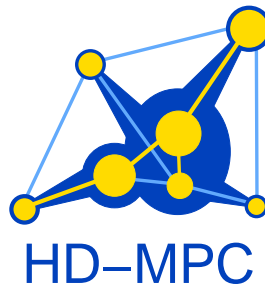


SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]



Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC



Deliverable Number:	D4.1.2
Deliverable Type:	Report
Contractual Date of Delivery:	01/09/2009
Actual Date of Delivery:	01/08/2009
Title of Deliverable:	Overview, toolbox and tutorial manual of existing state-of-the-art distributed optimization algorithms
Dissemination level:	PU
Workpackage contributing to the Deliverable:	WP4
WP Leader:	Moritz Diehl (KUL)
Partners:	TUD, KUL, USE, UWM
Author(s):	Dang Doan, Tamás Keviczky, Bart De Schutter

Table of contents

Executive Summary	3
1 Introduction	4
2 Distributed and parallel optimization software packages	5
2.1 DistOpt: a distributed optimization software environment	5
2.2 PDSDP: Parallel Dual Semidefinite Programming	6
2.3 SDPARA and SDPARA-C: SemiDefinite Programming Algorithm paRAllel version .	6
2.4 APPSPACK: Asynchronous Parallel Pattern Search	7
2.5 OOPS: Object oriented parallel solver	7
2.6 TAO: Toolkit for Advanced Optimization	8
2.7 Parallel Computing Toolbox for MATLAB	9
2.8 AIMMS: Advanced Integrated Multidimensional Modeling Software	9
2.9 Parallel CPLEX [7]	10
3 Comparison overview of the software packages	12
Bibliography	15

Project co-ordinator

Name: Bart De Schutter
Address: Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 Delft, The Netherlands
Phone Number: +31-15-2785113
Fax Number: +31-15-2786679
E-mail: b.deschutter@dcsc.tudelft.nl
Project web site: <http://www.ict-hd-mpc.eu>

Executive Summary

This report presents an overview of existing state-of-the-art distributed optimization algorithms. It discusses the advantages, strong points, and weaker points for each package. Moreover, for each software package we indicate the problem classes for which it is suited as well as the available interface, if any, with other solvers and programming environments that are frequently often used in the field of hierarchical and distributed model predictive control.

For an overview of toolboxes involving these software packages as well as for (tutorial) manuals and worked examples we refer to the references for each software package at the end of this report.

Chapter 1

Introduction

This report presents an overview of existing state-of-the-art distributed optimization algorithms. It discusses the advantages, strong points, and weaker points for each package. Moreover, for each software package we indicate the problem classes for which it is suited as well as the available interface, if any, with other solvers and programming environments that are frequently often used in the field of hierarchical and distributed model predictive control.

For an overview of toolboxes involving these software packages as well as for (tutorial) manuals and worked examples we refer to the references for each software package at the end of this report.

Chapter 2

Distributed and parallel optimization software packages

2.1 DistOpt: a distributed optimization software environment

DistOpt (a distributed optimization software environment) [17] is a software environment useful for modeling and evaluating the solutions of large-scale optimization problems in distributed environments. DistOpt approaches an optimization problem from a decomposition/coordination viewpoint. It exploits a particular decomposition/coordination methodology called Auxiliary Problem Principle [16] (APP), that allows to split a large problem into several subproblems (decomposition). The subproblems are solved iteratively, each one can be solved using any optimization technique, and the goal is to drive the solutions of the subproblems to the one of the original problem (coordination).

DistOpt is implemented in Ptolemy classic. Ptolemy [15] is an extensible object-oriented framework that can be used to model complex heterogeneous systems.

Strong and weak points

The advantage of using the APP methods shows itself when dealing with large problems where the speed of convergence and the computation time can be extremely scattered, and often very large.

Another advantage offered by DistOpt is the versatility in the phase of subproblem solution, where the user is free to use many optimizations techniques from the solvers' palette.

The disadvantage of DistOpt is that it only handles smooth, convex problems, due to the convergence theory of the underlying APP methods.

Problem classes

DistOpt can be used for continuous, convex nonlinear programs (NLP).

Interfaces to solvers and mathematical modeling environments

DistOpt has interfaces to the following solvers: CFSQP [28], QLD [33], LSGRG2 [27], SGRA [29], WNNLP [32].

DistOpt also has an interface to the environment MATLAB.

2.2 PDSDP: Parallel Dual Semidefinite Programming

The DSDP (Dual Semidefinite Programming) software [4, 12, 13] is a free open source implementation of an interior-point method for semidefinite programming (SDP). It provides primal and dual solutions, exploits low-rank structure and sparsity in the data, and has relatively low memory requirements for an interior-point method. It allows feasible and infeasible starting points and provides approximate certificates of infeasibility when no feasible solution exists. The dual-scaling algorithm implemented in this package has a convergence proof and worst-case polynomial complexity under mild assumptions on the data. The software can be used as a set of subroutines, through Matlab, or by reading and writing to data files. Furthermore, the solver offers scalable parallel performance for large problems and a well-documented interface. Some of the most popular applications of semidefinite programming and linear matrix inequalities are model control, truss topology design, and semidefinite relaxations of combinatorial and global optimization problems. Included in DSPD package is PDSDP, a parallel implementation of DSDP.

Strong and weak points

One advantage of PDSDP is that it can handle general semidefinite programs efficiently and with high robustness. Moreover, due to ease of use and the interface to MATLAB, PDSDP has been used in many applications, and has become a benchmarking technique for comparison with other parallel solvers for semidefinite programs, e.g., SDPARA.

A weak point is that DSDP only handles mixed linear and semidefinite programs.

Problem classes

PDSDP can be used for Semidefinite Programs for Combinatorial Optimization.

Interfaces to solvers and mathematical modeling environments

PDSDP has interface to the environment: MATLAB.

2.3 SDPARA and SDPARA-C: SemiDefinite Programming Algorithm paRAllel version

SDPA [19] (SemiDefinite Programming Algorithm) is one of the most efficient and stable software packages for solving SDPs based on the primal-dual interior-point method. It fully exploits the sparsity of given problems. The SDPARA [34] (SemiDefinite Programming Algorithm paRAllel version) is a parallel version of SDPA suitable for running on multiple processors with distributed memory, using MPI¹ [25] and ScaLAPACK [14]. The SDPARA-C [31] (SDPARA with the positive definite matrix Completion) is appropriate for sparse large-scale SDPs arising from combinatorial optimization.

Strong and weak points

Strong point: In control problems SDPARA achieves both faster total time and higher scalability than the PDSDP. This is mainly due to the fact that SDPARA computes the elements of the Schur

¹MPI: Message Passing Interface

complement matrix in parallel without any communication between the processors. This enables the SDPARA to obtain high scalability for control problems where most of the computational time is spent for computation of the Schur complement matrix.

Weak point: SDPARA is a standalone package, it does not have interface with MATLAB or other mathematical modeling languages. And like PDSDP, SDPARA only handles semidefinite programs (and SDPARA-C handles combinatorial semidefinite programs).

Problem classes

SDPARA: Semidefinite Programs.

SDPARA-C: Semidefinite Programs for Combinatorial Optimization.

Interfaces to solvers and mathematical modeling environments

Although in the SDPA family there is SDPA-M that has an interface to MATLAB, the parallel versions SDPARA and SDPARA-C do not have such interface.

2.4 APPSPACK: Asynchronous Parallel Pattern Search

APPSPACK (Asynchronous Parallel Pattern Search) [24, 26] is serial or parallel, derivative-free optimization software for solving nonlinear unconstrained, bound-constrained, and linearly-constrained optimization problems, with possibly noisy and expensive objective functions. APPSPACK implements asynchronous parallel generating set search, which handles bound and linear constraints by choosing search directions that conform to the nearby boundary. APPSPACK is written in C++ and uses MPI for parallelism. There is (optional) asynchronism that enables better load balancing.

Strong and weak points

Strong point: with APPSPACK, *only* function values are required for the optimization, so it can be applied to a wide variety of problems. The user needs simply to provide a program (written in any language such as Fortran, C, C++, Perl, etc.) that can evaluate the objective function at a given point.

Weak points: APPSPACK is a derivative-free method, so the optimization does not use the advantage of derivatives for smooth problems. Another point is that APPSPACK does not have interfaces with mathematical modeling environments.

Problem classes

APPSPACK can be applied to nonlinear unconstrained, bound-constrained, and linearly-constrained optimizations.

Interfaces to solvers and mathematical modeling environments

APPSPACK does not have interfaces with mathematical modeling environments.

2.5 OOPS: Object oriented parallel solver

OOPS (Object oriented parallel solver) [20–23] is a parallel interior point code that exploits any special structure in the Hessian and Jacobian matrices. The solver is an implementation of the primal-dual interior point method with multiple centrality correctors. The solver is implemented using object-oriented programming techniques. It solves linear (LP), quadratic (QP), and nonlinear (NLP) problems. OOPS is used together with a modeling language to retrieve the underlying structure of the optimization problem, called Structure Exploiting Tool (SET) [18]. The authors also developed a simple interface called Structure Passing Interface (SPI) to help passing the structure information from the optimization model to OOPS.

Strong and weak points

Strong point of OOPS: the solver exploits structure in linear algebra operations of interior-point methods, which saves memory and thus allows using the solver for extremely large problems with structure (more than one billion variables [21]).

Weak point: although SET has a modular approach that allows interfacing to other structure exploiting solvers, so far the only solver attached to it is OOPS.

Problem classes

OOPS can solve large-scale linear (LP), quadratic (QP), and nonlinear (NLP) problems.

Interfaces to solvers and mathematical modeling environments

OOPS has interfaces to the mathematical modeling environments AMPL and MPS.

2.6 TAO: Toolkit for Advanced Optimization

The TAO (Toolkit for Advanced Optimization) [11] project focuses on the development of software for large-scale optimization problems. TAO uses an object-oriented design to create a flexible toolkit with strong emphasis on the reuse of external tools where appropriate. TAO's design enables bi-directional connection to lower-level linear algebra support (for example, parallel sparse matrix data structures) as well as higher-level application frameworks.

The Toolkit for Advanced Optimization (TAO) is aimed at the solution of large-scale optimization problems on high-performance architectures. The main goals of TAO are portability, performance, scalable parallelism, and an interface independent of the architecture. TAO is suitable for both single-processor and massively-parallel architectures. The current version of TAO has algorithms for unconstrained and bound-constrained optimization.

Strong and weak points

Strong point: TAO used object-oriented design with emphasis on the reuse of external tools.

Weak point: Currently TAO only has algorithms for unconstrained and bound-constrained optimization, still does not handle nonlinearly constrained problems.

Problem classes

TAO is able to handle unconstrained optimizations, bound constrained optimizations, and mixed complementarity problems.

Interfaces to solvers and mathematical modeling environments

TAO uses the solvers in the package PETSc 2.3.3 [10], a suite of parallel linear and nonlinear equation solvers.

There also is an interface to CCA (Common Component Architecture) [3] and GA (Global Arrays) [6].

2.7 Parallel Computing Toolbox for MATLAB

The Parallel Computing Toolbox [30] is a package of MATLAB, it allows offloading work from one MATLAB session (the client) to other MATLAB sessions, called workers. Multiple workers can be used to take advantage of parallel processing. The Parallel Computing Toolbox simplifies the development of parallel applications in two significant ways. First, the toolbox supports distributed arrays, which treat related data distributed across processors as a single array rather than as independent arrays on different processors. Through distributed arrays, researchers can perform operations directly on arrays without having to manage the tedious details of message passing. The toolbox provides more than 150 MATLAB functions for distributed arrays, including linear algebra routines based on ScaLAPACK [14]. Second, the toolbox includes a new interactive parallel mode (pmode) of execution so that researchers can work interactively with a parallel job running simultaneously on several processors or cores. This mode facilitates iterative exploration, design development, and problem solving.

Strong and weak points

The advantage of the Parallel Computing Toolbox is that it acts as a set of tools that help distributing the computations in MATLAB. Therefore, many MATLAB scripts can be parallelized based on the functions provided by Parallel Computing Toolbox.

The weak points of Parallel Computing Toolbox are: firstly, the cost of licenses of the Parallel Computing Toolbox for all computers that are involved in a parallel sessions; secondly, the built-in optimization solvers of MATLAB are not very robust and efficient, while the highly robust and efficient solvers from third-parties do not make use of this toolbox.

Problem classes

With the Parallel Computing Toolbox, any optimization problem solvable by the Optimization Toolbox of MATLAB can be parallelized.

Interfaces to solvers and mathematical modeling environments

The Parallel Computing Toolbox is a MATLAB toolbox.

2.8 AIMMS: Advanced Integrated Multidimensional Modeling Software

AIMMS (Advanced Integrated Multidimensional Modeling Software) [1] is an advanced development environment for building optimization-based operations research applications and advanced planning systems. AIMMS is a mathematical modeling language, with a full graphical user interface both for developers and end-users. AIMMS includes solvers for all major mathematical programming types (such as linear programming, mixed integer programming, and nonlinear programming) and can be further extended to incorporate any of the advanced commercial solvers.

AIMMS offers the Generated Mathematical Program (GMP) library, a set of procedures that allows users to gain fine-grained control over the generation, manipulation, and solution of a mathematical program instance, and to manage a collection of solutions and solver sessions associated with such mathematical program instances. The AIMMS GMP library also supports the activation of parallel solver sessions, allowing an application to solve multiple mathematical programs in parallel on one computer.

Strong and weak points

Strong points: AIMMS can use many well-known solvers for all major mathematical programming types (like CPLEX, KNITRO, MINOS, SNOPT, etc.). Another advantage is that AIMMS has graphical interface for both developers and users, which allows easy adoption of the software.

Weak point: Users need to pay for the licenses of both AIMMS and an appropriate solver for each type of problems.

Problem classes

AIMMS can handle a wide range of problems, by using appropriate solvers. The list of problem classes includes: linear programming (LP), mixed integer linear programming (MILP), quadratic programming (QP), mixed integer quadratic programming (MIQP), quadratic constraint programming (QCP), mixed integer quadratic constraint programming (MIQCP), nonlinear programming (NLP), mixed integer nonlinear programming (MINLP), mixed complementarity programming (MCP), and mathematical programming with complementarity constraints (MPCC).

Interfaces to solvers and mathematical modeling environments

AIMMS has an interface to the following solvers: CONOPT, CPLEX, SNOPT, GUROBI, MINOS, KNITRO, XA, AOA, MOSEK, PATH, XPRESS, BARRON, and LGO. More information is available on the AIMMS web site [8].

AIMMS does not have interface with other mathematical modeling environments.

2.9 Parallel CPLEX [7]

ILOG CPLEX (often informally referred to simply as CPLEX) provides flexible, high-performance optimizers used when solving linear programming, quadratic programming, quadratically constrained programming, and mixed integer programming problems.

CPLEX provides a parallel CPLEX option allowing users to take advantage of the availability of additional CPUs to speed up performance while a specific model is solved. CPLEX 11 extends the

functionality of the parallel MIP optimizer to include two modes of operation. In the deterministic mode, a newly implemented search algorithm exploits parallelism in solving nodes of the branch-and-cut tree, but produces a repeatable, invariant solution path. In the opportunistic mode, the search algorithm takes full advantage of parallelism; it performs less synchronization between threads and allows random tie breaking, which may result in different solution paths but potentially faster performance.

Strong and weak points

Strong points: CPLEX is an industry-standard solver that has proved its robustness and efficiency for years. CPLEX is widely supported by mathematical modeling languages.

Weak point: Users need to pay for the license of CPLEX.

Problem classes

CPLEX (and parallel CPLEX) can solve Linear Programming (LP), Mixed Integer Programming (MIP) and Quadratic Programming (QP/QCP/MIQP/MIQCP).

Interfaces to solvers and mathematical modeling environments

CPLEX is supported by the interfaces of AIMMS, AMPL [2], GAMS [5], and TOMLAB [9], and some other packages.

Chapter 3

Comparison overview of the software packages

In Table 3.1, we provide a comparison between the software packages described in this paper. To list the information compactly in the table, we use a number of abbreviations for the types of optimization problems, whose meaning is provided below:

- LP: Linear Program.
- QP: Quadratic Program.
- SDP: Semidefinite Program.
- QCP: Quadratic Constraint Program.
- NLP: Nonlinear Program.
- MIP: Mixed Integer Program.
- MIQP: Mixed Integer Quadratic Program.
- MIQCP: Mixed Integer Quadratic Constraint Program.
- NLP: Nonlinear Program.
- MINLP: Mixed Integer Nonlinear Program.
- MPCC: Mathematical Program with Complementarity Constraints.
- MCP: Mixed Complementarity Program.
- GO: Global Optimization.

Many of the software packages in this review were originally solvers on single-core platforms that aimed at solving large-scale optimization problems. Thanks to the development of MPI, which is now a standard for message passing between multiple processors in parallel computing, the solvers could be implemented in parallel. The main goals of MPI are to share the computations and memory between processors that are suitable for parallel computing. However, from the viewpoint of distributed model predictive control (MPC), it is difficult to use parallel solvers in that fashion. This is

because the issues of distributed MPC not only arise from the computational aspects but also from the communications (limitation of bandwidth, losses, and delays), and control theory aspects (feasibility, stability).

Table 3.1: Comparison table of the software packages for distributed optimization

Package	Free use / commercial	Problem classes	Single / multiple processors	Language	Available interface	Year of latest release
DistOpt	Free	Continuous, convex NLP	Single	Ptolemy Classic	MATLAB	2002
PDSDP	Free	LP, QP, SDP, SDP for combinatorial optimization	Multiple	C	MATLAB	2005
SDPARA, SDPARA-C	Free	LP, QP, SDP, SDP for combinatorial optimization	Multiple	C++	MATLAB	2009
APPSPACK	Free	Unconstrained, bound constrained, linearly constrained NLP	Multiple	C++		2007
OOPS	n/a (unavailable for public)	LP, QP, NLP	Multiple	C	AMPL	2006
TAO	Free	Unconstrained, bound constrained NLP, MCP	Multiple	C	CCA, GA	2007
Parallel Computing Toolbox	Commercial	Any problem that can be solved by the Matlab Optimization Toolbox	Multiple	C, Java	MATLAB	2009
AIMMS	Commercial	LP, QP, QCP, NLP, MIP, MIQP, MIQCP, MINLP, GO, MPCC, MCP	Multiple	C++		2009
Parallel CPLEX	Commercial	LP, QP, QCP, MIP, MIQP, MIQCP	Multiple	C, C++, C#, Java, Python	AMPL, AIMMS, GAMS, TOMLAB	2009

Bibliography

- [1] AIMMS Web page, 2009. <http://www.aimms.com>.
- [2] AMPL Web page, 2009. <http://www.ampl.com>.
- [3] Common Component Architecture forum, 2009. <http://www.cca-forum.org/>.
- [4] DSDP: Software for Semidefinite Programming, 2009.
<http://www.mcs.anl.gov/hs/software/DSDP/>.
- [5] GAMS Web page, 2009. <http://www.gams.com>.
- [6] Global Arrays, 2009. <http://www.emsl.pnl.gov/docs/global/ga.html>.
- [7] ILOG parallel CPLEX Web page, 2009.
<http://www.ilog.com/products/cplex/product/parallel.cfm>.
- [8] List of AIMMS supported solvers, 2009.
<http://www.aimms.com/licensing/solvers/supported-math-program-types>.
- [9] TOMLAB Web page, 2009. <http://tomopt.com/tomlab/>.
- [10] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2009. <http://www.mcs.anl.gov/petsc>.
- [11] S. Benson, L. Curfman McInnes, J. Moré, T. Munson, and J. Sarich. TAO user manual (revision 1.9). Technical Report ANL/MCS-TM-242, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, 2007. <http://www.mcs.anl.gov/tao>.
- [12] S. J. Benson and Y. Ye. DSDP5 user guide — software for semidefinite programming. Technical Report ANL/MCS-TM-277, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, September 2005. <http://www.mcs.anl.gov/~benson/dsdp>.
- [13] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.
- [14] L. S. Blackford, J. Choi, A. Cleary, E. D’Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users’ Guide*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1997.
- [15] J. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt. Ptolemy: A platform for heterogeneous simulation and prototypes. In *Proceedings of European Simulation Conference*, Copenhagen, Denmark, June 1991.

- [16] G. Cohen. Auxiliary problem principle and decomposition of optimization problems. *Journal of Optimization Theory and Applications*, 2(3):277–305, November 1980.
- [17] J. Contreras, A. Losi, M. Russo, and F. F. Wu. DistOpt: A software framework for modeling and evaluating optimization problem solutions in distributed environments. *Journal of Parallel and Distributed Computing*, 60(6):741–763, June 2000.
- [18] E. Fragniere, J. Gondzio, R. Sarkissian, and J.-P. Vial. Structure exploiting tool in algebraic modeling languages. *Management Science*, 46:1145–1158, 2000.
- [19] K. Fujisawa, M. Kojima, K. Nakata, and M. Yamashita. SDPA (semidefinite programming algorithm) user’s manual — version 6.2.0. Technical report, Dept. Math. and Comp. Sciences, Tokyo Institute of Technology, Tokyo, Japan, December 1995. Revised September 2004.
- [20] J. Gondzio and A. Grothey. Reoptimization with the primal-dual interior point method. *SIAM Journal on Optimization*, 13(3):842–864, 2003.
- [21] J. Gondzio and A. Grothey. Direct solution of linear systems of size 10^9 arising in optimization with interior point methods. In R. Wyrzykowski, J. Dongarra, N. Meyer, and J. Wasniewski, editors, *Parallel Processing and Applied Mathematics PPAM 2005*, number 3911 in Lecture Notes in Computer Science, pages 513–525. Springer-Verlag, Berlin, 2006.
- [22] J. Gondzio and A. Grothey. Parallel interior point solver for structured quadratic programs: Application to financial planning problems. *Annals of Operations Research*, 152(1):319–339, 2007.
- [23] J. Gondzio and A. Grothey. Exploiting structure in parallel implementation of interior point methods for optimization. *Computational Management Science*, 6:135–160, 2009.
- [24] G. A. Gray and T. G. Kolda. Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization. *ACM Transactions on Mathematical Software*, 32(3):485–507, September 2006.
- [25] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable parallel programming with the message-passing interface*. MIT Press, Cambridge, Massachusetts, 1999.
- [26] T. G. Kolda. Revisiting asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Optimization*, 16(2):563–586, December 2005.
- [27] L. S. Lasdon and A. D. Waren. Generalized reduced gradient software for linearly and nonlinearly constrained problems. In H. Greenberg, editor, *Design and Implementation of Optimization Software*. Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, 1979.
- [28] C. T. Lawrence, J. L. Zhou, and A. L. Tits. *Users Guide for CFSQP Version 2.5d*. University of Maryland, Institute for Systems Research, College Park, Maryland, January 1998. <http://www.aemdesign.com>.
- [29] A. Losi. *Users Guide for SGRA - Version 0.2*. Universit degli Studi di Cassino, Italy, 1999. <ftp://sistelet.ing.unicas.it/Sgra>.
- [30] The Mathworks. MATLAB Parallel Computing Toolbox Web page, 2009. <http://www.mathworks.com/products/parallel-computing/>.

- [31] K. Nakata, M. Yamashita, K. Fujisawa, and M. Kojima. A parallel primal-dual interior-point method for semidefinite programs using positive definite matrix completion. *Parallel Computing*, 32:24–43, 2006.
- [32] W. Naylor. *WNLIB Library - Release 6*. Spike Technologies, Milpitas, California, May 1996. <http://www.willnaylor.com>.
- [33] K. Schittkowski. *QLD: A FORTRAN Code for Quadratic Programming, Users Guide*. Mathematisches Institut, Universität Bayreuth, Bayreuth, Germany, 1986.
- [34] M. Yamashita, K. Fujisawa, and M. Kojima. Semidefinite programming algorithm parallel version. *Parallel Computing*, 29:1053–1067, 2003.