メタヒューリスティクスによる汎用ソルバーの構築

野々部宏司

組合せ最適化問題に対する汎用ソルバーの実現を目指し，著者らが開発してきた制約充足問題に対するメタヒューリスティック・アルゴリズムの紹介を行う．このソルバーはこれまでに数多くの組合せ最適化問題に適用されており，最適化エンジンとして商用の数理計画パッケージにも組み込まれている．本稿では，ソルバーの概要について述べた後，汎用ソルバーを用いたアプローチの有効性を示すために参加した国際コンペティションの結果を報告する．さらに，利便性を高めるために行ったソルバーのExcelアドイン化についても述べる．

キーワード：組合せ最適化，汎用ソルバー，メタヒューリスティクス，制約充足問題

1. はじめに

日本オペレーションズ・リサーチ学会のホームページには，オペレーションズ・リサーチについて，

問題を科学的，つまり「筋のとおった方法」を用いて解決するための「問題解決学」

であると書かれている．本特集のキーワードである「最適化技術」は，ここでいう科学的方法を支える主要技術のひとつであるといえる．

最適化技術を用いた問題解決のプロセスを大雑把に捉えると，図1に示すように，

1. 現実の問題を最適化問題として定式化し，入力データを与え，最適化モデル（インスタンス）を生成する，
2. 最適化計算によってモデルの最適解，もしくは近似最適解を得る，
3. 得られた結果を分析し，モデルの妥当性や有効性を検証する，

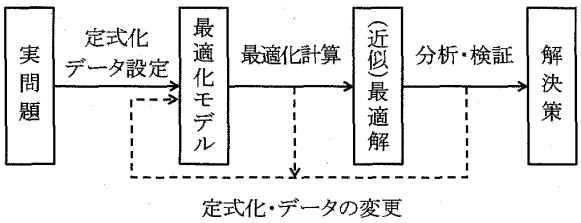

図1 最適化技術を用いた問題解決プロセス

といった手順を，満足・納得のいく解決策が得られるまで，適宜，定式化やデータの変更を行いながら繰り返し行うものであると考えることができる．本稿では，この一連のプロセスにおける「最適化計算」に焦点を当て，特に問題が組合せ最適化問題として定式化される場合を想定する．

上述のプロセスにおける「最適化計算」では，定式化された最適化問題に対し，十分な性能のソルバーをいかに用意するかがポイントとなるが，実務においてはソルバーの性能のみならず，ソルバーが利用できるようになるまでの手間や時間，費用も考慮する必要がある．利用可能なソルバーは利用すればよく，モデルを作成する段階で，既存のソルバーが利用できないか考えることも大切である．汎用ソルバーは，このような状況で，いろいろな問題に適用できることを目的としたものである．組合せ最適化問題に対する汎用ソルバーとしては混合整数計画（mixed integer programming, MIP）ソルバーが代表的であり，商用，非商用を含め，多数のパッケージが存在する．近年では，大規模なMIPモデルも厳密に解くことが可能になってきており，今後のさらなる性能向上も期待されている．

しかしながら，すべての組合せ最適化問題をMIPソルバーで現実的に解けるというわけではなく，状況に応じて定式化を見直したり，問題をいくつかの部分問題に分割したり，MIPソルバーの適用をあきらめて別のソルバーを利用したり，問題に特化したアルゴ

ののべ　こうじ
法政大学 デザイン工学部
〒102-8160 千代田区富士見2-17-1

リズムを新たに設計・開発したりするなどの必要が出てくることも多い．結局のところ，ありとあらゆる組合せ最適化問題を，あるひとつの汎用ソルバーで扱うことは実用上困難であり，適用範囲や性質の異なる複数のソルバーを使い分けることが（少なくとも現時点では）現実的な方法であるといえる．この考えのもと，茨木俊秀・京都大学名誉教授らの研究グループでは，様々な組合せ問題を最適化技術によって解決する方策として，いくつかの標準問題を「問題解決エンジン」と呼ばれる高性能なソルバーとともに用意しておき，解くべき問題を適当な標準問題に帰着して解く，というアプローチを提案，実践している[1][2]．これらのソルバーの特徴は，厳密な意味での最適化を追求せず，良質の解を短い計算時間で求めることを目指している点にあり，アルゴリズムの枠組みとしては，基本的にメタヒューリスティクスを用いている．

著者もこのグループのメンバーとして，これまでに汎用ソルバーの開発を行ってきた．以下では，そのひとつである重み付き制約充足問題（weighted constraint satisfaction problem, WCSP）ソルバーを取り上げ，その概要を述べた後，ソルバーの性能評価の一環として参加した国際コンペティションの結果を報告する．さらに，利便性を高めるために行ったソルバーのExcelアドイン化についても述べる．

2. WCSPソルバー

本稿で紹介するWCSPソルバー[8][9]は，組合せ最適化問題に対する汎用ソルバーとして数多くの問題に適用されており，最適化エンジンとして商用の数理計画パッケージにも組み込まれている．さらに，より広範な問題に適用できるよう機能拡張されている[6]．

2.1 問題設定

WCSPは，制約充足問題（constraint satisfaction problem, CSP）の自然な拡張のひとつである．CSPは，n個の変数$X_i(i=1, 2, ..., n)$と，各X_iについてX_iのとり得る値の集合D_i，およびm個の制約$C_l(l=1, 2, ..., m)$で定義され，すべての制約を満たすよう各変数X_iに値$j \in D_i$を割当てる問題である．ここで，変数X_iとその値$j(\in D_i)$の組のそれぞれに対し，0-1変数x_{ij}を，

$$x_{ij}=\begin{cases}1, & \text{変数 } X_i \text{ が値 } j \text{ をとる} \\ 0, & \text{その他}\end{cases}$$

と定義し，割当てを0-1ベクトル

$$\boldsymbol{x}=(x_{ij} | i=1, 2, ..., n, j \in D_i)$$

を用いて表すことにする．このとき，各変数には値が1つずつ割当てられることから，

$$\sum_{j \in D_i} x_{ij}=1, i=1, 2, ..., n, \tag{1}$$

でなくてはならない．以下では，条件(1)を満たす0-1ベクトル$\boldsymbol{x}$をCSPあるいはWCSPの解と呼ぶ．

CSPの目的がすべての制約を満たす解を求める（もしくはそのような解が存在しないことを示す）ことであるのに対し，WCSPは，制約違反が最小の解を求めることを目的とする．より厳密には，WCSPにおいて各制約C_lは，その重要度を表す重みw_l (≥ 0)と制約違反度を表すペナルティ関数$p_l(\boldsymbol{x})$を持ち，WCSPは以下の最小化問題

$$\text{minimize} \quad p(\boldsymbol{x})=\sum_l w_l p_l(\boldsymbol{x}) \tag{2}$$

$$\text{subject to} \quad (1)$$

として定式化される．ただし，各p_lは，解$\boldsymbol{x}$がC_lを満たすとき$p_l(\boldsymbol{x})=0$，満たさないとき$p_l(\boldsymbol{x})>0$であるとする．

なお，制約C_lを絶対制約として扱いたい場合は，重みw_lを十分大きな値に設定することで対処する．また，WCSPは目的関数を陽には持たないが，これを考慮制約として扱うことができる．例えば，非負の値をとる関数$g(\boldsymbol{x})$を最小化したい場合には，$g(\boldsymbol{x})$をそのままペナルティ関数と見なして，「$g(\boldsymbol{x})=0$」という制約を追加すればよい．

CSPでは，各制約C_lの記述方法について特に制限はなく，線形や非線形の等式や不等式，論理式，制約を満たす値の組の集合など，形式を自由に選ぶことができる．WCSPでは，ペナルティ関数p_lを記述することがC_lを記述することに対応し，p_lの記述方法について原理的には特に制限はない．この自由度の高さが，CSP，WCSPの最大の特長である．

2.2 アルゴリズム概要

文献[8][9]で提案されたWCSPソルバーはタブー探索法に基づいている．探索空間は解集合全体（制約(1)を満たす0-1ベクトル集合）であり，解$\boldsymbol{x}$の近傍$N(\boldsymbol{x})$は，$\boldsymbol{x}$から，ある1つの変数の値を変更することで得られる解（近傍解と呼ぶ）の集合で定義される．すなわち，解$\boldsymbol{x}$に対し，変数X_iの値を現在の値j'から他の値jへ変更することで得られる解を$\boldsymbol{x}(X_i \leftarrow j)$で表せば，

$$N(\boldsymbol{x})=\left\{\boldsymbol{x}(X_i \leftarrow j) \middle| \begin{array}{l} i \in \{1, 2, ..., n\}, j \in D_i, \\ x_{ij}=0 \end{array}\right\}$$

である．ただし，各反復の近傍探索において，$N(\boldsymbol{x})$

内の解すべてを移動先の解候補とするのではなく，最近の t 反復に値が変更された変数の値を，再度変更することで得られる近傍解については，解候補から除外する（ただし，例外規則を設けている）．ここで，t はタブー期間と呼ばれるプログラム・パラメータであり，その値は探索状況に応じて動的に変化する仕組みになっている．さらに，現在の解 $\boldsymbol{x}$ が違反している制約 C_l（すなわち，$p_l(\boldsymbol{x})>0$ である制約）のすべてに対して，$p_l(\boldsymbol{x}')\geq p_l(\boldsymbol{x})$ である解 $\boldsymbol{x}'\in N(\boldsymbol{x})$ についても解候補から除外することとしている．これには，近傍探索の高速化に加え，未探索の領域に探索を移す（多様化と呼ばれる）効果がある．

近傍探索中，近傍解 $\boldsymbol{x}'\in N(\boldsymbol{x})$ の評価に目的関数(2)を用いると，ペナルティ重み w_l の大きな制約（絶対制約など）を満たす解に探索が制限されてしまい，効果的ではない．そのため，近傍探索における解の評価には，以下の評価関数

$$f(\boldsymbol{x})=\sum_l v_l p_l(\boldsymbol{x})$$

を用いている．ここで，$v_l(l=1, 2, ..., m)$ は，$0\leq v_l \leq w_l$ を満たす実数である．これにより，v_l の値によっては，重みの大きな制約に違反するような解への移動も起こりやすくなり，結果として，解空間のより広い範囲を探索できるようになることが期待される．ただし，本来の目的は元の目的関数(2)を最小化することであるため，これが実現されるよう，v_l の値は探索状況に応じて自動調整される．

2.3 制約の実装

上述の通り，ペナルティ関数 p_l の記述方法について，原理的には特に制限はないが，実装上，p_l は非負の整数値を取るものとしている．さらに計算効率および探索効率を高めるため，各制約 C_l について，与えられた解 $\boldsymbol{x}$ に対し，$p_l(\boldsymbol{x})$ の値を返すルーチンや，変数 X_i の値を j に変更したときのペナルティ変化量 $p_l(\boldsymbol{x}(X_i\leftarrow j))-p_l(\boldsymbol{x})$ を計算するルーチンが用意されている必要がある．

0-1 変数 x_{ij} に対する線形，および 2 次の等式・不等式制約，all-different 制約（与えられた変数集合 V_l に対し，V_l に含まれる変数はすべて異なる値を取らなくてはならないとする制約）などの標準的な制約については，これらのルーチンはすでに実装されている．

ユーザが新たに制約（ペナルティ関数）を定義することもできる．その際，他の制約のペナルティ関数値を用いた記述も可能である．例えば，それぞれのペナルティ関数が 0 または 1 の値をとる 3 つの制約 C_1, C_2, C_3 のうち，2 つ以上に違反してはならない，といった制約 C_4 は，そのペナルティ関数を

$$p_4(\boldsymbol{x})=\max\{p_1(\boldsymbol{x})+p_2(\boldsymbol{x})+p_3(\boldsymbol{x})-1, 0\}$$

と定義することで記述できる．この場合，ユーザは $p_4(\boldsymbol{x})$ の値を返すルーチンを用意するだけでよく，ペナルティ変化量を計算するルーチンを実装する必要はない．

3. INRC2010

WCSP ソルバーを用いたアプローチの有効性を評価するため，国際コンペティション First International Nurse Rostering Competition 2010（INRC2010）[5]に参加した．INRC2010 は，ナーススケジューリング問題に対するアルゴリズムの性能を競う国際コンペティションであり，2010 年 3 月に開始され，同年 8 月に開催された PATAT2010（8th International Conference on the Practice and Theory of Automated Timetabling）において結果報告がなされている．本節では，その概要と結果について述べる（詳細については，文献[7]をご参照いただきたい）．

3.1 問題定義

INRC2010 で提示された問題は，与えられた N 人のナースの T 日間の勤務スケジュールを作成する問題であり，各日の各ナースに，どの勤務シフトを割当てるか，もしくは割当てないかを決定することが求められている（勤務シフトが割当てられない日は休暇日となる）．各勤務シフトには，勤務時間帯（例えば，8：30〜16：30 など）と，そのシフトに必要なスキル（勤務経験や役職など）が定められており，それらのスキルを有しないナースに当該シフトを割当てることはできない．また，各ナースには，そのナースの有するスキルと雇用契約が与えられている．

勤務スケジュールを作成する際の制約条件を以下に列挙する．

(i) 各ナースには 1 日あたり高々 1 つの勤務シフトしか割当てることはできない．

(ii) 各日，各勤務シフトについて，あらかじめ設定された人数のナースを過不足なく確保しなくてはならない．

以下の制約は個々のナースのスケジュールに関するものであり，制約の有無や上下限値，週末の定義（「土日」，「金土日」など）は雇用契約によって異なる．

(iii) 各勤務シフトについて，T 日間の割当て回数が上下限の範囲内になくてはならない．

(iv) 連続勤務日数（連続して勤務シフトが割当てられている日数）が上下限の範囲内になくてはならない．

(v) 連続休暇日数（連続してどの勤務シフトも割当てられていない日数）が上下限の範囲内になくてはならない．

(vi) 勤務シフトが1つでも割当てられている週末が，上限を超えて連続してはならない．

(vii) 週末に勤務シフトを割当てる場合，その週末に休暇日を設けてはならない．

(viii) 週末に勤務シフトを割当てる場合，その週末に含まれるすべての日に同じ勤務シフトを割当てなくてはならない（制約(vii)よりも厳しい条件）．

(ix) 夜勤シフト（深夜0時を含む勤務シフト）を割当てる場合，その直後2日間に，夜勤シフト以外の勤務シフトを割当ててはならない（夜勤（夜勤が連続する場合は最終日の夜勤）を終えた後に，2日間の休暇日を確保することを意図している）．

(x)「夜勤の翌日に日勤」など，望ましくない勤務シフトの並びを避けなくてはならない．

また，個々のナースの要望として以下の制約がある．

(xi) あるナースの，ある日について，その日を休暇日にしてはならない，もしくは休暇日にしなくてはならない．

(xii) あるナースの，ある日に，ある特定の勤務シフトを割当ててはならない，もしくは割当てなくてはならない．

これらの制約条件のうち，(i)，(ii)は絶対制約であり，必ず満たす必要がある．一方，(iii)以降はすべて考慮制約であり，必ずしも満たす必要はないが，満たさない場合にはペナルティが課される．各考慮制約にはペナルティ重みが付与されており，絶対制約をすべて満たしたうえで，考慮制約の違反ペナルティの加重和を最小化することが目的である．

INRC2010では，Sprint，Medium，Longの3つのトラックが設定されている．問題設定はすべてのトラックにおいて共通であるが，ナース数，勤務シフト数，雇用契約の内容（制約条件の有無）などの違いにより問題の規模や難易度が異なっており，それぞれ，10秒，10分，10時間をおおよその計算時間として想定している．

3.2 コンペティション実施概要

コンペティションでは，まず2010年3月の開始時に，Sprintトラック10問，Medium，Longトラックそれぞれ5問の問題例が公開され，その後，同じ数の問題例が追加で公開される．参加者は，自身が参加するトラックの公開問題例をすべて解いたうえで，得られた解と計算に用いたプログラム（実行ファイル）をあらかじめ定められた期日までに主催者へ送る必要がある．

主催者は，参加者から送付された解のペナルティに基づいて各トラック5チームずつのファイナリストを決定した後，それまで隠されていた問題例をファイナリストのプログラムを用いて解き，その結果によってファイナリストの最終順位を決定する．

3.3 WCSPソルバーの適用

INRC2010の問題は，ナース $i(i=1, 2, ..., N)$ の第 j 日目 $(j=1, 2, ..., T)$ を変数 X_{ij} に対応させ，その領域 D_{ij} を，ナース i に割当てることのできる勤務シフトの集合に休暇日を表すダミーシフトを加えた集合とすることで，WCSPとして定式化できる．ここで，制約(i)は，WCSPの制約(1)によって自動的に満たされる．その他の制約については，基本的に，変数 X_{ij} とその値 $s \in D_{ij}$ の組に対応する0-1変数 x_{ijs} に関する線形もしくは2次の等式・不等式によって記述するが，制約(vi)，(viii)については，ユーザ定義制約を導入することで対処することとした．

3.4 コンペティション結果

WCSPソルバーによって得られた最良解をプログラムとともに主催者に送付した結果，すべてのトラックにおいてファイナリストに選ばれ，最終的にSprintトラックで2位，Mediumトラックで3位，Longトラックで4位を獲得することができた．

WCSPソルバーによるアプローチの最大のメリットは，与えられた問題を解く際に，アルゴリズムの設計や実装に要する手間や時間を削減できることである．性能に関しても，ナーススケジューリング問題に対しては非常に高いことが，今回のコンペティションの結果によって示されたと考えている．

なお，2002年および2007年には，時間割作成を対象とした同様の国際コンペティションInternational Timetabling Competition 2002, 2007（ITC2002, ITC2007）が実施されている．著者らは，今回と同様，WCSPソルバーを用いてITC2007に参加し，設定さ

れた3つのトラックにおいて，それぞれ3位，2位，3位の好成績を収めている[3]．このことからも，WCSPソルバーを用いたアプローチの有効性を確認することができる．

4. Excelアドイン化

WCSPソルバーは，C++クラス・ライブラリとして実装されており，ユーザが自身のプログラムからこれを呼び出して利用することが可能である．ユーザが手軽に利用できるよう，簡易的な入出力インタフェースをつけた実行ファイル形式での提供を行っており，また，モデリング言語AMPLからの呼び出しにも対応している．今回，さらに利便性を高めるため，WCSPソルバーのExcelアドイン化を行った．

Microsoft社製の表計算ソフトExcelには，最適化問題を解くためのソルバー機能（Excelソルバー）がアドインとして付属している．大規模な問題を扱うことはできず，性能面では他の商用パッケージに劣るものの，SUMやSUMPRODUCTなどのExcel関数やセル参照機能を利用しながらワークシート上に最適化モデルを構築することができるという大きな利点がある．WCSPソルバーをExcelアドイン化することで，この利点を生かしつつ，Excelソルバーでは扱うことのできない規模の問題も解くことができるようになる．なお，現時点では，線形等式・不等式制約のみに対応している．

例として，文献[4]のナーススケジューリング問題をExcelワークシート上にモデル化し，WCSPソルバーで解く様子を図2に示す．Excelソルバーと同様，「変化させるセル」や「制約条件」の左辺（「セル参照」），右辺（「制約条件」）にセル参照や値を入力することで問題の設定を行う．ただし，「目的セル」がなく，各制約の「重み」を入力する点や計算時間や乱数の初期値を設定する点などがExcelソルバーとは異なる．

図2の例の場合，ワークシート上部の「0-1変数」と記された領域内の各セルが，ナースiの第j日目にシフトsを割当てるか否かを表す0-1変数x_{ijs}に対応し，このセル範囲を「変化させるセル」としてソルバーに指定している．これらの値によって定まる「勤務表」がその下に表示（各ナースの各日に，日勤シフトを表す「日」や夜勤シフトを表す「夜」などが表示）されており，この領域内のセルを参照する数式によって種々の制約を記述することができる．例えば，計画期間中，あるナースに割当てられる日勤シフトの数は「=COUNTIF（H14:AK14,"日"）」といった数式で

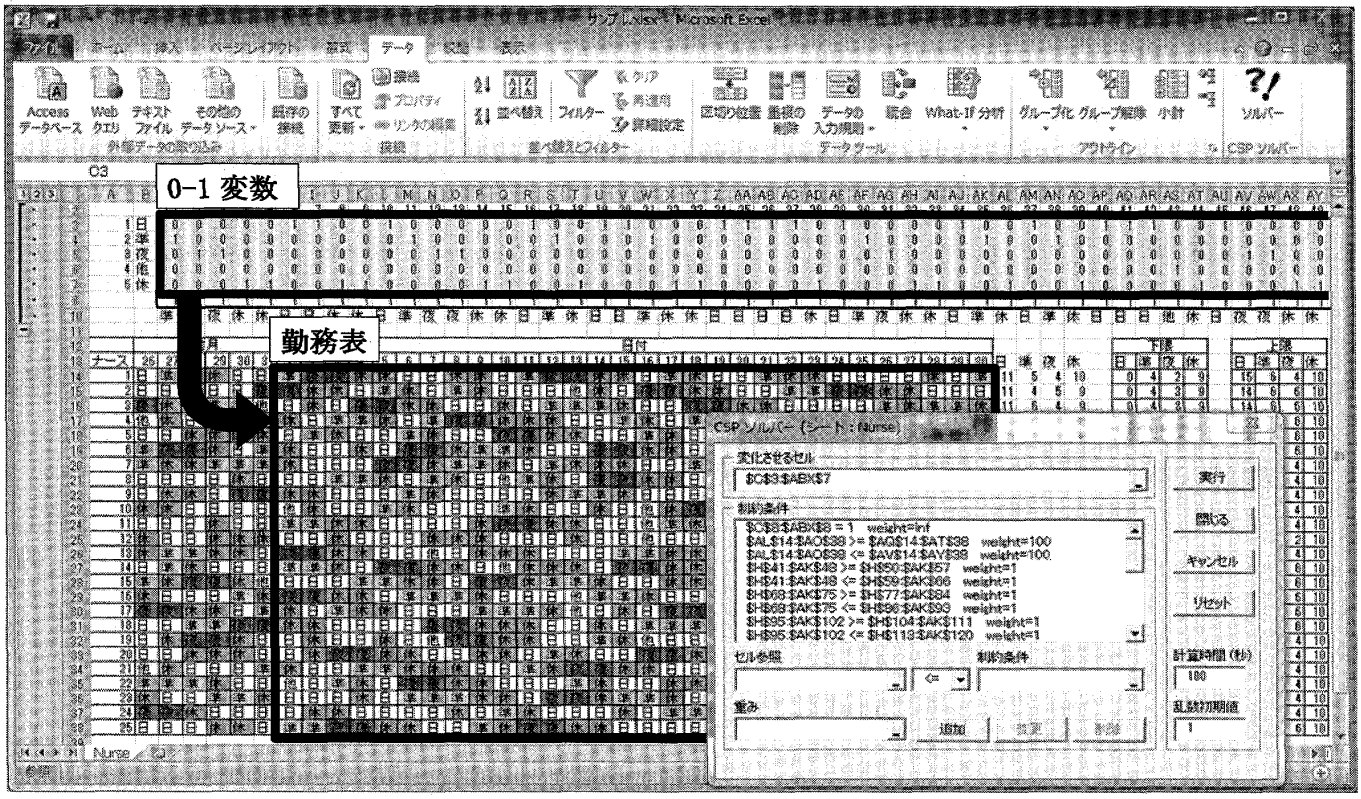

図2 WCSPソルバーのExcelアドイン化

与えられるため，この数式が入力されたセルを「制約条件」の左辺に指定することで，この上下限制約を記述することができる．同様に，夜勤3連続禁止といった制約を記述するには，隣接する3個のセル（連続する3日間に対応）のうち何個のセルに「夜」と表示されているかを表す数式「=COUNTIF（F14:H14, "夜"）」を，各ナースにつき計画日数分用意し，すべてに対して2以下とすればよい．このような制約記述は視覚的にも理解しやすく，また，条件付き書式機能によって制約違反のセルを強調表示すれば，モデルや解の検証にも役立つ．

Excel ワークシート上での最適化計算は，本格的な運用には適さないかもしれないが，「最適化問題をとりあえず試しに解いてみたい」といった状況で，手軽に試すことのできるひとつの方法として有用であると考えられる．

5. おわりに

本稿では，制約充足問題に対するメタヒューリスティック・アルゴリズムの紹介を行った．今後も，組合せ最適化問題に対する汎用ソルバーとして，いろいろな問題に適用していきたい．

ところで，冒頭で述べた問題解決プロセスにおいて，「最適化計算」はひとつのステップに過ぎず，「実問題のモデル化」や「入力データの収集」，そして「モデルの修正」を適切に行わない限り，問題の解決には至らない．例えば，INRC2010 の問題設定は実問題に基づいてはいるが，コンペティションで用意された問題例はアルゴリズム評価のための単なるベンチマークであり，これらが高速に解けるからといって，実際の現場の問題が即解決されるというわけではない．

汎用ソルバーの開発という観点においては，与えられた問題をいかに解くかが主題となるが，問題解決プロセス全体を考えると，良い解を高速に求められるようにすることに加え，汎用性やユーザの利便性を高めることが重要である．さらに，モデル分析に役立つ情報を出力できるようにすること（例えば，入力データの変更に伴い解がどのように変化するのか，厳密な感度分析は無理だとしても，大雑把な情報だけでも出力できるようにすることなど）も，問題解決や意思決定の効率化のためには必要であると考えられる．

参考文献

[1] 茨木俊秀：「問題解決エンジン」群とモデリング，オペレーションズ・リサーチ，Vol. 50, No. 4, pp. 229-232 (2005).

[2] T. Ibaraki : A personal perspective on problem solving by general purpose solvers, *International Transactions in Operational Research*, Vol. 17, pp. 303-315 (2010).

[3] 茨木俊秀，熱田光紀，野々部宏司：汎用ソルバによる時間割作成—国際コンペティション ITC2007 に参加して—，スケジューリング・シンポジウム 2008 講演論文集，pp. 173-176 (2008).

[4] A. Ikegami and A. Niwa : A subproblem-centric model and approach to the nurse scheduling problem, *Mathematical Programming*, Vol. 97, pp. 517-541 (2003).

[5] INRC2010 Home Page : http://www.kuleuvenp-kortrijk.be/nrpcompetition

[6] 野々部宏司：ユーザ定義制約の追加を考慮したメタヒューリスティクスに基づく制約最適化ソルバー，日本オペレーションズ・リサーチ学会 2010 年度春季研究発表会アブストラクト集，pp. 116-117 (2010).

[7] 野々部宏司：メタヒューリスティクスを用いた制約最適化ソルバーのナーススケジューリング問題への適用，電気学会研究会資料，システム研究会，ST-10-6, pp. 27-31 (2010).

[8] K. Nonobe and T. Ibaraki : A tabu search approach for the constraint satisfaction problem as a general problem solver, *European Journal of Operational Research*, Vol. 106, pp. 599-623 (1998).

[9] K. Nonobe and T. Ibaraki : An improved tabu search method for the weighted constraint satisfaction problem, *INFOR*, Vol. 39, pp. 131-151 (2001).