Evaluating Plans through Restrictiveness and Resource Strength

Federico Pecora*, **Amedeo Cesta**

Institute for Cognitive Science and Technology
Italian National Research Council

name.surname@istc.cnr.it

Abstract

The planning and scheduling framework which is the object of this paper consists in a loosely-coupled integration which is achieved by cascading a classical planner and a general purpose scheduler. The output of the planning phase yields a partially ordered plan which is then integrated with time and resource constraints to produce a scheduling problem. The research described in this paper is aimed at understanding the structural trademarks of the causal knowledge that a scheduling tool can inherit from STRIPS-based reasoners. Specifically, we analyze quality of the plans in terms of two well-known properties of scheduling problems, namely the Restrictiveness and the Resource Strength. To this end, we describe a set of experiments carried out on a series of state-of-the-art planners aimed at assessing the bias of different planning strategies in the context of the loosely-coupled framework.

Introduction

In recent years, increasing attention has been dedicated to integrating Planning and Scheduling (P&S), with the aim of extending the reach of automated solving to combinatorial optimization problems which require both causal and time/resource reasoning. Integrated P&S frameworks are typically obtained following two approaches. On one hand, integrated P&S architectures can be designed using a "contextual" approach, so called because time/resources and the causal sub-problem are dealt with contextually. This can be achieved, as for instance in (Gerevini, Saetti, & Serina 2003; Smith & Weld 1999; Currie & Tate 1991; Ghallab & Laruelle 1994), by expanding a strictly causal solving technique with algorithms and data structures capable of dealing with time and/or resources. Other contextual solvers adopt the opposite strategy, namely starting from a scheduling-centric approach and extending it with limited causal reasoning capabilities (Cesta, Fratini, & Oddi 2004; Jonsson *et al.* 2000; Muscettola *et al.* 1992). The main drawback of contextual P&S integration strategies is that typically the "enhanced" system does not aquire a full set of capabilities. This is the case, for instance, of planners which support PDDL2.1/2.2 (Fox & Long 2003; Edelkamp & Hoffmann 2004), an extension of the standard Planning Domain Definition Language (PDDL (Ghallab *et al.* 1998)) for modeling actions with durations, a characteristic which does not alone afford the versatility of common scheduling problem definition languages.

On the other hand, so-called "coupled" P&S consists in linking separate implementations of the two solving paradigms in order to achieve a bi-modular solver, in which each component deals with the respective sub-problem. This approach clearly presents some major difficulties since it requires the definition of forms of information sharing between the two subsystems. Moreover, this type of integration can be achieved by a strong coupling of the two solving components (i.e., in which every decision taken by one component is "propagated" by the other component), or, on the other end of the spectrum, by loosely-coupling the two solving sub-systems (i.e., the planner produces a plan and the scheduler then enforces the time and resource related constraints upon this plan). While a strongly-coupled architecture may seem to be more realistic, there is increasing evidence that often a loose-coupling of solving components is more applicable to solve certain classes of real-world problems[1].

The P&S framework which is the object of our work consists in a loosely-coupled integration which is achieved by cascading a classical planner and a general purpose scheduler. The output of the planning phase yields a partially ordered plan which is then integrated with time and resource constraints. The resulting scheduling problem is then given to the scheduler for resolution. The research described in this paper stems from the necessity for an in-depth comprehension of some fundamental aspects related to this form of integration. In particular, we address the issue of understanding the structural trademarks of the causal knowledge

*Also affiliated with the the Department of Systems and Informatics (DIS) of the University of Rome "La Sapienza", Rome, Italy — pecora@dis.uniroma1.it.

[1] This is the case, for instance, in military planning, where high-ranking officers take high-level planning decisions, disregarding issues such as resource allocation or specific synchronization constraints, while the more scheduling-related decisions which must be taken in order to make a plan operational occur afterwards. An interesting discussion on this topic was brought up during the ICAPS Workshop on Integrating Planning into Scheduling, June 2004.

that a scheduling tool can inherit from STRIPS-based reasoners. Thus, we set out to measure the structural properties of the scheduling problems yielded from the planning phase in the light of the subsequent scheduling. The present analysis extends the results obtained in (Pecora, Rasconi, & Cesta 2004; Cesta, Pecora, & Rasconi 2004), in which we focused on the bias produced by different planning approaches (heuristic search and planning graph based) in the light of makespan-optimizing scheduling. In this paper, our aim is to assess the bias of the planning phase on two well-known properties of scheduling problems, namely the Restrictiveness and the Resource Strength. These two structural properties of scheduling problems are commonly used[2] as control parameters for the random generation of Resource Constrained Project Scheduling Problems with minimum and maximum time lags (RCPSP/max) (Brucker *et al.* 1998). To this end, we describe a set of experiments carried out on a series of state-of-the-art planners aimed at generating scheduling problems which exhibit varying degrees of these two parameters.

This paper is organized as follows. We first present the general experimental setup within which our analysis is carried out. To this end, we first describe more precisely the loosely-coupled P&S framework (an architecture similar to REALPLAN-MS (Srivastava 2000)) which our work focuses on, as well as the benchmark set on which our analysis is performed. We then describe the details of the structural properties we wish to measure, while the section that follows describes an extention to one of the planners analyzed in the experiments aimed at increasing the detail of our observations. The main results of this paper consist in analyzing and comparing the performance in terms of the measures introduced earlier of four state-of-the-art planners. The results are then put together and further analyzed, and we conclude with a summary of our findings as well as an outlook for future work.

Experimental Setup

In an integrated P&S context, time and resource constraints as well as causal dependencies are contemplated in the initial problem definition. Every operator is inherently associated to a time duration and requires a certain quantity of consumable multi-capacity resources that ensure its executability.

The general schema we use in this investigation is as follows (see figure 1). A causal model of the environment is given as input to a planner, i.e., the domain representation and the problem definition, both expressed in a STRIPS-like formalism. This model does not contemplate time and resource related constraints, which are accommodated after the planning procedure has taken place. In order to produce a problem specification which can be reasoned upon by the scheduling procedure, the Partial Order Plan (POP) produced by the planner is integrated with time and resource related information by means of a plan adaptation procedure. This procedure produces a minimal-constrained deordering (Bäckström 1998) of the POP and integrates it with

[2] See for instance (Schwindt 1998).

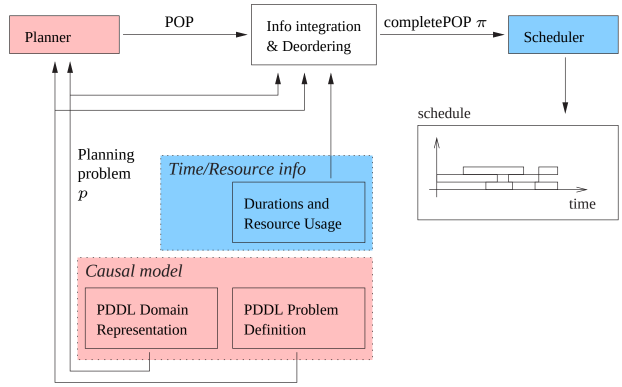

Figure 1: The loosely-coupled reference framework, in which a planning problem p is solved by the planner, ultimately yielding a scheduling problem π.

time and resource related information to produce what we shall call a completePOP. In more formal terms:

Definition 1. *A* completePOP P *is a tuple* $\langle \mathcal{T}, \mathcal{P}, \mathcal{R}, \mathcal{C}, \mathcal{U}, \mathcal{D} \rangle$ *where*

- $\mathcal{T} = \{T_1, \ldots, T_n\} \cup \{T_0, T_{n+1}\}$ *is the set of* tasks *which correspond to the* n *activities in the POP produced by the planner;* T_0 *and* T_{n+1} *are called* source *and* sink *activities;*
- $\mathcal{P}$ *is a set of* precedence constraints *between the tasks, where* $T_i \prec T_j$ *means that task* T_i *must be completed before the execution of task* T_j *can begin;*
- $\mathcal{R} = \{R_1, \ldots, R_m\}$ *is a set of renewable* resources*;*
- $\mathcal{C} = \{C_1, \ldots, C_m\}$ *are the* capacities *of the resources;*
- $\mathcal{U} : \mathcal{T} \times \mathcal{R} \rightarrow \mathbb{N} \cup \{0\}$ *is the function which determines the* resource usage *attributes of a task, that is,* $\mathcal{U}(T, R) = u$ *if task* T *uses* u *units of resource* R *(*$u = 0$ *indicates that* T *does not use resource* R*);*
- $\mathcal{D} : \mathcal{T} \rightarrow \mathbb{N} \cup \{0\}$ *is the function which determines the* durations *of the tasks, that is,* $\mathcal{D}(T) = d$ *if the duration of task* T *is* d *time units.*

According to the definition above, a completePOP coincides with a Resource Constrained Project Scheduling Problem (RCPSP) (Brucker *et al.* 1998) and can be visualized in the form of a precedence graph.

A General Multi-Agent Problem Template

Our aim in this paper is to analyze the performance in terms of plan quality of different planners within the above mentioned loosely-coupled framework. More specifically, we are interested in assessing the influence of different planning strategies on the structural characteristics of the scheduling problems they lead to. This section briefly illustrates the main features of the benchmark problems employed in the experimental evaluation.

We focus on a multi-agent inspired planning domain which produces completePOPs with high levels of concurrency by encapsulating the notion of *executing agent*. Concurrency is obtained by inducing the presence of multiple agents in the POP. Having augmented the STRIPS subset of the PDDL language with some simple extensions to allow the specification of durations and resource usage for the operators (these directives are ignored by the planner and integrated into the POP to form the completePOP), the general structure of a problem which leads to completePOPs which exhibit high degrees of concurrency is shown in figure 2, where `:capacity 0` denotes an object which is not a resource. Given this structure, the number of agents in the completePOP is determined by the number of objects with non-zero capacity, i.e., $|\{\texttt{A1}, \ldots, \texttt{An}\}|$, and the resource usage of each task is given by the `:uses` clause in the abstract operator specification.

```
(define (domain ... ) ...
  (:action op
    :parameters (?a - agent ... )
    :precondition ( ... )
    :effect ( ... )
    :uses (?a USAGE)
    :duration DUR) ... )
```

(a)

```
(define (problem ... ) (:domain ... )
  (:objects
    A1, ..., An - agent :capacity CAP
    B1, ..., Bm - type :capacity 0 ... )
  (:init ... )
  (:goal ... ))
```

(b)

Figure 2: General structure of augmented PDDL domain (a) and problem specifications (b) which lead to completePOPs with multiple agents. Bold typeface indicates directives which are ignored by the planner.

A very simple example "instantiation" of this somewhat general template could be the following: a team of robotic `agent`s can move from one `room` of an environment to another (so long as there is a door connecting the two adjacent rooms), and they can perform a simple `operation` on certain types of `object`s (iff they are in the same room as the object). In order to model the fact that agents do not perform an `operation` on two objects at the same time, each action `:uses` one unit of the resource `agent`. This equates to modeling the agents as binary resources for the scheduler. The idea is that, once the planning has taken place (disregarding this resource usage information), the scheduler will enforce the resource usage constraints, adding as a consequence a precedence constraint during the solving phase which serializes two operations on the same object which otherwise were independent.

Notice also that it is possible to model this binary resource constraint causally, by adding `not (busy ?a)` to the preconditions and `busy ?a` to the effects of action `operation`. The `busy` predicate can be seen as a "causal excuse" for the resource capacity constraint modeled with the `:uses` clause, and can be reset by a dummy `finish` action (whose effect is `not (busy ?a)`). Notice that in this manner we would be producing scheduling problems without resource contention peaks, thus transferring the burden of resolving these contentions to the planning phase. It has been shown in (Pecora & Cesta 2002; R-Moreno *et al.* 2002) on a very similar domain that adopting this modeling strategy heavily affects the efficiency of the planning procedure, making it very difficult to solve even problems of small size.

Restrictiveness and Resource Constraints

There is a consolidated body of work in the field of scheduling which deals with classifying scheduling problems with respect to various quantifiable properties. Our aim in this paper is to interpret these structural characteristics in the light of different implementations of the planning phase which constitutes the first component of the loosely-coupled framework. In this section we present the two well-known structural factors of scheduling problems which we will focus on. In the remainder of this paper we will use these factors to measure the quality of the scheduling problems obtained with a number of state-of-the-art planners.

Order Strength. The first parameter we focus on quantifies the effect of the precedence constraints contained in the plan (i.e., the causal structure of the scheduling problem) on the possible execution sequences of the tasks. Our aim is to measure to some extent the magnitude of the search space which is explored by the scheduling phase. To this end, we borrow the notion of *restrictiveness* from Project Scheduling.

Given a completePOP $\langle \mathcal{T}, \mathcal{P}, \mathcal{R}, \mathcal{C}, \mathcal{D} \rangle$, let $l(\mathcal{P})$ be the number of linear extensions of the causal structure determined by the precedence constraints in $\mathcal{P}$. The restrictiveness of the completePOP[3] can be defined as follows:

$$\sigma = 1 - \log \frac{l(\mathcal{P})}{|\mathcal{T}|!}$$

where $|\mathcal{T}|!$ represents the maximum number of linear extentions of a partial order in the set of tasks $\mathcal{T}$. The restrictiveness measures the degree to which the precedences between activities restrict the number of feasible execution sequences of the tasks. For completely parallel causal structures, the restrictiveness is 0, while it is 1 for serial task networks. In general, the higher σ, the less different alternatives exist to resolve resource conflicts between tasks (by defining additional precedence constraints). As a consequence, the restrictiveness is directly linked to the hardness of many RCPSP/max problems (both with respect to feasibility and optimization) (Schwindt 1998).

[3] The restrictiveness of a directed graph is used as a control parameter for the random generation of RCPSP/max problems, e.g. (Schwindt 1998).

Unfortunately, calculating σ is #P-complete, since it involves counting linear extensions (Brightwell & Winkler 1991). As shown in (Mastor 1970), a good approximation of the restrictiveness is the *order strength*. Given a completePOP π, this measure can be determined very efficiently and is defined as follows:

$$OS_\pi = \frac{|\overline{\mathcal{P}}|}{|\mathcal{T}|\left(|\mathcal{T}|-1\right)/2} \quad (1)$$

i.e., the number of precedence relations, including the transitive ones ($\overline{\mathcal{P}}$ denotes the set of precedence relations in the transitive closure of the precedence graph), divided by the theoretical maximum number of constraints. It is shown in (De Reyck & Herroelen 1996) that OS outperforms other significant network measures regarding the correlation with the hardness of RCPSP/max instances.

The OS parameter thus expresses the properties of the causal structure determined by the planning phase, and has no connection with the resource constraints contained in the problem. As we will see, different planning strategies yield plans which differ in causal structure, thus the planning algorithms themselves are also connected to the hardness of the resulting scheduling problems instances.

Resource Strength. One of the main features of schedulers consists in the capability to deal with resources in addition to complex time-constraints. In order to understand the bias produces by different planning strategies on the resource-related characteristics of the resulting scheduling problems, we employ another well-known measure, namely the *resource strength*[4]. Given the k-th resource in the completePOP, we denote with $r^k_{\min}$ the maximum usage of this resource by a single task in the completePOP, that is

$$r^k_{\min} = \max_{T\in\mathcal{T}} \mathcal{U}(T, R_k)$$

Also, let $r^k_{\max}$ denote the peak demand of resource R_k in the precedence-preserving earliest start schedule (i.e., the infinite capacity solution). The resource strength of resource R_k is thus defined as

$$RS_k = \frac{C_k - r^k_{\min}}{r^k_{\max} - r^k_{\min}}$$

where $C_k \in \mathcal{C}$ is the capacity of resource R_k. The resource strength expresses the relationship between the resource demand and the resource availability in the given completePOP.

Notice that a scheduling problem in which the capacity C_k of the k-th resource is at most $r^k_{\max}$ (i.e., the maximum potential claim of resource k in the precedence-preserving earliest start schedule) is such that resource k is "well-dimensioned". Conversely, if C_k exceeds the peak demand of R_k, then this resource's capacity is unnecessarily large, since $r^k_{\max}$ is an upper bound on C_k such that the problem is resource-feasible. We call a problem in which R_k adheres to the first requirement *k-parsimonious*. Notice also that $r^k_{\min}$ is a lower bound for C_k, since it represents the maximum usage of resource R_k by a single task in the completePOP. Therefore, in a k-parsimonious completePOP we have that C_k is bounded in the interval $[r^k_{\min}, r^k_{\max}]$, since these bounds represent, respectively, the smallest and largest capacity which can be given to R_k in order for the scheduling problem to be resource feasible with respect to R_k. If this is the case, then clearly RS_k is also bounded in the interval $[0, 1]$, while RS_k may be greater than 1 of the completePOP is not k-parsimonious.

In conclusion, the RS parameter takes into account the resource constraints which are given by the requirements of the tasks on one hand, and the limited resource capacities on the other, thus it measures the scarcity of the resource availability with respect to the requirements. In the following sections, our analysis focuses on the average RS over the resources which are employed in the completePOP, that is

$$\overline{RS}_\pi = \frac{\sum_{k:R_k\in\mathcal{R}_u} RS_k}{|\mathcal{R}_u|} \quad (2)$$

where $\mathcal{R}_u \subseteq \mathcal{R}$ is the set of *used* resources, i.e., for which $\mathcal{U}(T, R) > 0$ for some $T \in \mathcal{T}$. Notice that a completePOP may not prescribe the use of all resources modeled in the original problem, since the particular action instantiations which achieve the goal (the POP produced by the planner) are obtained by means of a purely causal deduction, and as such does not explicitly take into account any resource-related metric. Indeed, some among the planners we analyze in the following sections over-commit more than others with respect to resource utilization. This reflects strongly on the average resource strength: the higher $\overline{RS}_\pi$, the less constraining are the capacities of the resources with respect to the task network.

While the scheduling literature points to numerous other parameters which could be useful in the context of this investigation, some of which we will be analyzing in future work, in the following section we observe how the planning phase affects the two properties defined above.

As a final note, it is interesting to notice that taking into account "classical" properties of project scheduling problems rather than the strong-coupling constraint density we have defined in our previous work, allows us to relax the restricting assumptions we have made with respect to resource capacities and usage. In particular, parameters such as the two we analyze in this work are defined for any causal structure in the scheduling problem, as well as for multi-capacity resources and multiple-resource usage constraints.

Enhancing BLACKBOX for Obtaining Multiple Causal Solutions

Before describing in detail the structural characteristics of the scheduling problems which we want to observe, we describe an extention to the loosely-coupled framework aimed

[4] This parameter was first defined in (Cooper 1976; Alvarez-Valdez & Tamarit 1989). The definition we use here is taken from (Kolisch, Sprecher, & Drexl 1995).

at enhancing the detail of our observations. Among the planners used in our experiments, we employ a planning subsystem which is capable of producing multiple plans, thus yielding, after the information integration and de-ordering procedure has taken place, multiple completePOPs. Having more solutions to the causal part of the P&S problem provides us with a richer set of scheduling problems on which to perform the structural analysis by means of the measures defined previously. This extends the results obtained in (Pecora, Rasconi, & Cesta 2004; Cesta, Pecora, & Rasconi 2004) because it enables us to observe how not only different planning strategies but also different methods for extracting solutions within a certain planning strategy can influence the structure of the underlying scheduling problem. This is particularly true in the case of planning graph based planning, in which a planning graph contains many valid plans. Moreover, these plans are partitioned into sets according to the level of expansion at which they are obtained. In other words, given the minimum level l at which there exists a solution to the planning problem, instead of obtaining one scheduling problem π^l, we obtain the set of scheduling problems $\{\pi_1^l, \pi_2^l, \ldots\}$ which derive from the solution subgraphs of the planning graph[5].

In order to obtain multiple completePOPs in a single level of planning graph expansion, we have implemented a modified version of the BLACKBOX planning system, which we call MBLACKBOX (for multiple-BLACKBOX). This enhancement is obtained by replacing the ZCHAFF SAT-solver (Moskewicz *et al.* 2001) originally integrated within the BLACKBOX system with MCHAFF, an enumerative variation of ZCHAFF (Moskewicz *et al.* 2001; Moskewicz 2004), i.e., which is capable of finding multiple models of a given well-formed formula (see figure 3). The experiments described in the following sections thus focus also on the variation of OS and RS within the sets of solutions produced by MBLACKBOX.

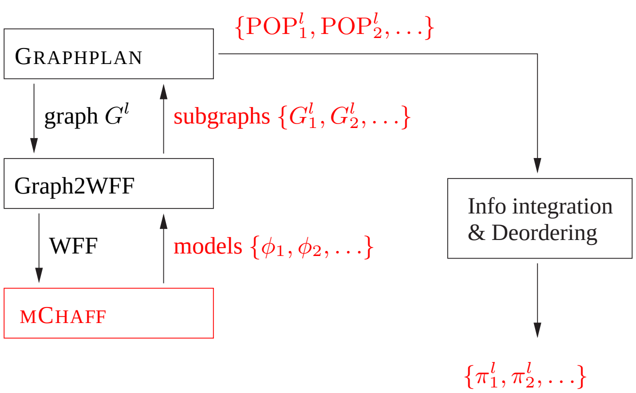

Figure 3: Enhancing BLACKBOX with enumerative capabilities by replacing ZCHAFF with MCHAFF. G^l denotes the planning graph expanded up to level l.

[5] Moreover, by increasing the level of expansion of the planning graph we can produce the sets of scheduling problems $\{\pi_1^l, \pi_2^l, \ldots\}, \{\pi_1^{l+1}, \pi_2^{l+1}, \ldots\}, \ldots, \{\pi_1^m, \pi_2^m, \ldots\}$, where m is the index of planning graph level-off. The analysis of OS and RS on further levels of expansion of the planning graph is outside the scope of this paper, and will be addressed in future work.

Experiments

The two parameters shown above represent attributes of scheduling problems which are widely accepted as meaningful. In the context of this investigation, OS characterizes the causal structure of planner-derived scheduling problems, measuring the restrictiveness of the task network determined by the planning phase, thus indicating the "degree of freedom" available to the scheduler in resolving conflicts. RS on the other hand quantifies the constrainedness of completePOPs with respect to the resources allocation determined by the planner. Both OS and RS provide a means to interpret the effect of the decisions taken by the planner with respect to the causal and resource allocation problem represented by the input planning problem.

Benchmark Problems

For the purpose of the experiments described in this section, the general multi-agent domain definition described earlier was instantiated according to the dependency graph shown in figure 4. The domain consists in a total of six operators, each requiring an object as well as a certain type of agent (type one or two). Moreover, each operator `:uses` a particular amount of the executing agent, and has a given `:duration`. Our benchmark consists of 100 randomly generated problems within this domain in the form $p = \langle A_1, A_2, \left[C_1^{\min}, C_1^{\max}\right], \left[C_2^{\min}, C_2^{\max}\right], O \rangle$, where

- A_1 (A_2) is the number of `agents` of type one (two);
- $\left[C_1^{\min}, C_1^{\max}\right]$ ($\left[C_2^{\min}, C_2^{\max}\right]$) is an interval within which lies the `:capacity` of every agent of type one (two);
- O is the number of `objects` (i.e., with `:capacity 0`).

Each problem was generated by randomly choosing A_1, A_2, O, as well as the capacity of each agent of type one and two within the corresponding interval. Un-resolvable problems were avoided by ensuring that the $C_1^{\min} \geq 4$ and $C_2^{\min} \geq 3$.

Notice that the absence of negative effects in this domain make the corresponding planning problems polynomial. This choice is determined by our interest in aspects related to the solution quality rather than the performance of the planners we consider. To this end, the 100 generated benchmark problems are intended to provide an easy set of problem instances from the planning point of view. Conversely, the additional resource and time related information poses a more elaborate scheduling sub-problem, with respect to which the tested planners exhibit a varying degree of commitment.

As we will see in the experimental analysis, the regularity of this domain introduces a strong dependency of the results obtained by the various planners on the size of the problem instance. In addition, notice that the domain is such that we expect to obtain little variation in the causal structure of the plans obtained with different planners[6]. Nonetheless, as we

[6] This domain can be interpreted as a workflow-management application, in which there is no real difficulty in determining the se-

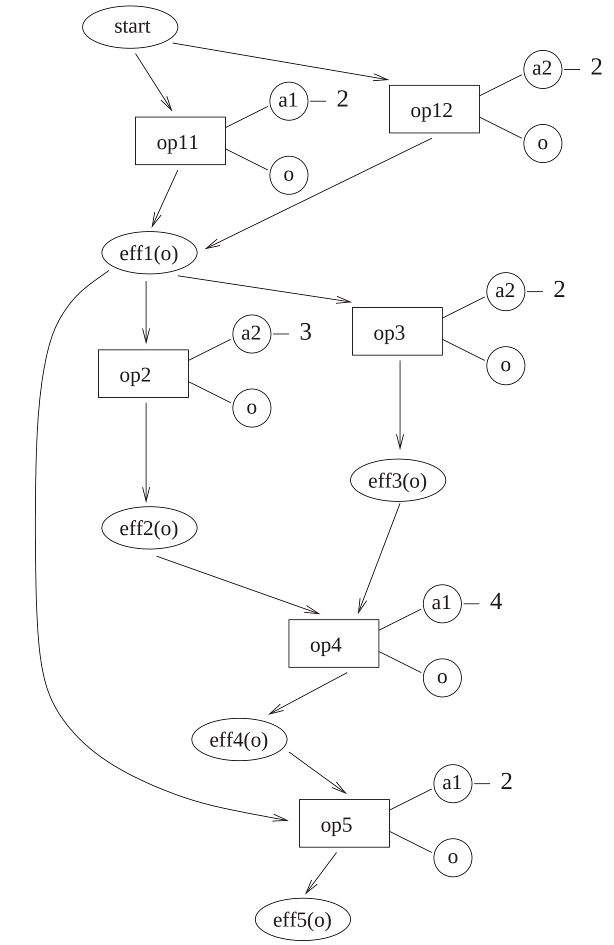

Figure 4: The dependency graph of the multi-agent domain used for the generation of the benchmark set. Ovals represent predicates and rectangles represent operators.

will see the relative regularity of the benchmark set does not affect the generality of our observations.

From POPs to Scheduling Problems

To begin with, the benchmark problems described above were solved using mBLACKBOX. As mentioned, this planner employs mCHAFF in order to produce multiple solutions to a single planning problem. On the 100 benchmark problems, mBLACKBOX yields a total of 7046 plans (the enumeration performed by mCHAFF yields between 10 and 200 unique plans for each problem). Each solution represents a completePOP π_i^j, where $i \in [1, 100]$ and j refers to the j-th unique solution to problem i. We then calculate $OS_{\pi_i^j}$ and $\overline{RS}_{\pi_i^j}$. Moreover, we extract from these measures $OS_{\pi_i}^{\max}$, $OS_{\pi_i}^{\min}$, $\overline{RS}_{\pi_i}^{\max}$ and $\overline{RS}_{\pi_i}^{\min}$, which represent the maximum and minimum values of OS and $\overline{RS}$ obtained for the i-th P&S problem.

The same set of 100 benchmark problems were solved by FF (Hoffmann & Nebel 2001), LPG (Gerevini, Saetti, & Serina 2003) and CPT (Vidal & Geffner 2004), and OS_{π_i} and $\overline{RS}_{\pi_i}$ were also computed for the 100 solutions obtained from these planners.

Resources

We begin by analyzing the resource related aspects of the obtained completePOPs. Figure 5 shows the value of $\overline{RS}$ for the various planners considered. In particular, only the first plan produced by mBLACKBOX is considered (i.e., the original non-enumerative version of BLACKBOX). Recall

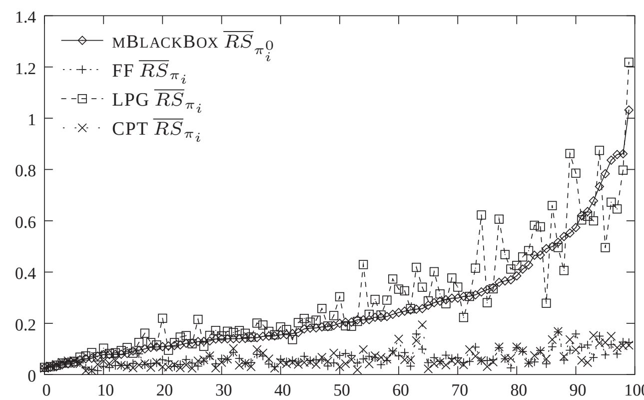

Figure 5: Average resource strength for the various planners (mBLACKBOX in non-enumerative mode).

that high values of $\overline{RS}$ indicate that the corresponding plan is under-constrained with respect to resource capacity constraints, thus that the resources are assigned in such a way that their capacity limitations are less likely to entail resource conflicts. It is immediate to notice the gap in performance between FF and CPT on one hand, and mBLACKBOX and LPG on the other. In most problems the former planners employ only one or two agents out of the overall pool of available agents. The strongly sub-optimal allocation of resources (agents) to tasks clearly has no impact on the quality of the solution to the causal sub-problem (the POP), while it heavily affects the quality of the completePOP as a whole. The scheduling problems produced by FF and CPT are thus very constrained from the point of view of resources. Overall, this may have some disadvantages with respect to the scheduling phase, such as longer makespans, less robust solutions, and so on.

The poorer quality of the plans obtained with FF and CPT demonstrates the fact that the over-committing nature of the performance-oriented heuristics employed by these planners appears to be counter productive in the light of the subsequent scheduling phase. Conversely, BLACKBOX and LPG have the nice property of committing less to resource peak leveling decisions, thus maintaining a lower level of invasiveness in the decision space of the scheduler. This phenomenon is even more interesting in the light of the fact that both BLACKBOX and LPG are (though in different ways)

quence of operators which achieves a goal, and in which the planner's task is essentially that of deciding an allocation of agents to operations.

planning graph based planners. These results thus point to an interesting load balancing quality of the planning graph paradigm.

Let us now observe the results obtained with the enumerative planner MBLACKBOX. As shown in figure 5, the performance of LPG is in line with that of the classical single-plan BLACKBOX, reaching for many problems higher peaks. Conversely, plotting the results obtained with the enumerative variant MBLACKBOX yields the results shown in figure 6. These results show that extracting more solutions

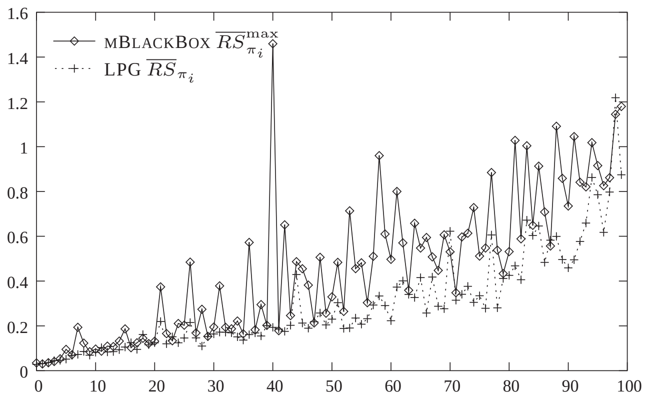

Figure 6: Average resource strength for LPG and the best performance of MBLACKBOX.

to a given planning problem affords us a choice between "equivalent" completePOPs with potentially very different characteristics from the point of view of resource allocation. In conclusion, notice that the enumeration of solutions performed by MCHAFF also ignores the resource related information in the problem instance. Indeed, these results suggest the possibility to equip the SAT solver with a more informed heuristic for solution space exploration that takes into account resource related metrics.

Causal Structure

We now turn our attention to the causal characteristics of the POPs produced by the planners considered above. As described previously, the output if the planner consists in a sequence of tasks which are then de-ordered to produce the precedence graph $\langle \mathcal{T}, \mathcal{P} \rangle$. The structural characteristics of this graph affect the scheduling phase in a number of ways, and, as noted earlier, a good predictor for the hardness of a scheduling problem is represented by the restrictiveness, which can be in turn approximated by OS.

Figure 7 shows the values of OS obtained with the four planners considered in the previous section. Again, we start by analyzing MBLACKBOX in non-enumerative mode. We can immediately notice three characteristics of the results: the presence of plateaus, the fact that OS_{π_i} for all planners except MBLACKBOX coincide on all problems, and that the values of OS appear to be scaled. These phenomena are due to the particular structure of the domain used to produce the benchmark problems.

The first two aspects can be easily explained in terms of the particular structure of the domain employed for the gen-

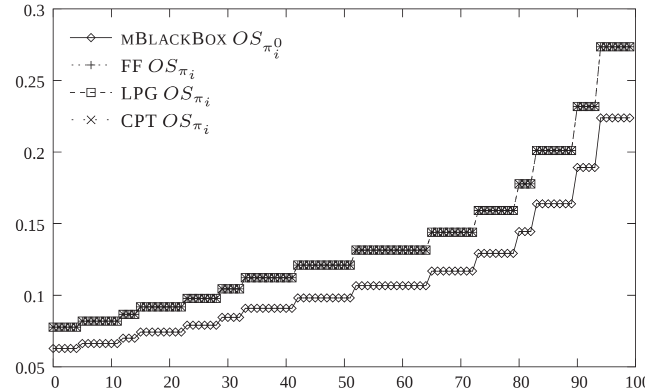

Figure 7: Order strength for the the various planners (MBLACKBOX in non-enumerative mode).

eration of the benchmark problems. In fact, the domain in question yields planning problems whose solutions have a structure which depends strongly on the size of the problem (see figure 8). In addition to the plateau effect, the results

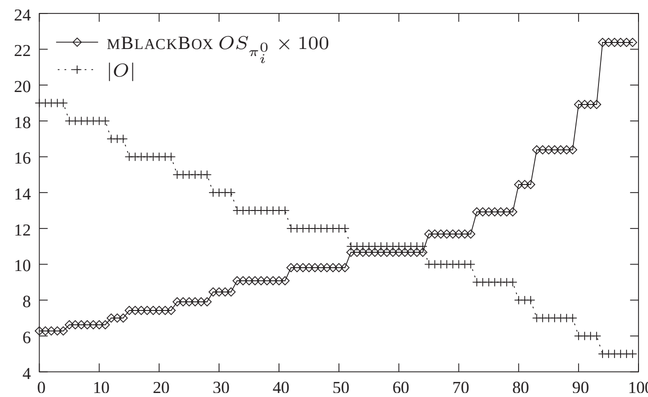

Figure 8: Dependency of the order strength on problem size in the benchmark set.

show that LPG, CPT and FF all tend to produce very similar solutions, so similar that from the strict point of view of the causal structure (i.e., not considering how resources are allocated) the plans are identical. Again, this is a direct consequence of the "simplicity" (from the planning point of view) of the domain. Indeed, further experiments show that this does not occur on more complex planning problems (such as the planning competition benchmarks), where the three planners yield solutions with values of OS that are not as easily comparable. This observation indicates that the benchmark problems considered in this paper yield a search space whose topology points all three heuristics to solutions with the same structural characteristics. Indeed, the simplicity of the domain is also the reason for the apparent scaling between the values obtained with BLACKBOX and those obtained with the other three planners. In fact, experiments on more complex domains do not yield scaled values of OS.

However, given that the benchmark set is so regular, it is somewhat unexpected that BLACKBOX obtains solutions with a different causal structure. These results seem to emphasize the difference between the SAT-solver based plan extraction mechanism and the search procedures employed by the other planners. The fact that the value of OS is lower for BLACKBOX than for the other planners confirms the results obtained in (Pecora, Rasconi, & Cesta 2004), in which we have shown, by means of a different experimental analysis, that planning graph based planners tend to produce scheduling problems which are easier than those produced by planners based on heuristic search (recall that OS is an indicator of the hardness of a scheduling problem).

A natural question at this point is the following: how does the enumeration of solutions performed by MBLACKBOX affect OS? Figure 9 shows the highest (worst) and lowest (best) order strength of the multiple solutions obtained by MBLACKBOX. An interesting aspect of these results is

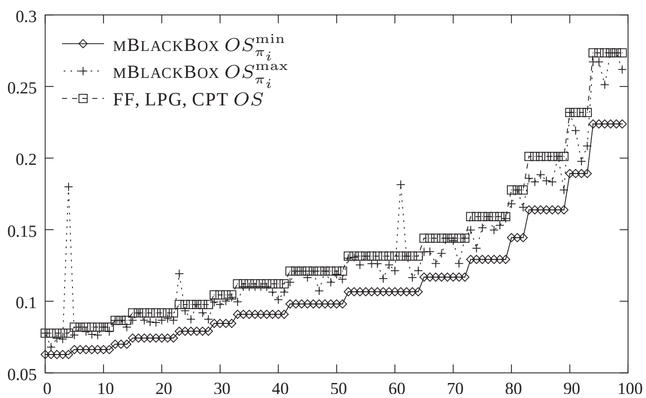

Figure 9: Order strength for the the various planners, with best ($OS^{\min}_{\pi_i}$) and worst ($OS^{\max}_{\pi_i}$) performance of MBLACKBOX.

that the best values of OS obtained by MBLACKBOX are also the first, i.e., $OS^{\min}_{\pi_i} = OS_{\pi_i^0}, \forall i$. Recall that with respect to the quality of resource allocation ($\overline{RS}$), we find that exploring different solutions to the planning problem is advantageous (see figures 5 and 6 in the previous section). Conversely, exploring further solutions does not yield an improvement with respect to the order strength, which is also increasing (i.e., there exist less different alternatives to resolve resource conflicts). This is to be expected: increasing the resource strength makes the scheduling problems more robust with respect to resource failures, but decreases the ease with which the scheduler can resolve resource conflicts.

Further Analysis of the Results

The experimental results described in this paper allow us to draw some interesting conclusions on how different planning strategies can benefit loosely-coupled P&S.

Planners as Resource Allocators. First, we have observed the characteristics of the various planning strategies on resource allocation by analyzing the average resource strength of the scheduling problems which result from the planning procedure. In this context we have seen how two of the most representative heuristic search based planners perform poorly with respect to resource allocation. Conversely, BLACKBOX and LPG, which base the search space representation on planning graphs, achieve much better results. The results point to the fact that planning graph based representations retain the intrinsic quality of favoring the extraction of plans with a higher degree of load balancing. Dually, the search strategies employed by FF[7] and CPT favor the exploration of states achieved by means of a minimal use of resources for execution. Notice that while it is true that resource capacities are ignored by the planning phase, the presence of resources for execution is not. There is no real "reason" for a planner to prefer the use of many agents rather than the strict amount necessary for causal satisfiability. Our results show that nonetheless, the planning graph data structure, which is a compact representation of the entire search space save the states pruned by mutex propagation, allows planners such as BLACKBOX and LPG to "inadvertently" provide solutions which make use of a greater number of resources for execution, thus effectively doing a better job at load balancing.

It is interesting to notice that a further uninformed exploration of the solution space achieved with the enumerative SAT-solver MCHAFF improves these results further. This makes a case for employing an enumerative planner for the purposes of loosely-coupled P&S, in which it is better to have more "equivalent" scheduling problems in the presence of invalidating run-time conditions.

The Bias of Planning on Restrictiveness. Our interest in the restrictiveness of the scheduling problems produced by the planners analyzed in this paper lies in the fact that it represents a way to put a number on the causal structure of the precedence graph which constitutes the scheduling problem. Also, measuring the restrictiveness is interesting because this measure quantifies the degree to which the planner restrains the search space of the resulting scheduling problem. If a planner obtains higher degrees of restrictiveness compared to another planner on the same problems, then this is an indication that the first planner invades the decision space of the scheduler more than the latter. Our analysis has shown that BLACKBOX retains the capability of guiding its search procedure towards a solution with lower degrees of restrictiveness, thus maintaining a lower commitment with respect to the subsequent scheduling phase. These result confirm a previous analysis (Pecora, Rasconi, & Cesta 2004) based on different structural properties of the scheduling problem, but which relied on the assumption that all resources were binary ($C_i = 1, \forall i$). The use of OS and RS has allowed us to relax this assumption.

The Tradeoff Between Robustness and Resource Efficiency. The results obtained by means of the enumera-

[7]Notice that FF employs relaxed planning graph propagation to compute the heuristic estimator. Nonetheless, the heuristic value so obtained represents only an estimate of goal distance, and does not incorporate any other information derivable from the planning graph representation.

tive planner MBLACKBOX show that exploring the neighborhood of the fist satisfying assignment yields plans with higher degrees of restrictiveness (higher values of OS), but also improved performance with respect to resource capacity limitations (higher values of RS). Indeed, this does not come as a surprise. In fact, it is reasonable to expect a tradeoff between these two characteristics of the completePOPs. On one hand, if the precedences between the tasks in the scheduling problem allow a high number of valid execution sequences (i.e., low OS), then it is reasonable to expect that the only way to safeguard against solutions which are fragile with respect to resource failures is to obtain a constraint network with higher restrictiveness in order to exclude these fragile solutions. Dually, it is straightforward to see that in order to increase the number of solutions to a P&S problem it is necessary to revert to a completePOP which potentially forebodes more resource contention peaks (lower RS), but at the same time allows the scheduler more degrees of freedom for computing a solution which makes efficient use of the resources. In the light of these considerations, we can see high restrictiveness (and resource strength) as a symptom of robustness, while low restrictiveness (which entails low resource strength) implies fragility.

It is important to keep in mind that in the context of loosely-coupled P&S, OS and RS represent characteristics of alternative causal solutions to a given P&S problem (i.e., alternative scheduling problems). We have shown that it is possible to obtain a multitude of "equivalent" scheduling problems (using different planners or an enumerative planner). If we can choose, within the context of one P&S problem, which scheduling problem to schedule for, the considerations made above become quite important. Depending, for instance, on the projected execution scenario, it may by useful to immediately select the scheduling problem which exhibits the appropriate tradeoff between (low) restrictiveness and (high) resource strength.

Conclusions and Future Work

The work described in this paper stems from the general question of how to extend a strictly causal solver with time and resource reasoning capabilities. In this context, we focus on a loosely-coupled P&S integration, that is, a framework in which the output of a classical planner is piped into a scheduler. This approach to P&S integration is suited for applicative contexts which require scheduling decisions to occur without the possibility of intervening on the planning decisions, which have already taken place and are irrevocable. Notice also that this component-driven approach is attractive because of its ease of implementation using off-the-shelf general purpose tools. In this context, the present investigation, which extends the results obtained in (Pecora, Rasconi, & Cesta 2004), is aimed at assessing the pros and cons of different planning strategies within the loosely-coupled framework.

The results described in this paper provide an assessment of how suited different planning strategies are with respect to two important characteristics of the resulting scheduling problems. We have shown how restrictiveness and resource strength, two well-known attributes of scheduling problems, provide novel and interesting quality metrics for plan quality. The results of these measurements have exposed new strengths and weaknesses of classical planners, which complement commonly accepted criteria for plan quality such as those employed in the planning competition, and may point towards new application scenarios for current AI planning technology.

We believe that the issues put forth in this paper forebode interesting new directions of research for planning, such as new planning strategies which explicitly take into account the measures described in this paper, as well as investigating other applicable measures for plan quality.

Acknowledgments. This research is partially supported by MIUR (Italian Ministry of Education, University and Research) under project ROBOCARE: "A Multi-Agent System with Intelligent Fixed and Mobile Robotic Components", L. 449/97 (`http://robocare.istc.cnr.it`). The Authors wish to thank Riccardo Rasconi and Simone Fratini for having fostered interesting common research.

References

Alvarez-Valdez, A., and Tamarit, J. 1989. Heuristic Algorithms for Resource-Constrained Project Scheduling: a Review and an Empirical Analysis. In Slowinski, R. and Weglarz, J., ed., *Intelligent Scheduling Systems*. Elsevier. 113–134.

Bäckström, C. 1998. Computational Aspects of Reordering Plans. *Journal of Artificial Intelligence Research* 9:99–137.

Brightwell, G., and Winkler, P. 1991. Counting Linear Extensions. *Order* 8:225–242.

Brucker, P.; Drexl, A.; Mohring, R.; Neumann, K.; and Pesch, E. 1998. Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods. *European Journal of Operations Research*.

Cesta, A.; Fratini, S.; and Oddi, A. 2004. Planning with Concurrency, Time and Resources. A CSP-Based Approach. In Vlahavas, I., and Vrakas, D., eds., *Intelligent Techniques for Planning*. Idea Group Publishing (to appear).

Cesta, A.; Pecora, F.; and Rasconi, R. 2004. Biasing the Structure of Scheduling Problems Through Classical Planners. In *Proceedings of the Workshop on Integrating Planning into Scheduling (WIPIS) at ICAPS, Whistler, Canada*.

Cooper, D. 1976. Heuristics for Scheduling Resource-Constrained Scheduling Projects: an Experimental Investigation. *Management Science* 22:1186–1194.

Currie, K., and Tate, A. 1991. O-Plan: The Open Planning Architecture. *Artificial Intelligence* 52(1):49–86.

De Reyck, B., and Herroelen, W. 1996. On the use of the complexity index as a measure of complexity in activity networks. *European Journal of Operational Research* 91:347–366.

Edelkamp, S., and Hoffmann, J. 2004. PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition. Technical report, Technical Report 195 Computer Science Department, University of Freiburg.

Fox, M., and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research* 20:61–124.

Gerevini, A.; Saetti, A.; and Serina, I. 2003. Planning through Stochastic Local Search and Temporal Action Graphs. *Journal of Artificial Intelligence Research (JAIR)*.

Ghallab, M., and Laruelle, H. 1994. Representation and Control in IxTeT, a Temporal Planner. In *Proceedings of the Second International Conference on AI Planning Systems (AIPS-94)*.

Ghallab, M.; Howe, A.; Knoblock, C.; McDermott, D.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL — The Planning Domain Definition Language, AIPS 98 Planning Competition Committee.

Hoffmann, J., and Nebel, B. 2001. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research* 14:253–302.

Jonsson, A.; Morris, P.; Muscettola, N.; Rajan, K.; and Smith, B. 2000. Planning in Interplanetary Space: Theory and Practice. In *Proceedings of the Fifth Int. Conf. on Artificial Intelligence Planning and Scheduling (AIPS-00)*.

Kolisch, R.; Sprecher, A.; and Drexl, A. 1995. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science* 41(10):1693–1703.

Mastor, A. 1970. An Experimental and Comparative Evaluation of Production Line Balancing Tehniques. *Management Science* 16:728–746.

Moskewicz, M.; Madigan, C.; Zhao, Y.; Zhang, L.; and Malik, S. 2001. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*.

Moskewicz, M. 2004. mChaff Website at Princeton: `http://www.princeton.edu/~chaff/mchaff.html`.

Muscettola, N.; Smith, S.; Cesta, A.; and D'Aloisi, D. 1992. Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Architecture. In *IEEE Control Systems, Vol.12, N.1*, 28–37.

Pecora, F., and Cesta, A. 2002. Planning and Scheduling Ingredients for a Multi-Agent System. In *Proceedings of UK PLANSIG02 Workshop, Delft, The Netherlands*.

Pecora, F.; Rasconi, R.; and Cesta, A. 2004. Assessing the Bias of Classical Planning Strategies on Makespan-Optimizing Scheduling. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04), August 22nd - 27th, Valencia, Spain*.

R-Moreno, M.; Oddi, A.; Borrajo, D.; Cesta, A.; and Meziat, D. 2002. Integrating Hybrid Reasoners for Planning & Scheduling. In *Proceedings of UK PLANSIG02 Workshop, Delft, The Netherlands*.

Schwindt, C. 1998. Generation of Resource-Constrained Project Scheduling Problems Subject to Temporal Constraints. Technical report, Universität Karlsruhe. Report WIOR-543.

Smith, D. E., and Weld, D. S. 1999. Temporal planning with mutual exclusion reasoning. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 326–337. Morgan Kaufmann Publishers Inc.

Srivastava, B. 2000. RealPlan: Decoupling Causal and Resource Reasoning in Planning. In *AAAI/IAAI*, 812–818.

Vidal, V., and Geffner, H. 2004. Branching and Pruning: An Optimal Temporal POCL Planner based on Constraint Programming. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-04), San Jose, CA*.