機械学習を用いた MIP ソルバーの パラメータチューニング

理化学研究所 革新知能統合研究センター 離散最適化ユニット 前原貴憲
NTT データ数理システム 石橋保身、大槻兼資、藤井浩一

1 概要

分枝限定法をコアアルゴリズムとした MIP ソルバーは運転計画やスケジューリング問題など様々な現実問題の解決に役立っている。現代的な MIP ソルバーは切除平面法や前処理、ヒューリスティクスなど様々な機能が寄せ集めてあり、そのパラメータ設定値は性能に大きく影響する。一方、現在機械学習を利用したパラメータチューニングに関する研究が活発に行われている。
本講演では機械学習の技術を MIP ソルバーのパラメータチューニングに応用した成果を発表する。

2 MIP ソルバーの基本要素

MIP ソルバーが対象としているのは「混合整数計画問題」と呼ばれる問題の種類となる。

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \\ & x \geq 0, \\ & x \in \mathbb{Z}^n \times \mathbb{R}^m \end{array} \tag{1}$$

MIP ソルバーの核となっているのは分枝限定法というアルゴリズムである。これは変数を場合分け・逐次固定していくことにより、解を列挙し厳密解を求めていくアルゴリズムである。ただし一部固定をして得られた緩和解が暫定解（整数性が満たされている解）よりも悪かった場合は、それ以上の固定は行われない（限定操作）。
現代的な MIP ソルバーは主に以下の技術要素から成り立っている。

- 前処理（Presolve）
- 切除平面法（ Cutting Plane ）
- ヒューリスティクス (Heuristics)

以下は Numerical Optimizer における分枝限定法の処理フローとなる。

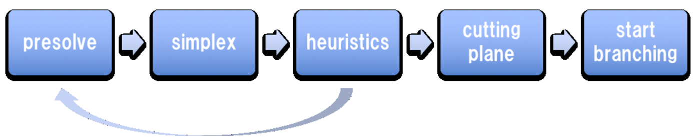

図 1: 分枝限定法処理フロー

各技術要素ごとにパラメータがあり、非公開パラメータまで全て合わせるとおよそ 80 ほどになる。以下パラメータの一部を掲載する。

パラメータ名	カテゴリー	内容
clevel	Cutting Plane	加える切除平面の on/off/aggressive
probing	Presolve	Probing の on/off
decompose	Presolve	非連結成分分解の on/off
rins	Heuristics	ヒューリスティクス RINS の on/off
neighbourSearch	Heuristics	ヒューリスティクス NeighbourSearch の on/off
diving	Heuristics	ヒューリスティクス diving の on/off/aggressive
branch-p	Branching	分枝操作における深さ優先探索・幅優先探索の重視度
scaling	その他	スケーリング手法の切り替え

表 1: Numerical Optimizer 分枝限定法関連パラメータ一覧（一部抜粋）

3 パラメータチューニング技術について

近年は機械学習の隆盛に伴い、自動パラメータチューニングについての研究も盛んに行なわれている。機械学習のタスクにおいてもニューラルネット

ワークにおけるノード数、決定木の深さなど学習に際して多くのパラメータがある。このようなパラメータチューニングにおいて、幾つかの手法が提案されており以下のその一部を挙げる。

- Randomization
- Baysiean optimization（ModelOptimizer [1] SMAC [2]）
- Hyperband [3]

最適化の文脈ではブラックボックス最適化などとも言われコンペティションも行なわれている（BBComp, `https://bbcomp.ini.rub.de/`）。現状ではどの手法が常に優位、ということはなく、問題設定によってはRandomization（パラメータを乱数によって選択する手法）が良い、などとする結果もある [4]。

4 random-MIP に対するパラメータチューニング

MIP は NP 困難な問題群であり、それぞれごとに分枝限定法の挙動は大きく異なる。ところで、MIP の実問題への応用では性質の類似した問題を複数個解くことが多い。たとえば、少しずつ違うパラメータをもつ問題群すべてについて解を求めたいケースが代表的である。本研究では、このような場合のパラメータチューニング手法として以下のアプローチを考える。

> 少数の問題を様々な MIP パラメータ（テストパラメータ）で解き、その計算時間や分枝数などの情報を取得する。その結果を機械学習を用いて処理することで「いま解いている問題群がどのような問題か」を認識する。その結果に応じて適切と思われるパラメータに設定する。

このアプローチの利点として以下が期待できる。

- テストパラメータの数だけ MIP を独立に解けば良いため、パラメータチューニングにかかる時間が推定しやすく、並列性も高い。多くの場合、前節で述べた一般的な方法よりも MIP の求解回数が少なくなる。
- 単にパラメータを最適化するだけでなく、現在解いている問題について構造的な知見が得られる。例えば、今解いている問題が巡回セールスマン問題と近いパラメータで解ける、などといったことがわかる。

以下ではこのアプローチが有力であることを確認する。以下のように定式化されるクラスの MIP をランダムに生成することを考える。

$$
\begin{array}{ll}
\text{maximize} & \sum_{r\in R} d_r z_r + \sum_{k\in K} d_k x_k - \sum_{c\in C} d_c w_c - \sum_{r\in R} y_r \\
\text{subject to} & a_r z_r + \sum_{k\in K} a_{rk} z_k - \sum_{c\in C} w_c - y_r \le b_r, \quad (r\in R), \\
& x_k \le u_k, \quad (k\in K), \\
& w_c \le v_c, \quad (c\in C), \\
& z_r \in \{0,1\}, \quad (r\in R), \\
& x_k \in \{1,\dots,100\}, \quad (k\in K), \\
& y_r \in \{0,100,200,\dots,10000\}, \quad (r\in R), \\
& w_c \in [1,20], \quad (c\in C).
\end{array} \tag{2}
$$

ここで各係数は以下の範囲でランダムに生成される.

$$
\begin{aligned}
b_r &\in \{951,\dots,999\}, \\
u_k &\in \{1,2,3,4\}, \\
v_c &\in \{1,\dots,20\}, \\
d_c &\in \{101,\dots,200\}, \\
a_r &\in \{1000,\dots,1800\}, \\
a_{kr} &\in \{1,\dots,2b_r\}, \\
d_r &\in \{1,\dots,1800\}.
\end{aligned}
$$

整数変数の個数 $|K|$、実数変数の個数 $|C|$、制約式の個数 $|R|$ は問題生成時に指定するパラメータである。これらの問題は、問題の生成に用いるパラメータ $|K|,|C|,|R|$ が同じであれば、生成された問題も近い性質をもつと考えられる。そこで、様々なテストパラメータにおいて問題を解き、その実行結果から用いたパラメータを推定する。

実験では、各パラメータを $|K| \in \{250,500\}$、$|C| \in \{5,10,20\}$、$|R| \in \{50,75,100\}$ の範囲で動かし，各設定ごとに 50 個ずつ・合計 900 個の問題を生成した。各問題について、それぞれを 50 個のテストパラメータで解いて計算にかかった時間を計測した。

以下では計算時間を機械学習を用いて解析する。前処理として計算時間は平均が 1 になるように正規化した。まず、計算時間から正しいパラメタを推定できるかどうかを確認した。900 個の問題を 810 個の学習データセットと 90 個のテストデータセットに分け、勾配ブースティング決定木 [5] (xgboost) を用いて学習データセットで学習し、テストデータセットのどれだけの問題について正しくラベルを復元できるかを調べた。その結果、$|K|$, $|C|$, $|R|$ について，それぞれ精度 0.92，0.8, 0.85 で正しいラベルを推定することに成功した。このことから、復数のテストパラメータでの実行結果から問題の特性を判定するアプローチは有力であると考えられる。

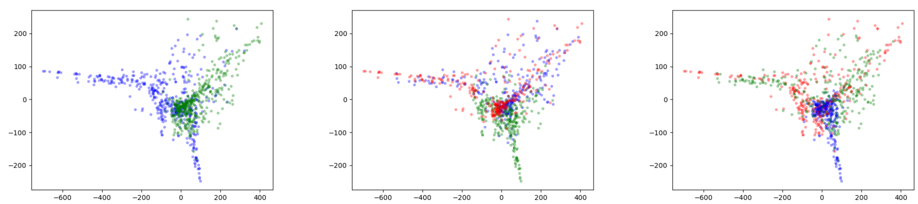

図 2: 計算時間を Isomap により 2 次元に埋め込んだもの．色付けは左から $|K|$, $|C|$, $|R|$ に対応．

続いて、計算時間から問題がどのように分類されるかを見るため，多様体学習の手法である Isomap [6] を用いて 50 次元の計算時間データを 2 次元に縮約し、可視化を行った。結果を図 2 に示す。この結果から、$|K|$ の違いは多様体上の右上とそれ以外に分類され、$|C|$ および $|R|$ の違いは多様体上の下とそれ以外に分類されていることがわかる。

本講演では、さらにこれらの random-MIP 問題に対して最適なパラメータを学習する機械学習モデルを構築する方法を紹介する。

5　おわりに

本稿では MIP ソルバーとパラメータチューニングについて論じ、ある限られたクラスの問題に対する有効性を論じた。今後はさらにこの技術をより広汎なクラスの問題に一般化していくことを考える。

References

[1] VMStudio 8.3 新機能紹介 http://www.msi.co.jp/vmstudio/vmstudio83.html

[2] Hutter, Frank, Holger H. Hoos, and Kevin Leyton-Brown. “Sequential Model-Based Optimization for General Algorithm Configuration.” LION 5 (2011): 507-523.

[3] Li, Lisha, et al. ”Hyperband: A novel bandit-based approach to hyperparameter optimization.” arXiv preprint arXiv:1603.06560 (2016).

[4] http://www.argmin.net/2016/06/20/hypertuning/

[5] T. Chen and C. Guesterin: “Xgboost: A scalable tree boosting system”. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2016): 785–794.

[6] J. B. Tenenbaum, V. de Silva, J. C. Langford: "A Global Geometric Framework for Nonlinear Dimensionality Reduction." Science 290, (2000): 2319–2323.