1 Optimization problems

We study optimization problems involving linear and nonlinear constraints:

\[
\text{minimize } \phi(x) \quad \text{subject to } \ell \leq \begin{pmatrix} x \\ Ax \\ c(x) \end{pmatrix} \leq u,
\]

where \( \phi(x) \) is a linear or nonlinear objective function, \( A \) is a sparse matrix, \( c(x) \) is a vector of nonlinear constraint functions \( c_i(x) \), and \( \ell \) and \( u \) are vectors of lower and upper bounds. We assume the functions \( \phi(x) \) and \( c_i(x) \) are smooth: they are continuous and have continuous first derivatives (gradients). Sometimes gradients are not available (or too expensive) and we use finite difference approximations. Sometimes we need second derivatives.

We study algorithms that find a local optimum for problem NP. Some examples follow. If there are many local optima, the starting point is important.

\begin{itemize}
  \item **LP** Linear Programming
    \[ \min c^T x \text{ subject to } \ell \leq \begin{pmatrix} x \\ Ax \end{pmatrix} \leq u \]
    MINOS, SNOPT, SQOPT, LSSOL, QPOPT, NPSOL (dense)
    CPLEX, Gurobi, LOQO, HOPDM, MOSEK, XPRESS
    CLP, lp_solve, SoPlex (open source solvers [7, 34, 54])
  \item **QP** Quadratic Programming
    \[ \min c^T x + \frac{1}{2} x^T H x \text{ subject to } \ell \leq \begin{pmatrix} x \\ Ax \end{pmatrix} \leq u \]
    MINOS, SQOPT, SNOPT, QPBLUR
    LSSOL (\( H = B^T B \), least squares), QPOPT (\( H \) indefinite)
    CLP, CPLEX, Gurobi, LANCELOT, LOQO, MOSEK
  \item **BC** Bound Constraints
    \[ \min \phi(x) \text{ subject to } \ell \leq x \leq u \]
    MINOS, SNOPT, LANCELOT, L-BFGS-B
  \item **LC** Linear Constraints
    \[ \min \phi(x) \text{ subject to } \ell \leq \begin{pmatrix} x \\ Ax \end{pmatrix} \leq u \]
    MINOS, SNOPT, NPSOL
  \item **NC** Nonlinear Constraints
    \[ \min \phi(x) \text{ subject to } \ell \leq \begin{pmatrix} x \\ Ax \\ c(x) \end{pmatrix} \leq u \]
    MINOS, SNOPT, NPSOL
    CONOPT, LANCELOT
    Filter, KNITRO, LOQO (second derivatives)
    IPOPT (open source solver [30])
\end{itemize}

Algorithms for finding local optima are used to construct algorithms for more complex optimization problems: stochastic, nonsmooth, global, mixed integer. Excellent examples are SCIP [50] for MILP and BARON [3] for MINLP.

2 AMPL, GAMS, NEOS

A fuller picture emerges from the list of problem types and solvers handled by the AMPL [2] and GAMS [20] modeling systems and the NEOS server [41]. NEOS is a free service
developed originally at Argonne National Laboratory and now hosted by the University of Wisconsin – Madison. It allows us to submit optimization problems in various formats (AMPL, GAMS, CPLEX, MPS, C, Fortran, …) to be solved remotely on geographically distributed machines.

**NEOS usage** (Jan 1–Dec 31 of each year):

<table>
<thead>
<tr>
<th>Year</th>
<th>Total jobs</th>
<th>Top solvers</th>
<th>Top inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>80,000</td>
<td>XpressMP, MINLP, MINOS, SNOPt, SBB</td>
<td>AMPL, GAMS, Fortran</td>
</tr>
<tr>
<td>2003</td>
<td>136,000</td>
<td>XpressMP, SNOPt, FortMP, MINOS, MINLP</td>
<td>AMPL, GAMS, Fortran</td>
</tr>
<tr>
<td>2004</td>
<td>148,000</td>
<td>MINLP, FortMP, XpressMP, PENNON, MINOS</td>
<td>AMPL, GAMS, Mosel</td>
</tr>
<tr>
<td>2005</td>
<td>174,000</td>
<td>Filter, MINLP, XpressMP, MINOS, KNITRO</td>
<td>AMPL, GAMS, Fortran</td>
</tr>
<tr>
<td>2006</td>
<td>229,000</td>
<td>Filter, MINOS, SNOPt, XpressMP, MOSEK</td>
<td>AMPL, GAMS, MPS</td>
</tr>
<tr>
<td>2007</td>
<td>551,000</td>
<td>SNOPt, MINLP, KNITRO, LOQO, MOSEK</td>
<td>AMPL, GAMS, MPS</td>
</tr>
<tr>
<td>2008</td>
<td>322,000</td>
<td>MINOS, Bonmin, KNITRO, SNOPt, IPOpt</td>
<td>AMPL, GAMS, CPLEX</td>
</tr>
<tr>
<td>2009</td>
<td>235,000</td>
<td>KNITRO, BPMPD, MINTO, MINOS, SNOPt</td>
<td>AMPL, GAMS, CPLEX</td>
</tr>
<tr>
<td>2010</td>
<td>236,000</td>
<td>KNITRO, Concorde, SNOPt, SBB, BARON</td>
<td>AMPL, GAMS, TSP</td>
</tr>
<tr>
<td>2011</td>
<td>75,000</td>
<td>SBB, Gurobi, filter, MINLP, XpressMP, MINTO, KNITRO, PATH, MINOS, MOSEK, LOQO, sdp3</td>
<td>GAMS, AMPL, Matlab_Binary, Fortran, Sparse_SDPA, MPS</td>
</tr>
<tr>
<td>2012</td>
<td>353,000</td>
<td>Gurobi, MINOS, CONOPT, SBB, KNITRO, XpressMP, MINTO, MINOS, IPOpt</td>
<td>AMPL, GAMS, Sparse_SDPA, MPS, Fortran, MOSEL, TSP, CPLEX, C</td>
</tr>
<tr>
<td>2013</td>
<td>1,865,000</td>
<td>MINOS, MINLP, KNITRO, Gurobi, IPOpt, SNOPt, csdp, DICOPT, Cbc, XpressMP, MINTO, BARON</td>
<td>AMPL, GAMS, Sparse_SDPA, MPS, TSP, CPLEX, C</td>
</tr>
<tr>
<td>2014</td>
<td>1,139,000</td>
<td>MINLP, Gurobi, filterMPEC, KNITRO, BARON, MINOS, Cbc, sdp, concorde, LOQO, MOSEK</td>
<td>AMPL, GAMS, TSP, Sparse_SDPA, MPS, C, MOSEL, FORTRAN, CPLEX</td>
</tr>
<tr>
<td>2015</td>
<td>645,000</td>
<td>MINLP, Gurobi, cbc, cplex, KNITRO, concorde, BARON, Bonmin, XpressMP, Couenne, MOSEK, MINOS</td>
<td>AMPL, GAMS, Sparse_SDPA, MPS, C, MOSEL, FORTRAN, MATLAB_BINARY</td>
</tr>
<tr>
<td>2016</td>
<td>1,039,974</td>
<td>CPLEX, Gurobi, knitro, cbc, moose, FICO-Xpress, baron, Bonmin, minos, couenne, ipopt, scip, concorde, conopt</td>
<td>AMPL, GAMS, LP, MPS, TSP, MOSEL, Sparse_SDPA, MATLAB_BINARY, C</td>
</tr>
</tbody>
</table>

**NEOS problem categories:**

<table>
<thead>
<tr>
<th>Category</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCO</td>
<td>Bound Constrained Optimization</td>
</tr>
<tr>
<td></td>
<td>L-BFGS-B</td>
</tr>
<tr>
<td>COIP</td>
<td>Combinatorial Optimization and Integer Programming</td>
</tr>
<tr>
<td></td>
<td>BiqMac, concorde</td>
</tr>
<tr>
<td>CP</td>
<td>Complementarity Problems</td>
</tr>
<tr>
<td></td>
<td>Knitro, MILES, NLPEC, PATH</td>
</tr>
<tr>
<td>EMP</td>
<td>Extended Mathematical Programming</td>
</tr>
<tr>
<td></td>
<td>DE, JAMS</td>
</tr>
<tr>
<td>GO</td>
<td>Global Optimization</td>
</tr>
<tr>
<td></td>
<td>ASA, BARON, Couenne, icos, LINDOGlobal, PGAPack, PSwarm, scip</td>
</tr>
<tr>
<td>LNP</td>
<td>Linear Network Programming</td>
</tr>
<tr>
<td></td>
<td>RELAX4</td>
</tr>
<tr>
<td>LP</td>
<td>Linear Programming</td>
</tr>
<tr>
<td></td>
<td>BDMLP, bmond, Clp, CPLEX, FICO-Xpress, Gurobi, MOSEK, OQOQP, SoPlex80bit</td>
</tr>
<tr>
<td>MPEC</td>
<td>Mathematical Programming with Equilibrium Constraints</td>
</tr>
<tr>
<td></td>
<td>filterMPEC, Knitro, NLPEC</td>
</tr>
<tr>
<td>MILP</td>
<td>Mixed Integer Linear Programming</td>
</tr>
<tr>
<td></td>
<td>Cbc, CPLEX, feasump, FICO-Xpress, Gurobi, MINTO, MOSEK, proxy, qsopt_ex, scip, SYMPHONY</td>
</tr>
<tr>
<td>MINCO</td>
<td>Mixed Integer Nonlinearly Constrained Optimization</td>
</tr>
<tr>
<td></td>
<td>AlphaECP, BARON, Bonmin, Couenne, DICOPT, FilMINT, Knitro, LINDOGlobal, MINLP, SBB, scip</td>
</tr>
<tr>
<td>MIOCP</td>
<td>Mixed Integer Optimal Control Problems</td>
</tr>
<tr>
<td></td>
<td>MUSCOD-II</td>
</tr>
</tbody>
</table>
**NDO** Nondifferentiable Optimization

**condor**

**NCO** Nonlinearly Constrained Optimization

CONOPT, filter, Ipopt, Knitro, LANCELOT, LOQO, MINOS, MOSEK, PATHNLP, SNOPT

**SOCP** Second-Order Conic Programming

FICO-Xpress, MOSEK

**SIO** Semi-infinite Optimization

nsips

**SDP** Semidefinite Programming

csdp, penbmi, pensdp, scipsdp, SDPA, sdplr, sdpt3, sedumi

**SLP** Stochastic Linear Programming

bnbs, ddsip, sd

Most solvers use double-precision floating-point hardware.

For LP, SoPlex80bit mostly uses double-precision, but includes an iterative refinement procedure with rational arithmetic to obtain arbitrarily high precision \[23, 22\].

For LP and RLP models, GAMS will soon offer the QUADMINOS solver, in which hardware double-precision floating-point is replaced by software quad-precision.

For SDP, the SDPA solver on NEOS allows you to choose double-precision, quadruple-precision, octuple-precision, or variable-precision versions \[51\].

### 3 Interactive optimization systems

Several systems provide a **graphical user interface** (GUI) or **integrated development environment** (IDE) for mathematical optimization.

**MATLAB** [37] has an Optimization Toolbox with a selection of dense and sparse solvers (none of the above, except ktrlink uses KNITRO [31]). Matlab Version 7.2 (R2015a) has the following functions (see help optim):

- fminbnd, fmincon, fminsearch, fminunc, fseminf, ktrlink
- fgoalattain, fminimax
- lsqmin, lsqnonneg, lsqcurvfit, lsqnonlin
- fzero, fsolve
- intlinprog, linprog, quadprog

**TOMLAB** [57] provides a complete optimization environment for MATLAB users. There are many problem types and solvers (CGO, CONOPT, CPLEX, GENO, GP, Gurobi, KNITRO, LGO, LSSOL, MINLP, MINOS, NLPLQ, NLSSOL, NPSOL, OQNLP, PENBMI, PENSDP, PROPT, QP, SNOPT, SQOPT, Xpress), a unified input format, automatic differentiation of M-files with MAD, an interface to AMPL, a GUI for selecting parameters and plotting output, and TomSym: a modeling language with complete source transformation.

**Note:** TOMLAB is available on the Stanford Linux cluster. See [http://stanford.edu/group/SOL/download.html](http://stanford.edu/group/SOL/download.html)

**CVX** [11] is a MATLAB-based modeling system for convex optimization problems (and for geometric programs). It allows objectives and constraints to be specified using MATLAB syntax.

**AIMMS** [1] includes solvers for CP, GO, LP, MILP, MINLP, NLP, QCP, QP. Its modeling language has historical connections to GAMS.

**COMSOL Optimization Lab** [10] provides the LP and NLP capabilities of SNOPT to users of COMSOL Multiphysics [9]. Currently, most problem classes are handled by SNOPT. A Nelder-Mead “simplex algorithm” is included for unconstrained optimization without derivatives.

**Frontline Systems** [19] provides Excel spreadsheet and Visual Basic access to optimizers for many problem classes: LP, QP, SOCP, MILP, NLP, GO, NDO.

**GAMS IDE** [20] provides an IDE for GAMS Windows installations.

**IBM ILOG CPLEX Optimization Studio** [29] provides an IDE for LP, MILP, and constraint-programming applications, along with a high-level modeling language (OPL).
4 More optimization systems

We cite a few systems in order to include them in the references: COIN-OR [8], CPLEX [28], Galahad [24, 25], Gurobi [26], Lindo [33], MOSEK [39], Optizelle [42], PENOPT [47], SeDuMi [52], TFOCS [56].

5 Sparse linear systems

Underlying almost all of the optimization algorithms is the need to solve a sequence of linear systems $Ax = b$ (where “$x$” is likely to be a search direction). We will study some of the linear system software below. Most codes are implemented in Fortran 77. MA57 includes an F90 interface, and newer HSL packages [27] are in F90. UMFPACK is written in C.

When $A$ is a sparse matrix, MATLAB uses MA57 for $\text{ldl}(A)$ and UMFPACK for $A \backslash b$ and $\text{lu}(A)$ (both direct methods).

If $A$ is a sparse matrix (or a linear operator defined by a function handle), MATLAB has the following iterative methods for solving $Ax = b$ or $\min \|Ax - b\|_2$: bicg, bicgstab, bicgstabl, cgs, gmres, lsqr, minres, pcg, qmr, symmlq, tfqmr.

Some of the iterative solvers are available in F77, F90, and MATLAB from SOL [53]. PETSc [48] provides many direct and iterative solvers for truly large problems.

Direct methods factorize sparse $A$ into a product of triangular matrices that should be sparse and well defined even if $A$ is singular or ill-conditioned.

$LUSOL$ [21, 35, 36] Square or rectangular $Ax = b$, $A = LU$, plus updating

$MA48$ [16] Square or rectangular $Ax = b$, $A = LU$

$MA57$ [15] Symmetric $Ax = b$, $A = LDL^T$ or $LBL^T$ (MATLAB’s $[L,D,P] = \text{ldl}(A)$)

$MUMPS$ [40] Square $Ax = b$, $A = LU$, $LDL^T$ or $LBL^T$ (massively parallel)

$PARDISO$ [46] Square $Ax = b$ (shared memory)

$SuperLU$ [14, 32, 55] Square $Ax = b$ (uniprocessor or shared or distributed memory)


$UMFPACK$ [58, 12] Square $Ax = b$ (MATLAB’s $[L,U,P,Q] = \text{lu}(A)$)

Iterative methods regard $A$ as a black box (a linear operator) for computing matrix-vector products $Ax$ and sometimes $A^T y$ for given $x$ and $y$.

$CG$, $PCG$ [37, 48] Symmetric positive-definite $Ax = b$

$SYMMLQ$ [43, 53, 48] Symmetric nonsingular $Ax = b$ (may be indefinite)

$MINRES$, $MINRES-QLP$ [43, 53, 48, 4, 5, 18, 6] Symmetric $Ax = b$ (may be indefinite or singular)

$GMRES$ [49, 48] Unsymmetric $Ax = b$

$CGLS$, $LSQR$, $LSMR$, $LSRN$ [44, 45, 53, 48, 17, 38] $Ax = b$, $\min \|Ax - b\|_2$, $\min \left\| \begin{pmatrix} A \delta I \\ b \end{pmatrix} \right\|_2$

References


[34] lp_solve open source LP and MILP solver. http://groups.yahoo.com/group/lp_solve/.


