実践講座

スケジューリング問題の新解法

(7)：エキスパートシステム手法のスケジューリング問題への応用

渡辺　正信

1．はじめに

エキスパートシステム(以降ESと略す)は，1970年代後半に米国大学で提唱された．その後，1980年代前半に産業化が試みられ，1980年代後半には，実用問題への利用が活発化した．1990年代に入り実用化され実際に日常的に利用されているESの多くはスケジューリングESである．特に，乗務員スケジューリングES[1]とか生産計画ES[2][3][4]は，企業における戦略的業務の効率化を目指して構築され，その有効性が確認されている．これらは，オペレーションズリサーチ(OR)に代表される従来手法では実現困難であったが，ESアプローチにより始めて開発・運用に成功したものである．

本稿では，まず最初に，人工知能(以降AIと略す)の基本的考え方を紹介する．次に，AIの考え方をスケジューリング問題へ適用した代表としてスケジューリングESのイメージを示す．その後で，現状のスケジューリングESの利点と問題点について整理し，次世代スケジューリングESに対するニーズ分析と技術開発要件を確認し，次世代スケジューリングESの有力な候補として，制約ベース推論を応用したスケジューリングESを紹介する．

1.1　人工知能(AI)システムの基本的考え方

システム構築に対するAIの基本的考え方は，『システムの柔軟性と効率性をいかに両立させるか』にある．この観点から「おもちゃの問題」ではなく「現実の問題」を現実的に解決する(つまり，合理的な処理時間で妥当な品質の解を導出できる)システムとして，エキスパートシステムが提案された．

エキスパートシステムは，システム・アーキテクチャ・レベルで知識ベースと推論エンジンを分離するという工夫を行なって，以下のように柔軟性と効率性の両立を図っている．

(1) 柔軟性を高める工夫：

知識ベースと推論エンジンを分離して，各々の変更／修正を容易にしている．これにより，段階的に知識ベース内を充実させることが比較的容易となって，環境の変化(システムに対するユーザ要求機能の変化など)に追随しやすくなっている．つまり，段階的に問題解決能力を容易に高めることを可能としている．

(2) 効率性を高める工夫：

ストレートフォワード(単純で機械的，または，一般的)な探索手法を現実問題に適用すると，組み合わせ爆発が発生して合理的な処理時間で解を導出できなくなるのを回避するため，特定の対象問題をうまく解決する専門家のノウハウを知識ベースに蓄積して，それを適宜活用し，組み合わせ爆発が起こらないように現実的に解を導出する工夫をしている．

さらに，知識ベース内の各々の知識を推論エンジンが逐次的に解釈実行すると実行処理が遅くなるので，知識ベースを前もってコンパイルすることで実行処理を高速化する工夫も行なわれている．

なお，本稿の後半で述べる制約ベース推論によるスケジューリングESでは，柔軟性と効率性とをより高めて，かつ両立するために，上述の他に，さらに，以下の工夫を行なうことを提案している．

(1) 柔軟性を高める工夫：

解が満足すべき制約条件を並べる形式で，スケジューリング問題を宣言的に記述できる言語を提供することで，ユーザが解きたい問題を容易に指定できる．つまり，その問題をどのように探索して解を求めるかは，システムに任せることができる．この時，ユーザは，

わたなべ　まさのぶ　NEC　C&C研究所ソフトウェア研究部
〒216 川崎市宮前区宮崎4－1－1

問題の制約条件が変更されたとき，変更した制約条件を追加／修正していけばシステムを容易に変更できる．対象問題を解決している専門家が存在しなくてもよい．

(2) 効率性を高める工夫：

制約条件をベースに宣言的に記述した問題記述を解釈して，その問題に対する効率的処理手順（アルゴリズム）を自動的に導出する．たとえば，制約の集合で記述されたスケジューリング問題に対する解を求める手順を，複数のフェーズに分割して，各フェーズに有効な探索ヒューリスティックスを適宜選別して，各々を巧く組み合わせ（コンパイルし）効率的なアルゴリズムとする．

現状のスケジューリングESは知識ベースに専門家が持つHOWの知識を蓄積して活用することを主としているが，次世代スケジューリングESでは，ユーザが提示するWHATの形式の問題表現から最適なHOWの知識を自動的に生成する機構を保持する．

1.2 スケジューリング・エキスパートシステムとは

現在，企業のSISとして重要な位置を築きつつあるスケジューリングESの具体例を紹介し，そのイメージを確認する．

[1] 生産計画ES

工場における生産計画の効率化／自動化を狙うものである．

生産計画の全体フローは，1カ月から1年先に対する販売予測／所要計画に始まり，その数カ月分の所要を数日から1週間単位毎に複数のラインでいかに生産するかの負荷調整(山積み山崩し計画)，さらに，日々の実際の作業実施計画をたてる製造実施計画，最後に生産されたものをユーザ先へ配送する物流計画からなる．

[2] 交通ダイヤ編成／乗員割付ES

航空機，バス，トラック，船などの交通機関の発展に伴い，それらのダイヤも大規模複雑化してきている．それに伴い，ダイヤ編成とその運用に関するスケジューリング作業（どの便にどの乗員，機材，貨物を割り付けるか)も大変なものとなっており，ES化が進展している．

[3] 時間割編成ES

学校での時間割(教室／先生／クラスの割当)，レストランなどでのウエイター／ウエイトレスの作業時間割，看護婦の勤務時間割編成作業なども，学校，レストラン，病院などが大規模化したり，その機能が複雑多様化するに伴い，人手作業の限界となってきてES化が進展している．

1.3 現状のスケジューリングESの利点と問題点

現状のスケジューリングESは，スケジューリング専門家のスケジューリング・ノウハウを知識ベース化し，これを活用することでスケジューリングを自動化・半自動化することを大前提にしている．以下，その利点と問題点を整理する．

[1] 利　点

・[専門家ノウハウの利用]：

スケジューリング専門家のノウハウを利用して高い品質の計画立案結果を合理的な処理時間で提供できる．従来のOR手法に代表される数理計画手法では，数式による定式化を行ない，その解法は，ブラックボックス化されていて，人間の計画作業に近い方式を組み込むことが困難であった．このため，現実問題の場合は膨大な組み合わせ問題を解決することとなり，実用時間内に解を出すことが困難であった．

しかし，ES手法では，現実の問題を解決している人間の専門家が保持している計画ノウハウのシステム化が可能となり，実用時間で解を導出できる．また，計画に関する種々の制約条件を記述しやすい知識表現が可能となり，計画の導出手順が人間に分かりやすくなるため，その導出手順の変更をしやすい．さらに，従来計画専門家に占有されていた計画ノウハウを，知識ベース化して共有化できるようになる．

・[変更の柔軟性]：

計画立案の前提条件が変更された時にES（知識ベース，または，推論エンジン）が，その変更に追従できる．プロトタイピング方式のシステム構築を前提としているので，計画立案の前提条件が変更された時にも柔軟に対応できる．つまり，システムを運用しながら改良/拡張が可能である．

・[フレンドリーGUIインタフェース]：

計画立案に向けた条件設定や，ESが提示した解をユーザが容易にGUIを使って修正できる．

[2] 問題点

・[知識獲得が困難]：

専門家のノウハウを知識ベース化する知識獲得工数が，多大となる．

・[計画専門家が必要]：

複雑すぎて，人手による緻密な計画立案が実施されていない問題に対しては，お手上げである．つまり，

スケジューリングの専門家がいないので，現状ではES化を諦めざるを得ない．

この時，現場では，「経験と勘と度胸」によるその場逃れの実行指示で済ましている．

・［調整機構の実現が困難］：

リアルタイムの計画調整（一度立案された計画結果の変更・修正・調整）機構の実現が困難である．

2．次世代スケジューリングESに向けたニーズ分析と技術開発要件

現在運用されているか，または今後開発の要望が強いスケジューリングESに関する市場分析結果を図1に示す．図1で，縦軸はES導入前の状況，横軸は問題の難易度を，それぞれ表わしている．

ここで，ES導入前の状況（縦軸）というのは，スケジューリングES導入前に人手による計画立案が実施されていたかどうかということである．従来のスケジューリングESでは，専門家による計画立案が実施されていることが前提にあり，スケジューリングESは，その専門家の計画立案ノウハウを知識ベース化することで開発された．しかし，生産工程が複雑過ぎるとか，工数が足りない等の理由で緻密な計画立案が困難で，まともに実施されていない場合も多い．

一方，問題の難易度（横軸）では，人手によるスケジューリング工数を尺度にして，4～5人日以内で計画が立案できる程度のものを難易度が低いとし，100人日以上が必要なものを難易度が高いとしている．また，現状で，複雑過ぎて人手による計画立案を諦めているものも問題の難易度が高いとしている．

現在，スケジューリングESが開発され実運用されているものの多くは，問題の難易度は比較的低く，ES導入前には人手による計画立案が実施されていたものである．そこでは，計画立案ノウハウが比較的明確化されており，計画作業自身が実施されていた．また，難易度が高い計画立案作業も一部ES化されているが，そのES開発には非常に多くの工数と費用を費やしている．この場合，計画立案ノウハウは，部分的に明確であるが不明確な面も多い．しかし，組織の戦略的業務であるが故に，多大な工数をかけて計画立案が実施されていた．この作業を効率化できれば，たとえ膨大な開発投資でも十分に見合うとの判断からスケジューリングESが開発されたわけである．

ここで重要な視点は，スケジューリングES化に対するニーズの本質は何か，それに応えるためには，どのような技術開発が必須かを見極めることである．まず，ニーズの本質は，人手で計画立案が行なわれていようがいまいが，「目標を達成する，合理的なスケジュールを立案する作業を効率化（自動化）したい」ということにある．したがって，次世代スケジューリングESを展開していくための技術開発のポイントは，難易度が低く，すでに人手による計画立案が行なわれている分野と，難易度が高く人手による計画立案を諦めている分野の2つの両極のタイプに対して必要な技術を確立することにある．これら両極の問題を解決する技術を確立することにより，上記の本質的ニーズに応えることができる．

つまり，必要な技術開発要件は，以下の3点となる．

(1) ［業務モデル構築の容易化技術の開発］：

問題の難易度が低く，すでに人手による計画立案が実施されている分野に有効な技術である．

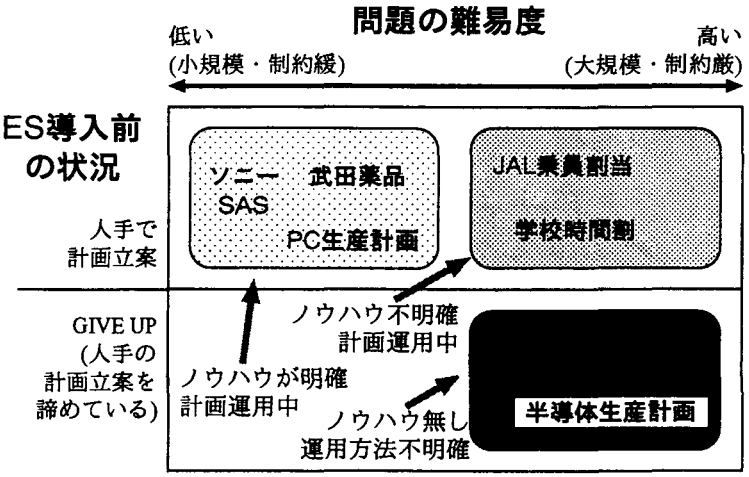

図1　スケジューリングES市場分析

(2)［スケジューリング問題（制約）記述言語と汎用最適化アルゴリズムの開発］：

現在，計画立案ノウハウが不明確な問題に対して，対象の計画問題を(宣言的に)記述できる枠組みと，その問題の（準）最適解を自動導出できる方式の確立．

(3)［スケジューリングES構築方法論の開発］：

上記の2つの技術を使って具体的スケジューリング問題を解決するシステムをいかに構築し運用していくかの方法論の確立．

以上(1)と(2)の技術を確立し，(3)のES開発方法論を整備することにより，開発コストが低く，計画立案環境の変化に追従できる次世代スケジューリングESの構築・運用が可能になる．

以下，筆者らが，上記の技術開発要件を前提に，次世代スケジューリングESを構築するために有効と考えている『制約ベース・スケジューリングシェルCOASTOOL』と，その適用例について述べる．

3．制約ベース・スケジューリングシェル COASTOOL［5］

3.1 設計思想

前述の技術開発要件のもとで，筆者らは，スケジューリングシステム構築シェルの研究開発を進めている［5］．その設計思想として以下の4点をあげる．ポイントは，制約処理基盤の上で，専門家立案ノウハウと汎用割付アルゴリズムの適宜活用と，ユーザ協調割付を実現するGUI部品の提供にある．

・［制約処理・管理基盤の提供］：

制約評価(制約の充足・満足度の評価)，制約伝播(値伝播，範囲伝播)，制約緩和・最適化（制約の重要度や満足度から目的関数を定義して制約緩和・最適化を図る）の各機能を提供する．

・［計画立案知識の活用］：

専門家立案ノウハウ・ベース割り付け機構と汎用割り付けアルゴリズム・ベース割り付け機構のいずれも実現できる環境を提供することで，初期割り付けや自動調整機能を問題にフィットさせて実現可能とする．

・［宣言的問題記述環境］：

業務モデル記述，問題定義，制約定義を宣言的に容易に実現できる言語と環境を提供する．

・［GUI部品の提供］：

スケジュール結果の表示や，ユーザによる割り付け指定・修正・調整を容易にするGUIベース協調割り付け環境を実現する．

3.2 基本アーキテクチャ

COASTOOLの基本アーキテクチャは，汎用の制約最適化問題解決システムである．制約最適化問題は変数の集合，各変数の候補値の集合，制約の集合，目的関数から構成される．問題は，目的関数を最小化するように各変数に候補集合内の値を割り付けることであ

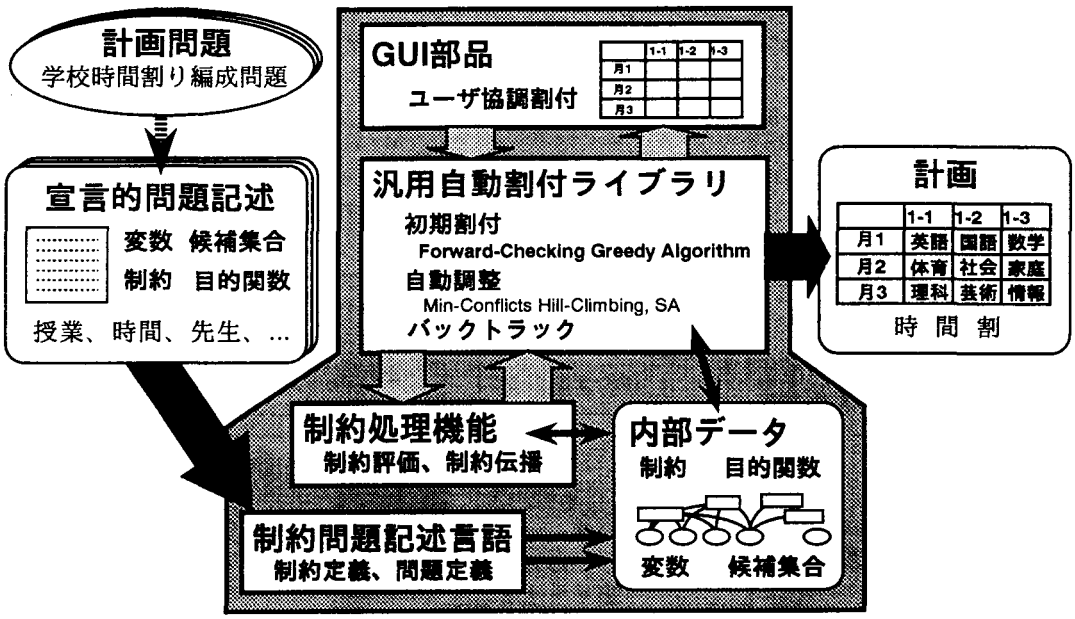

図2 COASTOOL基本アーキテクチャ

る．ここで，制約は目的関数により参照される．

図2にCOASTOOLの基本アーキテクチャを示す．ユーザは計画問題を制約最適化問題としてモデル化し，宣言的問題記述を作成する．COASTOOLの入力は宣言的問題記述，出力は計画立案結果である．宣言的問題記述は，制約問題記述言語により内部データに変換される．

汎用自動割付けライブラリは，初期割付け・自動調整・バックトラック等の自動割付けルーチン群を提供する．これらのルーチンは，制約評価・制約伝播（値伝播・候補集合伝播）など基本的な制約処理機能を用いて実現されている．

ユーザは問題の目的・性質に相応しい割付けルーチンを選択し利用することにより計画を自動立案する．GUI部品は立案結果を提示し，ユーザ協調割付け機能によりインタラクティブな修正を施し，計画を完成する．

次にCOASTOOLにおける制約定義の例を示す．

```
(define-constraint 先生の時間の一意性
 :level  5                          ;重要度
 :subjects ((:set 同先生授業 '同じ先生の授業)   ;制約定義の対象
  (:set 授業1 授業2 同先生授業))     →授業1/2は対象パラメータ
 :variables                         ;制約変数
  ((:name (時間1 授業1) (時間2 授業2)))   →時間1/2は変数パラメータ
 :condition (not (eql 時間1 時間2))  ;満足するための条件
 :evaluation (if (eql 時間1 時間2) 0 1))   ;満足度の計算方法
                                    (0～1)
                                    →1は完全に満足
```

これは時間割り編成問題における「先生の時間の一意性」と呼ばれる制約の定義であり，「同じ先生の授業のすべての組合せに対して，異なる時間を割り付けなければならない」ことを表わす．制約は，その制約を満足するための必要条件だけでなく，重要度や満足度を持ち，目的関数にもとづく制約緩和・最適化等に利用される．COASTOOLは種々の点で制約論理言語と類似するが，特に制約緩和・最適化が可能な点で異なっている．

COASTOOLでは計画問題を宣言的に記述し自動割付けルーチンを選択するだけでスケジューリングESの開発が可能であり，生産性が高い．また，スケジューリング環境の変化に伴い，制約の追加・修正・削除等を施すことにより容易なメンテナンスが可能である．

さらに，自動割付けルーチンとして最適化アルゴリズムを用いることにより，最適化・制約緩和・インタラクティブな計画修正・再計画等，高度な機能が提供可能となる．

3.3 汎用自動割り付けアルゴリズム

多くの汎用自動割り付けアルゴリズムは，まず1パスで割付けを行なう初期割付けと，その後，割付け結果を繰り返し修正・改良する自動調整の2つのフェーズからなる．

従来，初期割付けには，すでに割付け済みの変数と現在の割付け対象である変数との間で目的関数を最小化する値を順次割り付けるGreedy Algorithmが広く用いられている．

一方，近年，解の品質に対する初期割付けの重要性が理論的・実験的に明らかにされてきている[6]．

Greedy Algorithmは高速である反面，未割付けの変数との間の制約を考慮しないため初期割付け結果の品質が低いという問題がある．そこで，COASTOOLでは，制約伝播を用いて未割付けの変数との間の制約を前以てチェックすることにより非常に優れた初期割付けを作成するReally-Full-Lookahead-Greedy Algorithm（RFLGアルゴリズム）を提供している．

以下，図3に示すFCGアルゴリズムを詳述す

```
STEP0: 初期化
  1. V::=変数の集合，R::=空集合とする．
STEP1: Really-Full-Lookahead-Checking 無矛盾割付け
  2. Vが空集合になったら6へ進む．
  3. V中で制約を伝播し候補集合を絞り込む．
  4. 候補集合が空集合になった変数がなければ，
       V中の変数vを1つ選びVから削除し，
       vの候補値をランダムに選びvに割り付け，
       2へ戻る．
  5. 3で候補集合が空集合になった変数がある場合，
       その変数をVから削除しRに登録し，
       候補集合を制約伝播前（2の時の値）に戻し，
       2へ戻る．
STEP2: Greedy-Algorithm
  6. R中の変数の候補集合を初期の候補集合に戻す．
  7. Rが空集合になったら終了．
  8. R中の変数vを1つ選びRから削除し，
     目的関数を最小化するvの候補値をランダムに選び
     vに割り付け，7へ戻る．
```

図3　Really-Full-Lookahead Greedy Algorithm

る．

STEP 1は制約伝播（Arc Consistency または Waltz's Filtering）を用いて制約違反を起こさない範囲で可能な限り多くの変数に値を割り付ける処理である．この処理は後戻りしない点を除いて Really-Full-Lookahead Backtracking と等価である．無矛盾な割付けが不可能な場合，後戻りする代わりに不可能な変数を R に記録する．STEP 2は，STEP 1で制約違反を余儀なくされた R 中の変数に対する Greedy Algorithm による割付け処理である．

このように，STEP 1で割付け前に制約伝播により全変数間の制約を考慮することにより，制約違反が少ない品質の高い初期割付けを作成することができる．また，「可能な候補数が小さい変数を優先する」など，バックトラック手法における変数選択・値選択のヒューリスティックを併せて利用することにより，より違反制約が少ない初期割付けを作成することが可能となる．

4．適用事例

COASTOOL の適用事例として，学校の時間割編成システム［5］を紹介する．なお，上位生産計画立案システムへの適用例は，文献［8］を参照されたい．

［1］時間割編成問題

一般に，学校時間割り編成問題は複雑で制約条件が厳しく大規模であるため，計算機による自動化が困難な問題として知られている．特に埼玉県立久喜北陽高校は，各学年 10 クラス・先生 70 人（内非常勤 10 人）と最大級規模の高校であり，普通科・商業科（情報科/簿記科）・外国語コースと多彩なコースを持ち，多種の同時展開授業・連続授業等を展開している．時間割り編成は非常に困難であり，通常，先生が 100 人日（10 人×10 日）程度の工数を掛けて時間割りを編成している．

問題は，授業(808 個)に対して 54 種類(述べ 24,990 個）の制約を考慮しながら時間（34 個）を割り付ける問題である．制約は，「先生(クラス)の時間の一意性」など必須の制約と，「週に 2 単位の授業をなるべく連日に割り付けない」など緩和可能な制約とがあり，それぞれ違反点数が定められている．目的関数は違反制約の違反点数の合計を最小化することである．なお，各授業を担当する先生は前もって与えられている．

［2］評　価

埼玉県立久喜北陽高校の H 4 年度／H 5 年度時間割り編成問題を用いて COASTOOL の実験評価を行なった．なお，最適化アルゴリズムとして，RFLG アルゴリズムによる初期割付けと Min-Conflicts Hill-Climbing (MCHC) ［7］による自動調整の組合せを用いた．

・［生産性］：

H 4 年度の問題記述に要した開発工数は 3 人日であり，COASTOOL の開発性の高さを確認した．問題記述は 2,000 行で，うち 1,500 行がデータ記述であった．また，H 5 年度用に問題記述を書き換える工数は 1.5 人日であった．さらに，先生からのフィードバックに対し，制約の追加・変更・削除と違反点数の変更のみで適切な対応が可能であり，メンテナンス性の高さを確認した．

・［性　能］：

時間割り編成に要する時間は H 4 年度で約 1 時間，H 5 年度で約 6 時間であり実用に十分な性能を確認した(NEC EWS 4800／350 上の NXLISP，EXCORE／CL を利用)．

・［品　質］：

COASTOOL が立案した最適な時間割りは違反点数が最低（1 点）の制約を 1 個違反するのみ（違反点数の合計が 1 点）であった（H 4 年度）．時間割り担当の先生により，「人手の時間割りの 95％以上のレベルに達しており，ほぼ利用可能である」との評価が得られた．H 5 年度はかなり難しい問題であった(違反制約 19 個 41 点）が，90％以上の満足度が得られた．また，これを先生が机上で修正するために要した時間は半日以下であった．

・［RFLG アルゴリズムの評価］：

久喜北陽高校時間割りにおける解の組合せ数は 10 の 1200 乗を越える．このように非常に大規模な問題に対してバックトラック手法が有効でないことは明らかである．

比較評価のため，Greedy Algorithm と Simulated Annealing（SA）との組合せによる実験を行なった．SA は局所最適解に陥りにくいという特長を持つが，弱いバイアスのもとでランダムに探索するアルゴリズムであるため，大規模な問題に対して長時間を必要とする．H 5 年度問題では RFLG アルゴリズムによる初期割付け結果（所要 5 時間）と同等の品質の解に辿り

着くために約3日間の計算時間を要した．一方，より高速に冷却した場合，制約が厳しいため局所最適解に陥り，解の品質がRFLGアルゴリズムよりも低くなった．

時間割り担当の先生による満足度はGreedy AlgorithmとMCHCの組合せが70％以下であるのに対して，RFLGアルゴリズムとMCHCの組合せで95％以上と，大幅な品質の改善が確認できる．RFLGアルゴリズムは計算時間が大きいが実用的な範囲内である．ただし，制約に依存して計算時間が大きく変化するという問題があり，H5年度ではH4年度の5倍程度の時間を要している．ただし，STEP1で扱う変数・制約を制限することにより，解の品質は比較的低いが高速に初期割付けを行なうことが可能である．

・[リアルタイム計画調整]：

すでに作成した時間割結果や，一部制約違反を起こした時間割結果に対して，ユーザが不完全な修正を行なった場合，システムは，制約違反を最小化することで自動的な時間割調整を行なう．この機能によって，ユーザは，部分的で不完全な時間割変更を指示するだけで，容易に希望の時間割を作成することができる．

以上のように，RFLGアルゴリズムとMCHCとの組合せにより，制約の厳しい大規模問題に対して，実用時間内に高い品質の解を作成することが確認された．

5．おわりに

ESアプローチの利点と問題点等を確認した後，現状のスケジューリングESのニーズ分析を行ない，計画ESに関する技術開発要件を整理した．その後で，従来のESアプローチの長所を活かし欠点を克服するものとして，制約ベース・スケジューリングシェルを紹介し，その具体的適用事例を通して有効性を示した．そこでは，制約処理機構をベースにすることで，汎用自動割り付けアルゴリズムと専門家の計画立案ノウハウを適宜利用できる枠組が整備され，ユーザとの対話的な協調的スケジュール（リアルタイム計画調整）作業環境を容易に実現できた．

参考文献

[1] 森：運航乗務員乗務割作成支援システム，C&C SYSTEM REVIEW, No. 21, 1990.

[2] 片岡：ソニーグループ殿における生産計画エキスパートシステム，NEC技報，平成4年5月号，pp 8-12.

[3] 塩津：武田薬品工業(株)殿におけるエキスパートシステム事例，NEC技報，平成4年5月号，pp 13-18.

[4] 潮田，岩本，山之内，渡辺：プロセス産業を対象とした生産計画エキスパートシステム構築ツール，生産スケジューリングシンポジウム'94，p 25-30, 1994. 10.

[5] Yosikawa, M., Kaneko, K., Nomura, Y. and Watanabe, M. :A Constraint-Based Approach to High-School Timetabling Problem : A Case Study, Proc. of AAAI 94, 1994. 7.

[6] Musick, R. and Russell, S. : How Long Will It Take?, AAAI-92.

[7] Minton, S., et. al. : Minimizing Conflicts : A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems, Artificial Intelligence 58.

[8] 椎名，吉川，渡辺：パソコン生産計画エキスパートシステムの開発と評価，生産スケジューリングシンポジウム'94，p 112-117, 1994. 10.