

分枝限定法における分枝戦略選択のための 計算過程の可視化

宮村(中村)浩子^{†1} 品野勇治^{†2}
宮代隆平^{†2} 斎藤隆文^{†2}

分枝限定法を用いて整数計画問題を解く際に、どのような分枝戦略を選択するかは重要な問題である。分枝戦略の良し悪しは、生成される子問題の数、分枝限定木の深さ、総計算時間などに大きな影響を与える。しかしながら、大規模な数理計画問題では、分枝限定木の生成過程における出力は大量のログデータとなってしまう、それぞれの分枝戦略がどのように影響を与えているのか直感的な把握が難しい。そこで本研究では、分枝限定木の生長過程を可視化するシステムを提案する。本システムにより、分枝戦略の違いが子問題の生成過程に及ぼす影響を視覚的にとらえることができる。

Visualization of Runtime Behavior of Branch-and-bound Algorithms

HIROKO NAKAMURA MIYAMURA,^{†1} YUJI SHINANO,^{†2}
RYUHEI MIYASHIRO^{†2} and TAKAFUMI SAITO^{†2}

In branch-and-bound algorithms for integer programming, runtime behavior of the algorithms depends much on its branching strategy. However, from a huge computation log of a large program, it is difficult to explore key factors for effective branching. To analyze which factor of branching strategy is essential, we develop a system for visualization of growing process of a large branch-and-bound tree. The proposed system provides intuitive understanding how branching strategy affects branch-and-bound process.

^{†1} 日本原子力研究開発機構システム計算科学センター
Center for Science and e-Systems, Japan Atomic Energy Agency

^{†2} 東京農工大学
Tokyo University of Agriculture and Technology

1. はじめに

分枝限定法を用いて整数計画問題を解く際には、どのような分枝戦略を選ぶかにより、子問題の生成過程およびトータルの計算時間に大きな違いが出る。したがって、整数計画問題を高速に解くためには、子問題の生成過程を観察することにより適切な分枝戦略を選ぶことが重要である。子問題の生成過程はログデータにより得られるが、概して解析対象としたい整数計画問題は解くことが困難なものであり、生成されるログは大量の数値テキストデータとなる。しかし、大量の数値テキストであるログデータから、分枝戦略が子問題の生成過程に与えている影響を直感的に把握することは困難である。そこで、分枝限定木の形状および生成過程の様子を視覚的にとらえるために、分枝限定法の計算過程を可視化するシステム VAULT (Visualization and Analysis Utility for Large Tree structure) を開発する。

分枝限定法の計算過程を可視化するシステムとして、過去には Sapal-BB¹⁾ や Explanation of Branch-and-Cut Tree Pictures²⁾ が提案された。これらは、分枝限定木の評価関数値の変動を木の形で示すが、大規模なログデータに適用すると、描画要素が混み合い、個々の要素を認識しにくくなる。また、大規模階層データを可視化するための既存手法として、Tree-Maps³⁾、Hyperbolic Tree⁴⁾、ConeTrees⁵⁾ などがあげられるが、各ノードに複数の数値が与えられている分枝限定木の表現には不向きである。

そこで、大規模かつ多変数階層データである分枝限定木を可視化するために、空間効率を考慮したノード配置、簡略化木構造表現を備えた可視化システムを提案する。また、評価関数値(下界値)¹⁾、暫定値(上界値)など、複数の数値データの時間的変動も同時に示す。本システムを利用して分枝限定木の生成過程を観察することで、計算過程の傾向の把握・予測、異なる動作の発見、分枝戦略が影響を与える範囲の考察ができ、分枝変数、探索順序、緩和問題の選択などアルゴリズムの改良につながる事が期待できる。また、商用ソルバでの分枝限定法の実行時におけるパラメータ設定の決定にも有効である。

2. 背景と目的

本章では、分枝限定法のログデータ、およびその可視化による分枝戦略改良の取り組み、さらに大規模階層データの可視化手法を紹介する。

*1 本論文は、一般性を失うことなく、解く対象である問題は最小化問題を仮定して記述する。本論文の記述における評価関数値は、一般的には線形緩和問題の最適値によって与えられる下界値である。

2.1 分枝限定法およびログデータ

分枝限定法とは、整数計画問題を解くための一般的なフレームワークであり、整数計画問題を解くほとんどすべての商用/非商用ソフトが分枝限定法を用いている^{*1}。

ただ、ひとくちに分枝限定法といっても、絶対的な分枝の方法があるわけではなく、これまでにさまざまな分枝戦略が提案されている。分枝戦略の良し悪しにより、分枝限定法のスピードは大幅に変化する。さらに、すべての問題に共通して効果的な分枝戦略というものではなく、問題の性質に依存するところが大きい。一般に、大規模で難しい問題を解きたいときは、性質が同じで入力サイズが小さい問題を作り、複数の分枝戦略を試してみても効果のあるものを選ぶなどのチューニングが施される。

分枝戦略のチューニングには、分枝限定法のログからチューニングに必要な情報を解析する必要がある。これらの情報には「生成ノード数」「残存ノード数」「暫定解の出現頻度」「最良下界値の改善具合」などがあるが、これらは単一のノードにおけるデータを見るよりも、分枝限定法の進行にともないどのように値が変化していったかがより重要である。しかし問題の性質によっては、分枝限定法を実行した結果、巨大な分枝木が生成され、上記のデータの相関を直感的に把握することが困難になる。

これらの問題点を解決するために、過去にいくつかの研究が提案されてきた。次節では、分枝限定法に関するもの以外も含め、巨大な階層データの可視化における先行研究について述べる。

2.2 階層データ可視化システムの先行研究

並列分枝限定法の計算過程を視覚的にとらえるために開発された Sapal-BB¹⁾ は、並列分枝限定法の挙動を調べるためのシステムであり、シミュレート実行と同時に、生成される分枝限定木を動的に可視化する。このシステムを用いることで、逐一変化する分枝木の生成の様子を観察できる。Sapal-BB が分枝限定木の生成プロセスの観察に有効であるのに対して、本研究は、分枝戦略の影響などを解析するために、分枝限定法を終了するまで走らせたような、大規模ログデータを効果的に観察することを目指す。そのためには、大規模な数値データを表示することが必要である。大規模データを扱った研究としては、大規模な巡回セールスマン問題を分枝カット法により解いた際の分枝限定木を可視化する Explanation of Branch-and-Cut Tree Pictures²⁾ があげられる。この研究では、各ノードの評価関数値を木構造の高さ方向の座標と対応付けてノードを配置することで、値が大きく変化する枝を発

見できる。これによって、評価関数値の変動に大きな影響を与えている分枝変数を知ることができる。しかし、やはり大規模なデータに対しては、ノードが混雑し解析が困難になる。

分枝限定木のような大規模階層データを可視化するための手法は、パイオインフォマティクス分野を中心に、いくつかの取り組みがなされている⁶⁾⁻⁸⁾。たとえば、同じ画素に重ね描きする際、描画順を操作し重要な情報が隠れないようにする TreeJuxtaposer⁶⁾ が提案された。これはハイライト表示とよばれ、ユーザが指定した領域は他の領域によって背後に隠れないようにする。しかし、ユーザが指定するハイライト領域もまた大量である場合にはすべての要素を表示することが難しい。大規模階層データを簡略化によって表現する方式では、階層データのフラクタル性を利用した Fractal Views⁹⁾ があげられる。この手法は、大局的特徴と局所的特徴の繰返し構造を利用して表示するノード数を調整している。これはノード数を一定範囲内に抑えられることから、大規模データを効率的に表示できる。だが、表示されていない情報をユーザが見落とす恐れがある。

描画空間を効率的に利用する方法としては、木構造の表現方式自体を工夫する H-tree, Radial layout, Balloon layout¹⁰⁾、また、Radial layout で作成したツリーグラフを非ユークリッド空間に投影する手法⁴⁾ が提案されている。しかし、これらの手法においても大規模な階層データに対してノードが密集してしまうという問題は残る。階層データのリンクの表現形式を工夫する研究も行われている。Tree-Maps³⁾ は、ノード間の階層関係を入れ子で表現することで、直接的なリンクの表示を略している。これは限られた空間に多数のノードを配置できるという利点はあるが、ノード間の関係やノードの深さを直感的にとらえにくいだけでなく、上位階層のノードに与えられた情報は示せない。同様の入れ子による階層データの可視化手法として Data Jewelry Box¹¹⁾ が提案されたが、Tree-Maps と同様にノード間の関係や階層の深さの直感的な認識が難しい。

データをユーザの操作によって選択的に表示する手法も提案されている。たとえば、3次元空間に階層データを配置する ConeTrees⁵⁾ では、ユーザは必要な階層以下のノードを表示しないで観察することができる。また、Radial layout を採用し、リンク情報を領域の隣接状態で示すことで空間効率を高めた Information Slices¹²⁾ では、選択したノードの下位階層を新たに新しいグラフとして作成する。

これら大規模階層データを可視化する先行研究では、ノードやリンクを表現する描画要素の形状や配置を工夫することで空間効率の良い階層データの提示を実現してきた。また、表示する要素を制限することで大規模データに対処してきた。我々は、これらのグラフ表現手法を拡張し、分枝限定法の計算過程を可視化することに特化した可視化システムを提案す

*1 本論文では、整数計画問題および分枝限定法の詳細な説明は省く。文献 17) を参照のこと。

る．この可視化システムでは，大規模データを効率的に配置すること，変動する数値データを木構造データと連動して観察できるように示すことを実現する．また，分枝限定法で重要な情報を残しつつ簡略表現を実現することを目指す．

3. 階層データの木構造化

分枝限定法の計算過程データから木構造を構築するためには，親子関係を認識し，木の形で表現しなくてはならない．しかしながら，本研究で扱うデータは大規模であり，木構造にした場合は大量のノードを含むため，効率的な可視化のためには数多くの工夫が必要である．本章では，データの格納，グラフ生成のための描画要素配置，多変量データの提示，さらに大規模階層データを可視化するための段階的表示機能について説明する．なお，本システムは ILOG CPLEX¹³⁾ により，整数計画問題を解いた際に生成されるログデータ，およびそれと同等の情報を持つログデータを入力データとして使用している．ログデータに記載されている代表的な情報としては，以下のものがある（ほかにもいくつかあるが本論文では省略する）．

- 現在のノード ID
- 親ノードの ID
- 当該ノードのノード深さ
- 当該ノードでの残存ノード数
- 当該ノードでの線形緩和値
- 現時点での最良暫定解の値
- 現時点での最良下界値

3.1 データ格納

分枝限定法の計算過程データから親子関係を構築し，格納する．まず，入力データにはそれぞれのノードの親ノード ID が記録されている．データ入力時には，各ノードが読み込まれるたびに親ノード ID を記録するとともに親ノードに対して読み込んだノードの ID を子ノードリストに加える．さらに，子ノード数のカウントを 1 加える．木構造を構築するためには，親ノード ID，もしくは子ノードリストのどちらか一方の情報があれば可能である．しかし，ここでは可視化結果に対する対話的な操作の実現を見込んで，双方の情報を保持させることとする．

次に，交差なくリンクを提示するために，各ノードに与える領域を求める．ここでは，自身より下の階層に存在する葉ノードの個数を数え，その個数分領域を確保する方式をとる

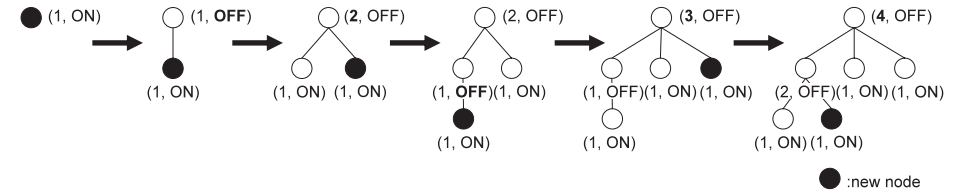


図 1 木構造グラフのデータ格納

Fig. 1 Storing a hierarchical datasets.

(3.2 節)．葉ノードの個数 (leaf_no) は以下の手順でデータを読み込むたびに更新される．

- 1: 新しいノードが読み込まれたら，leaf_no = 1 に設定し，葉ノードであるかどうかのフラグ leaf_flag を ON に設定する．
- 2: 親ノードの leaf_flag が ON である場合は OFF に変更，OFF である場合はルートノードまでの経路をたどり，経路上のノードの leaf_no を 1 加算する．

以上の処理をデータ読み込み時に行うことで，各ノードには自身以下の階層に存在する葉ノードの個数である leaf_no を記録できる．データを読み込むたびに更新される様子を図 1 に示す．図中にノードに付随するカッコ内は，leaf_no, leaf_flag をそれぞれ表している．

入力データはルートノードから順に木の成長に沿う．そのため，新しく読み込まれるノードはつねに葉ノードであり，その親ノード ID はログデータに記録されているため $O(1)$ で親ノードにアクセスできる．親ノードを順々にルートノードまでの経路をたどるが，そのオーダは木の全ノード数を n とした場合， $O(n)$ で抑えられる．ただし，この処理はデータ読み込み時に 1 回行われるだけであるため，描画など，可視化結果を観察する際には必要ない．

3.2 大量グラフ要素の配置

大規模階層データを限られた空間に表示するには，空間を効率的に利用したノードの配置や，クラッタリングを抑えたリンクの提示が必要である．本システムでは，Top-down layout, Radial layout を採用し，それぞれに対する効果的なノード配置，リンク提示を検討する．

まず，ノードの配置では，自身以下の階層に存在する葉ノードの個数 (leaf_no) 分だけ領域を確保しながら配置する．Top-down layout では， x 軸方向の表示領域幅を木全体の葉ノード数で割った値を 1 つの葉ノードの領域とする．この値をもとに，各ノードは leaf_no 分だけ x 軸方向に領域を確保しながら配置する (図 2(a))．親子関係をより明確に示すた

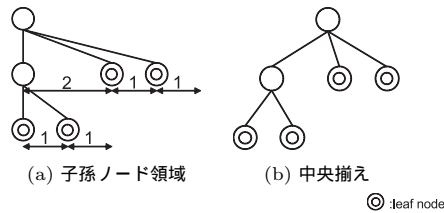


図2 ノード配置

Fig. 2 Node arrangement: (a) The area of the descendant node, (b) Center arrangement.

めには、最下層から順に子ノードの中心に親ノードを再配置する(図2(b)).以降、子ノードの中心に親ノードを再配置したものを「中央揃え」、再配置前のものを「始点揃え」とよぶ。ノードの中央揃えでは、親子関係を持つノード間のユークリッド距離が近くなり、リンクの把握を助ける。また、階層の深さは y 座標で表す。そのため、ルートノードは最上部に置かれ、深い階層のノードほど下部に配置される。ノードの配置では、データ入力時に格納した葉ノードの個数 (leaf_no), 階層の深さから x, y 座標ともに1次式で求めるため、そのオーダは木の全ノード数を n とした場合、 $O(n)$ で抑えられる。

Radial layout では、表示角を木全体の葉ノード数で割った値を1つの葉ノードの領域とする。本システムでは、表示角を 360° としている。ここでは領域は角度で与えられ、各ノードは leaf_no 分だけ中心角領域を確保しながら配置する。再配置では、中心からの角度が子ノードの中心になるように親ノードを移動する。階層の深さは中心からの距離で表現する。ここではルートノードは表示領域の中心に存在する。

リンクの提示には、「直線」によってリンクを提示する方法(図3(a),(b),(g),(h)), 階層に沿った線(弧)で x 座標(角度)を移動するような「折れ線」でリンクを提示する方法(図3(c),(d),(i),(j)), 線のようなオブジェクトを表示しないで、領域の「塗りつぶし」からリンクを提示する方法(図3(e),(f),(k),(l))を提供している。

まず、直線によってリンクを提示する方法では、親子関係を示すノード間を直線で結ぶ。この方法では、親子関係の直感的理解を促すが、大規模データに対してはクラッタリングが発生する。Radial layout では一般的にクラッタリングの発生を抑えられるが、全体構造が把握し難い。次に、階層に沿った線で移動する折れ線提示について考える。この方法は、直線の混雑が緩和されるだけでなく、同階層のノードを明確にできるため階層の深さに着目した観察に適している。また、リンクを示す線が整理されるため、親子関係が把握しやすく、

クラッタリングの発生も抑えられる。最後に、領域の塗りつぶしによる親子関係の提示を検討する。この方法では、親子、兄弟ノードで境界を共有しているため、塗りつぶす色の変化が大きいノードの発見が容易である。ユーザはこれらのリンク提示方法を目的に応じて選択して利用できる。

なお、図3は分枝限定法によって計算された分枝限定木の可視化結果である。ノードに与えられた色は、評価関数値を表している。

3.3 多変量データの提示

ノードが持つ数値の表現には、高さ方向である z 座標、ノードの色、背景領域色を用いることができる。また、色については、色相、明度、彩度に表現したい数値を割り当てられる。図4(a)に、木構造の背景領域を塗りつぶす際の明度に計算ステップ数を割り当てた結果を示す。この結果から、木構造のどの部分が早い段階で計算されたのかを木の形状、目的関数値の変動とともに認識できる。このように複数の属性値をさまざまな座標や色などに割り当てることで、木構造の形状とともに数値の変動や相関を把握できる。また、 z 座標に評価関数値を割り当てた結果を図4(b)に示す。これは $x-z$ 平面投影した結果である。

3.4 木構造の段階的表示

表示空間に構成要素が収まりきらないほど大規模な分枝限定木を観察することを考える。一般的には、木構造が配置されている領域を表示可能な大きさに分割し、それぞれの領域ごとに独立の部分木を表示する。しかし、この方法では領域間の木構造の連続性が保たれないだけでなく、全体構造をとらえにくいという問題がある。ConeTrees⁵⁾では、ある階層以下のノードをユーザの選択によって表示しないことで大規模データを簡略化している。本システムでは、簡略化された木構造でも、元の木構造の大局的な特徴を示すために、ポリゴンメッシュの形状特徴を保持しながら簡略化メッシュを作成する段階的メッシュ法に着目する¹⁴⁾。

大規模階層データから段階的木構造を作成するために、稜線縮退化操作をリンクに対して適用する(図5)。階層データは、見た目の類似性とデータの特性による類似性が異なるため、形状特徴を保持することは効果的でない。そこで、リンク縮退の際の評価関数をユーザが選択できるようにすることでデータの特性を考慮した類似性を保つ。本システムでは、簡略化に使用する特徴量にノード間の評価関数値の変動量、子ノード数の2つを用意し、ユーザインタフェースによるボタンで選択できる。図5では、評価関数値の変動量を評価関数に使用し、評価関数値の変動が小さいリンクを縮退化操作によって削除する。また、暫定解が得られたノードは非常に重要であるため、削除の対象から外す。なお、削除されたノー

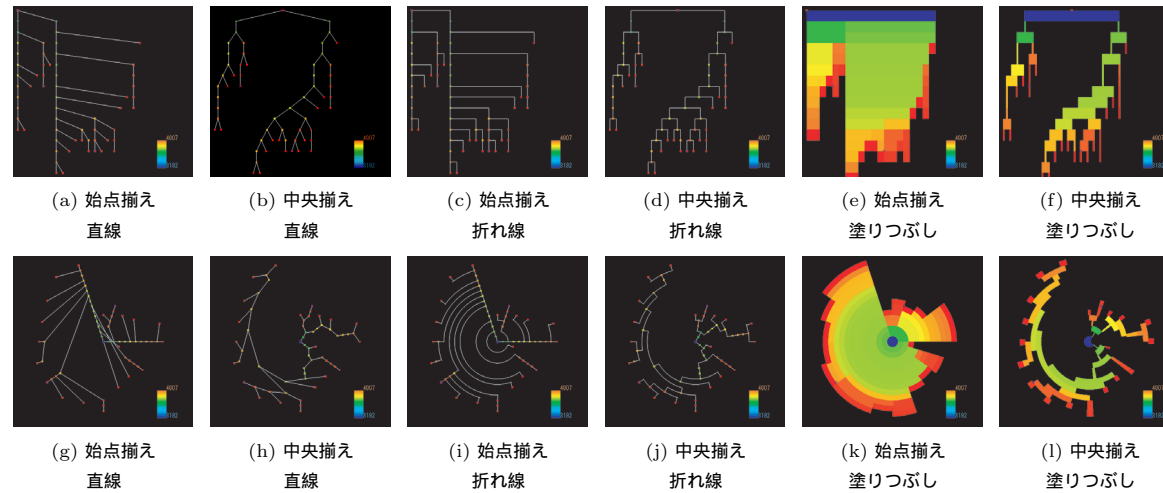


図3 ノード配置，リンク提示のパリエーションによる木構造表示；上段：直交座標系，下段：極座標系
 Fig. 3 Tree Graph Variation: “Starting point arrangement” style are (a), (c), (e), (g), (i), and (k) and “Center arrangement” style are (b), (d), (f), (h), (j), and (l). “Straight line” style are (a), (b), (g), and (h); “Straight line with corner” style are (c), (d), (i), and (j); and “Drawing” style are (e), (f), (k), and (l).

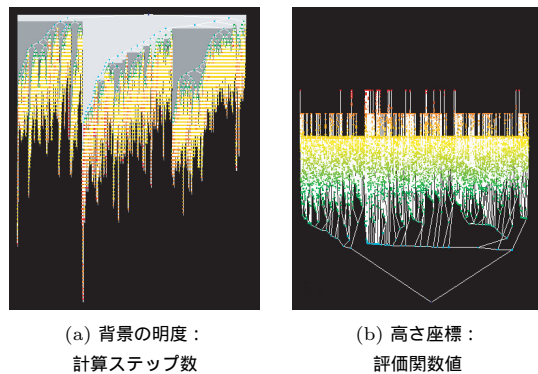


図4 多変量データの可視化例
 Fig. 4 Example of multivariate visualization: (a) The calculation timing step is mapped the lightness of background, (b) The objective value is allocated in the vertical axis.

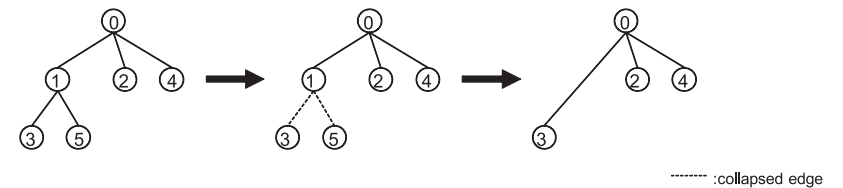


図5 リンク縮退化操作
 Fig. 5 Link collapse operation.

ドは明度を落として表示したり，縮退先のノードの大きさを変えて表示したりすることで，ユーザに情報を提示する．

簡略化に際しては，ユーザが選択的に復元できるように，簡略化過程を記録する．図6に簡略化による木構造データ（Data）の更新と簡略化過程データ（Simplified Data）を示す．木構造データには，ノードIDと親ノードIDを記録し，簡略化過程データでは，縮退ノードID，縮退先のノードID，これら2つのノードの関係（P：親ノード，C：子ノード）を

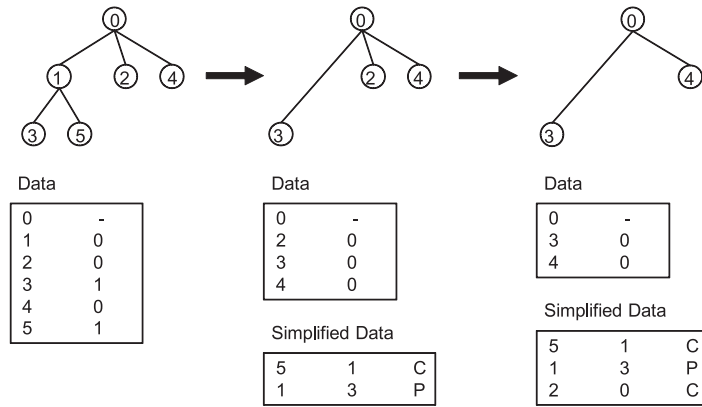
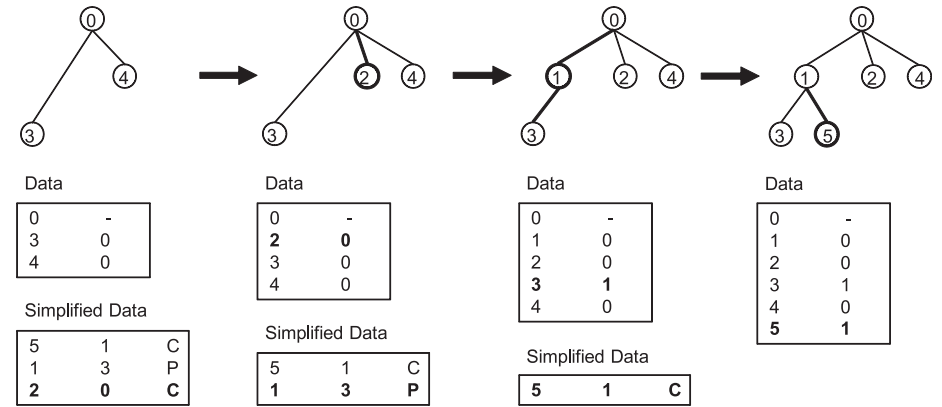


図 6 木構造データ更新

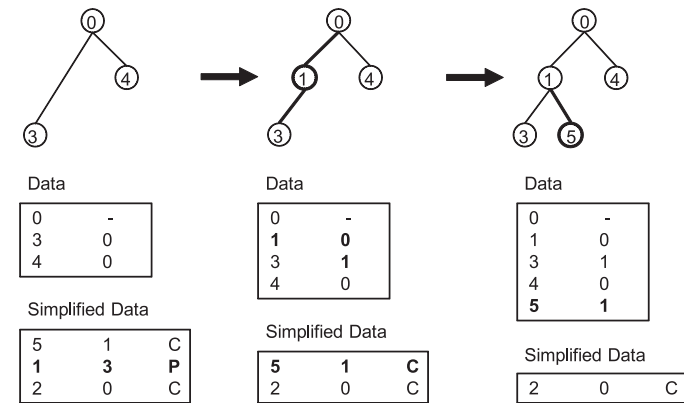
Fig. 6 Redefining a hierarchical datasets.

記録している．なお，実際の木構造データでは他に子ノード数，子ノード ID，目的関数値，上界値，暫定解など複数の数値データが格納されているが，ここでは省略する．この簡略化データを作成する場合には，まずリンクに対する特徴量の評価が必要である．たとえば，親ノードと子ノードの評価関数値の変動量を特徴量とする場合，これら 2 ノードの評価関数値の差を求める．この際の計算量は，簡略化によって削除するノード数を n_c としたとき， $O(n_c)$ となる．次に，これらの特徴量を基準にソートする．その際の計算量は $O(n_c \log n_c)$ となる．ここで，ソートしたリンクから，葉ノードを端点に持ち，かつ暫定解でないリンクを縮退させる．ただし，ユーザインターフェースによって特徴量や簡略化する割合を変えながら対話的に簡略化過程データを作成する場合にはそのつど計算量が変化する．

この簡略化過程データを用いて，簡略化された木構造を元の精度の木構造に復元する様子を図 7 に示す．全体的に復元するときは，記録した簡略化過程データを簡略化とは逆順に，ユーザが指定した復元率に達するまで復元する．選択的に復元するときは，ユーザが表示されたノードをマウスクリックで選択し，選択されたノードに縮退したノードを復元し，復元したノードに縮退したノードを次に復元する．この処理を繰り返すことで選択的な復元を実現する．復元に際しては，簡略化過程データを逆順に走査するため，復元するノード数を n_s とすると $O(n_s)$ で実現できる．なお，兄弟ノードは計算ステップ数（ノード ID）によって計算された順に配置し直すことができ，兄弟ノードが簡略化操作で入れ替わらないよ



(a) 全体的復元



(b) 選択的復元

図 7 木構造データの復元

Fig. 7 Restructuring a hierarchical datasets: (a) All the datasets, (b) A portion of datasets.

うにできる．

3.5 並列環境での実行結果

筆者らが今回開発したシステムは，ユーザがクライアント PC でログデータを解析することを前提としている．しかし，整数計画問題の計算に関しては並列環境でも行われることを考慮しており，サブツリーに分割されたデータにも対応する．まず，サブツリーをノード

として木構造を生成する．その後、各サブツリー内で木構造を生成する．このとき、各サブツリーでは、サブツリー内の葉ノード数を保持する．この葉ノード数を、サブツリーの親子関係から上位階層に送ることで各ノードの葉ノード数を更新する．

今後、一度に読み込めないような大規模ログデータが生成された際には、まずサブツリーを表示し、次にユーザが指定したサブツリーの情報を読み込み、表示するなど工夫が必要である．

4. 大規模階層データ可視化システム VAULT

大規模階層データ可視化システム VAULT は、木構造などの可視化結果を表示する領域と対話的操作のための操作パネルを持つ．ユーザインタフェースの構築には、GUI ライブラリ ADV_AutoGL¹⁵⁾ を使用する．また、分枝限定法を用いて整数計画問題を解いたときの計算過程データの観察のために、表示ウィンドウは以下の領域に分け、さまざまな情報を提示する．

- 木構造表示領域 (図 8 左上)
- 計算ステップ表示領域 (図 8 左下)
- グラフ表示領域 (図 8 右下)
- ノードデータ表示領域 (図 8 右上)

木構造表示領域には、前章で紹介した木構造グラフを表示する．ここで、分枝限定法の計算過程の観察で重要な、暫定解が得られたノードにマーク付けをし、さらに分枝戦略が影響を与える分枝変数を特定するために、特定ノードと同じ分枝変数のノードにマーク付けをする．計算ステップ表示領域では、各ノードが計算されたタイミングを y 座標にとり、プロットする．このとき、木構造との対応をとるため x 座標を合わせる．グラフ表示領域では、計算ステップ表示領域の y 軸に合わせて、プール内ノード数、下界値、暫定解が変化の様子を表示する．最後に、ノードデータ表示領域では、ユーザが選択したノードの詳細情報を数値、テキストで表示する．ノードの選択は、木構造表示領域、または計算ステップ表示領域に表示されているノードをクリックすることでできる．計算ステップ表示領域、グラフ表示領域では共通の時間軸を持ち、赤線は暫定解が見付かったノードから時間軸に垂直に引いたものであり、暫定解であるノード、およびそのときの残存ノード数、下界値、暫定解を把握することができる．

ユーザは、木構造表示領域の分枝限定木でノードの木構造や評価関数値の分布を把握し、計算ステップ表示領域、グラフ表示領域で分枝限定法の計算過程の時間変化を観察できる．

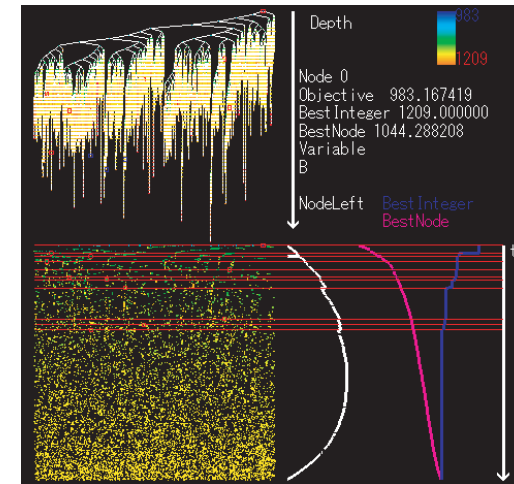


図 8 VAULT による分枝限定木の可視化
Fig. 8 VAULT for branch-and-bound tree.

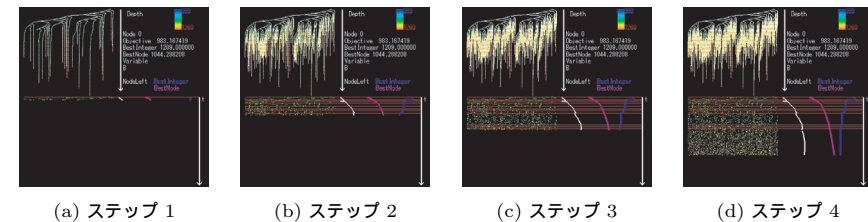


図 9 分枝限定木の生長の様子
Fig. 9 Growing process of a branch-and-bound tree.

これらの可視化結果を観察したうえで、興味あるノードに対しては、ノードデータ表示領域で詳細情報を得られる．

また、本システムは計算ステップに応じて表示する機能も持つ．表示ステップ幅を操作パネルに入力し、ボタンをクリックすることで計算過程を進めたり戻したりしながら観察できる．図 9 に計算ステップに応じてアニメーション表示した結果を示す．

5. 実行例

図 10 に、分枝戦略を変えて整数計画問題を解いた計算過程の可視化結果を示す。可視化実験には、Windows XP (CPU: Intel Pentium M 1.60 GHz, RAM: 2 GB RAM) を用いた。ここで扱っている問題は、整数計画問題のベンチマーク集 MIPLIB 2003¹⁶⁾ に含まれる mas76 (変数 151 個, 制約式 12 本) である。この問題を、4 つの分枝戦略を用いて解いた結果が図 10 (a)~(d) である。分枝戦略の細かい設定は省くが、(a) が許容解を求めることに重点をおいた戦略、(d) が下界値を改善することに重点をおいた戦略、(b) は標準的な設定、(c) は (b) と (d) の中間に位置する分枝戦略である。分枝限定木のノード数はそれぞれ (a) 973,238, (b) 375,730, (c) 553,636, (d) 747,954 となった。ノードの色は目的関数値 (38,894 ~ 40,560) をマッピングし、Radial layout, 始点揃え、塗りつぶしでグラフを表示した。これらの結果から、解が求まるまでの過程の相違を視覚的にとらえられる。図からも分かるように、(a) は整数解を求めることに重点をおいているため、早い段階で目的関数値が大きくなっている。これに対して、(d) では整数解を求めることに重点をおかないため、値の変動が小さい。バランス型の (b) では、先にいくつかの適した整数解を求めるため、何本かのリンクが深い階層まで伸びている。

また、図 10 (a) を Top-down layout, 中央揃え、直線でグラフ表示し、高さ方向に目的関数値をマッピングした結果 (図 11 (a)) の段階的な詳細度表現を示す (図 11 (b)~(d))。なお、この木の元データサイズは、ノード数 973,238, ログデータサイズ 108,825 KB である。ノード数を 1/6,000 まで簡略化した木構造グラフで、削除したノードの情報を明度を落として表示したものと、ノードサイズで表示したものをそれぞれ図 11 (e), (f) に示す。これらの結果を観察しながら、ユーザは興味ある領域を図 11 (g) のように選択し、復元できる。部分的に復元した木構造グラフは表示ノードだけを対象として x 座標を正規化することで、復元領域部分を拡大して観察できる (図 11 (h))。

処理に要する時間は、元データを 1/6,000 まで簡略化するのに約 3.1 秒要する。しかし、一度簡略化データを作成すれば、簡略化データを参照しての復元は約 0.6 秒、簡略化データを参照しての再簡略化は約 0.5 秒でそれぞれ実現できた。部分的復元も同等の時間で実現できる。なお、図 11 (a) の表示には約 13.5 秒要していたが、図 11 (d) の表示は毎秒 30 フレームの描画速度を実現できる。

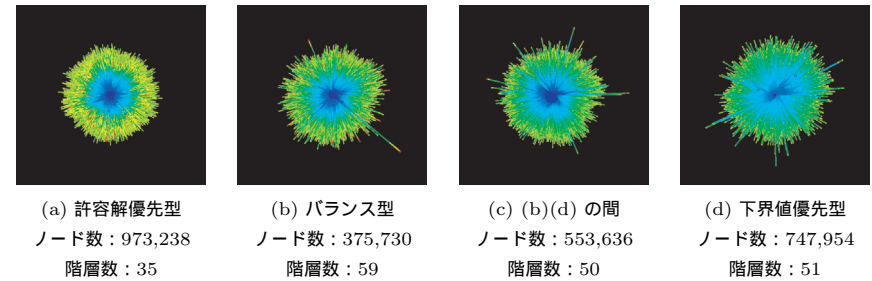


図 10 分枝限定木生成過程の比較; (a) は整数解を求めることに重点をおいているため、早い段階で値が大きくなっている。(b) は、先にいくつかの適した整数解を求めるため、何本かのリンクが深い階層まで伸びている。これらに対して、(d) では整数解を求めることに重点をおかないため、値の変動が小さい

Fig. 10 Comparison of branch-and-bound processes.

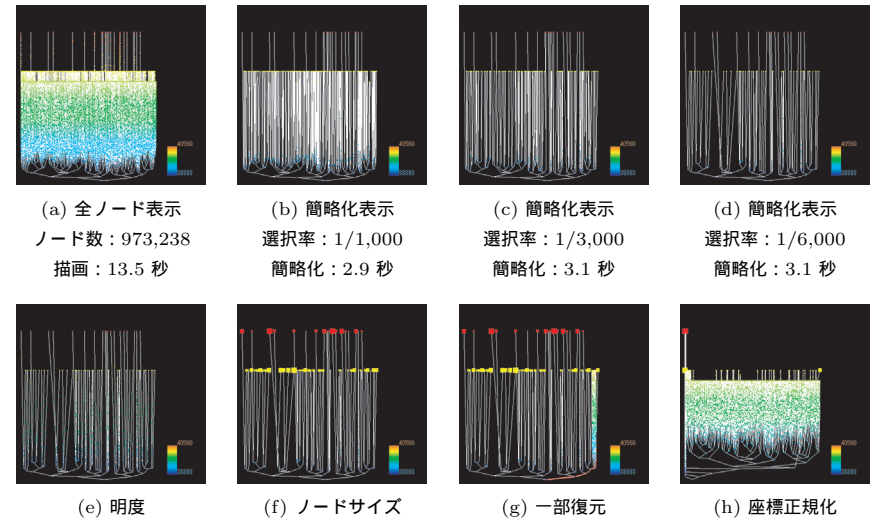


図 11 段階的な詳細度表現と復元

Fig. 11 Multilevel representation and reconstruction.

6. おわりに

大規模な分枝限定法の計算過程を可視化するシステムを提案した。この可視化システムでは、大規模データを効率的に配置することで、混雑する木構造グラフを見やすく表現することを実現した。また、変動する複数の数値データを木構造データと連動して観察できるように示す機能を開発し、さらに大規模階層データに適用するために分枝限定法で重要な情報を残しつつ簡略化する表現方法を提案した。ユーザインタフェースとして、グラフの表現形式を選択できるだけでなく、選択的に表示したい数値データを選択したり、部分木を選択したりできる。また本システムを分枝限定法の解析に適用することで、本システムの有効性を検証した。

謝辞 本研究の一部は平成19年度文部省科学研究費補助金基盤研究C(課題番号18510118)の助成による。

参考文献

- 1) 大西克実, 榎原博之, 中野秀男: 並列分枝限定法に対するビジュアライゼーションシステム, 電子情報通信学会論文誌, Vol.J79-D-1, No.7, pp.400-408 (1996).
- 2) Cook, W.: Traveling Salesman Problem, Sweden Home. <http://www.tsp.gatech.edu/sweden/>
- 3) Johnson, B. and Shneiderman, B.: Tree-Maps: A Space-filling Approach to the Visualization of Hierarchical Information Structures, *Proc. IEEE Visualization '91*, pp.284-291 (1991).
- 4) Lamping, J., Rao, R. and Pirolli, P.: A Focus+context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies, *Proc. ACM SIGCHI '95*, pp.401-408 (1995).
- 5) Robertson, G.G., Mackinlay, J.D. and Card, S.K.: Cone Trees: Animated 3D Visualizations of Hierarchical Information, *Proc. ACM SIGCHI '91*, pp.189-194 (1991).
- 6) Munzner, T., Guimbertiere, F., Tasiran, S., Zhang, L. and Zhou, Y.: TreeJuxtaposer: Scalable Tree Comparison Using Focus+Context with Guaranteed Visibility, *ACM Trans. Graphics (Proc. SIGGRAPH 2003)*, Vol.22, No.3, pp.453-462 (2003).
- 7) Zmasek, C.M. and Eddy, S.R.: ATV: Display and Manipulation of Annotated Phylogenetic Trees, *Bioinformatics*, Vol.17, No.4, pp.383-384 (2001).
- 8) Rost, U. and Bornberg-Bauer, E.: TreeWiz: Interactive Exploration of Huge Trees, *Bioinformatics*, Vol.18, No.1, pp.109-114 (2002).
- 9) Koike, H.: Fractal Views: A Fractal-Based Method for Controlling Information Display, *ACM Trans. Information System*, Vol.13, No.3, pp.305-323 (1995).

- 10) Reingold, E.M. and Tilford, J.S.: Tidier Drawings of Trees, *IEEE Trans. Softw. Eng.*, Vol.7, No.2, pp.222-228 (1981).
- 11) Itoh, T., Yamaguchi, T., Ikehata, Y. and Kajinaga, Y.: Hierarchical Data Visualization Using a Fast Rectangle-Packing Algorithm, *IEEE Trans. Visualization and Computer Graphics*, Vol.10, No.3, pp.302-313 (2004).
- 12) Andrews, K. and Heidegger, H.: Information Slices: Visualising and Exploring Large Hierarchies Using Cascading, Semicircular Discs, *Proc. IEEE Information Visualization '98*, pp.9-12 (1998).
- 13) ILOG: CPLEX. <http://www.ilog.co.jp/>
- 14) Hoppe, H.: Progressive Meshes, *Proc. ACM SIGGRAPH 96*, pp.99-108 (1996).
- 15) Kawai, H.: ADVENTURE AutoGL: A Handy Graphics and GUI Library for Researchers and Developers of Numerical Simulations, *Computer Modeling in Engineering and Sciences*, Vol.11, No.3, pp.111-120 (2006).
- 16) Martin, A., Achterberg, T. and Koch, T.: MIPLIB. <http://mipilib.zib.de/>
- 17) 今野 浩, 鈴木久敏: 整数計画法と組合せ最適化, 日科技連 (1982).

(平成20年8月9日受付)

(平成20年10月12日再受付)

(平成20年11月4日採録)



宮村(中村)浩子(正会員)

平成16年お茶の水女子大学大学院人間文化研究科複合領域科学専攻博士後期課程修了。同年より東京農工大学大学院生物システム応用科学府助手, 助教を経て, 平成21年11月より(独)日本原子力研究開発機構研究員。ボリュームビジュアライゼーション, インフォメーションビジュアライゼーションの研究に従事。博士(理学)。電気学会会員。



品野 勇治(正会員)

平成9年東京理科大学大学院工学研究科経営工学専攻博士後期課程修了。同年より同大学助手。平成11年東京農工大学講師, 平成16年同大学助教授を経て, 平成19年より同大学准教授。数理計画法の理論と応用, 組合せ最適化問題に対する並列・分散アルゴリズムとその実装に関する研究に従事。博士(工学)。オペレーションズ・リサーチ学会, IEEE, ACM

各会員。



宮代 隆平

平成 16 年東京大学大学院情報理工学系研究科数理情報学専攻博士後期課程修了。同年より同研究科学術研究支援員，東京農工大学大学院共生科学技術研究部助手を経て，平成 19 年より同大学大学院共生科学技術研究院助教。組合せ最適化，数理計画の研究に従事。博士（情報理工学）。日本オペレーションズ・リサーチ学会会員。



斎藤 隆文（正会員）

昭和 62 年東京大学大学院工学研究科情報工学専攻博士課程満期退学（平成 2 年修了）。同年より日本電信電話（株）NTT 研究所勤務。平成 9 年東京農工大学工学部助教授，平成 14 年東京農工大学大学院生物システム応用科学研究科教授。コンピュータグラフィクス，可視化，形状処理等の研究に従事。工学博士。画像電子学会，ACM SIGGRAPH 会員。