Subgoal Partitioning and Resolution in SGPlan*

Yixin Chen, Chih-Wei Hsu, Benjamin W. Wah
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
URL: http://manip.crhc.uiuc.edu/programs/SGPlan
Email: {chen, chsu, wah}@manip.crhc.uiuc.edu

Introduction

In this demo, we illustrate the idea and operations of SGPlan (Chen, Hsu, & Wah 2004; Wah & Chen 2005), a PDDL2.2 planner that won the first prize in the suboptimal temporal metric track and a second prize in the suboptimal propositional track in the Fourth International Planning Competition (IPC4), 2004. SGPlan is the only planner that won in two tracks of the competition. Since SGPlan is a suboptimal planner, it did not participate in the optimal track.

SGPlan was designed based on the key observation that many planning applications have a clustered structure of their constraints. Specifically, we have found that constraints are highly localized by their subgoals. Based on this observation, we have proposed to partition problem constraints by their subgoals into multiple subsets, solve each subproblem individually, and resolve inconsistent global constraints across subproblems based on a penalty formulation.

In this demo, we illustrate the key observation of constraint locality in some application domains, the constraint partitioning approach we have used in SGPlan, and the process of resolving inconsistent global constraints across partitioned subproblems.

Observations on Constraint Locality

Our approach is based on the observations that the constraints in many planning applications are not purely random but highly structured, and that these constraints can be clustered by their subgoals.

For example, in AIRPORT, an airport scheduling domain, each subgoal is a destination in an airport. Most mutual-exclusion constraints are localized within a subgoal relating two actions from the same subgoal, and only a small fraction of global constraints relate multiple subgoals. We illustrate this observation as follows.

Slide 1 illustrates the topology of a part of the Munich airport in the AIRPORT-4 instance. The example involves a planning task for moving three airplanes from their starting positions to some destination gates.

Slide 2 illustrates the constraints in a temporal planning problem whose actions have durations. Constraints in PDDL2.2 temporal planning problems are mutual-exclusion constraints proposed in Graphplan (Blum & Furst 1997). We show examples of violated mutual-exclusion constraints in the AIRPORT-4 instance. A plan is a solution plan if there are no violated mutual-exclusion constraints.

Slide 3 plots the structure of the mutual-exclusion constraints in three solution plans generated by LPG on the AIRPORT-4 instance. These plans correspond to, respectively, one, two, and three airplanes. We plot actions in boxes and a line between two actions if they are related by a mutual-exclusion constraint. We see that the number of actions and the number of constraints grow in proportion to the number of subgoals, which lead to an exponential growth in search complexity.

Slide 4 shows that the seemingly random constraints in the solution plans generated by LPG are in fact highly structured and can be clustered by their subgoals. To see this, we generate a plan for each of the three subgoals in the AIRPORT-4 instance, compose the plans together, and plot all the actions and constraints. We show that most constraints are local constraints relating two actions from the same subproblem, and that only a few global constraints (shown in red lines) relate two actions from different subproblems. This observation is intuitively sound because the movements of airplanes are largely independent. Two airplanes interact with each other only when they are at the same position and can be scheduled independently most of the time.

We also show in the demo that the constraint-locality property is observed in other IPC4 domains. Slide 5 plots the ratio of global constraints to all constraints under subgoal partitioning in four other IPC4 domains. It shows that the fraction of global constraints with respect to the total number of constraints is consistently low.

Constraint Partitioning in SGPlan

In SGPlan, we use a planning algorithm based on constraint partitioning. The approach partitions a planning problem into multiple subproblems by its subgoals, and

*Research supported by National Science Foundation Grant IIS 03-12084 and National Aeronautics and Space Administration Grant NCC 2-1230.

solves each subproblem individually before composing the solutions. As there are global constraints relating multiple subproblems, the main technical problem addressed in the design of SGPlan is the resolution of inconsistent global constraints.

Resolving global constraints for temporal planning is particularly difficult than some previous partitioning approach for several reasons. First, we are interested in fully-automated planning and do not have any domain-specific knowledge about the application problems. Second, existing efficient decomposition methods for linear and convex optimization cannot be used because the constraint functions are symbolic, nonlinear, and do not have linearity or convexity properties. Third, mathematical conditions requiring continuity and differentiability cannot be derived because the constraints may be discrete and not in closed form. Due to these difficulties, previous constraint programming methods, such as penalty and Lagrangian methods, cannot be applied for resolving inconsistent global constraints.

To overcome these difficulties, Slide 6 presents our proposed Extended Saddle Point Condition (ESPC) based on a new ℓ_1^m-penalty function (Wah & Chen 2005). Our theory provides a necessary and sufficient condition for characterizing constrained local optimal solutions without continuity and differentiability assumptions on constraints. Moreover, the condition can be decomposed into a partitioned form, where each subproblem is associated with a necessary local condition. The partitioned condition greatly reduces the search space of each subproblem.

Slide 7 illustrates the architecture of our SGPlan planner based on constraint partitioning. In SGPlan, a planning problem is partitioned into multiple subproblems by its subgoals, and may be further partitioned using Landmark analysis (Porteous, Sebastia, & Hoffmann 2001). SGPlan then uses a modified version of the FF planner (Hoffmann & Nebel 2001) to find a solution plan for each subproblem that satisfies all local constraints and that minimizes a modified objective function. In order to bias the search towards resolving global inconsistencies, our modified objective function includes the original objective function and the penalty terms for violated global constraints. In each iteration, the penalty values are updated after solving all the subproblems and evaluating the global constraints.

Resolution of Inconsistent Global Constraints

Next, we illustrate the rapid reduction of violated global constraints in SGPlan. In addition, we show the effectiveness on using ESPC in SGPlan by comparing it with a greedy search without ESPC.

Slide 8 demonstrates the solution process of SGPlan on the AIRPORT-4 instance with three subgoals. In the first iteration, we generate a plan for each subgoal. We see that there are some violated global constraints (shown in arrows) at the start of the second iteration. In the second iteration, SGPlan solves each subproblem individually, using the ℓ_1^m-penalty function to bias the search towards resolving the violated global constraints. It is clear that the number of global constraints is reduced quickly, and a solution plan is found after the second iteration.

Slide 9 demonstrates the effectiveness of using ESPC on four example instances in four application domains. In each case, we plot the number of violated global constraints with respect to the number of subproblems solved by SGPlan. As a comparison, we also tested a greedy search algorithm without using ESPC and a penalty function to bias the search. In each instance, we generated three alternative plans and accepted the one with the minimum number of violated global constraints. Clearly, SGPlan using ESPC can resolve inconsistent global constraints much more efficiently than the greedy algorithm.

Slide 10 summarizes the evaluation results of SGPlan on the seven IPC4 domains. For each domain, we show the total number of instances and the corresponding number of instances solved by SGPlan and five other leading IPC4 planners within the 30-minute CPU-time and 1-GB memory limits. The results show that SGPlan is consistently better than other planners across all IPC4 domains.

References

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

Chen, Y. X.; Hsu, C.-W.; and Wah, B. W. 2004. SGPlan: Subgoal partitioning and resolution in planning. In *Proc. Fourth Int'l Planning Competition*. Int'l Conf. on Automated Planning and Scheduling.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *J. of Artificial Intelligence Research* 14:253–302.

Porteous, J.; Sebastia, L.; and Hoffmann, J. 2001. On the extraction, ordering, and usage of landmarks in planning. In *Proc. European Conf. on Planning*, 37–48.

Wah, B., and Chen, Y. X. 2005. Fast temporal planning using the theory of extended saddle points for mixed nonlinear optimization. *Artificial Intelligence*.