使ってみよう線形計画ソルバ

品野　勇治，藤井　浩一

本稿では，線形計画問題を解くソフトウェアである線形計画ソルバの標準的な利用方法と，原稿執筆時点で利用可能なソルバを可能な限り紹介する．筆者らはソルバ開発コミュニティの中で仕事をしているので，この機会に現在のソルバ開発現場の様子も紹介するとともに，各ソルバの特徴として何を紹介するべきかは，できる限り開発者，または，開発者に近い研究者に問い合わせて記述した．本稿は，多くの線形計画ソルバの選択肢がある中で，読者がもつ問題を解くのに最も適当なソルバを選択するための指針を与えることを主たる目的としている．

キーワード：線形計画，ソフトウェア，ベンチマーク

1. はじめに

本稿の執筆依頼に際して，「最近，SCIP Optimization Suite が改良されたと話題である」という内容のメールをいただいた．SCIP Optimization Suite とは筆者の所属する Zuse Institute Berlin が主として開発する最適化ソフトウェアパッケージ群であり，整数計画ソルバとして機能する SCIP (Solving Constraint Integer Programs) を軸とする．実際には，改良は日々行われている．本稿の本題から外れるが，まずソフトウェア開発の現状を紹介し，ソフトウェアが日々改善されている様子を見ていく．

商用，アカデミックを問わず，最適化ソルバの開発には，ソフトウェア工学における**継続的インテグレーション** (CI: Continuous Integration) が利用されている．開発のワークフローが定義されており，ある機能の追加に対しては，バージョン管理システムの管理下にあるオブジェクト（ソースコードファイル，ディレクトリツリーなど）を複製した**ブランチ** (Branch) が作られ，特定の機能を追加したソルバがそのブランチで開発されテストされる．そのブランチがメインのブランチに統合される際には，複数人での**コードレビュー** (Code Review) が行われ，統合されると新機能が追加される前の機能が正常に動作していることを確認する**リグレッションテスト** (Regression Test) が自動的に動作する．その後，一連のテストインスタンスにおける**性能テスト** (Performance Test) が自動的に実行され，その結果はデータベースで管理される．よって，ある機能追加が，どのインスタンスに対してどの程度のインパクトを与えたかが常に把握されている．

SCIP Optimization Suite 開発グループでは，ワークフローの定義のための議論開始から，利用するバージョン管理ソフトウェアの選定，各種管理のためのシステム構築には，約 2 年を要したように思うが，そのおかげでソフトウェアのリリースは容易になった．現在のリリース作業は，本質的には，日々追加されている機能のどこまでをリリースに含めるかどうかを決めるだけである．

本稿では，現在利用可能な**線形計画ソルバ**を紹介する．ここでいう線形計画ソルバとは，以下のような制約式および目的関数が線形となる線形計画問題を解くソフトウェアのことである．

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to:} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{x} \in \mathbb{R}^n. \end{aligned} \tag{1}$$

ここで，$A \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{c} \in \mathbb{R}^n$ はユーザが定義するデータとなる．問題 (1) は**等式標準形**で記述されているが，ソルバでは適当な前処理を行い不等式も扱うことができる．紹介するソルバのバージョンは原稿執筆当時（2018 年 12 月 1 日）におけるもので，リンクも当時確認したものになる．

2 節では，線形計画ソルバのインストールから標準的な利用方法について解説する．実際に利用する際に注意してもらいたい数値的困難への対応について解説し，近年利用可能になった**厳密線形計画ソルバ** (Exact LP solver) の利用方法を紹介する．厳密線形計画ソルバとは，有理数による係数および右辺値の記述を許容

しなの　ゆうじ
Zuse Institute Berlin
Takustrasse 7, 14195 Berlin, Germany
shinano@zib.de
ふじい　こういち
株式会社 NTT データ数理システム
〒 160–0016 東京都新宿区新宿 35 煉瓦館 1F
fujii@msi.co.jp

し，任意の精度で最適解を得るソルバのことである．

3節および4節では，商用およびアカデミック線形計画ソルバを紹介する．節タイトルで各ソルバの正式名称を記述し，文章内の紹介では通称を用いている．各ソルバへのアクセス情報は脚注に示す．線形計画ソルバに限らず，最適化ソフトウェアの多くはHans Mittelmannによって頻繁にベンチマークテストが行われており，BENCHMARKS FOR OPTIMIZATION SOFTWAREというWebページ[1]に，実行結果が日付とともにまとめられている．線形計画ソルバのベンチマークとしてはCLP, Google-GLOP, HiGHS, lp_solve, MATLAB, MOSEK, GLPK, SAS-OR, SoPlex, QSopt[2]に対して以下の分類により結果が示されている[3]．

- Benchmark of Simplex LP solvers
- Benchmark of Barrier LP solvers
- Large Network-LP Benchmark (commercial vs free)

このベンチマーク結果はベンチマークインスタンスによる結果であり，特定の問題に限れば結果は大きく変わりうることに注意してほしい．各ソルバのパラメタ設定を変更した場合も同様である．加えて，ソルバの評価は単に短時間で問題が解ければよいというものでもない．ベンチマークの結果は一つの目安にはなるが，ユーザが解きたいインスタンスに対してどのソルバが適当であるかを知るには，複数のソルバを自ら利用してみることを強く勧める．本稿では，ここに示されているソルバを紹介することに加えてBPMPD, COPL_LP, CPLEX, Gurobi, HOPDM, LINDO, MIPCL, PIPS, Xpressおよび日本国産のNumerical Optimizerを紹介する（各種スプレッドシートにも，線形計画問題を解く機能は付属しているが本稿では扱わない）．

2. 線形計画ソルバの利用方法

本節では線形計画ソルバの利用方法についてSCIP Optimization Suiteに付属する線形計画ソルバSoPlexを用いて説明する．ほかのソルバについても基本的な利用方法はほとんど同様である．

2.1 準備

Linux環境およびMac環境におけるSoPlexのインストール手順を述べる．

2.1.1 インストーラの利用

インストーラを用いる場合はSCIPのWebページ[4]へ行き，左側のタブから“download”を選ぶと，各プラットフォームのインストーラのダウンロードリンクが表示される．たとえばMacでインストールする場合は「SCIPOptSuite-6.0.1-Darwin.dmg」と書かれたリンクをクリックすると名前や所属などを記入するフォームが表示される．ライセンス条項を読み同意したらダウンロードを行う．

2.1.2 ソースからのコンパイル

SoPlexのWebページ[5]へいき，左側のタブから“download”を選ぶ．「Source code： SoPlex」と書かれたリンクをクリックするとインストーラの場合と同様名前や所属などを記入するフォームが表示され，ダウンロードを行うことができる．

ソースファイルのコンパイルにはビルドツールCMake[6]が必要である．SoPlexバージョン4.0.0ではCMakeのバージョンが3以上である必要があった．手元のCMakeのバージョンが古い場合は，適宜更新してほしい．

適当なフォルダを作成し，フォルダ上でダウンロードしたSoPlexのフォルダのパスを指定をして`cmake`コマンドを実行する．続けて`make`コマンドを実行するとバイナリファイル`soplex`が作成される．以下はコンパイル実行例である．

```
$tar xzvf soplex-4.0.0.tgz
$cd soplex-4.0.0
$mkdir build
$cd build
$cmake..
$make
```

以上のように実行すると`build`以下の`bin`フォルダ以下にバイナリファイル`soplex`が作成される．

2.2 LPフォーマットとMPSフォーマット

線形計画ソルバに対する標準の問題記述フォーマットには，MPS (Mathematical Programming System) フォーマットとLPフォーマットの二つがある．MPSフォーマットは，古い計算機の仕様を意識した可読性

[1] http://plato.asu.edu/bench.html
[2] Hans Mittelmannの Webページにおける表記
[3] 2019年1月10日段階では，以前まで行っていたCPLEX, Gurobi, Xpressのベンチマークを行っていない．その経緯については「END OF A BENCHMARKING ERA」というタイトルでWebページに記載されている．
[4] https://scip.zib.de/index.php
[5] https://soplex.zib.de/index.php
[6] https://cmake.org

の低いフォーマットなので，ここではLPフォーマットで問題を記述する．LPフォーマットの詳細については文献 [1] などを参考にしてほしい．

ここでは次のような線形計画問題を例とする．

$$\begin{aligned} &\text{min} && 180x_1 + 160x_2 \\ &\text{subject to:} && 6x_1 + x_2 \geq 12 \\ & && 4x_1 + 6x_2 \geq 24 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

ファイル `sample.lp` に上記問題をLPフォーマットで記述すると以下のようになる．

```
Minimize
obj: 180 x1 + 160 x2
Subject to
r_1: 6 x1 + x2 >= 12
r_2: 4 x1 + 6 x2 >= 24
Bounds
 x1 >= 0
 x2 >= 0
End
```

計算するために，次のコマンドで実行する．

```
$./soplex sample.lp
```

実行すると最適化計算が行われ，計算終了時のステータス，計算時間，反復回数および目的関数値等最適化計算の情報が標準出力される．

```
SoPlex status       : problem is solved
                      [optimal]
Solving time (sec) : 0.00
Iterations          : 2
Objective value     : 7.50000000e+02
```

解を見たいときは，オプション`-x`を付与する．上の例でオプションを付与すると以下のように出力される．

```
Primal solution (name, value):
x1          1.5000000000000000e+00
x2          3.0000000000000000e+00
```

LPフォーマットは直感的に記述できるものの，比較的最近考案されたフォーマットであるためソルバが対応していない場合がある．同じインスタンスに対する各ソルバでの性能を比較したい場合には，あらゆるソルバが対応しているMPSフォーマットへ変換する必要がある．このような場合はSoPlexに付属しているファイルフォーマットの変換機能を使うのがよい．実行例を以下に示す．

```
$./soplex sample.lp --writefile=out.mps
```

上記の例では，LPフォーマットで記述された`sample.lp`をMPSフォーマットで記述された`out.mps`に出力する．

線形計画ソルバのほかのインターフェイスとしてはプログラム言語Pythonのインターフェイス (CPLEX, Gurobi, Numerical Optimizer, Xpress)，Excelインターフェイス (LINDO, Numerical Optimizer) などが提供されている．

2.3 ライブラリとしての使い方

SoPlexをはじめ多くの線形計画ソルバはライブラリとして利用することを可能とするAPI (Application Program Interface) が用意されている．ライブラリとして利用するには，各ソルバのマニュアルによりAPIを調べて使うことになるが，サンプルは一般的には少ない．

整数計画ソルバSCIPはLP Solver Interfaceといういくつかの線形計画ソルバの共通インターフェイスを提供している[7]．たとえばLP Solver Interfaceには`SCIPlpiChgBounds`という関数があるが，SCIPはリンクしている線形計画ソルバに対して，この関数を通じて変数の領域変更を適用することができる．ほかにも制約式を追加する関数`SCIPlpiAddRows`や目的関数を変更する関数`SCIPlpiChgObj`などがある．これによりたとえば線形計画問題を一度解き，さらに制約を加えた後再最適化する，なども可能である．各線形計画ソルバのライブラリとしての使い方はLP Solver Interfaceの各ソルバの実装を調べるとよい．

2.4 線形計画における数値的困難

実際に現実問題を線形計画問題として定式化しソルバで解こうとする場合，数値的困難に見舞われることもある．たとえば実行可能であるはずの問題がソルバに実行不可能と判定されるなどのケースが考えられる．ここで**実行可能**とは設定された閾値の範囲で制約式を満たす解が存在することを言い，**実行不可能**とは解が存在しないことを言う．ここでは，線形計画法の中でも数値的困難に対処しやすい単体法に焦点を絞り，どのように対処するかを見ていく．

線形計画問題の等式標準形 (1) を考える．最適解における基底行列を B とすると，最適解は $B\boldsymbol{x} = \boldsymbol{b}$ と

[7] https://scip.zib.de/doc-6.0.0/html/group__LPIS.php

いう一次方程式の解で表現できる．今右辺値ベクトル $\boldsymbol{b}$ を $\boldsymbol{b}+\delta\boldsymbol{b}$ と微小摂動したときの解を $\boldsymbol{x}+\delta\boldsymbol{x}$ とすると，以下のような関係式が成り立つ．

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}+\delta\boldsymbol{x}\|} \leq \kappa(B)\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}+\delta\boldsymbol{b}\|}. \tag{2}$$

ここで $\kappa(B)$ は基底行列の条件数 $\|B\|\|B^{-1}\|$ を表している．

ソルバにはパラメタとして **実行可能性に関する閾値** (feasibility tolerance) と **最適性に関する閾値** (optimality tolerance) の二つがある．ソルバは最適化計算において制約違反量が feasibility tolerance 以内の解を許容する．

関係式 (2) から条件数が大きいほど解はデータの誤差の影響を受ける可能性があることがわかる．たとえば IEEE 254 の規格では double の精度は約 $1.0e-16$ であるためソルバの feasibility tolerance が $1.0e-6$ であれば条件数は $1.0e+10$ 程度の範囲までにおさまるのが望ましい．

条件数が大きい問題がどのように数値的不安定さを引き起こすかを見るために，以下のような問題を考える．

$$\begin{array}{ll} \min & x_1+x_2 \\ \text{subject to:} & 0.33333x_1+0.66667x_2=1 \\ & x_1+2x_2=3 \\ & x_1,x_2\geq 0. \end{array}$$

SoPlex ではオプション `-q` を付与することにより，条件数（正確には条件数の近似値）を含めた諸々の統計情報を出力できる．上の問題を実行すると

```
Condition Number : 4.00000000e+05
```

と表示され条件数が非常に大きい値になっていることがわかる．これは一番目の制約式と二番目の式がほとんど平行になっているためである．ここで一番目の制約式の右辺値のみ微小に値を変更する．

$$\begin{array}{ll} \min & x_1+x_2 \\ \text{subject to:} & 0.33333x_1+0.66667x_2=1.000001 \\ & x_1+2x_2=3 \\ & x_1,x_2\geq 0. \end{array}$$

上のようにすると変数 x_1 の値は 1.00 から 0.80 と大きく変化し目的関数値も 2.0 から 1.9 に変化する．再度関係式 (2) を確認すると，右辺値の内 $\kappa(B)\|\delta\boldsymbol{b}\|$ が 0.4 となり，解が相対的に大きく変動しても不思議ではないことがわかる．

数値的困難に見舞われたときは，データの誤差とソルバの閾値に応じて条件数の値を確認し，もし条件数の値が大きすぎる場合は定式化を見直すなどが推奨される．文献 [2] には条件数以外にもほかに数値的困難が引き起こされるケースについて記述されている．

2.5 厳密線形計画ソルバ (Exact LP Solver)

前節では数値的困難性と線形計画ソルバの関連について述べたが，浮動小数点の誤差を排除した有理数解が必要になることもある．ここでは，近年 SoPlex で利用可能になった厳密線形計画ソルバの機能を紹介する [3, 4]．

SoPlex の導入方法については 2.1 節で述べたが，厳密線形計画ソルバとして利用するには `cmake` コマンド実行時のオプションを変更する必要がある．

```
$cmake.. -DGMP=true
$make
```

ここで重要なことは，オプションとして`-DGMP=true` と指定していることである．これにより **多倍長ライブラリ** GMP が SoPlex にリンクされる．GMP は GNU プロジェクト[8]から提供されている多倍長精度の演算を可能にするライブラリである[9]．

SoPlex を厳密線形計画ソルバとして利用する際には，有理数記述した LP フォーマットの問題記述を扱うことが可能である．LP フォーマットで有理数係数の線形計画問題を記述したファイル `rational.lp` を用意する．

```
Minimize
obj: 180 x1 + 160 x2
Subject to
r_1: 103/17 x1 + 20/19 x2 >= 12
r_2: 85/21 x1 + 139/23 x2 >= 24
Bounds
 x1 >= 0
 x2 >= 0
End
```

以下のコマンドで実行する．

```
$./soplex -X --real:feastol=0
--real:opttol=0 --int:solvemode=2
--int:syncmode=1 --int:readmode=1
--int:checkmode=2 rational.lp
```

[8] https://www.gnu.org
[9] https://gmplib.org

オプション-X を用いることで有理数の値を出力することができる．上の例では以下のように出力される．

```
Primal solution (name, value):
x1 7372764/5047783
x2 15107964/5047783
```

上記オプションの詳細は SoPlex の Web ページ[10]に記述されている．

3. 商用線形計画ソルバ

ここで紹介するすべての商用線形計画ソルバは，線形計画ソルバを機能の一部として含み，通常，混合整数計画問題も解け，非線形計画問題を扱えるソルバもある．本稿では線形計画ソルバとしての機能に限定して記述する．線形計画ソルバとしては，通常，単体法，および，内点法の実装を含む．また，複数 CPU コアをもつ計算機上でマルチスレッド並列処理として動作させた場合，複数のアルゴリズムが同時に実行され，先に解を得たアルゴリズムの終了をもって計算終了とするソルバも複数存在する．

3.1 IBM ILOG CPLEX Optimizer[11]

Robert E. Bixby により C 言語で開発され単体法（CPLEX の名前は，the simplex method as implemented in the C programming language からきている）の実装が，1988 年に CPLEX Optimization Inc. から販売された．その後，CPLEX は，ILOG 社を経て，現在は IBM 社の製品として販売されている．もちろん現在は内点法の実装も含む．長期にわたり継続的に開発・利用されていて，ライブラリとしての API も充実している．線形計画ソルバとしては，単体法の内部において制約式を動的に調整するオプションがある．ユーザはオプションを有効にすると内部では核単体法 [5] という CPLEX が独自に開発した単体法が動作し，基底行列のサイズが動的に変更されている．

3.2 Gurobi Optimizer[12]

CPLEX の元開発者である Zonghao Gu，Edward Rothberg および Robert E. Bixby によって 2008 年に始められた．商用ソルバの中では比較的歴史は浅いが CPLEX 元開発者が開発陣に加わっていることもあり開発は活発である．単体法を拡張することにより区分的線形関数を目的関数とした問題も取り扱うことができるのが特徴である[13]．

3.3 LINDO[14]

LINDO は 1979 年に Linus Schrage と Kevin Cunningham によって開発された．「What'sBest!」という Excel アドイン機能を提供しており，Excel から直接または Visual Basic 言語からソルバを呼び出すことが可能である．

3.4 MATLAB[15]

MATLAB の最適化ツール MATLAB Optimization Toolbox は 1993 年にリリースされ，線形計画問題をはじめ多くの問題のソルバを提供している． MATLAB の行列形式で問題を記述することも可能だが，モデリング言語 AMPL[16]の提供する MATLAB API を使うことによって，AMPL モデルを読み込むことも可能である．自動摂動などの適応戦略が実装されていて，ソルバとしての安定性に寄与している．

3.5 MOSEK[17]

1997 年に Erling D. Andersen および Knud D. Andersen によって始められた．初期の段階で 線形計画問題に対する同次自己双対内点法が実装されており，同手法は現在では多くのソルバーでも採用されている [6].

3.6 Numerical Optimizer[18]

株式会社 NTT データ数理システムが開発する数理最適化ソフトウェアである．山下浩により 1980 年代に開発が始まり，その後田辺隆人が加わり主双対内点法による非線形最適化とモデリング言語を備えたパッケージとして V2 が発表され，以来バージョンアップを重ねている． Excel 上から実行できる Excel アドインおよび開発支援を行う Nuorium という独自の GUI を備えており，プログラミングに不慣れなユーザでも容易に扱えるのが特徴である．線形計画ソルバとしては主双対内点法以外にも単体法はもちろん制約式行列が一台の計算機のメモリに収まりきらない超大規模線形計画問題に対応した matrix free 内点法を備えている．

3.7 SAS/OR Software[19]

統計解析ソフトウェアである SAS は初期の段階から線形計画ソルバを有していたが，2004 年より本格的なソルバ開発が Yan Xu, Manoj Chari および Radhika

[10]https://soplex.zib.de/doc/html/EXACT.php
[11]https://www.ibm.com/jp-ja/marketplace/ibm-ilog-cplex
[12]http://gurobi.com
[13]https://mip2014.engineering.osu.edu/sites/mip2014.engineering.osu.edu/files/uploads/mip2014_Gu.pdf
[14]https://www.lindo.com
[15]https://www.mathworks.com/products/optimization.html
[16]https://ampl.com
[17]https://www.mosek.com
[18]https://www.msi.co.jp/nuopt
[19]https://www.sas.com/ja_jp/software/or.html

Kulkarni によって開始され，2006 年に SAS 9.1.3 でリリースされた．「OPTMODEL」というモデリング言語を使うことにより SAS プラットフォームでの結果可視化などができる．線形計画ソルバとしては Dantzig–Wolfe 分解法が適用可能である．ブロック構造をユーザが指定することも可能だが，自動的にブロック構造を検出させるアルゴリズムの適用も可能である．

3.8 FICO Xpress Optimization[20]

1983 年に Robert Ashford と Bob Daniel によって当時 Dash Optimization 社によって開発が始まった [7]．同社は 2008 年に FICO 社に買収されたが，開発は継続され定期的にバージョンアップされている．また，CPLEX と同程度の操作を可能とする API も用意されている．線形計画ソルバとしては LP folding という問題の対称性を活かして高速化する技術[21] や双対単体法および内点法から基底解を求めるクロスオーバーの並列化が搭載されている．

4. アカデミック／非商用線形計画ソルバ

本節では本原稿執筆時点で入手可能なアカデミック，または，非商用線形計画ソルバを可能な限り列挙する．単体法の実装として，CLP, GLPK, Google-GLOP, lp_solve, MIPCL, PIPS-S, QSOpt, QSOpt_ex, SoPlex を紹介する．内点法の実装として，BPMPD, COPL_LP, HOPDM および PIPS-IPM を紹介する．ソルバ GLPK, lp_solve, MIPCL は混合整数計画問題も扱うことができる．

アカデミックソルバは，博士課程の学生たちによって開発・拡張されていることも多い．このような場合博士論文を書き終わると開発・拡張が中断してしまうケースが多々ある[22]．一方で多数の研究者によって分担してメンテナンスされる場合もあるなど，アカデミックソルバの開発体制は多種多様であるため，各ソルバの開発者に関する情報は基本的に省略する．

4.1 BPMPD[23]

Csaba Mészáros によって開発された内点法ソルバである．最後のリリースが 1998 年と古いが Windows 環境のバイナリがダウンロードでき，本原稿執筆時においても Windows10 上で動作することが確認できた．

4.2 CLP[24]

CLP (Coin-or linear programming) は，COIN-OR[25]（Open Source のコードを提供している非営利団体）で開発している Open Source の線形計画ソルバである．コードは C++ 言語で書かれている．二次割当計画問題の線形緩和に有効な Idiot crash と呼ばれる初期基底解構成方法が実装されている．

4.3 COPL_LP[26]

Yinyu Ye によって始まり杉数科技有限公司および上海財経大学によって運営されているオープンソースの最適化ソフトウェアプラットフォーム Leaves において提供されている内点法ソルバである．インターフェイスは Python になっている．Ubuntu 16.04.5 でビルドする場合には Python の cython パッケージおよび scipy パッケージが必要でありなおかつソースコードの修正が必要だった．付属のドキュメントによると同次主双対内点法が実装されている．

4.4 GLPK[27]

GNU プロジェクトによって提供されているソルバである．モデリング言語，GNU MathProg が付属する．

4.5 Google-GLOP[28]

GLOP は，Google Optimization Tools (OR-Tools) に付属する線形計画ソルバである．コードは C++ で書かれているが，Python，C#，Java から呼び出せるようにラッパーが用意されている．MPS ファイルのモデルを読むドライバが `examples` フォルダ以下に `mps_driver.cc` として含まれているが，本原稿執筆段階では，Google OR-Tools のページ[29] からダウンロードできるアーカイブでは自動的にビルドされず，ソースコードの修正が必要であった（MacOS のみでのテスト）．したがって，本稿では GitHub のページを主として掲載した．

4.6 HiGHS[30]

Julian Hall および Qi Huangfu によって開発された「hsol」を前身としその後 Julian Hall および Ivet Galabova に引き継がれ 2018 年に公開された．双対単体

[20] https://www.msi-jp.com/xpress
[21] https://community.fico.com/s/blog-post/a5Q80000000Drp2EAC/fico1299
[22] 学術機関で，どのように継続的に優れたソフトウェアを維持管理する仕組みを作るかという課題の解決を試みるプロジェクト (http://www.zib.de/projects/sustainable-infrastructures-archiving-and-publishing-high-performance-optimization-software) もドイツでは立ち上がっている．
[23] http://old.sztaki.hu/~meszaros/bpmpd/
[24] https://projects.coin-or.org/Clp
[25] https://www.coin-or.org
[26] https://www.shanshu.ai/product/leaves
[27] https://www.gnu.org/software/glpk/
[28] https://github.com/google/or-tools
[29] https://developers.google.com/optimization/
[30] https://github.com/ERGO-Code/HiGHS

法のみが実装されていて「部分最適化」という 1970 年代に考案されたピボット戦略手法を用いて単体法の並列化が実装されているのが特徴である [8].

4.7 HOPDM[31]

Jacek Gondzio によって開発されたインフィージブル主双対内点法のソルバである．バージョン 2.13 のソースコードが公開されているが（最新のバージョンは 2.30）Ubuntu 16.04.5 ではビルドすることができなかった.

4.8 lp_solve[32]

lp_solve はフリー（ライセンスは LGPL）のソルバである．フリーのソルバとしては，さまざまな入力ファイルフォーマットを扱える．LP フォーマットファイルも読める． また，ライブラリとしての利用も柔軟で，C, VB, .NET, Delphi, Excel, Java などから呼び出せる．さらに，ドライバを通して，AMPL, MATLAB, O-Matrix, Scilab, Octave, R などからも呼び出せる．さまざまな環境から呼び出せるので用途によっては利用しやすい.

4.9 MIPCL[33]

MIPCL は整数計画ソルバとして開発されているが CLP という線形計画を解くための C++ クラスも 提供している．ソースファイルは提供されていないが，実行ファイルおよびライブラリは Web ページからダウンロードできる.

4.10 PIPS[34]

PIPS は，大規模 2 段階確率線形計画問題を解く並列ソルバとして開発されたソフトウェア・パッケージである． 線形計画ソルバとしては，以下の構造をもつ線形計画問題を分散メモリ環境上で解くソルバである.

$$\min_{\boldsymbol{x}\in\mathbb{R}^{n_1},\boldsymbol{y}\in\mathbb{R}^{n_2\times s}} \boldsymbol{c}^\top\boldsymbol{x}+\sum_{i=1}^{s}\boldsymbol{q}_i^\mathsf{T}\boldsymbol{y}_i,$$

subject to:

$$\begin{aligned}
&A\boldsymbol{x} && && && && && = \boldsymbol{b}_0,\\
&T_1\boldsymbol{x} && +W_1\boldsymbol{y}_1 && && && && = \boldsymbol{b}_1,\\
&T_2\boldsymbol{x} && && +W_2\boldsymbol{y}_2 && && && = \boldsymbol{b}_2,\\
&\vdots && && && \ddots && && \vdots\\
&T_s\boldsymbol{x} && && && && +W_s\boldsymbol{y}_s && = \boldsymbol{b}_s,
\end{aligned}$$

$$\boldsymbol{l}\le\boldsymbol{x}\le\boldsymbol{u},$$
$$\boldsymbol{l}_i\le\boldsymbol{y}_i\le\boldsymbol{u}_i,\quad \forall i\in\{1,2,\ldots,s\}.$$

内点法の実装である PIPS-IPM[35] と単体法の実装である PIPS-S を含む．線形計画問題そのものが巨大で計算機の 1 ノードに問題データが収まらない大規模な問題を直接的に解く[36].

4.11 QSOpt[37], QSOpt_ex[38]

QSOpt は，巡回セールスマン問題ソルバ Concorde[39] 内で線形緩和問題を解く際に利用されている線形計画ソルバである． QSOpt は，ヘッダーファイル，ライブラリ，実行ファイルは提供されているが，ソースファイルは公開されていない．一方，厳密線形計画ソルバである QSOpt_ex のほうは，ソースファイルが公開されている.

4.12 SoPlex[40]

SCIP Optimization Suite に付属する線形計画ソルバである．近年，精度保証付きのソルバへと拡張された [3, 4]．厳密線形計画ソルバとしての利用方法は 2 節で紹介した．筆者らの知る限り，有理数形式で記述された LP フォーマットを解釈して実行する唯一の線形計画ソルバである．混合整数計画ソルバとしての SCIP は，LP Solver Interface を介して複数の線形計画ソルバを利用可能としている．線形計画ソルバとして CPLEX, Gurobi, MOSEK, Xpress に加え非商用ソルバとして CLP, QSOpt が利用できるが，SCIP では通常 SoPlex によりテストされている．線形計画問題を解くために SoPlex をリンクした SCIP を利用することができ，その場合は線形計画法の適用前に SCIP に実装されている前処理が適用される．また SCIP Optimization Suite に付属するモデリング言語 ZIMPL や，Python によるモデルの記述が可能となるので，使い勝手はよくなる反面，厳密線形計画ソルバとして利用はできないなど制限のある使い方になる．注意するべきこととして，SCIP Optimization Suite のソースコードは公開され

[31] https://www.maths.ed.ac.uk/~gondzio/software/hopdm.html
[32] http://lpsolve.sourceforge.net
[33] http://www.mipcl-cpp.appspot.com/
[34] https://github.com/Argonne-National-Laboratory/PIPS
[35] 内点法の実装に関しては，線形計画と同様の並列アルゴリズムが動作するため非線形計画ソルバである PIPS-NLP も含まれている.
[36] 大規模 2 段階確率線形整数計画問題を解くために，PIPS を利用して分枝限定法を動作させる PIPS-SBB [9]，および，分枝限定木の並列処理を含めたソルバ [10] も実験的には開発されている.
[37] https://www.math.uwaterloo.ca/~bico/qsopt/index.html
[38] https://www.math.uwaterloo.ca/~bico/qsopt/ex/index.html
[39] http://www.math.uwaterloo.ca/tsp/concorde.html
[40] https://soplex.zib.de

ているが無償使用はアカデミックユーザの研究利用のみに許可されていて商用利用に関しては有償であることが挙げられる．

5. おわりに

本稿では，線形計画ソルバの利用方法と，可能な限り現在入手可能な（独立して動作可能な）線形計画ソルバを紹介した．本稿に紹介したように，現在では利用できる線形計画ソルバに多くの選択肢がある．繰り返しになるが，ソルバの性能を知るには他人の実験結果だけに頼らず，自らいくつかのソルバを使って自分のインスタンスを解いてみることを奨励する．ベンチマークの結果は，マーケティングのために利用されがちであり，ベンチマーク結果の利用が議論[41, 42]になることもある．

実際利用するには，それぞれのソルバのデフォルトのパラメタ（たとえば，feasibility tolerance など）が，どのように設定されているかにも注意を払う必要がある．前述したが，これらのパラメタが異なれば，当然，計算時間に差がでる．まずは自分自身で，自分の解きたい問題を複数の線形計画ソルバで解いてみてほしい．

商用ソルバを利用する場合には，提供されるソルバの性能だけでなく会社のサポート体制にも注意を払う必要がある．モデル化に対してもサポートされるなら，インスタンスデータそのものを，より安定的に解けるようにモデル化を助けてくれる会社もある．適切なモデルを作ることのほうが，悪いモデルを高速に解く場合よりもはるかによいソリューションが得られる．

もし，読者が研究としてソルバの開発にも興味があるなら最新のソルバも使ってみよう．研究としても挑戦的なアカデミックソルバ（たとえば，分散メモリ環境上で動作する PIPS や，厳密線形計画ソルバ）を利用する場合は，インストールそのものが困難なソルバも多い．そのような場合でも，くじけずに開発者に状況を正確に伝えて使う努力をしてみるとよい．筆者の経験では，多くのソルバ開発者はユーザを大切にし，ユーザとのコミュニケーションを好む．なぜなら，ソフトウェアがユーザによって作られることを知っているからである．オープンソースで開発されているソルバであっても，開発者に誰を加えるかに関してはそれほどオープンではない．ある程度の信頼関係が必要なのである．問い合わせの繰り返しが，研究者として参画する機会になることも多い．だから，果敢に使ってみよう線形計画ソルバ．

謝辞　ソルバについての情報を株式会社 NTT データ数理システム田辺隆人，Mosek 社 Erling Andersen，Mathworks 社 Mary Fenelon，Zuse Institute Berlin Matthias Miltenberger，IBM 社 Roland Wunderling，SAS 社 Phillip Christophel，Lehigh University Ted Ralphs および FICO 社 Timo Berthold （敬称略）から提供いただいた．ここに記して謝意を表する．

参考文献

[1] 宮代隆平，“整数計画ソルバー入門，”オペレーションズ・リサーチ：経営の科学，**57**(4), pp. 183–189, 2012.
[2] E. Klotz, “Identification, assessment, and correction of ill-conditioning and numerical instability in linear and integer programs,” *Bridging Data and Decisions*, pp. 54–108, 2014.
[3] A. M. Gleixner, D. E. Steffy and K. Wolter, “Improving the accuracy of linear programming solvers with iterative refinement,” In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, pp. 187–194, 2012.
[4] A. M. Gleixner, D. E. Steffy and K. Wolter, “Iterative refinement for linear programming,” *INFORMS Journal on Computing*, **28**, pp. 449–464, 2016.
[5] R. Wunderling, U.S. Patent No. 9,177,256, U.S. Patent and Trademark Office, 2015.
[6] E. D. Andersen, “The homogeneous and self-dual model and algorithm for linear optimization,” Technical Report TR-1-2009, MOSEK ApS, 2009. http://docs.mosek.com/whitepapers/homolo.pdf（2018 年 12 月 1 日閲覧）
[7] R. W. Ashford and R. C. Daniel, “LP-Model: Xpress-LP’s model builder,” *IMA Journal of Management Mathematics*, **1**, pp. 163–176, 1986.
[8] Q. Huangfu and J. J. Hall, “Parallelizing the dual revised simplex method,” *Mathematical Programming Computation*, **10**, pp. 119–142, 2018.
[9] L. Munguía, G. Oxberry and D. Rajan, “PIPS-SBB: A parallel distributed-memory branch-and-bound algorithm for stochastic mixed-integer programs,” In *Proceedings of 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 730–739, 2016.
[10] L. Munguía, G. Oxberry, D. Rajan and Y. Shinano, “Parallel PIPS-SBB: Multi-level parallelism for stochastic mixed-Integer programs,” ZIB-Report, No. 17-58, 2017.

[41] https://community.fico.com/s/page/a5Q2E000000Dt0JUAS/fico1421
[42] http://www.gurobi.com/company/news/announcement