Solving Large-Scale Nonlinear Programming Problems by Constraint Partitioning

Benjamin W. Wah and *Yixin Chen*

Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Email: wah@uiuc.edu
URL: http://manip.crhc.uiuc.edu

Outline

- Observation
  - Constraints in many applications are structured
- Approach
  - Partition problem by its constraints into subproblems
- Issues addressed
  - Resolution of violated global constraints
  - Automated analysis of problem structure and its partitioning
  - Optimality of partitioning
  - Demonstrations of improvements
- Future work

Real Constraints Are Structured

- Constraints model entities and actions with structural or temporal locality
- They may model:
  - Relations among components in close proximity for problems of physical structures
  - Relations among actions close to each other in time for scheduling problems

Constraint Locality in MINLP

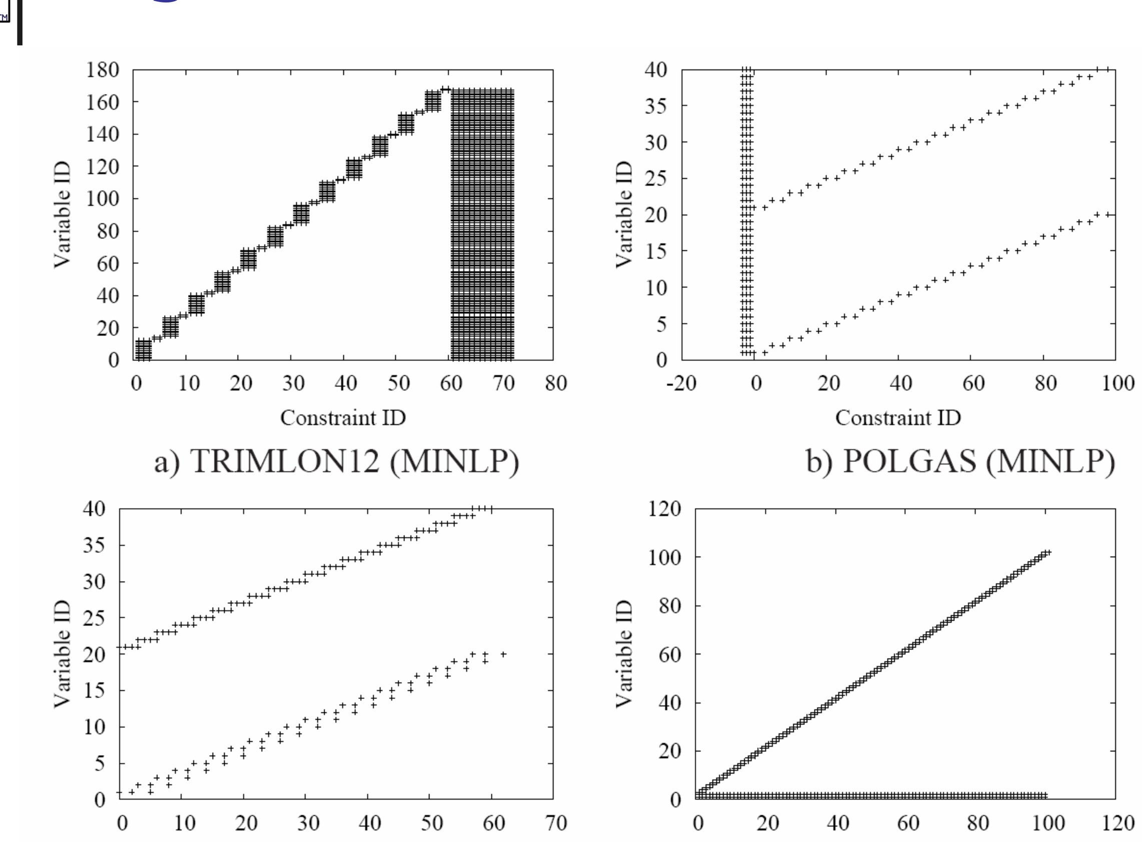
- TRIMLON
  - Minimize the trim loss in producing a set of paper rolls from raw paper rolls
  - Trimlon12 has 168 variables (integer *n*, real *y*, *m*) and 72 constraints
    - Not solvable by any existing MINLP solver from starting point specified

variables: $y[j],\ m[j],\ n[j,i]$ where $i = 1,\cdots,I;\ j = 1,\cdots,J$

objective: $\min_{z=(y,m,n)} f(z) = \sum_{j=1}^{J}(c[j]\cdot m[j] + C[j]\cdot y[j])$ (OBJ)

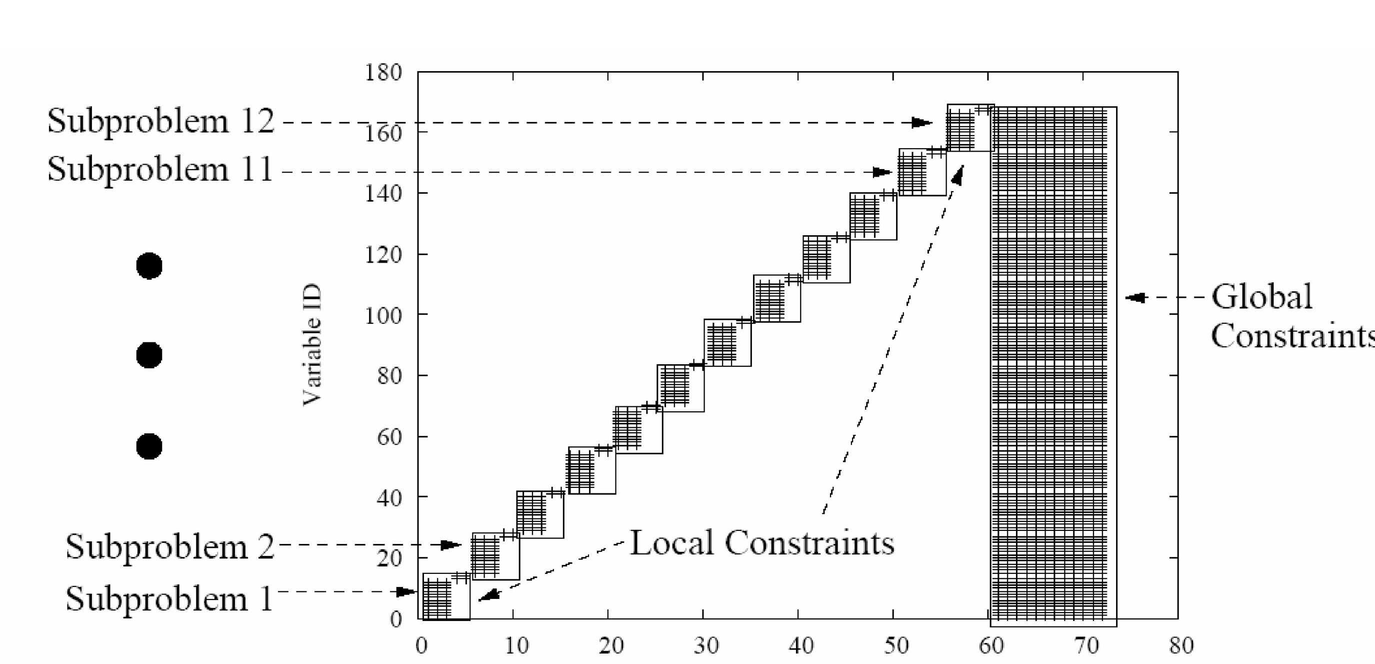
subject to:
$B_{min} \le \sum_{i=1}^{I}(b[i]\cdot n[i,j]) \le B_{max}$ (C1)
$\sum_{i=1}^{I} n[i,j] - N_{max} \le 0$ (C2)
$y[i] - m[j] \le 0$ (C3)
$m[j] - M\cdot y[j] \le 0$ (C4)
$Nord[i] - \sum_{j=1}^{J}(m[j]\cdot n[i,j]) \le 0.$ (C5)

Regular Constraint Structure



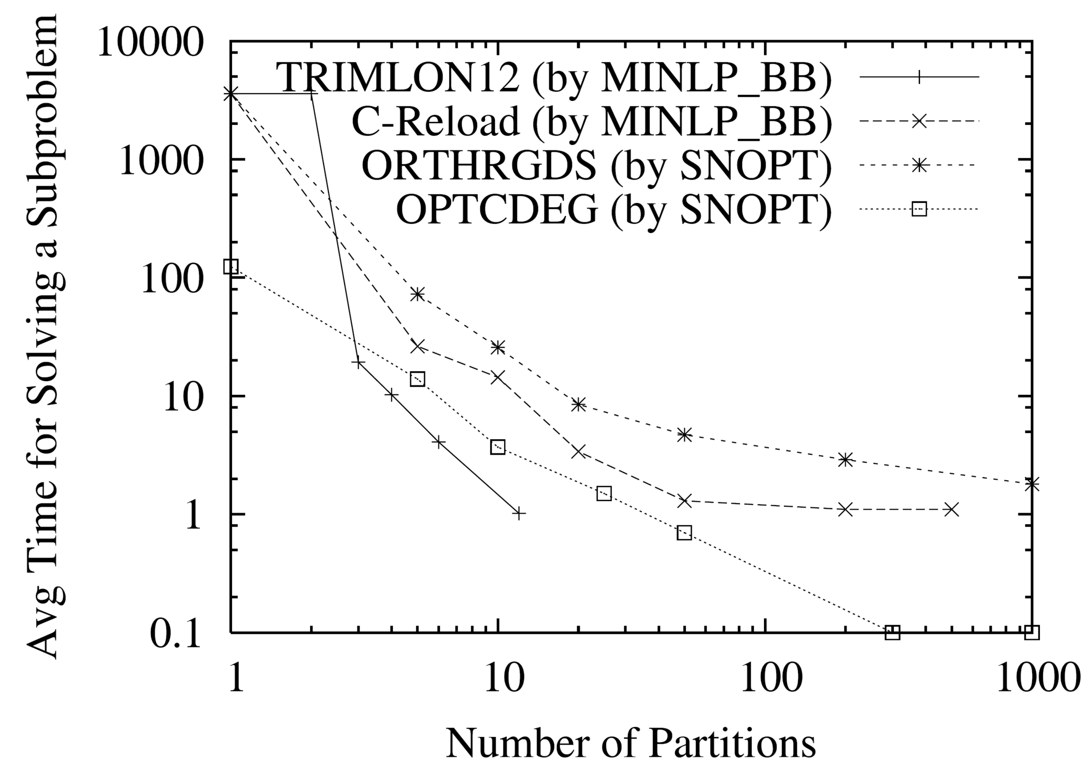
a) TRIMLON12 (MINLP)
b) POLGAS (MINLP)
c) OPTCDEG3 (CNLP)
d) ORTHRGDS (CNLP)

Partitioning of TRIMLON12



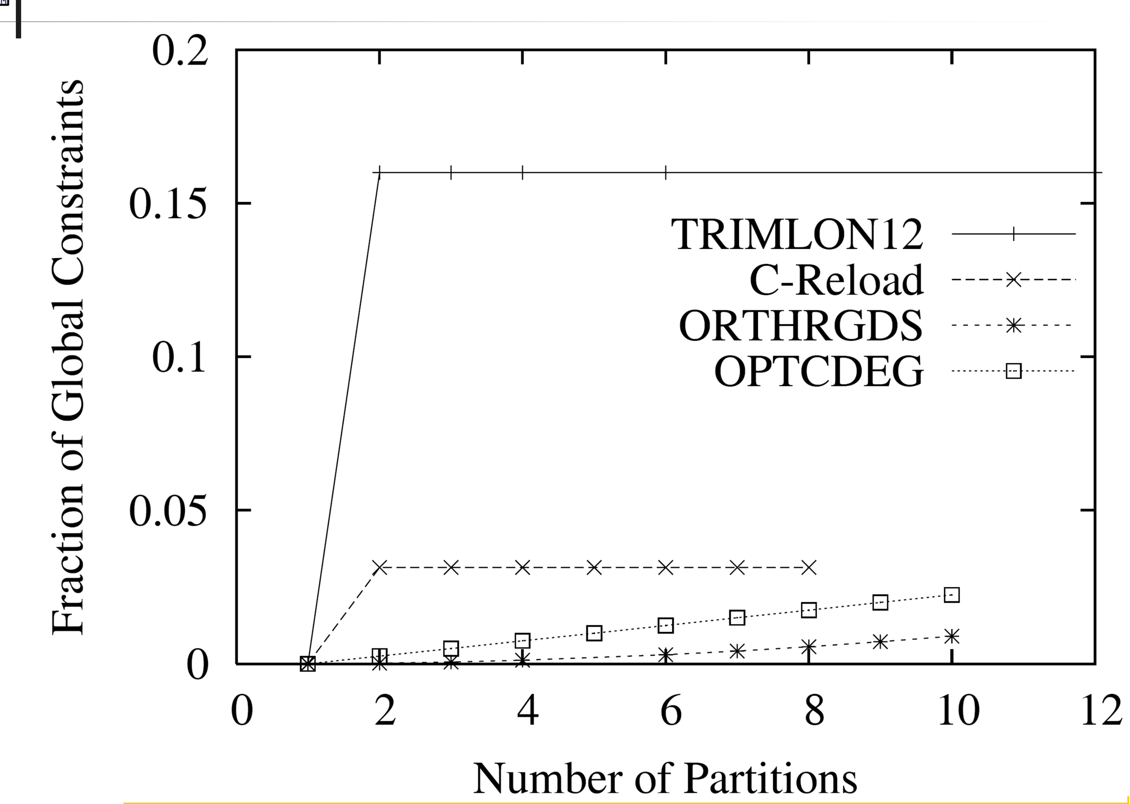
12 out of 72 constraints (16.7%) are global constraints

Substantial Decrease in Subproblem Time



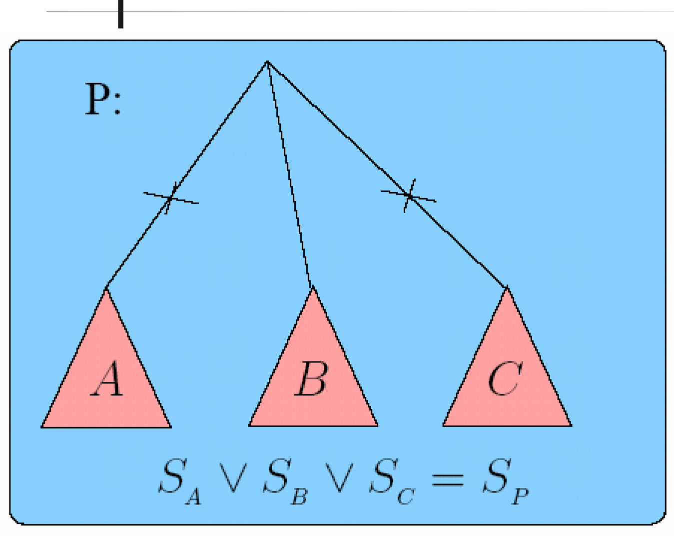
Substantial decrease in overhead as number of constraints in a subproblem is reduced

Monotonic Increase in Global Constraints

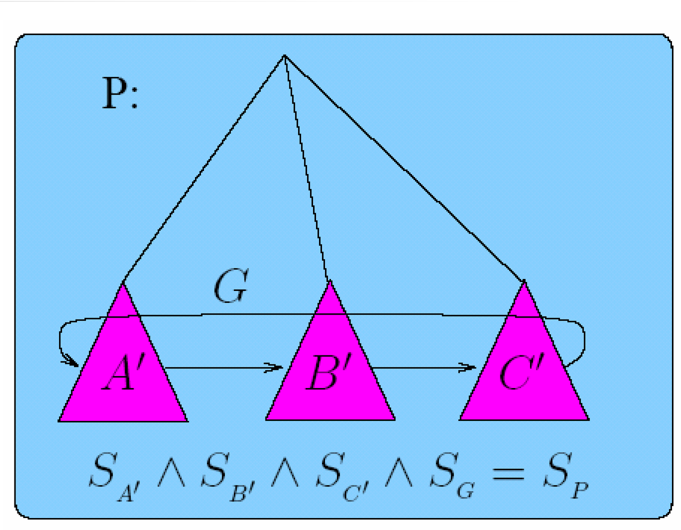


More overhead for resolving global constraints as number of partitions increaes

Two Partitioning Approaches



Subspace Partitioning



Constraint Partitioning

Constraint partitioned subproblems are much simpler

Previous Work: Penalty Methods

- Global minima of penalty functions

$$L_s(z,c) = f(z) + c \cdot \left[\sum_{i=1}^{m} (h_i(z))^{\rho} + \sum_{i=1}^{r} (\max(0, g_i(z)))^{\rho} \right]$$

- Local minima of penalty functions
  - KKT: differentiability and continuity assumptions
    - System of nonlinear equations that cannot be partitioned
  - L-1 penalty formulation

$$\ell_1(z,c) = f(z) + c \cdot \max\Big(0, |h_1(z)|, \cdots, |h_m(z)|, g_1(z), \cdots, g_q(z)\Big)$$

Previous Work: Partitioning

- Subspace partitioning
  - Branch on discrete variables into subproblems
  - Prune inferior subproblems by derived bounds
  - Subproblems must be solvable easily
  - Examples: GBD, OA, GCD, B&R, B&B
- Separable programming
  - Solve dual problem that can be partitioned by its constraints into much simpler subproblems
  - Require linearity or convexity of functions

Issues Addressed

- Resolution of violated global constraints
- Automated analysis of problem structure (in some standard form) and its partitioning
- Optimality of partitioning
  - Tradeoffs between the number of global constraints to be resolved and the time to evaluate a subproblem
- Demonstration of improvements over existing methods

Resolution of Global Constraints

- Necessary and sufficient condition governing constrained local minima [AI05]
  - Loose assumptions, without continuity and differentiability of constraint functions
  - Easy to satisfy: looking for penalties that are larger than some thresholds
- Partitioning of the N&S condition into a set of necessary conditions that are sufficient collectively
  - One necessary condition for each subproblem
  - One necessary condition for the global constraints

The Partition-and-Resolve Framework

$$
\begin{array}{llll}
(P_t): & \min_z \ J(z) & & \\
& \text{subject to } \ h^{(t)}(z(t)) = 0, & g^{(t)}(z(t)) \le 0 & \text{(local constraints)} \\
& \text{and } \ H(z) = 0, & G(z) \le 0 & \text{(global constraints)}
\end{array}
$$

$L_m(z, \gamma, \eta) \uparrow_{\gamma,\eta}$ to find γ^{**} and η^{**}

$\min_{z(1)} J(z) + \gamma^T |H(z)| + \eta^T \max(0, G(z))$ subject to $h^{(1)}(z(1)) = 0$ and $g^{(1)}(z(1)) \le 0$

• • •

$\min_{z(N)} J(z) + \gamma^T |H(z)| + \eta^T \max(0, G(z))$ subject to $h^{(N)}(z(N)) = 0$ and $g^{(N)}(z(N)) \le 0$

Issues Addressed

- Resolution of violated global constraints
- Automated analysis of problem structure (in some standard form) and its partitioning
- Optimality of partitioning
  - Tradeoffs between the number of global constraints to be resolved and the time to evaluate a subproblem
- Demonstration of improvements over existing methods

Implementation of P&R Framework

```
procedure CPOPT
    call automated_partition(); // automatically partition the problem //
    γ ⟵ γ0; η ⟵ η0; // initialize penalty values for global constraints//
    repeat      // outer loop //
        for t = 1 to N // iterate over all N stages to solve (8) in each stage //
            apply an existing solver to solve (8)
            call update_penalty(); // update penalties of violated global constraints //
        end_for;
    until stopping condition is satisfied
end_procedure
```

$$
\begin{array}{ll}
\min_{z(t)} & J(z) + \gamma^T a + \eta^T b \\
\text{subject to} & h^{(t)}(z(t)) = 0 \ \text{ and } \ g^{(t)}(z(t)) \le 0, \\
& -a \le H(z) \le a \ \text{ and } \ G(z) \le b
\end{array}
$$

Partitioning Constraints into Subproblems

- *Index vector* for representing variables and constraints in AMPL models
  - Indexing is essential for representing complex problems
  - Partitioning by index vectors can be interpreted meaningfully
- *Partitioning index vector*

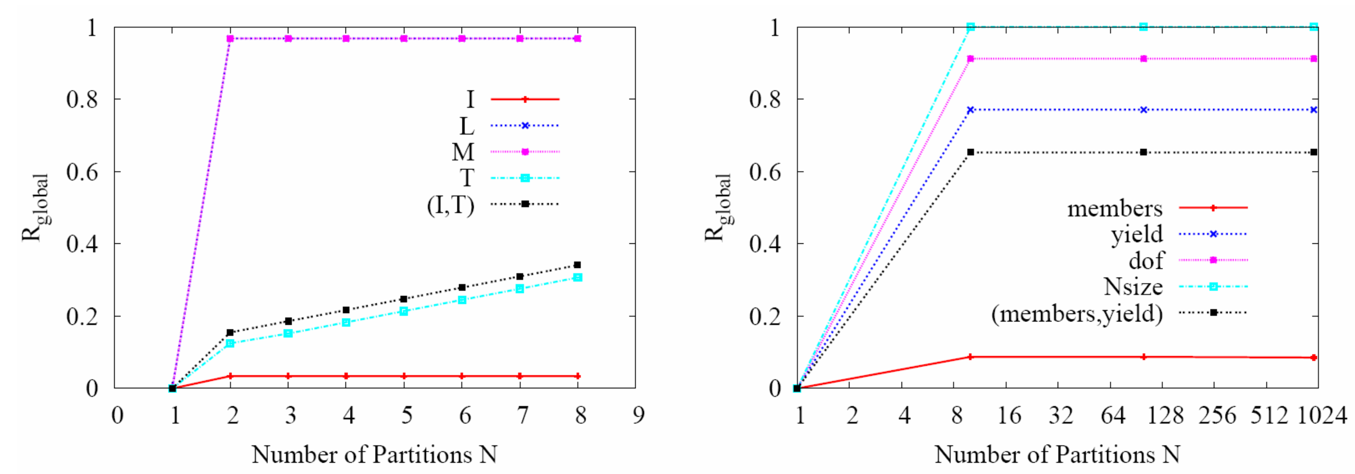
$$\text{PIV} = \{J\}, \text{and } S_1 = \{1\}, \cdots, S_{12} = \{12\}$$

Issues Addressed

- Resolution of violated global constraints
- Automated analysis of problem structure (in some standard form) and its partitioning
- Optimality of partitioning
  - Tradeoffs between the number of global constraints to be resolved and the time to evaluate a subproblem
- Demonstration of improvements over existing methods

Metric of Partition-ability

$$R_{global} = \frac{\text{number of global constraints}}{\text{total number of constraints}}$$



a) C-RELOAD-r-104 b) SPACE-960-r (MINLP)

Index that minimizes R_{global} is indepdent of N

Optimal Number of Partitions

Convex relationship between N and total solution time

- Start from large N
- Estimate time per iteration
- Reduce N until time per iteration increases

Space-960-r MINLP

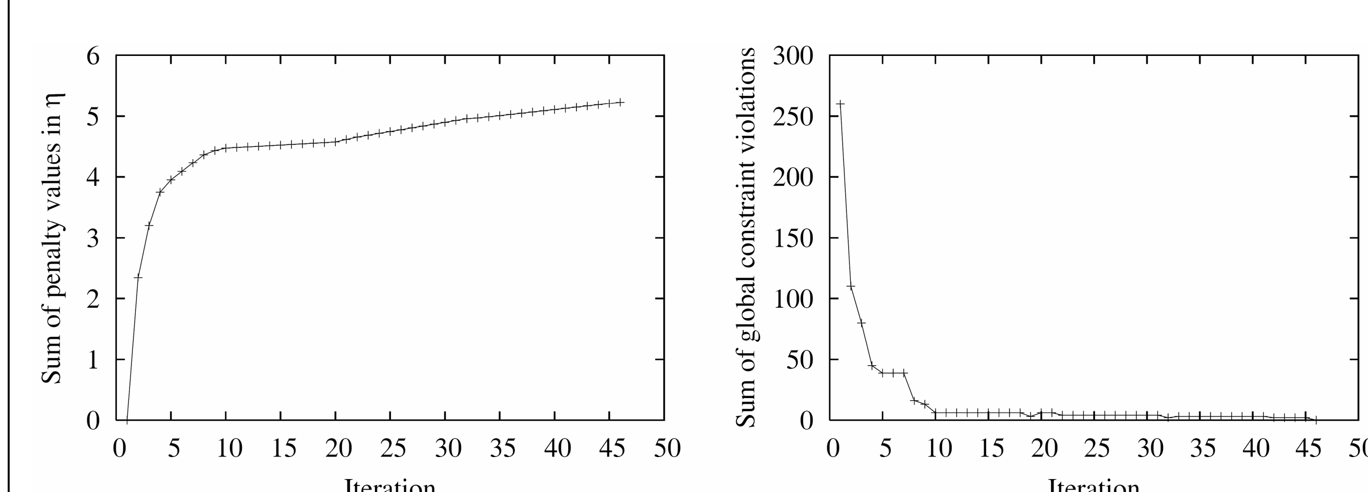
Number of partitions N	1	15	30	60	120	240	480
Time per subproblem	>3600	8.4	$\boxed{3.3}$	3.1	2.8	2.7	2.6
Time per iteration	>3600	126	$\boxed{99}$	186	336	648	1248
Number of iterations	1	1	$\boxed{1}$	2	2	2	5
Total time to solve problem	>3600	126	$\boxed{99}$	372	672	1296	6240

Estimated Subproblem Times

Time for each subproblem has little variance when the original problem is partitioned evenly

Problem instance	ORTHRGDS	ORTHRGDS	SPACE-960-r	SPACE-960-r
Number of partitions N	1000	20	100	10
Avg. time per subproblem ($T_p(N)$)	1.8	8.5	2.8	9.4
Std. dev. of time per subproblem	0.021	0.31	0.013	0.015

Solving TRIMLON12 by CPOPT



a) Sum of penalty values in η b) Sum of global constraint violations (C5)

46 iterations to resolve all global constraints

Issues Addressed

- Resolution of violated global constraints
- Automated analysis of problem structure (in some standard form) and its partitioning
- Optimality of partitioning
  - Tradeoffs between the number of global constraints to be resolved and the time to evaluate a subproblem
- Demonstration of improvements over existing methods

Difficult-to-Solve MINLP Benchmarks

ID	n_c	n_v	Quality	Time	Quality	Time	Quality	Time
MINLP Test Problem			MINLP_BB		BARON		CPOPT(MINLP_BB)	
C-RELOAD-q-49	1430	3733	–		–	–	$\boxed{-1.13}$	69.45
C-RELOAD-q-104	3338	13936	–	–	–	–	$\boxed{-1.14}$	353.74
Ex12.6.3	57	92	$\boxed{19.6}$	23	$\boxed{19.6}$	423.1	$\boxed{19.6}$	13.43
Ex12.6.4	57	88	$\boxed{8.6}$	70	$\boxed{8.6}$	478.2	$\boxed{8.6}$	2.94
Ex12.6.5	76	130	15.1	4	$\boxed{10.3}$	845.5	10.6	3.33
Ex12.6.6	97	180	$\boxed{16.3}$	18	$\boxed{16.3}$	937.4	$\boxed{16.3}$	149.40
PUMP	34	24	–	–	131124	977	$\boxed{130788}$	84.53
SPACE-960-i	6497	5537	–	–	–	–	$\boxed{7.65E6}$	187.43
SPACE-960-ir	3617	2657	–	–	–	–	$\boxed{7.64E6}$	145.76
SPACE-960	8417	15137	–	–	–	–	$\boxed{7.84E6}$	1206.43
SPACE-960-r	5537	12257	–	–	–	–	$\boxed{5.13E6}$	160.45
STOCKCYCLE	97	480	–	–	436341	n/a	$\boxed{119948.7}$	6.45
TRIMLON4	24	24	12.2	10	$\boxed{8.3}$	11.0	$\boxed{8.3}$	2.73
TRIMLON5	30	35	12.5	14	$\boxed{10.3}$	55.3	$\boxed{10.3}$	24.5
TRIMLON6	36	48	18.8	19	$\boxed{15.6}$	1092.9	$\boxed{15.6}$	15.94
TRIMLON7	42	63	–	–	$\boxed{17.5}$	990.7	18.1	65.34
TRIMLON12	72	168	–	–	–	–	$\boxed{95.5}$	345.50
TRIMLOSS4	64	105	10.8	99	–	–	$\boxed{10.6}$	9.76
TRIMLOSS5	90	161	12.6	190	–	–	$\boxed{10.7}$	76.85
TRIMLOSS6	120	215	–	–	–	–	$\boxed{22.1}$	69.03
TRIMLOSS7	154	345	–	–	–	–	$\boxed{26.7}$	59.32
TRIMLOSS12	384	800	–	–	–	–	$\boxed{138.8}$	323.94

Difficult-to-Solve NLP Benchmarks

ID	n_c	n_v	Quality	Time	Quality	Time	Quality	Time
CNLP Test Problem			Lancelot		SNOPT		CPOPT(SNOPT)	
CATENARY	166	501	-	-	-	-	$\boxed{-1.35E5}$	245.64
DTOC6	5000	10001	-	-	-	-	$\boxed{1.02E6}$	58.05
EIGMAXB	101	101	$\boxed{0.91}$	1.34	-	-	1.87	24.33
GILBERT	1000	1000	2459.46	1.12	4700.61	689.18	$\boxed{2454.67}$	39.55
HADAMARD	256	129	-	-	-	-	$\boxed{0.99}$	7.88
KISSING	903	127	0.84	123.43	-	-	$\boxed{0.77}$	73.45
OPTCDEG	4000	6001	-	-	$\boxed{45.76}$	10.23	46.98	19.65
ORTHREGC	5000	10005	-	-	3469.05	557.98	$\boxed{2614.34}$	143.65
ORTHREGD	5000	10003	-	-	8729.64	208.27	$\boxed{7932.92}$	123.49
ORTHRGDM	5000	10003	$\boxed{1513.80}$	4.56	10167.82	250.00	2340.34	20.34
ORTHRGDS	5000	10003	912.41	4.20	-	-	$\boxed{894.65}$	105.34
VANDERM1	199	100	-	-	-	-	$\boxed{0.0}$	45.34
VANDERM3	199	100	-	-	-	-	$\boxed{0.0}$	36.70
VANDERM4	199	100	-	-	-	-	$\boxed{0.0}$	52.33

Future Work

- General solver for large-scale applications
  - Automated analysis of constraint structure
  - Interface to existing solvers for solving partitioned subproblems
- Many other interesting applications
  - VLSI design and layouts
  - AI planning applications [SGPlan, AI05]