Exploring Nonlinear Constraint Optimization and their Applications

Benjamin W. Wah

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

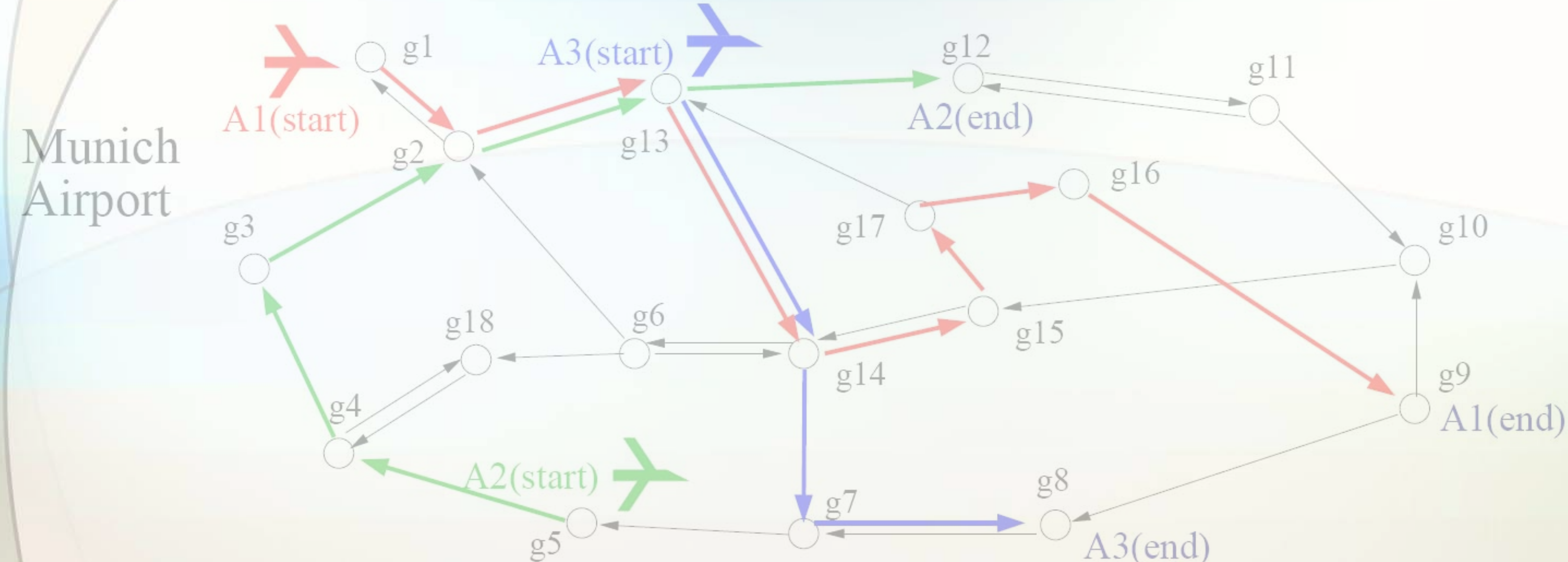
Outline

- Observation
  - Constraints in many application problems 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
- Conclusions

Nonlinear Constrained Optimization

- An application problem defined by
  - A set of mixed (integer and real) variables
  - A nonlinear objective function
  - A set of nonlinear constraints (conditions to be satisfied in the application)
- Exists in every engineering field
  - Planning of spacecraft and satellite operations
  - VLSI placement of components on a CPU chip
  - Design of aircrafts
  - Design of a petroleum pipeline

An Airport Planning Example



Facts: at(A1,g1), blocked(g1), unblocked(g1)

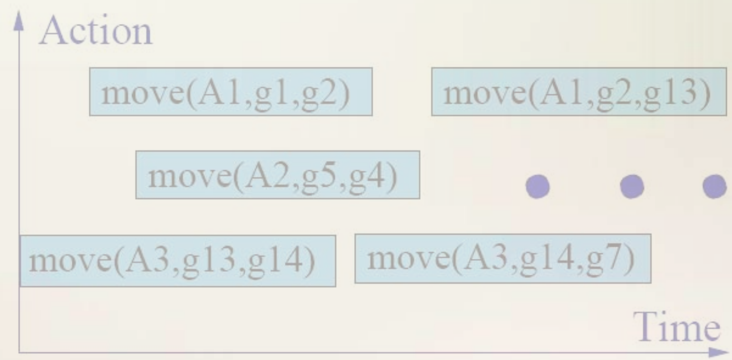
Actions: move(A1,g1,g2)

Initial Facts: at(A1,g1), at(A2,g5), at (A3,g13)

Subgoals: at(A1,g9), at(A2,g12), at (A3,g8)

Objective: minimize total time

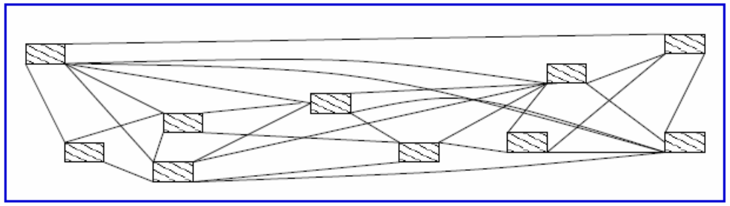
Problem Specification



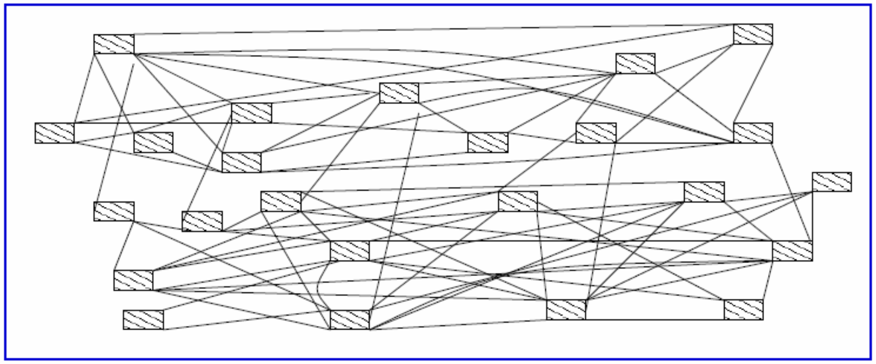
A Solution Plan

Constraints in AIRPORT

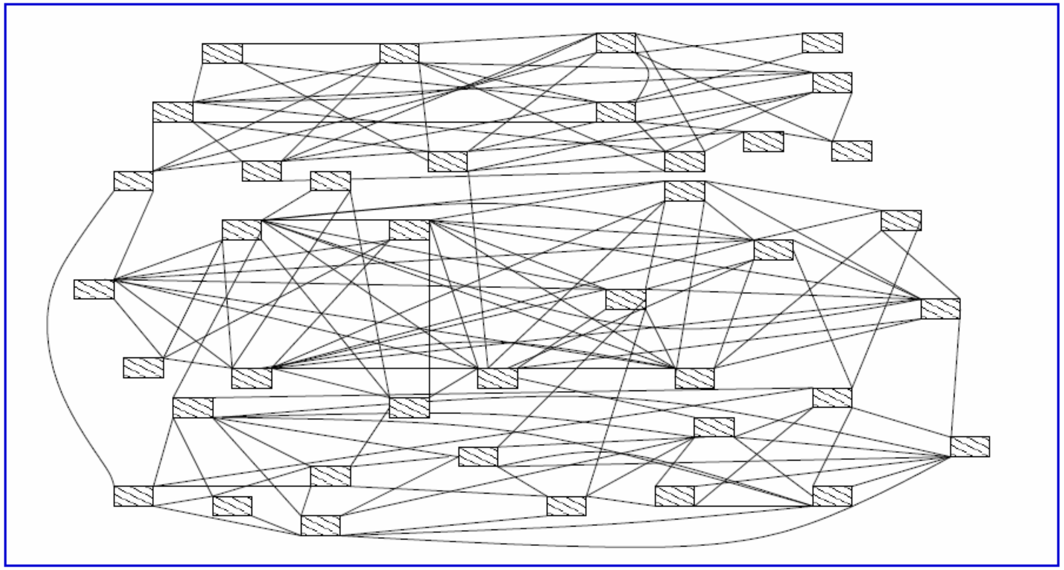
1 Subgoal (1 Plane)



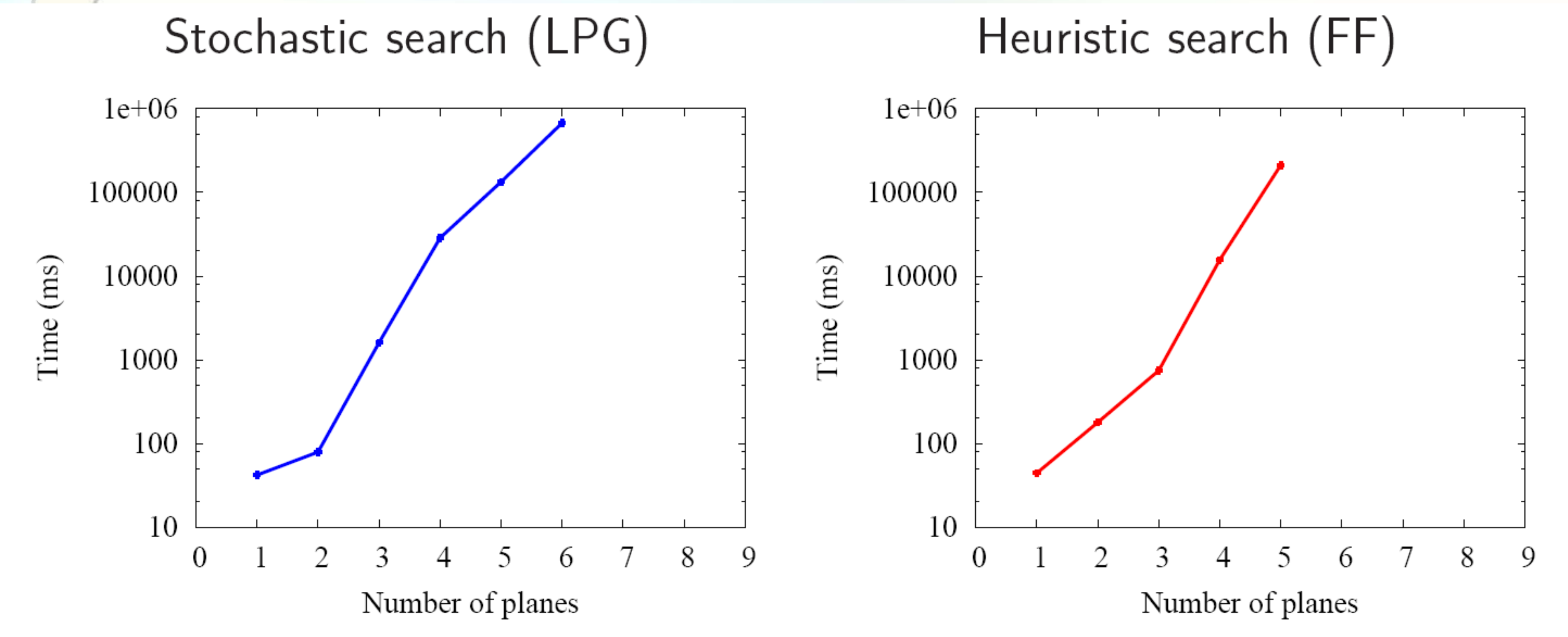
2 Subgoals (2 Planes)



3 Subgoals (3 Planes)



Exponential Complexity

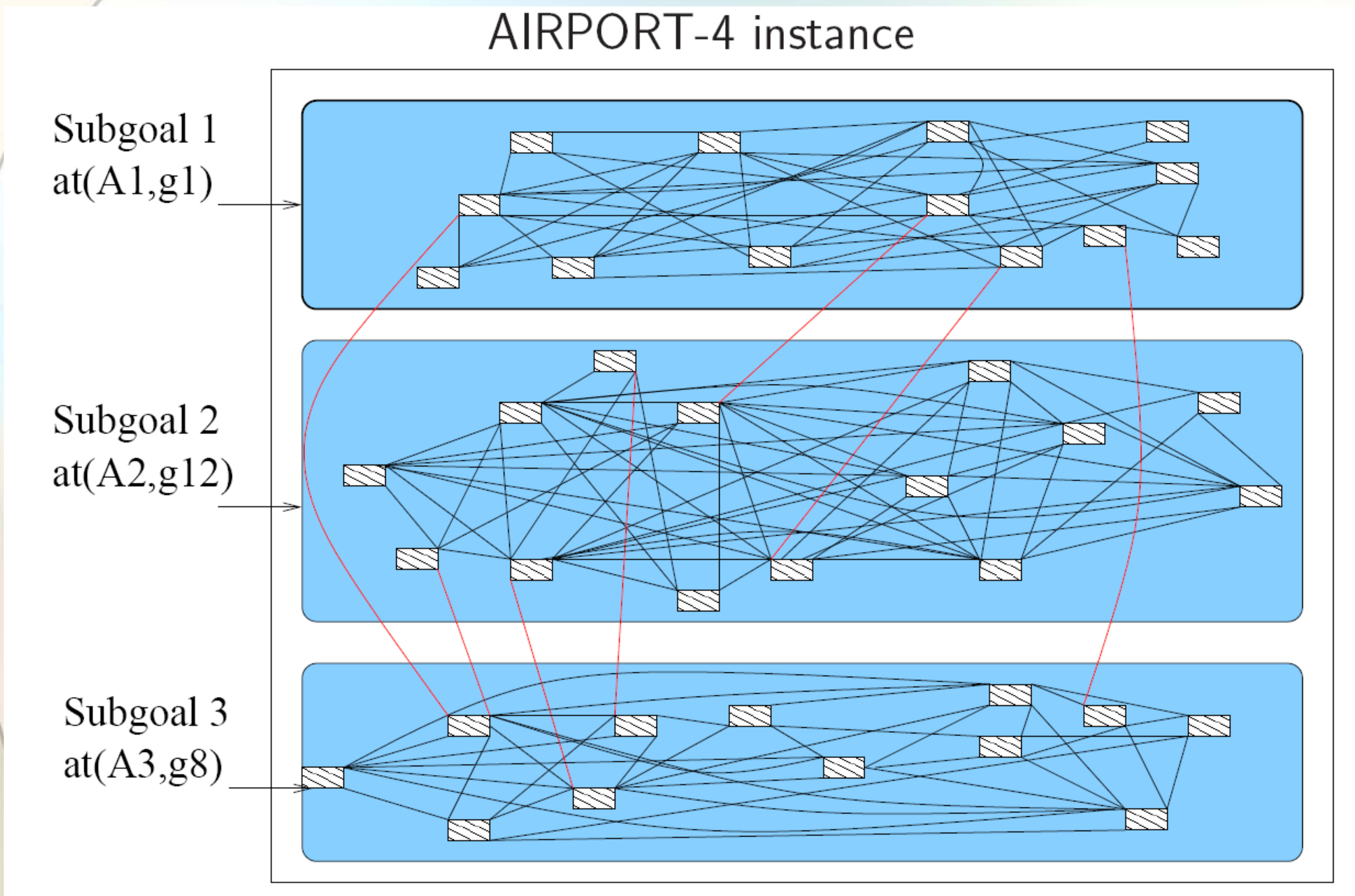


(Winners of 2nd and 3rd International Planning Competitions)

Real Constraints Are Structured

- Constraints model entities and actions with spatial 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
- A majority of the application problems encountered have structured constraints

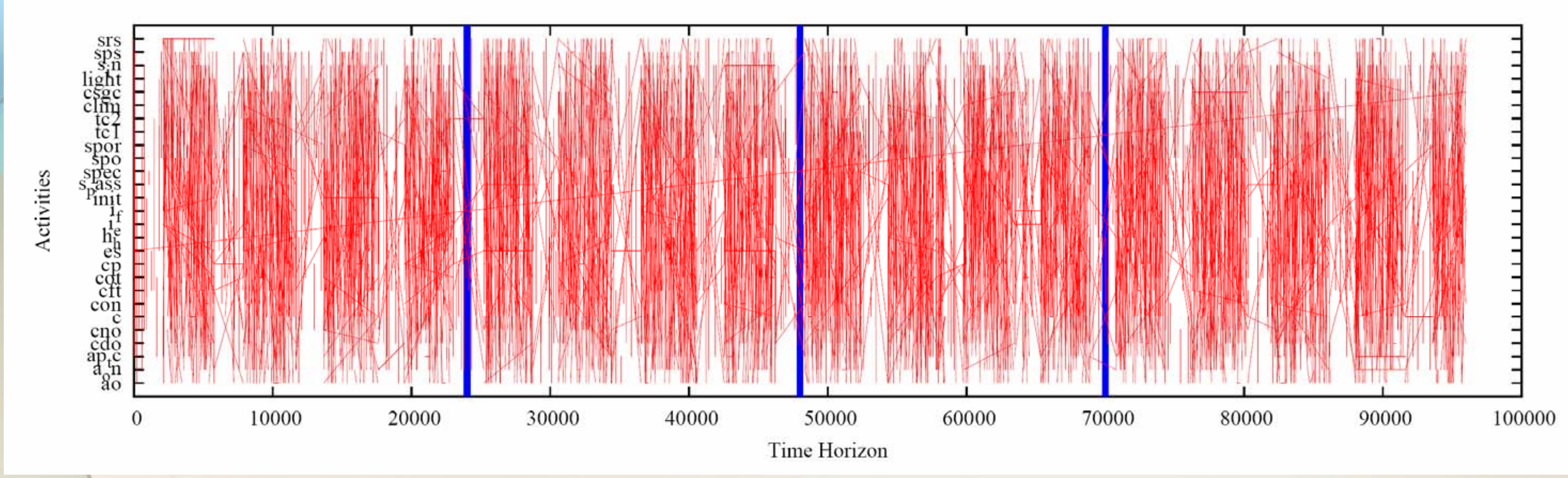
Constraint Locality in AIRPORT



Strong spatial locality of constraints

Operations Planning of CX-1 Satellite

Strong temporal locality of constraints



Measure ozone data and download to ground for analysis

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:

$$B_{min} \le \sum_{i=1}^{I} (b[i] \cdot n[i,j]) \le B_{max} \quad \text{(C1)}$$

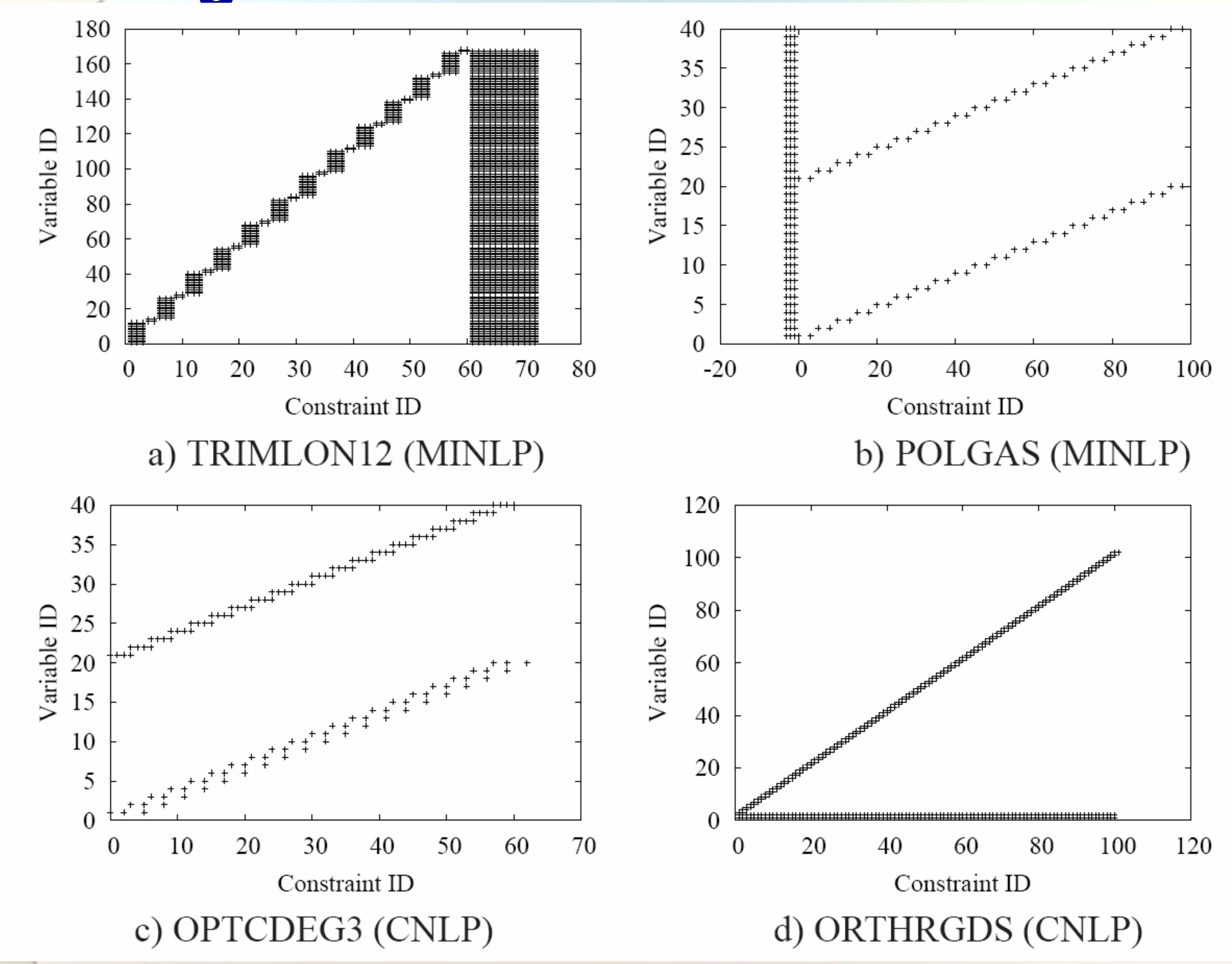
$$\sum_{i=1}^{I} n[i,j] - N_{max} \le 0 \quad \text{(C2)}$$

$$y[i] - m[j] \le 0 \quad \text{(C3)}$$

$$m[j] - M \cdot y[j] \le 0 \quad \text{(C4)}$$

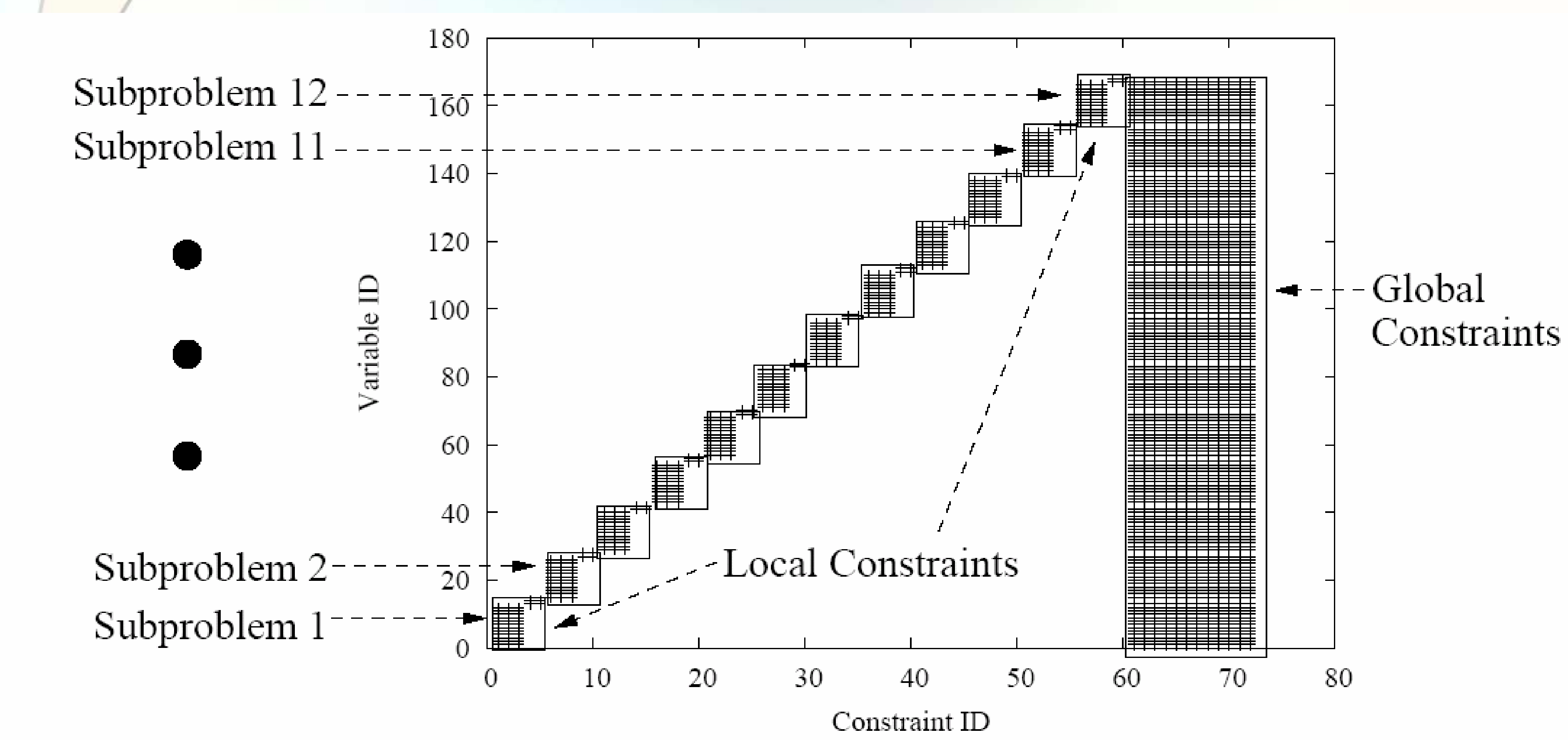
$$Nord[i] - \sum_{j=1}^{J} (m[j] \cdot n[i,j]) \le 0. \quad \text{(C5)}$$

Regular Constraint Structure



ILLINOIS

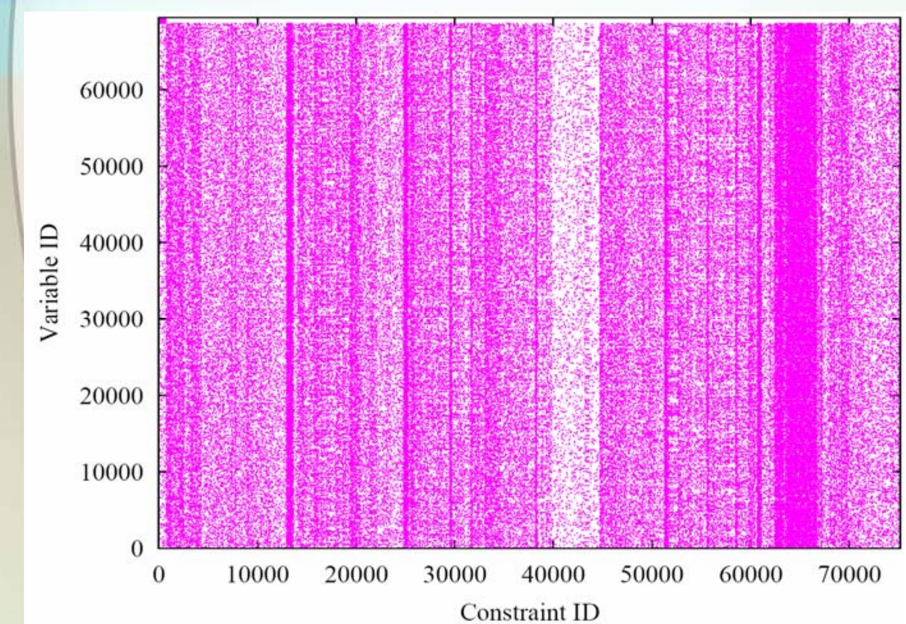
Partitioning of TRIMLON12

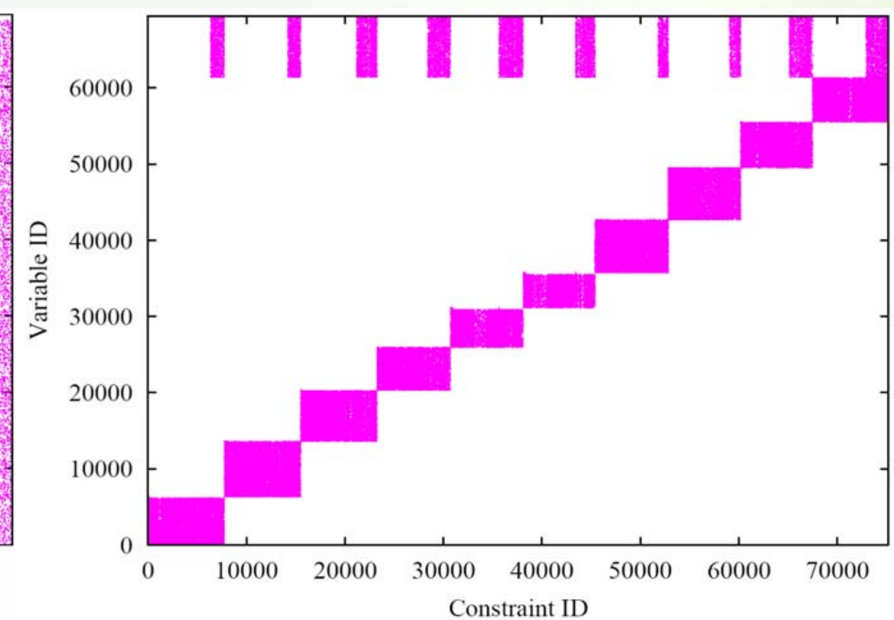


12 out of 72 constraints (16.7%) are global constraints

Standard Cell Placements

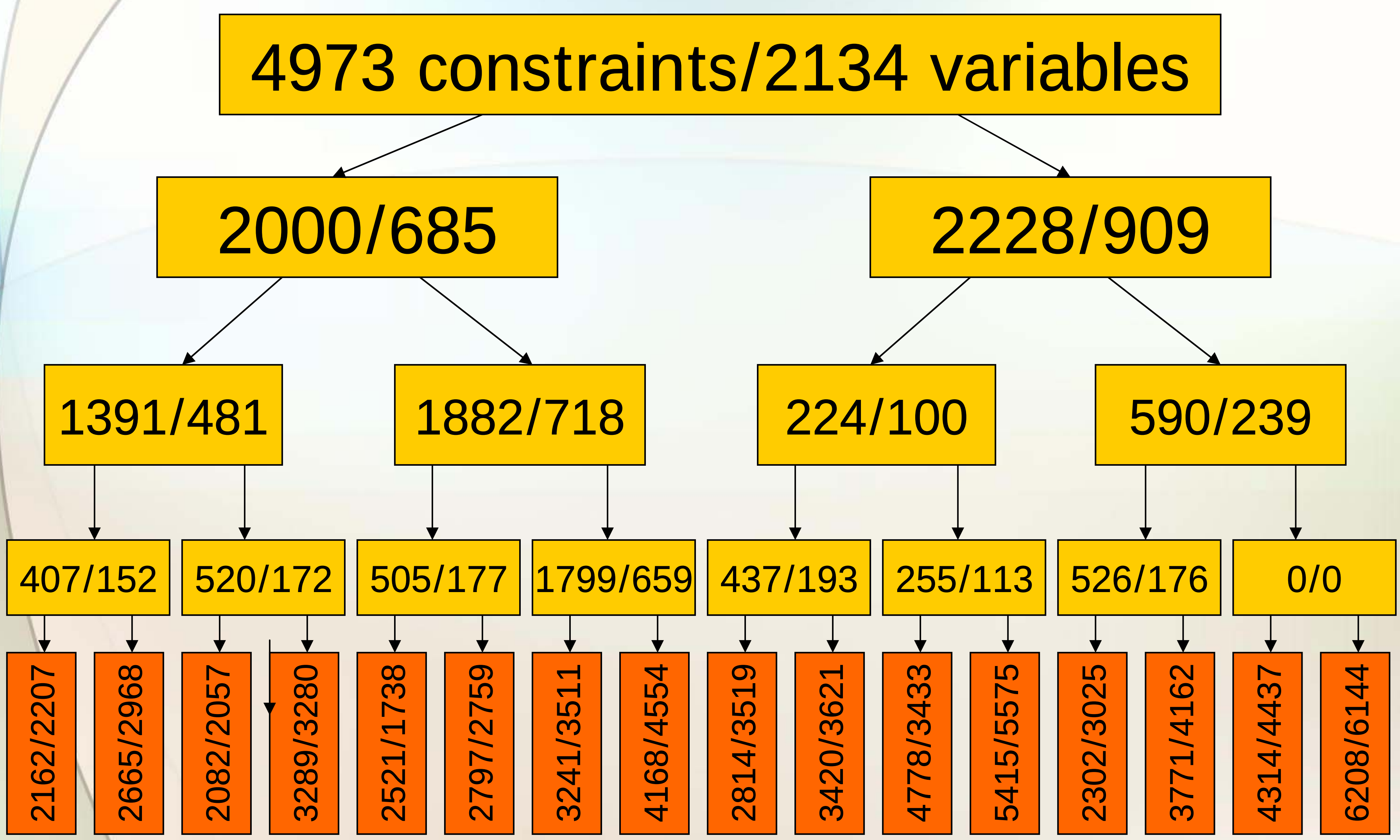
- **IBM10 benchmark with pads**
  - **69,429 cells (744 pads)**
  - **75,196 nets, 2-41 elements in each netlist**





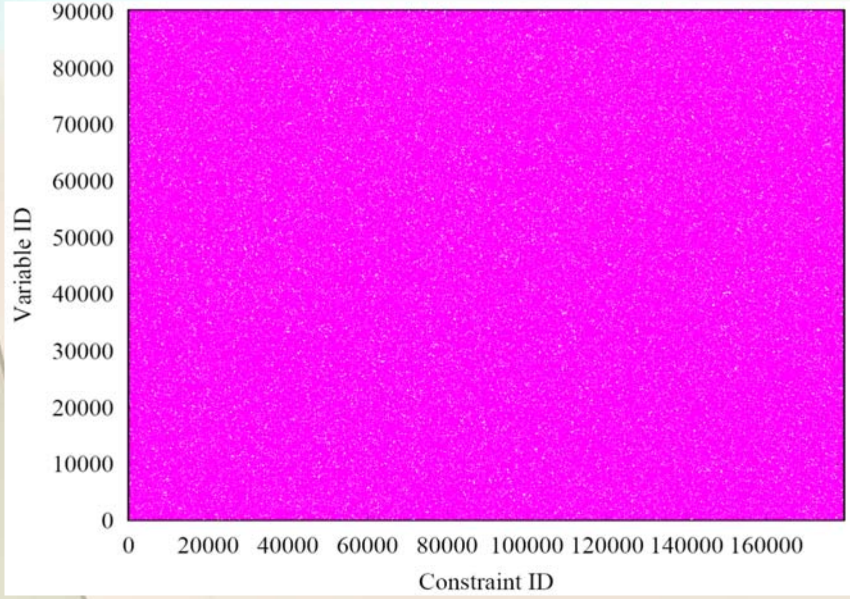
After applying METIS (Karypis et al.)

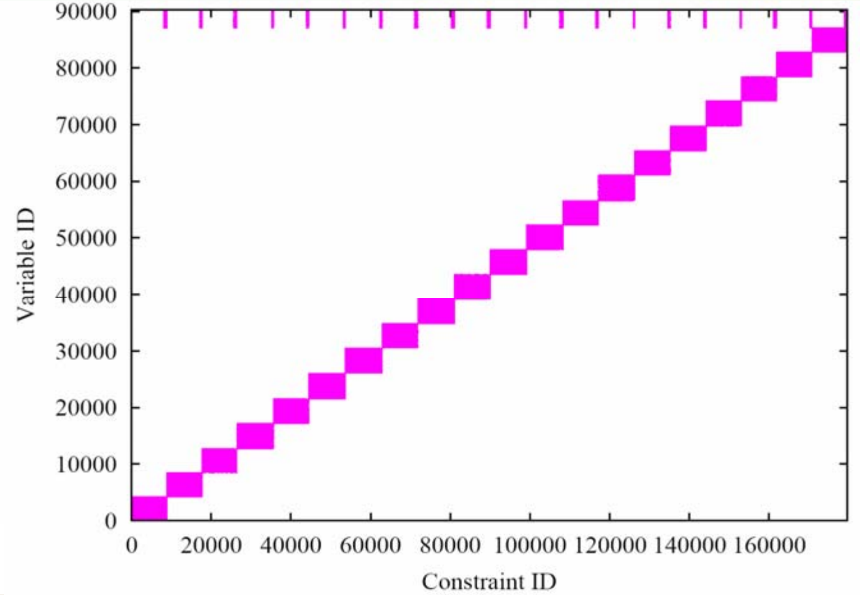
Hierarchical Decomposition of IBM10



Grid-Pebbling Problem (SAT Benchmark)

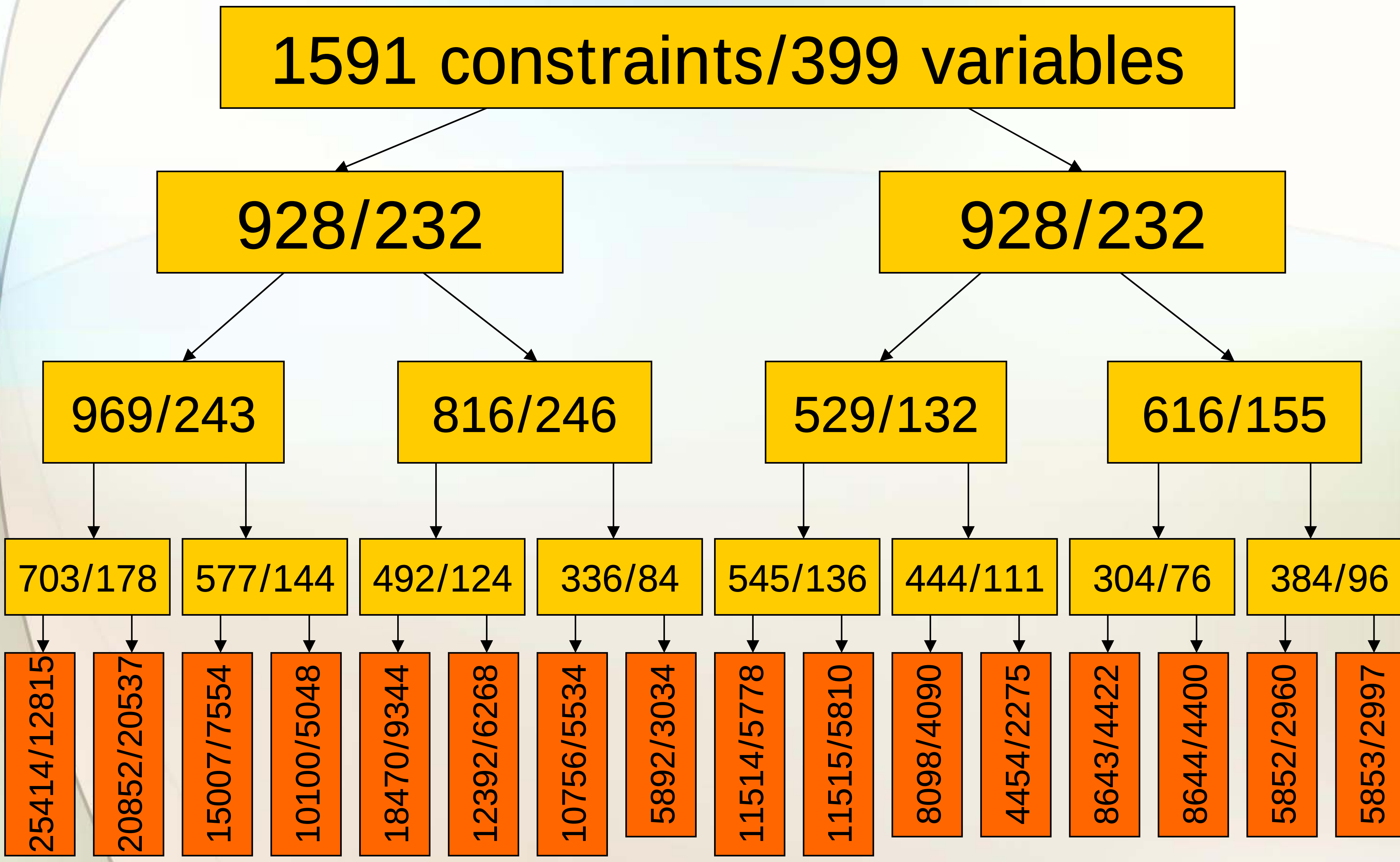
- **Scheduling precedence graphs in dependent task systems**
  - **90,300 variables, 179,701 constraints**



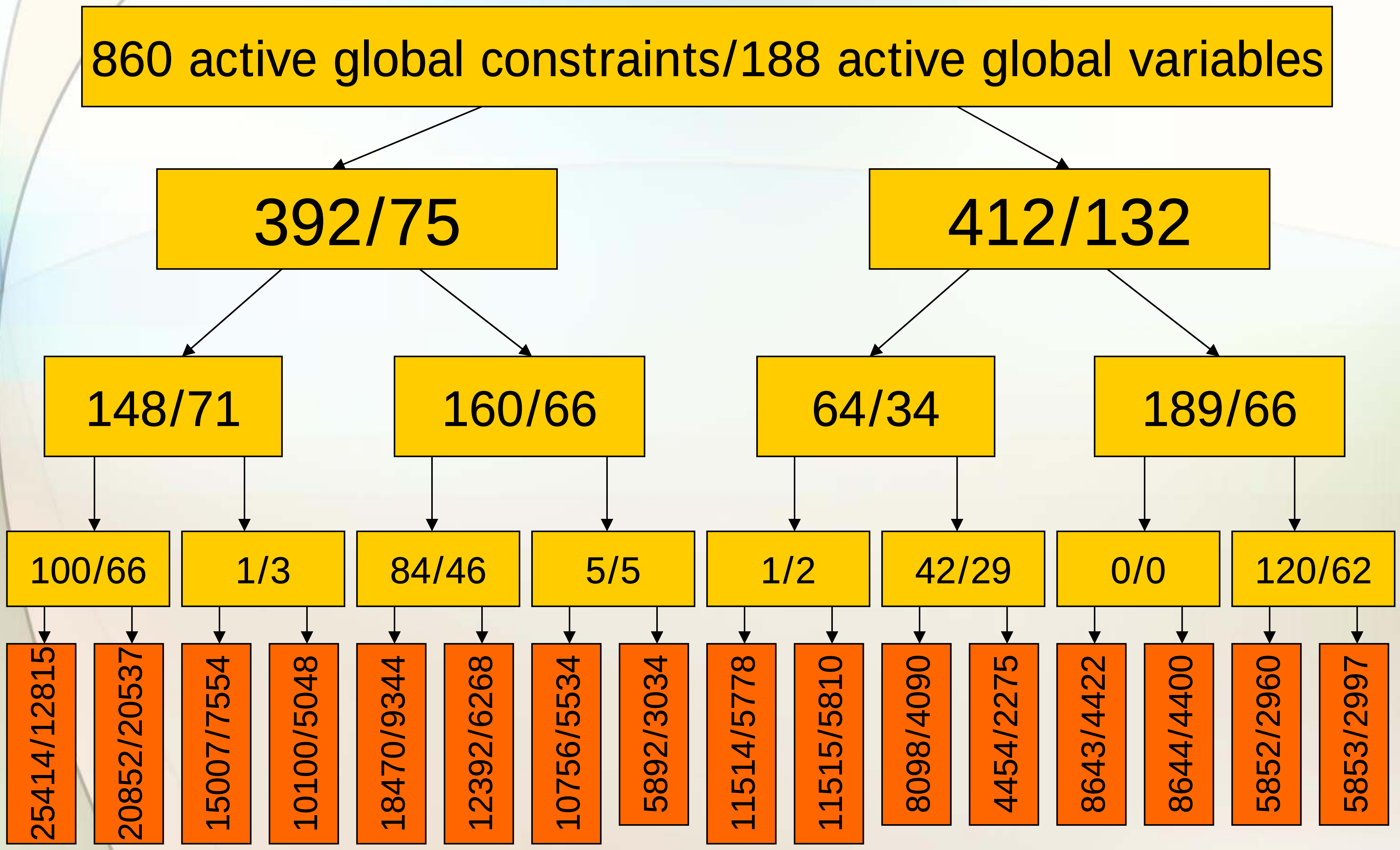


After applying METIS (Karypis et al.)

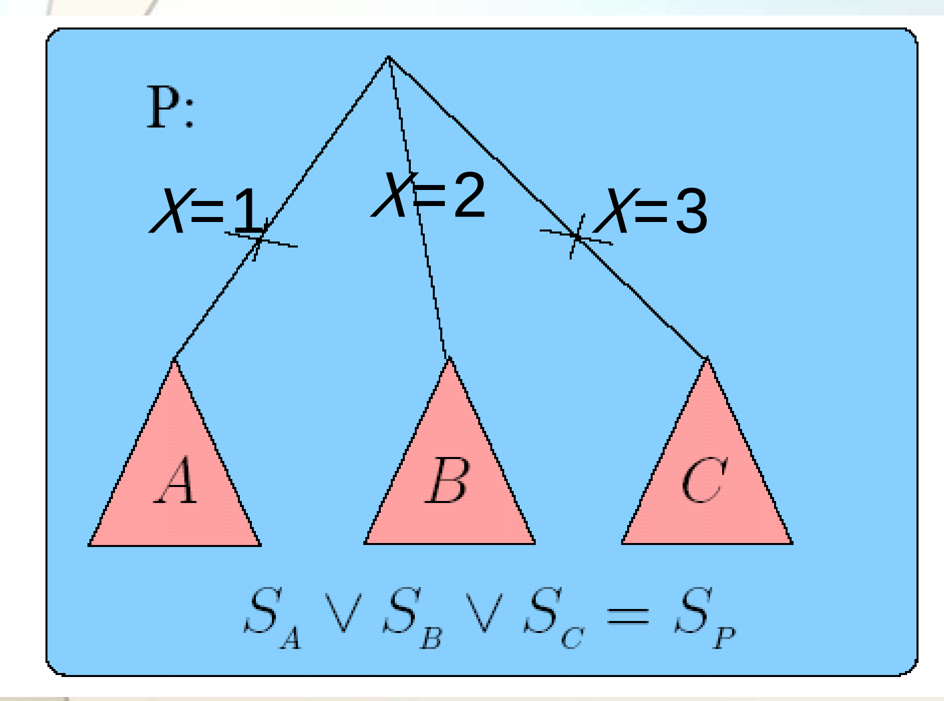
Hierarchical Decomposition of GridPbl



Small Number of Active Global Constraints



Subspace Partitioning



Subspace Partitioning

Partition **P by branching on the values of a variable**

Solve P by choosing the correct path and by solving the subproblem

Overhead for solving each subproblem *is similar* to that of P

Hierarchical Subspace Partitioning

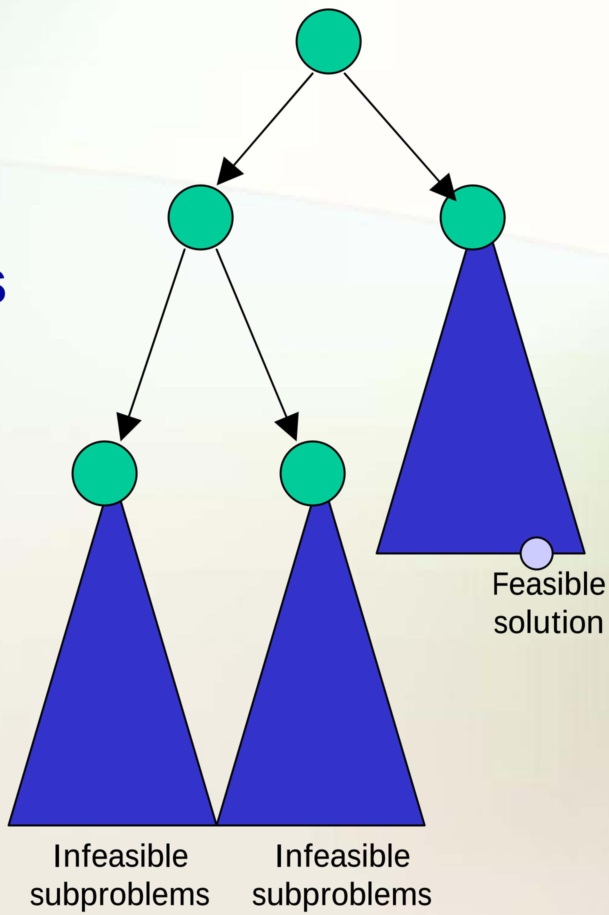
Recursively partition each subproblem by assigning values to variables (*guided* by heuristic functions)

Prune infeasible assignments by bounds or infeasibility (*easily computed*)

Backtrack to new variable assignments

Evaluate many subproblems to discover the "correct" variable assignments

Examples: B&B, B&R, GBD, OA, GCD



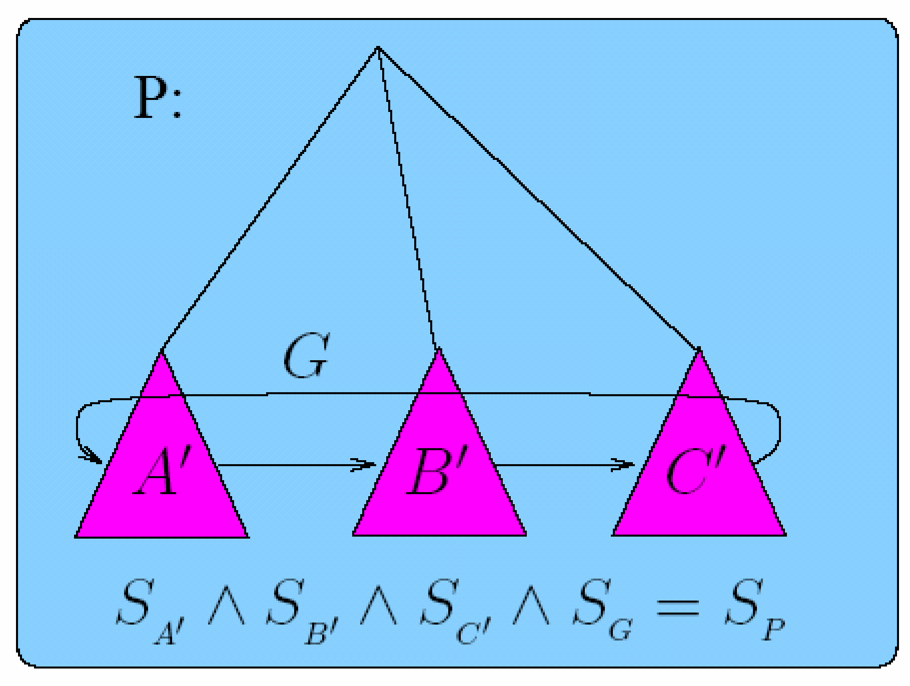
OR tree

Constraint Partitioning

Partition **P by its constraints into subproblems**

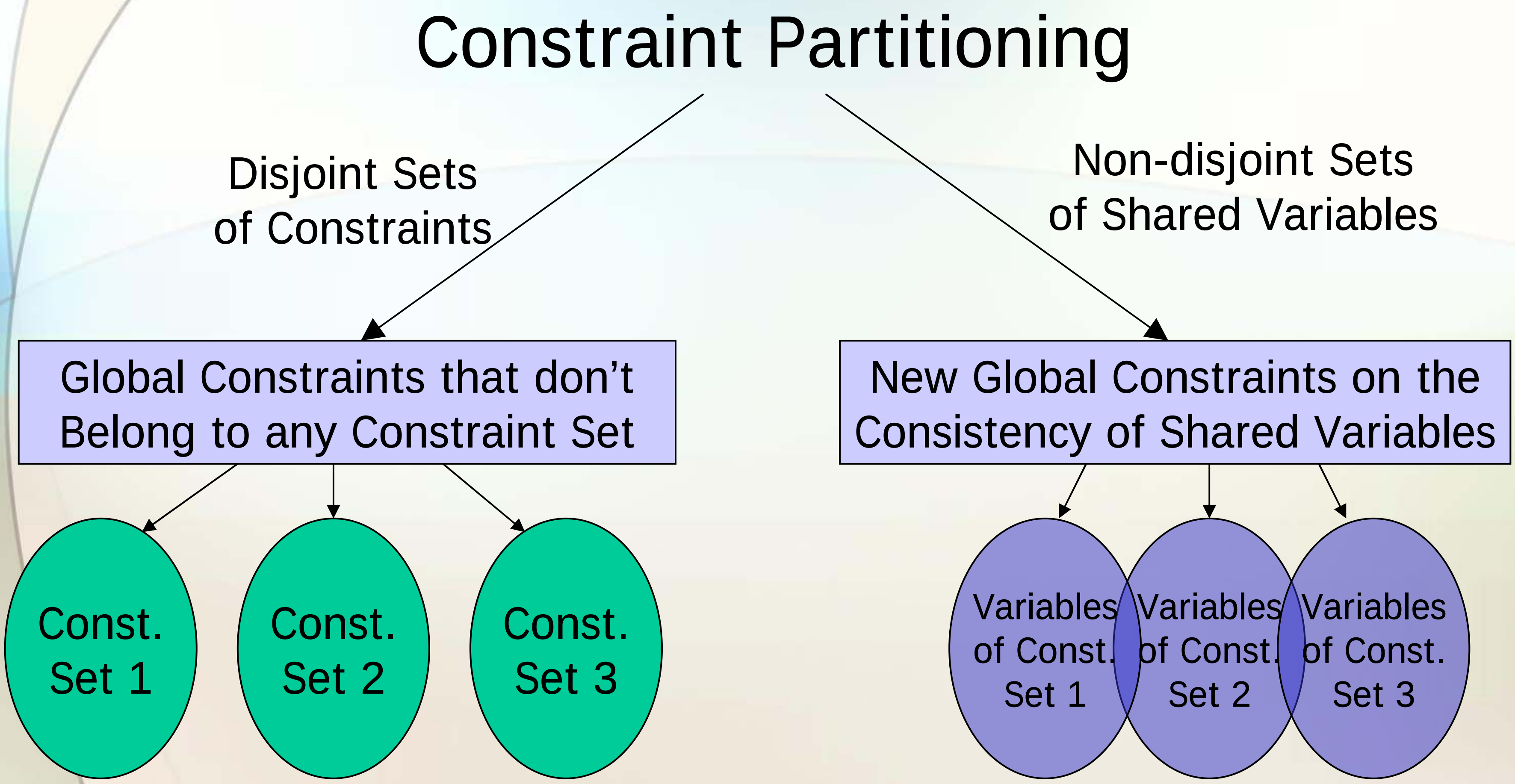
Solve P by solving all the subproblems and by resolving the inconsistent (active) global constraints

Overhead of each subproblem is exponentially smaller



Constraint Partitioning

Effects of Constraint Partitioning



Issues Addressed

- **Bottom-up resolution of violated global constraints**
- **Automated analysis of problem structure (in some standard form) and its partitioning**
- **Optimality of partitioning**
- **Demonstration of improvements over existing methods**

Previous Work: Penalty Methods

- **General penalty formulation**

$$L(variable, penalty) = objective + penalty \sum constraint\ violations$$

  - **When the penalty is large enough**
    - **Global minimum of penalty function corresponds to constrained global minimum of the original problem**
  - **Global minima of nonlinear functions are hard to find**

- **KKT: Local minimum of penalty function is a necessary condition for constrained local minimum**
  - **Differentiability and continuity requirements**
  - **System of nonlinear equations that cannot be partitioned**

Theory of Extended Saddle Points

- Necessary and sufficient condition of penalty formulations governing constrained local minima [AI06]
  - 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

Partition and Resolve Framework

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

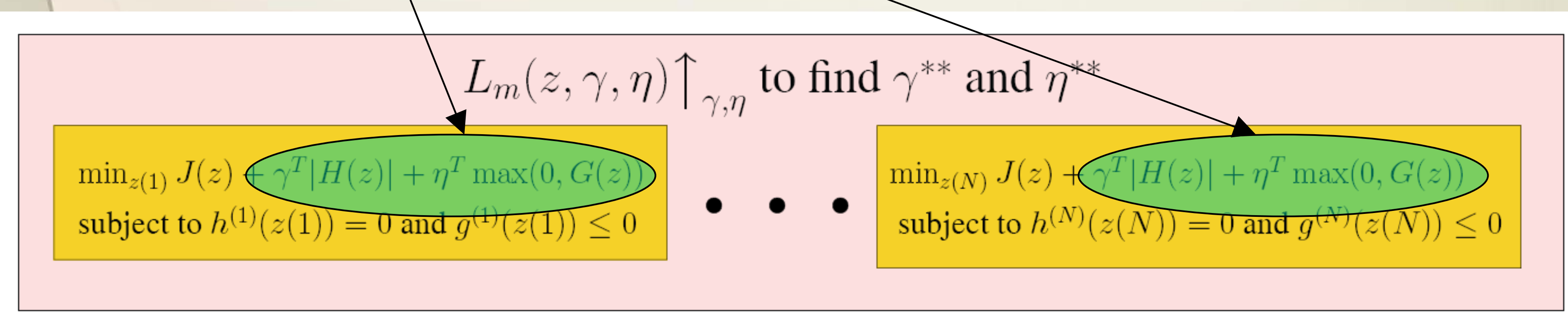
$L_m(z, \gamma, \eta) \uparrow_{\gamma,\eta}$ to find γ^{**} and η^{**}

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

Partition and Resolve Framework

Weighted active global constraints provide guidance in local subproblems

- Solving a subproblem
  - Satisfy local constraints
  - Minimize global objective
  - Minimize global constraint violations
- Increasing penalties on violated global constraints



Issues Addressed

- Resolution of violated global constraints
- **Automated analysis of problem structure (in some standard form) and its partitioning**
- Optimality of partitioning
- Demonstration of improvements over existing methods

Implementation of P&R Framework

procedure CPOPT
  call *automated_partition()*; // automatically partition the problem //
  $\gamma \longleftarrow \gamma_0$; $\eta \longleftarrow \eta_0$; // initialize penalty values for global constraints//
  repeat // outer loop //
    for $t = 1$ **to** N // iterate over all N stages to solve $P_t^{(t)}$ in stage t //
      apply an existing solver to solve $P_t^{(t)}$;
      call *update_penalty()*; // update penalties of violated global constraints //
    end_for;
  until stopping condition is satisfied;
end_procedure

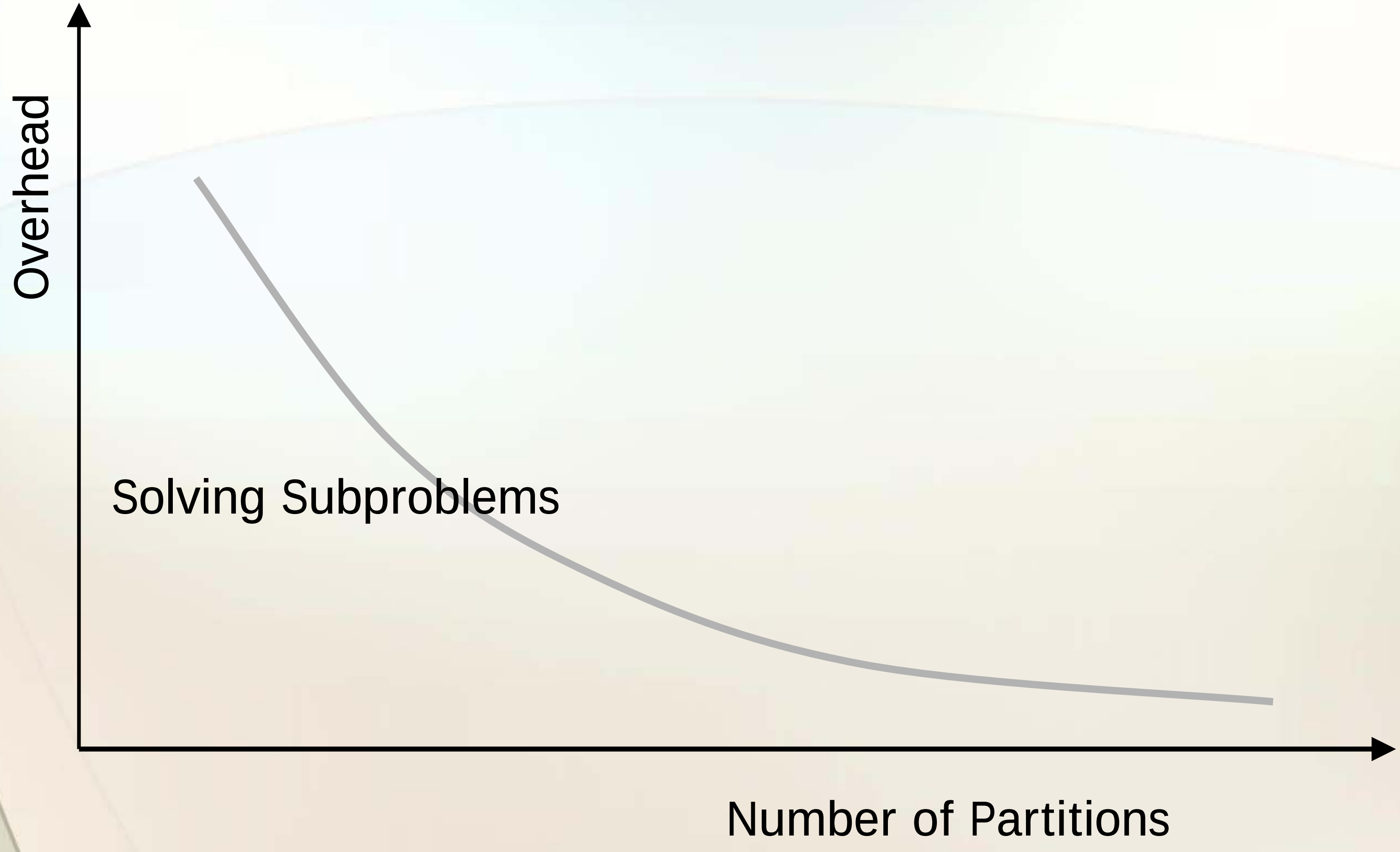
$$\min_{z(t)} \quad J(z) + \gamma^T a + \eta^T b$$

$$\text{subject to} \quad h^{(t)}(z(t)) = 0 \quad \text{and} \quad g^{(t)}(z(t)) \le 0,$$
$$-a \le H(z) \le a \quad \text{and} \quad G(z) \le b,$$

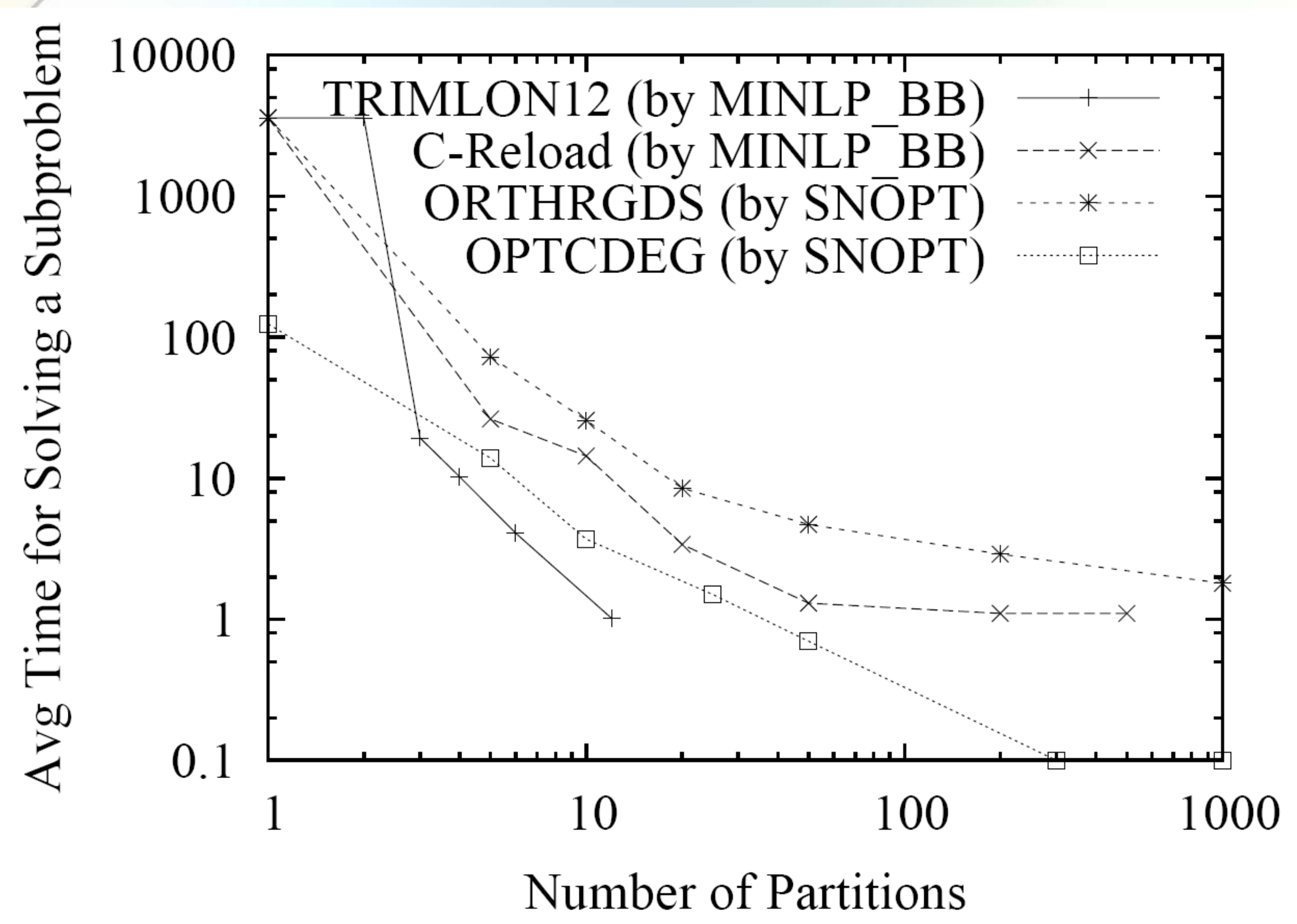
Issues Addressed

- 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

Trade-offs in Constraint Partitioning

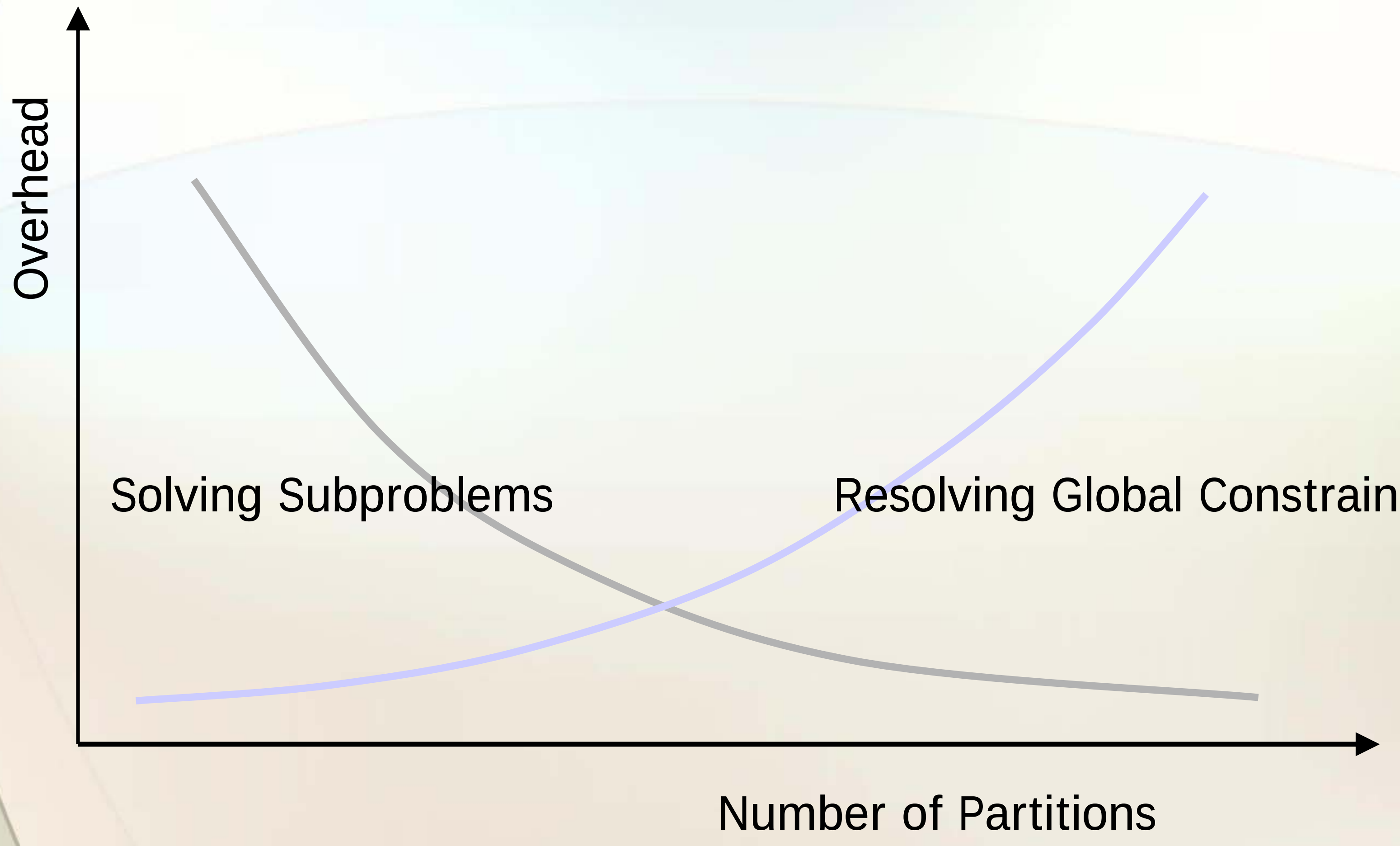


Substantial Decrease in Subproblem Time

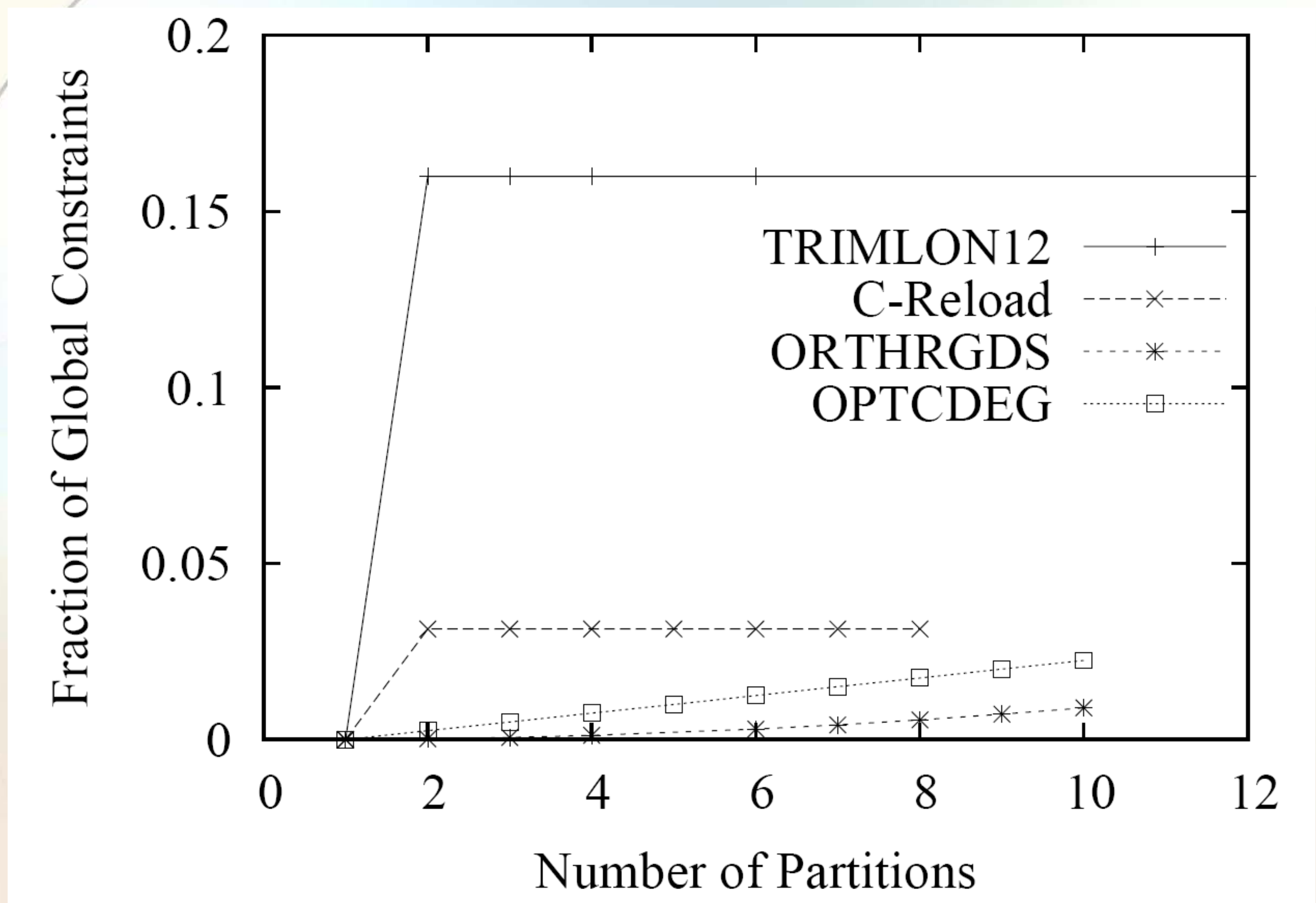


Substantial decrease in overhead as number of constraints in a subproblem is reduced

Trade-offs in Constraint Partitioning

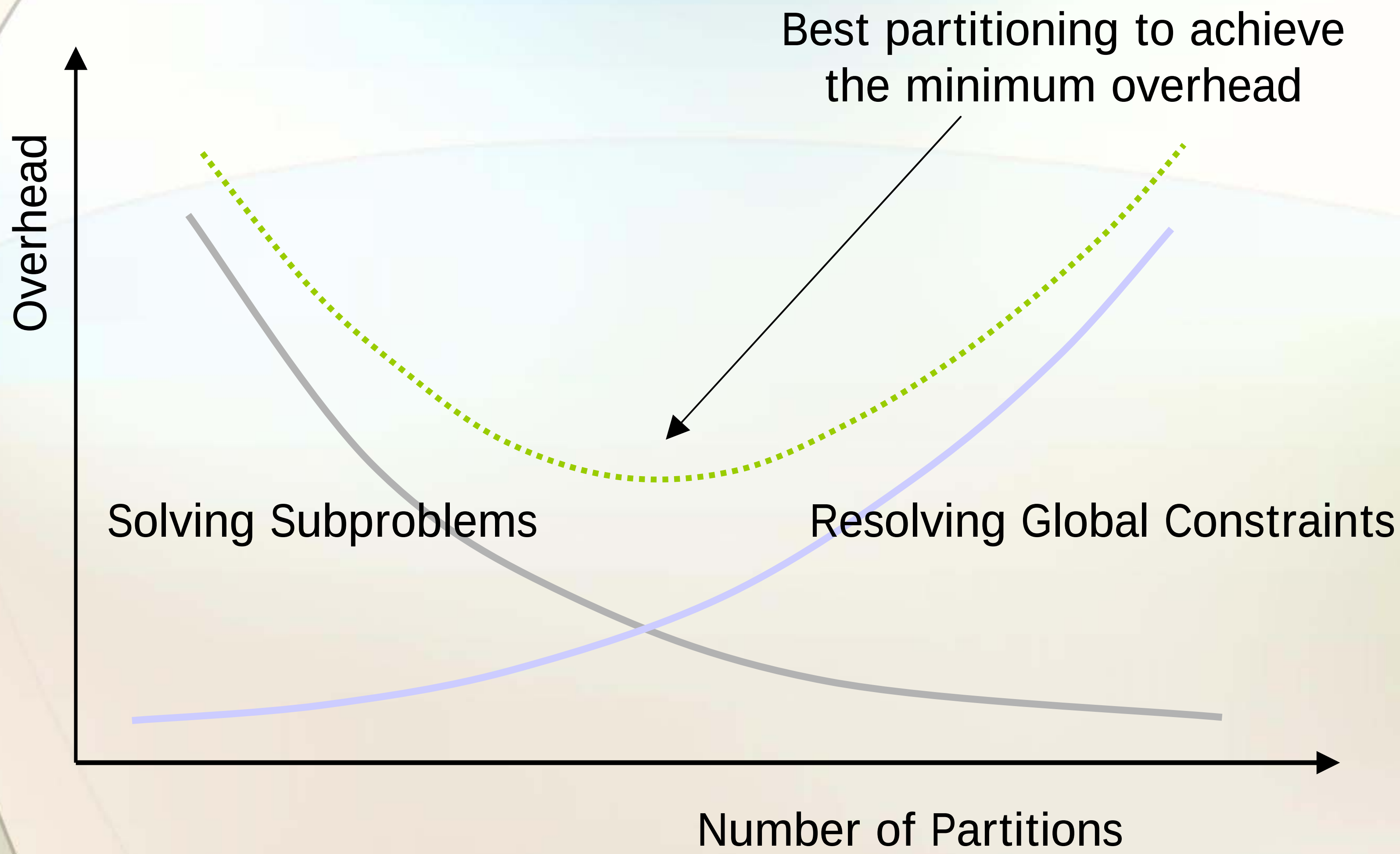


Monotonic Increase in Global Constraints



More overhead for resolving global constraints as number of partitions increases

Optimal Number of Partitions



Convex relationship between no. of partitions and total time

An Illustration

- **Exploit convex relationship between *N* and total time**

Total time spent
= 2.6 + 2.7 + 2.8 + 3.1
+ 3.3 + 8.4 + 99 = 121.9

Start from large *N*

Estimate time per iteration

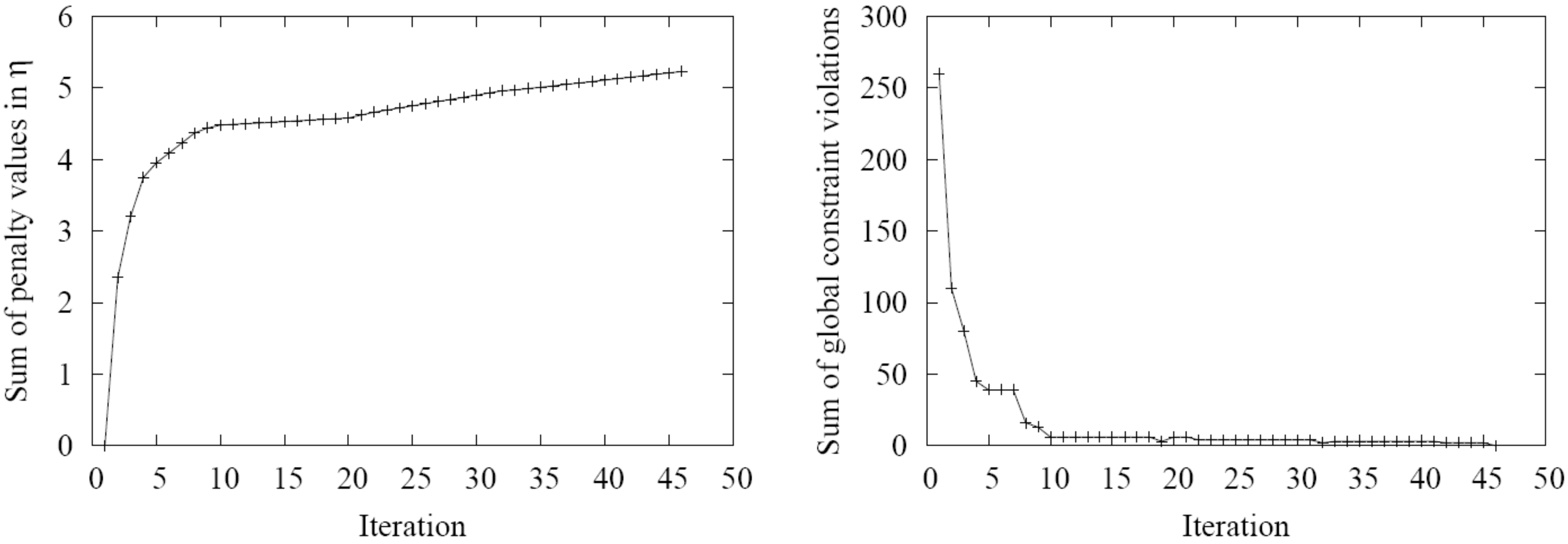
Assume that global constraints can be resolved quickly

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	3.3	3.1	2.8	2.7	2.6
Time per iteration	>3600	126	99	186	336	648	1248
Number of iterations	1	1	1	2	2	2	5
Total time to solve problem	>3600	126	99	372	672	1296	6240

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
- **Demonstration of improvements over existing methods**
  - **Temporal planning**
  - **Nonlinear constrained optimization**
  - **Neural network learning**

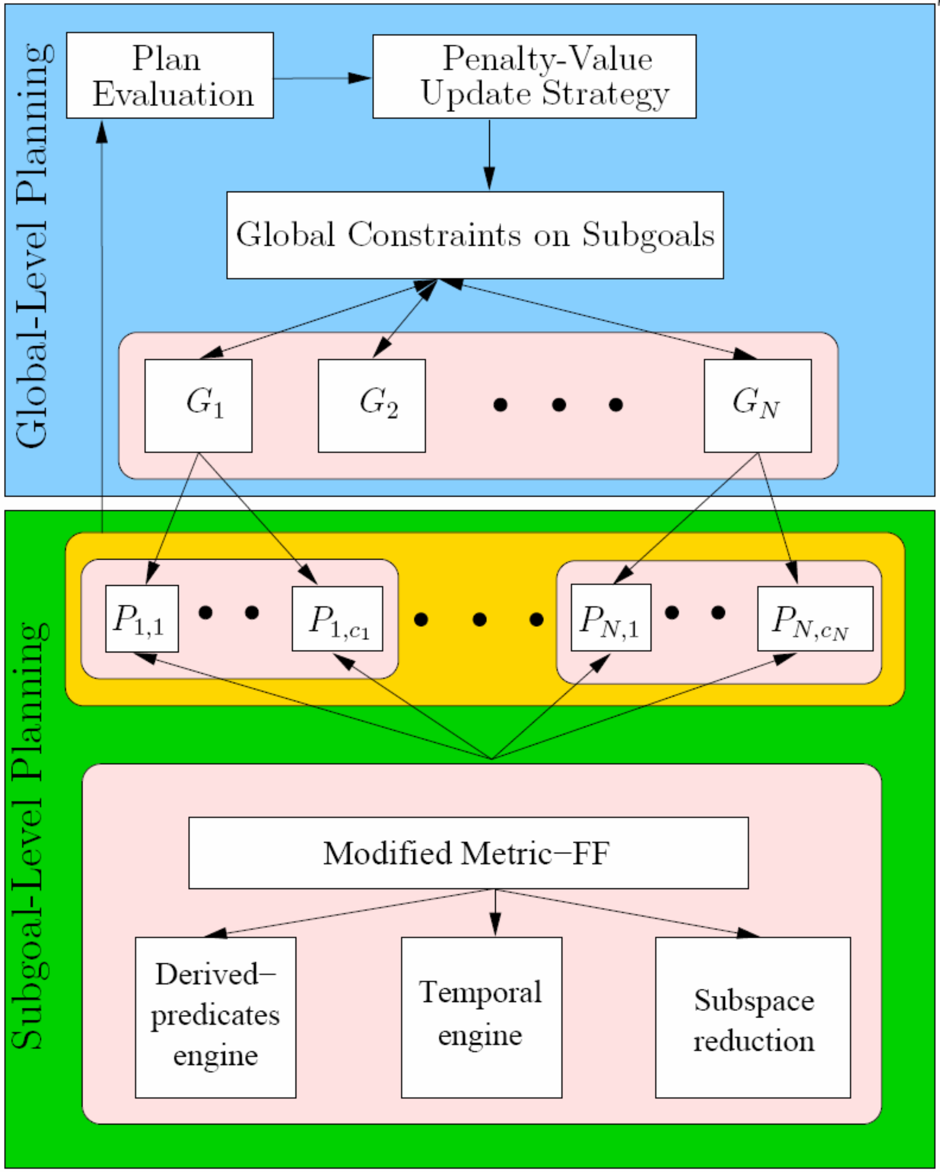
4^{th} Int'l Planning Competition (2004)

- Biennial competition since 1998
- Real-world application domains
  - Airport scheduling
  - Petroleum transportation
  - Communication channel scheduling
  - Power supply restoration
  - Satellite operations
  - Mobile communications
- Format of competition
  - Over a period of 8 weeks, starting March 2004
  - One application (multiple domains and multiple instances) each week over 7 weeks
  - Each instance limited to 1 GB memory and 30 minutes on a Linux computer
  - Planner designed to run on all instances with no human intervention

Participants in the Classic Part

- Heuristic search
  - Macro-FF, FAP, Marvin, Crikey, TP4, Downward, SGPlan, Diagonal-Downward, Tilsapa, Optop, P-MEP, YAHSP
- Transformation methods
  - Optiplan, Petriplan, SATPLAN
- Systematic search
  - Semsyn, CPT, BFHSP
- Local search
  - LPG-TD

Architecture of SGPlan [JAIR'06]



Techniques Studied

Global-Constraint Resolution

Producible Resources

Constraint Partitioning by Subgoals

Landmark Analysis

Temporal Engine

Derived-Predicates Engine

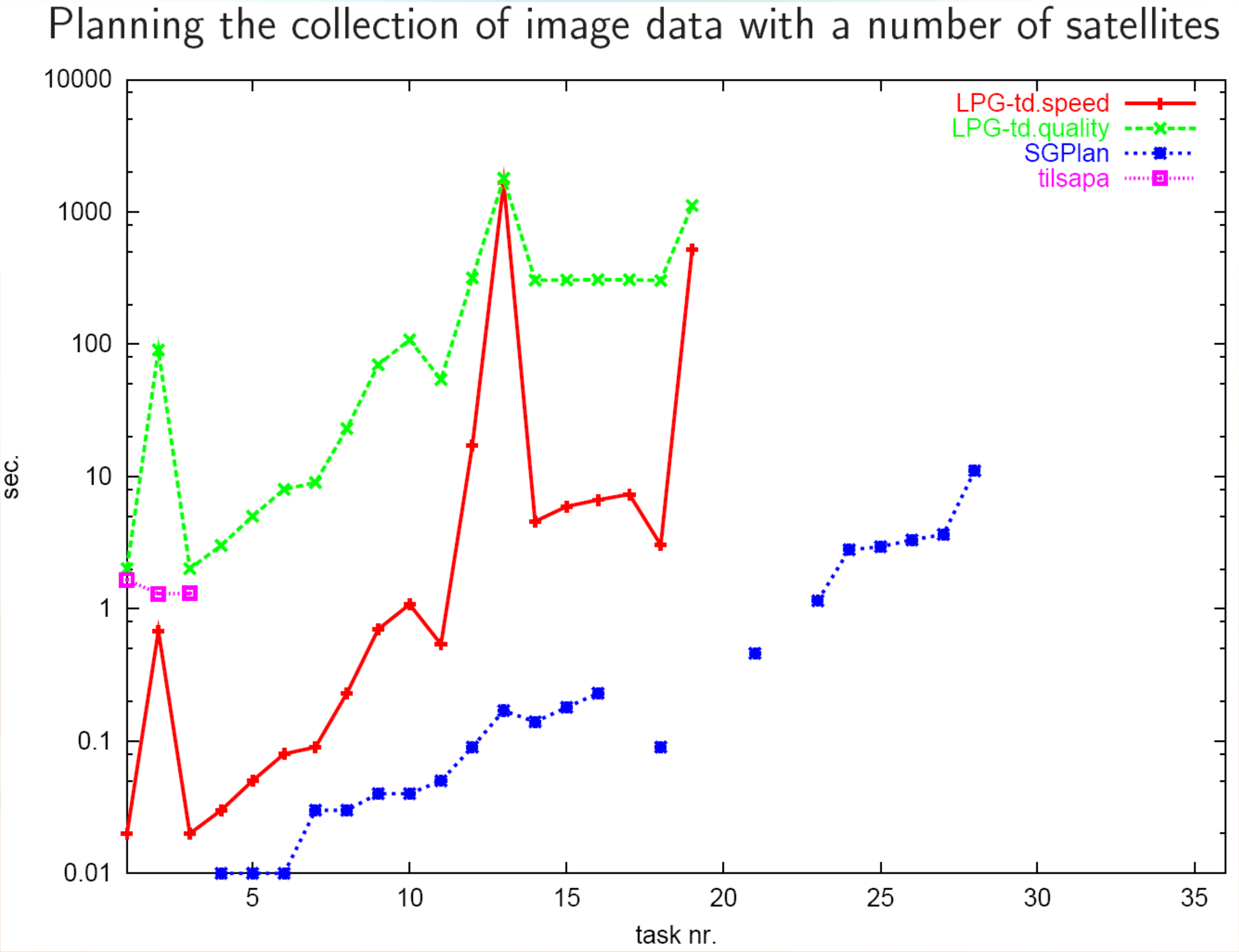
Search-Space Reduction

IPC4 Results

- **SGPlan was the only planner that won in two tracks**
  - **Suboptimal Temporal Metric Track: 1st Prize**
  - **Suboptimal Propositional Track: 2nd Prize**
  - **Optimal Track: did not participate**

Domain	Total	**SGPlan**	LPG	Downward	Macro-FF	YAHSP	Crikey
Airport	200	**155**	134	50	21	36	64
Pipesworld	260	**166**	113	60	62	93	111
Promela	272	**167**	83	83	38	42	13
PSR	200	**122**	99	131	32	48	29
Satellite	288	**207**	157	36	36	-	-
Settlers	20	**19**	13	-	-	-	-
UMTS	300	**274**	200	-	-	-	-
Overall	1540	**1110**	799	360	189	219	217

IPC4 Results: SATELLITE Domain



Difficult-to-Solve MINLP [CP'05]

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

Difficult-to-Solve NLP [CP'05]

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

Conclusions

- Constraint partitioning is a powerful approach for exploiting constraint structure in order to reduce complexity
  - Bottom-up resolution with guidance provided by top-level active global constraints
  - Using existing solvers to solve partitioned subproblems
  - Hierarchical partitioning is critical

Acknowledgments

- Y. X. Chen, Washington University, St. Louis
- C. W. Hsu, U. of Illinois, Urbana-Champaign
- S. M. Lee, U. of Illinois, Urbana-Champaign
- M. Richards, U. of Illinois, Urbana-Champaign
- Y. Shang, U. of Missouri, Columbia
- T. Wang, Synopsis
- Z. Wu, Oracle