最適化研究における数値実験を中心としたアプリケーション駆動研究サイクル

品野　勇治

最適化研究におけるアプリケーション駆動研究サイクルを紹介する．アプリケーション駆動研究サイクルは，学術機関での研究と企業における研究成果の利用とのつながりを良くする点では優れている．一方で，ソフトウェア開発・維持に多大な労力を要するため，日本の大学や研究機関における実施には困難さが伴う．ZIBにおいてアプリケーション駆動研究サイクルが，比較的うまく機能している背景を説明する．また，日本においてアプリケーション駆動研究サイクルを活性化するための第一歩として，論文投稿時に，論文中の数値実験に利用した全データ提出の義務化を提案したい．

キーワード：最適化問題，アプリケーション駆動研究サイクル

1. はじめに

本稿では，最適化研究における数値実験を中心としたアプリケーション駆動研究サイクルを紹介する．本稿において紹介する内容は，著者自身が Zuse Institute Berlin (ZIB)[1]において仕事をするなかで，参加したり，直接見たりしてきた研究・教育活動の紹介である．

最初に，ZIB と著者の所属する最適化部署に関して紹介したい．ZIB は，Fast Computer, Fast Algorithm をスローガンとするベルリン州立の研究機関である．研究機関名は，世界初のプログラム可能コンピュータ Zuse Z3[2]の開発者でベルリン生まれの Konrad Zuse の名前に由来する．最適化部署の長である Martin Grötschel 教授は，研究所の現所長であるとともに，IMU (International Mathematical Union) の Exective Committee メンバほか，多くの研究組織において指導的な役割を果たされている．「数学を現実社会を良くする方向に可能な限り活用したい」という Grötschel 教授の強い情熱に牽引されて，現在の ZIB の最適化部署が存在するように著者には思える．現在の ZIB の最適化部署は，多くの研究が企業との共同研究による外部資金により運営されており，実際に社会で利用されるプログラムの開発まで行っている．汎用的に利用可能な最適化ソフトウェア一式は，SCIP Optimization Suite[3] としてソースコードを含めてすべて公開している[4]．その中心となる SCIP (Solving Constraint Integer Programs) は，混合整数計画問題を含む，より広いクラスの最適化問題を扱えるように開発された最適化ソルバフレームワークである．その最適化ソルバとしての性能およびソフトウェアとしての質も，研究機関としての開発物としては，極めて高く維持されている．

ZIB は最適化を扱う研究機関としては，理論面の研究と実社会へのつながりのバランスが極めて良い研究機関と言える．また，SCIP Optimization Suite の開発は，ベルリンにある大学だけでなく，ダムシュタット工科大学，アーヘン工科大学とも共同で開発されている．ZIB の正規スタッフとして SCIP Optimization Suite の開発グループで長期間働くという経験から，短期の滞在経験者よりは広く見聞きしてきたつもりではあるが，本稿の内容には著者の私感を伴っている点には注意していただきたい．

2. アプリケーション駆動研究サイクル

著者が理解している「アプリケーション駆動研究サイクル」の特徴は以下である．

1. 具体的な現実問題の解決が研究対象
2. 必要な現実データを注意深く収集
3. 問題解決のための理論とアルゴリズムの構築
4. 現実データを利用して実際に問題を解くプログラム開発

しなの　ゆうじ
Zuse Institute Berlin, Department Optimization, Takustr 7, D-14195 Berlin-Dahlem, Germany

[1] http://www.zib.de/
[2] チューリング完全であることが 1998 年に証明されている．
[3] http://scip.zib.de/
[4] アカデミックでは自由に利用できるが，商用利用は有料である．以前は，ZIB Optimization Suite として配布されていたが，現在は，ZIB 外の開発者も加わっているので名称変更された．

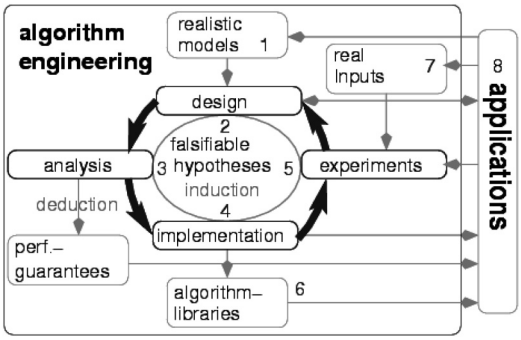

図 1　DFG Priority Programme 1307 Algorithm Engineering (2007–2013) における研究サイクル

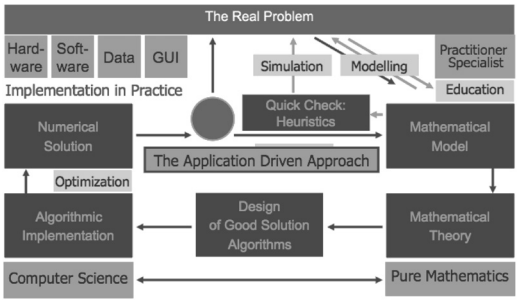

図 2　Problem Solving における研究サイクル

5. 現実データによる数値実験の実施
6. 数値実験結果の検証と現実問題に対する解決策の妥当性の検証
7. 現実問題が解決できた場合には，その手法の一般化やプログラムのライブラリ化による再利用可能性の検討

「研究サイクル」と呼ばれるのは，この一部，あるいは，全部が繰り返されることを意味する．また，そのサイクルの中心には，現実データによる数値実験の実施と検証の実施が必然であり，そのため「反証可能である」という仮定が成立しなければならない．

著者はアプリケーション駆動研究サイクルを，ドイツのいくつかの研究プロジェクトにおいて見てきた．基本的には上記が各プロジェクトの特徴に応じて図式化されて説明されていた．例えば，図 1 は，Algorithm Engineering プロジェクトにおけるアプリケーション駆動研究サイクルである．図 1 で示されているサイクルは，洗練された最新の Algorithm を常に実際のプログラムの中で利用可能とするための工学的なプログラム開発手法とみなせる．図 1 はプロジェクトの WEB ページや，研究成果として出版されている文献 [1] の表紙となっている．特に，文献 [1] では，ほぼ図 1 の各ボックスに 1 章を設けて，行うべき内容がまとめられている．図 2 に示されているサイクルは，ZIB の最適化部署での Problem Solving の取り組み方を示したものである．最初にモデリングとヒューリスティックによるクイックチェックのサイクルがある点，および，最後に教育を含んでいる点に注意されたい．いずれの図においても，論文執筆を目的とした数値実験のためのソフトウェア開発と比べると，大規模なソフトウェア開発が要求される．Algorithm Engineering では，開発するソフトウェアはライブラリとして再利用することを当初から意図している．このようなソフトウェアは，一般的に再利用可能な構造をプログラム中に埋め込むので，規模は単機能を実現するプログラムよりもかなり大きくなる．特にライブラリやソフトウェアフレームワークの設計は，対象が本質的に整理されていないと困難である．また，Problem Solving では，GUI (Graphical User Interface) や，解の可視化も実装に含まれている．こちらは，研究成果が本当に利用されるためには，開発手法から得られる解をユーザ側の立場で眺める方法の提供が必然だからである．いずれも，研究サイクルを繰り返すことで，最新の研究成果の理論と実践のギャップを徐々に埋めていくような工学的なソフトウェア開発手法を模索し，研究サイクルそのものを洗練していこうとしている．

このように文章や図だけを眺めると，とても当たり前に思える研究スタイルである．しかし，研究機関において現実問題を対象とした最適化問題を実際に解くことを前提とするのは，かなり困難である．まず，実際に解くなら，問題の定義，データ収集，プログラム開発などに莫大な時間と労力を要するが，それらは研究者としての業績につながりにくい．現実問題なのでユーザが存在し，解決策はユーザが納得するものである必要がある．対象は素直に定義されるような問題ではないほうが一般的であり，ユーザですら解決すべき問題が本質的にはわかっていないことも多い．解くべき問題が定義できたとしても，**本当に必要なデータ**の収集に莫大な時間を要することも稀ではない．実際，解くべき問題を定義し，定義された問題を解くために必要なデータを必要な精度で収集するだけに終わったプロジェクトも存在したようである．研究者であれば，論文の執筆が主たる仕事となるが，理論面での魅力に欠ける場合も多く論文にもなりにくい．このような状況を理解したうえで，何年も現実問題を解くことに取り組む意志が必要となる．よって，実施にあたっては，それでも現実問題に取り組むことを指向する環境と，

それを行う意志を持つ人材を育てる教育が必要となる．次章以降，アプリケーション駆動研究サイクルに連動していると思えた，予算申請や教育を紹介したい．

3. DFG Research Center Matheon でのプロジェクト予算申請

DFG Research Center Matheon (mathematics for key technologies)[5]は，ベルリンの3つの大学（ベルリン自由大学，フンボルト大学，ベルリン工科大学）と2つの研究機関 (WIAS, ZIB) によって運営されている仮想的な研究組織である（建物があるわけではない）．名称から想像できるように，応用数学を対象とした研究組織であり，プロジェクトは以下の研究対象カテゴリに対して行われている（A〜V のカテゴリ識別＋番号でプロジェクトが識別される）．

A Life sciences
B Networks
C Production
D Electronic and photonic devices
E Finance
F Visualization
V Education, Outreach, Administration

予算申請後のプレゼンテーションは公開で行われる．ベルリンの大学の学生たちも，ベルリンで行われる，あるいは，すでに実施されている研究内容を知るために参加していた．申請書類は事前に審査されており，プレゼンテーション時には審査結果に対する反論もできる[6]．最盛時[7]のプレゼンテーションは，1プロジェクト申請に対するプレゼンテーションが15分で，午前9時から午後6時まで，昼休みを除くとほとんど休みなく3日間であった．1プロジェクトの予算規模は，日本の科研費 A から科研費 S に相当する額だと記憶しているが，単位は金額ではなく，プロジェクトを推進するうえで必要となる研究者を何人・何カ月雇用するかで示される．一般的なプロジェクト内容に関する質問に加えて，以下の内容の提示が必ず求められていた．

- 数学分野に対する貢献
- 産業界に対する貢献
- ほかのプロジェクトとの連携／関連
- 生産物

生産物の多くはソフトウェアであり，新規プロジェクトの場合は，「どのようなソフトウェアを開発するのか」継続プロジェクトの場合は，「**プロジェクト終了後，どのようにソフトウェアを維持するか**」の提示が求められていた．このような研究費のあり方は，アプリケーション起動の研究サイクル推進には有効である．実際，予算で雇用される研究員は基本的に博士課程の学生であり，研究機関における主戦力である．

アプリケーション駆動は，ニーズから研究を進めるアプローチである．一方，研究者がすでにもっているシーズをアプリケーションへつなげるというアプローチのプロジェクト提案も存在する．シーズからのアプローチは，最盛時の Matheon におけるプロジェクト提案では採択されにくかった．理由は，アプリケーション駆動で研究を進めている研究者のほうが現場を知っているため，議論になると論破されてしまうケースが多いのである．注意していただきたいのは，研究としてのシーズからのアプローチの優劣の議論ではないことである．優れた研究であることが理解されている場合もある．しかし，Applied mathematics を対象としているので，産業界で実際に利用することを指向し，実現可能性の高いプロジェクトに Matheon では予算をつけるというポリシーである．よって，最終的には現実問題の解決とは乖離するなら，研究内容が優れていても，Industrial motivated mathematics であり，Matheon でのプロジェクト提案にはそぐわないという考え方である．つまり，Matheon の目標とプロジェクト目標の整合性の優先度が高かった．Matheon でのプロジェクトには適さなくても，優れた研究プロジェクトであれば，それに適したほかの予算枠があるはずである．

4. Combinatorial Optimization at Work

Combinatorial Optimization at Work[8]は，基本的には Martin Grötschel 教授による集中講義である．最も大規模に行われたのは，2009年の ZIB での講義で，ZIB のスタッフを総動員して17日間，連日午前9時から午後6時頃まで行われた．講義内容，および，講義のプレゼンテーションの多くは，今でも Combinatorial Optimization at Work Ⅱ の WEB ページ[9]から得られるので参照されたい．基本的には数学科の博士課程の

[5] http://www.matheon.de
[6]「これだけオープンに審査されれば公正な審査になって政治的な力は働かないですね」と当時尋ねた．返事は，審査会がオープンでも政治的な力関係は働くとのことであった．しかし，少なくとも学生を含めて審査結果を予測したり，研究内容に関するプレゼンに対して議論しているので，極端な例は排除されていると思う．
[7] 現在は DFG の予算によるプロジェクトとしては終わっている．
[8] 過去の講義の内容は，http://co-at-work.zib.de/参照．
[9] http://co-at-work.zib.de/berlin2009/

学生を対象としている集中講義ではあるが，極めて実践的でソフトウェアの利用方法やプログラム開発が多く含まれていることがわかる．

ここでは例として，Thorsten Koch による 2009 年 9 月 22 日の Real World Data[10] および，2009 年 10 月 9 日の Real World Data Again[11] について紹介する．9 月 22 日の講義では，集中講義期間中に参加者（100 名程度）の講義室における最適座席割当問題を実際に解くことが提案される．その目的のために，参加者全員から講義室の各座席の好みを数値化したデータの電子メールによる提出を求めている．各座席の好みを数値化する際の条件，および，プログラムによる処理を前提として，データ提出時のメール文面やフォーマット等が細かく指定された（詳細に関しては，脚注に講義プレゼンへの URL を示したので，そちらを参照されたい）．その後は，毎朝，講義の前にデータの収集状況と，メール中に示されたデータにおいて整合性の取れてないメールに対しては，修正し再提出することが要求された．その様子は，最後の 10 月 9 日の講義資料に示されている．結局，毎日訂正箇所や修正を必要とする対象者へ修正し再提出を促したにもかかわらず，最適化問題を解くために必要とされる全データは集まらず，プロジェクトが失敗する例を実践をもって示す講義となっている．

最後の Martin Grötschel 教授の Summary, future perspectives and closing では，数学科の学生を対象にしているのにソフトウェアの利用方法やプログラミングが多かったことに関しても説明されている．数学が強力なツールであるということを示すには，それを使って現実問題を解決して見せるしかない．そのためには，ソフトウェアの利用方法やプログラミングが避けられないなど，一般的な話だけでない．講義では，最適化問題を極めて適切に解いたとしても，それが現場で利用されるかどうかは別問題である．人間は，それほど合理的に活動しない場合もあるなど，経験に基づく話には説得力があった．

5. ソフトウェア開発に関して

ここでは，ソフトウェアを研究所内で開発している点について触れたい．ソフトウェア開発とその維持には莫大な労力を必要とする．最適化研究を対象とすると，ソフトウェア開発により論文が書けるというわけでもない．ZIB では，ソフトウェアは研究所内で開発している．これを可能とする背景を説明したい．

主戦力は博士課程の学生であるが，ドイツでは一般的に研究員として雇用されているので日本における博士課程の学生と経済面では異なる．日本のように年度末に修士学生が全員そろって修了するわけではない．修士論文は，学生自身が「半年後には仕上がる」と判断したとき，自ら提出期限を決めて宣言し，自ら決めた期限に修士論文を提出する（その後，修士論文が合格になっても口頭試験もある）．よって，年齢もさまざまで，学生本人が納得するまで研究する習慣は，修士課程が修了した時点で備わっている[12]．博士課程も同様ではあるが，何年も雇用されるかどうかは指導教授による．ZIB の最適化部署の博士課程の学生は，ドイツ国内の大学の学生と比べても博士取得年齢が高い．これは，Martin Grötschel 教授が，自分自身で選んだ学生を雇用し続ける教授だからと言える．

ZIB の最適化部署におけるソフトウェアは，数学に長けた学生が，自らソフトウェア開発を学び開発している．開発を通して，新たに解決すべき課題を見いだした場合は，自ら理論構築し解決するか，あるいは，その専門知識を持つ人材を世界中からスカウトしてくる[13]．したがって，それぞれ得意分野を持った開発者集団と言えるし，開発そのものが組織的に行われていると言える．また，修士学生や博士学生の在籍年限はとても長く修了時期も異なるので，世代が重なる時期も長くなり一斉に学生がいなくなることもない．そのため，ソフトウェア開発に関する引き継ぎも比較的世代間でスムーズに行われ，継続的なソフトウェア開発を可能としている．研究所内に，ソフトウェアの継続的な開発を可能とする環境があることが，アプリケーション駆動の研究サイクルを成功させる基盤になっている．ただし，現在のデータ整理から解の可視化まで，ほぼ何もかも研究所の人材で行うという部分は，行き過ぎに感じるところもある．一方で，ZIB の最適化部署で開発経験を積んだ研究者は，何もかも行っているので即戦力として企業で活躍できるだけの開発力を有し，就職に困ることはないようである．つまり，博士課程の期間が長くてもキャリアパスとしての選択肢は

[10] http://co-at-work.zib.de/berlin2009/downloads/2009-09-22/2009-09-22-1600-TK-Real-World-Data.pdf
[11] http://co-at-work.zib.de/berlin2009/downloads/2009-10-09/2009-10-09-0900-TK-Real-World-Data.pdf

[12] こちらはポジティブな表現であるが，一方でいつまでも学生をやっているケースがあるという見方もできる．
[13] 一般的に，研究者つながりで候補が決められ，ZIB のスタッフの誰かが会いに行くか，候補者を ZIB に招聘して講演してもらいスタッフで採用に関して議論する．

一般的には多い．

日本では，大学や研究機関で大型の予算を取ったプロジェクトにおけるソフトウェア開発は，プログラム開発を企業へ委託するケースも多いと思う．プログラム開発が委託されると，理論研究とソフトウェア開発は完全に切り離される．開発者側は，理論研究に対する方向性やビジョンを引き継ぐわけではなく，形式的に書かれた仕様[14]に従ってプログラム開発が行われる．この場合，アプリケーション駆動の研究サイクルを素直に機能させることは難しい．開発物であるソフトウェアの再利用性だけでなく，プロジェクト終了時には開発したソフトウェアの継続利用さえ困難な場合も多い．実際，研究機関において 10 年以上継続的に開発・維持されているソフトウェアは，研究者個人の意志で開発されているものであり，プロジェクトの成果として存在するものは極めて少ないのではないだろうか．一般的に，プログラム開発を企業へ委託した場合，開発を通しての新たな研究課題の発掘はあまり期待できない．

日本に限らず，予算の切れ目がソフトウェアの寿命に直結する場合も少なくない．また，主として開発していた博士課程の学生の修了に伴い，ソフトウェアが廃れることも多い．Matheon でのプロジェクト申請の仕組みなどは，継続的な開発を促す仕組みと捉えることができる．

6. 研究機関におけるアプリケーション駆動研究サイクルの機能

研究機関や大学におけるアプリケーション駆動研究サイクルがうまく回るようになると，当該研究機関や大学は，当該アプリケーションに関連する研究分野での優位性が確立される傾向にある．その結果，特定の業種や企業と特定の研究機関や大学の結びつきが強くなり，研究分野への新規参入が難しくなる．公共研究機関や大学で開発された技術が，特定企業で独占されることは好ましくない．また，新規参入を拒む研究分野ができるのも好ましくない．このような状況を避ける手段の一つは，研究成果の公開であると思われる．実際，ZIB では共同研究の契約にあたって以下に注意するように教育している．

- 知的財産の扱い（特許）
- サードパーティコードの扱い（ソフトウェアを販売する場合）
- 保守契約は結んではいけない
- 論文を書く権利の保持
- 学会発表の権利の保持
- ほかの企業との共同研究の権利の保持
- 契約後の研究の継続の権利の保持（競合企業との共同研究）

そして，開発したソフトウェアや，収集したデータは公開するように試みている．このように，企業との共同研究であっても，研究成果の公開を可能とする契約により研究が遂行されているので，公共研究機関としての役割を果たしている．それでも，特定のアプリケーションに関連する研究分野での優位性が確立さることには変わりはない．成功した研究機関に，アプリケーション分野に関連する技術が集積されることになり，技術的には研究機関や大学が先導する仕組みとなっている．

7. 日本において可能なこと

最適化分野におけるアプリケーション駆動研究サイクルを，その研究スタイルだけを日本の特定の研究機関や大学で導入しても，これまでの説明から機能しないと予測できる．日本においても，アルゴリズムの開発と計算機の進歩によって，論文として，現実の最適化問題を解いた結果が報告されるケースも増えているように思う．それらの論文は，アプリケーション駆動の入り口になると考えられる．しかし，残念ながら，「反証可能である」という仮定は成立しないケースのほうが多いように思われる．論文であるので，手法の新規性により採択されているとは思われるが，ほかの手法との優位性の確認は一般的には困難である．著者の経験として，論文の筆者へ追試するために，論文中の数値実験に利用されたデータをお願いした結果，会社の規定で外には出せないとのことであった．対象問題が NP 困難であった場合には，その性能は強くデータに依存するので，少なくとも同じデータで比較しないと，論文での提案手法とほかの手法との比較は困難である．論文誌掲載と同時に，論文中で利用している数値実験のデータを論文誌の WEB ページ上に公開することは，現在ではそれほど手間はかからないと思われる．企業のデータをそのまま出せない場合には，ある種の妥当な変換後のデータで数値実験が実施されていても，データが公開されずに検証できない状態よりは良い．少なくとも，論文の読者が自ら検証できるだけのデータは公開するべきである．仮にデータ数は少なくても公開された際の効果は大きい．例としてナーススケジューリング問題を取り上げたい．ナーススケジューリング

[14] 最先端の数学を扱っている場合に，委託先のプログラマが理解できるような仕様を書くこと自体大変だと思う．

問題は，池上敦子先生がフィールドワークとして日本の医療現場でのデータを収集した．池上敦子先生自身による [2] および，当時の既存論文 [3] の数値実験データは，ご自身の WEB ページ[15]において公開され，その後に立ち上げられた勤務表作成のベンチマークサイト[16] においても公開された．その結果，これらのデータを用いた妥当な比較が行えることとなり，ナーススケージューリング問題に対する数値実験を通してのアルゴリズム開発に貢献している．そこで，比較的簡単で大きな効果が期待できる以下を提案したい．

> 数値実験を伴う最適化問題の解法に関する論文では，論文投稿と同時に，論文中の全数値実験データの提出を義務づける（あるいは，数値実験データ生成用プログラムと，利用したパラメタ設定の提出）．論文採択時には，全データ（あるいは，実験データ生成プログラム）も同時に公開されることに必ず同意を得る．

もちろん長期的には教育制度まで変更していくという考え方はあるだろうが，早急な教育制度の変更は，教育現場の混乱につながり教員の研究者としての生産性を著しく阻害する可能性が高い．教育制度の変更に関しては慎重に検討し，計画的に時間をかけて変更されるべきであるので，その変更は当面期待できない．また，Combinatorial Optimization at Work のような集中講義の実施も，日本の現状においては困難ではないかと思う．実際に，著者は 2013 年の 3 月日本開催を目指して，その 1 年以上前に Grötschel 教授の訪日予定も確保していただいて実現に向けて努力した．日本の OR 学会関係者の 5 名以上と連絡も取り相談した．Combinatorial Optimization at Work の実施にあたっては，明確な達成点を決めるとともに，参加者が自ら，その達成点に向かってソフトウェアを使うかプログラムを書くかにより何か実際に PC で計算するような，演習を含む集中講義であることに意味がある．実現できなかった主たる理由は，そのような講義への参加を希望する学生が日本では集まらないということであった．このような状況は，急に変わるものではない．

予算申請の制度に関しては，一部，Matheon の考え方を導入してみても良いのかもしれない．特に，Matheon のプロジェクト申請のように，継続的な開発によりノウハウを蓄積するような仕組み作りを奨励するような予算配分方法は検討する価値はある．日本の場合は，継続的に開発／維持されているソフトウェアは，研究者個人の努力によるところが大きい．よって，具体的には，新規のプロジェクトよりも，研究者個人やプロジェクトが，ソフトウェアを公開し，継続的に開発／維持しておりユーザのいるようなプロジェクトを奨励するような予算配分があっても良いと思う．つまり，予算申請時には，以下の提示を求め，すでに継続的に開発／維持してきているプロジェクトに対して予算を配分する．

- ソフトウェアの公開年
- 当該ソフトウェアのユーザ数見積り
- ソフトウェアの研究分野におけるインパクト
- ソフトウェアの今後の発展性
- 当該プロジェクト終了後のソフトウェアの維持管理方法

新規参入を拒まないように，1 プロジェクト期間 3 年程度とし継続申請は 1 回のみとする．このような予算配分が継続的に，例えば 10 年間行われることがわかっていれば，ソフトウェア開発／維持を積極的に行うプロジェクトが育つ可能性はある．

8. おわりに

本稿では，最適化研究分野における数値実験を中心としたアプリケーション駆動研究サイクルについて紹介した．このような研究サイクルがドイツ国内で一般的であるかどうかに関しては，著者は広く調査しているわけではないのでわからない．ただし，本稿で紹介した研究サイクルは，ZIB とドイツ国内外の関連研究機関や大学における適用が進んでいると感じる．それは，Combinatorial Optimization at Work のような集中講義を何度も実施して，その実績を広める努力をしているためである．Combinatorial Optimization at Work Ⅱ への参加者は，23 カ国から 105 名となっており，もちろんドイツからの参加者が最も多いが，EU 圏のドイツ国外からの参加者も相当数得ている．加えて，同様の Summer School はドイツ国外でも実施されており，若手研究者間のネットワークづくりに貢献している．最適化分野に限れば，ある程度，世界的な広がりをみせている研究スタイルであると感じる．

最適化研究分野では，研究機関や大学によるアプリケーション駆動研究サイクルの実施は，学術機関での研究の成果と実社会の要望をうまく結びつける研究スタイルである．最適化手法は特定のアプリケーションと結びついて威力を発揮するケースが多いので，一般

[15] http://cleo.ci.seikei.ac.jp/ atsuko/DATA/data.html
[16] http://www.cs.nott.ac.uk/ tec/NRP/

的には最適化分野には向いている研究スタイルとも言える．それでも，研究機関や大学においては，開発に要する時間と労力を考えるとアプリケーションとしての向き不向きはある．比較的，社会インフラに関連するプロジェクトは成功しているように思うが，ZIB でもすべて成功しているわけでもない．開発の主戦力が博士課程の学生であることを考えると，短期のプロジェクトでは博士論文につながらないので困る．一方，長期のプロジェクトの場合には，現実社会に適用される技術進歩の方向性が，当初構築していた最適化モデルにそぐわなくなったケースなどもあったようである．このように失敗したケースですら，「現実を学ぶ」ということに関しては良い機会となっている．そのような経験を含めて，企業へのキャリアパスがあるのかもしれない．

アプリケーション駆動研究サイクルは，学術機関の独立性が損なわれるという見方もできる．著者が学生だった頃には，「企業からの紐付きの研究は一切しない」というスタンスの先生方もいらっしゃった気がする．本当に優れた研究は，今すぐに役に立たなくても 100 年後や 1,000 年後に評価されるような研究だと思う．アプリケーションを全く想定せず，知的好奇心にかられて行われる研究があっても良いし[17]，すべての最適化分野における研究がアプリケーション駆動の研究スタイルに合致するはずもない．実用性を求める方向にだけ予算が配分される状況も好ましくない．

アプリケーション駆動研究サイクルに限らず，求める研究スタイルが効率的に実施されるためには，大学や研究機関，あるいは各種研究費によるプロジェクトに対して，本質的に何を求めるかを明確にするだけでなく，その目的のために，とりまく環境がいかに有機的につながるかが成功の鍵になる．本稿が，日本におけるバランスのとれた良い研究環境の構築に関して考える際の一助となることを願う．

参考文献

[1] M. Mueller-Hannemann and S. Schirra, “Algorithm Engineering,” *LNCS Vol. 5971*, Springer, 2010.

[2] A. Ikegami and A. Niwa, “A Subproblem-centric Model and Approach to the Nurse Scheduling Problem,” *Mathematical Programming*, **97**, 517–541, 2003.

[3] H. H. Millar and M. Kiragu, “Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming,” *European Journal of Operational Research*, **104**, 582–592, 1998.

[17] ただし，取り組んでいる研究者が感じる面白さを，著者のような凡人にもわかるように伝える努力はしてほしい．