Constraint Partitioning for Solving Planning Problems with Trajectory Constraints and Goal Preferences

Chih-Wei Hsu, Benjamin W. Wah, Ruoyun Huang, Yixin Chen

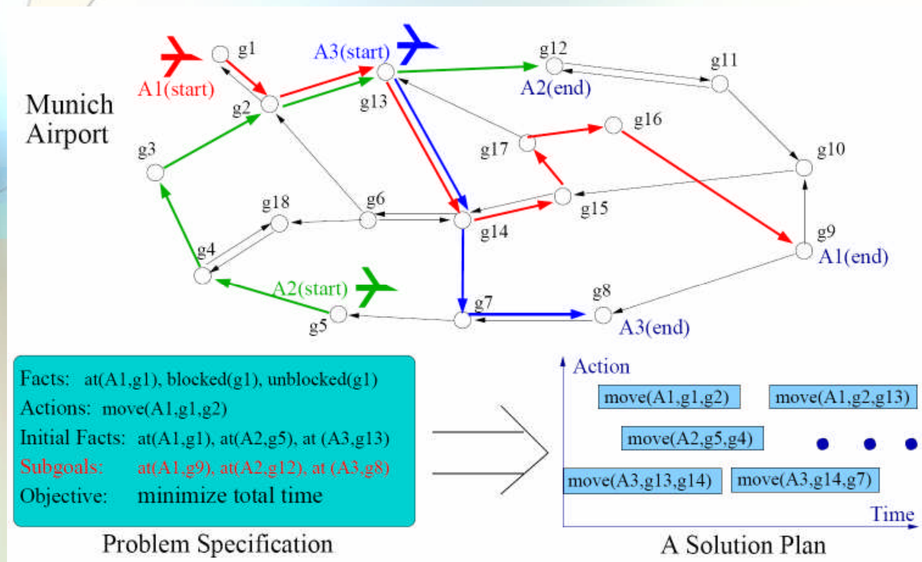
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
URL: http://manip.crhc.uiuc.edu

IJCAI-07

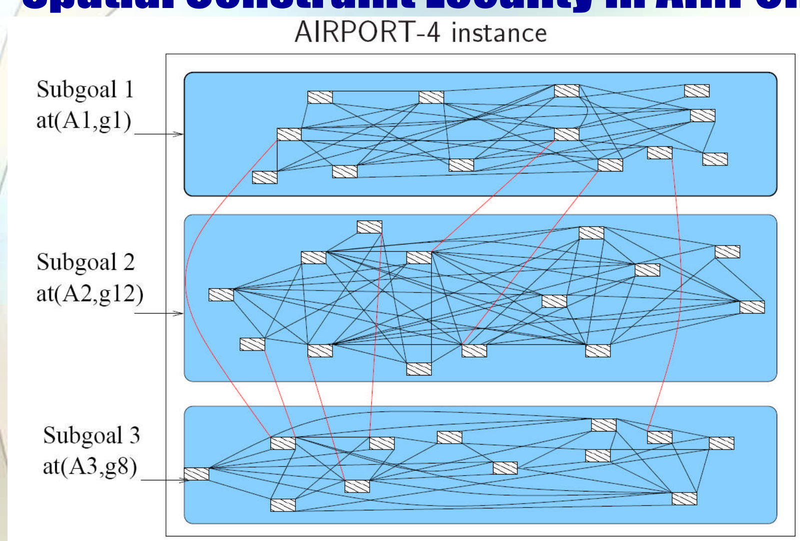
Outline

- Observation on constraint locality
- Partition-and-resolve approach
  - Partition a problem by its constraints into subproblems
- Observation in PDDL3 domains
- Architectures of $SGPlan_5$
  - Handling soft goals
  - Handling trajectory constraints
  - Subproblem-level planning
- Conclusions

An Airport Planning Example



Spatial Constraint Locality in AIRPORT

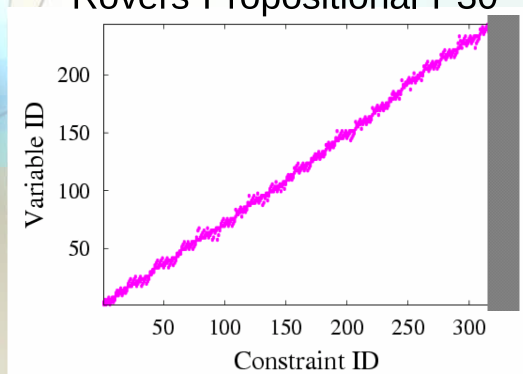


Mutual exclusion between 2 subgoals:
2 planes cannot be on the same runway at the same time

Depicting Constraint Localities

Problem Formulation

Rovers-Propositional-P30



Constraints: mutual exclusions, trajectory constraints and goals
Variables: initial solution plan
Global constraints: constraints span across multiple subproblems

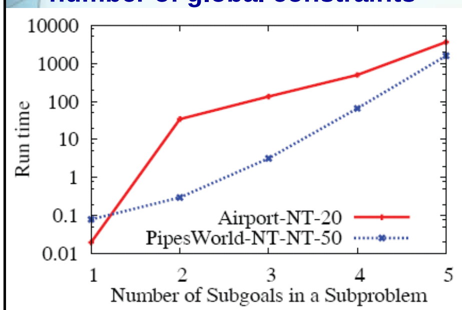
Attribute used for partitioning: **subgoals**

Outline

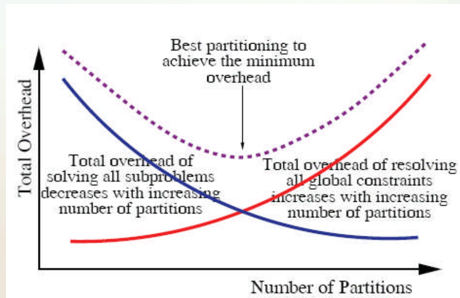
- Observation on constraint locality
- **Partition-and-resolve approach**
  - **Partition a problem by its <u>constraints</u> into subproblems**
- Observation in PDDL3 domains
- Architectures of SGPlan$_5$
  - Handling soft goals
  - Handling trajectory constraints
  - Subproblem-level planning
- Conclusions

Motivations of Approach

- Significantly smaller aggregate complexity of solving all subproblems
- Large reduction in solution time if global constraints can be resolved quickly
- Study of constraint locality and partitioning granularity to reduce the number of global constraints

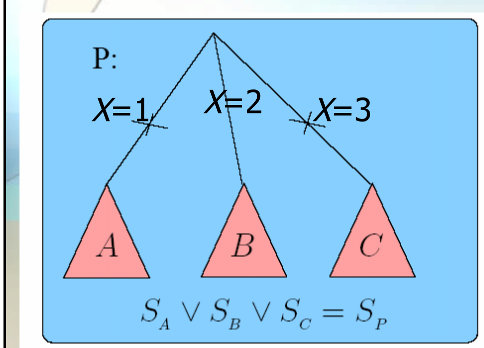


Exponential growth in solution time



Convex relationship between no. of partitions and total time

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

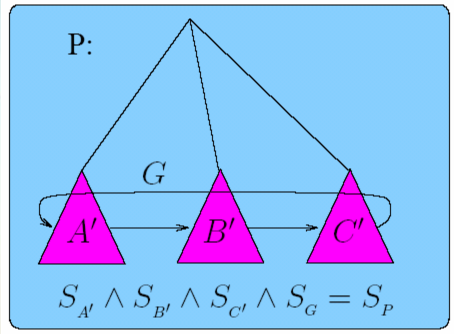
Overhead for solving each subproblem *is similar* to that of P

Constraint Partitioning

Partition **P by its constraints into subproblems**

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

Overhead of each subproblem is substantially smaller



Constraint Partitioning

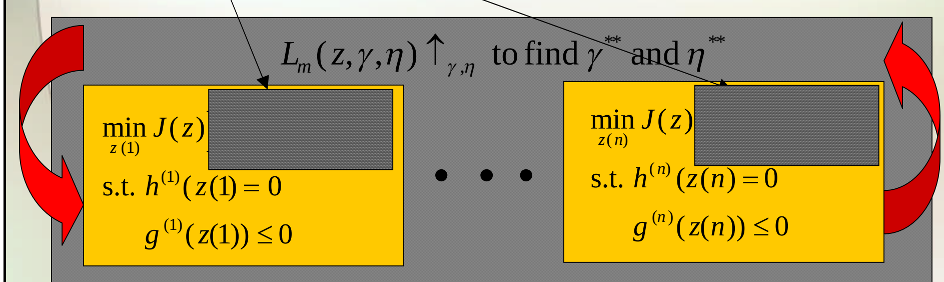
Each subproblem is significantly relaxed with a much larger solution space

Our Partition-and-Resolve Framework

Weighted active global constraints provide guidance in local subproblems

Similar solver as original problem

- **Solving a subproblem**
  - **Satisfy local constraints**
  - **Minimize global objective**
  - **Satisfy (soft) global constraints**
- **Increasing penalties on violated global constraints**



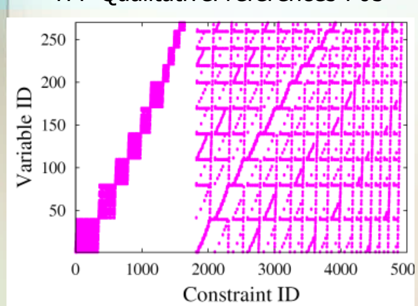
Outline

- Observation on constraint locality
- Partition-and-resolve approach
  - Partition a problem by its constraints into subproblems
- **Observation in PDDL3 domains**
- Architectures of $SGPlan_5$
  - Handling soft goals
  - Handling trajectory constraints
  - Subproblem-level planning
- Conclusions

Constraint Locality in PDDL3

- Planning benchmarks in PDDL 3.0
  - New features over PDDL 2.2
    - Soft goals or preferences on action preconditions
    - Hard and soft constraints on plan trajectory
- Different locality of constraints caused by new features

TPP-QualitativePreferences-P05



- Poor locality by subgoal partitioning
  - Many global constraints due to implicit mutual exclusions
  - Trajectory constraints not considered in partitioning

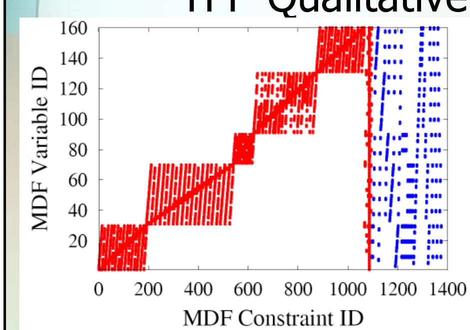
Multi-Valued Domain Formulation

- Multi-valued state variables
  - Traditionally encoded in multiple binary facts
  - Now encode in one variable that can take multiple values
- Example
  - AT(TRUCK1, LOCATION1), …, AT(TRUCK1, LOCATION8) → LOC(TRUCK1) = LOC1, …, LOC(TRUCK1) = LOC8
- More compact representation than STRIPS
- New problem representation for reducing the number of global constraints

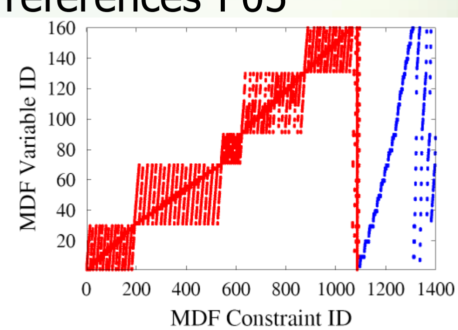
Constraint Locality by Other Variables

- Guidance variable: variable in goal-state constraint
  - Ex: Stored quantity of a product
- Cluster constraints by guidance variables

TPP-QualitativePreferences-P05



Clustering by hard constraints



Clustering by hard/soft constraints

Granularity in Partitioning

- **Bottleneck variable: variable in a group that other variables depend on**
  - **Ex: Position of Truck in TPP; Position of Hoist in Storage**

Table 1: Trade-offs on the number of partitions for the *Trucks-TimeConstraints-20* instance.

Partitioning Strategy	No Partitioning	Bottleneck	Guidance
# partitions	1	4	22
# global constraints	0	20	573
avg. # local const. per subproblem	21274	1404	230.1
time/subproblem	>1800 sec.	2.04 sec.	0.16 sec.

Bottleneck: # partitions = min(# bottleneck var., # guidance var.)
Guidance: # partitions = # of guidance variables
Subgoal: subgoal partitioning

Summary of Partitioning Approach

- **Problem representation: MDF**
- **Partitioning attributes: guidance variables**
- **# partitions = min(# guidance var., # bottleneck var.)**
- **Cluster guidance variables using information of constraints/preferences**

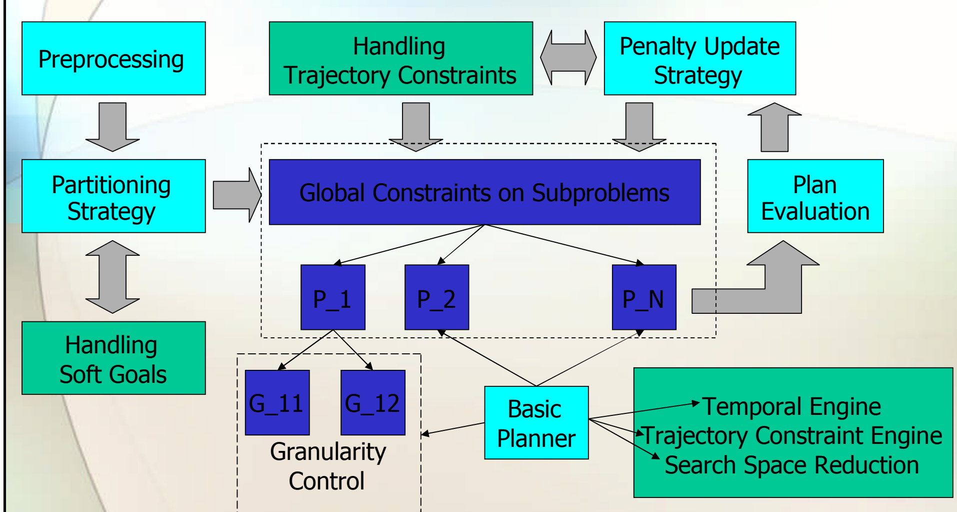
Table 2: Average fraction of constraints that are active global constraints initially across all the instances of each IPC5 *QualitativePreferences* domain under three partitioning strategies. (See keys in Table 1.)

Domain	Bottleneck	Guidance	Subgoal
TPP	0.0153	0.0733	0.2892
OpenStacks	–	0.1728	0.1834
Trucks	–	0.0749	0.1623
Storage	0.0007	0.0032	0.0109
Rovers	0.0227	0.0464	0.0488

Outline

- Observation on constraint locality
- Partition-and-resolve approach
  - Partition a problem by its constraints into subproblems
- Observation in PDDL3 domains
- **Architectures of $SGPlan_5$**
  - **Handling soft goals**
  - **Handling trajectory constraints**
  - **Subproblem-level planning**
- Conclusions

Architecture of $SGPlan_5$



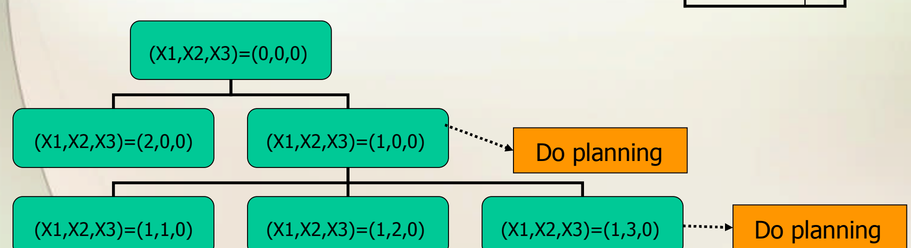
Partitioning of Problems with Soft Goals

- State is a conjunctive list of state variables
- Naïve approach
  - Enumerate all reachable values of state variables in (soft and hard) goals
  - Choose assignment that optimizes plan metric
- To overcome high complexity of enumeration
  - Decompose enumeration according to locality of constraints in subgoals
  - Conduct heuristic search for large space

Handling Soft Goals

- If enumeration cost is small and metric has only soft-goal violations (on final state)
  - Enumerate and plan to optimal assignment
- Otherwise
  - Perform DFS with backtracking
    - Use plan metric value as heuristic value
    - Plan for each explored node

Soft Const	Penalty
$X1 \leq 3$	2
$X2 = X3$	3
$X2 \leq X1$	4
$X3 \neq 5$	5

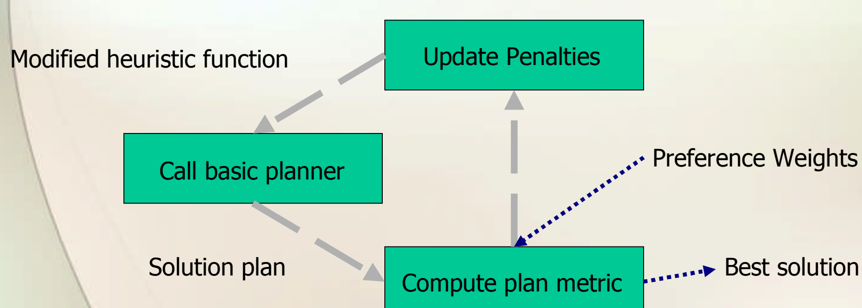


Handling Trajectory Constraints

- General difficulties
  - Huge complexity to enumerate over intermediate states
  - Difficult to detect violation of constraints beforehand
  - Need support from basic planner
- Approach
  - Adapt partition-and-resolve strategy and basic solver for trajectory constraints

Handling Soft Trajectory Constraints

- Approach: relax-and-tighten
  - Assign a penalty to each soft constraint
  - Initially set all penalties to zero
  - Call solver with modified heuristic function
  - Augment penalties of violated constraints



Subgoal-level planning

- Modified Metric-FF for PDDL3
  - Support temporal features and partitioned approach (done in IPC4)
  - New search heuristic MCDC, similar to that in Fast Downward
  - Penalize actions whose relaxed plan violates trajectory constraints
- Search-space reduction
  - Eliminate irrelevant actions that only relate to facts or subgoals in other subproblems (done in IPC4)
  - Prioritize actions that do not cause inconsistencies of bottleneck state variables (variable ordering)
- Subgoal-level decomposition techniques
  - Necessary if subproblem is too complicated
  - Incremental planning on subproblems with multiple subgoals

IPC5 Results (2006)

- SGPlan$_5$ solved 871/978 (89%) of the benchmarks
  - 1st prize in satisficing planners

Table 4: Number of instances solved by SGPLan$_5$ with respect to the total number of instances (in parenthesis) in each domain variant. (– means no instances in that domain.)

Domain	Prop.	MetTime	Simp.	Qual.	Comp.	Const.	Total
TPP	30(30)	79(80)	20(20)	20(20)	20(20)	18(30)	187(200)
OpenStacks	30(30)	40(40)	20(20)	20(20)	–	–	110(110)
Trucks	28(30)	30(30)	20(20)	20(20)	20(20)	20(20)	138(140)
Storage	30(30)	30(30)	20(20)	20(20)	20(20)	9(30)	129(150)
Pathways	30(30)	30(30)	30(30)	–	30(30)	–	120(120)
Rovers	40(40)	32(40)	20(20)	20(20)	–	–	112(120)
PipesWorld	30(50)	30(50)	–	–	15(18)	0(20)	75(138)

Comparison with Other IPC5 Planners

Top Planners in IPC5 [46]	1st/2nd Places	Variants Solved
SGPlan_5 [70]	32/3	35
Downward [58]	1/4	5
MIPS-BDD [34, 33]	0/1	1
MIPS-XXL [37, 36]	0/14	14
HPlan-P [7]	0/5	5
YochanPS [10, 30]	1/8	9

Conclusions

- Constraint partitioning is a powerful approach for exploiting constraint structure and for reducing complexity
  - Bottom-up resolution with guidance provided by top-level active global constraints
  - Using existing solvers to solve partitioned subproblems
- Planning with trajectory constraints and goal preferences
  - New partitioning strategy from MDF analysis: partitioning attributes and granularity control
  - Handling soft goals: enumeration or DFS
  - Heuristics for trajectory constraints
  - Integration of techniques

Questions?