2003年日本オペレーションズ・リサーチ学会
秋季研究発表会

混合整数計画問題の解法における前処理の効果検証

東京農工大学 *鴻池祐輔 KOUNOIKE Yuusuke
01207140 東京農工大学 品野勇治 SHINANO Yuji
01507034 神戸商科大学 藤江哲也 FUJIE Tetsuya

1. はじめに

混合整数計画問題(MIP) に対する分枝カット法において，その高速化を目的とした前処理や種々のカット生成法など，様々な技法が考案されてきた．また，多くの計算実験による評価も行われてきている．しかし，各種技法は互いに影響し合うため，それぞれの技法の効果を検証することは困難である．

筆者らは，様々な技法を容易に実装できること，各技法を様々な組み合わせで適用できること，それぞれの技法を対等な条件で実装できることを重視し，各技法を外部モジュールとして実装できるソルバーを設計し，その実装を行った．本稿では，特に開発したソルバーを利用して行ったMIPに対する前処理[2]の効果について報告する．

MIPに対する前処理は，分枝限定法におけるルート問題でのみ適用されることが多い．これは前処理に要する処理時間と，前処理の効果として少なくなる分枝数とのトレードオフの結果であることが多い．しかし，もし前処理の効率的な実装が可能であるなら，全ての分枝ノードにおいて前処理を行うことによる計算時間の短縮も見込める．そこで，開発したソルバーを利用して，全てのノードにおける前処理の効果の検証を行った．

2. ソルバーの設計

ソルバーは，分枝限定法のフレームワークを提供し，各種技法の効果を調べるために，フラグインとして置き換え可能なモジュールをいくつかのタイミングで呼び出せる設計とした．一般的な商用のソルバーで提供される機能との違いは，1つの呼び出しタイミングに対して，複数のモジュールを追加・削除できる点である．この機能により，種々の解法における相互作用を調べることができる設計となっている．例えば，カット生成のアルゴリズムなどは，いくつも提案されているが，その内の幾つかを組み合わせた時の効果がどのようになるかなどを，既存の実装の変更なしに，簡単に検証できる．

図1は，分枝限定法のどのタイミングで，各種外部モジュールが呼び出されるのかを示している．外部モジュールとして置き換え可能な部分は，影のついたボックスで示している．1つのボックスに対して複数のモジュールを登録し，呼び出し可能である．

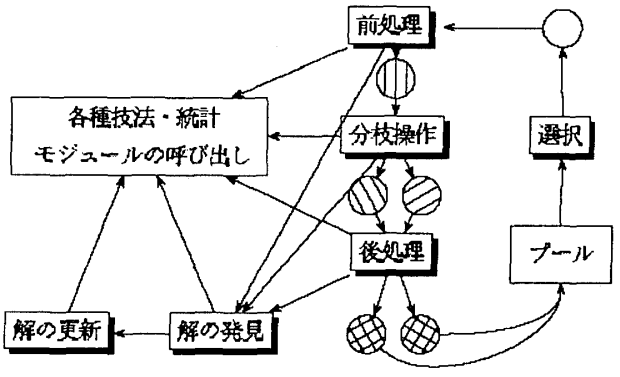

図1．外部モジュール呼び出しのタイミング

3. MIPにおける前処理

前処理は，制約式の左辺の変域と右辺値の比較により，定式化を強化する手法である．[2]で詳細に示されている手法の中で以下を実装している．

- identification of infeasibility
- identification of redundancy
- fixing variables
- improvement of bound
- improvement of coefficient

4. 評価実験のための外部モジュール

本稿での評価実験における前処理以外の技法としては，擬似コストによる分枝変数選択と，以下を外部モジュールとして実装した．

切除平面法による下界値の強化

切除平面法の実装には，COIN[1]が提供するカット生成ライブラリを用い，Gomory cut, Knapsack cover cut, Odd hole cutを提供した．

最良優先探索と深さ優先探索の組合せ

8個の子問題は深さ優先探索し，その後，新た

な最良優先で選ばれる子問題から８個を深さ優先探索することを繰り返すハイブリッド探索を適用した．

5. 数値実験結果

MIPLIB の問題のうち 42 問を解き，その計算時間と探索したノード数の比較を行った．図２は，各問題について，前処理によって計算時間と探索ノード数が何%減少したかを示す．図３は，前処理の有無による各問題の計算時間の変化を示す．図２より，ノード数は多くの場合に減少傾向にあるが，計算時間は必ずしも減少しているとは言えない．ただし，図３より，計算時間が増加した問題の多くは，比較的短時間で解けた問題であることがわかる．よって，前処理は規模の大きな問題で特に有効であることが分かる．

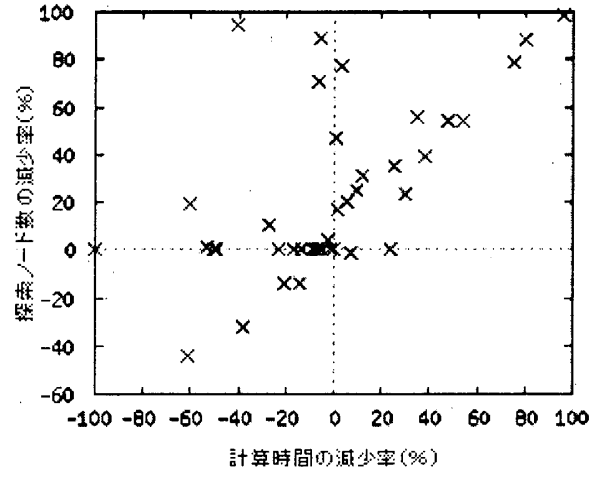

図２．探索ノード数と計算時間の減少率

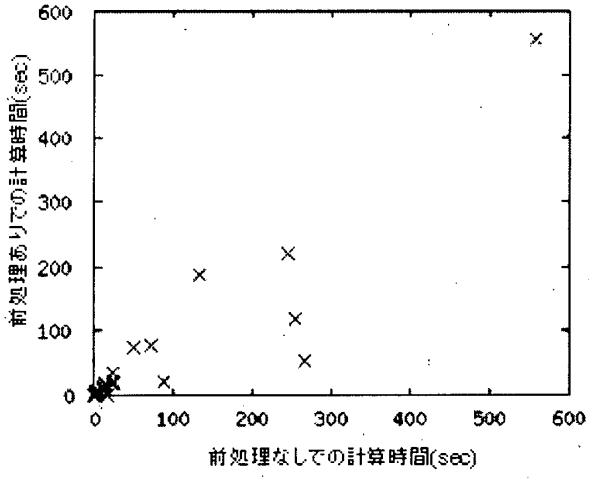

図３．前処理の有無による計算時間の変化

個々の問題の分枝限定法の進行にどのような違いが生じたか，特に良い効果のあった問題を図４に，悪い効果のあった問題を図５に示す．図４では前処理を適用したことによって，下界値の増加がより急速になっているのに対し，図５では逆に遅くなっている．また，図５では暫定解の更新が遅れることがあり，特に最適解の発見が遅くなっていることが分かる．

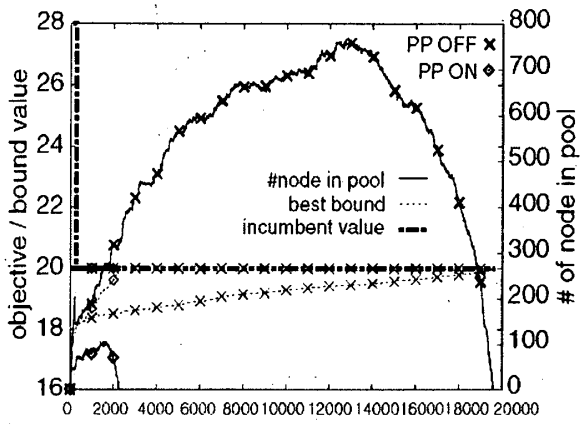

図４．MIPLIB の vpm1 での実行結果

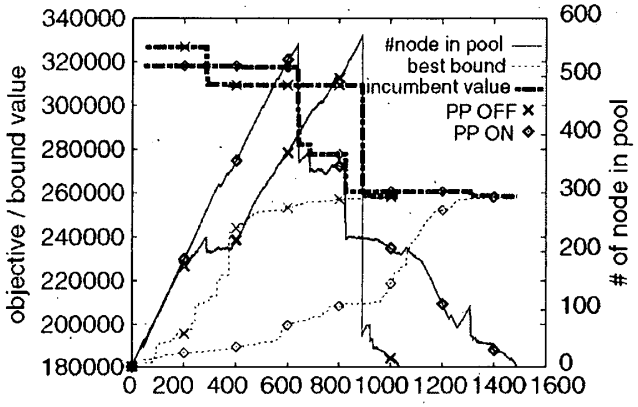

図５．MIPLIB の p0282 での実行結果

6. おわりに

本稿では，各種技法の評価のためのソルバーの設計と，紙面が許す範囲で前処理の効果を示した．より詳細な結果については当日報告する．

参考文献

[1] IBM. COIN(COmputational Infrastructure for Operations Research) http://www.coin-or.org/.

[2]M.W.P.Savelsbergh, Preprocessing and probing for mixed integer programming problems. ORSA J. on Computing Vol.6. pp.445-454,1994.