COIN-OR：ORにおけるオープンソースの事例

Robin Lougee-Heimer

ソフトウェアはORにとって根底をなすものである．計算の面でのORのソフトウェアへの依存状態は，この分野において，どのようにソフトウェアを開発，管理，配布するかが重要であるということを示している．本稿では，ORにおけるオープン・ソフトウェアの開発，管理，配布に関する議論についてまとめる．2000年に開始したORコミュニティのための「オープンソース」・ソフトウェアを促進するための先進的な活動であるCOIN-OR (the Computational INfrastructure for Operations Research) について紹介する．現在，COIN-ORは，教育目的の非営利団体によって管理され，INFORMSが世話人を受け持っている．COIN-ORのウェブサイト (http://www.coin-or.org/) では，オープン・ソフトウェアと標準とデータの共同開発と配布の基盤が整備されており，研究，教育，応用のための20以上のオープンソース・プロジェクトに関する本拠地となっている．

キーワード：COIN-OR，オープンソース

1. はじめに

計算の面でのORのソフトウェアへの依存状態は，この分野において，どのようにソフトウェアを開発，管理，配布するかが重要であるということを示している．2000年に，ORコミュニティのためのソフトウェア開発，管理，配布に対して，「オープンソース」によるアプローチが可能であるかの試みが開始された．COIN-OR (the COmputational INfrastructure for Operations Research) はORにおけるオープンソース・ソフトウェア推進のための先進的な活動である．開始から7年経過し，COIN-ORは世界最大のORの専門家グループにより主催され，教育目的の非営利財団により管理されることになり，20以上のプロジェクトの本拠地となって急成長しているコミュニティである．第2節では，オープンソース・アプローチへの動機についてまとめ，第3節ではオープンソースの定義と哲学についてレビューする．第4節では，COIN-ORの目標，歴史および現状について報告する．本稿は，文献[5]を加筆，修正したものである．

2. 典型的なソフトウェア開発と配布の実際

ピアレビューされたORに関する公刊物に掲載されている定理や証明，アルゴリズムは，計算機実験の結果によって補強されることが多い．例えば，INFORMSの旗艦誌であるOperations Researchには，2001年に掲載された論文のおおよそ75%に何らかの計算実験の結果が含まれている．このようにアルゴリズムや定理はピアレビューを経て公開されるが，ソフトウェアは通常はそうならない．このことは以下に示すような悪影響を及ぼしている．

- **結果は再生不能である．** 計算実験の結果はアルゴリズムの複雑さと実装の効率性に依存する．実装の詳細が公表されないと，結果を再生することが困難もしくは不可能である．例えば，もしもカーマーカーがコンピュータに実装したものが公的に利用可能であったならば，彼の研究はもっと早く再現されたはずであり，彼の学術的貢献にまつわる混乱はずいぶんと軽減されたはずである[2][8]．
- **比較は不公平である．** 既存の実装にアクセスできないと，研究者が自分の新しいアプローチとすでに公表されているアプローチを比較しようとする場合，既存のアルゴリズムを再度自分で実装しなければならない．この結果は，「他人のアイデアを自分が実装したもの」と「自分のアイデアを自分で実装したもの」という不公平な比較でしかない．
- **ツールは繰り返し作られる．** ソースコードへアクセスできないと，研究者は既存のコードを再利用できずに新たに最初から作らなければならない．多くの場合科学の進歩が，地道な積み重ねで進むという性質を考えると，研究者にとって再度作るということは時間と努力を浪費して

IBM Research, T. J. Watson Research Center
robinlh@us.ibm.com

いるといえる．

- **知識移転は限定的である．** ソースコードにアクセスできないと，ORの研究者はその分野で最も優れたソフトウェアのアイデアから学んだり作り出したりすることができない．実装状態が不明確では，知識移転の機会が喪失される．
- **標準の欠如が共同作業を抑制する．** ソフトウェア配布を常設の場所から行うことは，ソフトウェアの標準化にとって大変重要である．現在標準と見なされているMPSフォーマット（Mathematical Programming System file format）ですら商用の最適化ソルバーによって解釈に違いがある．今日，多くのアルゴリズムが商用のソルバーに実装されている．商用ソルバーはその入出力形式とデータ構造を利用するため，そして組み込みエンジンとして特定の部分問題を解くために使用される．商用エンジンはその製品のAPI（application programming interface）を利用して呼び出される．これにより実装が特定の製品に依存してしまい，同じ製品を利用できない研究仲間との共同作業を妨げることになってしまっている．

これらの結果は，個人がOR研究においてソフトウェアの開発や利用を困難にし，遅延させてしまっている．コミュニティにおいて，理論に関しては公開されている文献から便益を摘み取ることができても，公開された支援ソフトウェアを配布しないことで，同様の便益を遺失していることに対する歴史的展望については文献[5]を参照されたい．

3. オープンソースの選択

3.1 オープンソースの定義

オープンソースの哲学は，独立したピアレビュー，迅速なソースコードの進化を支持することで，ソフトウェアの信頼性と品質をより高めようというところにある．この哲学は，非伝統的な方法で著作権を利用するということが，現実には先進的である．従来，「ソフトウェアを購入する」というときには，実際には著作権保持者からソフトウェアを利用するためのライセンスを購入する．今日では，購入者は通常，前もってコンパイルされたライブラリあるいは実行プログラムを受け取るが，それはソースコードではない．このことは，購入者はそのプログラムを行使することのみしか許されていないということを意味する．ソースコードがなければ，購入者はバグ修正をすることも，そのソフトウェアを修正することもできない．それとは対照的に，オープンソースの場合は，ユーザはソースへアクセスすることができ，規約でも修正や再配布が許されているので，コードを修正したり再配布することができる．この節では，オープンソースについて簡単に紹介する．オープンソースに関するディスカッションやビジネス面での分析に関しては文献[11][12]を参照されたい．

技術的には，「オープンソース」という単語は，非営利団体のOpen Source Initiative（OSI）によって定義・推進されているソフトウェア・ライセンスのカテゴリーである．用語の誤用を避けるために，OSIは認証プログラムを運用している．「OSI Certified」のラベルのあるソフトウェアは，Open Source Definition（http://www.opensource.org/）が満たすライセンスの下で配布されている．Open Source Definitionのversion 1.9では，単にソースコードにアクセスできることよりも広い，以下の10項目の基本的な基準が明記されている．(i)再配布は自由であること，(ii)ソースコードにアクセスできること，(iii)派生物を妨げない，(iv)作者のソースコードはオリジナルの状態を保つこと，(v)利用者やグループによる差別をしないこと，(vi)利用する分野による差別をしないこと，(vii)ライセンスの配布，(viii)特定の製品のみにライセンスを与えないこと，(ix)他のソフトウェアを制限しないこと，(x)ライセンスは技術的に中立であること．

本稿執筆時点では，OSIのウェブ・サイトには60以上のOSI認証を受けたライセンスのリストがある．すべてのOSI認証を受けたライセンスはOpen Source Definitionを満たす共通の特性を持っているが，同様にオープンソース・ライセンスはそれぞれ特有の特徴も持っている．例えば，すべてのオープンソース・ライセンスはソースコードの再配布を許可しなければならないが，いくつかのライセンスでは再配布が必要条件である．Apache®[1]とBerkeley System Distribution（BSD）のライセンスでは，ソースコードの配布は認められているものの，コンパイルしたものや派生物については義務ではない（これらはオープンソース・ライセンスの最も制約の少ないものである）．逆に，GNU Public License（GPL）ではそれらを必要としている．

[1] Apacheは Apache Software Foundationの登録商標

おそらく，最も重大な相違と論点は，ライセンスが「ウィルス的」であるかないかではないか．つまり，ライセンスを受けたソフトウェアの派生物や修正部分に対して同じライセンスが留保するかどうかということであろう．この条項の入ったライセンスは繁殖するような伝染経路があるため「ウィルス的」であるといわれる．そのようなライセンスの下でソフトウェアをコンパイルするためには，ライセンスを取らなければならないことになる．つまり，GPLはウィルス的であり，Mozilla™[2] Public Licenseはウィルス的ではないといえる．

オープンソースはパブリック・ドメインと同義語であると広く誤解されている．これら2種類のライセンスは明らかに異なる．パブリック・ドメインのソフトウェアとは違い，オープンソース・ソフトウェアには明らかな著作権がある．また，オープンソースはバイナリ・ファイルのみが配布されるシェアウェアやフリーウェアとも異なる．さらに，「アカデミック利用のみフリー」のものとも異なり，オープンソースは多様で最も広い範囲からの参加が可能であり，利用者の区別をしていない．

3.2 オープンソース・コードの開発

オープンソース・ソフトウェア・プロジェクトの典型的な成功（例えば，Linux®[3] OS）では，ユーザの中から自然発生的にボランティアの開発者のバーチャルなコミュニティが立ち上がっている．ユーザはウェブからソース・コードをダウンロードしそのまま利用するか，あるいはそれぞれの目的に応じて修正して利用する場合もある．ユーザはバグ修正したり，拡張機能をつけたり，新しいプラットフォームに移植したりすることもある．通常彼らは（場合によってはライセンスによる必要条件により）彼らなりの修正をベースコードに加えて配布するという形で貢献をする．非常に多くの開発者が同時に作業をするため，コードの修正は急速に進んでいく．

バーチャルのコミュニティは価値観を共有しておりピラミッド型の構造となっている．ピラミッドの中でのそれぞれのメンバーのポジションは，その人の貢献の価値によって決まる．底辺にはたくさんのユーザ，中位には開発者が，そして頂上には，プロジェクトの公式な配布コードの変更を管理している少数のコア・メンバーがいる．コア・チームは総意もしくは多数決ルールにより意思決定をするかもしれないし，最高権威を持つ「善意の独裁者」が方向性を決定することもある．オープンソース・コミュニティは自我と評判の上で成り立っており，開発者はその仕事に誇りを持つボランティアである．ソフトウェアは所有者を有さず，メンテナンスをするメンバーがいるだけである．バグは隠されず公開される．公開された交換が高く評価され，情報を秘匿することは好ましくない．

3.2.1 オープンソース・プロジェクトの管理

一般には，オープンソース・プロジェクトは，少なくとも，製品バージョンと開発バージョンという2通りのコードを配布する．製品バージョンは，メンテナンスの変更のみが含まれているもので，比較的安定している．それに比べて，開発バージョンは不安定である．アクティブなプロジェクトでは，一日に何度も更新されたりもする．この急速なコードの進展はバーチャルなコミュニティの中で並行して開発が進められた結果起こる．それぞれのボランティアの開発者は同じソースコードを同時に編集する．こういった，表面上は実用的とはいえないプロセスを管理するために，バージョン管理システムが利用される．

バージョン管理システムの中枢は，基本の配布コードを保存しておくリポジトリである．個々のユーザは，リポジトリに保存されているコードを各々のコンピュータにコピーして使う．基本の配布コードの変更は，リポジトリに書き込みを許可された人たちのみが可能である．優れたバージョン管理システムでは，ソース・ファイルの変更の履歴を，差異を保存することでうまく記録している．このようなシステムでは，互換性を保つようにコミットを順次適用するようにして，ユーザはロックなしで同時に修正作業を行うことができる．最新のアップデートの状態から，ローカルの修正されたコードをさらにアップデートするために，ユーザは2つのバージョンを同期させるコマンドを実行する．アップデートすると，ユーザのローカルのコピーに現在のリポジトリ・レベルに更新する「パッチ」が当てられる．もし，バージョン管理システムが自動的に差異を解消することができないならば，競合のメッセージが表示される．競合状態はユーザが手作業で解消しなければならない．オープンソース・プロジェクトで広く使われている管理システムにCVS（Concurrent Versions System）がある[1]．同じような多

[2] Mozillaは Mozilla Foundationの商標
[3] Linuxは Linus Torvaldsの米国およびその他の国における登録商標または商標

くのツールもオープンソース開発で使われており，それ自身もまたオープンソース・プロジェクトである．

3.3 ORとオープンソース

このような非同期でバーチャルのボランティア・コミュニティが，高性能，高品質，高信頼性で安全なコード生産できたということは，直感に反すると最初は思われたかもしれない．しかしそれはできたのである．多くのインターネット・サイトがオープンソースを利用して稼動している（おおよそ50%程度のウェブサイトがオープンソースのApache® HTTPによるサーバで，35%がMicrosoftのIISを利用している[9]）．オープンソースのパラダイムの恩恵は，成功した多くのプロジェクトにより証明されている．これらの恩恵はOR研究におけるソフトウェアの開発，配布における典型的な悪影響に対処している．ピアレビューのためのソースコードの公開と，オープンソース・ライセンスによる急速な進展により，計算機実験の結果を再現することができ，公正にアルゴリズムの性能評価を行うことができる．また，最良のプログラムが保存され，コードを再度実装する手間は最小限に抑えられる．さらに，実装上で新しく得られた知識を移転可能にし，共同作業とソフトウェアの標準化を進めることができる．

オープンソースがORコミュニティにとって魅力的な選択肢であっても，それが万能薬であるわけではない．オープンソースはコンピュータ科学の分野に端を発しており，ORにも計算的な要素が大きいとはいえ，2つのコミュニティの文化や特徴には異なる点もある．例えば，オープンソースはボランティアの開発者の多いコミュニティから恩恵を受けるが，ORは比較的専門的な領域であり，開発者の数はそれほど多くない．成功したオープンソース・プロジェクトはソフトウェアのスタックが低いレベル（例えばLinux® OS）についてのものが典型的であり，OSのソフトウェアについては，ソフトウェア・スタックが中高位のアプリケーション領域に属する．オープンソースの可能性とこの領域における進化を推進することができるかを実証するために，COIN-ORは始動した．

4. ORのためのオープンソース

4.1 COIN-ORの始まりとリポジトリ

COIN-OR（The COmputational INfrastructure for Operations Research）はORコミュニティのためのオープンソースを推進するために始動したものである[3]～[7][13]．COIN-ORの趣旨は，公開された文献がOR理論の「リポジトリ」として存在するのと同様に，ORソフトウェアのリポジトリを提供することであり，文献と同様の恩恵をコミュニティにもたらすことをねらっている．リポジトリはコミュニティなしでは構築も持続もすることができない．COIN-ORの最終目的は，教育を行い，気づきを促し，議論を刺激し，開発者とユーザを後押しし，そしてORのオープンソース・コミュニティを構築することである．

http://www.coin-or.org/のリポジトリは，さまざまな独立して管理されている「プロジェクト」により構成されている．メーリング・リストやバグの追跡，ウェブ・ページやその他必要な基盤といった関連するサービスはオープンソース・ライセンスの下で，ソフトウェア・プロジェクトを開発，配布しようとするOR研究者が利用可能である．それぞれのプロジェクトは，そのプロジェクトのすべての面（例えば，機能性，信頼性，ドキュメント）の進行に関して責任を持つ「プロジェクト管理者」によって進められる．プロジェクト管理者は，活発で効果的に管理された高品質のプロジェクトを作りあげるようにガイドラインとその手続きを示す．以下に示すリストは，これまでに参加したボランティアにより作られた現在利用可能なプロジェクトである．個々のプロジェクトの詳細については，http://www.coin-or.org/で参照することができる．

- **BCP**. Branch-Cut-Price Framework：混合整数線形計画のための並列分枝切除価格アルゴリズムのためのフレームワーク．
- **Bonmin**. Basic Open-source Nonlinear Mixed INteger programming：一般の混合整数非線形計画問題のための実験的なC++コード．
- **BuildTools**. COIN-OR Unix開発者用ツールとドキュメント．さまざまなCOIN-ORプロジェクトをLinux®, Unix, Cygwin上で環境管理しコンパイルするためのツール．
- **CBC**. COIN-OR Branch and Cut：線形計画問題のための分枝カット・ライブラリ．
- **CGC**. COIN-OR Graph Classes：ネットワーク表現とアルゴリズムのコレクション．
- **CGL**. Cut Generator Library：切除平面を生成するのためのライブラリ．
- **CLP**. COIN-OR Linear Program solver：シンプレックス法ソルバー

- CoinBinary. COIN-OR Binary Distributions: COIN-ORプロジェクトのコンパイル済みバイナリ配布ファイル.
- CoinMP. CLP, CBC, and CGLのための軽量なAPIとDLL.
- CoinUtils. COIN-ORプロジェクトのためのユーテリティ，データ構造，線形代数.
- CoinWeb. COIN-OR Web Services: COIN-ORのウェブ・ページのリポジトリ，Tracの情報など.
- COPS. COIN-OR Open Parallel Search Framework：線形計画ベースの分枝切除価格法を含む並列分枝木探索アルゴリズムのためのフレームワーク.
- CppAD. C++関数の微分を行うツール.
- CSDP. 半正定値計画のための内点法.
- DFO. Derivative-Free Optimization：微分を利用できない場合に用いる一般の非線形計画問題のためのパッケージ.
- DyLP. Dynamic LP：動的シンプレックス法のためのツール.
- FLOPC++. C++の代数的モデル言語.
- GAMSlinks. GAMS (General Algebraic Modeling System) とCOIN-OR内のソルバーとのリンク.
- Ipopt. Interior-Point Optimizer：一般の大規模非線形最適化.
- LaGO. Lagrangian Global Optimizer：非凸混合整数非線形問題の大域解を求める.
- MSVisualStudio. さまざまなCOIN-ORプロジェクトをMicrosoft Visual Studioコンパイラでビルドするためのツール.
- NLPAPI. Non Linear Programming API：非線形計画問題の定義，解法のサブルーチン・インターフェイス.
- OBOE. Oracle Based Optimization Engine：実行可能集合や目的関数の台 (support) などの一次条件をユーザが与える凸問題の最適化.
- OSI. Open Solver Interface：線形および混合整数計画法を呼び出す単一API.
- OTS. Open Tabu Search：禁断探索法のためのフレームワーク.
- SMI. Stochastic Modeling Interface：不確実性下の最適化.
- SVM-QP. Support Vector Machine Quadratic Programming：二次計画法によるサポート・ベクター・マシン.
- SYMPHONY：混合整数線形計画法のライブラリ.
- VOL. Volume Algorithm：主問題の近似解を計算するための劣勾配アルゴリズム.

4.2 最初の7年間の進展

ORの専門家が興味を持つようなオープンなツールのための活発なコミュニティ主導のファウンドリ (foundry) を目標として過去7年間堅実に発展してきた. IBM基礎研究部門が先導し，2000年8月のthe 17th International Symposium for Mathematical Programmingにおけるプレゼンテーション[10]と組織ミーティング，そしてプロジェクトのウェブサイトの立ち上げから，最初の一歩が始まった. ソフトウェアのリポジトリはIBMが所有していたサーバ上にあった4つのプロジェクトから広がり，最初のコア・チームは2つの組織の代表として7名のメンバーにより構成されていた. 本稿を執筆している2007年時点では，COIN-ORのリポジトリは，INFORMSのサーバに置かれ，25のプロジェクトを有し，8名の委員が8つの組織を代表する，教育目的の非営利のCOIN-OR Foundation Inc. により管理されている.

COIN-ORの成長をプロジェクトの数で測定することは簡単ではあるが，それはあまりに単純化しすぎている. というのは，既存のプロジェクトの進行と，既存のプロジェクトのユーザに対するインパクトを見落としてしまうからである. COIN-ORのソフトウェアの利用状況を測定することはさらに困難である. COIN-ORリポジトリのソフトウェアはプロジェクトのウェブサイト上から必要に応じて利用可能である. だれでもウェブにアクセスでき，COIN-ORリポジトリをウェブ・インターフェイスもしくは日次更新されるtarファイル (UNIXのテープ・アーカイブ・フォーマット) でダウンロードすることができる. したがって，ソフトウェアが使われているかどうか知るための，またどこにどのようにあるかを正確に追跡するための正式なメカニズムはない. インターネット・トラフィックの統計以上に，利用に関しては (ユーザがバグや専門的問題に出くわしたときに起こる) メーリング・リストでの議論のやり取りや個人的なコンタクトから推定できる. 本稿の執筆時点では，コミュニティのメーリング・リスト全体について1,027のユニー

ク・メール・アドレスから合計2,003の登録がある．

ソフトウェアの利用に関する一つの定量的尺度は文献に引用された数である．しかし引用は普通COIN-ORの特定のプロジェクトに対してであり，全体を対象としていない．いくつかのプロジェクトでは引用を追跡している．例えば，CSDPプロジェクトはこのプロジェクトを利用した公刊論文のリストを保持しており，20以上の文献がプロジェクト（https://projects.coin-or.org/Csdp/wiki/CSDPUsed）に列挙されており，COIN-ORが目標としているソフトウェア再利用の促進を実証している．

研究者の間でのCOIN-ORの興味レベルの代理となるものが，関連するカンファレンスでの活動の度合である．2001年の年次大会では，INFORMSのオープンソースに関する最初のクラスタにCOIN-ORチームが招待された．6つの異なる組織からの9名の発表者が5つの研究発表を行い，分枝切除価格法に関するオープンソースのツールに関するワークショップが行われ，ILOG, Dash Optimization, IBM Maximal Software, AMPLなどを含めたMPSフォーマットの新たな方向性を探るためのパネル・ディスカッションが行われた．2006年のINFORMSの年次大会においても同様のクラスタが組織され，20の組織を代表する58名の発表者による42の研究発表があり，2nd Annual COIN-OR Cup Competition prizeの表彰があった．さらに2006年には，the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) がCOIN-ORについて最初のワークショップを行った．4日間のワークショップでは，アメリカ，カナダ，ヨーロッパ，南アメリカから11の企業，2つの行政機関，21の大学を代表し75名の参加者を得た．

企業の中でCOIN-ORを利用している証拠を見つけることはさらに困難である．2007年に新たに寄せられたYahoo! Research, Royal Dutch Airlines/Air France, SmartFolio, SOSlibからのユーザ・レポートが，継続的な進展を示している．さらに包括的なユーザ・レポートのリストはCOIN-ORのウェブサイトの2007 Annual Reportに掲載されている．

4.3 COIN-ORへの参加

COIN-ORでは，ユーザ，開発者もしくはアイデア創発者などすべてのレベルで，参加・貢献いただける方をお待ちしている．既存のプロジェクトを拡張していく以外にも新しいプロジェクト（例えばモデル言語，ビジュアル化，シミュレーション，表計算ソフトウェアのプラグイン，ソルバー）やそれをリードする人材も歓迎している．COIN-ORのコードは，実務家・学者・学生が，ビジネス・研究・教育のために利用可能である．ORのためのオープンソースに興味を持っている人はメーリング・リストに加入いただき，COIN-ORのウェブサイト（http://www.coin-or.org/）にアクセスいただきORのオープンソースのビジョンを実現化するのにご協力いただければと思う．

謝辞 本稿に関して，コードと熱意，アイデアのつまったCOIN-ORに対する基盤管理者，開発者とユーザの貢献に対して感謝します．

（訳：生田目崇　専修大学）

参考文献

[1] Concurrent Versions System, http://www.cvshome.org

[2] J. J. H. Forrest and J. A. Tomlin, "Implementing interior point linear programming methods in the Optimization Subroutine Library," *IBM Systems Journal*, 31 (1), 26-38 (1987).

[3] Robin Lougee-Heimer, "The COIN-OR Initiative: open source software for optimization," *Proceedings of CP-AI-OR '01*, Wye, England, pp. 307-319.

[4] Robin Lougee-Heimer, Francisco Barahona, Brenda Dietrich, J. P. Fasano, John Forrest, Robert Harder, Laszlo Ladányi, Tobias Pfender, Theodore Ralphs, Matthew Saltzman and Katya Scheinberg, "The COIN-OR Initiative: open source accelerates operations research progress," *ORMS Today*, 28 (4), October 2001.

[5] Robin Lougee-Heimer, "The Common Optimization INterface for Operations Research," IBM *Journal of Research and Development*, 47 (1), 57-66, January 2003.

[6] Robin Lougee-Heimer, "COIN-OR Ready to Roll in Pittsburgh," *ORMS Today*, 33 (5), October 2006.

[7] Robin Lougee-Heimer, "Making Heads or Tails of COIN-OR," *ORMS Today*, 34 (1), February 2007.

[8] Gill Murray, M. Saunders, J. A. Tomlin and M. Wright, "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method," *Mathematical Programming*, 36, 183-209 (1986).

[9] Netcraft, http://www.netcraft.com, September 2007.

[10] William Pulleyblank, Brenda Dietrich, John Forrest and Robin Lougee-Heimer, "Open source for optimization software," *International Symposium for Mathematical Programming* (*ISMP*), Atlanta, Georgia, August 2000.

[11] Eric S. Raymond, *The Cathedral and the Bazaar*, O'Reilly, 1999, http://www.catb.org/~esr/writings/cathedral-bazaar/hacker-history

[12] Eric S. Raymond, *The Magic Cauldron*, June 1999. http://www.catb.org/~esr/writings/magic-cauldron

[13] Matthew J. Saltzman, "COIN-OR: an open-source library for optimization," in Soren S. Nielsen (ed.), *Programming Languages and Systems in Computational Economics and Finance*, Kluwer, Boston, forthcoming.