# 問題構造の解析に基づく組合せ最適化アルゴリズムの自動構成

梅谷　俊治

産業や学術の幅広い分野において，組合せ最適化の専門家でない多くの利用者が，日々新たに生じる現実問題を短期間で解決できる環境を実現するためには，これらの問題を整数計画問題や制約充足問題などの汎用的な組合せ最適化問題に定式化し，その問題に対する高性能なアルゴリズムを開発することが望ましい．しかし，問題の汎用性が高まればアルゴリズムの性能向上に利用できる特徴的な構造が失われるため，汎用的な組合せ最適化問題に対して高性能なアルゴリズムを開発することは容易ではない．本稿では，個別の入力データからアルゴリズムの性能向上に役立つ情報を発見し，アルゴリズムの設定や構成を自動的に決定することで，汎用的かつ高性能な組合せ最適化ソルバーを実現する試みとその現状をいくつか紹介する．

キーワード：組合せ最適化，整数計画問題，メタヒューリスティクス

## 1.　はじめに

組合せ最適化の分野では，これまで多くの問題に対して，個々の問題に含まれる特徴的な構造を利用した高性能なアルゴリズムが開発されてきた．しかし，産業や学術の幅広い分野において，専門家でない多くの利用者が，日々新たに生じる現実問題を適切な組合せ最適化問題に定式化することに加え，適切なアルゴリズムを適用することは困難である．

利用者が現実問題を短期間で解決できる環境を実現するためには，多様な現実問題を自然な形に定式化できる汎用的な組合せ最適化問題を定義し，その問題に対する高性能なアルゴリズムを開発することが望ましい．しかし，問題の汎用性が高まればアルゴリズムの性能向上に利用できる特徴的な構造が失われるため，汎用的な組合せ最適化問題に対して高性能なアルゴリズムを開発することは容易ではない．

本稿では，このような状況に対し，個別の入力データからアルゴリズムの性能向上に役立つ情報を発見し，アルゴリズムの設定や構成を自動的に決定することで，汎用的かつ高性能な組合せ最適化ソルバーを実現する試みとその現状をいくつか紹介する．

## 2.　計算困難な組合せ最適化問題

近年，産業や学術の分野における多くの重要な問題が組合せ最適化問題として定式化できることが知られるようになり，大規模かつ複雑な組合せ最適化問題を効率良く解くことがますます強く求められている．しかし，実務上重要な組合せ最適化問題の多くはNP困難と呼ばれるクラスに属する難しい問題であり，厳密な最適解を効率良く求めるアルゴリズムは知られていない．これらの難しい組合せ最適化問題に対する代表的なアプローチとして，問題を整数計画問題に定式化して分枝カット法に基づくアルゴリズムを適用する方法（図 1）や，個々の問題に含まれる特徴的な構造を利用してメタヒューリスティクスに基づくアルゴリズムを開発する方法（図 2）などがある．

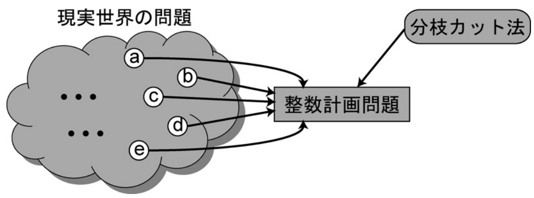


**図 1**　整数計画問題に対する分枝カット法

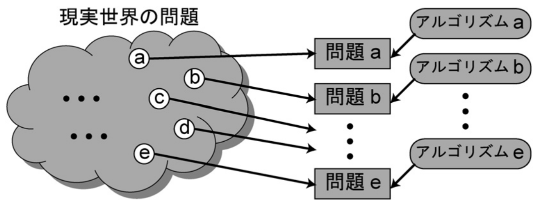


**図 2**　問題の特徴的な構造を利用したメタヒューリスティクス

整数計画問題は多くの現実問題を定式化できる汎用

うめたに　しゅんじ
大阪大学大学院情報科学研究科
〒 565–0871 大阪府吹田市山田丘 2–1

  

的な組合せ最適化問題の一つであり，厳密な最適解を求める分枝カット法にさまざまなアイデアを盛り込んだ高性能なソルバーが多数公開されている [5, 16, 17]．しかし，定式化に際して多くの変数や制約条件の導入を必要とする問題や，定式化自体は簡潔であっても分枝カット法が苦手とする問題も多く，すべての組合せ最適化問題が整数計画ソルバーだけで効率良く解けるわけではない．

現実問題では厳密な最適解が常に求められているわけではなく，むしろ精度の高い近似解で十分に問題解決に役立つ場合が多い．メタヒューリスティクスは近似解法のさまざまなアイデアを組み合わせて，探索の集中化と多様化をバランス良く実現する枠組みである [6, 14, 23]．しかし，現実問題に対して高性能なメタヒューリスティクスを開発するには，個々の問題に含まれる特徴的な構造を利用する必要があるため，組合せ最適化の専門家であっても新たな問題に取り組むたびにアルゴリズムの開発に多くの時間と労力を要する場合が多い．

茨木俊秀・京都大学名誉教授らのグループは，現実の幅広い問題に適用できる汎用性とアルゴリズムの性能向上に利用できる構造を合わせ持つ組合せ最適化問題を標準問題としていくつか選び，各標準問題に対して「問題解決エンジン」と呼ばれる高性能なメタヒューリスティクスを開発し，現実問題を適切な標準問題に定式化して解くことを提案し，数々の事例を通じてその有効性を示している [10, 11, 12]．しかし，組合せ最適化の専門家の助力なしに，利用者が現実問題を適切な標準問題に定式化することは容易ではない．また，現実問題に対応するためにアルゴリズムの修正が必要となる場合も多く，これらの問題解決エンジンが組合せ最適化の専門家の手を離れて広く普及するには至っていない．

## 3. 問題構造の解析に基づくアルゴリズムの自動構成

整数計画問題は産業や学術の幅広い分野に現れる問題を記述できる汎用的な組合せ最適化問題であるが，$\min\{\boldsymbol{c}^T\boldsymbol{x} \mid A\boldsymbol{x}=\boldsymbol{b},\ \boldsymbol{x}\in\mathbb{Z}^n\}$ という記述からアルゴリズムの性能向上に役立つ特徴的な構造を見いだすことは困難である．一方で，現実問題を定式化して得られる個別の入力データは無秩序でまったく構造を持たないわけではなく，実際にはグラフ，論理，順序，割当などの典型的な離散構造で記述される特徴的な部分から構成されている場合が多い．例えば，整数計画問題のベンチマーク問題集 MIPLIB [1, 13] の問題例は $\sum x_j = 1$ などの特殊な形の制約式を多く含み，図 3 に示すような特徴的な制約行列で構成される傾向がある．また，あらかじめアルゴリズムの仕様をすべて決定する必要はなく，実行時にアルゴリズムの一部の仕様を変更すれば，個別の入力データから抽出した情報をアルゴリズムの性能向上に利用することは可能である．

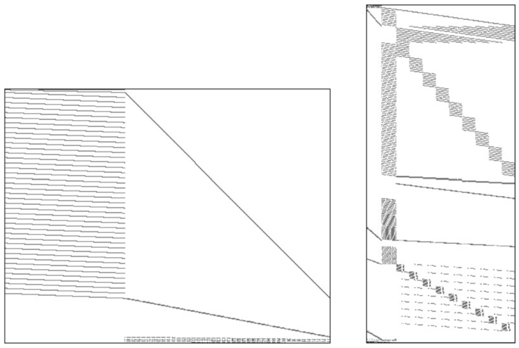

**図 3**　特徴的な制約行列を持つ整数計画問題の係数行列の非ゼロ成分パターン（左: stp3d, $159488 \times 204880$, 右: acc-tight4, $3285 \times 1620$）

以降では，個別の入力データからアルゴリズムの性能向上に役立つ情報を発見し，アルゴリズムの設定や構成を自動的に決定することで，汎用的かつ高性能な組合せ最適化ソルバーを実現する試みとその現状をいくつか紹介する．

## 4. 分解法の自動化

大規模な整数計画問題では，制約行列のほとんどの成分が 0 となり（疎行列と呼ばれる），非ゼロ成分が角状に固まって現れるブロック対角と呼ばれる構造を持つ場合が多い．このような構造を持つ大規模な整数計画問題では，一部の変数や制約条件からなる小規模な部分問題を逐次解くことにより，元の問題の解を効率良く求める Danzig–Wolfe 分解や Benders 分解などの分解法が古くから知られている．しかし，分解法を適用するにはどのような部分問題に分解するかあらかじめ指定する必要があるため，ブロック対角構造を持つことが明らかにわかる特殊な整数計画問題に対してしか分解法を適用できなかった．

文献 [15, 21] では，制約行列の行と列の並び順を入れ替えてブロック対角構造を抽出する問題をハイパーグラフ分割問題に帰着することで，一般の大規模な整数計画問題に対する Danzig–Wolfe 分解の適用を実現している（図 4）．具体的には，制約行列の各非ゼロ成分を頂点，各行・各列をそれらが含む非ゼロ成分を

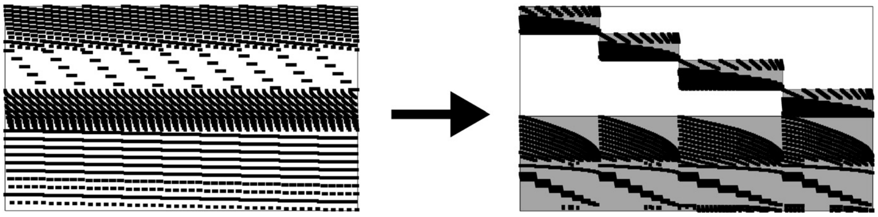

**図 4**　ブロック対角構造の抽出 [15]

張るハイパー辺とするハイパーグラフを定義し，あらかじめ決められた数の連結成分に分割するために，このハイパーグラフから各行に対応するハイパー辺を取り除く．しかし，ハイパーグラフを分割しても整数計画問題の制約行列における密な部分がブロックとして抽出されるだけで，抽出されたブロックがアルゴリズムの性能向上に役立つ特徴的な構造を持っているわけではない．また，ハイパーグラフを分割するに当たって適切な連結成分の数を決めることも容易ではなく，Danzig–Wolfe 分解の自動化の試みはまだ始まったばかりである．

## 5. パラメータの自動設定

多くのアルゴリズムはその動作を決定するいくつかのパラメータを持ち，パラメータの値の設定によって性能が大きく変化する．整数計画問題や制約充足問題などの汎用的な組合せ最適化問題は多様な現実問題を含むため，個別の入力データに対して最適なパラメータ値を見つけることは，利用者はもちろん組合せ最適化の専門家であっても多くの試行錯誤を必要とする困難な作業である．しかし，多くの利用者は個々の業務において類似した現実問題を繰返し解いている場合が多く，これらの入力データには共通した特徴が含まれることが期待できる．

文献 [4] では，過去の入力データの蓄積を訓練データとして，複数通りの異なるパラメータ値を設定してそれぞれ訓練データにアルゴリズムを適用し，ほかよりも明らかに性能の劣るパラメータ値があればランダムな値に再設定する手続きを繰り返す手法を提案しており，2 次割当問題や巡回セールスマン問題に対するメタヒューリスティクスのパラメータ設定に利用している．文献 [8, 9] では，現在のパラメータ値に対して微小な変更を加えて性能が改善すれば，新しいパラメータ値に置き換える手続きを繰り返す局所探索法に基づく手法を提案しており，充足可能性問題や整数計画問題に対するソルバーのパラメータ設定に利用している．ただし，これらの手法はパラメータ値を変更するたびに訓練データにアルゴリズムを適用するため，適切なパラメータ値の探索に多くの計算時間を必要とする．

実はパラメータの自動設定は必ずしも訓練データを必要とするわけではない．例えば，文献 [19] では，大規模な計算機上で整数計画ソルバーの 1 つである SCIP を並列実行しており，アルゴリズムの実行開始時に異なるパラメータ値を設定した複数のプロセスを並列実行することで，与えられた入力データに対する適切なパラメータ値を探索している．

パラメータの自動設定は，文献 [3, 7] に詳しくまとめられており，このほかにも応答反応曲面法など目的関数がブラックボックスとなっている最適化問題をシミュレーションで解く手法などが適用されている．

## 6. アルゴリズムの自動選択

巡回セールスマン問題やグラフ彩色問題など有名な組合せ最適化問題に対しては，これまで多くのアルゴリズムが提案されている．しかし，任意の入力データに対してほかよりも性能が良いアルゴリズムがそうあるわけではなく，実際には各アルゴリズムが得意とする入力データの傾向は異なる場合が多い．そこで，組合せ最適化問題に対して複数の異なるアルゴリズムを用意し，個別の入力データに対して性能が最も良いと期待できるアルゴリズムを自動選択する手続きが提案されている．

文献 [18] では，グラフ彩色問題に対して 6 個のアルゴリズムを用意し，教師付き学習を用いて個別の入力データに対して性能が最も良いと期待できるアルゴリズムを自動選択する手法を提案している．あらかじめ選んだ 78 種類の特徴を用いて 859 問の訓練データを学習し，152 問のテストデータのうち 107 問（70.39%）について性能が最も良いアルゴリズムを選択することに成功している．また，文献 [22] では，充足可能性問題に対して 7 個のアルゴリズムを用意し，教師付き学習を用いて各アルゴリズムに計算時間を適切に振り分けるポートフォリオに基づくアルゴリズムの自動選択を提案し，2007 年の充足可能性問題の国際競技会 (SAT

  

Competition) で優秀な成績を収めている．

最近の整数計画ソルバーは分枝カット法を基本戦略とする一方で，実際には内部で多くのアルゴリズムを適宜切り替えて探索している．例えば，整数計画ソルバーの1つであるSCIP3.0では，制約式に特有な前処理，変数の定義域の伝播，切除平面の生成，分枝変数の選択，実行可能解を生成するヒューリスティクスなど126個のアルゴリズムがプラグインとして実装されている [2]．これらのアルゴリズムの切り替えはパラメータで制御されているため，整数計画ソルバーでは，前節で紹介したパラメータの自動設定をアルゴリズムの自動選択と見なすこともできる．

## 7. 局所探索法における探索空間の削減

最後に，著者が集合分割問題に対して個別の入力データの解析に基づいて局所探索法の探索空間を削減した手法を紹介する．

集合分割問題は，与えられた集合のすべての要素をちょうど1回ずつ含む部分集合の組合せの中で，コストの総和が最小となるものを求める問題であり，すべての制約式が $\sum x_j = 1$ となる特殊な整数計画問題に定式化できる．集合分割問題は乗務員スケジューリング，配送計画問題，施設配置などの現実問題を応用に持つが，これらの現実問題を集合分割問題に定式化すると制約条件の数に対して変数の数が非常に多く，制約行列の行と列を並び替えてもブロック対角構造をうまく抽出できない場合が多い．

著者は，文献 [20] で，現在の解 $\boldsymbol{x}$ のうち1つの変数の値を $x_j = 0 \rightarrow 1$ もしくは $x_j = 1 \rightarrow 0$ と反転させる1反転近傍に基づく局所探索法を紹介した．実際に，集合分割問題に対して精度の良い近似解を求めるためには，複数の変数の値を同時に反転するより大きな近傍を持つ局所探索法を開発する必要がある．しかし，大規模な集合分割問題では2反転近傍であっても近傍内の候補解の数が非常に多くなるため，大きな近傍を探索する際には改善する見込みの高い候補解にのみ絞り込む必要がある．そこで，以下のような方法を用いて2反転近傍における候補解の絞り込みを実現した．

現在の解 $\boldsymbol{x}$ が1反転近傍の局所最適解であれば，2反転近傍では現在の解 $\boldsymbol{x}$ のうち2つの変数の値を $x_{j_1} = 0 \rightarrow 1$, $x_{j_2} = 1 \rightarrow 0$ と同時に反転させる場合のみを考慮すればよく，さらに，同じ制約条件に同時に現れる頻度の高い変数の組を同時に反転させると改善解が得られやすい．そこで，各変数 $x_{j_1}$ に対して同じ制約条件に同時に現れる頻度の高い変数 $x_{j_2}$（上位1%）を図5（左）示すリストに格納し，局所探索法では2反転近傍の候補解をこのリストに含まれる変数の組に絞り込むことで探索の効率化を実現する．また，このリストをグラフの隣接リストと見なせば図5（右）に示す変数間の関係を表すネットワークが得られる．

大規模な集合分割問題ではリストの生成に多くの計算時間を要するため，実際には，リストのすべての行を前処理で生成するのではなく，各変数 $x_{j_1}$ を含む2反転近傍を適用する際にリストの変数 $x_{j_1}$ に対応する行のみを遅延生成する．

整数計画問題のベンチマーク問題集MIPLIB [1, 13] に含まれる7つの集合分割問題に対して，整数計画ソルバーの1つであるCPLEX12.5.1と提案手法（局所探索法）をMacBookPro (Intel Core i7, 2.7 GHz) 上で実行した結果を表1に示す．いずれのアルゴリズムも1スレッドで実行し，計算時間の上限を問題例air04, air05では600秒に，ほかの問題例では3,600秒に設定した．表中のLP下界値は線形計画緩和問題の最適値を，リスト生成率は提案手法の実行終了時での変数の数に対する生成されたリストの行数の割合を，CPLEX, 提案手法はそれぞれ実行終了時におけるCPLEX12.5.1と提案手法の暫定解（実行可能解）の目的関数値を表す．また，アスタリスク (*) の印は厳密な最適値が得られたことを表す．

表1から見てとれるように，ds-bigやivu06-bigなどの大規模な問題例に対して，分枝カット法を基本戦略とする整数計画ソルバーでは限られた時間内に精度の良い近似解を求めることは容易ではない[1]．提案手法では，2反転近傍の候補解となる変数の組を全体の1%に絞り込んでいるうえに，ds-bigやivu06-bigなどの大規模な問題例ではリストの生成率が非常に低いことから，2反転近傍内のごくわずかな候補解を探索するだけで精度の良い近似解を求めていることがわかる．一方で，air04やair05などの小規模な問題例では，整数計画ソルバーで厳密な最適解が求まる一方で，提案手法では最適解に近い解を求めることは容易ではない．また，集合分割問題は実行可能解を1つ求めること自体が非常に難しい問題であるため，アルゴリズムの仕様やパラメータの値を少し変更するだけで結果が大きく変わる．例えば，問題例ds-big, ivu06-bigでは提案手法を実装する際の試行錯誤でそれぞれ1053.06,

1 ちなみに，問題例dsでは大規模な計算機上で整数計画ソルバーを長時間並列実行することで最適値93.52が得られている [19].

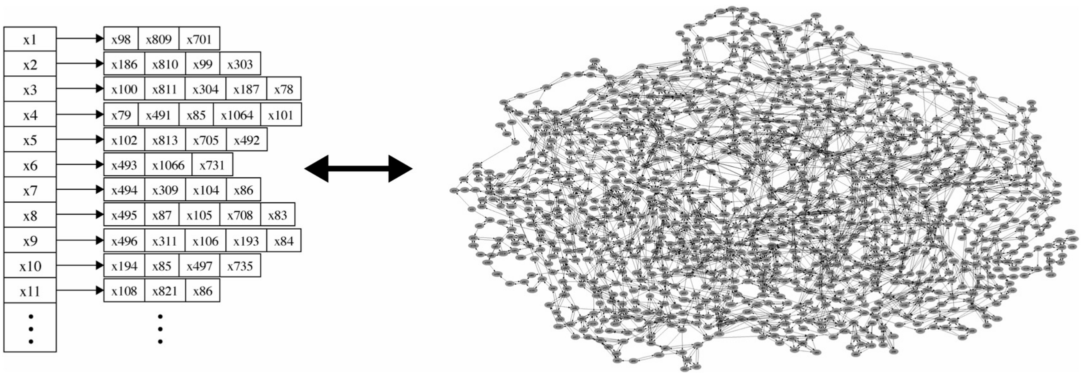


**図 5**　同じ制約条件に同時に現れる頻度の高い変数の組を格納したリスト（左）と変数間の関係を表すネットワーク（右）

**表 1**　MIPLIB の問題例に対する数値実験の結果

| 問題例 | 制約 | 変数 | 密度 | LP 下界値 | CPLEX | 提案手法 | リスト生成率 |
|---|---|---|---|---|---|---|---|
| air04 | 823 | 8904 | 1.00% | 55535.43 | ∗56137 | 60635 | 62.55% |
| air05 | 426 | 7195 | 1.70% | 25877.60 | ∗26374 | 27565 | 65.76% |
| t1717 | 551 | 73,885 | 0.80% | 134531.02 | 188704 | 202276 | 35.12% |
| t1722 | 338 | 36,630 | 1.08% | 98815.40 | 122532 | 122371 | 38.59% |
| ds | 656 | 67,732 | 2.31% | 57.23 | 384.25 | 235.13 | 34.00% |
| ds-big | 1042 | 174,997 | 2.54% | 86.82 | 1771.07 | 1227.96 | 5.92% |
| ivu06-big | 1177 | 2,277,736 | 0.87% | 135.42 | 672.49 | 188.58 | 0.21% |

176.61 の暫定値が得られており，まだ改善の余地は大きいと予想される．

## 8.　おわりに

汎用的な組合せ最適化問題では，任意の入力データに対して精度の良い近似解を求めるアルゴリズムを設計することは非常に困難である．しかし，現実問題を定式化して得られる個別の入力データは無秩序でまったく構造を持たないわけではなく，特徴的な構造を持つ部分から構成されている場合が多い．また，あらかじめアルゴリズムの仕様をすべて決定する必要はなく，実行時にアルゴリズムの一部の仕様を変更すれば，個別の入力データから抽出した情報をアルゴリズムの性能向上に利用することは可能である．

本稿では，個別の入力データからアルゴリズムの性能向上に役立つ情報を発見し，アルゴリズムの設定や構成を自動的に決定する手法をいくつか紹介した．実世界から収集された大規模データに基づく現実問題は十分に整理されていない場合が多く，最近の情報爆発による大規模・複雑化も相まって，組合せ最適化の専門家であっても問題構造を把握することが困難な事例が急増している．本稿で紹介した手法はこのような大規模かつ複雑な現実問題を解決するための重要な基盤技術となることが期待できる．

**謝辞**　本研究は，科学技術振興機構 (JST) 戦略的創造研究推進事業さきがけ（研究領域「知の創生と情報社会」）の一環として行われている．

### 参考文献

[1] T. Achterberg, T. Koch and A. Martin, “MIPLIB2003,” *Operations Research Letters*, **34**, 361–372, 2006.
[2] T. Berthold, A. M. Gleixner, S. Heinz, T. Koch and Y. Shinano, “Solving mixed integer linear and nonlinear problems using the SCIP Optimization Suite,” Technical Report ZIB-Report 12–24, Zuse Institute Berlin, 2012.
[3] R. Battiti, M. Brunato and F. Mascia, *Reactive Search and Intelligent Optimization*, Springer, 2008.
[4] M. Birattari, *Tuning Metaheuristics: A Machine Learning Perspective*, Springer, 2005.
[5] 藤江哲也，“最近の混合整数計画ソルバーの進展について，”オペレーションズ・リサーチ，**56**, 263–268, 2011.
[6] M. Gendreau and J. Y. Potvin (eds.), *Handbook of Metaheuristics* (2nd ed.), Springer, 2010.
[7] H. H. Hoos, “Automated algorithm configuration and parameter tuning,” in: Y. H. Hamadi, E. Monfroy and F. Saubion (eds.), *Autonomous Search*, Springer, 37–71, 2011.
[8] F. Hutter, H. H. Hoos and K. Leyton-Brown, “ParamILS: An automatic algorithm configuration framework,” *Journal of Artificial Intelligence Research*, **36**, 267–306, 2009.
[9] F. Hutter, H. H. Hoos and K. Leyton-Brown, “Automatic configuration of mixed integer programming solvers,” *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization*

  

*Problems* (CPAIOR2010), *Lecture Note in Computer Science*, **6140**, 186–202, 2010.
[10] 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, **50**, 229–232, 2005.
[11] 茨木俊秀,“問題解決エンジンとしてのメタヒューリスティクス・アルゴリズム,”第 19 回 RAMP シンポジウム論文集, 117–130, 2007.
[12] T. Ibaraki, “A personal perspective on problem solving by general purpose solvers,” *International Transactions in Operational Research*, **17**, 303–315, 2010.
[13] T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy and K. Wolter, “MIPLIB2010 Mixed integer programming library version 5,” *Mathematical Programming Computation*, **3**, 103–163, 2011.
[14] 久保幹雄, J. P. ペドロソ, メタヒューリスティクスの数理, 共立出版, 2009.
[15] M. E. Lübbecke, “Automatic decomposition and branch-and-price—A status report,” *Experimental Algorithms* (SEA2012), *Lecture Note in Computer Science*, **7276**, 1–8, 2012.
[16] 宮代隆平, 松井知己,“ここまで解ける整数計画,”システム/制御/情報, **50**, 363–368, 2006.
[17] 宮代隆平,“ここまで解ける整数計画—近年の発展—,”第 20 回 RAMP シンポジウム論文集, 1–21, 2008.
[18] M. Nysret and M. Schwengerer, “Algorithm selection for the graph coloring problem,” *Learning and Intelligent OptimizatioN Conference* (LION7), *Lecture Notes in Computer Science*, 7997, 389–403, 2013.
[19] 品野勇治, T. Achterberg, T. Berthold, S. Heinz, T. Koch, S. Vigerske and M. Winkler,“制約整数計画ソルバ SCIP の並列化,”統計数理, **61**, 47–78, 2013.
[20] 梅谷俊治, 柳浦睦憲,“メタヒューリスティクス事始め：まずは局所探索法から,”オペレーションズ・リサーチ, **58**, 689–694, 2013.
[21] J. Wang and T. Ralphs, “Computational experience with hypergraph-based methods for automatic decomposition in discrete optimization,” Technical Report, Department of Industrial and Systems Engineering, Lehigh University, USA, 2012.
[22] L. Xu, F. Hutter, H. H. Hoos and K. Leyton-Brown, “SATzilla: Portfolio-based algorithm selection for SAT,” *Journal of Artificial Intelligence Research*, **32**, 565–606, 2008.
[23] 柳浦睦憲, 茨木俊秀, 組合せ最適化—メタ戦略を中心として—, 朝倉書店, 2001.