# The SGPlan Planning System in IPC-6

**Chih-Wei Hsu** and **Benjamin W. Wah**
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois, Urbana-Champaign
Urbana, IL 61801, USA
{*chsu,wah*}*@manip.crhc.uiuc.edu*

**Abstract**

In this paper, we describe the the planning system $\text{SGPlan}_6$ participating in the satisficing tracks of the deterministic part of the Sixth Planning Competition (IPC6). $\text{SGPlan}_6$ is designed to solve both temporal and non-temporal planning problems specified in PDDL3 which may have soft goals, derived predicates or ADL features. $\text{SGPlan}_6$ inherited the parallel decomposition framework from $\text{SGPlan}_5$ which consists of the partitioning procedure, the resolution algorithm, and the subproblem solver. New enhancements are added in $\text{SGPlan}_6$ to address poor localities and to further exploit multi-valued domain structure. We identify domain-specific features that lead to the use of techniques for overcoming different scenarios of problem decomposition.

## Introduction

In this paper, we describe the the planning system $\text{SGPlan}_6$ participating in the satisficing tracks of the deterministic part of the Sixth Planning Competition (IPC6). $\text{SGPlan}_6$ is designed to solve both temporal and non-temporal planning problems specified in PDDL3 which may have soft goals, derived predicates or ADL features. $\text{SGPlan}_6$ inherited the parallel decomposition framework from $\text{SGPlan}_5$ (Hsu et al. 2007) which consists of the partitioning procedure, the resolution algorithm, and the subproblem solver. New enhancements are added in $\text{SGPlan}_6$ to address poor localities and to further exploit multi-valued domain structure.

In general a planning problem $\mathcal{T} = (\mathcal{A}, \mathcal{F}, \mathcal{I}, \mathcal{G})$ is a quadruple, where $\mathcal{A}$ is the set of possible actions in $\mathcal{T}$, $\mathcal{F}$ is the set of all state variables, $\mathcal{I}$ is the set of initial state defining the original values of all state variables, and $\mathcal{G}$ is the strict goal condition on the final state. In addition to binary and numeric state variables, $\text{SGPlan}_6$ supports multi-valued state variables which can be either explicitly defined using object fluents or implicitly detected by translating existing binary representation. A multi-valued formulation has been proved beneficial to planners as it provides a more compact representation than a binary-valued representation. Indeed, $\text{SGPlan}_5$ has demonstrated the success of utilizing multi-valued formulation for a more effective problem decomposition and goal utilities optimization.



Following the previous SGPlan framework, $\text{SGPlan}_6$ exploits the locality of planning constraints to parallel decompose a large planning problem into loosely coupled subproblems. *Parallel decomposition* (Meuleau et al. 1998), partitions a state space into subproblems in such a way that the combined state space is the cross product of the subproblem state spaces. Parallel decomposition leads to the partitioning of the variables into (possibly overlapping) subsets. Those variables that are shared among the subsets are called *shared* or *complicating variables*, and those in one subset are called *private* or *local variables*. As a result of the partitioning of the variables, the constraints are also partitioned into subsets: those involving variables in one subset are called *local constraints*, whereas those involving variables in more than one subset are called *complicating* or *global constraints*.

Given that the strong locality of constraints has been observed in many planning domains (e.g. (Chen, Wah, and Hsu 2006; Hsu et al. 2007)), parallel decomposition has the potential to exponentially reduce the complexity of the original problem. Besides the SGPlan systems, it has also been applied in *factored planning* (Amir and Engelhardt 2003) for solving STRIPS planning problems and in *factored MDPs* (Boutilier, Dearden, and Goldszmidt 2000) for solving probabilistic planning problems. Earlier work like localized planning (Lansky 1998) or HTN planning (Erol, Hendler, and Nau 1994) also employed similar methodology though their decomposition schema is not automatically generated.

The key problem of parallel decomposition is to resolve resolves those inconsistent complicating variables and those violated complicating constraints. Like previous versions of SGPlan, the resolution step in $\text{SGPlan}_6$ is based on the theory of partitioned *extended saddle point condition* (ESPC) (Wah and Chen 2006) which can tackle constraints defined on mixed-integer variables. On the other hand, *factored planning* algorithms which apply tree solving algorithms in *constraint satisfaction problems* (CSPs) (e.g. (Dechter 2003) can only resolve enumerable discrete variables.

We first briefly introduce the architecture of SGPlan and then describe new developments which are domain-specific features for identifying different scenarios of problem decomposition and the corresponding techniques.

## Architecture of SGPlan

We show the three inter-related steps when solving a problem partitioned by parallel decomposition.

- The *parallel decomposition* step partitions the variables into subsets and identifies the complicating variables and complicating constraints across the subsets.
- The *resolution* step resolves those inconsistent complicating variables and those violated complicating constraints. The resolution and solution steps may iterative a number of times before a feasible or an optimal solution is found.
- The *subproblem solver* solves a subproblem. As the subproblems are very similar to the original problem, they can be solved by the same solver. Some modifications to the subproblem solver may be needed in order to handle the complicating variables and complicating constraints.

The planer assumes a compact representation of planning constraints derived from multi-valued state variables. Next, we identify two domain-independent attributes that can relate a large number of constraints and that lead to better constraint localities. The first is the set of guidance variables that are multi-valued state variables in goal-state constraints. Intuitively, they are essential for formulating subproblem goals. Configurations of guidance variables that violate strict goals must be removed. The second attribute is the set of bottleneck state variables. These represent objects that limit parallelism and are the sources of mutexes because actions try to access them concurrently. For example, the locations of vehicles in the transportation domain are bottleneck variables because every action asserts the location of one vehicle. The number of bottleneck variables will be used to decide the number of subproblems. Besides, we eliminate variable dependency by detecting variables whose values can be derived from other variables via derived predicates or action effects.

Before partitioning the problem, the planner heuristically selects a subset of soft goals and solving them in conjunction with the other strict goals by our decomposition-based planner. Instead of enumerating the soft goals directly, the planner decides on the values of the guidance state variables which entail different configuration of soft goal satisfaction. The reason is guidance variables which are more orthogonal and they can alleviate the issue of soft goal dependency. Each assignment of guidance variables, though not necessarily optimal or feasible, corresponds to a classic planning subproblem without soft goals. We find the best assignment by a heuristic search on the space of guidance variables with backtracking. The heuristic function of search estimates the plan metric value corresponding to each assignment with the aid of relaxed plan provided by the subproblem solver.

The parallel decomposition is done by partitioning guidance state variables. The number of partitions is an important parameter of our partitioning procedure. First, the maximal number of partitions is the number of guidance variables since it is no-good to have non-disjoint clustering. Further, the maximal number of actions can be executed in parallel without inducing mutexes may be also a logical choice of the maximal number of partitions. As a result, we set the number of partitions to be the smaller of the number of guidance variables and the number of bottleneck variables. We then cluster the guidance variables by formulating a graph partitioning problem. We define a node in the graph to be a guidance variable, and an edge between two nodes when they are involved in the same planning constraint which can be a mutual exclusion, a goal condition, and an action condition.

The decomposition results in global constraints across subproblems which are mainly mutual exclusions among actions of different subproblems. To resolve the flaws in the composed plan, we apply ESPC to develop the resolution algorithm. The constraints and variables in ESPC formulation are very similar to those in Graphplan (Blum and Furst 1997) which searches a feasible plan in the planning graph. Violated constraints are in fact the planning flaws in the action graph (Gerevini, Saetti, and Serina 2003) of the composed plan. Using a penalty function of the objective function in each subproblem and the transformed global constraint functions weighted by their penalties, we bias the search of individual subproblem to satisfy global constraints. In each iteration, the basic planner solves a subproblem by generating a feasible plan that satisfies the local planning constraints as well as to minimize the weighted violations of the global constraints. The penalties are gradually increased and all the violated global constraints are resolved ultimately.

SGPlan solves each subproblem individually by a modified Metric-FF planner. The first major modification is to penalize global constraint violations in the heuristic search as the original Metric-FF (Hoffmann 2003) only consider reachability of the subproblem goal and the plan metric. SGPlan predicts the number of flaws introduced into the composed plan using the relaxed plan and then force the search to select the action with the best weighted sum of original heuristic value and the the global constraint violations. The second major modification is to extend Metric-FF to temporal planning. We study the problem of converting non-temporal plan to a temporal one, and the issue of inserting the solution of a subproblem into the composed schedule. The solution is a enhanced PERT algorithm that considers numeric resource constraints in its schedule. Finally the implementation of derived predicates and soft goals are added in SGPlan with the aid of planning axioms.

## New Developments in SGPlan$_6$

In this section, we describe the new developments in SGPlan$_6$. In general we start from study simplified version of problems which can be a smaller version or a relaxed version. we develop efficient techniques for some cases and identify the invariant attributes of those cases. We extend the results to the original problem and evaluate the performance for refining the attributes or annotated rules.

The first part is to detect a richer set of producible state variables. Every value or configuration of a producible state variable, no matter it is numeric or discrete, can be reachable anytime without affecting the feasibility of plan. A sufficient condition of producible state variables used in previous versions of SGPlan is to choose effects of actions without any preconditions. In SGPlan$_6$, we extend the condition in such a way that we examine always reachable actions and prove the effects of those actions are either producible state variables or ones have no impact on the feasibility of plan. We then decompose the original planning problem into twp parts: first is to generate sufficient amount of producible variables and then solve the original problem with constraints on producible variables relaxed. As the producible state variables tend to be competing resource among subproblems, this strategy eliminates lots of global constraint violations in the parallel decomposition. For domains which optimizing the amount of producible state variables, we can also isolate the plan quality optimization problem from the original problem. The dependency relationship among producible variables also leads to stage decomposition. The producible state variables also should not be treated as guidance variables as goal conditions on them can be achieved during post-planning.

For planning domains with poor localities or localities cannot be exploited with our partitioning procedure, we apply subproblem-level decomposition techniques on larger subproblems. In previous version of SGPlan, we have used goal agenda and landmark analysis (Hoffmann, Porteous, and Sebastia 2004). However, many domains do not have landmarks or provable goal orderings. Therefore, we do subproblem decomposition in a more relaxed setting in the sense that subtasks need not have provable ordering relationship. As the ordering may be infeasible now, we put the subtask ordering in a heuristic search framework. We develop an ordering heuristic and stopping condition of ordering enumeration based on the marginal improvement of plan metric. We also study the granularity of ordering to overcome the complexity of ordering space. The parameters and rules of the above ordering search is learned from the experience of solving existing and smaller benchmarks.

Finally, we exploit the multi-valued formulation further by generalizing the "ignore-delete-list" heuristic on problems with multi-valued state variables. The "ignore-delete-list" heuristic turns to a relaxed problem that domain values that have been achieved will not be invalidated. We utilize the domain transition graph of each multi-valued state variables during the subproblem decomposition as we can identify intermediate stages for reaching the subproblem goal by examine the transition graph. The causality relationship of multi-valued formulation is also used to identify producible state variables.

## References


Amir, E., and Engelhardt, B. 2003. Factored planning. In *Proc. Int'l Joint Conf. on Artificial Intelligence*, 929–935.

Blum, A. L., and Furst, M. L. 1997. Fast planning through planning graph analysis. *Artificial Intelligence* 90:281–300.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 2000. Stochastic dynamic programming with factored representations. *Artificial Intelligence* 121(1-2):49–107.

Chen, Y. X.; Wah, B. W.; and Hsu, C. W. 2006. Temporal planning using subgoal partitioning and resolution in SGPlan. *J. of Artificial Intelligence Research* 26:323–369.

Dechter, R. 2003. *Constraint Processing*. Morgan Kaufman.

Erol, K.; Hendler, J. A.; and Nau, D. S. 1994. HTN planning: Complexity and expressivity. In *Proc. National Conf. on Artificial Intelligence*, 1123–1128.

Gerevini, A.; Saetti, A.; and Serina, I. 2003. Planning through stochastic local search and temporal action graphs in LPG. *J. Artificial Intelligence Research* 20:239–290.

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered landmarks in planning. *J. of Artificial Intelligence Research* 22:215–278.

Hoffmann, J. 2003. The Metric-FF planning system: Translating "Ignoring Delete Lists" to numeric state variables. *J. of Artificial Intelligence Research* 20:291–341.

Hsu, C. W.; Wah, B. W.; Huang, R.; and Chen, Y. X. 2007. Constraint partitioning for solving planning problems with trajectory constraints and goal preferences. In *Proc. Int'l Joint Conf. on Artificial Intelligence*, 1924–1929. AAAI.

Lansky, A. L. 1998. Localized planning with action-based constraints. *Artificial Intelligence* 98(1-2):49–136.

Meuleau, N.; Hauskrecht, M.; Kim, K.; Peshkin, L.; Kaelbling, L. P.; Dean, T.; and Boutilier, C. 1998. Solving very large weakly coupled Markov decision processes. In *Proc. AAAI Conf. on Artificial Intelligence*, 165–172.

Wah, B., and Chen, Y. X. 2006. Constraint partitioning in penalty formulations for solving temporal planning problems. *Artificial Intelligence* 170(3):187–231.