NAME

runcutest − CUTEst interface to solvers.

SYNOPSIS

runcutest --package *pack* [−−architecture *arch*] [−−single] [−−help] [−−keep] [−−rebuild] [−−output *0|1*] [−−limit *secs*] [−−cfortran] [−−debug] [−−uncons] [−L*path/to/lib*] [−−blas *keyword*] [−−lapack *keyword*] [−−decode *problem[.SIF]*]

DESCRIPTION

runcutest is the CUTEst interface to solvers. It replaces its predecessor CUTEr's combination of **runpackage**, *pkg* and **sd***pkg*. The command accepts options in short or long form. Any option that is not directly recognized is passed unchanged to the SIF decoder, sifdecoder(1).

runcutest reads suitable architecture-dependent environment variables and then compiles and links all the relevant source files and libraries to form an executable of the package *package* running on problem *problem*.

The user has the opportunity to run commands before and after the run if need be. **runcutest** executes the script *package*_pre, if it exists, before the run. Similarly, it executes the script *package*_post, if it exists, after completion of the run.

runcutest Options

You can start runcutest with the following options. An option can be used either in short or long form.

−p, −−package *pack*
: Specifies the package or solver, *pack* to use. See the section **Currently Supported Packages** below. This is the only mandatory option.

−A, −−architecture *arch*
: Run the decoder using the architecture *arch*; the architeture is a string of the form machine.system.compiler as specified in the directory $CUTEST/versions. If no −A option is given, a valid architecture given by the environment variable $MYARCH will be used, but if $MYARCH is invalid or empty the decoder will terminate.

−sp, −−single
: Run *package* in single-precision mode if available. Double precision is the default.

−h, −−help
: Print a short help message with the available command-line options.

−k, −−keep
: Keep the generated executable after use. May be useful when solving a particular problem with the same solver with different parameters. Deleting the executable after use is the default.

−r, −−rebuild
: Force recompilation of the test problem. Default is to reuse object files.

−o, −−output *0|1*
: Regulates the output level. Verbose mode is **-o** 1, silent mode is **-o** 0. Silent mode is the default.

−l, −−limit *secs*
: Sets a limit of *secs* second on the *package* runtime. Unlimited cputime is the default.

−c, −−cfortran
: Causes specialized compiler options to be used to specify that the main subroutine of *package* is written in C. This is necessary with some compilers, such as the Intel Fortran Compiler.

−L*path/to/lib*
: This option is passed directly to the linker and causes the path *path/to/lib* to be searched for libraries. Useful to specify custom BLAS and LAPACK libraries.

−b, −−blas *keyword*

Overrides usage of the default *linpack* library packaged with CUTEst. Instead, use the BLAS library specified by *keyword*. The keyword *keyword* has one of two forms. The first, *-lmyblas* causes the linker to search for BLAS subprograms in the libmyblas.a library. The second, *none*, causes the linker to skip inclusion of any external BLAS. Use the first option if an optimized BLAS library is available on the host system, e.g., the ATLAS BLAS. The second option is useful for packages which already include the necessary BLAS subprograms. Use of *none* may be useful if *package* already includes the BLAS subroutines on which it relies.

−K, −−lapack *keyword*

Overrides usage of the default *linpack* library packaged with CUTEst. Instead, use the LAPACK library specified by *keyword*. The keyword *keyword* has one of two forms. The first, *-lmylapack* causes the linker to search for LAPACK subroutines in the libmylapack.a library. The second, *none*, causes the linker to skip inclusion of any external LAPACK. Use the first option if an optimized LAPACK library is available on the host system. The second option is useful for packages which already include the necessary LAPACK subprograms. Use of *none* may be useful if *package* already includes the LAPACK subroutines on which it relies.

−D, −−decode *problem[.SIF]*

Applies the SIF decoder to the problem *problem.SIF* to produce the OUTSDIF.d file and the problem-dependant Fortan subroutines. If this flag is not specified, **runcutest** assumes that the problem has been decoded prior to the call.

−u, −−uncons

When *package* is **mx**, the Matlab interface, this option specifies that the problem is unconstrained. This causes the appropriate MEX interface to be compiled and linked. The default is to link with the constrained tools.

additional command-line options

Any command-line option not documented in this manual page and/or in the help message of **runcutest** is passed unchanged to the SIF decoder. See the **sifdecode** manual page for more information.

CURRENTLY SUPPORTED PACKAGES

There are currently interfaces to the following packages:

algencan

See E. G. Birgin, R. Castillo and J. M. Martinez, *Numerical comparison of Augmented Lagrangian algorithms for nonconvex problems*, Computational Optimization and Applications 31, 31-56 (2005).

http://www.ime.usp.br/˜egbirgin/tango/codes.php

bobyqa

See M.J.D. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives*, Technical report NA2009/06 Department of Applied Mathematics and Theoretical Physics, Cambridge England, (2009).

cg_descent

See W. W. Hager and H. Zhang, *Algorithm 851: CG_DESCENT, A conjugate gradient method with guaranteed descent*, ACM Transactions on Mathematical Software, 32, 113-137 (2006).

http://www.math.ufl.edu/˜hager/papers/CG/

cgplus

The CG+ package is a nonlinear conjugate-gradient algorithm designed for unconstrained minimization by G. Liu, Jorge Nocedal and Richard Waltz (Northwestern U.).

http://users.eecs.northwestern.edu/˜nocedal/CG+.html

cobyla

See M.J.D. Powell, *A direct search optimization method that models the objective and constraint functions by linear interpolation*, In Advances in optimization and numerical analysis, Proceedings of the Sixth workshop on Optimization and Numerical Analysis, Oaxaca, Mexico, volume 275 of Mathematics and its Applications, pp 51--67. Kluwer Academic Publishers (1994).

derchk

This package checks the dervatives supplied in the problem SIF file, and is due to Dominique Orban from Ecole Polytechnique de Montreal.

dfo

See A. R. Conn, K. Scheinberg and Ph.L. Toint, *On the convergence of derivative-free methods for unconstrained optimization*, Approximation Theory and Optimization: Tributes to M. J. D. Powell , Eds. A. Iserles and M. Buhmann, 83-108, Cambridge University Press (1997).

https://projects.coin-or.org/Dfo

direstsearch

The Direct Search suite provides a variety of patern search methods for derivative-free optimization and was written by Liz Dolan, Adam Gurson, Anne Shepherd, Chris Siefert, Virginia Torczon and Amy Yates.

http://www.cs.wm.edu/˜va/software/DirectSearch/direct_code/

filtersd

See R. Fletcher *A sequential linear constraint programming algorithm for NLP*, SIAM Journal on Optimization, 22(3), pp. 772-79 (2012).

http://www.coin-or.org/projects/filterSD.xml

filtersqp

FilterSQP is a filter-based SQP method for large-scale nonlinear programming by Roger Fletcher and Sven Leyffer from the University of Dundee.

*gen77***,** *gen90***,** *genc*

These package simply illustrates how CUTEst tools may be called in fortran 77, fortran 90 and C; the result is of no consequence.

hrb

This package writes the matrix data for the given problem in Harwell or Rutherford-Boeing sparse matrix forrmat, and was provded by Nick Gould from the Rutherford Appleton Laboratory.

ipopt

See A. WÃ¤chter and L. T. Biegler, *On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming*, Mathematical Programming 106(1) 25-57 (2006).

https://projects.coin-or.org/Ipopt

knitro

See R. H. Byrd, J. Nocedal, and R. A. Waltz, *KNITRO: An Integrated Package for Nonlinear Optimization* in Large-Scale Nonlinear Optimization, G. di Pillo and M. Roma, eds, pp. 35-59 (2006), Springer-Verlag.

http://www.ziena.com/knitro.htm

la04

LA04 is a steepest-edge simplex method for linear programming by John Reid frm the Rutherford Appleton Laboratory.

http://www.hsl.rl.ac.uk/catalogue/la04.xml

lbfgs See D.C. Liu and J. Nocedal, *On the Limited Memory Method for Large Scale Optimization* Mathematical Programming B, 45(3) 503-528 (1989).

http://users.eecs.northwestern.edu/˜nocedal/lbfgs.html

lbfgsb

See C. Zhu, R. H. Byrd and J. Nocedal. *L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization* ACM Transactions on Mathematical Software, 23(4) 550-560 (1997).

http://users.eecs.northwestern.edu/˜nocedal/lbfgsb.html

loqo

See R. J. Vanderbei and D. F. Shanno *An Interior-Point Algorithm for Nonconvex Nonlinear Programming*, 13 (1-3) pp 231-252 (1999).

http://www.princeton.edu/˜rvdb/loqo/LOQO.html

matlab

Creates a Matlab binary to allow CUTEst calls from Matlab. See $CUTEST/src/matlab/README.matlab to see how to use the binary with Matlab. Note that there is a simplified interface **cutest2matlab** that may be used in preference.

minos

See B. A. Murtagh and M. A. Saunders. *A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints*, Mathematical Programming Study 16, 84-117 (1982).

http://www.sbsi-sol-optimize.com/asp/sol_product_minos.htm

nitsol

See M. Pernice and H. F. Walker, *NITSOL: a Newton iterative solver for nonlinear systems*, Special Issue on Iterative Methods, SIAM J. Sci. Comput., 19, 302-318 (1998).

http://users.wpi.edu/˜walker/NITSOL/

npsol

A linesearch SQP method for constrained optimization by Philip Gill, Walter Murray, Michael Saunders and Margaret Wright from Stanford University.

http://www.sbsi-sol-optimize.com/asp/sol_product_npsol.htm

newuoa
See M.J.D. Powell, *The NEWUOA software for unconstrained optimization without derivatives*, in, G. Di Pillo and M. Roma (eds), Large-Scale Nonlinear Optimization, volume 83 of Nonconvex Optimization and Its Applications pp 255-297, Springer Verlag, 2006.

pds
Direct search methods for unconstrained optimization on either sequential or parallel machines by Virginia Torczon from The College of William and Mary.

praxis
Brent's multi-dimensional direct search unconstrained minimization algorithm, as implemented by John Chandler, Sue Pinsk and Rosalee Taylor from Oklahoma State University.

http://people.sc.fsu.edu/˜jburkardt/f_src/praxis/praxis.html

snopt
See P. E. Gill, W. Murray and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Review 47(1) 99-131 (2005).

http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm

spg
See E. G. Birgin, J. M. Martinez and M. Raydan, *Algorithm 813: SPG - software for convex-constrained optimization*, ACM Transactions on Mathematical Software 27 340-349, (2001).

http://www.ime.usp.br/˜egbirgin/tango/codes.php

stats The package collects statistics about the types of variables and constraints involved in a given problem, and was written by Dominique Orban from Ecole Polytechnique de Montreal.

stenmin
See A. Bouaricha, *Algorithm 765: STENMIN - a software package for large, sparse unconstrained optimization using tensor methods*, ACM Transactions on Mathematical Software, 23(1) 81-90 (1997).

http://www.netlib.org/toms/765

tao TAO is an object-oriented package for large-scale optimization written by Todd Munson, Jason Sarich, Stefan Wild, Steven Benson and Lois Curfman McInnes,

http://www.mcs.anl.gov/research/projects/tao/

tenmin
See R.B. Schnabel and T.-T. Chow, R. B. Schnabel and T.-T. Chow, *Algorithm 739: A software package for unconstrained optimization using tensor methods*, ACM Transactions on Mathematical Software, 20(4) 518-530 (1994).

http://www.netlib.org/toms/739

tron See C. Lin and J. J. More', *Newton's method for large bound-constrained optimization problems*, SIAM J. Optimization 9(4) 1100-1127 (1999).

http://www.mcs.anl.gov/˜more/tron/

uncmin

See J. E. Koontz, R.B. Schnabel, and B.E. Weiss, *A modular system of algorithms for unconstrained minimization*, ACM Transactions on Mathematical Software, 11(4) 419-440 (1985).

vf13

VF13 is a line-search SQP method for constrained optimization by Mike Powell from the University of Cambridge.

http://www.hsl.rl.ac.uk/archive/index.html

Interfaces to the obsolete packages *hsl_ve12*, *osl*, *va15*, *ve09* and *ve14* previously supported in CUTEr have been withdrawn.

The packages *derchk*, *gen77/90/c*, *hrb* and *stats* are supplied as part of the CUTEst distribution and should work "as is". Anyone wishing to use one of remaining packages will need to download and install it first. See the README in the relevant subdirectory of $CUTEST/src for further instructions.

A file with each of supported package's name may be found in the directory $CUTEST/packages/ and indicates default locations for the package's binary and options files. These files may be edited if necessary, or copied into $CUTEST/packages/(arrchitecture)/(precision)/ to allow for architecture or precision specfic settings; **runcutest** will use the architecture/precision specfic directory version, if any, in preference to the default version.

ENVIRONMENT

CUTEST

Directory containing CUTEst.

SIFDECODE

Directory containing SIFDecode.

MYARCH

The default architecture.

MASTSIF

A pointer to the directory containing the CUTEst problems collection. If this variable is not set, the current directory is searched for *problem.SIF*. If it is set, the current directory is searched first, and if *problem.SIF* is not found there, $MASTSIF is searched.

AUTHORS

I. Bongartz, A.R. Conn, N.I.M. Gould, D. Orban and Ph.L. Toint

SEE ALSO

CUTEr (and SifDec): A Constrained and Unconstrained Testing Environment, revisited,
N.I.M. Gould, D. Orban and Ph.L. Toint,
ACM TOMS, **29**:4, pp.373-394, 2003.

CUTE: Constrained and Unconstrained Testing Environment,
I. Bongartz, A.R. Conn, N.I.M. Gould and Ph.L. Toint,
TOMS, **21**:1, pp.123-160, 1995.

sifdecoder(1), cutest2matlab(1).

NAME

runcutest − CUTEst interface to solvers.

SYNOPSIS

runcutest --package *pack* [−−architecture *arch*] [−−single] [−−help] [−−keep] [−−rebuild] [−−output *0|1*] [−−limit *secs*] [−−cfortran] [−−debug] [−−uncons] [−L*path/to/lib*] [−−blas *keyword*] [−−lapack *keyword*] [−−decode *problem[.SIF]*]

DESCRIPTION

runcutest is the CUTEst interface to solvers. It replaces its predecessor CUTEr's combination of **runpackage**, *pkg* and **sd***pkg*. The command accepts options in short or long form. Any option that is not directly recognized is passed unchanged to the SIF decoder, sifdecoder(1).

runcutest reads suitable architecture-dependent environment variables and then compiles and links all the relevant source files and libraries to form an executable of the package *package* running on problem *problem*.

The user has the opportunity to run commands before and after the run if need be. **runcutest** executes the script *package*_pre, if it exists, before the run. Similarly, it executes the script *package*_post, if it exists, after completion of the run.

runcutest Options

You can start runcutest with the following options. An option can be used either in short or long form.

−p, −−package *pack*
Specifies the package or solver, *pack* to use. See the section **Currently Supported Packages** below. This is the only mandatory option.

−A, −−architecture *arch*
Run the decoder using the architecture *arch*; the architeture is a string of the form machine.system.compiler as specified in the directory $CUTEST/versions. If no −A option is given, a valid architecture given by the environment variable $MYARCH will be used, but if $MYARCH is invalid or empty the decoder will terminate.

−sp, −−single
Run *package* in single-precision mode if available. Double precision is the default.

−h, −−help
Print a short help message with the available command-line options.

−k, −−keep
Keep the generated executable after use. May be useful when solving a particular problem with the same solver with different parameters. Deleting the executable after use is the default.

−r, −−rebuild
Force recompilation of the test problem. Default is to reuse object files.

−o, −−output *0|1*
Regulates the output level. Verbose mode is **-o** 1, silent mode is **-o** 0. Silent mode is the default.

−l, −−limit *secs*
Sets a limit of *secs* second on the *package* runtime. Unlimited cputime is the default.

−c, −−cfortran
Causes specialized compiler options to be used to specify that the main subroutine of *package* is written in C. This is necessary with some compilers, such as the Intel Fortran Compiler.

−L*path/to/lib*
This option is passed directly to the linker and causes the path *path/to/lib* to be searched for libraries. Useful to specify custom BLAS and LAPACK libraries.

−b, −−blas *keyword*

Overrides usage of the default *linpack* library packaged with CUTEst. Instead, use the BLAS library specified by *keyword*. The keyword *keyword* has one of two forms. The first, *-lmyblas* causes the linker to search for BLAS subprograms in the libmyblas.a library. The second, *none*, causes the linker to skip inclusion of any external BLAS. Use the first option if an optimized BLAS library is available on the host system, e.g., the ATLAS BLAS. The second option is useful for packages which already include the necessary BLAS subprograms. Use of *none* may be useful if *package* already includes the BLAS subroutines on which it relies.

−K, −−lapack *keyword*

Overrides usage of the default *linpack* library packaged with CUTEst. Instead, use the LAPACK library specified by *keyword*. The keyword *keyword* has one of two forms. The first, *-lmylapack* causes the linker to search for LAPACK subroutines in the libmylapack.a library. The second, *none*, causes the linker to skip inclusion of any external LAPACK. Use the first option if an optimized LAPACK library is available on the host system. The second option is useful for packages which already include the necessary LAPACK subprograms. Use of *none* may be useful if *package* already includes the LAPACK subroutines on which it relies.

−D, −−decode *problem[.SIF]*

Applies the SIF decoder to the problem *problem.SIF* to produce the OUTSDIF.d file and the problem-dependant Fortan subroutines. If this flag is not specified, **runcutest** assumes that the problem has been decoded prior to the call.

−u, −−uncons

When *package* is **mx**, the Matlab interface, this option specifies that the problem is unconstrained. This causes the appropriate MEX interface to be compiled and linked. The default is to link with the constrained tools.

additional command-line options

Any command-line option not documented in this manual page and/or in the help message of **runcutest** is passed unchanged to the SIF decoder. See the **sifdecode** manual page for more information.

CURRENTLY SUPPORTED PACKAGES

There are currently interfaces to the following packages:

algencan

See E. G. Birgin, R. Castillo and J. M. Martinez, *Numerical comparison of Augmented Lagrangian algorithms for nonconvex problems*, Computational Optimization and Applications 31, 31-56 (2005).

http://www.ime.usp.br/~egbirgin/tango/codes.php

bobyqa

See M.J.D. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives*, Technical report NA2009/06 Department of Applied Mathematics and Theoretical Physics, Cambridge England, (2009).

cg_descent

See W. W. Hager and H. Zhang, *Algorithm 851: CG_DESCENT, A conjugate gradient method with guaranteed descent*, ACM Transactions on Mathematical Software, 32, 113-137 (2006).

http://www.math.ufl.edu/~hager/papers/CG/

cgplus

The CG+ package is a nonlinear conjugate-gradient algorithm designed for unconstrained minimization by G. Liu, Jorge Nocedal and Richard Waltz (Northwestern U.).

http://users.eecs.northwestern.edu/˜nocedal/CG+.html

cobyla

See M.J.D. Powell, *A direct search optimization method that models the objective and constraint functions by linear interpolation*, In Advances in optimization and numerical analysis, Proceedings of the Sixth workshop on Optimization and Numerical Analysis, Oaxaca, Mexico, volume 275 of Mathematics and its Applications, pp 51--67. Kluwer Academic Publishers (1994).

derchk

This package checks the dervatives supplied in the problem SIF file, and is due to Dominique Orban from Ecole Polytechnique de Montreal.

dfo

See A. R. Conn, K. Scheinberg and Ph.L. Toint, *On the convergence of derivative-free methods for unconstrained optimization*, Approximation Theory and Optimization: Tributes to M. J. D. Powell , Eds. A. Iserles and M. Buhmann, 83-108, Cambridge University Press (1997).

https://projects.coin-or.org/Dfo

direstsearch

The Direct Search suite provides a variety of patern search methods for derivative-free optimization and was written by Liz Dolan, Adam Gurson, Anne Shepherd, Chris Siefert, Virginia Torczon and Amy Yates.

http://www.cs.wm.edu/˜va/software/DirectSearch/direct_code/

filtersd

See R. Fletcher *A sequential linear constraint programming algorithm for NLP*, SIAM Journal on Optimization, 22(3), pp. 772-79 (2012).

http://www.coin-or.org/projects/filterSD.xml

filtersqp

FilterSQP is a filter-based SQP method for large-scale nonlinear programming by Roger Fletcher and Sven Leyffer from the University of Dundee.

*gen77***,** *gen90***,** *genc*

These package simply illustrates how CUTEst tools may be called in fortran 77, fortran 90 and C; the result is of no consequence.

hrb

This package writes the matrix data for the given problem in Harwell or Rutherford-Boeing sparse matrix forrmat, and was provded by Nick Gould from the Rutherford Appleton Laboratory.

ipopt

See A. WÃ¤chter and L. T. Biegler, *On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming*, Mathematical Programming 106(1) 25-57 (2006).

https://projects.coin-or.org/Ipopt

knitro

See R. H. Byrd, J. Nocedal, and R. A. Waltz, *KNITRO: An Integrated Package for Nonlinear Optimization* in Large-Scale Nonlinear Optimization, G. di Pillo and M. Roma, eds, pp. 35-59 (2006), Springer-Verlag.

http://www.ziena.com/knitro.htm

la04

LA04 is a steepest-edge simplex method for linear programming by John Reid frm the Rutherford Appleton Laboratory.

http://www.hsl.rl.ac.uk/catalogue/la04.xml

lbfgs See D.C. Liu and J. Nocedal, *On the Limited Memory Method for Large Scale Optimization* Mathematical Programming B, 45(3) 503-528 (1989).

http://users.eecs.northwestern.edu/~nocedal/lbfgs.html

lbfgsb

See C. Zhu, R. H. Byrd and J. Nocedal. *L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization* ACM Transactions on Mathematical Software, 23(4) 550-560 (1997).

http://users.eecs.northwestern.edu/~nocedal/lbfgsb.html

loqo

See R. J. Vanderbei and D. F. Shanno *An Interior-Point Algorithm for Nonconvex Nonlinear Programming*, 13 (1-3) pp 231-252 (1999).

http://www.princeton.edu/~rvdb/loqo/LOQO.html

matlab

Creates a Matlab binary to allow CUTEst calls from Matlab. See $CUTEST/src/matlab/README.matlab to see how to use the binary with Matlab. Note that there is a simplified interface **cutest2matlab** that may be used in preference.

minos

See B. A. Murtagh and M. A. Saunders. *A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints*, Mathematical Programming Study 16, 84-117 (1982).

http://www.sbsi-sol-optimize.com/asp/sol_product_minos.htm

nitsol

See M. Pernice and H. F. Walker, *NITSOL: a Newton iterative solver for nonlinear systems*, Special Issue on Iterative Methods, SIAM J. Sci. Comput., 19, 302-318 (1998).

http://users.wpi.edu/~walker/NITSOL/

npsol

A linesearch SQP method for constrained optimization by Philip Gill, Walter Murray, Michael Saunders and Margaret Wright from Stanford University.

http://www.sbsi-sol-optimize.com/asp/sol_product_npsol.htm

newuoa

See M.J.D. Powell, *The NEWUOA software for unconstrained optimization without derivatives*, in, G. Di Pillo and M. Roma (eds), Large-Scale Nonlinear Optimization, volume 83 of Nonconvex Optimization and Its Applications pp 255-297, Springer Verlag, 2006.

pds

Direct search methods for unconstrained optimization on either sequential or parallel machines by Virginia Torczon from The College of William and Mary.

praxis

Brent's multi-dimensional direct search unconstrained minimization algorithm, as implemented by John Chandler, Sue Pinsk and Rosalee Taylor from Oklahoma State University.

http://people.sc.fsu.edu/˜jburkardt/f_src/praxis/praxis.html

snopt

See P. E. Gill, W. Murray and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Review 47(1) 99-131 (2005).

http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm

spg

See E. G. Birgin, J. M. Martinez and M. Raydan, *Algorithm 813: SPG - software for convex-constrained optimization*, ACM Transactions on Mathematical Software 27 340-349, (2001).

http://www.ime.usp.br/˜egbirgin/tango/codes.php

stats The package collects statistics about the types of variables and constraints involved in a given problem, and was written by Dominique Orban from Ecole Polytechnique de Montreal.

stenmin

See A. Bouaricha, *Algorithm 765: STENMIN - a software package for large, sparse unconstrained optimization using tensor methods*, ACM Transactions on Mathematical Software, 23(1) 81-90 (1997).

http://www.netlib.org/toms/765

tao TAO is an object-oriented package for large-scale optimization written by Todd Munson, Jason Sarich, Stefan Wild, Steven Benson and Lois Curfman McInnes,

http://www.mcs.anl.gov/research/projects/tao/

tenmin

See R.B. Schnabel and T.-T. Chow, R. B. Schnabel and T.-T. Chow, *Algorithm 739: A software package for unconstrained optimization using tensor methods*, ACM Transactions on Mathematical Software, 20(4) 518-530 (1994).

http://www.netlib.org/toms/739

tron See C. Lin and J. J. More', *Newton's method for large bound-constrained optimization problems*, SIAM J. Optimization 9(4) 1100-1127 (1999).

http://www.mcs.anl.gov/˜more/tron/

uncmin

See J. E. Koontz, R.B. Schnabel, and B.E. Weiss, *A modular system of algorithms for unconstrained minimization*, ACM Transactions on Mathematical Software, 11(4) 419-440 (1985).

vf13

VF13 is a line-search SQP method for constrained optimization by Mike Powell from the University of Cambridge.

http://www.hsl.rl.ac.uk/archive/index.html

Interfaces to the obsolete packages *hsl_ve12*, *osl*, *va15*, *ve09* and *ve14* previously supported in CUTEr have been withdrawn.

The packages *derchk*, *gen77/90/c*, *hrb* and *stats* are supplied as part of the CUTEst distribution and should work "as is". Anyone wishing to use one of remaining packages will need to download and install it first. See the README in the relevant subdirectory of $CUTEST/src for further instructions.

A file with each of supported package's name may be found in the directory $CUTEST/packages/ and indicates default locations for the package's binary and options files. These files may be edited if necessary, or copied into $CUTEST/packages/(arrchitecture)/(precision)/ to allow for architecture or precision specfic settings; **runcutest** will use the architecture/precision specfic directory version, if any, in preference to the default version.

ENVIRONMENT

CUTEST

Directory containing CUTEst.

SIFDECODE

Directory containing SIFDecode.

MYARCH

The default architecture.

MASTSIF

A pointer to the directory containing the CUTEst problems collection. If this variable is not set, the current directory is searched for *problem.SIF*. If it is set, the current directory is searched first, and if *problem.SIF* is not found there, $MASTSIF is searched.

AUTHORS

I. Bongartz, A.R. Conn, N.I.M. Gould, D. Orban and Ph.L. Toint

SEE ALSO

CUTEr (and SifDec): A Constrained and Unconstrained Testing Environment, revisited,
N.I.M. Gould, D. Orban and Ph.L. Toint,
ACM TOMS, **29**:4, pp.373-394, 2003.

CUTE: Constrained and Unconstrained Testing Environment,
I. Bongartz, A.R. Conn, N.I.M. Gould and Ph.L. Toint,
TOMS, **21**:1, pp.123-160, 1995.

sifdecoder(1), cutest2matlab(1).