SBB: A New Solver for Mixed Integer Nonlinear Programming

Michael R. Bussieck

GAMS Development Corp.

Arne Drud

ARKI Consulting & Development A/S

Overview

- *Introduction: The MINLP Model*
- *The B&B Algorithm*
- *Global/Local Solutions: What can go wrong*
- *The SBB Design Criteria and Implementation*
- *Some Results and Comparison with DICOPT*
- *Conclusions and Future Work*

The MINLP Model

- Min or max $f(x,y)$

  s.t. $g(x,y) = b$

  $l \le x \le u$ continuous

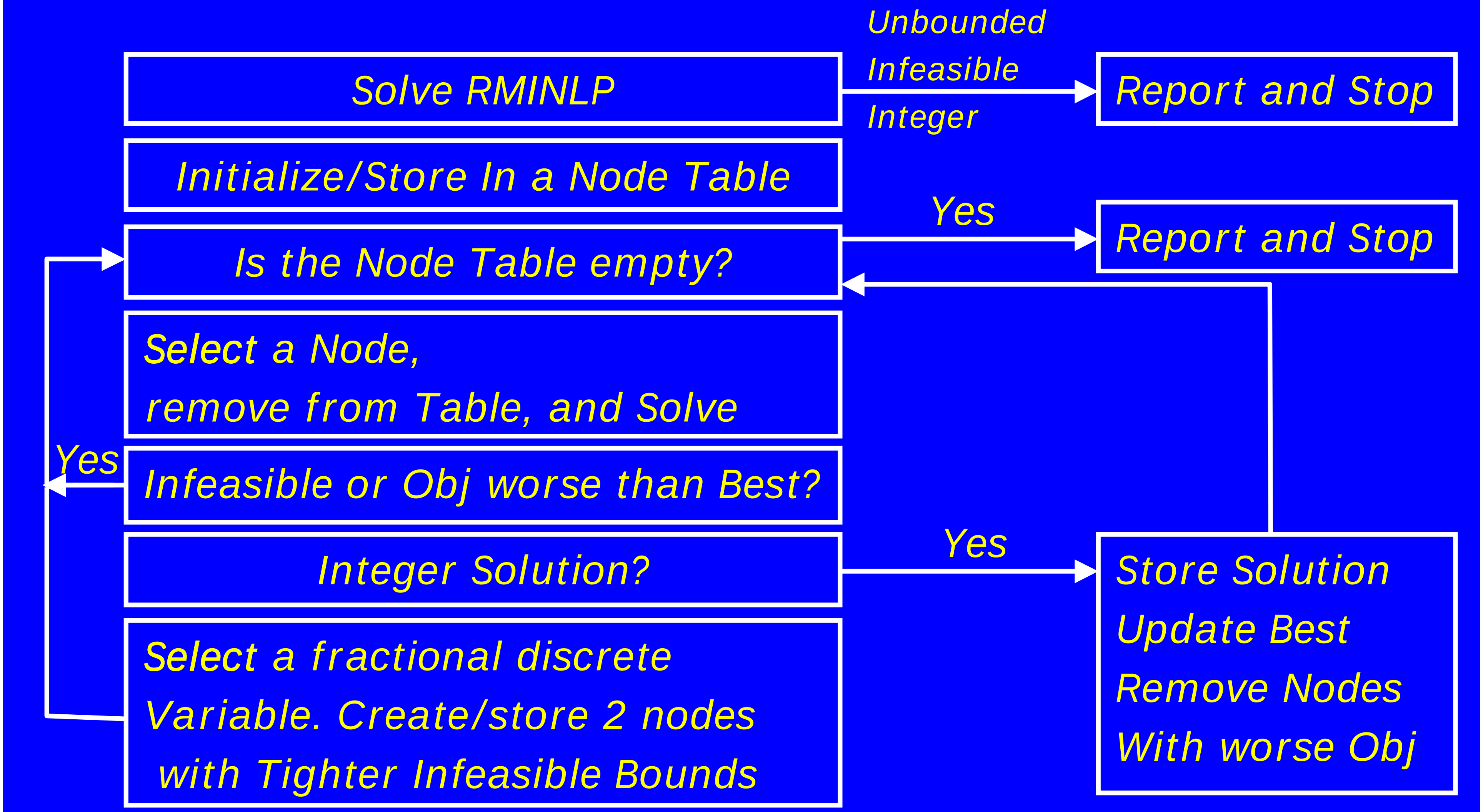
  $li \le y \le ui$ integer

- *For convenience we only discuss “Min”*
- *SBB handles also: SOS1, SOS2, SemiContinuous, and SemiInteger Variables (not included in this talk)*

Some Definitions

- *When the Integrality Constraints on y are removed we get the corresponding RMINLP model*
- *A Node is the RMINLP model in which the bounds on y may be tigthened*

The B&B Algorithm



Solve RMINLP

Unbounded

Infeasible

Integer

Report and Stop

Initialize/Store In a Node Table

Is the Node Table empty?

Yes

Report and Stop

Select a Node,
remove from Table, and Solve

Infeasible or Obj worse than Best?

Yes

Integer Solution?

Yes

Store Solution
Update Best
Remove Nodes
With worse Obj

Select a fractional discrete
Variable. Create/store 2 nodes
with Tighter Infeasible Bounds

Assumptions Needed to Get a Global Optimum

- If a Node is declared “Infeasible” then it is indeed Globally Infeasible
- The objective for a Node is a valid lower bound for all integer solutions satisfying the bounds of the Node
- These assumptions are satisfied if the RMINLP model is convex in both x and y.

Non-Convex Models: Potential Problems

- *The nodes are solved using an NLP Local Optimization Algorithm. Potential Problems are:*
  - *Branches or the Search Tree can incorrectly be cut off because:*
    - *A node is declared “Infeasible” even though it is only “locally infeasible” and there exist feasible regions.*
    - *A node is fathomed because the objective for the node is a poor approximation for the global optimum for the node.*
- *In addition: Problems will occur if the NLP solver fails.*

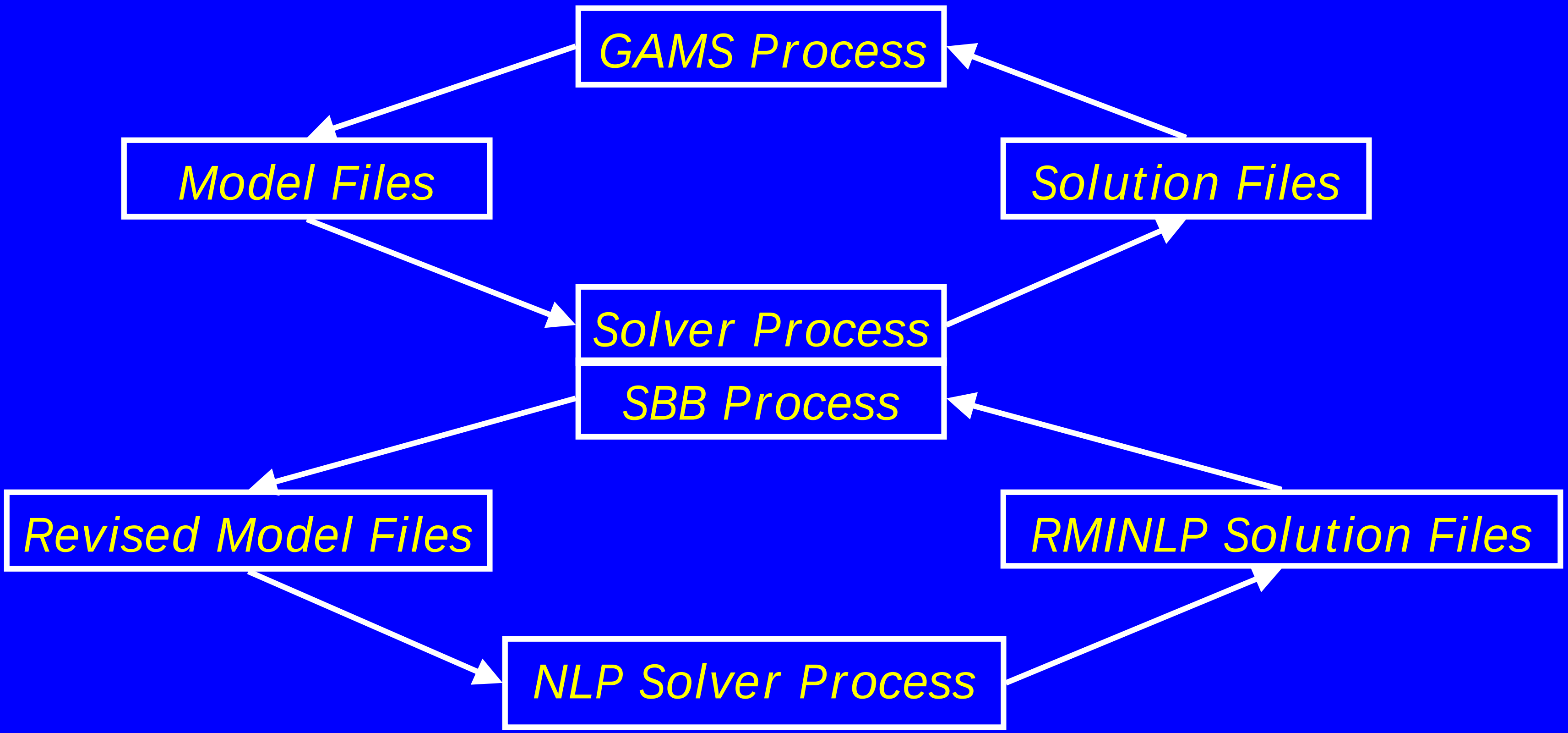
Desirable Properties of MINLP Models

- *The feasible region for the RMINLP is connected and bound tightening in the nodes does not change this property.*
- *The nodes are not likely to be caught in bad local optima.*

Design Objectives for SBB

- Should be able to use any existing GAMS NLP solver
- Should be able to handle solver failures
- Should have an option that can help against incorrect infeasibilities
- Should report information on observed non-convexities.

The SBB Implementation
The Concept



GAMS Process

Model Files

Solution Files

Solver Process

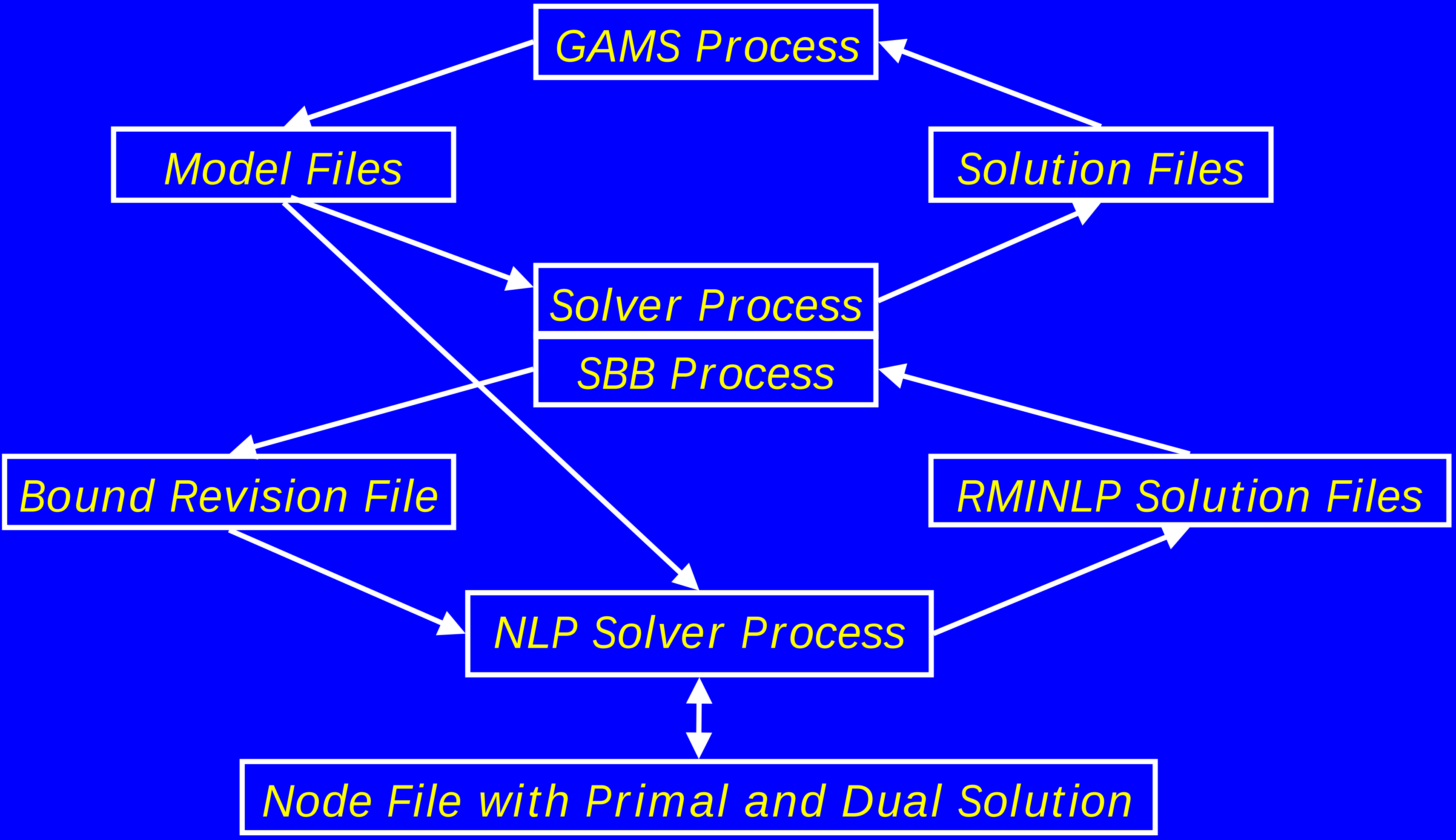
SBB Process

Revised Model Files

RMINLP Solution Files

NLP Solver Process

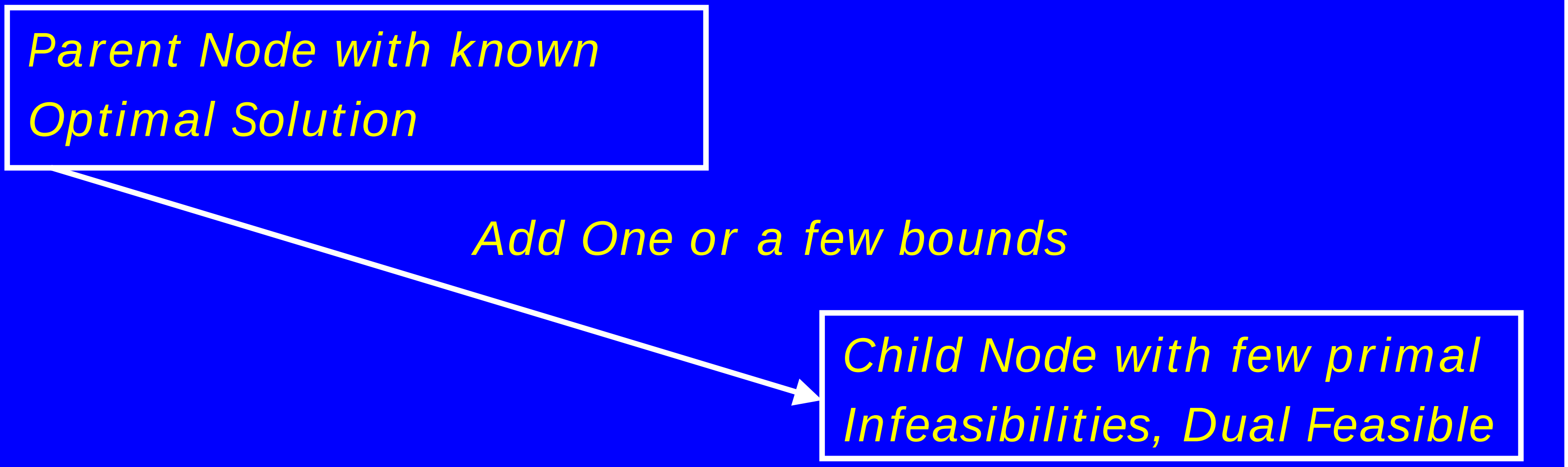
Efficient SBB Implementation



Making a GAMS NLP Solver SBB-aware

- *The GAMS Link is revised so*
  - *It knows whether it is called by SBB or not*
  - *It can read the Bound Revision File*
  - *It can get Bounds of parent plus initial primal and dual values from a node file and it can store the solution in the Node File*
- *Most of this is handled in the GAMS interface library*
- *Currently, no special solver changes*

Solving the NLP Submodels



Parent Node with known Optimal Solution

Add One or a few bounds

Child Node with few primal Infeasibilities, Dual Feasible

- Solution Approaches:
  - Standard Phase 1 – Phase 2
  - Specialized Dual-Type Parametric Programming

Some SBB Options

- `failseq solver1.opt1 solver2.opt2`
  - *Try solver1 with options defined by opt1. If it fails, try solver2 with options defined by opt2, etc. If all fail,* ***ignore*** *the node and continue the search*
- `infeasseq level solver1.opt1 solver2.opt2`
  - *If a node with depth ≤ level is locally infeasible then try solver1 with options defined by opt1, etc.*
- `rootsolver solver.opt`
  - *Use a special solver/option pair for the root node*
- `subiter/subres max`
  - *Avoid that one node uses all resources*

67 Test Models

Name	Rows	Columns	Discrete	NonZeros	NL NNZ
4stufen	99	150	48	319	87
alanbar	9	10	4	26	3
batchdesb	21	21	9	55	10
beuster	115	158	52	398	159
cecil_13ba	900	842	180	2814	360
contvar	285	297	87	1281	530
deb10bar	131	184	22	694	432
deb6	508	476	20	2342	1432
deb7	898	814	10	4116	2816
deb8	898	824	10	4136	2816
deb9	918	814	10	4156	2816
detf1	6206	4408	400	26610	15400
ecop	1547	1536	19	3932	759
ecop2	927	937	2	2215	440
enpro48ba	216	155	92	744	29
enpro56ba	193	129	73	653	24
ex1222	4	4	1	9	2
ex1223	14	12	4	40	17
ex1223a	10	8	4	32	9
ex1224bar	9	13	8	33	6
ex1225bar	12	10	6	29	2
ex1233bar	66	54	12	223	28
ex1243bar	98	70	16	331	36
ex1244bar	131	97	23	471	52
ex1252bar	45	41	15	120	36
ex1263aba	37	26	24	155	32
ex1263bar	57	94	72	243	32
ex1264aba	37	26	24	155	32
ex1264bar	57	90	68	239	32
ex1265aba	46	37	35	224	50
ex1265bar	76	132	100	349	50
ex1266aba	55	50	48	304	72
ex1266bar	97	182	138	478	72
ex3bar	33	34	8	102	5
ex4	31	37	25	237	127
feedtrayba	93	99	7	453	282
fuelbar	17	17	3	41	6
gear	5	9	4	13	4
hdabar	720	724	13	2208	464
johnall	193	195	190	958	573
meanvarxb	46	37	14	113	7
nuclear104	3340	13011	10920	67870	44884
nuclear24b	635	994	600	6364	4944
nuclear25b	661	1060	650	6769	5240
nuclear49b	1433	3343	2450	19503	14109
ortez	75	88	18	269	54
parallelbar	117	207	25	754	155
procselbar	9	12	3	28	2
pumpbar	36	26	9	96	36
ravembar	188	114	54	613	28
saa_1bar	554	557	50	2408	1650
saa_2bar	6054	6108	50	27201	6000
saa_gams	554	557	50	2408	1650
spectra2ba	74	71	30	411	240
springbar	10	19	12	46	14
synheatba	66	58	12	227	28
trimlossba	55	50	48	305	72
utilbar	169	147	28	469	10
var_con10b	466	575	12	2559	1920
var_con5ba	466	575	12	2559	1920
water3	138	196	28	757	46
water4	138	196	126	757	46
waters	138	196	14	645	46
watersbp	138	196	28	645	46
waterxbar	56	72	14	243	60
waterz	138	196	126	645	46

Capability Issues

- *Although discrete Variables in non-linear terms work in principal with DICOPT it usually results in a bad problem formulations. Therefore, it is disabled in GAMS/DICOPT. We recommend not to use discrete non-linear variables but permit their use.*
- *DICOPT does not support SOS and Semi Continuous Variables.*

Solution Status for 67 Models

		DICOPT						
		O	U	R	N	C	Total	
	O	36	2		10	12	60	“Optimal”
SBB	U		1		2	3	6	Unfinished (Int. Sol)
	R			1			1	Root Node Failure
	N							No Integer Solution
	C							Capability Issues
	Total	36	3	1	12	15	67	

Subsolvers: CONOPT2, CPLEX6.6.

No options and no Alternatives at failure/infeasibility

Comparison of 36 “Optimal” Models

	Objective	Solution Times
The Same	20	11
SBB Better	16	4
DICOPT Better	0	21

Future Work

- Today MINLP is at the stage where MIP was 10-15 years ago.
- Model Formulations:
  - Recommendations for good formulations and warnings against bad formulations
  - Preprocessing
- Algorithmic/Theoretical Work:
  - Preprocessing, Probing
  - Cut Generation, Pseudo Cost, ...
- System Work
  - Parallel NLP Solvers