Planning - Scheduling Connections through Exogenous Events

Minh B. Do & Subbarao Kambhampati
Department of Computer Science and Engineering
Arizona State University, Tempe AZ 85287-5406
{binhminh,rao}@asu.edu

Terry Zimmerman
Carnegie Mellon University, Robotics Institute
5000 Forbes Avenue, Pittsburgh, PA 15213
wizim@cs.cmu.edu

Abstract

Exogenous events appear in many practical planning and scheduling problems but until recently only the latter methodology has explicitly dealt with them. While exogenous events are commonly viewed as instantaneous actions occurring at specific time points and interacting with actions, they can sometimes be seen as enforcing *earliest starting time* and *latest finishing time* constraints on actions. Such constraints are common in scheduling, suggesting that exploiting related techniques from the scheduling community might be effective in a planning context. In this paper, we look at heuristic techniques from both the planning and scheduling fields that are relevant to this problem, and consider combined approaches that may be more effective when planning in the presence of the type of constraints imposed by exogenous events.

Introduction

An exogenous event can be defined as a happening that occurs at a specific point along a conceptual time-line and changes the world state by altering the value of one or more state variables. In the context of automated planning, exogenous events may affect the planning process by either enabling or interfering with the execution of certain domain actions. In this sense, they impose additional time constraints on action execution.

Examples of exogenous events and their potential interactions with actions in a plan are:

- Electricity will be available at 10 AM. Thus, actions that need electricity will only be able to execute after 10 AM.
- The store will be closed starting next month. Purchase actions requiring the store cannot be executed thereafter.

Exogenous events can also occur in counteracting pairs and make up time windows. Some examples of time windows are:

- A satellite can communicate with an earth station only between 8-10 AM or 9-11 PM.
- At a certain waypoint X on Mars, it's *sunny* (thus allowing the Rover to recharge its battery at that location) between 10-11 AM.
- The bus operates between ASU and downtown Phoenix in the period between 5 AM and 11 PM.

Automated planning deals explicitly with causal and interference relations between actions, so a natural interpretation of exogenous events is as instantaneous actions that interact with actions in the plan. From the scheduling point of view, a subclass of exogenous events (especially in the case of time-windows) can be seen as providing the earliest starting time (est) and latest finishing time (lft) constraints to activities in the plan. The est and lft constraints are fundamental to scheduling, suggesting that effective handling of exogenous events in temporal planning problems might benefit by combining techniques from both the planning and scheduling communities.

A critical component in the performance of any given planner is often the quality of the heuristic(s) guiding its search. Many of the top-performing planners such as HSP (Bonet & Geffner 1997), FF (Hoffmann & Nebel 2001), AltAlt (Nguyen et. al. 2001), Sapa (Do & Kambhampati 2003) use heuristic estimation based on propagating reachability information either directly or via the planning graph structure (Blum & Furst 1997), then follow up with a second phase that extracts a "relaxed plan" from the graph. By ignoring negative interactions between actions, the relaxed plan idea quickly produces approximate (but not provably correct) plans for achieving candidate goal sets. In this paper we discuss how the interactions between exogenous events and other actions in the plan affect this two-phase heuristic framework. Given the two different views of exogenous events (instantaneous actions vs. est/lft constraints) we also discuss how effective heuristic techniques in scheduling such as slack-based (Smith & Cheng 1993) or texture-based (Beck et. al. 1997) can be used to improve the quality of the relaxed plan based heuristic extracted from the planning graph. Thus, this paper examines how scheduling techniques can be adapted to improve planning heuristics and performance in the presence of exogenous events.

The remainder of this paper is organized as follows: We start with a view of exogenous events as instantaneous actions that should be included in any plan and how they affect the current planning search framework, especially the heuristic estimation process. We then look at different approaches of using scheduling techniques to improve relaxed-plan based heuristics guiding the planners. We end with a section on future work and our conclusions.

Representing Exogenous Events

In general an exogenous event might change the value of a state variable or a function value (e.g. room temperature), or it might maintain some state variable or function value during some period (e.g. fuel level should be kept higher than 5 during $[t_1, t_2]$). In this paper, we only concentrate on exogenous events that change the value of certain state variables from true to false or vice versa. In the PDDL2.2 language for expressing temporal planning domains (to be used in this year's International Planning Competition (IPC4)), an exogenous event is referred to as *timed initial fact*. Examples of PDDL2.2 exogenous event syntax include: *(at 10 (open-station city0))* and *(at 20 (not (open-station city0)))*. At an abstract level, an exogenous event is represented as a tuple $e = \{p, o, t\}$ in which p is the predicate affected by e, $o = +/-$ is an operation on p, and t is the time point at which e occurs. If $o = +$, then p changes value from $false$ to $true$; if $o = -$, then p changes from $true$ to $false$. Combinations of two counteracting instantaneous events that happen at different time points t_1 and t_2 constitute a *time window*. They may occur repetitively, as in the case of the visibility time windows for satellite communication and bus schedules.

Normally, exogenous events affect state variables whose values cannot be changed by actions in the domains. For example, the agent (planner) is not likely to be able to change the satellite's communication time windows, which is decided by the geometric position alignment between the satellite and the communication center. Similarly, the planner is not likely to be able to change the time when tickets are on sale or the time schedule of the bus. This type of exogenous events has been discussed in the HSTS planning/scheduling system (Muscettola et. al. 1992) and was refered to as non-controllable state variables. However, throughout this paper, we make no assumption concerning the planners ability to change the state variables affected by exogenous events. Nevertheless, if the problem only involves exogenous events that do not affect state variables changed by other actions, then fewer constraints (action interactions) are involved.

Example: We will take here the sample problem provided by the organizers of the IPC4 in which we need to move passengers between different cities (ZenoTravel domain) and have airplanes that can fly at two different speeds. Actions in the domain are: $board(person, airplane, city)$, $debark(person, plane, city)$, $fly(plane, city1, city2)$, $zoom(plane, city1, city2)$ (fly faster), and $refuel(plane, city, flevel1, flevel2)$.

Exogenous events (or timed-initial facts) are represented as follows:

e_1: (at 25 (open-station city0))
e_2: (at 75 (*not* (open-station city0)))
e_3: (at 275.02 (open-station city1))
e_4: (at 375.03 (*not* (open-station city1)))
e_5: (at 475.05 (open-station city2))
e_6: (at 575.06 (*not* (open-station city2)))

These exogenous events impact the $refuel$ actions, which are described as follows:

```
(:durative-action refuel
:parameters ( ?a - aircraft ?c - city ?l ?l1 - flevel)
:duration (= ?duration 50)
:condition
    (and (at start (fuel-level ?a ?l)) (at start (next ?l ?l1))
    (over all (at ?a ?c)) (over all (open-station ?c)))
:effect
    (and (at end (not (fuel-level ?a ?l)))
    (at end (fuel-level ?a ?l1)))))
```

The description shows that exogenous events are represented as instantaneous actions that change the values of predicates $(openstation\ cityX)$. Positive events (e_1, e_3, e_5) appear to satisfy a precondition of $refuel(aircraft, cityX)$ and negative events (e_2, e_4, e_6) delete a precondition and prevent it from executing.

From a different point of view, pairs of the exogenous events can be seen as defining the $est - lft$ time windows familiar in scheduling. Thus, in this case, we can discard the precondition $(overall\ (open - station\ ?c))$ from the description of the $refuel()$ action along with the exogenous events and replace them with temporal constraints such as: $est(refuel(aircraft, city0)) = t_{e_1}$ and $lft(refuel(aircraft, city0)) = t_{e_2}$. From this perspective, the possibility of using techniques from the scheduling community, where those constraints are very common, becomes more apparent.

Heuristic Estimation with Exogenous Events

Most of the current automated planners find a valid plan by incrementally selecting and adding actions to a partial plan which is initially empty (contains no actions). Syntactically, exogenous events are like instantaneous actions that change the values of state variables. However, unlike other actions in the problem, the planner has no choice over whether or not to select an exogenous event. There are at least two feasible approaches to handling such events: (i) Compile the exogenous events down to normal actions in a manner that forces the planner to choose all exogenous events related to actions needed to reach the problem goals; (ii) Reason explicitly about exogenous events and extend the notion of initial state or initial plan to include them. With respect to the first approach, Fox & Long (2003) present polynomial transformations to compile down exogenous events into normal actions in PDDL2.1. An extension of the LPGP planner (Cresswell & Coddington 2003) implemented this approach.

Alternatively, the second approach entails extending the capability of a planning system to reason directly with exogenous events, and we argue that this is the more promising of the two. Here the planner is provided a non-empty initial plan containing fixed-time actions representing the exogenous events. We have extended the initial state/plan representation in the forward temporal planner *Sapa* to include exogenous events and the planner is able to solve the sample example problem discussed above in roughly 10 seconds on a Pentium IV machine. Nevertheless, without any modification to its heuristic estimation, the planner searches through more than 22000 nodes before succeeding. It is evident that solving larger problems will quickly become infeasible unless the quality of the heuristic estimates guiding the planner can be improved to account for exogenous events. We con-

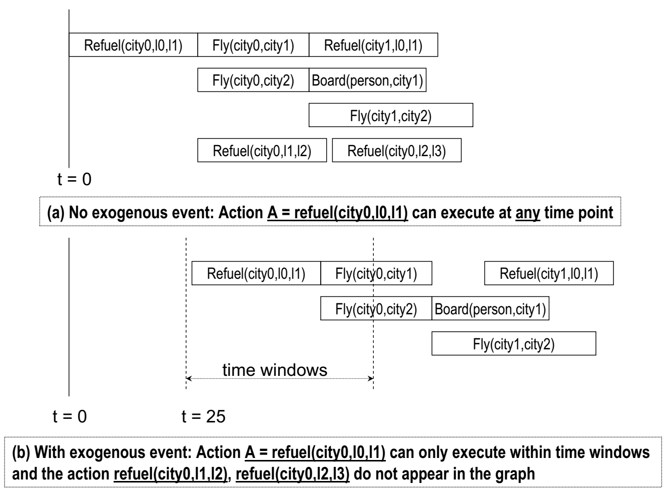

Figure 1: Temporal planning graph with/without exogenous events

sider here the effects of exogenous events on the heuristic approach of extracting an effective relaxed plan from the relaxed planning graph.

The framework consists of 2 steps: (i) building the (relaxed) planning graph; and (ii) using it to extract the relaxed plan. When extracting a relaxed plan, we ignore interference relations between actions that arise from negative interactions (i.e. one action deletes the precondition, or effect of another) or from two actions competing for the same resource. We will first look at the problem of how to build the planning-graph in the presence of exogenous events, then consider where scheduling techniques can help to improve the quality of heuristics generated via relaxed-plan extraction.

Building the Planning Graph with Exogenous Events

Both the classical planning graph (Blum & Furst 1997) and the temporal planning graph (Smith & Weld 1999) are built by adding all actions and facts at their optimistically estimated earliest possible execution times. Starting from the initial state and going forward in time, the graph alternates between possible executable actions and possible achieved facts, and represents the estimated earliest time at which we can achieve each goal as well as the causal relations between actions and fact at each time point. The estimation can be improved if we also propagate *mutual exclusion* relations between actions when building the planning graph. In addition to normal actions, we can also have *noop* (or persistent) actions that carry forward a fact from its earliest achieved time to infinity.

One key feature in this type of approach is that when a given action/fact appears in the planning graph at a certain level or time point, then it can persist through all later levels/time-points[1]. Distance-based heuristics such as the

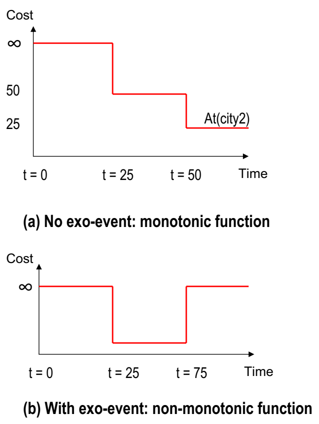

Figure 2: Cost functions with/without exogenous events

relaxed plan heuristic depend heavily on determining the level/time-points at which goals first appear non-mutex or alternately, the set of relaxed actions that optimistically achieve goals at the earliest time or with lowest cost. When exogenous events are introduced, this determination becomes much more complicated. Facts appearing in the planning graph may later be deleted by exogenous events, and thus actions supported by those facts will no longer be executable.

Exogenous events behave like actions in that they change the values of predicates, but unlike normal actions they must automatically be included in any plan. In terms of a standard planning graph semantics, if separate actions establish fact f and f at time point t, we assume that it may still be possible to achieve f at t because there may exist a plan that does not include the action deleting f. (This follows from the fact that each level of the planning graph is a disjunctive representations of all facts/actions possible at that level.) However, if there is an exogenous event e that deletes f at t, then f cannot be achieved at time point t, regardless. The effects of exogenous events *overwrite* effects of all other actions, and accounting for such negative events greatly complicates the difficulty of incorporating such events in a (relaxed) planning graph. Given a relaxed plan extracted from such a planning graph, there is no informedness loss incurred due to the fact that we've ignored negative effects from (relaxed) actions *that do not appear in the (relaxed) plan*. However, the negative effects associated with any exogenous 'actions' that are ignored in building a relaxed planning graph may seriously degrade heuristic informedness since they are guaranteed to exist in the final plan.

Figure 1 illustrates one portion of the planning graph with and without exogenous events. Without exogenous events, actions and facts are represented by their earliest possible achievement time. With exogenous events, the times at which actions and facts appear in the planning graph are represented as disconnected intervals, depending on the actual causal relations between actions in the plan and the exogenous events.

In addition to time, we can estimate the cost to achieve

[1]This enables an efficient bi-level implementation of the planning graph structure wherein each action/fact is only associated with the earliest level/time-point at which it first appears.

the goals by propagating the (reachability) cost of achievement while building the planning graph (Do & Kambhampati 2003). Reasoning over cost in planning is similarly complicated by the introduction of exogenous events. Assume that the cost of executing action $zoom(city1, city2)$ doubles the cost of action $fly(city1, city2)$ and takes half the time to finish. Figure 2 shows the cost function of one fact $at(city2)$ assuming that the aircraft is at $city1$ in the initial state. When exogenous events are present actions and facts introduced in the planning-graph can be deactivated later, so the cost functions for some actions and facts will no longer have the property of *monotonically* decreasing. Actions and facts may disappear from the planning graph or their achievement costs can be reset after some certain durations (this problem has also been discussed in (Smith 2003)).

In Figure 2, we show only the simple case of the fact $open\_station(city0)$ which is directly affected by exogenous events. A slightly more complicated example would be a given exogenous event disables action A, which is the current (estimated) best supporter for a fact f. Thus, the best action that can support f is changed from A to the next best action A' and thus the cost to achieve f increases. As a result, the cost function of f is no longer monotonically decreasing. In general, cost functions can increase and decrease in a complicated manner depending on the interactions between actions and exogenous events. We will show later that it becomes harder to reason about action selection when extracting a relaxed plan based on the cost functions if they are not monotonically decreasing.

After constructing cost functions by propagating the cost information when building the planning graph, they can be used to extract the relaxed plan during the final heuristic estimation. The relaxed plan is extracted by first picking a goal, then by looking at the cost function of the goal, we can pick the most promising action to achieve that goal. For example, if the goal is $at(city2)$ then the cost function in Figure 2 shows that the lowest cost action to achieve it at time $t \geq 50$ is $fly(city1, city2)$ but if $t < 50$ then the most promising one is $zoom(city1, city2)$. When an action is picked, its preconditions are added to the goal list. Thus, if action $zoom(city1, city2)$ is picked to support goal $at(city2)$ at time t, then its precondition $at(city1)$ is added as a new goal at time $t - Dur(zoom(city1, city2)$. Going backward and *ignoring* (relaxing) any interferences between actions), the relaxed plan extraction routine is guaranteed to finish in linear time without backtracking. The resulting action set along with its causal structure can be used as an estimate of the real set of actions (ie. the plan) that achieves the goals.

As mentioned above, without exogenous events the cost functions monotonically decrease, and thus to support a goal g at time t_g, we only need to choose the (lowest cost) action A supporting g at t_g because A gives a lower cost *and* allows more time to achieve its preconditions. With non-monotonic cost functions engendered by the presence of exogenous events, the lowest cost that we can achieve g at t_g may not be the lowest overall cost. There can be a time point $t < t_g$ at which $C(g, t) < C(g, t_g)$ because an exogenous event reset the cost of g after t. If we choose the lowest cost action A' that supports g at t, the cost-function indicates that it can be better cost-wise choice. However, because we have less time to achieve the precondition of A' than A it may not be the best choice when we take the interference relation into account and have to order mutual exclusive actions (due to logical or resource contention constraints) one after another.

The cost-functions can provide effective guidance in selecting actions that lead to a relaxed plan with least cost. However, to select actions that are less likely to violate the temporal constraints (and thus constitute a better relaxed plan), we will next consider heuristic techniques from the scheduling community such as slack-based and texture based approaches.

Exogenous events as *est* and *lft* time constraints

As we discussed in the earlier part of the paper, exogenous events can be seen as providing the time constraints on actions in the plan. We showed that in the sample problem (provided by the IPC4 organizers), all the exogenous events that are in the form of *e = (at 10 (open-station city0))* and *e' = (at 20 (not (open-station city0)))* can be discarded and replaced by the est/lft constraints on actions that have the predicates *P = (open-station city0)* as preconditions. Thus, for a given action *A = refuel(planX,city0)*, the precondition P can be discarded and the interactions between e, e' and A can be represented as the time constraints $est(A) = 10$ and $lft(A) = 20$. Those types of constraints are common in deriving slack-based and texture-based scheduling heuristics. From that point of view, we can use scheduling techniques to improve the heuristic estimation framework in planning, specifically the relaxed-plan extraction routine.

Scheduling Heuristics for Relaxed-plan Extraction

The key to the relaxed plan heuristic approach is to derive a relaxed plan that is as close to a good quality valid plan as possible. Extracting such a relaxed-plan from the planning graph involves (i) finding a good quality action set according to the user's objective function (e.g. if the user wants a low-cost plan, then the actions selected in the relaxed plan should be biased towards lower costs); and (ii) minimizing constraint violations.

With respect to the first criterion, the previous section has discussed how to change the time-sensitive cost functions guiding the relaxed-plan extraction in the presence of exogenous events. For the second criterion, constraint violations arise from the relaxed negative effects. Actions in the relaxed plan may contradict each other or compete for the same resources. Without the *est* and *lft* constraints imposed by the exogenous events, all of the pair-wise interactions between actions in the relaxed plan can be solved by ordering two interacting actions one after the other. However, with the new *est* and *lft* constraints, some orderings may lead to violations of those temporal constraints. To reduce the number of constraint violations while extracting the relaxed plan, we can guide the process via some scheduling heuristic techniques.

Slack-based Heuristic: Let $S = \{A_1, ...A_k\}$ be the set of actions that have been selected so far in the relaxed-plan extraction process. Because there is no backtracking in this process, all the actions in S will be included in the final relaxed plan. Suppose that g is the next goal that we need to support at time t_g. Assume that the set of actions that can support it is $S_g = \{A_1^g, A_2^g, ...A_m^g\}$ (which can be sorted according to their execution cost estimated by the cost functions). We would like to select greedily one action from S_g such that if included in S, it will minimize the chance of a temporal constraint violation in the final relaxed plan. Unlike a standard scheduling problem in which the goal is to allocate as many tasks as possible in the plan, we have several choices of actions in S_g and only one of them will be included in the final relaxed plan, while all of the others will be discarded. In the following paragraph, we discuss the selection criterion based on the slack value. To facilitate the discussion, for each action A in S or S_g, the earliest starting and latest finishing times are: est_A and lft_A. Because the selected action from S_g needs to achieve the goal g at t_g, the latest finishing time of action $A \in S_g$ is $lft_A \leftarrow min\{lft_A, t_g\}$.

To calculate the slack values, for each action A in S_g, we first identify a subset S_A, of actions in S that must be ordered with regard to A. Each action in S_A either logically interferes with A or competes with A for some resource. In scheduling, feasible orderings between two resource competing actions depends only on whether the $slack(i \rightarrow j)$ and $slack(j \rightarrow i)$ are positive or negative. However, in a temporal planning problem, feasible orderings between two interfering actions depend on whether an action deletes a causal link instituted by the other. Therefore, possible orderings in planning problems involving exogenous events depend on both the causal relationships between actions and the temporal constraints on action starting times. Thus, per (Smith & Cheng 1993), the ordering $A_i \rightarrow A_j$ is possible if $slack(i \rightarrow j) \geq 0$, while in a temporal planning problem $A_i \rightarrow A_j$ is possible if $slack(i \rightarrow j) \geq 0$ *and* A_j does not interfere with any causal relation started from A_i (e.g. A_j deletes p, and there is a causal link $A_i \rightarrow^p A_k$). Using such conditions, we can eliminate violating candidate actions from S_g.

Now, assume that for each action A in S_g and A_i in $S_A \subset S$, the slack value of $slack(A \rightarrow A_i)$ and $slack(A_i \rightarrow A)$ can be calculated if $A \rightarrow A_i$ or $A_i \rightarrow A$ is possible according to the two constraints listed above, then:

$$slack(A, A_i) = min\{slack(A \rightarrow A_i), slack(A_i \rightarrow A)\} \quad (1)$$

We have a choice between actions in the S_g set as to which is the most suitable to support the subgoal g. Therefore, we define an additional slack value:

$$slack(A, S_A) = min_{A_i \in S_A} slack(A, A_i) \quad (2)$$

If A is the next action to be selected to support g and thus definitely be included in the final relaxed-plan, then equation (2) can be used to select the *most constrained action* with regard to A and the slack heuristic. However, A is only one of the candidate actions in S_g supporting g: we wish to compare all such actions with respect to their slack values. The goal is to select the least constrained actions

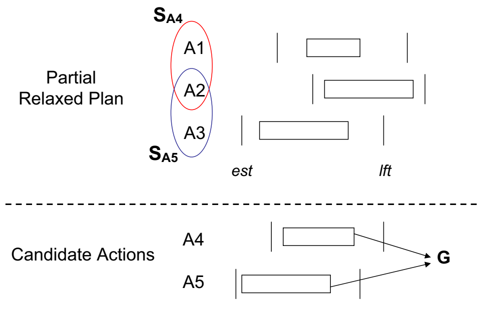

Figure 3: Relaxed-plan heuristic example

among all those in S_g. Based on the $slack(A_i, S_{A_i})$ values of all actions $A_i \in S_g$, we then can choose support for g such that it has minimal chance of leading to a temporal constraint violation later. Thus we can choose an action A such that:

$$slack(A, S) = max_{A_i \in S} slack(A_i, S) \quad (3)$$

Unlike traditional slack based variable ordering in scheduling where the interest lies with the most constrained pair of actions based on the smallest slack value between them (most constrained variable first heuristic), the process of extracting a relaxed plan with no backtracking motivates selecting the action with highest slack $slack(A, S)$ value to minimize the chance of an eventual temporal constraint violation. In other words, if we consider g as a variable, S_g as its domain and actions $A \in S_g$ as values, then equation (3) is used to select the *least constrained value*. In this sense, the max equation is used over the top of min equation (2) to be the value ordering for variable g. From this point of view, we can even use the overall slack values calculated in equation (3) as the mean to select most-constrained variables by selecting g in all the remaining goals that lead to the smallest slack value calculated by (3).

Example: Figure 3 shows one example that will illustrate the procedure discussed above. Assume that the current relaxed-plan contains 3 actions $S = \{A_1, A_2, A_3\}$ and there are two actions that can support the goal g in $S_g = \{A_4, A_5\}$. Among actions in S, $S_{A_4} = \{A_1, A_2\}$ are actions that need orderings with A_4 (competing for the same resource or logical interference), and $S_{A_5} = \{A_2, A_3\}$ are actions interacting with A_5. To choose between A_4 and A_5, we first compute the slack values between A_4 and actions in $S_{A_4} = \{A_1, A_2\}$) and slack values between A_5 and actions in $S_{A_5} = \{A_2, A_3\}$) using equation (1) for individual pair of actions. Then, we will use equation (2) to compute the slack value $slack_{A_4} = slack(A_4, S_{A_4})$ and $slack_{A_5} = slack(A_5, S_{A_5})$. Those values indicate the worse case scenario if we select A_4 or A_5 to add to the current relaxed plan. Then, we select the maximum value among $slack_{A_4}$ and $slack_{A_5}$ to choose the better of the two.

Texture-based Heuristics: Besides the slack-based heuristic another popular heuristic in scheduling such as texture measurement (Fox et. al. 1989; Beck et. al. 1997) based heuristic can also be used to guide the action extraction process. This approach may be particularly effective in a planning problem involving multi-capacity resources. Such problems entail reasoning beyond the causal relations and interference relations associated with pairs of actions.

Individual and aggregated resource demand curves can be built for all actions in the partial relaxed plan at any stage. Then, for each action A in the S_g that can potentially support subgoal g, we can build the individual demand curve for A and see how the aggregated demand curve changes when A is included in the action set S of actions already selected. Based on the aggregated curves for all individual actions in S_g, we can use variations of texture-based heuristics discussed by Beck et. al. (1997) to find the action that is least likely to lead to resource contention failure. One example could be to select an action in S_g such that if it is included in S then resulting aggregated demand curve has the smallest maximum height increase. The hope is that this will minimize the chances of eventually exceeding resource capacity. We will use the same example shown in Figure 3, to illustrate briefly how it can be done.

Example: Adopting the same assumptions , A_4 and A_5 are two candidate actions that we need to choose one from to support a goal g; $S = \{A_1, A_2, A_3\}$ are actions already selected in the relaxed-plan. $S_{A_4} = \{A_1, A_2\}$ is the set of actions that use a resource R_4 which is also used by A_4, and similarly, $S_{A_5} = \{A_2, A_3\}$ contain actions sharing resource R_5 with A_5. Note that unlike the slack-based heuristic where we only consider one set of actions that have to be ordered with regard to a candidate action, there can be multiple sets of actions for each candidate action in the texture-measurement approach. For example, if A_5 uses both resources R_5 and R_4 then we need to consider both S_{A_4} and S_{A_5} when we calculate the effect of A_5 on the resource demand curves of the relaxed plan.

Now assume that A_4 only uses resource R_4 and A_5 only uses R_5. We first build the aggregated demand curves for R_4 using actions in S_4 (within their est/lft constraints) and for R_5 using actions in S_5. After building the individual demand curve of R_4 for A_4 and of R_5 for A_5, we will incorporate them into the overall aggregated demand curves of R_4 (i.e. the aggregated curve of R_4 for $\{A_1, A_2, A_4\}$) and R_5 (aggregated demand curve for $\{A_2, A_3, A_5\}$). We then can decide which of the actions A_4 or A_5 has the lesser effect (increases the maximum height of the demand curve the least) and select that one to support g. If a given candidate action A uses multiple resources, then we can take the maximum effect height increment for those resources in the presence of A as its texture-effect measurement. We then take action with the minimum value among all the candidate actions as the next one to be included in the relaxed plan.

Conclusion and Future Work

Exogenous events are a natural component of real world planning problems and have been introduced into the standard planning language PDDL2.2 for the ICAPS planning competition this year. In this paper, we have examined exogenous events from both planning and scheduling points of view. From the planning side, exogenous events appear as instantaneous actions that change the state description of the world and thus interact with other actions in the plan. From the scheduling point of view, we can consider an important class of exogenous events as imposing *earliest starting time* and *latest finishing time* to other actions in the plan. Such constraints are seldom considered in the planning field but are quite common from the scheduling perspective.

Looking at the problem from both planning and scheduling sides, we discussed approaches for combining heuristic techniques from both fields to solve temporal planning problems involving exogenous events, especially in cases where actions also compete for resources.

Our future work involves first, identifying classes of problems most suitable for the techniques discussed in this paper, and then implementing the approaches and testing on a set of planning problems involving exogenous events.

References

Beck, C., Davenport, A., Sitarski, E., and Fox, M. 1997. Texture-Based Heuristics for Scheduling Revisited. In *Proc AAAI-97.*

Blum, A., and Furst, M. 1997 Fast Planning Through Planning Graph Analysis. In *AIJ 90 (p.281–300).*

Bonet, B., Loerincs, G., and Geffner, H. 1997. A robust and fast action selection mechanism for planning. In *Proc AAAI-97*

Cresswell, S., and Coddington, A. 2003. Planning with Timed Literals and Deadlines In *UK PlanSIG*

Do, M. and Kambhampati, S. 2003. Sapa: a multi-objective metric temporal planer. In *JAIR 20 (p.155-194)*

Fox, M., Long, D., and Halsey, K. 2004. An Investigation into the Expressive Power of PDDL2.1 *Unpublished manuscript*

Fox, M.S., Sadeh, N., and Baykan, C. 1989. Constrained Heuristic Search. In *Proc. of IJCAI-89.*

Hoffmann, J. and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. In *JAIR 14 (p.253-302).*

Hoffmann, J. and Edelkamp, S. 2004. IPC4 Website: *http://ipc.icaps-conference.org/*

Nguyen, X., Kambhampati, S., and Nigenda, R. 2001. Planning Graph as the Basis for deriving Heuristics for Plan Synthesis by State Space and CSP Search. In *AIJ 135 (p.73-123).*

Muscettola, N., Smith, S.F., Cesta, A. and D'Aloisi, D. 1992. Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Framework In *IEEE Control Systems, 12(1).*

Smith, S. and Cheng C. 1993. Slack-Based Heuristics for Constraint Satisfaction Scheduling. In *Proc. of AAAI-93.*

Smith, D. and Weld, D. 1999. Temporal planning with mutual exclusion reasoning. In *Proc. of IJCAI-99.*

Smith, D. 2003. The Mystery Talk. *Plannet Summer School*