DERIVATIVE-FREE OPTIMIZATION
Algorithms, software and applications

Nick Sahinidis
National Energy Technology Laboratory
Department of Chemical Engineering
Carnegie Mellon University
sahinidis@cmu.edu

Acknowledgments:
Luis Miguel Rios
NIH and DOE/NETL

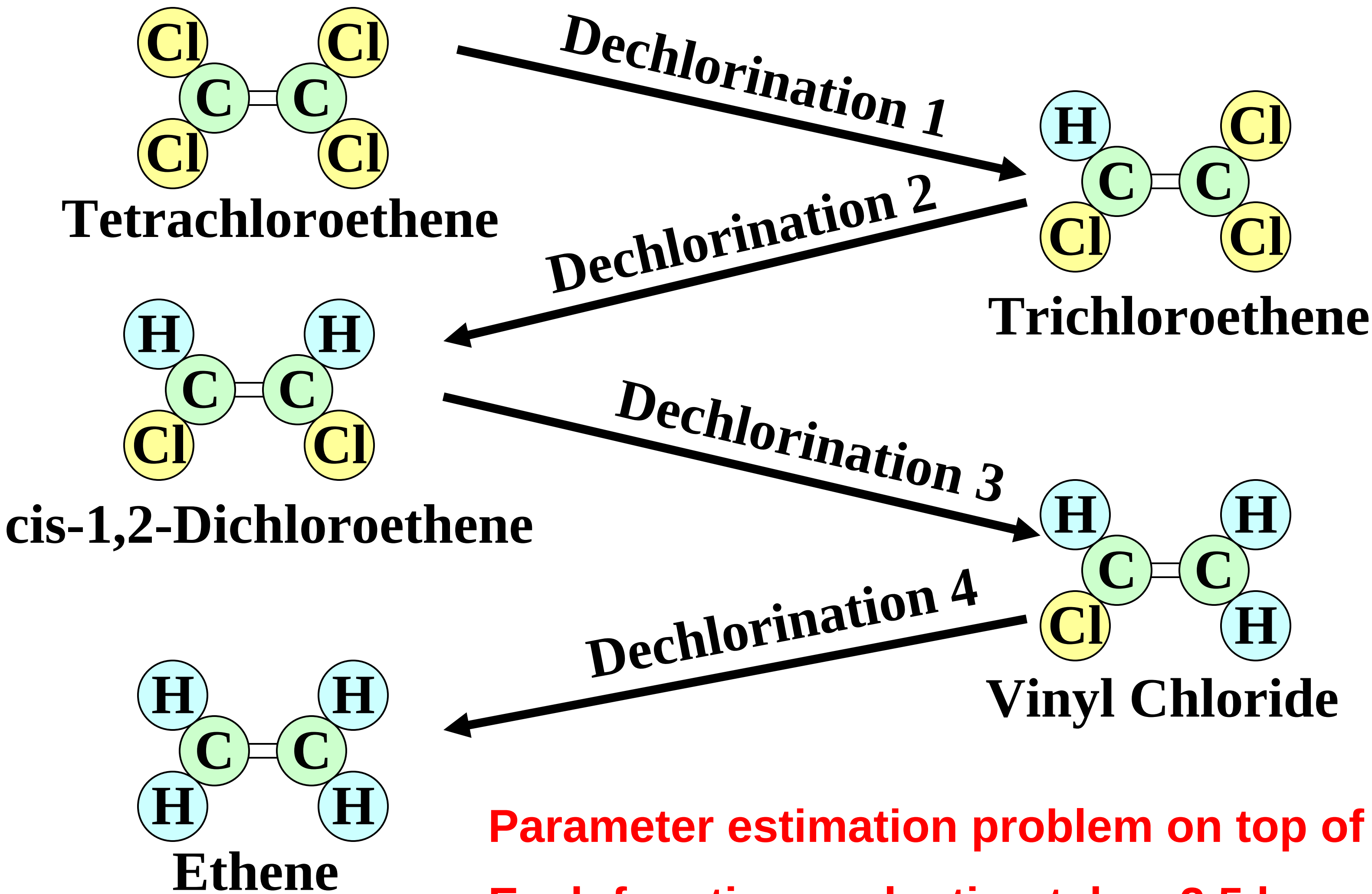
DERIVATIVE-FREE OPTIMIZATION

- **Optimization of a function for which**
  - **derivative information is not symbolically available**
  - **derivative information is not numerically computable**

- **Talk outline**
  - **Motivation**
  - **Review of algorithms and software**
  - **Application to protein-ligand binding**
  - **Two new algorithms**

MODEL CALIBRATION

(Maguthan and Shoemaker, 2005)



Parameter estimation problem on top of PDEs

Each function evaluation takes 2.5 hours

APPLICATIONS

- Parameter estimation over differential equations
- Optimal control problems
- Simulation-based optimization
  - Objective computation may involve sampling
- Automatic calibration of optimization algorithms
- Experimental design/optimization

TIMELINE OF INNOVATION

1960

- 1961 Hooke and Jeeves algorithm is proposed
- 1962 First simplex-based optimization algorithm
- 1965 Nelder-Mead simplex algorithm is proposed
- 1969 First use of trust-region quadratic-based models

1970

- 1973 First published monograph
- 1975 Genetic algorithms are proposed
- 1979 Hit-and-run algorithms are proposed

1980

- 1983 First use of simulated annealing in optimization
- 1989 DACE stochastic model is proposed

1990

- 1991 Convergence of multi-directional search algorithm is shown
- 1993 Ideas from Lipschitzian optimization introduced
- 1994 Geometry considerations for points in trust-region methods
- 1995 Implicit filtering and particle swarm algorithms proposed
- 1997 DACE surrogate model introduced
- 1998 First use of radial basis functions in surrogate models
- 1999 Introduction of multilevel coordinate search

2000

- 2002 First use of augmented Lagrangian in pattern search methods
- 2003 Generating set nomenclature introduced
- 2004 Incorporation of filters and simplex derivatives in pattern search
- 2009 First textbook

MOST CITED WORKS

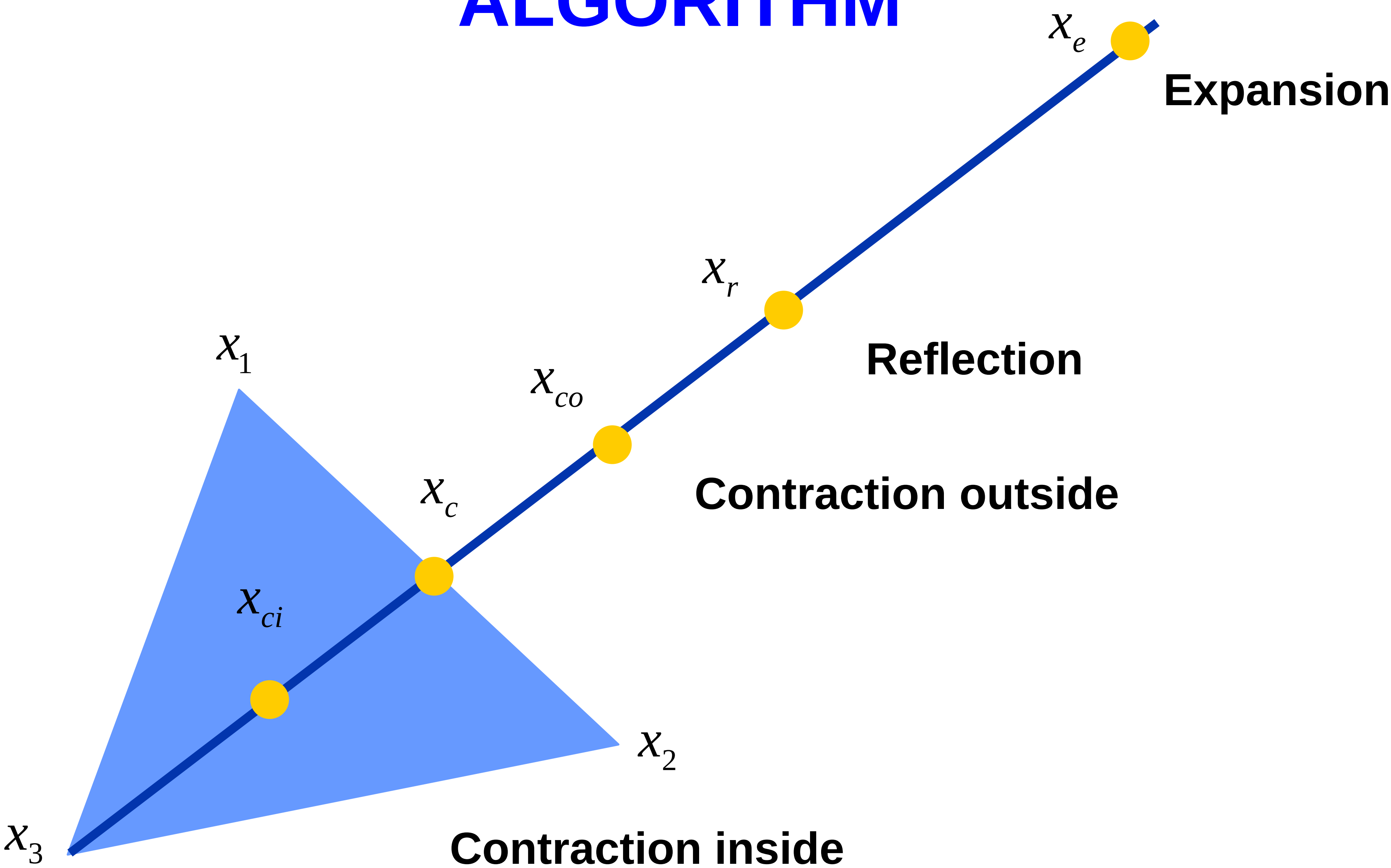
Publication	Year appeared	Citations[1]
Hooke and Jeeves [59]	1961	1567
Nelder and Mead [92]	1965	9101
Holland [53]	1975	22277
Kirkpatrick *et al.* [72]	1983	16109
Eberhart and Kennedy [39, 71]	1995	10099

1. From Google Scholar on 16 December 2009.

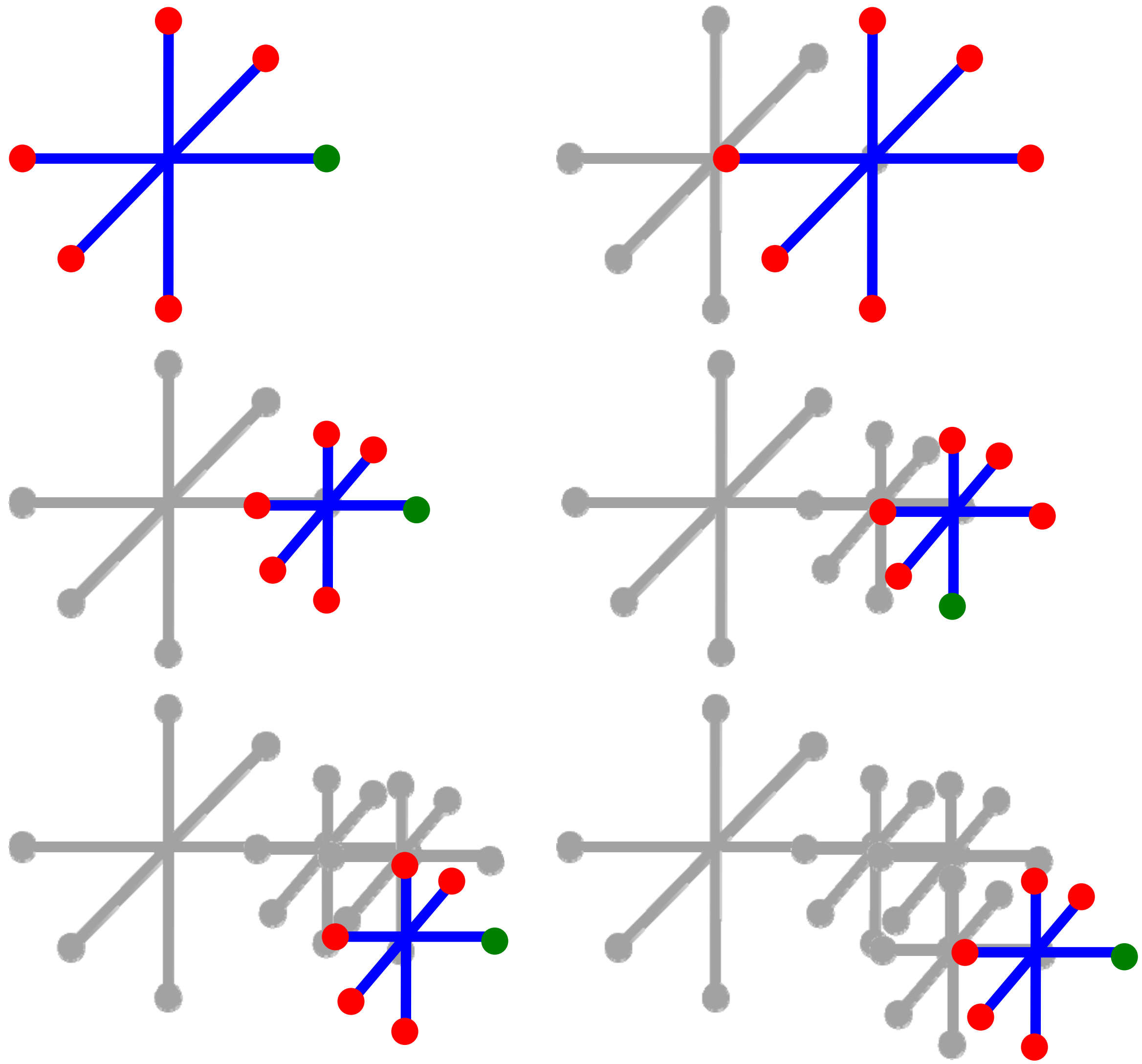
DERIVATIVE-FREE OPTIMIZATION ALGORITHMS

- LOCAL SEARCH METHODS
  - Direct local search
    - Nelder-Mead simplex algorithm
    - Generalized pattern search and generating search set
  - Based on surrogate models
    - Trust-region methods
    - Implicit filtering
- GLOBAL SEARCH METHODS
  - Deterministic global search
    - Lipschitzian-based partitioning
    - Multilevel coordinate search
  - Stochastic global optimization
    - Hit-and-run
    - Simulated annealing
    - Genetic algorithms
    - Particle swarm
  - Based on surrogate models
    - Response surface methods
    - Surrogate management framework
    - Branch-and-fit

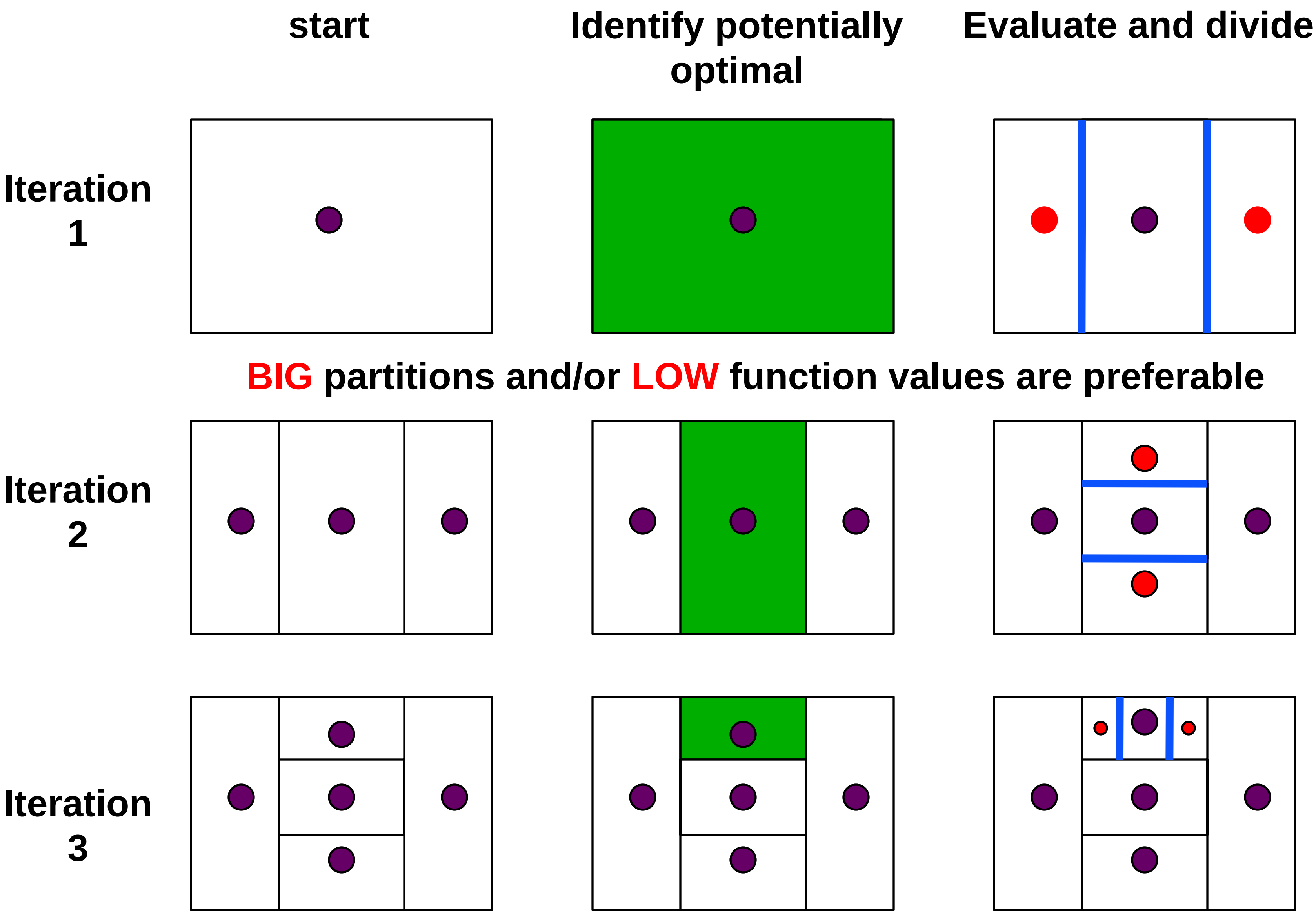
NELDER-MEAD SIMPLEX ALGORITHM



PATTERN SEARCH ALGORITHMS



DIRECT ALGORITHM



ALGORITHMIC COMPONENTS

- **Random elements**
  - **Deterministic vs. stochastic**
- **Set of points considered in each iteration**
  - **None; One; Many**
- **Partitioning**
  - **Without: local optimality**
    - » **Torczon (1991)**
  - **With: global optimality, provided search is “dense”**

DERIVATIVE-FREE OPTIMIZATION SOFTWARE

LOCAL SEARCH

FMINSEARCH (Nelder-Mead)
DAKOTA PATTERN (PPS)
HOPSPACK (PPS)
SID-PSM (Simplex gradient PPS)
NOMAD (MADS)
DFO
(Trust region, quadratic model)
IMFIL (Implicit Filtering)
BOBYQA
(Trust region, quadratic model)
NEWUOA
(Trust region, quadratic model)

GLOBAL SEARCH

DAKOTA SOLIS-WETS (Direct)
DAKOTA DIRECT (DIRECT)
TOMLAB GLBSOLVE (DIRECT)
TOMLAB GLCSOLVE (DIRECT)
MCS (Multilevel coordinate search)
TOMLAB EGO (RSM using Kriging)
TOMLAB RBF (RSM using RBF)
SNOBFIT (Branch and Fit)
TOMLAB LGO (LGO algorithm)

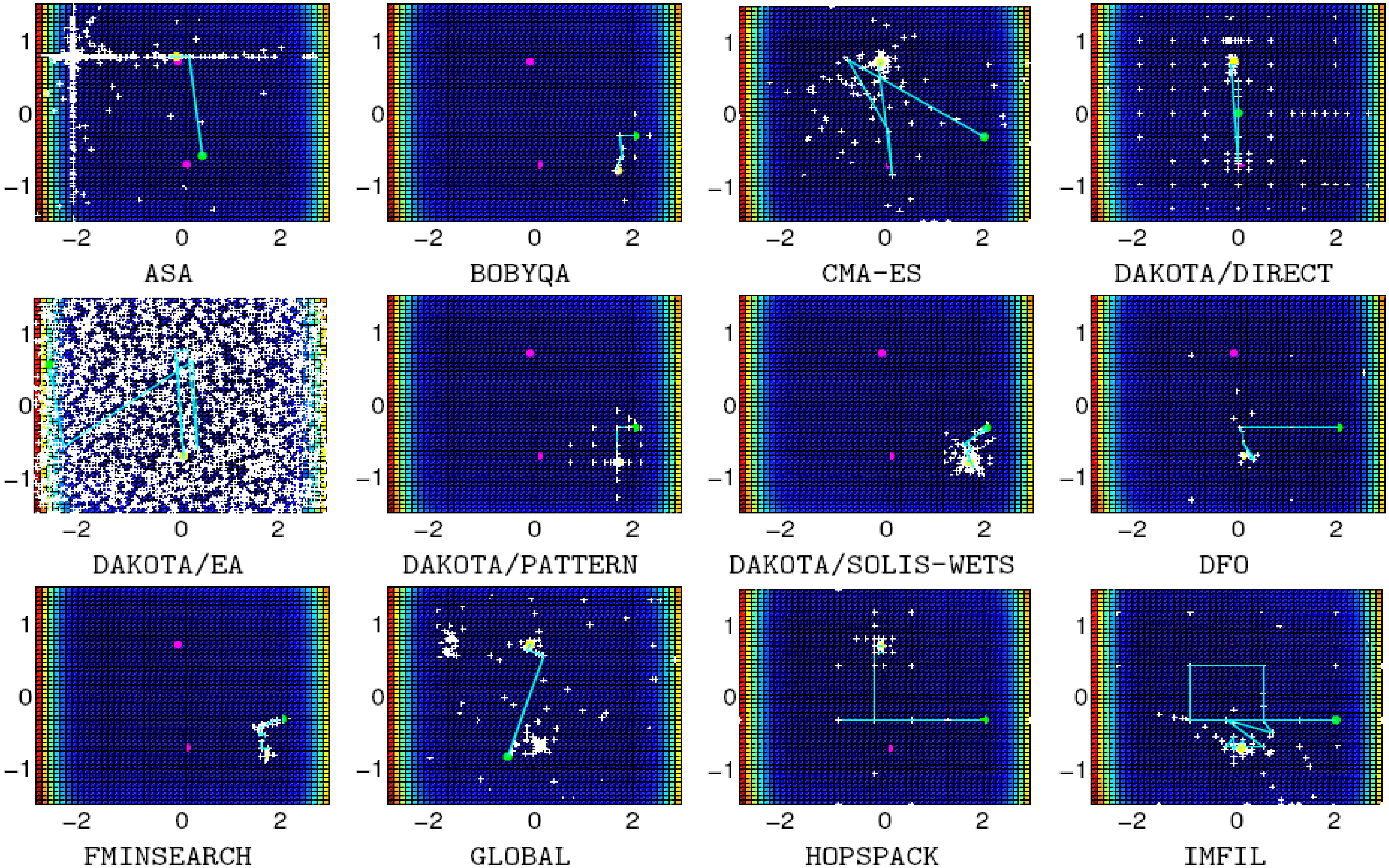
STOCHASTIC

ASA (Simulated annealing)
CMA-ES (Evolutionary algorithm)
DAKOTA EA (Evolutionary algorithm)
GLOBAL (Clustering - Multistart)
PSWARM (Particle swarm)

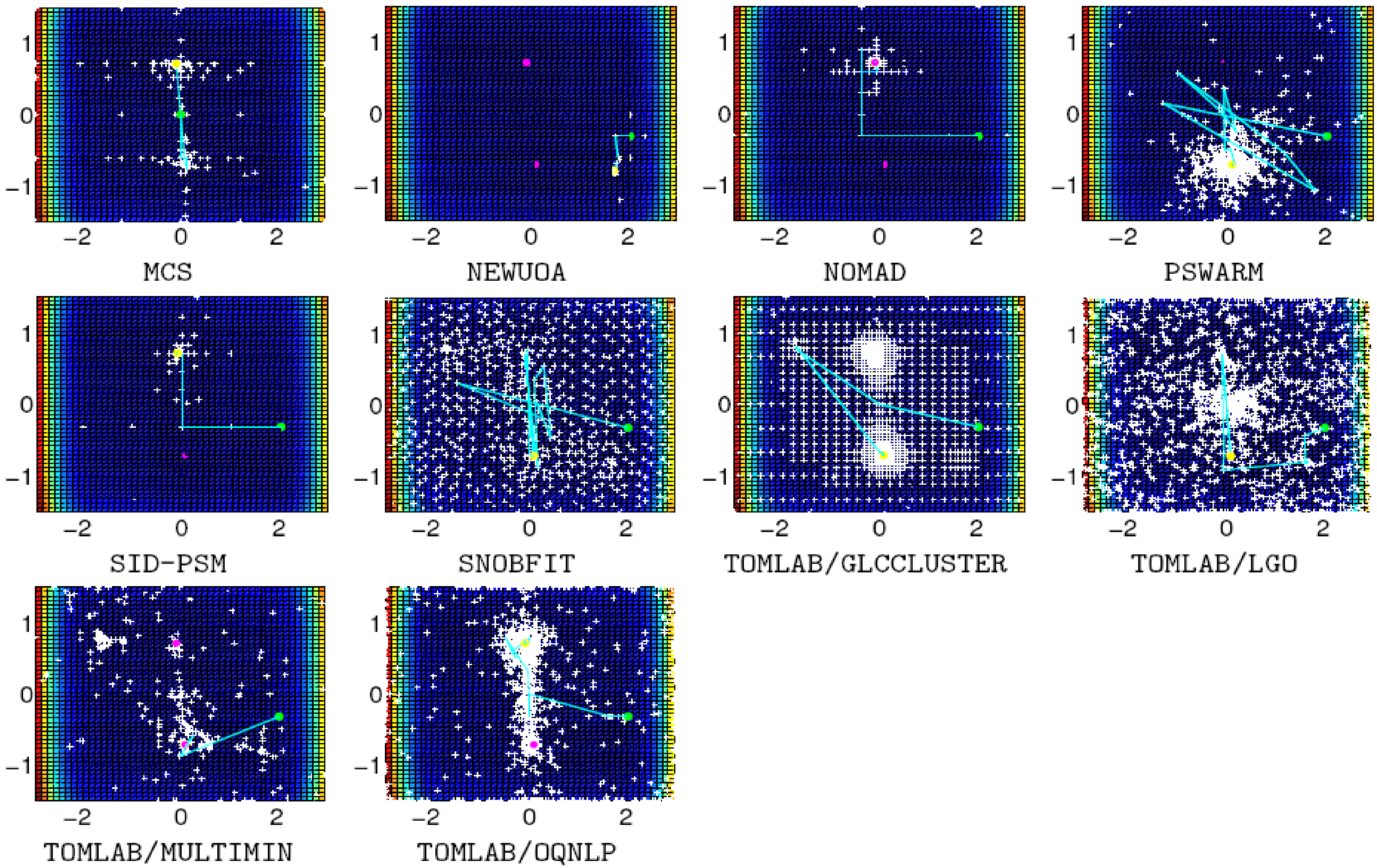
SOLVERS CONSIDERED

Solver	URL	Version	Language	Bounds	Constraints Linear	Constraints Black-box
ASA	www.ingber.com	26.30	C	required	no	no
BOBYQA	N/A	N/A	Fortran	required	no	no
CMA-ES	www.lri.fr/~hansen/ cmaesintro.html	3.26.beta	Matlab	optional	no	no
DAKOTA/DIRECT	www.cs.sandia.gov/dakota/	4.2	C++	required	yes	yes
DAKOTA/EA	www.cs.sandia.gov/dakota/	4.2	C++	required	yes	yes
DAKOTA/PATTERN	www.cs.sandia.gov/dakota/	4.2	C++	required	yes	yes
DAKOTA/SOLIS-WETS	www.cs.sandia.gov/dakota/	4.2	C++	required	yes	yes
DFO	projects.coin-or.org/Dfo	2.0	Fortran	required	yes	yes
FMINSEARCH	www.mathworks.com	N/A	Matlab	not allowed	no	no
GLOBAL	www.inf.u-szeged.hu/~csendes	1.0	Matlab	required	no	no
HOPSPACK	software.sandia.gov/trac/ hopspack	2.0	C++	optional	yes	yes
IMFIL	www4.ncsu.edu/~ctk/imfil.html	0.86	Matlab	required	no	yes
MCS	www.mat.univie.ac.at/~neum/ software/mcs/	2.0	Matlab	required	no	no
NEWUOA	N/A	N/A	Fortran	not allowed	no	no
NOMAD	www.gerad.ca/nomad/	3.3	C++	optional	no	yes
PSWARM	www.norg.uminho.pt/aivaz/ pswarm/	1.3	C, Matlab	required	yes	no
SID-PSM	www.mat.uc.pt/sid-psm/	1.1	Matlab	optional	yes	no
SNOBFIT	www.mat.univie.ac.at/~neum/ software/snobfit/	2.1	Matlab	required	no	no
TOMLAB/GLCCLUSTER	tomopt.com	7.3	Matlab	required	yes	yes
TOMLAB/LGO	www.pinterconsulting.com/	7.3	Matlab	required	yes	yes
TOMLAB/MULTIMIN	tomopt.com	7.3	Matlab	required	yes	yes
TOMLAB/OQNLP	www.opttek.com	7.3	Matlab	required	yes	yes

SEARCH PROGRESS FOR camel6



SEARCH PROGRESS FOR camel6—Continued



TEST PROBLEMS

1. Richtarik's [112] piece-wise linear problems:

$$\min_x \max_i \{ |\langle a_i, x \rangle| : i = 1, 2, \ldots, m \},$$

2. Nesterov's [94] quadratic test problems:

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + \|x\|_1,$$

3. a variant of Nesterov's test problems without the nonsmooth term:

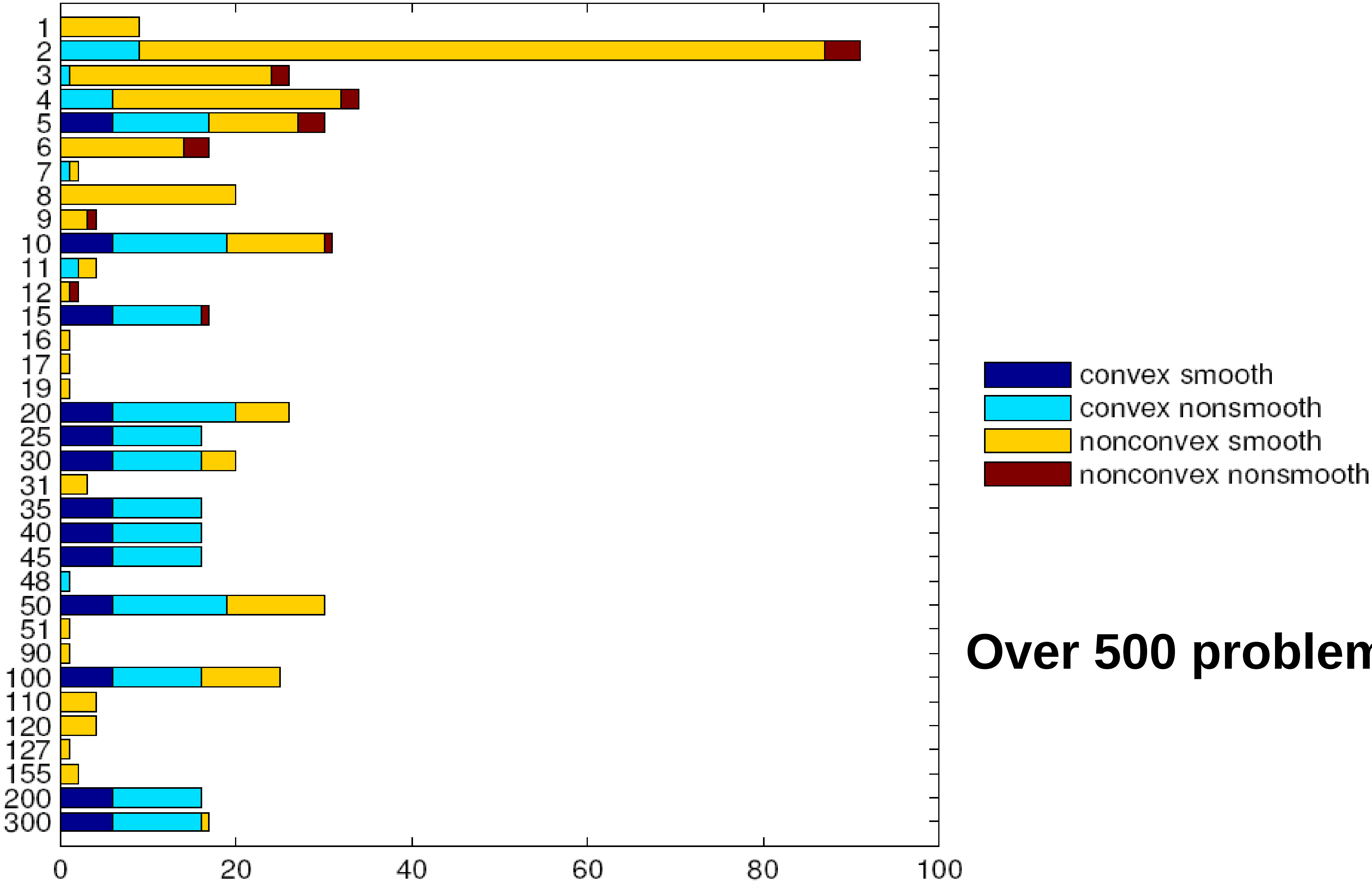
$$\min_x \frac{1}{2} \|Ax - b\|_2^2,$$

4. The ARWHEAD quadratic test problem from Conn *et al.* [28]:

$$\min_x \sum_{i=1}^{n-1} (x_i^2 + x_n^2)^2 - 4x_i + 3,$$

5. 248 nonconvex problems from the `globallib` [47] and `princetonlib` [109],
6. and 49 nonsmooth problems from the collection of Lukšan and Vlček [85].

TEST PROBLEM CHARACTERISTICS



Over 500 problems

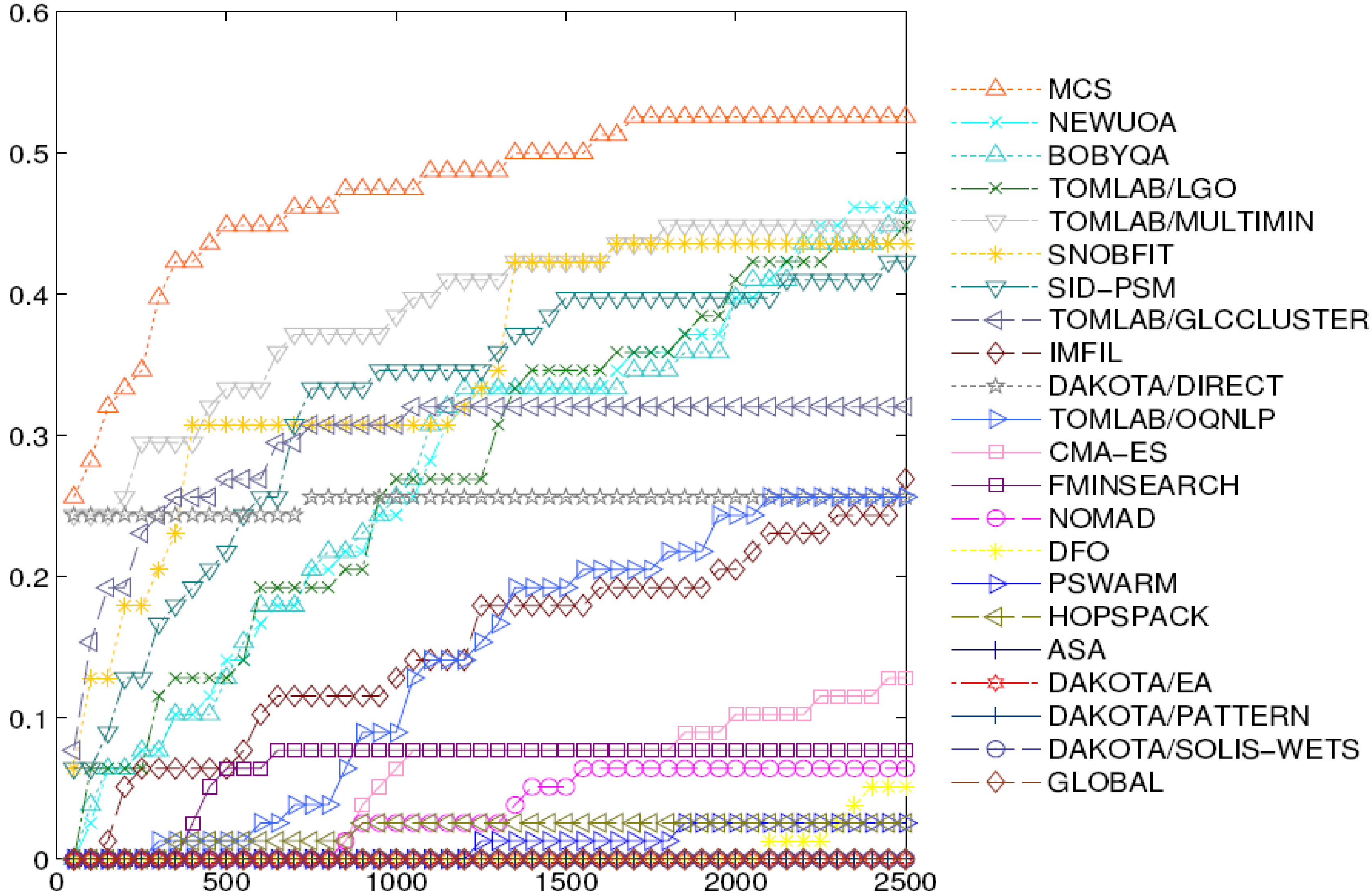
EXPERIMENTAL SETUP

- **For all solvers**
  - **Default settings / non-intrusive interface**
  - **Same bounds; only if required by solver; mostly [-10000, 10000]**
  - **Same starting points**
  - **Limit of 2500 iterations and 600 CPU seconds**
- **BARON and LINDOGlobal used to find global solutions for all problems**
- **Absolute Tolerance of 0.01 or Relative Tolerance of 1% used for solver comparisons**
- **Average-case comparisons based on median objective function value of 10 runs from randomly generated starting points**
  - **But DAKOTA/DIRECT, MCS, TOMLAB/CLUSTER**

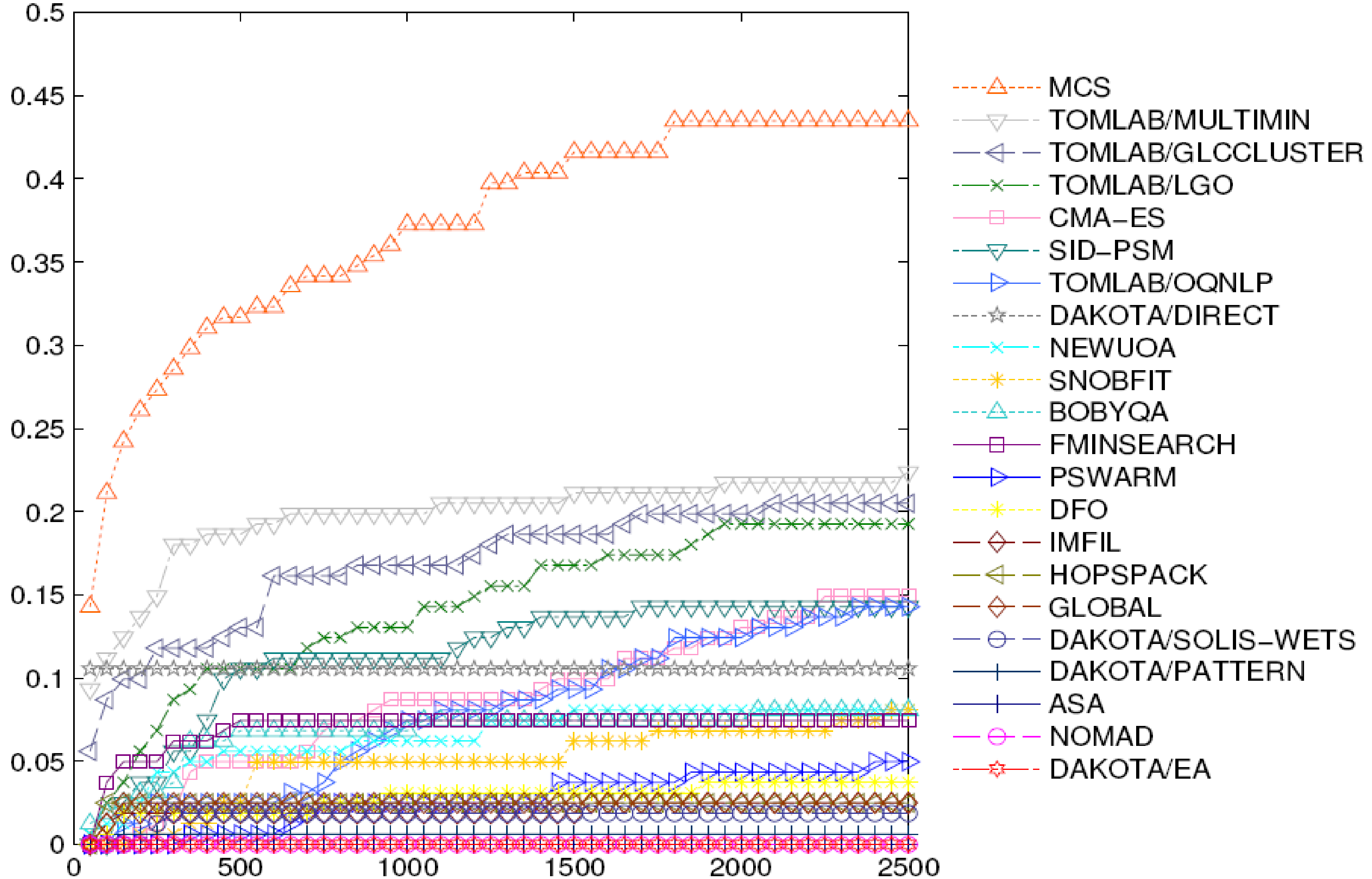
QUESTIONS ADDRESSED

- What is the quality of solutions obtained by current solvers for a given limit on the number of allowable function evaluations?
- Does quality drop significantly as problem size increases?
- Which solver is more likely to obtain global or near-global solutions for nonconvex problems?
- Is there a subset of existing solvers that would suffice to solve a large fraction of problems?

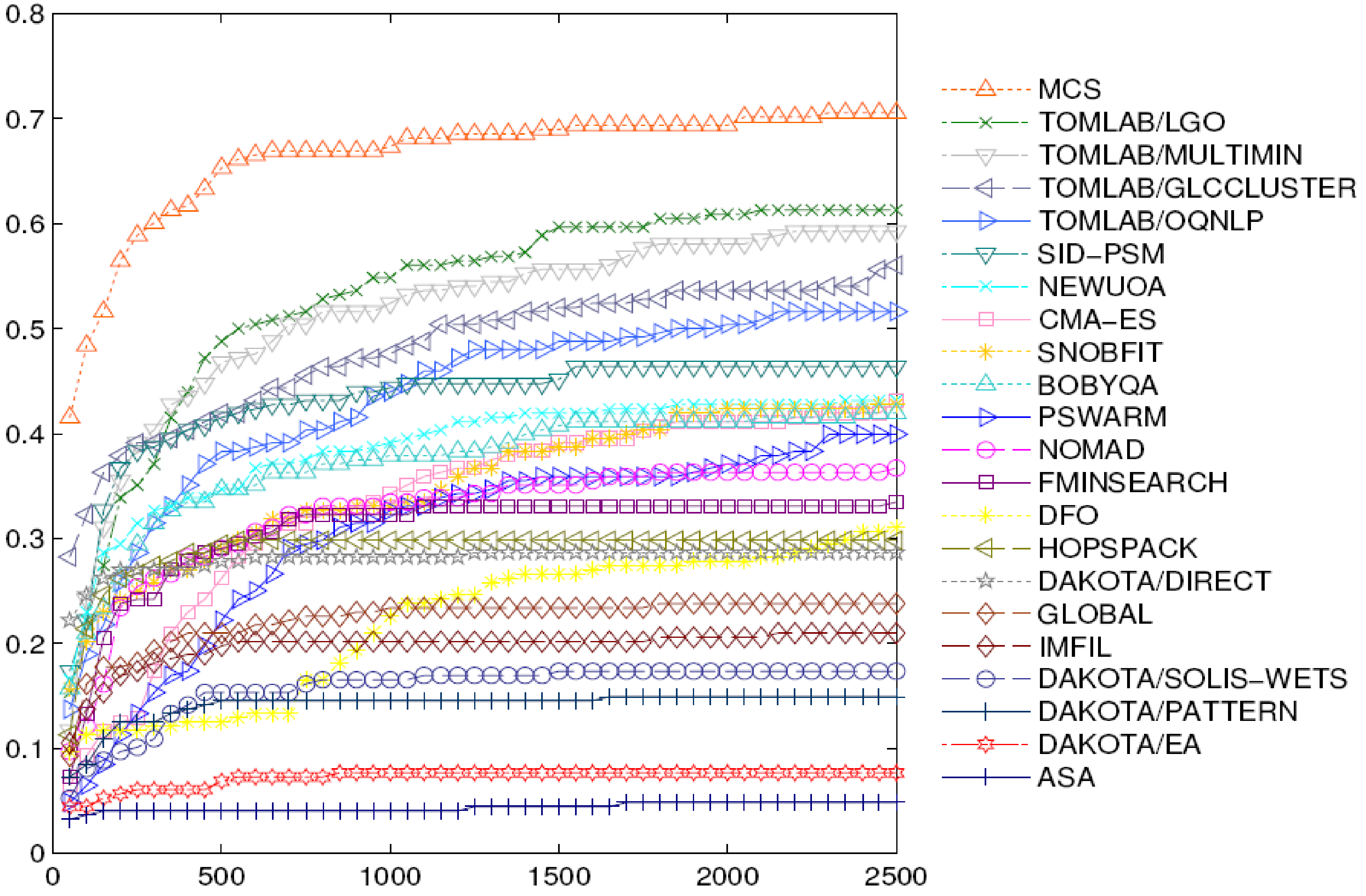
FRACTION OF PROBLEMS SOLVED: CONVEX SMOOTH



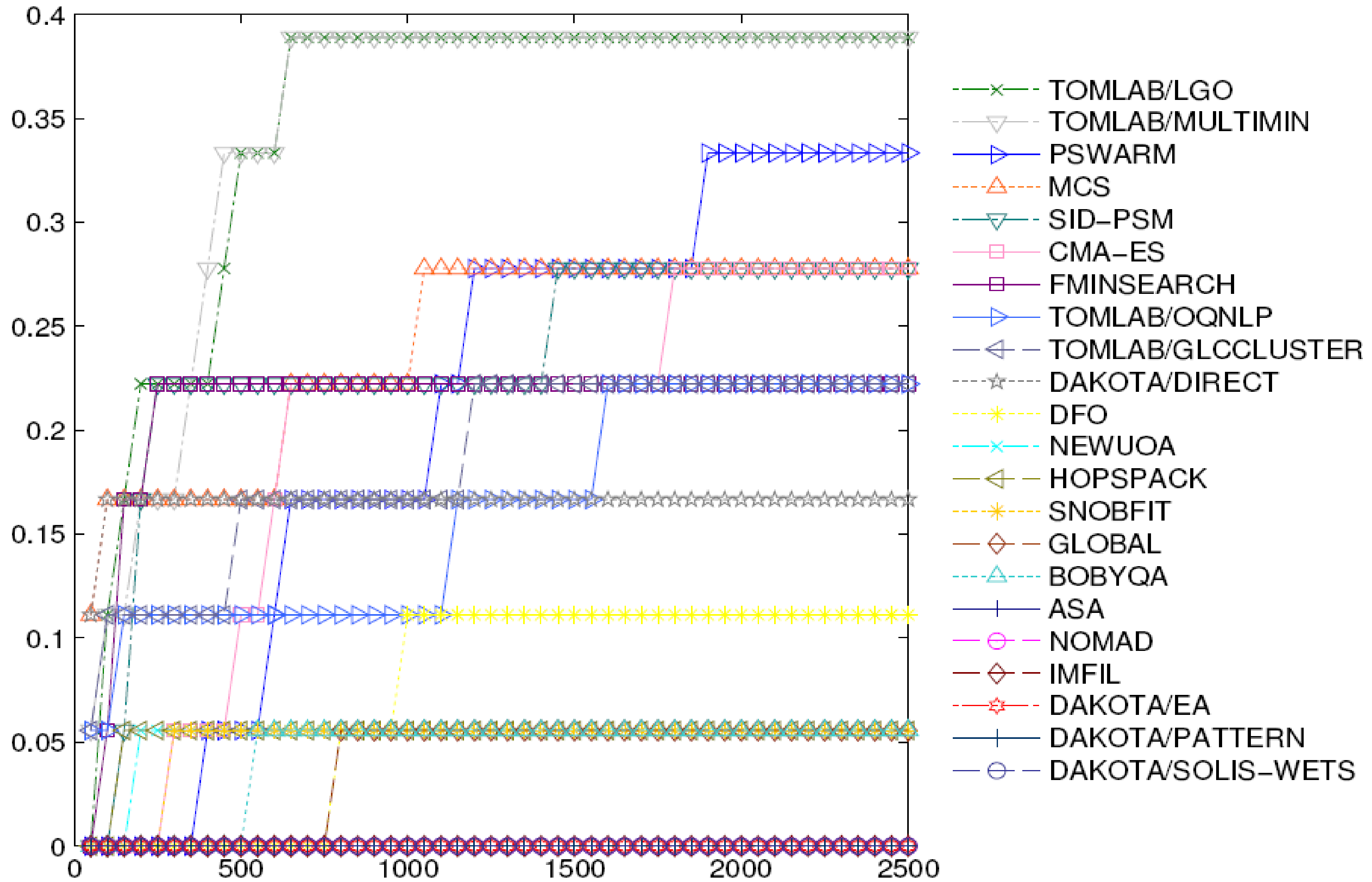
FRACTION OF PROBLEMS SOLVED: CONVEX NONSMOOTH



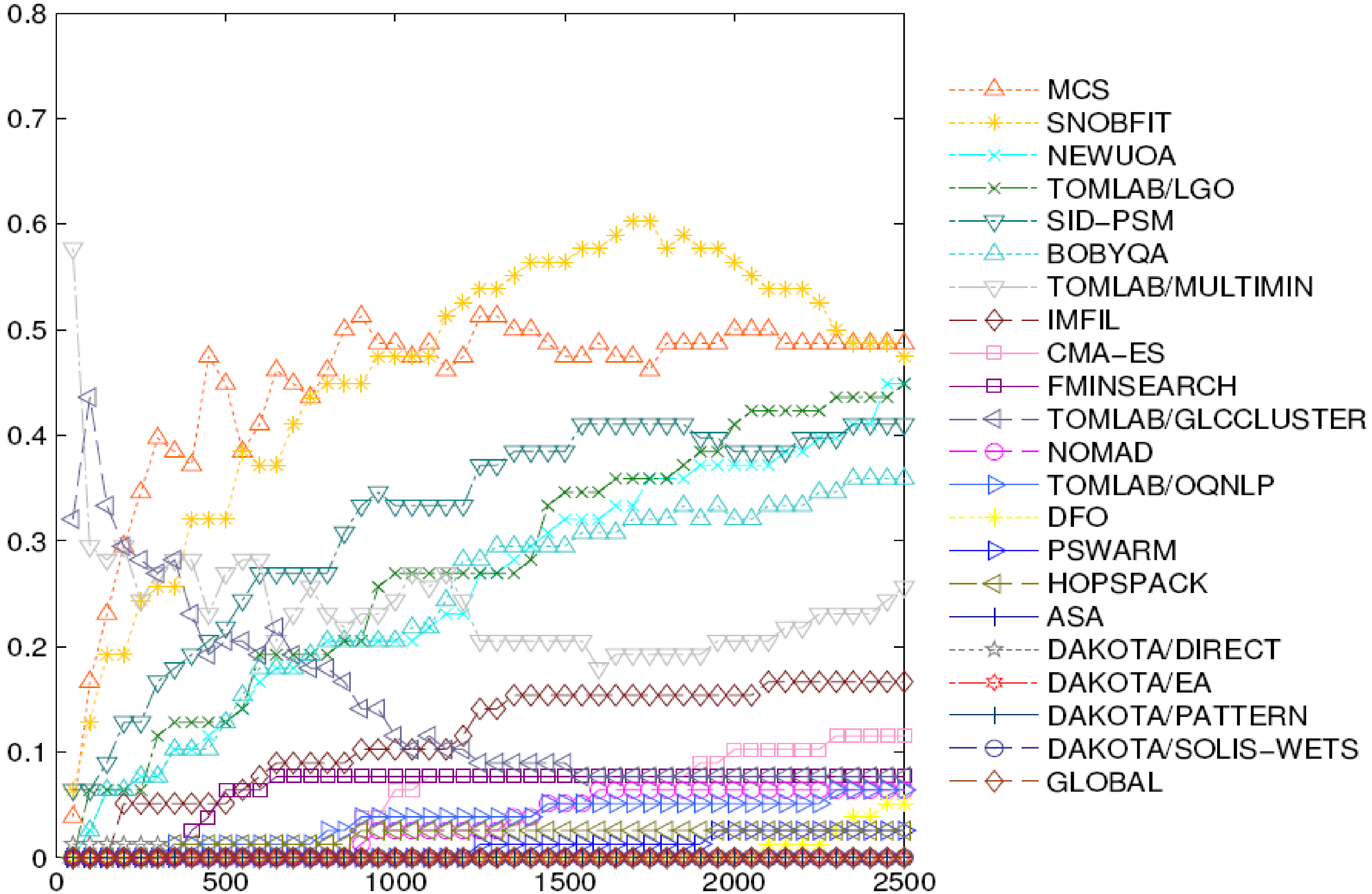
FRACTION OF PROBLEMS SOLVED: NONCONVEX SMOOTH



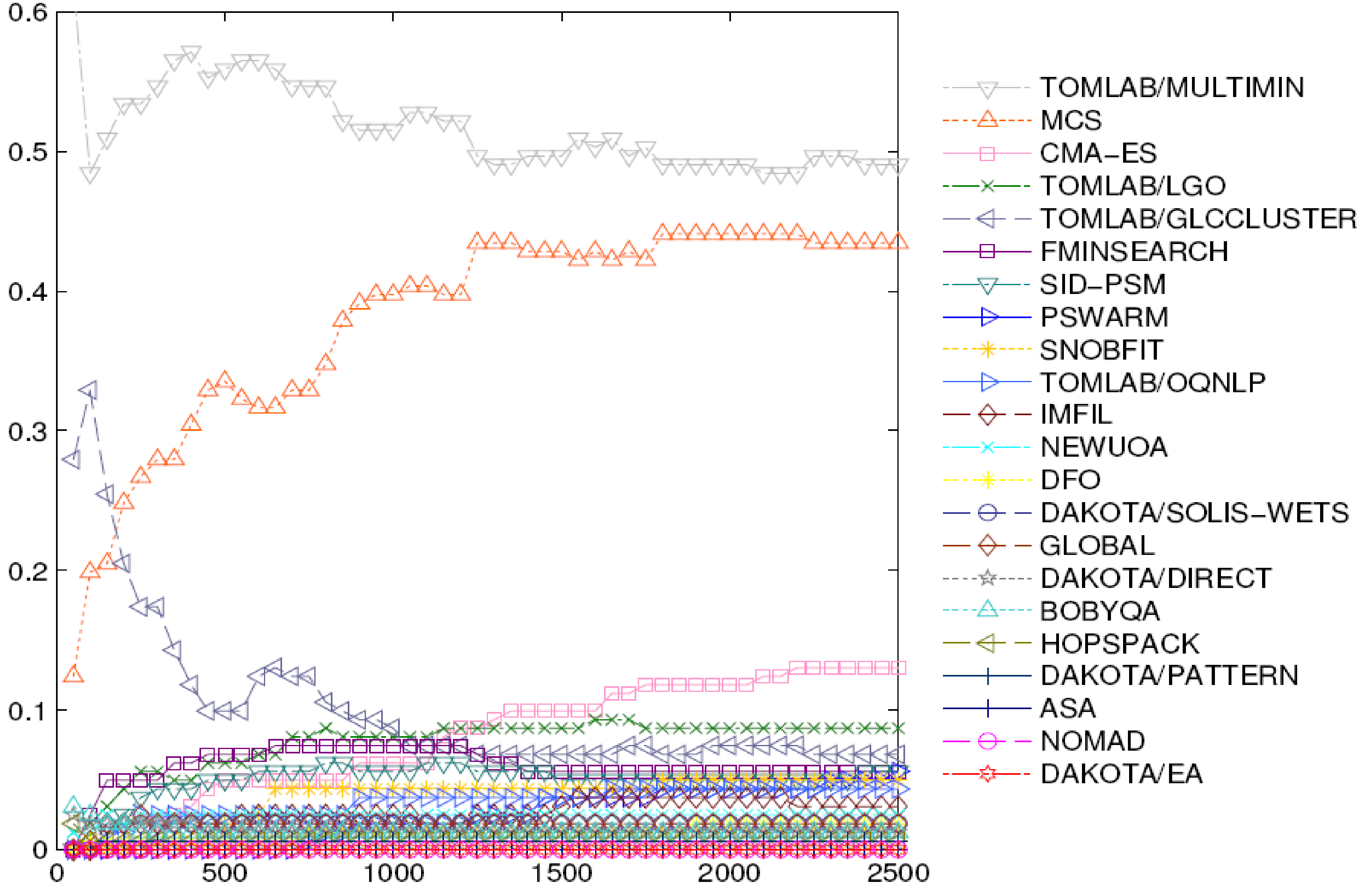
FRACTION OF PROBLEMS SOLVED: NONCONVEX NONSMOOTH



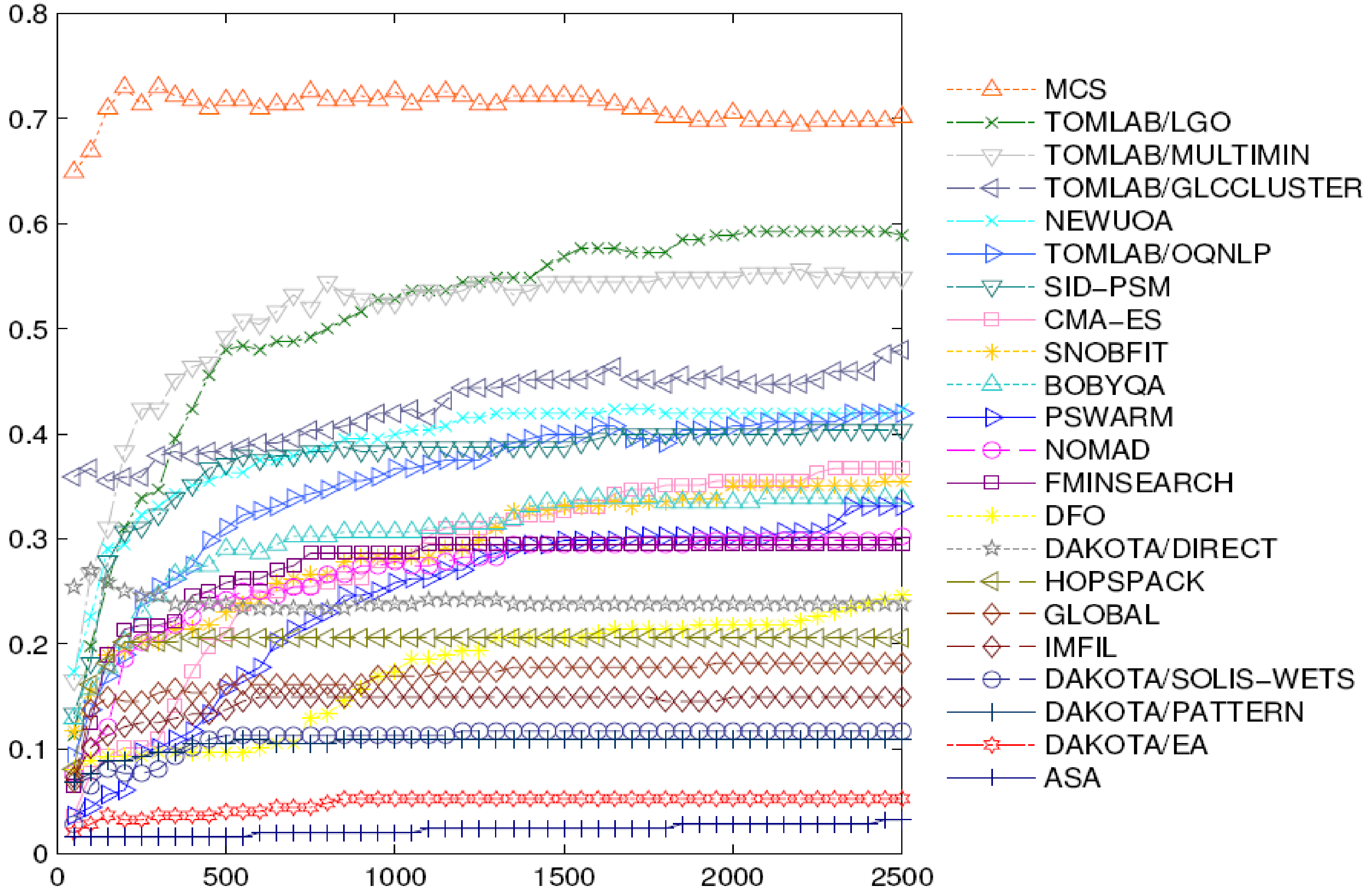
FRACTION OF PROBLEMS SOLVER WAS BEST: CONVEX SMOOTH



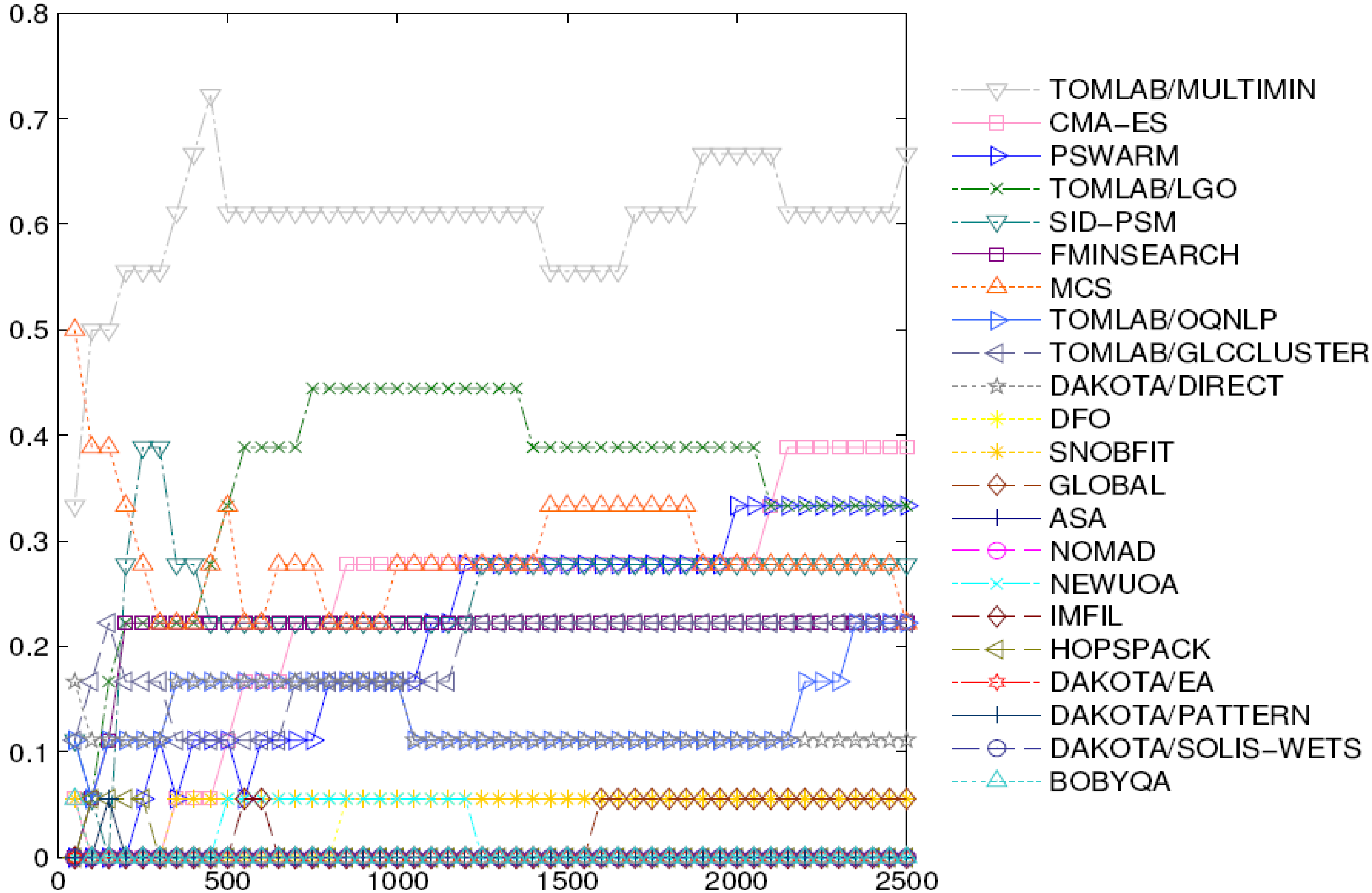
FRACTION OF PROBLEMS SOLVER WAS BEST: CONVEX NONSMOOTH



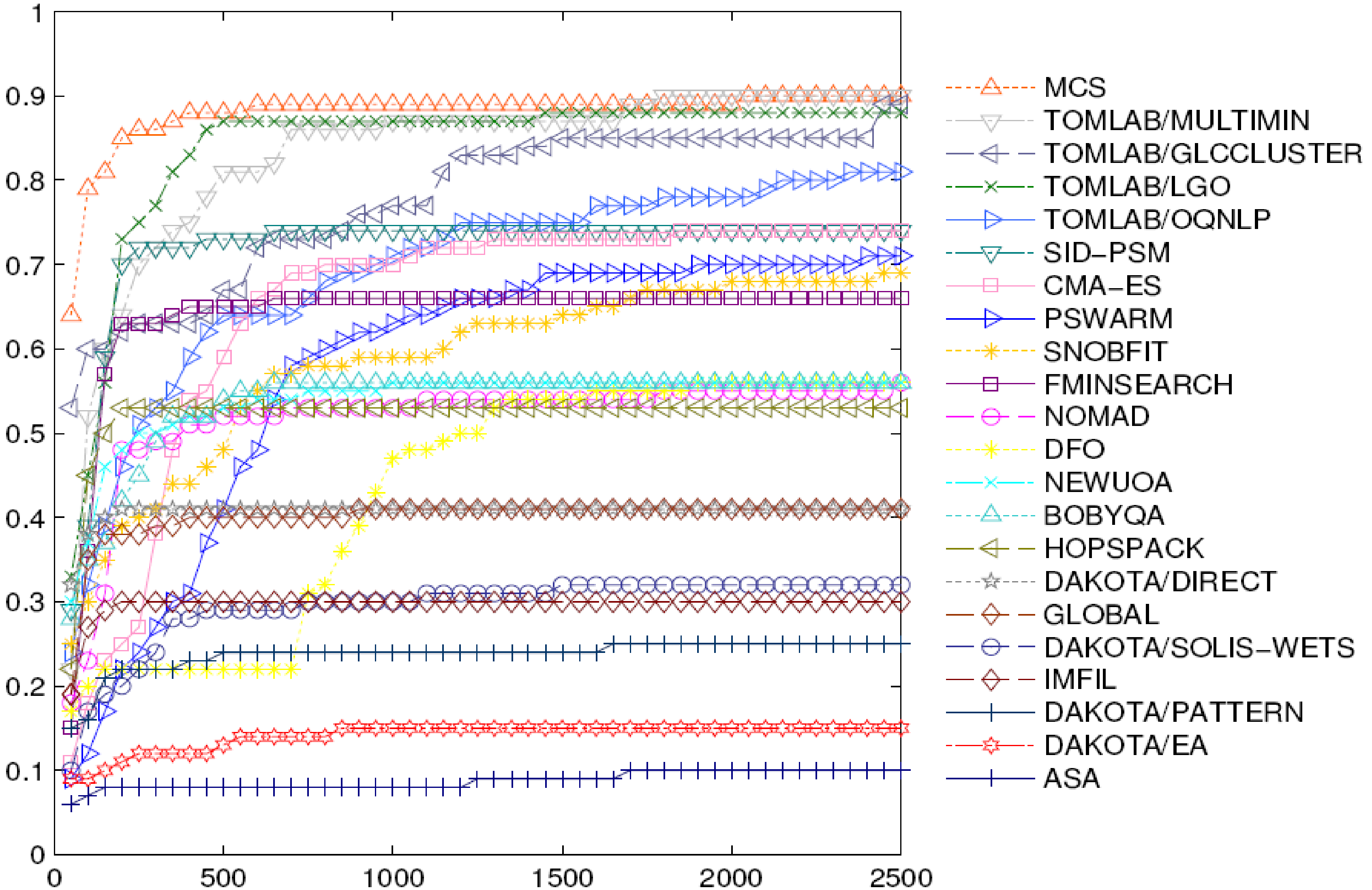
FRACTION OF PROBLEMS SOLVER WAS BEST: NONCONVEX SMOOTH



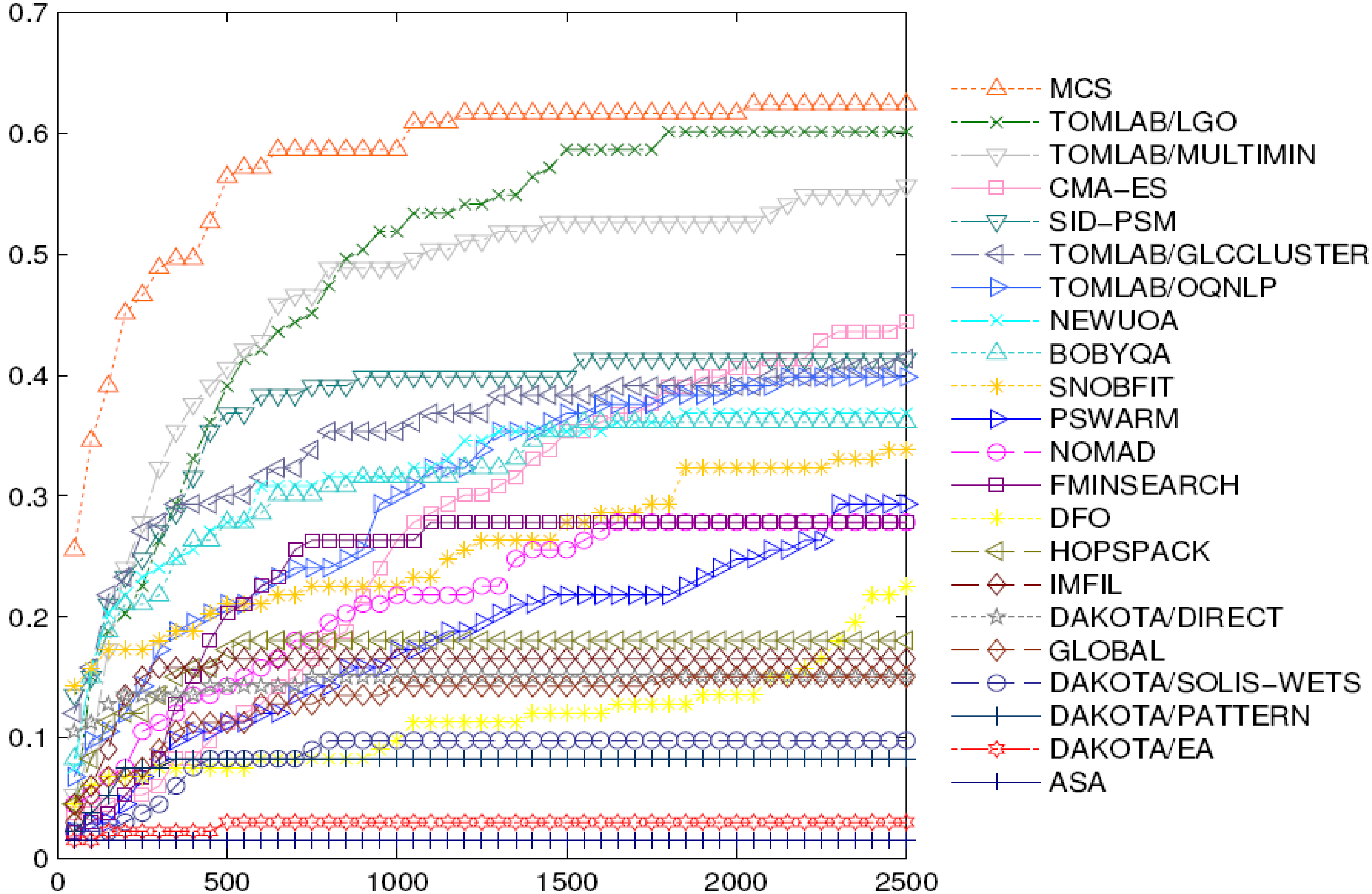
FRACTION OF PROBLEMS SOLVER WAS BEST: NONCONVEX NONSMOOTH



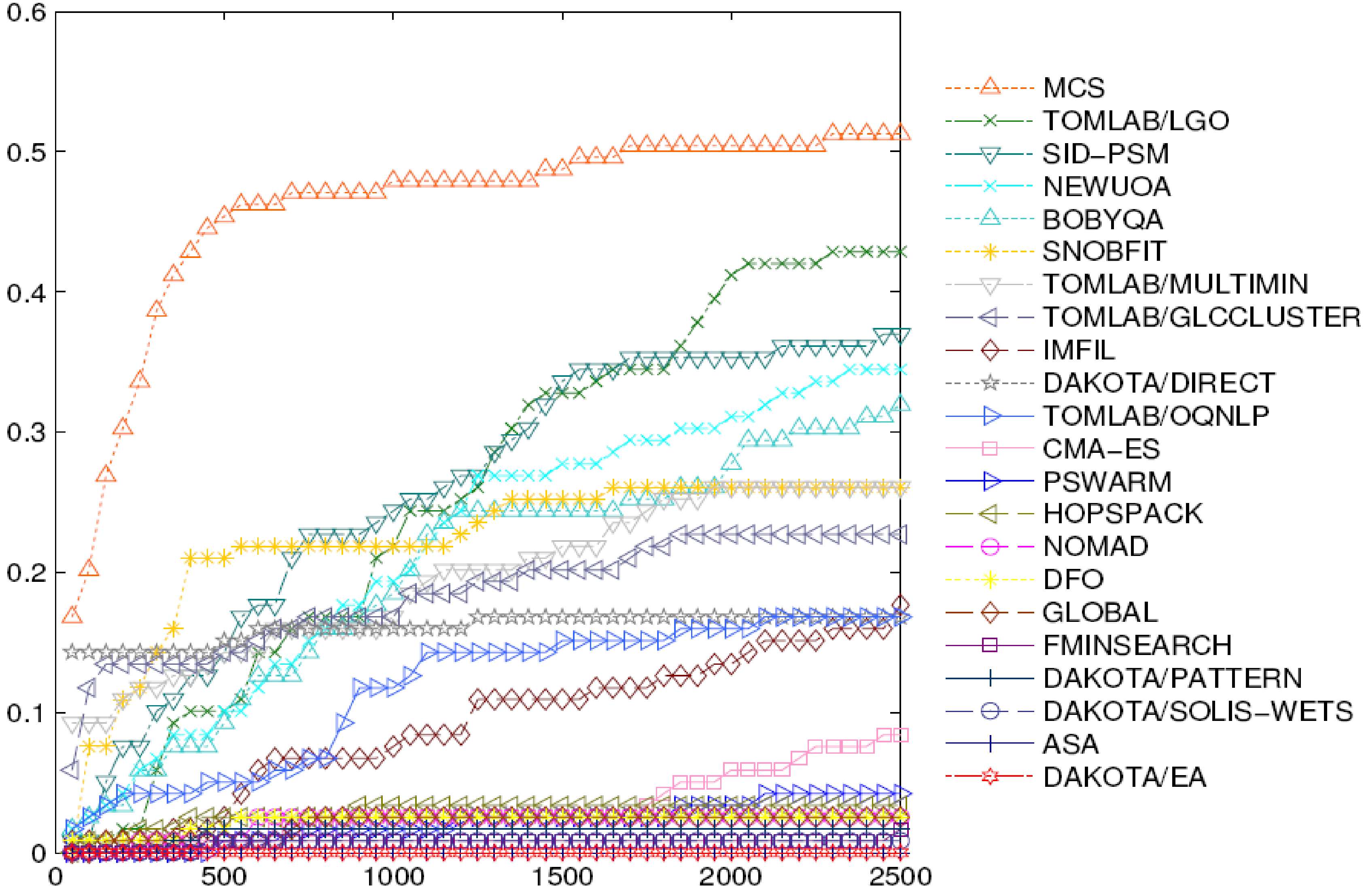
FRACTION OF PROBLEMS SOLVED: 1 TO 2 VARIABLES



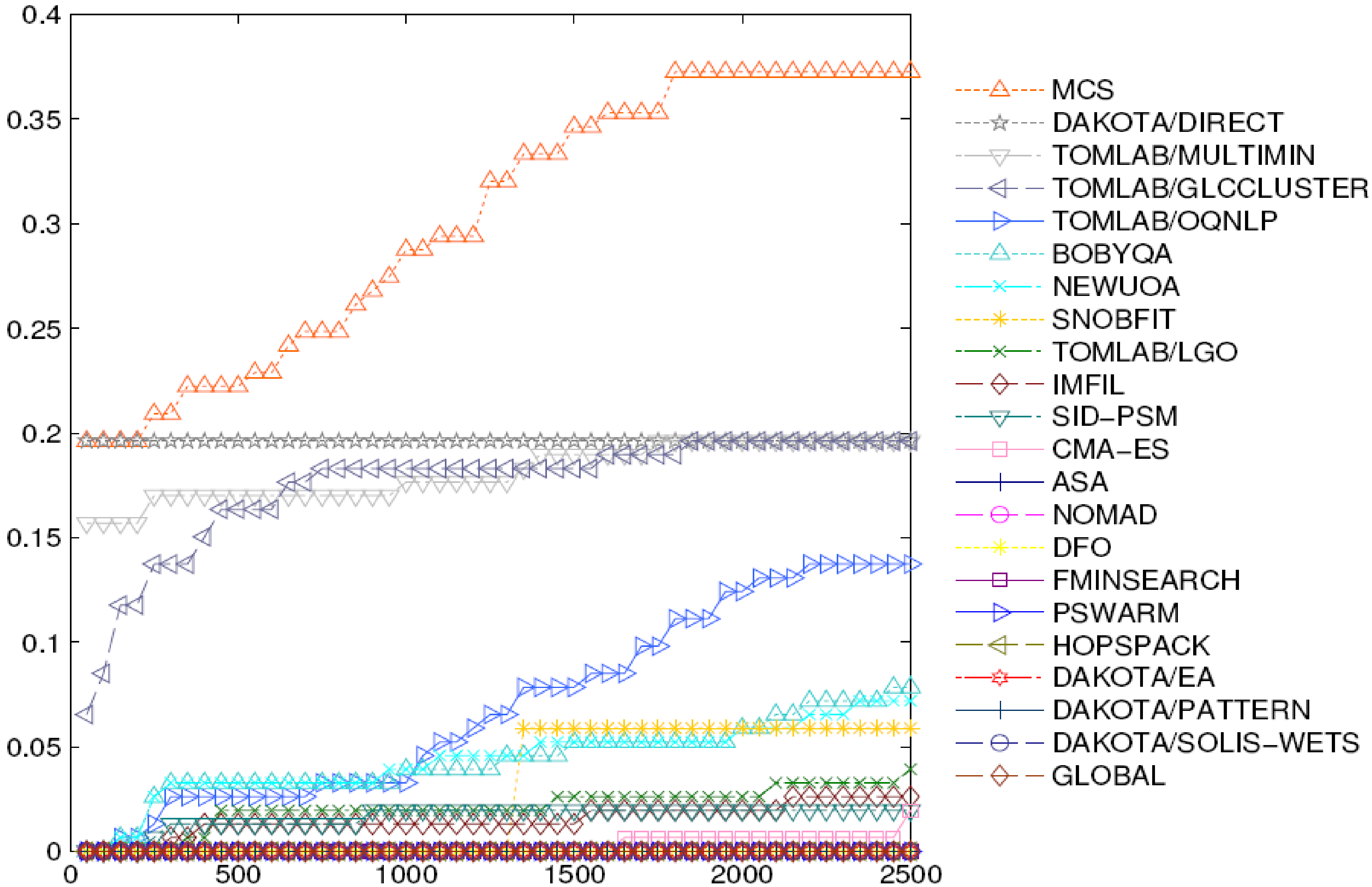
FRACTION OF PROBLEMS SOLVED: 3 TO 9 VARIABLES



FRACTION OF PROBLEMS SOLVED: 10 TO 30 VARIABLES



FRACTION OF PROBLEMS SOLVED: 31 TO 300 VARIABLES



STARTING POINT IMPROVEMENT

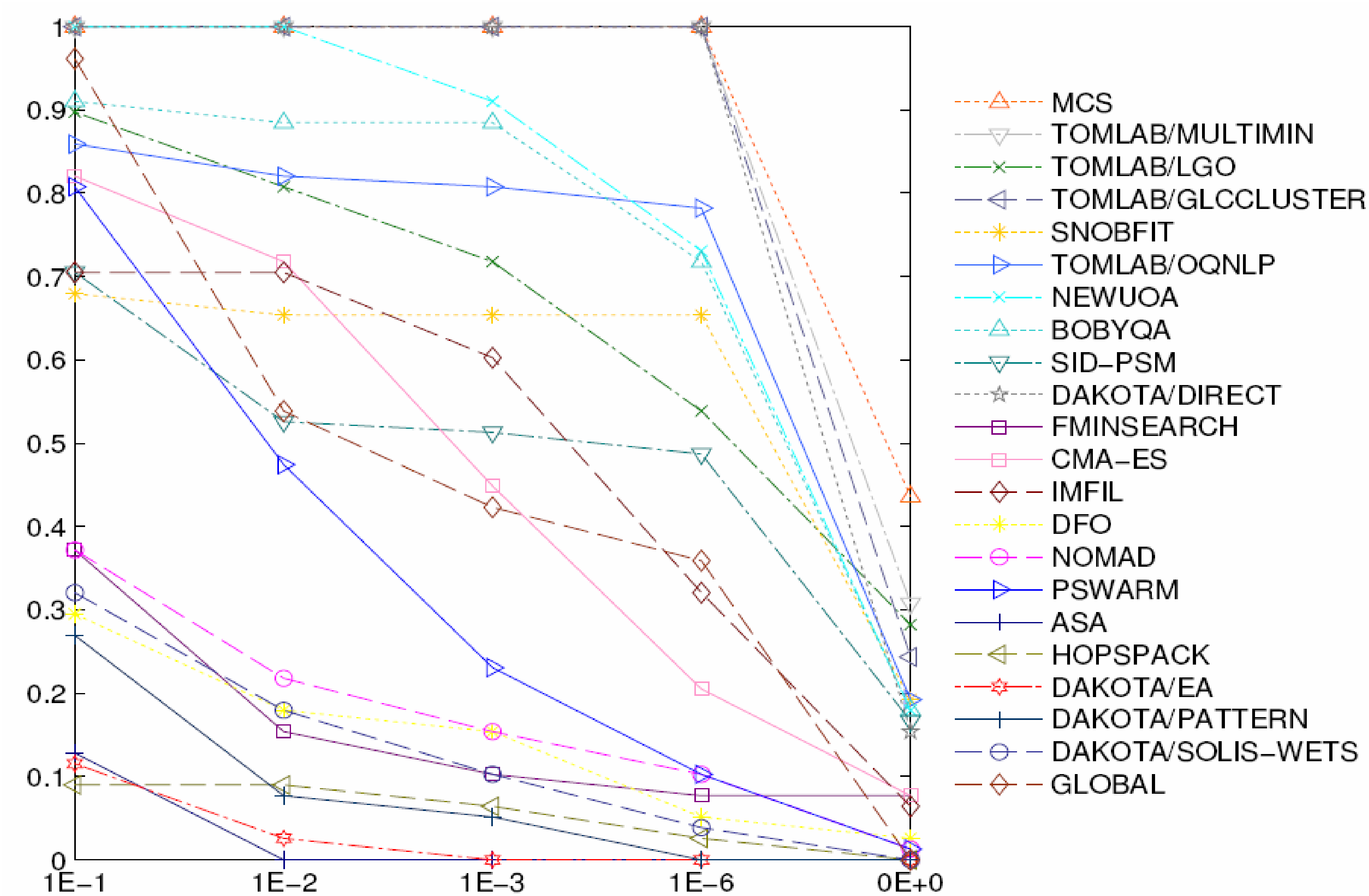
- **For a given τ between 0 and 1, and a given starting point x_0, a solver improves the starting point if**

$$f(x_0) - f_{\text{solver}} \geq (1-\tau)(f(x_0) - f_L)$$

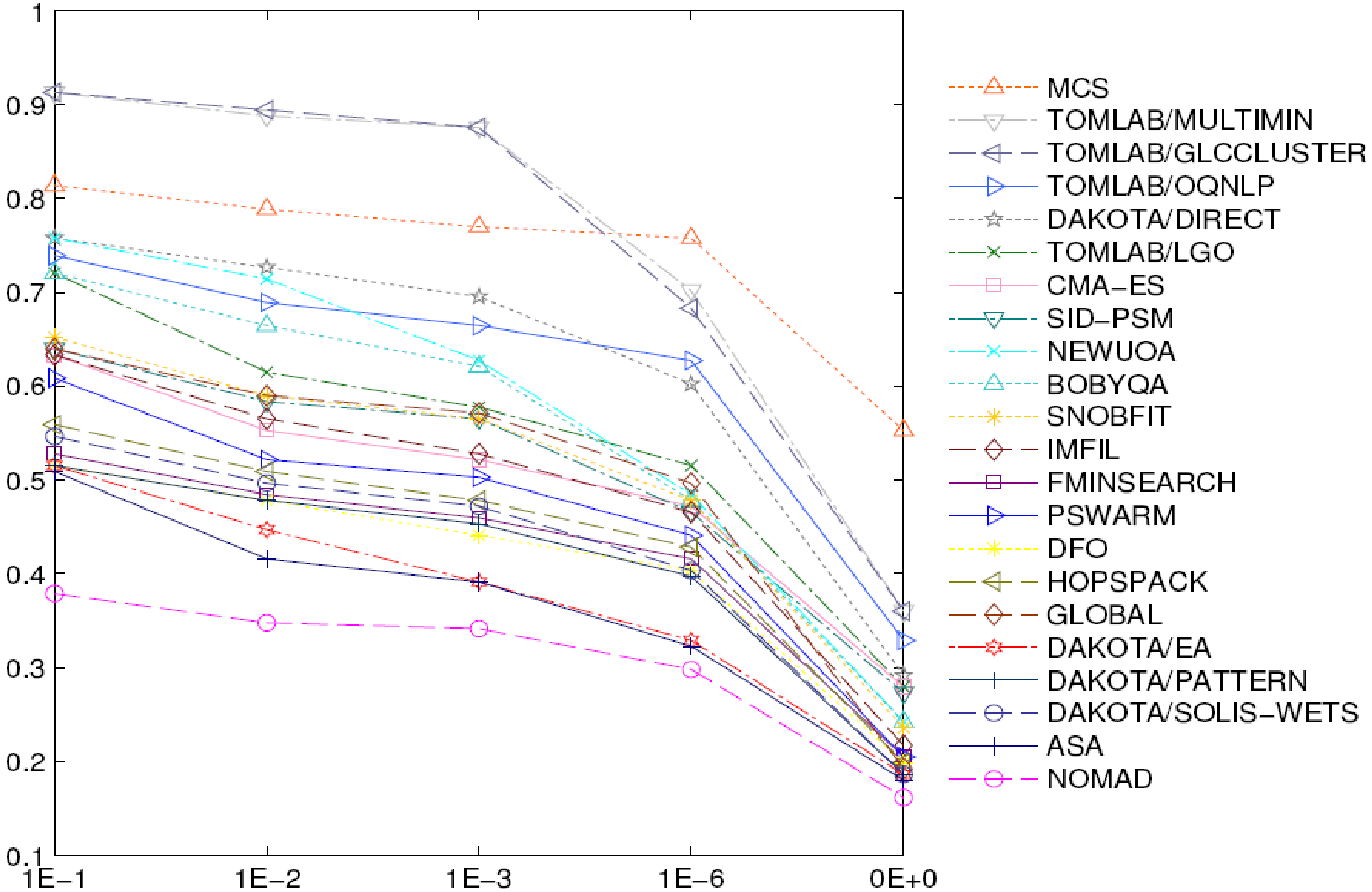
where f_L is the best possible solution for the problem

- **Problem considered solved if one or more runs satisfied this requirement**

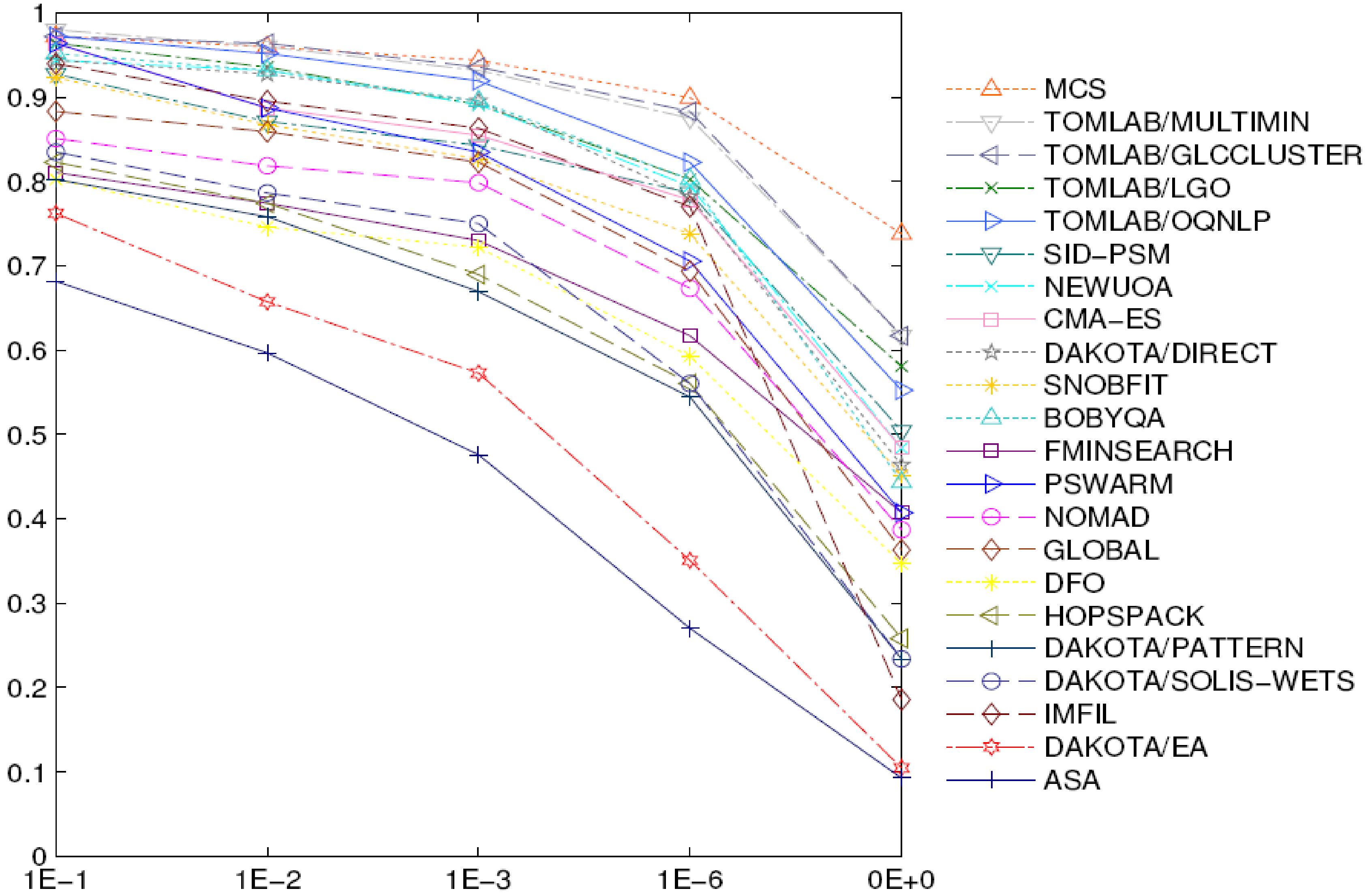
FRACTION OF PROBLEMS IMPROVED: CONVEX SMOOTH



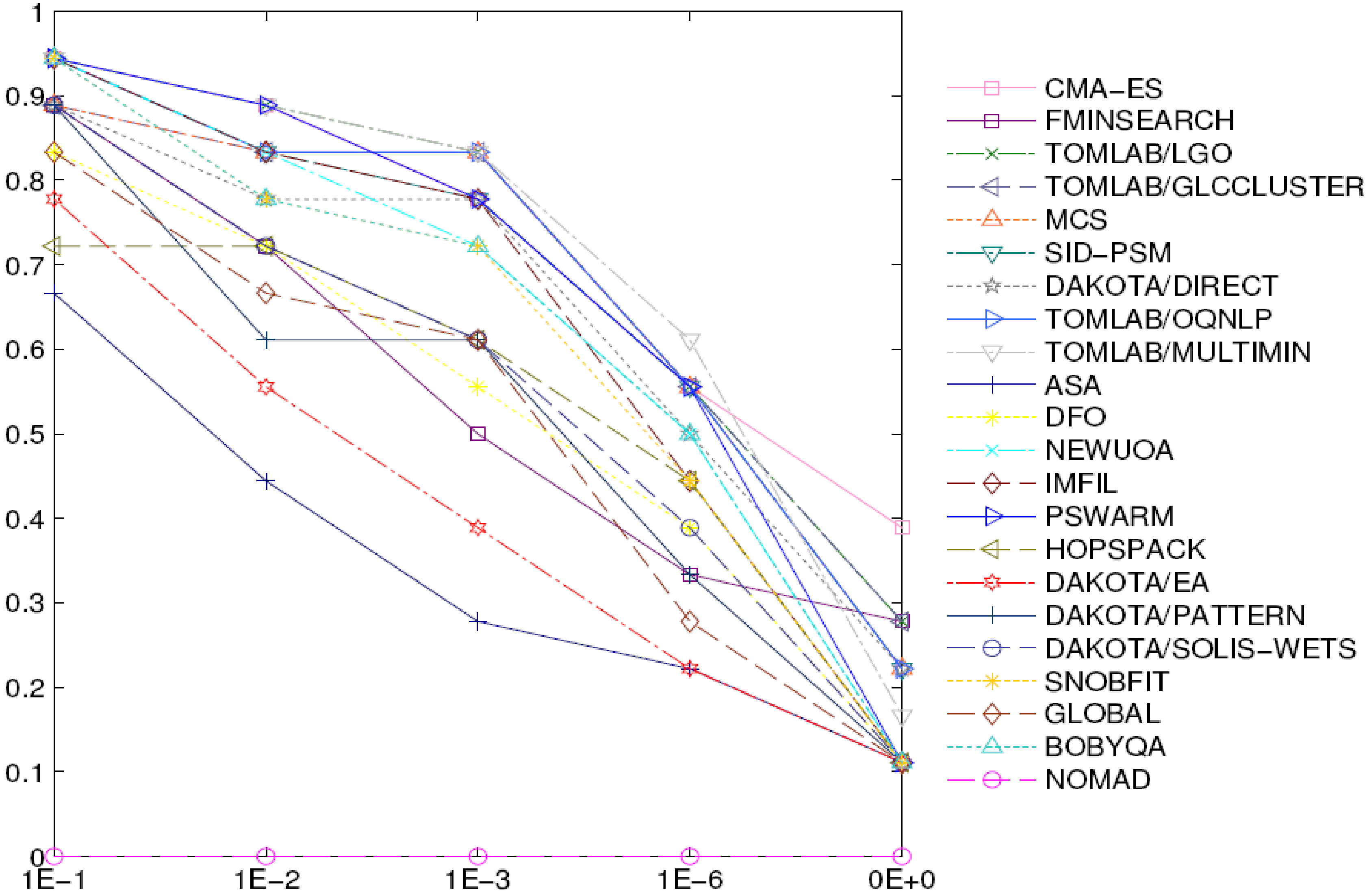
FRACTION OF PROBLEMS IMPROVED: CONVEX NONSMOOTH



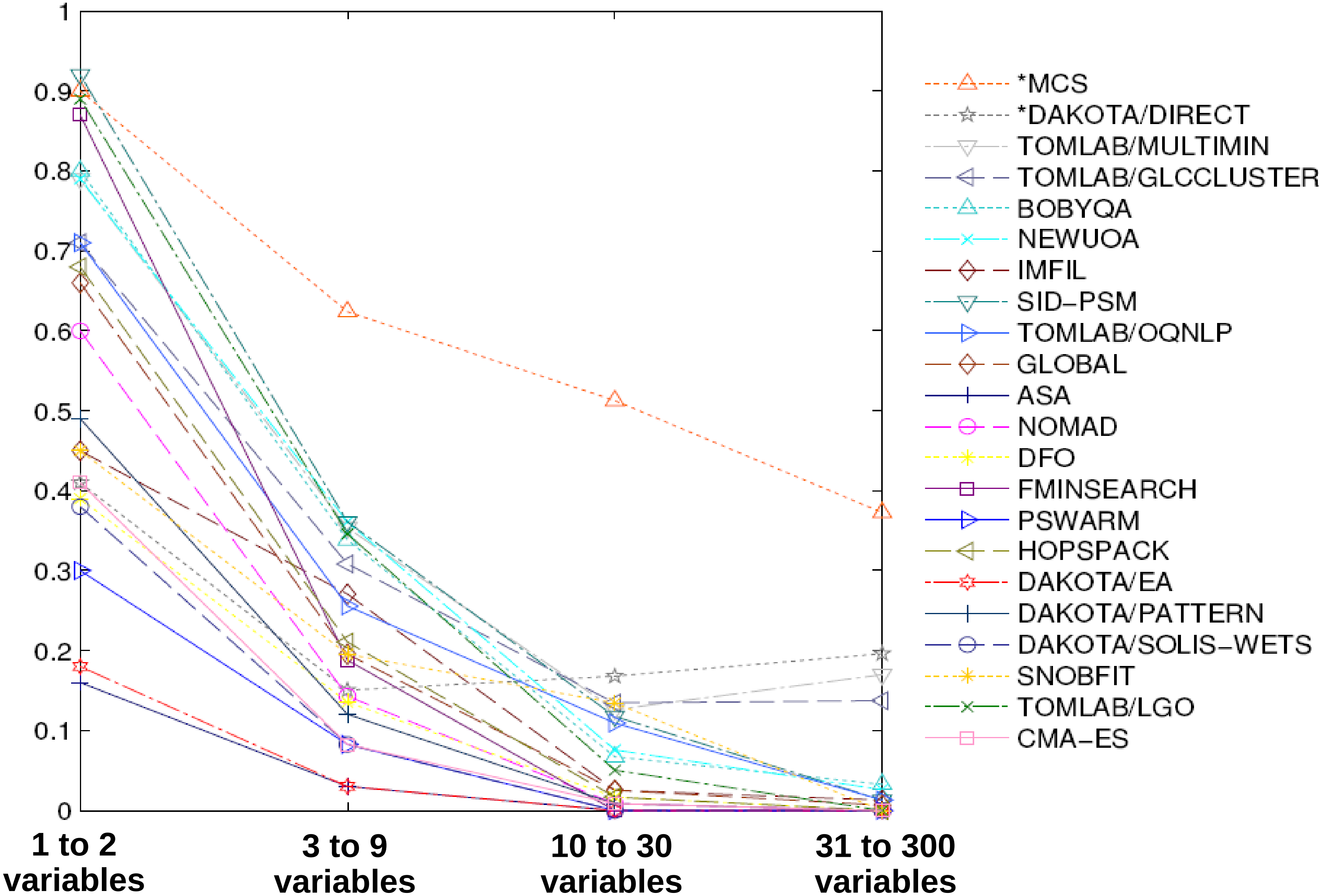
FRACTION OF PROBLEMS IMPROVED: NONCONVEX SMOOTH



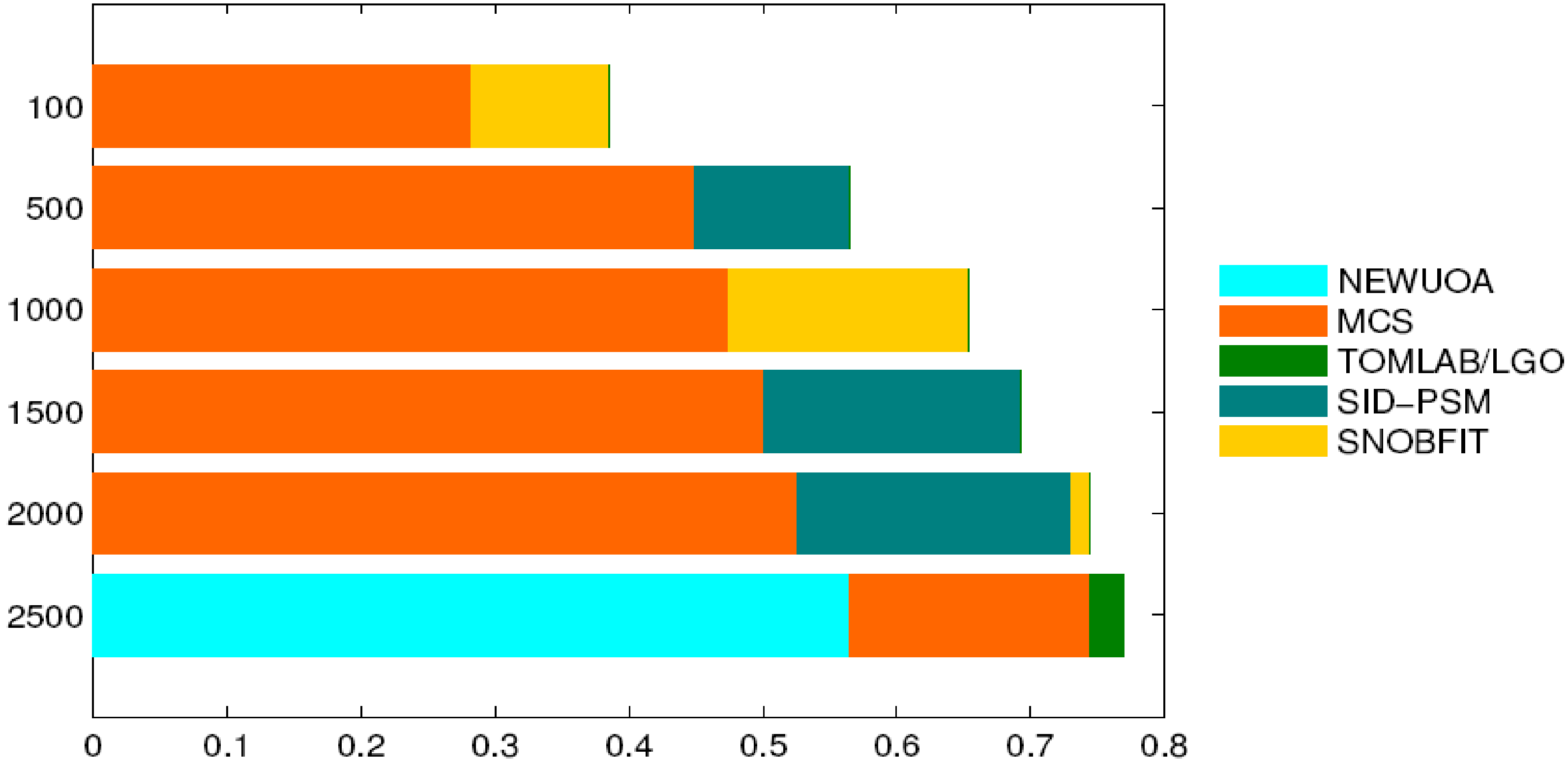
FRACTION OF PROBLEMS IMPROVED: NONCONVEX NONSMOOTH



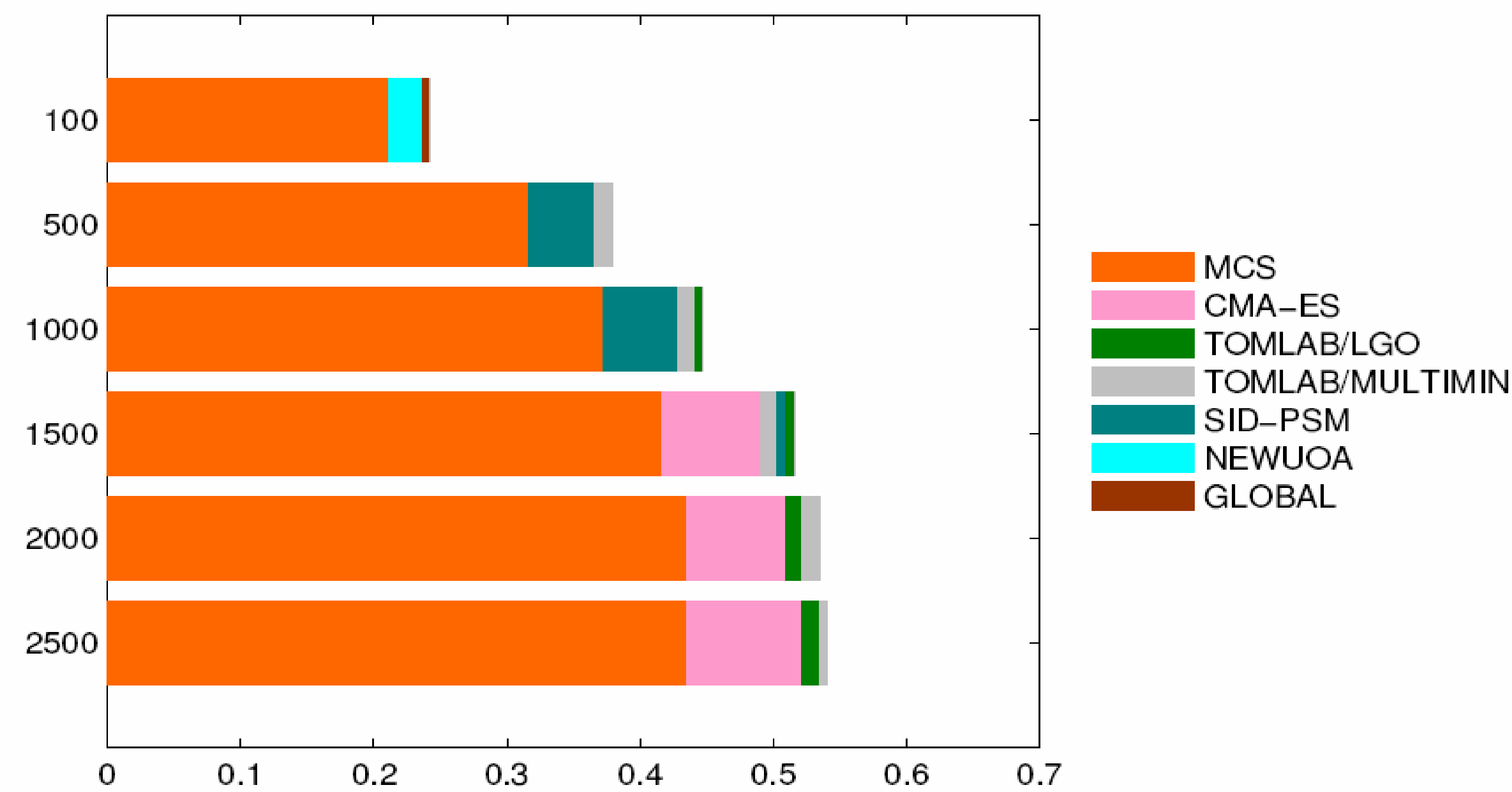
FRACTION OF PROBLEMS SOLVED: MULTISTART STRATEGY



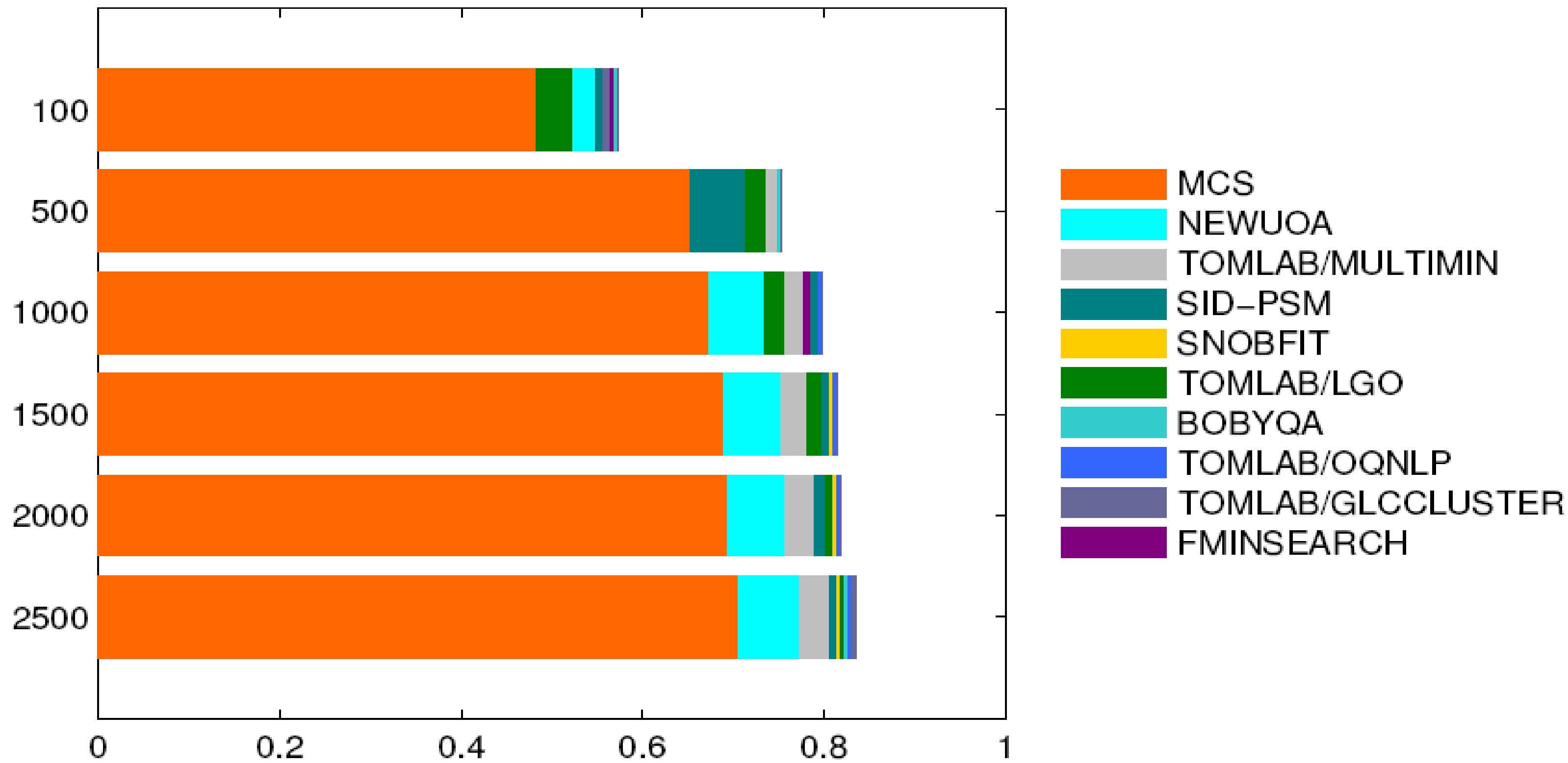
MINIMUM SET OF SOLVERS CONVEX SMOOTH PROBLEMS



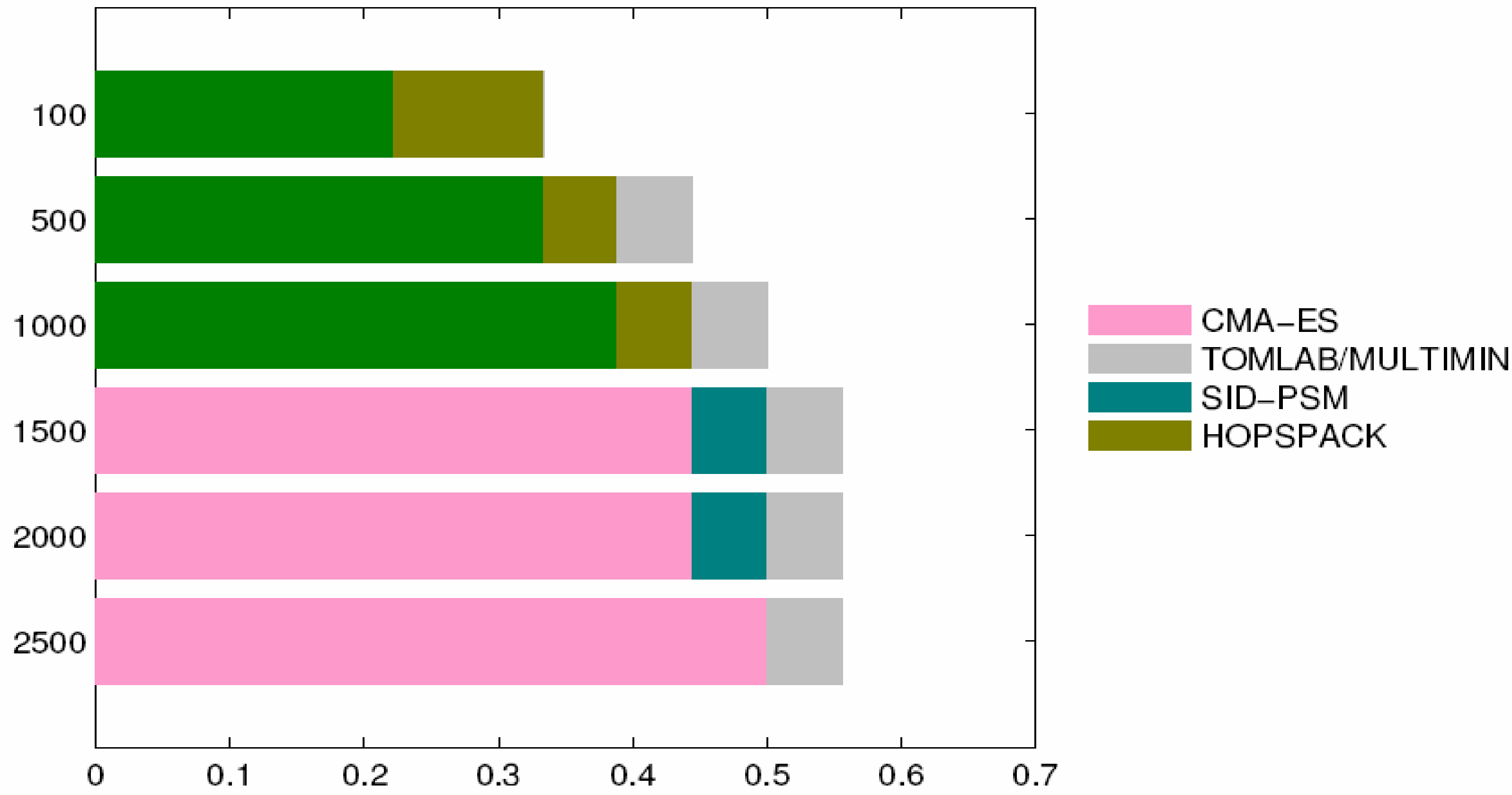
MINIMUM SET OF SOLVERS
CONVEX NONSMOOTH PROBLEMS



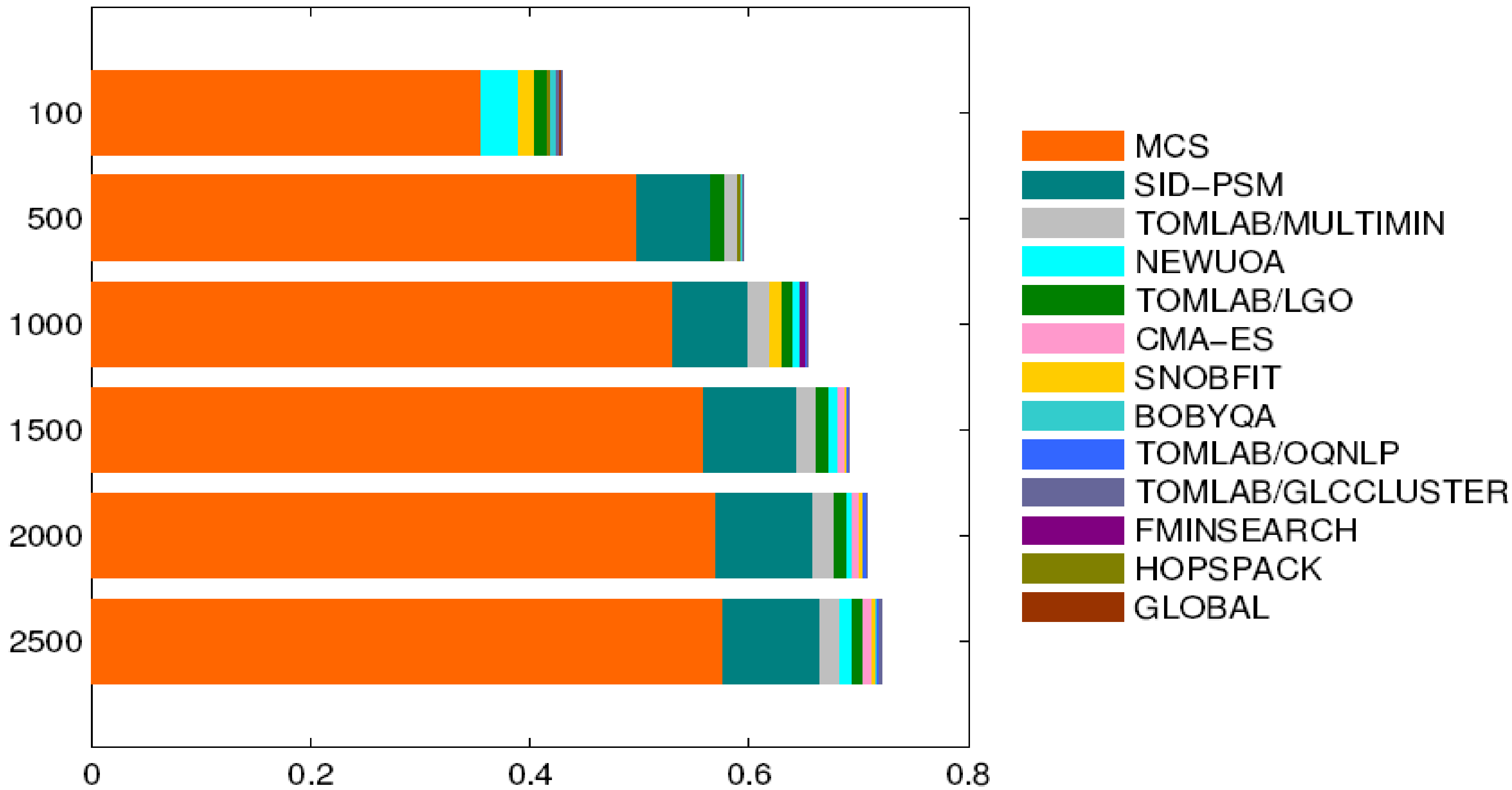
MINIMUM SET OF SOLVERS NONCONVEX SMOOTH PROBLEMS



MINIMUM SET OF SOLVERS NONCONVEX NONSMOOTH PROBLEMS



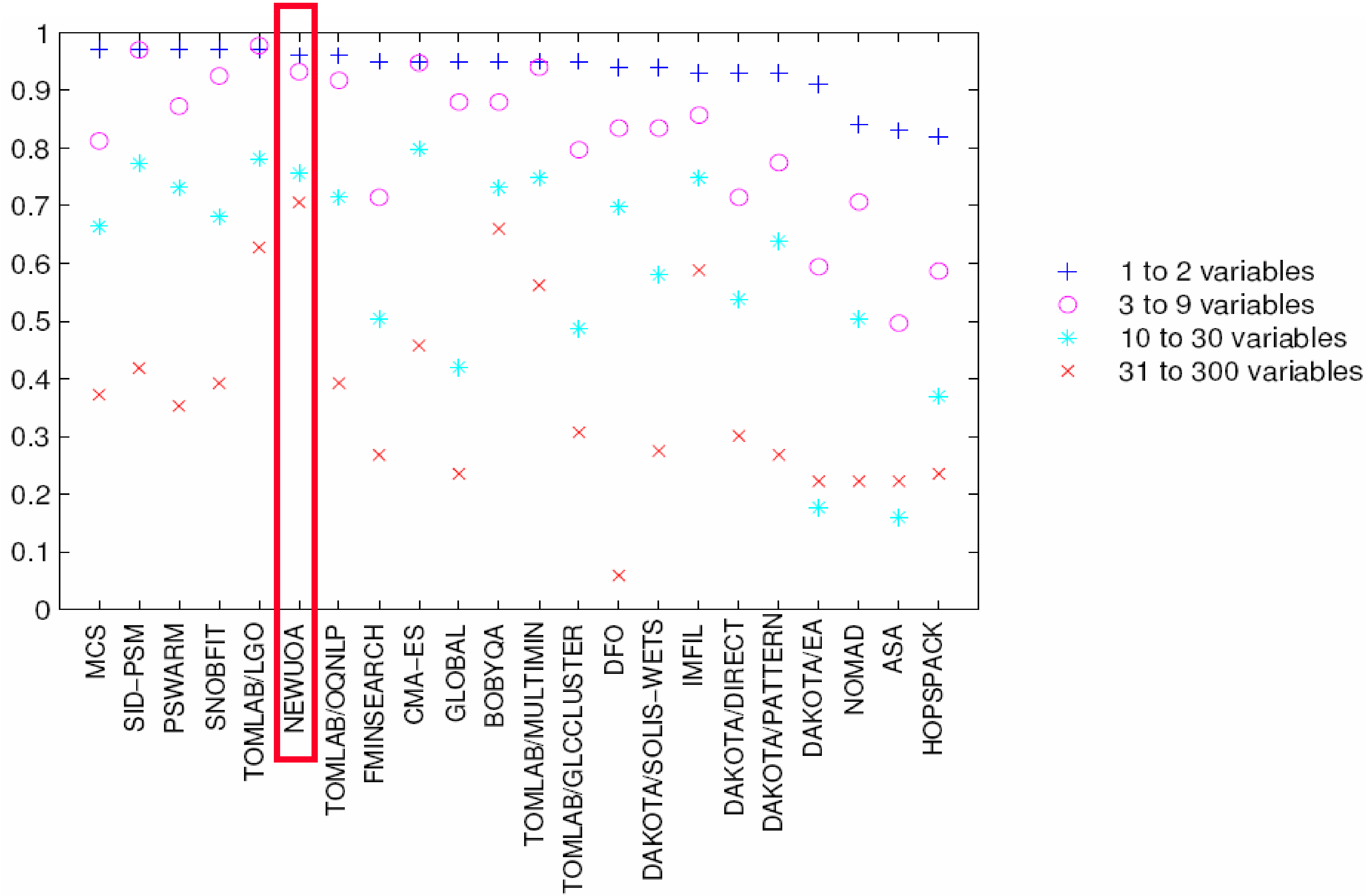
MINIMUM SET OF SOLVERS
ALL PROBLEMS



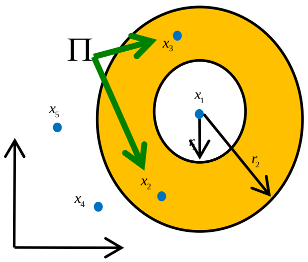
REFINEMENT ABILITY

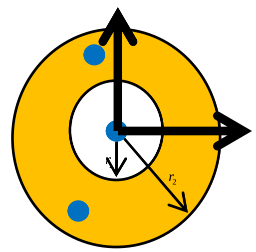
- **Solvers were started from an starting point close to a global minimum of the problem**
- **A range of 0.2 for each variable was used (unless problem bounds were tighter)**

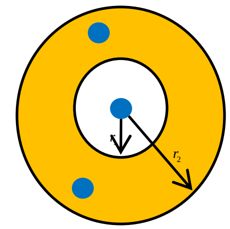
FRACTION OF LOCAL PROBLEMS SOLVED: ALL PROBLEMS



MODEL-AND-SEARCH
LOCAL ALGORITHM







Collect points around current iterate →

Scale and shift origin to current iterate →

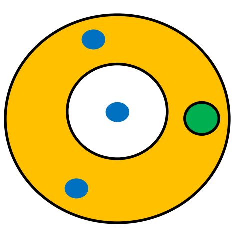
Add points: n linearly independent points

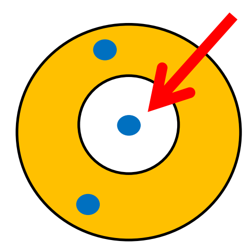
↓

Evaluate points in specific order →

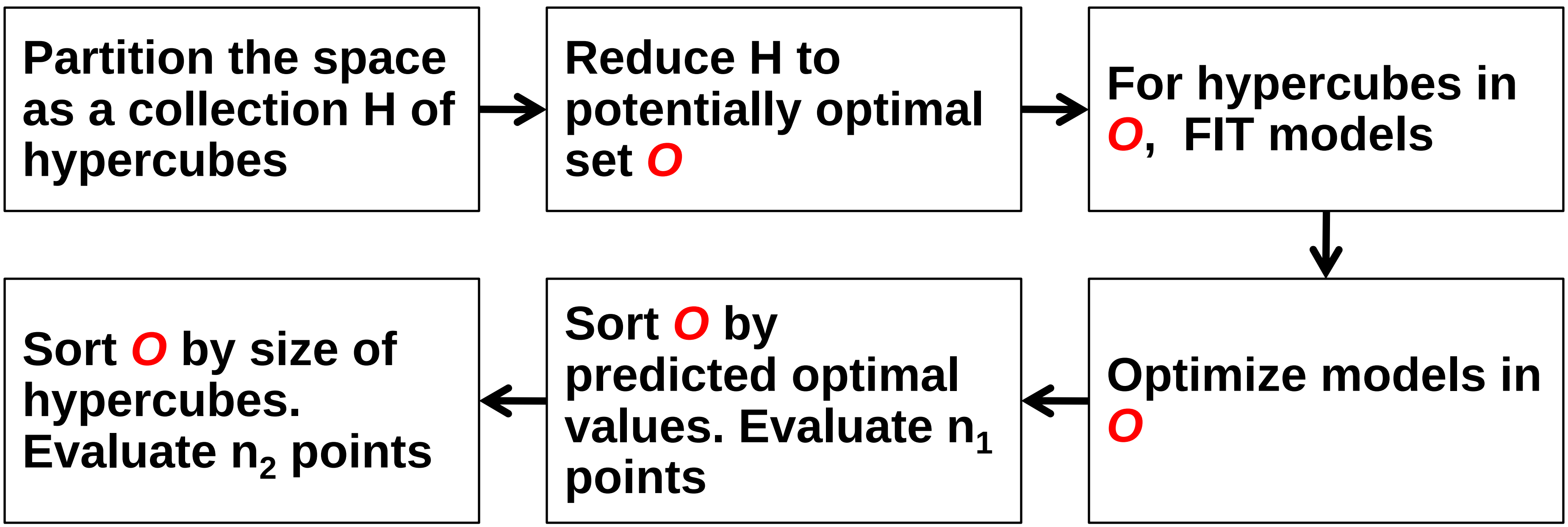
Build and optimize interpolating model →

Check for positive basis. Add points if necessary

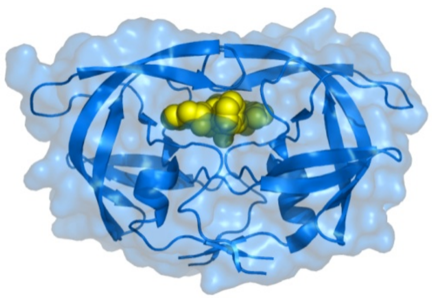




BRANCH-AND-MODEL GLOBAL ALGORITHM



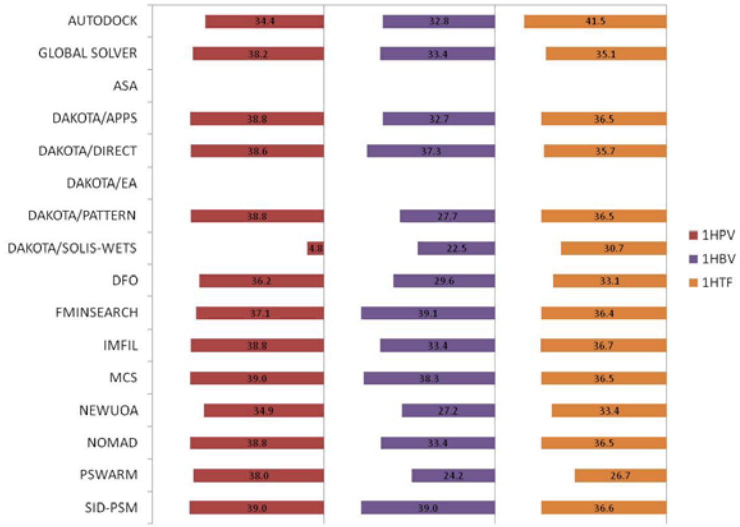
PROTEIN-LIGAND DOCKING



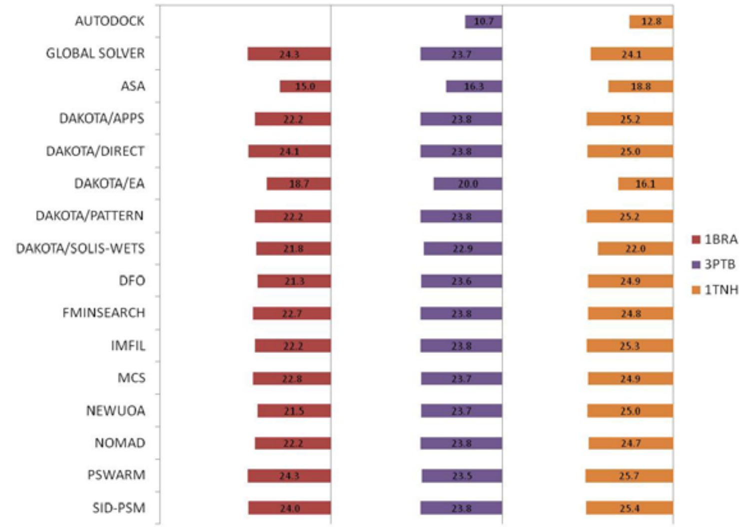
- **Identify binding site and pose**
- **Conformation must minimize binding free energy**
- **Docking packages**
  - **AutoDock, Gold, FlexX …**
  - **Most rely on genetic and other stochastic search algorithms**

BINDING ENERGIES

ASPARTIC

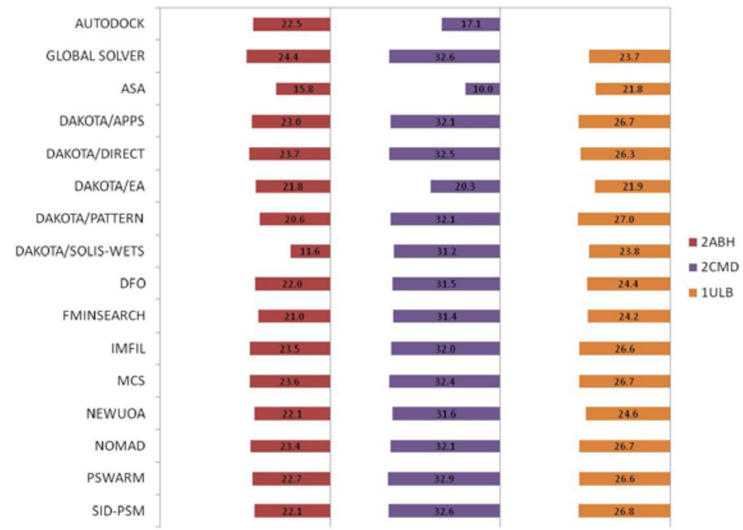


SERINE



METALLOPROTEASES

MISCELLANEOUS



B&M outperformed AutoDock in 11 out of 12 cases,
and found the best solution amongst all solvers for 3 complexes

CONCLUSIONS

- MCS, LGO, and NEWOA/BOBYQA stand out
- Stochastic solvers do not perform as well as deterministic ones
  - CMA-ES and PSWARM are occasionally competitive
- Many opportunities
  - New algorithms needed
  - Applications abound
- Readings
  - Rios and Sahinidis (2010)
  - Conn, Scheinberg and Vicente (2009)