な開発努力が必要となるのが現状である．

そこで，筆者らはエンドユーザレベルでの制約問題記述を可能とするGUIツールとしてハイパーデータシートを開発し，時間割り問題の簡易記述に成功した[1, 18, 19, 24]．ハイパーデータシート（図9）は，リスト・二次元表など構造を持つデータシート間の演算と制約を基本とする新しい表計算モデルを持ち，スプレッドシートの簡易性とデータベースの頑強性や再利用性を併せもつ表計算ツールである．構造変換・フィルタ・セル値演算などデータシートの構造に基づく新しい表演算系と，表間演算処理の自動再計算機構および一時的な違反を許す柔軟かつ強力な制約管理を特徴とする．特に制約違反表と呼ぶRDB表形式のシートを定義することにより，SQLのクエリに相当する任意の制約条件を効率的に定義可能である．

ハイパーデータシートによりスクールマジックの制約問題を定義したところ，すべての制約条件の簡易記述に成功した．図10に「先生の1日授業数の制限」（制約C6）の記述例を示す．全制約条件の記述（図11）に要した記述量は，COASTOOL/C++で約1,000行であるのに対して，ハイパーデータシートでは約50のシート演算オペレータで記述可能である．また，WYSWYGインタフェースにより簡易に記述可能であり，SEレベルの能力を持たない一般エンドユーザでも制約問題の記述が可能となることが期待される．

今後，ハイパーデータシートとCOASTOOL/C++を有機的に結合することにより，汎用制約問題解決パッケージツールが実現可能となろう．これは，現在の数理計画パッケージと類似だが，記号処理による制約が可能であり，学校時間割り自動編成システム

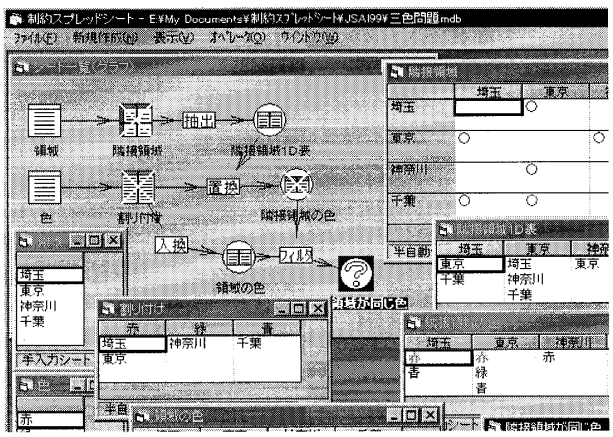

図9　ハイパーデータシート画面イメージ

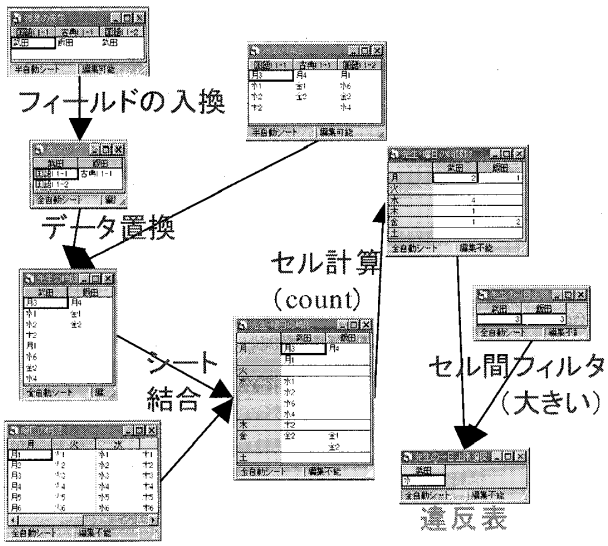

図10　ハイパーデータシートによる制約定義の例

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

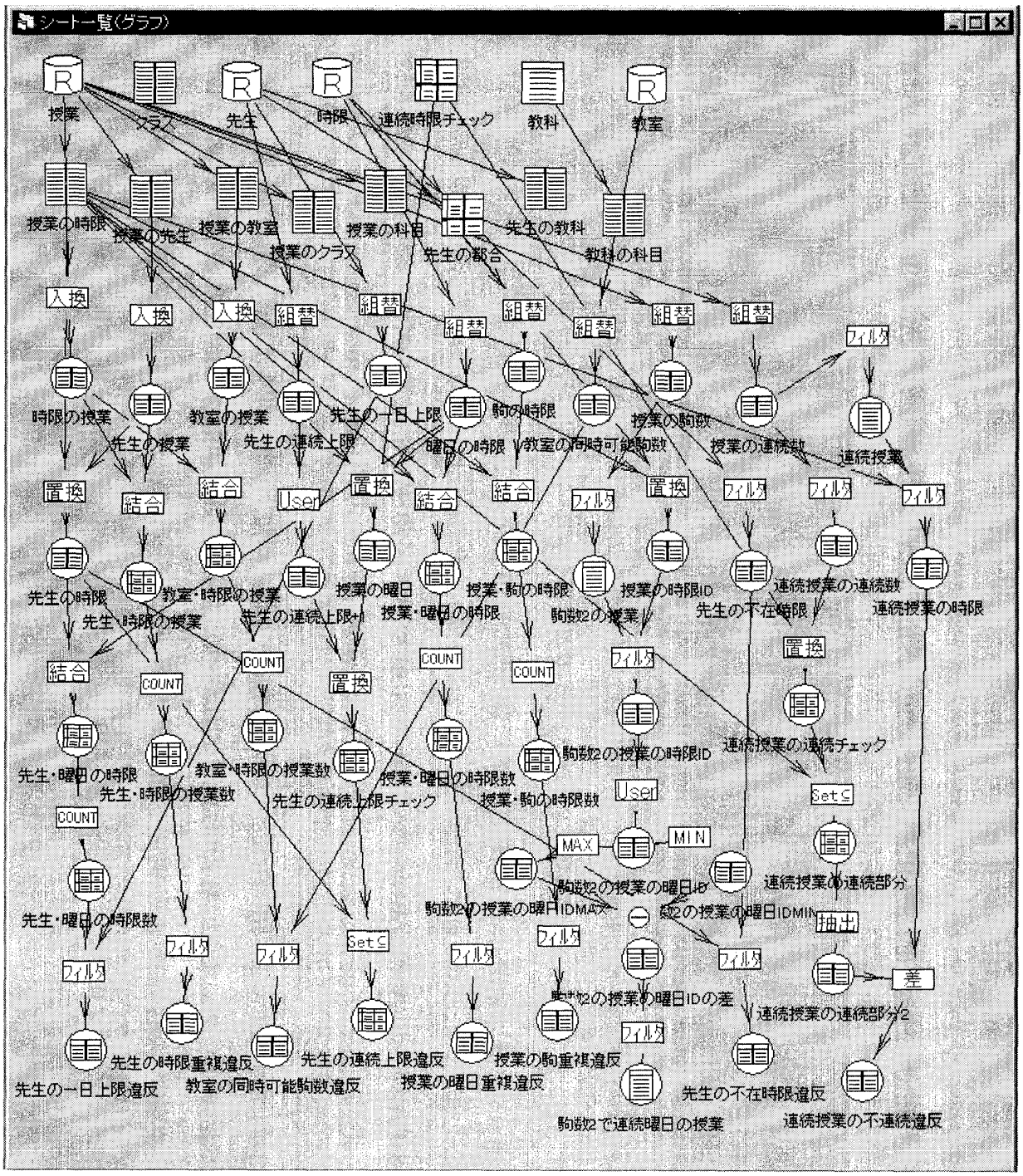

図11　ハイパーデータシートによる学校時間割り編成問題の全制約の記述

や要員業務割当システム等を簡易に構築可能なツールである．

8.　おわりに

本論文では，筆者らの時間割り自動編成システム開発を概観し，汎用的なアプローチによる制約問題解決への試みについて述べた．筆者の経験では，学校時間割りと類似のニッチな問題は多々存在する[10]．例えば，看護婦・郵便局・交通機関乗務員など要員配置型サービス業における勤務スケジュール，自動車学校や軍隊・研修センターなどにおける教育や講習のスケジュール，スポーツ大会の対戦などイベントスケジュール，コマーシャルの放映スケジュールなど，いずれも類似の問題であり枚挙に暇がない．これらサービスシステムのスケジューリングに対して，本稿の内容が参考になれば幸いである．

Garilは，“アルゴリズム設計にレシピはない．これは応用面では不運であり毎回いちから設計しなけれ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

ばならない．しかし，研究者は幸運であり研究テーマがなくなることはない．”と述べた．NP完全性に立ち向かえば，これは必然的な結論であるといえる．しかし，それは，上記のような多種多様の問題に対して，簡単に技術を適用することはできないという結論を示している．Garilの言葉を完全に否定することはできないであろうが，筆者らは高校と大学の時間割りに同一のアルゴリズムを適用し実用化に成功した．また，ハイパーデータシートの開発により，多種多様な問題を簡易にモデリング可能とした．今後，汎用制約問題解決パッケージツールの実現により，多様な問題が簡易に解決可能となればと願っている．

参考文献

[1] 安藤, 吉川：“制約データシート(2)：システムの試作と評価”, AI全大, 1999/6.

[2] 金子, 吉川, 山之内, 渡辺：“学校時間割自動編成システムにおける制約緩和手法の提案と評価”, AI全大, 1995/7.

[3] 金子, 吉川, 山之内, 渡辺：“学校時間割編成問題における段階的編成手法の提案と評価”, 情処全大, 1997/3.

[4] 金子, 吉川, 山之内：“大学時間割編成問題への制約緩和手法の適用と評価”, 情処全大, 1997/9.

[5] Kaneko, Yoshikawa, Nakakuki：“Improving a Heuristic Repair Method for Large School Timetabling Problems”, Int Conf on Principles & Practice of Constraint Programming (CP99), 1999/10.

[6] Machworth, “The Logic of Constraint Satisfaction”, Artificial Intelligence, 58：3-20, 1992.

[7] Minton, et. al., “Minimizing Conflicts：A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems”, Artificial Intelligence, 58：160-205, 1992.

[8] Minton, “Specification-By-Demonstration：The ViCCS Interface”, AAAI Spring Symposium Seriese, 1996.

[9] Minton, “Automatically Configuring Constraint Satisfaction Programs：A Case Study”, Constraints 1(1), 1996.

[10] The International Series of Conferences on the Practice and Theory of Automated Timetabling (http://www.asap.cs.nott.ac.uk/ASAP/ttg/patat-index.html)

[11] NECソフトウェア，スクールマジックホームページ (http://www.necsoft.com/soft/edusoft/smagic/)

[12] NECソフトウェア，スクールマジックホームページ (http://www.necsoft.com/soft/edusoft/smagicu/)

[13] 吉川, 野村, 渡辺：“制約ベース計画シェルCOASTの開発と評価”, AI全大, 1993/7.

[14] 吉川, 金子, 山之内, 渡辺：“学校時間割自動編成システムの開発と評価”, AI全大, 1995/7

[15] 吉川：“制約最適化技術のスケジューリング問題への応用”, AI学会誌, 1998/5.

[16] 吉川：“制約最適化技術のスケジューリング問題への応用”, COM・SCM・スケジューリング研究部会, 1998/9.

[17] 吉川：“学校時間割り自動編成システム”, 日本OR学会 シンポジウム, 1999/9.

[18] 吉川, 安藤：“制約データシートによる制約充足問題記述”, AI学会KBS研究会, 1999/6.

[19] 吉川, 安藤：“制約データシート(1)：シート間制約に基づく新しい表計算モデルの提案”, AI全大, 1999/6.

[20] 吉川：“学校時間割り自動編成技術の紹介”, NIME Newsletter 第20号, 2000/9. (http://www.nime.ac.jp/nnl/news_cont.html)

[21] Yoshikawa, Wada：“Constraint Satisfaction with a Multi-Dimensional Domain”, Int Conf on AI Planning Systems (AIPS-92), 1992/6.

[22] Yoshikawa, Kaneko, Nomura, Watanabe：“A Constraint-Based Approach to High-School Timetabling Problems：A Case Study”, Nat. Conf on Artif. Intel. (AAAI-94), 1994/8.

[23] Yoshikawa, Kaneko, Yamamouchi, Watanabe：“A Constraint-based High-School Scheduling System”, IEEE Expert, 1996/02.

[24] Yoshikawa：“Specifying Constraint Satisfaction Problems with HyperDataSheet”, Int Conf on Practice & Theory of Automated Timetabling (PATAT 2000), 2000/8.

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．