解説

MIPLIBとHans Mittelmann's benchmarks

品野　勇治

混合整数計画法 (Mixed Integer Programming: MIP) は，MIP を解くソフトウェアである MIP ソルバが大規模な現実問題を解けるようになったこともあり，現実問題を解く有用な OR の手法として広く知られるようになった．しかしながら，MIP ソルバの開発に欠かせないベンチマーク・データセットおよび性能測定方法についてはそれほど広く知られているとは言い難い．ベンチマーク・データセットは注意を払って作成しないと，多くのバイアスがかかってしまう．それらのバイアスを可能な限りのぞき，真に有用なベンチマーク・テストの結果を得るためには複数の人数で多大な労力を割く必要がある．本稿では，そのような MIP ソルバ開発の背景として重要な役割を果たしてきた MIPLIB と Hans Mittelmann's benchmarks について解説する．また，本稿において Hans Mittelmann's benchmarks は，BENCHMARKS FOR OPTIMIZATION SOFTWARE のページ (http://plato.asu.edu/bench.html) に示されているベンチマークである．

キーワード：MIP，MIPLIB，ソフトウェア，ベンチマーク

1. はじめに

筆者が所属する Zuse Institute Berlin(ZIB) は，現在，混合整数線形／非線形計画問題を解くためのソフトウェアである MIP ソルバの開発者が集まっている研究機関である．ZIB により配布されている MIP[1] ソルバである SCIP の開発者の多くが商用ソルバの開発者となり，ZIB 内にオフィスをもっている．本原稿執筆時点において，元 SCIP ソルバ開発者の 3 名が Gurobi ソルバの開発を行っており，2 名が Xpress ソルバの開発を行っている．各人は，SCIP の開発を 10 年以上続けたのちに，商用ソルバの開発者となっており，現在の SCIP 開発者の数人は既に商用ソルバ開発者になることが決まっている．この状況がいつまで継続するかはわからないが，現時点では ZIB は MIP ソルバ開発において世界の中心に位置すると考えられている．実際，混合整数線形／非線形計画問題を実際に解くこと (Computation) を研究する著名な研究者の多くは，頻繁に ZIB を訪問する．

2018 年 11 月 4 日から 11 月 7 日に開催された 2018 INFORMS Annual Meeting において，MIP ソルバ開発コミュニティにおいてターニングポイントとなり得るような出来事があった．この会議において，MIP ソルバ開発のコミュニティでは標準的なベンチマークである MIPLIB の新しいバージョン MIPLIB2017 がリリースされた．MIPLIB には，ソルバ開発者にとっては挑戦的なインスタンスなども含むが，最も慎重に設計されているのは，MIP ソルバの性能評価に利用するための Benchmark set（MIPLIB2017 では，**240** のインスタンスを含む）である．Gurobi Optimization 社は，MIPLIB2017 の代表的な作成メンバの一員であるにもかかわらず，この Benchmark set の一部である **99** インスタンスだけを選んだ計算結果に対して，MIP ソルバ開発コミュニティの中では一般的でない評価方法を利用して，Gurobi ソルバが，CPLEX ソルバより 2.69 倍速く，Xpress ソルバより 5.51 倍速いという広告を出した．また結果自体は後述の Hans Mittelmann's benchmarks に掲載されていた結果に基づいていたが，掲載した Hans D. Mittelmann は結果について「preliminary」と但し書きをつけていてまだ吟味が必要である結果であるという注釈をつけていた．

このような根拠の不明確な広告に対して，MIP ソルバ開発コミュニティは迅速に反応し，INFORMS 2018 の会議中に Twitter 等で大きな議論となった．FICO 社の Oliver Bastert と Timo Berthold は，FICO のブログの中で 2018 年 11 月 7 日に A Note on #fairbenchmarking という記事 (https://community.fico.com/s/blog-post/a5Q2E000000Dt0JUAS/fico1421) により，科学的な視点から問題点を的確に指摘している．

また，Gurobi Optimization 社の Zonghao Gu は彼

しなの　ゆうじ
Zuse Institute Berlin
Takustrasse 7, 14195 Berlin, Germany
shinano@zib.de

[1] 厳密には SCIP の扱う問題クラスは CIP(Consrtraint Integer Program) と定義され，MIP よりも広範囲の問題を扱える．詳細は，https://scip.zib.de 参照．

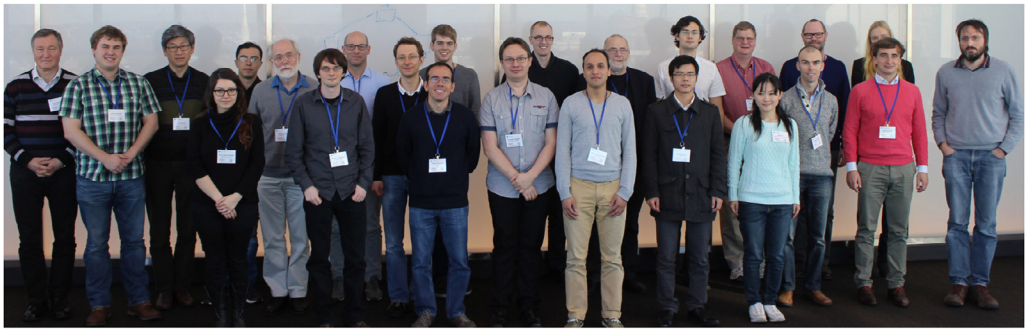

図 1　Remembering David S. Johnson (1945-2016)(http://dimacs.rutgers.edu/news_archive/memorial-johnson) に掲載されている 2014 年の第 11 回 DIMACS Implementation Challenge 参加者（左から 6 番目が David S. Johnson）

の講演の前に謝罪したことが，次の Twitter からもわかる．

> David Bernal@bernalde 2018 年 11 月 7 日 Zonghao Gu from @gurobi starts his talk clarifying the whole issue regarding the company's #MIPLIB misleading announcements. As Dan Heist says: "When you realize you've made a mistake, make amends immediately. It's easier to eat crow while it's still warm" #KnightsOfMIP #orms

筆者は，日本オペレーションズ・リサーチ学会 2019 年秋季研究発表会において講演を行い，その中で Hans Mittelmann's benchmarks に言及した．そのため，国内で誤解に基づく誤った記事がウェブに掲載されていることを知った[2]．このような記事が長期間にわたり何の訂正もされずに WEB 掲載されている現実を知り，筆者自身も MIPLIB2017 作成メンバの一員として，正しい情報を伝えるべく本稿を執筆することとした[3]．

[2] 2019 年 2 月に配信されたメールマガジンに基づく，Hans Mittelmann's benchmarks に関する日本語記事であるが，現在は ウェブ上の記事は削除されている．この記事の中では，昨年の INFORMS 2018 での出来事を知るための十分な情報源へのリンクが準備されていた．そのような事実にも関わらず，「アンフェア（不公正）な『Hans Mittelmann ベンチマーク』」や，「当教授自らは，継続的に何らベンチマーキングをしておらず倫理的にも問題で，多くの人に誤解と迷惑を掛けた責任は重く，単なる当サイト閉鎖宣言では済まされるものではありません．」などの誤解に基づく日本語での記述を含んでいた．

[3] 筆者は，上記記事の執筆者宛に，2019 年 9 月 23 日にメールを書き，記事の誤りを指摘し訂正を求めると共に，本稿執筆の動機としての記事について言及することを断っている．メールへの返事はなく，ウエブ上の記事の訂正もない．しかし，ウェブ上の記事は削除された．

2.　ベンチマーク・データセットの開発

ベンチマーク・データセットの開発は，アルゴリズム実装の性能評価において主要な役割を担ってきた．ここでは，その必要性と歴史を概観し，特に MIPLIB シリーズに関して解説する．

2.1　必要性と歴史

アルゴリズム開発が盛んになると，そのアルゴリズムを実装した際に，どの程度の規模の問題を扱えるのかを調べる研究も始まった．特に，David S. Johnson により確立された DIMACS Implementation Challenge (http://dimacs.rutgers.edu/programs/challenge/) は，アルゴリズムとその実装の性能に対する理解を深め，それらを改善するための研究を推進してきた．最悪ケースを示すアルゴリズムの理論解析は，あまりに悲観的な解析で実際にアルゴリズムを実装して利用する際の挙動とはかけ離れていることも多い．このギャップを埋めるべく，多くの代表的な計算問題に対して継続的に Implementation Challenge が実施されてきた．第 1 回の DIMACS Implementation Challenge は，1990–1991 年に開催され，課題は「Network Flows and Matching」であった．David S. Johnson が行った最後の第 11 回の DIMACS Implementation Challenge の課題は「Steiner Tree Problems」である（図 1 参照[4]）．その後も，DIMACS Implementation Challenge が継続したことは，その重要性が広く認められていることの証である．もちろん，このような Implementation Challenge には，性能評価のためのベンチマークが必要になり，そのためのベンチマーク・データセッ

[4] 筆者も参加しており左から 3 番目である．David S. Johnson 自身は，次回の challenge について，「初回に戻って Network Flow にしたい」と講演で話されていた．

トが用意されるのは必然である．ベンチマーク・データセットは，それを生成するための生成プログラムが用意されるか，あるいは，インスタンスのセットが用意されるかのいずれかである．DIMACS implementation challenge のベンチマーク・データセットは，結果の客観性が示せるため多くの論文において利用されている．

NP 困難な問題は，理論的には問題の規模が大きくなると，現実的な時間では解けなくなる問題を指す．特に，NP 困難な問題に対しては，ベンチマーク・データセットとしてインスタンスのセットが用意されることが多い．これは，インスタンス・データの特徴を利用した解法が設計されたり，逆に，恣意的に特定のアルゴリズムに対して困難なインスタンスを生成したりすることもできるため，理論的には大規模な問題は扱えないはずだが，インスタンスにどのような特徴がある場合に，どの程度の規模の問題が最新のアルゴリズム，ソフトウェア，ハードウェアの組み合わせで現実的に対処可能なのかを調べるためである．1990 年代には，多くの組合せ最適化問題に対してベンチマーク・データセットが開発された．たとえば，巡回セールスマン問題に対しては，Gerhard Reinelt が，過去に研究等で利用されたインスタンスに加えて，産業応用上生成されたインスタンスと地図上の都市から構成されるインスタンスを集めて TSPLIB [1] を 1990 年に開発した．TSPLIB には 100 以上のインスタンスが含まれ，14 都市から 85,900 都市の問題を含む[5]．また，二次割当問題に対する QAPLIB(http://anjos.mgi.polymtl.ca/qaplib/) [2] は，1991 年に最初のバージョンが公開されている．より一般的な数理計画問題のベンチマーク・データセットとしては非線形整数計画問題を対象とする MINLPLib[3]，二次計画問題を対象とする QPLIB[4]，そして MIPLIB がある．

標準的な混合整数計画問題のベンチマークである MIPLIB は，Robert E. Bixby, E. Andrew Boyd, Ronni R. Indovina により 1991 年に開発されている．その後，MIPLIB 2.0, MIPLIB 3.0, MIPLIB2003, MIPLIB2010, MIPLIB2017(https://miplib.zib.de/index.html) とデータセットが再設計されている．MIP は，その問題記述力が極めて高い．そのため，乱数により自動生成したインスタンスは，極端に困難な問題になるか，あるいは，極端に易しい問題になる傾向があり，実際に解きたい問題を解いた際の挙動を調べるには適当でない．MIP ソルバは，現実問題に対して，どの程度の規模の問題が現実的な時間で解けるかで評価される．そのため，MIPLIB は，基本的に具体的なアプリケーションを背景にもつインスタンスを集めている．理想的には，現実世界に存在するあらゆるアプリケーションを MIP に定式化したインスタンス群のそれぞれの特徴を備えたインスタンスの最小セットにしたいのである．

ここで，問題規模と性能評価に関して若干補足しておく．計算量の理論では，問題規模は入力データサイズにより測ることになるが，ここでは MIP を例として大雑把に整数変数の数で規模を測ることとして説明する．1990 年代初頭の頃には，整数変数の数が 100 を超えると大規模な問題として扱われていた．現在の MIP ソルバの性能を考えると，この規模はとても小さく思える．しかし，現在でも，0-1 変数だけによる問題を，単純な全列挙により解くことを考えると，スパコンを利用してさえ 70 変数程度が解ける問題規模の限界になると思われる[6]．つまり，その性能が，任意の問題が解ける MIP ソルバとして示されるなら，解ける問題規模は 70 変数程度になってしまう．このことから，MIPLIB のようなデータセットを，ソルバ開発のコミュニティで開発することの意味は，MIP ソルバが，今後，現実的な時間内で解けるようにする問題群を決める作業に相当する．

ベンチマーク・データセットの開発には，それぞれの最適化問題に対するアルゴリズム・ソルバ開発のコミュニティが，継続的に 20 年間以上，研究・議論してきた結果であることを強調したい．いずれのベンチマーク・データセットも，単なるインスタンスの集まりではなく，慎重に選択されている．

2.2 MIPLIB2017

筆者の私観ではあるが，最適化問題に対するデータセットとしては，MIPLIB シリーズは，最も洗練されたデータセットであると思う．

開発メンバ　筆者が，最初に MIPLIB の開発議論の様子に触れたのは，2009 年で MIPLIB2010 の開発のために研究者が ZIB に集まった際である．Robert E. Bixby, Hans D. Mittelmann を含み，主要な商用・非商用ソルバ開発者が ZIB に集まって，ほぼ終日議論

[5] 現在は，すべてのインスタンスが解かれている．本原稿執筆段階では，オリジナルのリンクは切れておりウェブ・ページが表示されなかった．ただし，Google で調べればデータは取れる．一方，TSP に対する新しいテストデータが http://www.math.uwaterloo.ca/tsp/data/index.html にある．

[6] 毎秒 1 京個の解の候補を列挙できるスパコンを利用したとして，70 変数で 1 日以上の計算時間を要する．

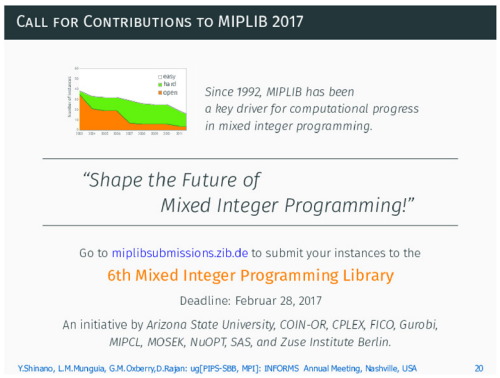

図 2　MIPLIB2017 作成メンバが国際会議などで利用していたインスタンスの投稿を促す案内（2016 年 Informs Annual Meeting の筆者のプレゼンより引用）

していた[7]．MIPLIB2017 については，さらに多くのメンバにより議論されており，開発メンバは，Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp Christophel, Mary Felenon, Koichi Fujii, Gerald Gamrath, Ambros Gleixner, Gregor Hendel, Kati Jarck, Thorsten Koch, Jeff Linderoth, Marco Lübbecke, Hans Mittelmann, Derya Ozyurt, Imre Pólik, Ted Ralphs, Domenico Salvagnin, Yuji Shinano, Franz Wesselmann, Michael Winkler である．これらのメンバは，次の組織，および，ソルバの開発メンバに属する：Arizona State University, COIN-OR, CPLEX, FICO, Gurobi, MathWorks, MOSEK, Numerical Optimizer, SAS, Zuse Institute Berlin．開発メンバによる最初の議論は，2016 年 7 月 15 日に行われた．

インスタンス・データの収集　MIPLIB2017 のインスタンス・データの収集は，公開で行われている．開発メンバのそれぞれが，国際会議や Workshop で講演を行う際にアナウンスするとともに，WEB, Twitter，各種メーリング・リストにアナウンスしている．図 2 に，MIPLIB2017 へのインスタンスの投稿を促す，開発メンバが利用していた国際会議等でのプレゼンテーションを示す．こららの活動を通して，できる限り多くのインスタンスを世界中から集める努力をしている．

Collection set と Benchmark set　世界中から集まったすべてのデータより，MIPLIB2017 の Collection set，および，Benchmark set へのインスタンスが慎重に選ばれている．何を基準に，どのように選択するかに関しては，開発メンバによる議論の結果に基づいている．選択プロセスの概要を図 3 に示す．特に，Benchmark set に関しては，benchmark-suitable instances が次のように定義され，

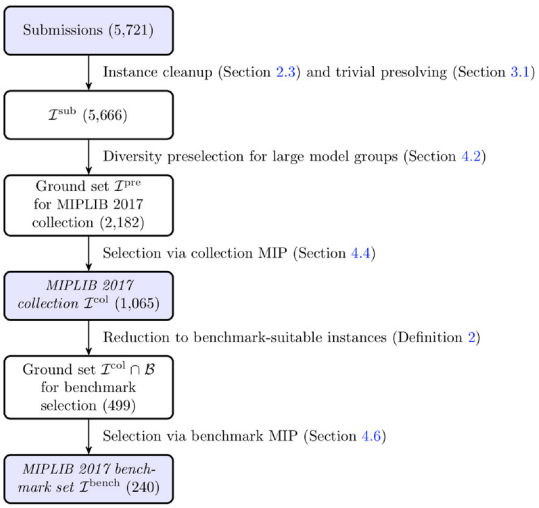

図 3　MIPLIB2017 の Collection set，および，Benchmark set 選択プロセスの概要．（）内の数字は，選択されたデータ数を示す（論文 [5] より引用）．

Definition 1（Benchmark-suitable instance）**.** We call an instance $i \in \mathcal{I}$ *benchmark-suitable* if

1. it can be solved by at least one considered solver within 4 hours;
2. it requires at least 10 seconds with 50 % of the solvers;
3. it has a constraint and objective dynamism of at most 10^6(see Section 3.3);
4. the absolute value of each matrix coefficient is smaller than 10^{10};
5. the results of all solvers on i are consistent (see Section 3.7);
6. it has no indicator constraints (see Section 3.4);
7. it is bounded;
8. the solution (objective) value of i is smaller than 10^{10};
9. it has at most 10^6 nonzero entries.

これを満たすインスタンスの中から，MIP モデルを解くことによりインスタンスが選択されている．上記，Definition 1 における “considered solver” は，具体的には次のソルバ・バージョンである：CBC 2.9.8, IBM

[7] 当時，筆者自身は議論に参加していないが，ディナーに誘われて話を聞く機会を得た．

CPLEX 12.7.1, Gurobi 7.5.1, MATLAB R2017b, MOSEK 8.1.0.30, SAS/OR 14.2, SCIP 4.0.0, FICO Xpress 8.2. つまり，提出された 5,721 のインスタンス全てを，これら全てのソルバにより 4 時間の時間制限で解いたのち，図 3 の工程を経て選択されている．上記，Definition 1 および図 3 における Section 番号は，MIPLIB2017 論文 [5] における Section 番号である．詳細は，MIPLIB2017 論文を参照されたい．ここで注意していただきたいのは，**MIPLIB2017 の Benchmark set は，240 インスタンスのすべてを解いた結果により評価されるように設計されている**点である．そのようにインスタンスが選択されているのである．

このように関係するコミュニティのメンバを集結して議論し，オープンにインスタンスデータを収集したのち，洗練された手法により Benchmark set が開発されてもバイアスはかかる．しかし，数年間にわたる時間と労力をかけて，バイアスを排除する努力を最大限にした結果得られている Benchmark set であることは読者にも理解されると信じる．

3. Hans Mittelmann's benchmarks

Hans D. Mittelmann のサイト：DECISION TREE FOR OPTIMIZATION SOFTWARE (http://plato.asu.edu/guide.html) は有名である．特に，今回，問題視されているのは，このサイト中の BENCHMARKS FOR OPTIMIZATION SOFTWARE (http://plato.asu.edu/bench.html) の部分であるが，まずはサイトの目的と構成から説明したい．

3.1 サイトの目的と構成

DECISION TREE FOR OPTIMIZATION SOFTWARE サイトの目的は，最初のページに示されているように，

> Welcome! This site aims at helping you identify ready to use solutions for your optimization problem, or at least to find some way to build such a solution using work done by others.... We do not aim at giving an overview over existing commercial products and recommend one of the other guides for that....

であり，次の情報により構成されている．

Problem & Software: software sorted by problem to be solved

Benchmarks: collection of testresults and performance tests, made by us or others

Testcases: example files ready to use with existing software, in different formats

Books & Tutorials: a short list of introductory texts, some online

Tools: software which helps formulating an optimization problem or simplifying its solution

WebSubmission: some software can be used directly via the net thanks to implementors who make their computing facilities available to you

Other sources: for more information provided by others

このようなサイトを継続して更新している背景にあるのは，Hans D. Mittelmann の**「ソフトウェアは正しくテストされなければならない」**という信念であるように感じる．サイト全体は，あらゆる最適化問題と，それを解くソフトウェア，および，ベンチマーク・データの所在とベンチマークに関する書籍・論文がまとめられているとともに，他の研究者が行ったベンチマークへのリンクも示されている．Hans Mittelmann's benchmarks はサイト中のごく一部である．

3.2 MIPLIB2017 の Benchmark set によるベンチマーク

Mittelmann's benchmarks は，

- COMBINATORIAL OPTIMIZATION (Concorde-TSP with different LP solvers),
- LINEAR PROGRAMMING,
- MIXED INTEGER LINEAR PROGRAMMING,
- SEMIDEFINITE/SQL PROGRAMMING,
- NONLINEAR PROGRAMMING,
- MIXED INTEGER QPS AND QCPS,
- MIXED INTEGER NONLINEAR PROGRAMMING,
- PROBLEMS WITH EQUILIBRIUM CONSTRAINTS

のカテゴリにおいて，さまざまなベンチマークが行われている．MIPLIB2017 の Benchmark set によるベンチマークは，Hans Mittelmann's benchmarks 中，さらにごく一部になる．それぞれについて，Hans D. Mittelmann は，それぞれのコミュニティと交流しながら，それぞれのコミュニティが推奨するベンチマークを行っている．論文の中には，独自の評価を行っているものも多い．また，計算環境が統一されているわ

けでもない．したがって，このように第三者によって統一された環境において数値実験の客観的結果が得られることは，いずれのソルバ開発者にとっても，他ソルバとの性能差を知る機会となり好ましい．

MIPLIB2017 の Benchmark set の開発には，Hans D. Mittelmann 自身もメンバとして加わっているが，先に示したように，Benchmark set そのものは開発メンバの議論に基づいて客観的に選択されている．ベンチマークの実行スクリプトは，MIPLIB2017 のアーカイブに含まれており，Hans D. Mittelmann によって作られているものではない．これは，MIP ソルバによってデフォルト・パラメタの値が異なるため，同じ条件で実行するためには，パラメタの設定が必要になるためである．また検証用プログラム Solution Checker もあり，得られた解が共通の許容範囲の数値誤差内に収まっているかを多倍長計算により慎重に確認する．同プログラムも MIPLIB2017 のアーカイブに含まれている．Hans D. Mittelmann は，MIPLIB2017 の提供するスクリプトを実行しており，読者も MIPLIB2017 に含まれるアーカイブを利用することにより自身の手で各ソルバの性能を調査できる．

計算結果の集計に関して，現在，MIP ソルバのコミュニティでは，Tobias Achterberg が彼の博士論文 [6] によって提案した以下の Shifted Geometric Mean を利用するのが標準である．

Definition 2. 値の集合 $N := \{t_1, t_2, \ldots, t_n\}$ と shift value s が与えられたとき，N に対する *shifted geometric mean* は，

$$SG(N) = \left(\prod_{k=1}^{n} (t_k + s) \right)^{\frac{1}{n}} - s,$$

となる．

このように集計するのは，短時間で解けるケースの影響を少なくするためである．計算時間を集計する際には，$s = 10$ を利用するのが一般的である．時間制限を付けて計算するため，制限時間内に解けなかったインスタンス，および，計算が異常終了したインスタンス，または，誤った解（誤った解とは言い切れないので，他のソルバによる解とは “missmatch” と表示される）の場合の扱いは，Hans Mittelmann’s benchmarks では計算時間を制限時間としている．これは，全体の集計時間としては本来よりかなり小さな値になりうる点に注意が必要である．また集計値が制限時間の設定に依存することもここの議論でわかる．

(F)SCIP/spx]-6.0.2: FiberSCIP (SCIP+SOPLEX on 1 thread)
CBC-2.10.0: CBC
GLPK-4.65: www.gnu.org/software/glpk/glpk.html
LP_SOLVE-5.5.2: lpsolve.sourceforge.net/
MATLAB-2019a: MATLAB (intlinprog)
MIPCL-2.5.0: MIPCL
SAS-OR-15.1: SAS

Table for single thread, Result files per solver, Log files per solver

Table for 8 threads, Result files per solver, Log files per solver

++
Unscaled and scaled shifted geometric means of run times

All non-successes are counted as max-time.
The third line lists the number of problems (240 total) solved.

1 thr	CBC	SCIP	MATLB	MIPCL	GLPK	LP_SOL	SAS
unscal	2113	1213	3351	1736	5032	5335	743
scaled	2.84	1.63	4.51	2.34	6.77	7.18	1
solved	88	119	59	99	23	20	147

the best commercial solvers would have a geomean of about .2 to .3

8 thr	CBC	FSCIP	MIPCL	SAS
unscal	1585	988	1246	580
scaled	2.73	1.70	2.15	1
solved	102	140	115	157

the best commercial codes would have a geomean of about .2

図 4 MIPLIB2017 に対する Hans D. Mittelmann の 2019 年 8 月 19 日掲載ベンチマーク結果 (http://plato.asu.edu/ftp/milp.html).

原稿執筆時点における MIPLIB2017 の最新のベンチマーク結果を図 4 に示す[8]．“unscale” の行が，Benchmark set 240 インスタンスの計算結果を Shifted Geometric Mean で集計した値であり，2 時間の制限時間内で解けなかったインスタンスなどは，2 時間の計算時間として集計されており，$s = 10$ である．“scaled” の行は，最も “unscale” において小さな値の結果を 1 として正規化された値が示されている．ここで注意すべきことは，制限時間内に解けたインスタンス数が少ないケースのほうが，Shifted Geometric Mean の値が小さくなりうる点である．これは，制限時間内に解けなかったインスタンスなどの場合に制限時間が使われているためである．そこで，“solved” の行に，240 インスタンス中，何インスタンスが制限時間内に解けたかが示されている[9]．これらの値は，240 インスタンスの結果を集計した値である．各インスタンスに関して，どのソルバが，どの程度速く解いたかを見るには，“Table for * threads” のリンク先を見ればよい．また，各ソルバによる結果の詳細は，“Result files per solver” のリンク先に示されており，さらに，各ソルバの実行ログが “Log files per solver” に置いてある．これらの

[8] ベンチマークサイトが閉鎖されているわけではない．CPLEX, Gurobi, Xpress の結果が削除されただけである．
[9] Hans D. Mittlemann 教授に解けたインスタンス数も示すように要求したのは筆者である．相手は研究者なので，合理的な要求には答えくれる．

内容を調べれば，**各インスタンスによって，各 MIP ソルバの性能が大きく異なる**ことを確認できる．

MIP ソルバの挙動として，*Performance Variability* [7] が広く知られている．これは，数学的に同一の問題でも，たとえば変数・制約式の順番が変わっただけで，同じ MIP ソルバにより同じ動作環境上で解いても，計算時間が大きく異なるという挙動である．MIP ソルバ内部では，その動作途中で複数のアルゴリズム上の選択が必要になる際，評価関数の値を利用する部分が多くある．これらの評価関数の値が同じであった場合には，データがメモリ上にどのように配置されているかによって，選択されるアルゴリズムが変わる．このことが最終的には計算時間の大きな違いに繋がる．一時は，この点を考慮して，1 インスタンスに対して変数・制約式の順番のみを変えた 5 種類のデータでの実験を行っていたこともある．おそらく，あまりにも長時間の計算を要するので，できなくなったのであろう．

4. 不適切な性能評価結果の作成による性能比較

これまで説明してきたように，MIPLIB2017 の Benchmark set は極めて慎重に設計されている．加えて，Hans D. Mittelmann は，公正なベンチマークはどうあるべきなのかを多くの研究者と議論し，自らも精力的に実験結果を公開し実践によって示してきている．

「1. はじめに」で述べた出来事が MIP ソルバ開発のコミュニティとして受け入れられなかった問題の本質は，ある一社がマーケティングのために，**性能評価結果を作った**点である．本稿の読者が，図 4 の結果のページを訪れ，各インスタンスにおける結果を調べれば，前述のようにインスタンスによって，各 MIP ソルバの性能が大きく変わることを確認できる．したがって，240 インスタンスの一部の結果だけを使うなら，特定の MIP ソルバに対して高評価を与える性能評価結果を作ることは容易である．さらに，制限時間内に解けなかったインスタンスなどに対する計算時間の値として，制限時間を 10 倍した値を使う PAR10 [8] と呼ばれる手法を利用して Shifted Geometric Mean を求めた．こちらも MIP ソルバ開発のコミュニティでの標準ではないので，性能差を大きく見せたとも考えられる[10]．

その広告では，Benchmark set の中で MIPLIB2017 に新たに加わった 99 インスタンスの結果を Hans Mittelmann's benchmarks 結果から選んで集計したとなっていた．しかし，過去に存在したインスタンスの実行結果が，MIP ソルバのバージョンが上がると遅くなることは常にあり，Benchmark set は 240 のインスタンスの実行結果を使うように設計されている．このような Benchmark set の一部を使う場合は，その理由を十分に説明する必要があるが，今回のケースにおいてはマーケティングのために恣意的に選んだと言わざるをえない[11]．

加えて，この性能評価結果を作ったのは，Hans D. Mittelmann ではない．Hans D. Mittelmann は，サイトにある実験結果を，このように不適切な方法で利用をされたことを知ったのち，即座に Gurobi の計算結果を削除した．

5. おわりに

MIPLIB2017 は問題の投稿・選定の過程において多くの研究者およびソルバ開発者の努力によって成立したプロジェクトである．その目的は単にソルバの優劣を比較する，という以上に MIP の研究を前進させるところにあり，整数計画の研究に携わる人にはできるだけ利用してほしい．

一方で，ユーザ側の立場からみた場合，それぞれの扱う問題のインスタンに対しては自身でテストする重要性も忘れてはならない．各 MIP ソルバの開発には，MIPLIB インスタンスを含めたソルバ独自のベンチマークインスタンスに対して開発・チューニングを通常行っている．その目的はそのベンチマークに対して「平均的に」良くなることを目指している．そのためユーザはそれぞれが扱う問題のインスタンスに対して，どの MIP ソルバが適しているのかについては個別に性能評価をするべきである．このような作業も，MIPLIB2017 に含まれているベンチマーク用のスクリプトを若干変更することで，比較的楽に行える．

また現在 MIP ソルバ開発のコミュニティにおいて標準的に利用されている Shifted Geometric Mean が決定的でもないうえ，ユーザの利用目的によっては評価方法自身も本来変更するべきである．並列ソルバの場合には，さらに考慮するべき点が増える．筆者らも，これら性能評価に関する問題点などを論文 [9] として

[10]制限時間内に解けなかったインスタンスなどに対する計算時間の扱い方に関して，MIP ソルバの開発コミュニティで今のところ合意はない．制限時間を使うのは過少評価ではあるが，10 倍することにも妥当な根拠はない．

[11]都合の良い結果だけを選択することは，cherry picking と呼ばれ，（数値）実験結果の評価において最も行っていはいけないこととされている．

まとめたので参考にしてほしい[12]．特に，論文 [9] では，資源消費（資源は時間・計算 core など）という概念に基づく性能評価のフレームワークを提案している．基本効率 (Baseline *Efficiency*) を，資源のセットを固定した際の性能とし，スケーラビリティ (*Scalability*) は，特定の資源量が変更された際の Efficiency の変化で測定するという提案である．詳細は論文での定義を読んで頂きたいが，多くの評価方法があるということを知って頂くために，本稿では未定義であることを承知で以下に記す．Efficiency を時間の消費量として測定する場合，Time to Optimality, Time to Fixed Gap, Time to First Solution, 計算の処理量として測定する場合，Number of Nodes（分枝数），Number of Bounding Problems Solved, Iteration Count, 計算の進捗量として測定する場合，Gap,Gap Integrals, Tree Size Estimation を定義している．また，論文 [9] では結果の集計方法・可視化方法に関しても議論している．

Hans Mittelmann's benchmarks の結果は，これまでも商用・非商用 MIP ソルバのいずれにおいても，それらのマーケティングに利用されたきたのは事実である．しかし，今回のように性能評価結果を恣意的に作成されて利用されたことはこれまでにない．MIP ソルバ開発のコミュニティでは，昨年 11 月に大きな議論になったが Gurobi の対応は素早かった．MIP ソルバ開発のコミュニティの研究者は議論を好み，間違いだと考えれば迅速に対応する．ZIB が開発している MIP ソルバ SCIP のトップページにも，以前は "The fastest non-commercial MIP Solver" として Hans Mittelmann's benchmarks の結果をグラフにして表示していた．しかし，このようなグラフを掲載することが，新たな研究者の参入を妨げるのではないかという議論の末，今では表示していない．

本記事で述べた事情により Hans Mittelmann's benchmarks では残念ながら MIPLIB2017 については商用ソルバの結果については本稿執筆段階では掲載はされていない．しかしながらアカデミックソルバの性能などを知るには有用であるし非線形計画など MIP 以外の結果も載っているので是非参考にしてほしい．

謝辞　本原稿を一読し，有意義なコメントをくださった藤井浩一，梅谷俊治，檀寛成，榊原静（敬称略）に感謝します．

参考文献

[1] G. Reinelt, "TSPLIB: A traveling salesman problem library," *ORSA Journal on Computing*, **3**, pp. 376–384, 1991.

[2] R. E. Burkard, S. Karisch and F. Rendl, "QAPLIB: A quadratic assignment problem library," *European Journal of Operational Research*, **55**, pp. 115–119, 1991.

[3] M. R. Bussieck, A. S. Drud and A. Meeraus, "MINLPLib: A collection of test models for mixed-integer nonlinear programming," *INFORMS Journal on Computing*, **15**, pp. 114–119, 2003.

[4] F. Furini, E. Traversi, P. Belotti, A. Frangioni, A. Gleixner, N. Gould, L. Liberti, A. Lodi, R. Misener, H. Mittelmann, N. Sahinidis, S. Vigerske and A. Wiegele, "QPLIB: A library of quadratic programming instances," *Mathematical Programming Computation*, **11**, pp. 237–265, 2019.

[5] A. Gleixner, G. Hendel, G. Gamrath, T. Achterberg, M. Bastubbe, T. Berthold, P. M. Christophel, K. Jarck, T. Koch, J. Linderoth, M. Lübbecke, H. D. Mittelmann, D. Ozyurt, T. K. Ralphs, D. Salvagnin and Y. Shinano, "MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library," preprint is in Optimization Online, submitted to MPC, 2019.

[6] T. Achterberg, "Constraint integer programming," der Technischen Universitat Berlin, D83, 2007, https://opus4.kobv.de/opus4-zib/files/1112/Achterberg_Constraint_Integer_Programming.pdf

[7] 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**, pp. 103–165, 2011.

[8] S. Kadioglu, Y. Malitsky, A. Sabharwal, H. Samulowitz and M. Sellmann, "Algorithm selection and scheduling: Principles and practice of constraint programming," CP 2011, pp. 454–469, 2011.

[9] S. J. Maher, T. K. Ralphs and Y. Shinano, "Assessing the effectiveness of (parallel) branch-and-bound algorithms," preprint is in Optimization Online, submitted to MPC, 2019.

[12] この種の論文は議論を通して執筆するため時間を要する．2017 年 5 月の Gerogia Tech でのセミナーでの講演から始まり，同年 SIAM Conference on Optimization (OP17), SIAM PP18, ISMP2018, UG Workshop 2019 での講演・議論を通して執筆されている．その結果，サーベイが長くなり過ぎたので，新提案は別論文に書くこととした．よって，性能評価の良いサーベイ論文になっていると思う．