Global Optimization and the GAMS Branch-and Cut Facility

Hernan Abeledo, Michael Bussieck, Leon Lasdon, Alex Meeraus, Hua Ni

George Washington University
GAMS Development Corporation
University of Texas
GAMS Development Corporation
George Washington University

CORS/INFORMS Joint International Meeting Banff
May 16-19 2004

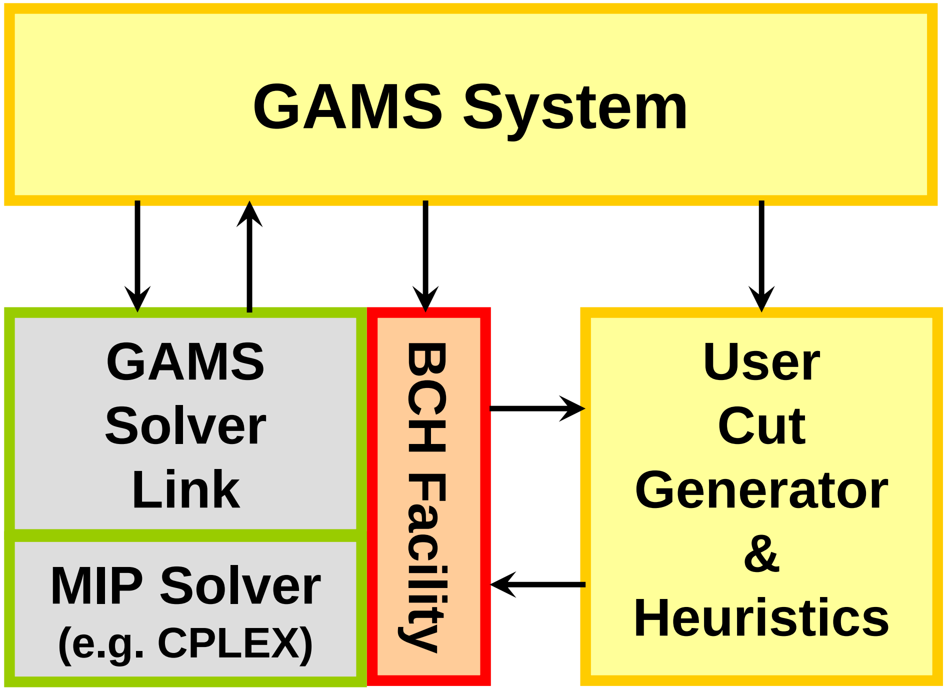
Branch-and-Cut (B&C)

- MIP problems benefit from *user supplied* cutting planes and integer solutions in a B&C algorithm
- Implementation facilities:
  - MIP solver callback functions (CPLEX, XPRESS, …)
  - B&C framework (ABACUS, COIN BCP, …)
- Required Knowledge for Implementation
  - IT knowledge (C/C++/JAVA, Solver APIs)
  - Mathematical programming knowledge
  - Application specific knowledge
- Supply GAMS users with an easy access to B&C: ***GAMS Branch-and-Cut & Heuristic (BCH) Facility***

BCH Unique Facility

- Cut Generator and Heuristic
  – Represented in terms of original GAMS problem formulation
  – Independent of the specific MIP solver
  – Use any other model type and solver available in GAMS

- Design Principle:



Example: Multi-Knapsack

http://www.gams.com/modlib/libhtml/bchmknap.htm

```
Binary variables x(j); Positive variables slack(i);
Equations mk(i), defobj; Variable  z;

defobj.. z =e= sum(j, value(j)*x(j));
mk(i)..   sum(j, a(i,j)*x(j)) =l= size(i);

model m /all/;  solve m max z using mip;
```

The original model formulation

Separation Problem for Cover Cuts:

z.l<1

Cover Cuts c(j)=y.l(j):

sum(c(j),x(j)) =l= card(j)-1;

```
Binary variable y(j) membership in the cover;
Equations defcover, defobj; Variable z;

defobj..      z =e= sum(j, (1-x.l(j))*y(j));
defcover..  sum(j, ai(j)*y(j)) =g= size_i+1;

model cover /all/; solve cover min z using mip;
```

Cover Cut Separation Model

- Master model (bchmknap.gms):

```
execute_unload 'multiknapdata', j, I, a, size          // Export of data
```

- Activate Cplex/BCH facility (cplex.opt):

```
usercutcall bchcover.gms
```

- Separation model (bchcover.gms):

```
$gdxin multiknapdata                        // Get data from original model
$load j i a size
$gdxin bchout                               // Get fractional solution from solver
$load x
$include cutmodel                           // model definition and solution
* Define additional cut for original problem:
If (z.l<1, numcuts = 1;
   x_c('1',j) = y.l(j);                     // cut matrix
   rhs_c('1') = sum(j, y.l(j)) - 1;
   sense_c('1') = 1);                       // 1 =l=, 2 =e=, 3 =g=
```

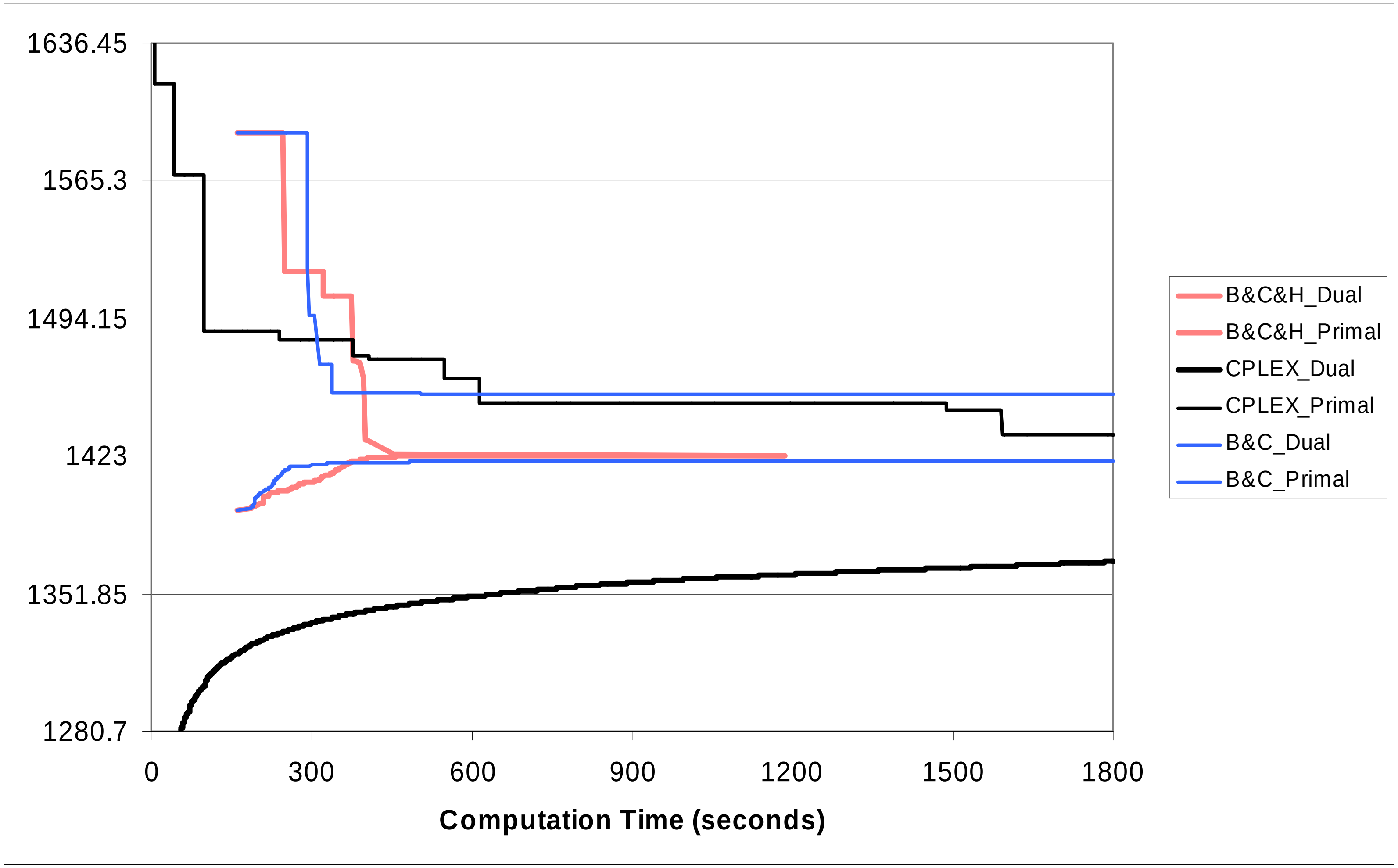
CPLEX with BCH Log

```
        Nodes                                               Cuts/
   Node  Left     Objective  IInf  Best Integer     Best Node    ItCnt     Gap
      0     0     4134.0741     2                   4134.0741        3
*** Calling cut generator. Added     1 cut
                  4090.6542     2                    User:  1        4
*** Calling cut generator. Added     1 cut
                  4001.4270     4                    User:  1        5
*** Calling cut generator. Added     1 cut
                  3950.4202     4                    User:  1        6
*** Calling cut generator. Added     1 cut
                  3871.4286     2                    User:  1        7
*** Calling cut generator. Added     1 cut
                  3841.6244     6                    User:  1        9
*** Calling cut generator. No cuts found
*** Calling cut generator. No cuts found
      1     1     3800.0000     3                   3800.0000       10
*** Calling cut generator. Added     1 cut
*** Calling cut generator. No cuts found
                                                     User:  1
*** Calling cut generator. No cuts found
*     2     1                   0     3800.0000     3800.0000       12    0.00%
*** Calling cut generator. No cuts found
```

Another Example

- Real Example: Oil Pipeline Design Problem
  - J. Brimberg, P. Hansen, K.-W. Lih, N. Mladenovic, M. Breton 2003. An Oil Pipeline Design Problem. Operations Research, Vol 51, No. 2 228-239
  - http://www.gams.com/modlib/libhtml/bchoil.htm
- Performance Improvements
  - Cplex/BCH: 20 minutes
  - Regular Cplex: 450 minutes
- Overhead of BCH
  - Time spent within the callback functions minus MIP computation on cuts and heuristics: 20% ~ 25%

Convergence – Pipeline Design



OK, but what has all this to do with Global Optimization?

- *Use any other model type and solver available in GAMS*
- Solve difficult MIP problems using GO for cut separation

Difficult Network Problem

- Single-commodity, uncapacitated, fixed-charge network flow problem

$$\min \sum_{(i,j)\in A} (f_{ij} y_{ij} + c_{ij} x_{ij}), s.t. \sum_{(j,i)\in\delta^-(i)} x_{ji} - \sum_{(i,j)\in\delta^+(i)} x_{ij} = b_i, 0 \le x_{ij} \le M y_{ij}, y \in \{0,1\}$$

- Problem class includes Steiner Tree Problem
- F. Ortega, L. Wolsey A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. Networks 41 (2003), no. 3, 143—158
- 83 instances (including 24 not proven optimal)

Dicut Inequalities

- Dicut: $\sum_{(i,j)\in\delta^-(S)} y_{ij} \geq 1$ if S⊂V and `b(S)>0`.
- Separation problem:

$$\xi = \min \left\{ \sum_{(i,j)\in A} \bar{y}_{ij} z_j (1 - z_i) \ : \ \sum_{i\in V} b_i z_i > 0, \ z_i \in \{0,1\} \ \forall \ i \in V \right\}$$

- Non-convex quadratic binary program:
  - Ortega/Wolsey: Greedy algorithm
  - Let's solve this exactly (small, #variables |V|)

Global versus Local

- Berlin52 – from SteinLib
  - 52 nodes (1 source, n'=15 sinks), 1326 edges

	CPLEX (no BCH)	BCH + SBB (local)	BCH + BARON (global)
# cuts	0	139	610
# nodes	168449	1	1
Time	1000	1000	223
Gap	45%	39%	0%

GAMS

Heuristics

- Local Branching (Fischetti/Lodi) or
  Delta/Onion Method (Guignard/Spielberg)
- Improve a given binary solution *x** by restricting search space. Add inequality:

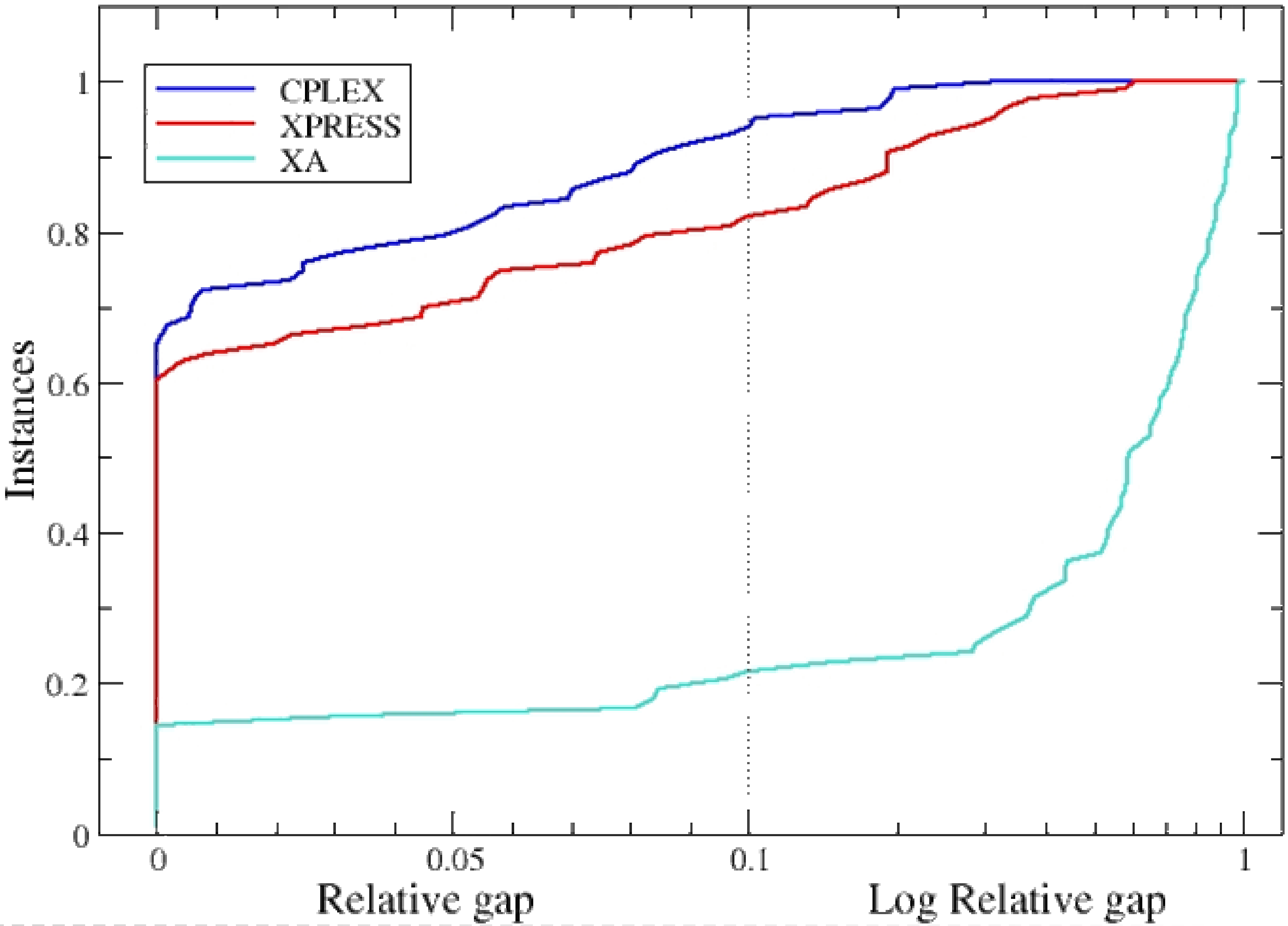
$$\sum_{j,x_j^*=1} x_j + \sum_{j,x_j^*=0} (1 - x_j) \le k$$

- Solve resulting problem as regular MIP
- Solve recursively/iteratively

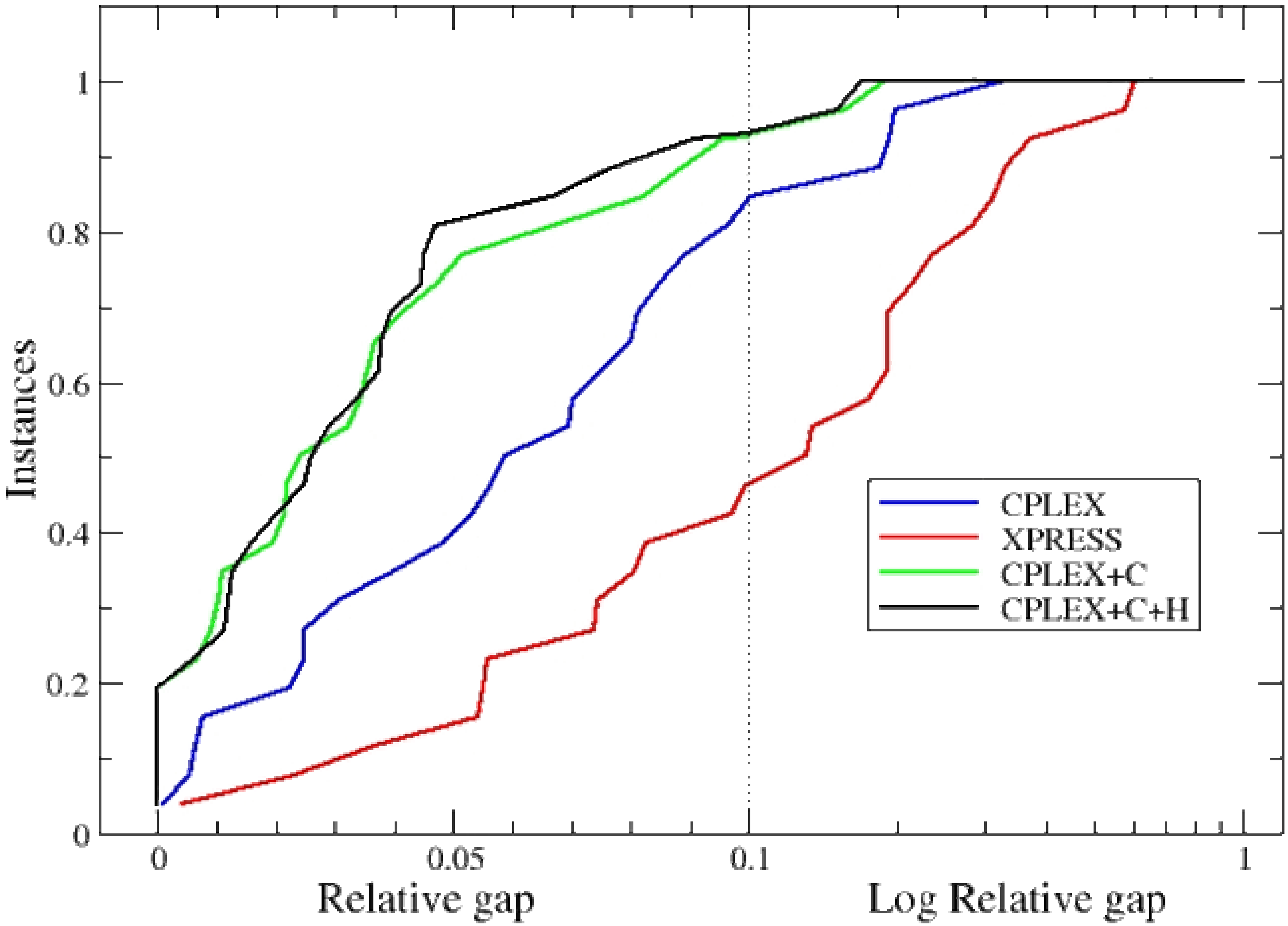
Computational Experiments

- 83 instances from Wolsey's web page: http://www.core.ucl.ac.be/wolsey/ufcn.htm
- Models in Xpress-Modeler & MPS format
- Translated in consistent GAMS model with 83 data files
- Verified translated instances:
  - Same relaxed objective value
  - Matrix statistics
  - When solved optimally, same objective value
- Computing Environment
  - Athlon 2800, 2 GB memory running Linux
  - GAMS 21.4, Cplex 9.0, Xpress 2003
  - Time limit 1800 seconds

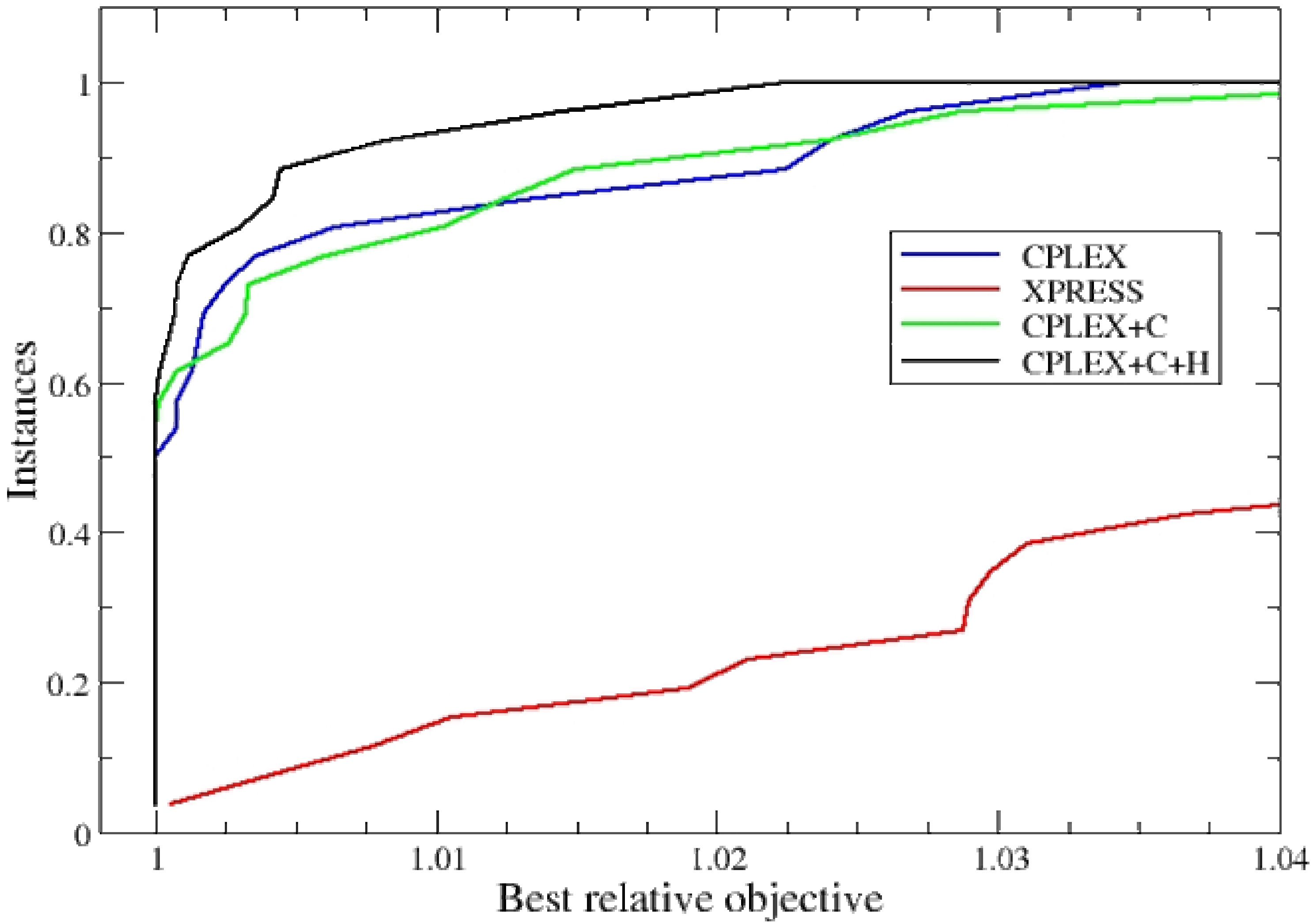
No BCH – 1800 s Time Limit



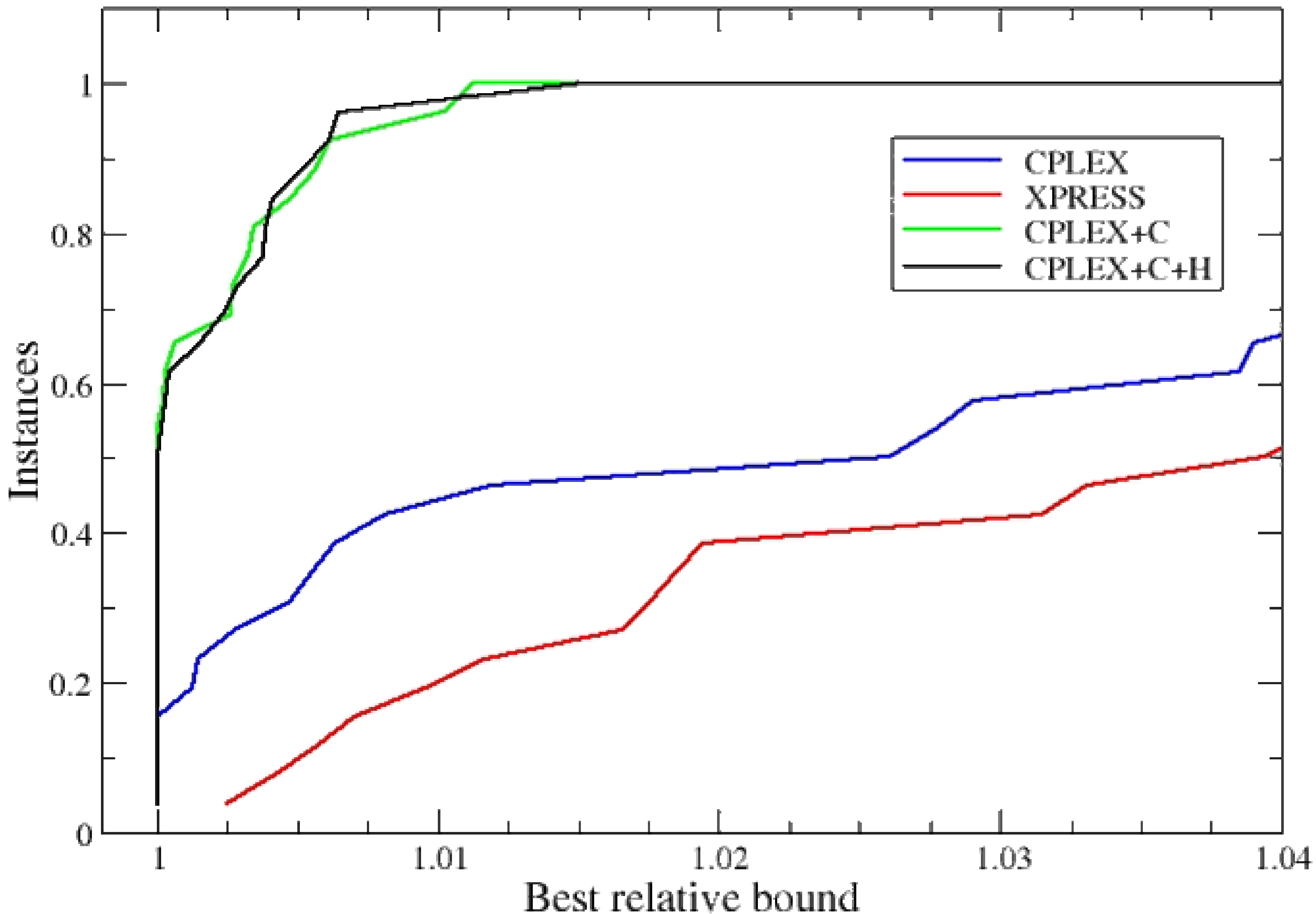
Unsolved Instances - Gap



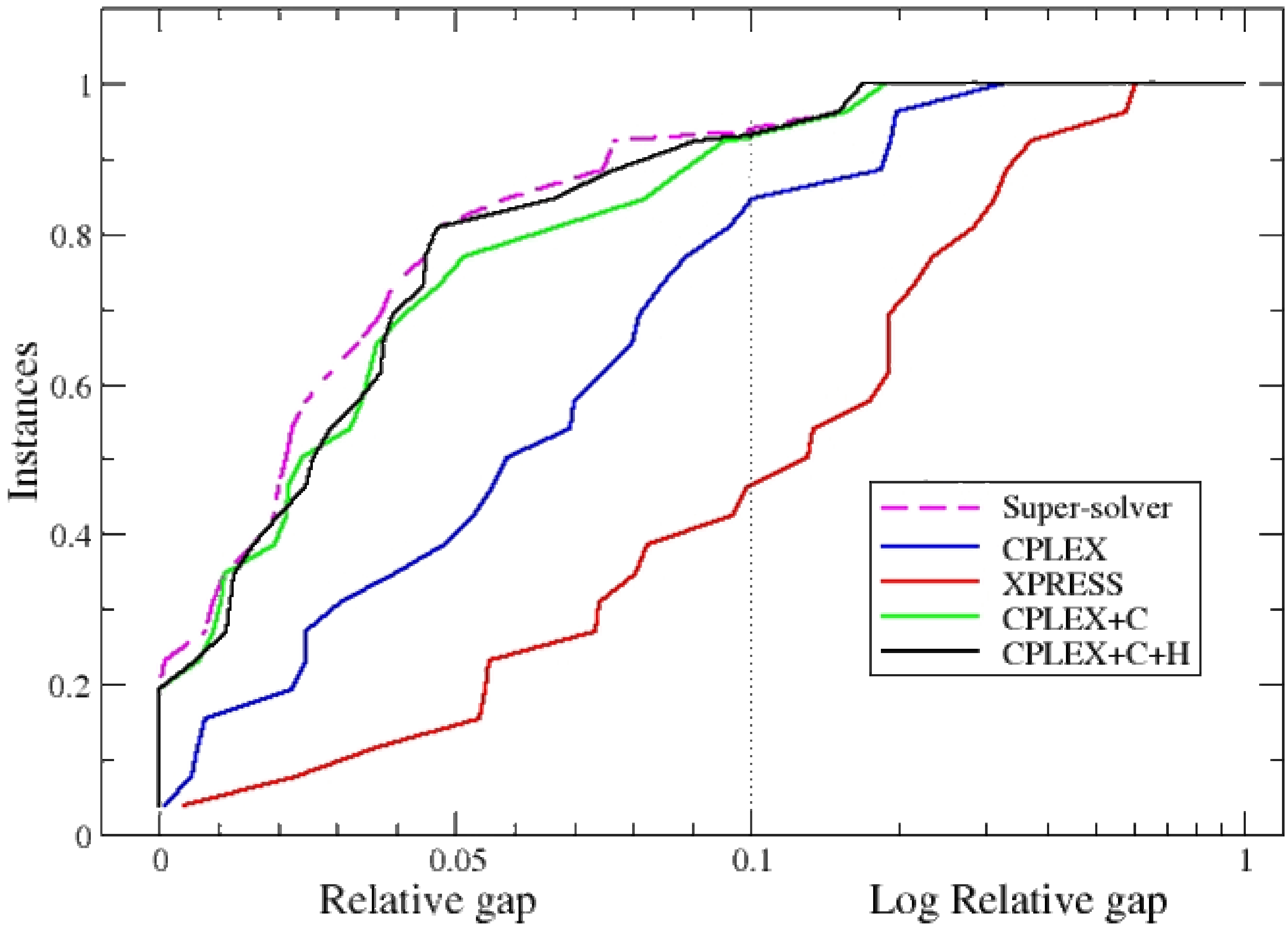
Unsolved Instances - Objective



Unsolved Instances - Bound



Unsolved Instances - Gap



Conclusion

- BCH Facility automates all major steps necessary to define, execute, and control the use of user defined routines within the framework of general purpose MIP codes
- Cut separation models provides a new application area for Global Optimization
- More information:
  - **`www.gams.com/docs/bch.htm`** **(BCH)**
  - **`www.gams.com/apps/fcnet.htm`** **(FC-Network)**
  - **`www.gams.com/presentations`** **(Presentation)**