Towards an Efficient Integration of Planning and Scheduling

Antonio Garrido and Ma. de Guadalupe García-Hernández and Eva Onaindía

Universidad Politécnica de Valencia Camino de Vera s/n, 46022 Valencia, Spain {agarridot,magarcia,onaindia}@dsic.upv.es

Abstract

AI planning and scheduling processes have been traditionally hardly related to each other. However, real-world problems require capabilities of both processes. This paper presents a description of three approaches for tackling these problems: i) temporal planning approach; ii) separate approach; and iii) integrated approach. For the last approach, we provide an efficient model, as part of our ongoing work, that interleaves planning and scheduling in a flexible and general way. We also describe the key points of this approach, which are the structure and the way the two processes interact.

Introduction

AI planning community is getting more and more involved in solving realistic problems that require the use of planning techniques to supply action selection together with feasible resource assignments, such as logistic problems, crisis management, manufacturing systems, ground traffic on airports, space applications and control of satellites, etc. (Hoffmann et al. 2004). Although these problems include features of both planning (determining which actions must be executed) and scheduling (determining when and which resources must be used), planning and scheduling processes have been traditionally hardly related to each other. Particularly according to (Boddy, Cesta, & Smith 2004), "it has been recognized for some time that classical planning and scheduling models are at opposite ends of a spectrum, with most interesting real-world problems falling somewhere in the middle and requiring characteristics of both". However, dealing with planning and scheduling in a disunited way does not seem sensible: selecting an action in a plan is usually conditioned to several temporal constraints, resource availability and criteria to be optimised. Consequently, planning and scheduling complement each other perfectly, particularly considering that they use similar techniques (search in graphs, use of heuristics, management and reasoning of constraints, etc.) (Smith & Zimmerman 2004; Smith, Frank, & Jónsson 2000).

Recent advances in planning research (planning graphs, local search, heuristic techniques, satisfiability and constraint satisfaction, etc.) allow to deal with a more realistic planning model that handles time, resources and multioptimisation criteria, thus incorporating more (scheduling) capabilities to planners for solving real-world problems. However, as the problems become harder and more complex, the difficulties to solve them become harder as well, and complexity in planners is growing dramatically (Ghallab, Nau, & Traverso 2004; Garrido & Onaindia 2004). Relaxation of numeric features in actions, thus disjoining the planning part from the scheduling one by separating the structural (propositional) part of the plan from the part dedicated to resource usage (numeric variables and resource management), is also possible (Garrido, Onaindia, & Hernandez 2005; Halsey 2004). The main idea is to make a clear distinction between sketching the structure of the plan (reasoning on conditions, effects, orderings and causal links) and fulfilling the numeric constraints (reasoning on time and numeric variables that usually involve resource usage). Nevertheless, there is an important drawback in class of problems with a strong dependency on the numeric conditions (Garrido, Onaindia, & Hernandez 2005). In these problems, actions to support the numeric conditions can only be applied in very particular cases, i.e. the numeric features are an intrinsic part of the original problem and do change the structure of the plan (there exists a complex interaction of numeric features that cannot be abstracted during plan generation). Let us imagine a rovers¹ application scenario, where rovers can only be recharged at points with sun (actions to recharge are very limited), and trying to separate the navigation and communication part from managing energy levels and recharging is nearly impossible. This is a clear example of a plan that needs to take into consideration the numeric features as they modify its structure; i.e. the plan generation is highly influenced by the numeric constraints and making a clear distinction between planning and scheduling turns unlikely.

This paper presents an evolution of different approaches for tackling planning and scheduling problems. First, we start with a temporal planning approach that follows the same ideas of most state-of-the-art planners that have participated in last international planning competitions. Second, we continue with a more modern approach that tries to simplify the original problem to make it easier and, consequently, more affordable by means of a relaxation of

¹See http://ipc.icaps-conference.org, International Planning Competitions (IPC-2002 and IPC-2004), for more information about this and other problems and domains

numeric variables on actions (separation of planning and scheduling). Finally, we describe some ideas to perform (what we understand as) an efficient integration of planning and scheduling, as a model that combines techniques of both planning and scheduling in an dynamic interleaved way. In this paper, we analyse the main features of each approach, and present the advantages and disadvantages, based on our own experience and current work.

A planning/scheduling problem from a planning perspective

Nowadays, many usual activities involve the execution of a sequence of actions, which must satisfy several constraints (both temporal and on resource availability), in order to achieve some goals, while trying to optimise a metric function defined on the problem. This kind of problem represents a planning and scheduling problem, which is hardly separable into two disjunctive parts. Hence, we define a planning and scheduling problem as the tuple $\mathcal{P}_{ps} = \langle \mathcal{I}, \mathcal{G}, \mathcal{A}, \mathcal{R}, \mathcal{C}, \mathcal{M} \rangle$, where each element means:

- \mathcal{I} = Initial state, with all the information that is true at the beginning of the problem.
- \mathcal{G} = Goals, with all the facts that must be achieved at the end of the problem.
- \mathcal{A} = Actions, defined on the problem domain that allow to achieve \mathcal{G} from \mathcal{I} .
- \mathcal{R} = Resources, available to execute the actions in the plan.
- C = Temporal constraints, with additional constraints the plan must satisfy.
- \mathcal{M} = Metric function, as a multi-criteria function that needs to be optimised.

To the authors' knowledge there is not a well accepted language to model real problems of planning and scheduling. However, from a planning perspective there exists a widely accepted language to define planning domains which is called PDDL (McDermott 1998). PDDL was designed as a common framework to define planning problems and test progress in planning techniques. PDDL has now evolutioned to PDDL2.1 and PDDL2.2 (Fox & Long 2003; Edelkamp & Hoffmann 2004), but none of them allows to explicitly define resources \mathcal{R} or temporal constraints \mathcal{C} . On the one hand, resources are *artificially* modelled as other objects in the problem, and are implicit in the definition (conditions and effects) of each action. On the other hand, temporal constraints to represent hard constraints among actions, finite persistence of action effects, deadlines for goals, etc. are not considered and difficult to be included in the problem. Fortunately, in addition to the initial state \mathcal{I} and goals \mathcal{G} , now in PDDL2.2 we can easily model: i) actions with duration (:duration (= ?duration (boarding-time ?a))); ii) actions with local conditions and effects ((at start (at ?p ?c)), (over all (at ?a ?c))); iii) actions with numeric features, conditions ((>= (fuel ?a) (* (distance ?c1 ?c2) (slow-burn ?a)))) and

effects ((assign (fuel ?a) (capacity ?a))); iv) multi-criteria problem metrics to optimise (:metric minimize (+ (* 0.5 (total-time)) (*

0.02 (total-fuel-used)))); and v) timed initial literals that express deterministic unconditional exogenous events and allow to represent a kind of temporal constraints in the form of time windows (:init (at 9 (shop-open)) (at 20 (not (shop-open)))).

Although there still exists a lack of capabilities in the planning domain definition languages to describe scheduling features such as resources, deadlines and complex temporal constraints, improvements in the expressivity of PDDL have done realistic planning and scheduling problems come within reach (Hoffmann *et al.* 2004). Therefore, most features of real-world planning/scheduling problems can be currently modelled by PDDL2.x, at least from a planning point of view.

Temporal planning (and scheduling) approach

When dealing with planning and scheduling problems, a straightforward approach is to push beyond classical planning assumptions to incorporate scheduling constraints (duration on actions, numeric features and multi-criteria metrics). Although this makes the planning solving process extremely more complex, progress in planning techniques have reached a great success as some planners such as SGPlan (Chen, Hsu, & Wah 2004), LPG-TD (Gerevini *et al.* 2004) or Mips (Edelkamp 2002) demonstrated in last planning competitions (Fox & Long 2003; Edelkamp *et al.* 2004).

Basically, the structure of a temporal planner is depicted in Fig. 1. Starting from the domain and problem definition (for most modern planners this is defined in PDDL2.x), the temporal planner extends classical planning techniques to reason on time, mainly on action duration, (start, invariant and end) conditions and timed initial literals. In order to find a plan, different types of techniques have been developed to improve search; planning/action graphs, estimations based on relaxed plans, resolution of global constraints, goal ordering, search space reduction, etc. are usually used to help in action selection. As can be thought, many of these techniques are action-based heuristics, but more powerful temporal planners calculate some cost for actions to be able to optimise the plan according to a multi-criteria metric. This way, the final plan, which is executable according to the problem constraints, provides a good value for the problem metric (though only a few planners can guarantee optimality).

This approach is currently one of the most widely used; most planners that participated, and were awarded as top performers, in last planning competitions worked under this approach. The main reason for this lies in the natural extension of planning algorithms to support new capabilities. Basically, the main advantages of this approach are twofold:

• The solving process turns into a homogeneous planning and scheduling system, and there is little distinction between action selection and ordering and resource usage decisions. Thus, the scheduling process does not need to



Figure 1: Outline of the temporal planning approach.

be very expressive or efficient; scheduling is just an intrinsic component of a broader planning process.

• The resulting plan is executable and does not need a postprocessing stage to check inconsistencies on resource availability and/or temporal constraints. Intuitively, this means that such a plan is straightly what a real problem requires as a solution.

On the contrary, this approach presents some important disadvantages:

- Temporal reasoning performed in the approach is adequate but slightly limited. Complex temporal constraints (such as disjunctive or metric constraints) on plans, actions and resources is quite complex without a specific temporal manager or scheduler.
- It is difficult to determine when the system is planning or scheduling (Garrido & Barber 2001). Since the system is a tight combination of planning and scheduling, it makes it much more difficult (or impossible) to find heuristic criteria to improve planning and scheduling separately.
- The extension of planning algorithms to deal with scheduling features makes these algorithms extraordinarily complex. Although currently some planners can cope with this complexity, it is obvious that the performance of planners cannot fit with such an increase in the complexity of the problems. This imposes a hard limit in planning scalability and jeopardises the future of planning in its application to real-world problems. This is shown in Fig. 2, which compares the runtimes of LPG-TD.quality (Gerevini et al. 2004) and SGPlan (Chen, Hsu, & Wah 2004) when solving STRIPS vs. some numeric problems. As can be seen, the runtimes are significantly different. This helps demonstrate how the complexity of dealing with scheduling features increases the complexity of the planners and their runtimes in several orders of magnitude in some cases, and even makes them unsolvable in others.



Figure 2: Comparison of two awarded temporal planners (LPG-TD.quality and SGPlan) solving STRIPS *vs.* numeric problems in rovers and satellite domains. All tests were censored after 300 s.

Separate approach for planning and scheduling

As indicated in previous section, one of the main disadvantages of a temporal planning approach is the limit that it imposes in planning scalability: the complexity of these planners in very difficult problems may become intractable. An intuitive modification to that approach is to remove part of the capabilities (and complexity) from the planning process and use a specific process to perform scheduling tasks, i.e. to make use of a separate approach for planning and scheduling with two different processes. A historical approach for this consists in drawing a line between planning and scheduling, where planning precedes scheduling (Garrido & Barber 2001). First, the planner generates a plan by the application of actions from the domain, just considering causal links. Second, the scheduler validates the problem constraints, considering action durations and allocating resources for the entire plan. More recent works based on this philosophy use a new kind of relaxation that separates the structural (propositional) part of the plan from the part dedicated to resource usage (numeric variables and resource management) (Garrido, Onaindia, & Hernandez 2005; Halsey 2004).

Fundamentally, this approach (see Fig. 3) works by separating the initial domain and problem features into two disjunctive components. On the one hand, planning features are managed by a classical planner that deals with propositional actions (no duration, local conditions/effects or numeric features are considered) under a STRIPS-based model. This way, the output of this process is a sequential plan that is propositionally sound (the plan would be directly executable in a scenario without numeric variables such as a pure-STRIPS planning problem) (Garrido, Onaindia, & Hernandez 2005). On the other hand, a very simple scheduling pro-



Figure 3: Outline of the separate planning+scheduling approach.

cess receives as an input the sequential plan and incorporates numeric features, thus satisfying the numeric conditions and constraints while parallelising the actions in the plan. Obviously, each planning and scheduling process uses specific heuristics mainly based on actions and problem metric, respectively. The final result is, consequently, an executable plan.

The main advantage of this approach is the possibility to abstract out planning tasks from scheduling ones, devoting the planning effort in the efficient construction of a plan and to use a specific scheduler to check whether the plan is schedulable. Moreover, if the problem is propositionally unsolvable, i.e. there are some propositional (sub)goals that cannot be satisfied, this approach does not need to perform any scheduling task. There exist, however, some drawbacks:

- Lack of cooperation between the two processes. Both processes work independently with no relation or knowledge on each other during its execution.
- Lack of global optimisation criteria. The planner returns a sequential plan that, after being parallelised, may provide a bad value for the problem metric. Unfortunately, this is not unusual since the heuristics used during planning do not take problem metric information into consideration, and the scheduling process does not have the capability to change the plan substantially.
- The sequential plan might not be properly parallelised by the scheduling process. For instance, let us suppose that the available energy level for a rover is scarce and no recharge action is available, i.e. a numeric constraint cannot be satisfied. If the planner returns a plan where the rover runs out of energy while navigating (note that the

Figure 4: Two resulting plans for a problem of the rovers domain: (a) pure-STRIPS sequential plan, and (b) parallel plan that satisfies the numeric constraints (time and resources).

planner does not check resource constraints), the scheduler will not be able to make the plan executable. Again, this situation is not unusual. Many problems have a strong dependency on the numeric features since they are an intrinsic part of the problem (propositional and numeric part are strongly coupled). This means that the scheduling constraints must be verified immediately after each action is planned; the plan highly depends on such constraints as they can change its structure.

The last drawback is the gravest, so we will illustrate it with an example on the rovers application scenario (see IPC-2002 for more information). Let us assume a problem where there are two samples to find (rock and soil) and communicate them back to a lander. Initially, rover0 has enough energy to start to operate, but not enough to finish the plan, and there is only one waypoint (wp3) where recharge is possible. Fig. 4-a shows the sequential plan that the planner of this approach returns. As can be seen, this plan is propositionally executable, but since no reasoning on resources has been done, the plan does not contain any action that visits wp3. Next, the scheduler starts to parallelise actions satisfying numeric conditions. The problem appears when the rover reaches an energy level that prevents it from executing any action, even a navigate action to wp3 to recharge. Since the scheduler cannot change the structure of the plan (add or delete actions according to causal links), it will fail to find an executable plan, though the executable plan (see Fig. 4-b) is nearly the same and it contains all the five actions present in the sequential plan. This clearly demonstrates that working with planning and scheduling in a disunited way may fail in many problems, and an integrated approach with a much stronger relation between both processes to deal with a broader quantity of problems is necessary.

Integration of planning and scheduling

Designing an integrated approach for planning and scheduling is not an easy task. The solution does not necessarily imply an embedded planning process into a scheduling one, or vice versa, as we will discuss in section . An efficient



Figure 5: Outline of the integrated planning and scheduling approach.

integrated model needs to be general and flexible, where both planning and scheduling processes play a similar role, and where the key points are the definition of the next two items: i) structure of the integrated approach, and ii) the way in which planning and scheduling processes cooperate and communicate to work under an integrated way.

Structure of the integrated approach

Fig. 5 shows the structure of our integrated approach. Obviously, the integrated module requires the domain and problem definition as an input. Moreover, a plan (either provided by the user as a set of activities or by a planner as a STRIPS parallel or sequential plan), is used as an additional input, which is later motivated. The integrated module contains an action network (AN) converter that transforms the input plan into a network that represents the plan with its actions, causal links and (temporal+resource) constraints. The planning process updates the action network to make actions propositionally executable; i.e. it mainly works as a replanner, repairing (adding or deleting) actions in the network when they cannot be executed. The scheduling process mainly works as a validator of constraints and checks the feasibility of the action network and its constraints. However, the scheduler's task is not only to validate the action network, but also to inform the planning process about the conflicting actions and the resources involved in such conflicts. This way, the scheduler helps the planner repair the plan by means of integrated heuristics that allow both processes to take common agreement decisions. Finally, an executable plan is extracted from the action network that has been generated by using planning and scheduling criteria.

When defining the structure, there are two elements that are essential to establish the foundations of the integrated module: the former concerns the input of the integrated module, whereas the latter concerns how the action network represents the plan, actions and their constraints. The input of the module needs to include the domain and problem definition. At this point, the integrated module could start from an empty plan, i.e. generating a plan from scratch. However, this does not seem to be sensible because of the high complexity that it entails. Nowadays, there are many state-of-the-art planners that generate plans in an efficient way (Fox & Long 2003; Edelkamp et al. 2004). Therefore, it seems much worthier starting from an initial plan as the basis (like in the separate approach for planning and scheduling, but now with the two processes working together in an interleaved way). The underlying idea is to use a classical planner, as simple (in terms of expressivity and calculus) and efficient as possible. This will allow us to use a pure-STRIPS plan provided by any planner. Note that this does not limit the approach: a relaxed plan that ignores the delete effects of actions (Bonet & Geffner 2001; Hoffmann & Nebel 2001) can also be used as an input, and even a plan generated by hand, making the conversion into an action network even simpler in both cases. Also note that this plan does not need to be propositionally executable since the replanner can repair unsupported conditions. This increases the opportunities to use efficient planners as a previous step to the integrated module.

The action network (see Fig. 6) follows the philosophy of temporal constraint networks (TCN (Dechter, Meiri, & Pearl 1991)) to represent actions and constraints. Nodes represent timepoints where actions start/end (.on/.off, respectively). There are three types of edges, all of them labelled with an interval, that represent: i) usage of a resource (time is considered as other resources) between the .on/.off timepoints of the same action; ii) causal links between timepoints of actions; and iii) temporal constraints between timepoints. For instance, in Fig. 6, action (calibrate rover0 cam0 obj wp0) increases time in 5 units (action duration) and decreases energy of rover0 in 2 units (note that these values can be also defined by an interval). Action (calibrate rover0 cam0 obj wp0).off has a causal link with (take_img rover0 wp0 obj1 cam0 low_res).on because the first action generates a proposition at end that the second action needs over all its execution. This definition of causal links between timepoints is an efficient way to represent causal links between local effects and conditions. Finally, temporal constraints allow to define additional constraints and deadlines between the execution of actions and the beginning of time: (sample_soil rover0 rover0st wp0) must finish before time 50.

Planning and scheduling processes working together

An interesting and compelling question for an efficient integration of planning and scheduling lies in the way in which these processes are interleaved. Here we outline a tentative working scheme as an attempt of creating a highly-coupled integration approach. Both processes are constantly interleaved during the process of creating a plan that satisfies the



Figure 6: Example of an action network. Solid and dashed lines represent usage (increase/decrease) of resources in actions (time and energy of rover0, respectively); thick lines represent causal links; and dotted lines represent temporal constraints. T0 identifies the beginning of time.

temporal and resource constraints. The key point of this approach, as indicated previously, is that the scheduler does not merely work as a validator but as a helpful source of information to help the planning process. Following, we describe the way both processes work together, along with its flow chart diagram (see Fig. 7). First, the planning tasks (propositional part of the plan) are:

- 1. Check propositional executability. The planning process selects a set of current actions from the action network. Next, it checks whether the actions are propositionally executable.
- 2. **Repair process**. This is the main task of the planning process, consisting in repairing a portion of the plan to make it executable. There are two identifiable subtasks:
- (a) Repair action preconditions. The reparation process may imply to repair a single action condition (flawrepair) or a set of conditions. During the reparation process all actions in the action network are considered and all necessary actions to satisfy the unsupported preconditions are inserted.
- (b) Remove actions from the action network. Another task within the planning module is to remove one or several actions from the network. This is a relevant activity since the current plan might not satisfy the time/resource constraints and the solution necessarily goes through the deletion of some actions in the network. Obviously, the recommendation on which actions should be removed is given by the scheduling process. In this case, a portion of the plan could be removed and the planning process will have to repair the plan to make it executable again.

Second, the two main tasks the scheduler undertakes are:

1. Check time/resource constraints + temporal action allocation. This task consists in checking the satisfaction



Figure 7: Flow chart diagram for the tasks during planning and scheduling interaction.

of all time and resource constraints in the problem. The scheduling process carries out this task to find out whether there exist over-constrained resources on the current portion of the plan under work while satisfying the temporal constraints. When there is no precise information on the temporal allocation of a particular action and, as long as it is possible, the scheduler applies a least-commitment temporal scheduling. However, at some point (when checking time/resource constraints) it may become necessary to supply a more precise temporal allocation for an action in order to check the exact resource consumption. For example, if it is known that action a is placed after b, then the scheduler will keep a temporal constraint $[0,\infty]$ between a and b. At the time of checking constraints, the scheduler must have more precise information about how long b is after a in order to detect the exact time point or interval where the problem runs out of a resource. In other words, it is necessary to know the planning state over which the action is going to be executed in order to check the constraint satisfaction.

2. Find conflicting actions and inform the planning process about them. This is the most complex task of the scheduler. If the scheduler detects that some resource constraints are violated, it will calculate (heuristic application) the most likely actions involved in the overconstrained resources (no-goods), such as the most consumable actions or the minimal action set involved in the resource conflict. This information will be provided to the planning process along with some recommendations for each action.

Note that in our approach, resources are directly allocated by the planning process under the recommendations of the scheduling process. Hence, if the replanner inserts an action to transport a package from city1 to city2 with truck1 and the scheduling process later discovers the fuel consumption exceeds the available amount, it might suggest to remove such an action because of its high consumption. Therefore, once the action is removed the planning process will have to find a different alternative to have the package at city2, which might involve using a different truck. The same can be applied to allocate continuous resources. The scheduling process might suggest not deleting the drive action but just indicating that there is a problem with the fuel resource in such an action. In this case, the replanner will opt to apply a flaw-repair routine to provide support to the numerical precondition of the action. In conclusion, resources are allocated by the planning process because they are invariantly attached to the execution of actions. However, the decision about which resource to use or when to supply more of a resource will be suggested by the scheduling process. An open point here is the improvement in the information flow between the scheduler and the planner; i.e. the scheduler could inform the planner about the actions that could be removed from the plan or advice the planner to help it make future action choices, such as warning the planner to avoid actions with over-subscripted resources.

Conclusions through related work

The separation of planning and scheduling processes when solving real-world problems has shown important drawbacks that are difficult to be solved without a combined approach of planning and scheduling. In this line of research, we can find two different perspectives. From a planning point of view, a planner is extended to handle time and resources (temporal planning approach) (Ghallab, Nau, & Traverso 2004; Chen, Hsu, & Wah 2004; Gerevini et al. 2004; Edelkamp 2002). From a scheduling point of view, a scheduler is extended to course some dynamic action selection (planning capabilities embedded within scheduling) (Smith & Zimmerman 2004). In the former, an indistinguishable mixture of planning and scheduling is achieved, where action and resource allocation decisions are considered while planning. This mixture of processes may increase the complexity of the overall approach, making problems with a strong scheduling component practically intractable. In the latter, planning is an adjunct to scheduling, i.e. scheduling borrows some of the conventions and objectives of planning. This approach requires as input demands specific top-level goals, a set of activities and probably the order in which these activities must be applied. Therefore, the planning component is already given to the problem. Hence, the main drawback of this approach is that planning is only considered as a subtask of the scheduling process, used to find the resource setup activities. Moreover, the argument for leveraging strengths of both scheduling and planning processes remains compelling.

In the last decade, the way in which planning and scheduling must be combined has been addressed as an interesting question and a hot topic of research (Muscettola 1994; Chien *et al.* 2000; Srivastava, Kambhampati, & Do 2001; Rodríguez 2003). Although some attempts have been successful in the domains they were designed for, such as HSTS (Muscettola 1994) or Aspen (Chien *et al.* 2000), this success cannot be easily demonstrated in other domains, i.e. they were designed for particular problems and consequently they cannot be considered as general integrated models of planning and scheduling. On the contrary, the integrated approach presented in this paper tries to be a general, flexible model that dynamically interleaves planning and scheduling processes (both playing a similar role) to tackle planning problems with a strong and weak component of resource/time management. Our approach combines both processes while benefits from their knowledge by separate, such as use of STRIPS or relaxed plans, plan execution and reparation, definition of timepoints, resource allocation, temporal constraint management, etc. Further, it also incorporates the use of common integrated heuristics, reasoning on an action network, and a clever interaction of the planning and scheduling processes, where the main issues are: i) the structure of the integrated approach (and its input as any kind of plan), and ii) the way in which planning and scheduling processes interact (cooperation and communication, where both processes help each other).

Acknowledgments

This work has been partially supported by the Spanish government CICYT projects TIC2002-04146-C05-04 and DPI2001-2094-C03-03, by the Valencian government project GV04A-388 and by the Universidad Politécnica de Valencia under project 20020681.

References

Boddy, M.; Cesta, A.; and Smith, S., eds. 2004. *Proc.* of the ICAPS-2004 Workshop on Integrating Planning into Scheduling.

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence* 129:5–33.

Chen, Y.; Hsu, C.; and Wah, B. 2004. SGPlan: Subgoal partitioning and resolution in planning. In *Proc. Int. Conf.* on Automated Planning and Scheduling (ICAPS-2004) – International Planning Competition, 30–32.

Chien, S.; Rabideau, G.; Knight, R.; Sherwood, R.; Engelhardt, B.; Mutz, D.; Estlin, T.; Smith, B.; Fisher, F.; Barrett, T.; Stebbins, G.; and Tran, D. 2000. ASPEN - automating space mission operations using automated planning and scheduling. In *Proc. SpaceOps 2000*.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.

Edelkamp, S., and Hoffmann, J. 2004. PDDL2.2: the language for the classical part of IPC-4. In *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS-2004)* – *International Planning Competition*, 2–6.

Edelkamp, S.; Hoffmann, J.; Littman, M.; and Younes, H., eds. 2004. *Proc. of the International Planning Competition IPC–2004*.

Edelkamp, S. 2002. Symbolic pattern databases in heuristic search planning. In *Proc. 6th Int. Conference on AI Planning and Scheduling (AIPS-2002)*, 195–204. AAAI Press.

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.

Garrido, A., and Barber, F. 2001. Integrating planning and scheduling. *Applied Artificial Intelligence* 15(5):471–491.

Garrido, A., and Onaindia, E. 2004. On the separation of propositional and numeric planning in realistic problems. In *Proc. 23rd UK Planning and Scheduling SIG Workshop*, 56–65.

Garrido, A.; Onaindia, E.; and Hernandez, D. 2005. A preliminary study on the relaxation of numeric features in planning. In *Frontiers in Artificial Intelligence and Applications*, volume 131, 355–362. IOS Press.

Gerevini, A.; Saetti, A.; Serina, I.; and Toninelli, P. 2004. Planning in PDDL2.2 domains with LPG-TD. In *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS-*2004) – International Planning Competition, 33–34.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning. Theory and Practice*. Morgan Kaufmann.

Halsey, K. 2004. The workings of CRIKEY – a temporal metric planner. In *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS-2004) – International Planning Competition*, 35–37.

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

Hoffmann, J.; Edelkamp, S.; Englert, R.; Liporace, F.; Thiebaux, S.; and Trug, S. 2004. Towards realistic benchmarks for planning: the domains used in the classical part of IPC-4. In *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS-2004) – International Planning Competition*, 7–14.

McDermott, D. 1998. *PDDL - The Planning Domain Definition Language*. AIPS-98 Planning Competition Committee.

Muscettola, N. 1994. HSTS: Integrating planning and scheduling. In Zweben, M., and Fox, M., eds., *Intelligent Scheduling*. San Mateo, CA: Morgan Kaufmann. 169–212.

Rodríguez, M. 2003. *Representing and Planning tasks with time and resources*. Ph.D. Dissertation, Universidad de Alcalá de Henares.

Smith, S., and Zimmerman, T. 2004. Planning tactics within scheduling problems. In *Proc. ICAPS-2004 Workshop Workshop on Integrating Planning Into Scheduling*, 83–90.

Smith, D.; Frank, J.; and Jónsson, A. 2000. Bridging the gap between planning and scheduling. *Knowledge Engineering Review* 15(1):47–83.

Srivastava, B.; Kambhampati, S.; and Do, M. 2001. Planning the project management way: Efficient planning by effective integration of causal and resource reasoning in RealPlan. *Artificial Intelligence* 131:73–134.