Multiple Relaxations in Temporal Planning

Keith Halsey and **Derek Long** and **Maria Fox**[1]

Abstract. CRIKEY is a planner that separates out the scheduling from the classical parts of temporal planning. This can be seen as a relaxation of the temporal information during the classical planning phase. Relaxations in planning are used to guide the search. However, the quality of the relaxation greatly affects the performance of the planner, and in some cases can lead the search into a dead end. This can happen whilst separating out the planning and scheduling problems, leading to the production of an unschedulable plan. CRIKEY can detect these cases and change the relaxation accordingly.

1 CRIKEY

CRIKEY is a planner that uses relaxations to solve temporal planning problems. By successively reducing the complexity of the problem, the solutions to these relaxed problems are used to either guide the search for the unrelaxed version, or as a skeleton solution, round which a full solution can be built.

CRIKEY performs forward heuristic search and can cope with metric and temporal domains and is closely based on MetricFF [5]. For this reason, it is the temporal aspects of the planner and the relaxations of the temporal information, that is the focus of this paper. CRIKEY is a re-implementation, with improvements, of the system described in [3], and follows the architecture as detailed in Figure 1. The principle is to firstly extract the temporal information from a temporal planning problem. The resulting metric planning problem is solved with forward heuristic search, using a relaxed plan graph as a guide to finding a totally ordered plan. A partial order is extracted from this plan using a modified Veloso algorithm [6] to find ordering constraints based on logical and metric reasoning. Then, along with the temporal information, a Simple Temporal Network is used to produce a valid temporal plan of time stamped actions. Significantly, there is no feedback from the scheduling to the planning phase; These are separate, happening one after the other. Whilst this will not find an optimal plan, that is to say one that exploits all concurrency possible, it is complete and sound.

This architecture separates the planning and scheduling problems found in temporal problems solving each half separately. In this aspect it is similar to MIPS [1]. Actions, that can have a duration, must both be chosen for their logical and metric effects (planning) and then arranged in time so as not to break any constraints (scheduling). This has the advantage of making each problem on its own easier to solve and has proved sucessful in the domains of the IPC'02 (International Planning Competition 2002) [2]. The problem with this architecture as it stands is where the planning and scheduling interact. Here, the relaxation fails and an unschedulable plan can be produced.

[1] University of Strathclyde, Glasgow, UK email: keith.halsey@cis.strath.ac.uk

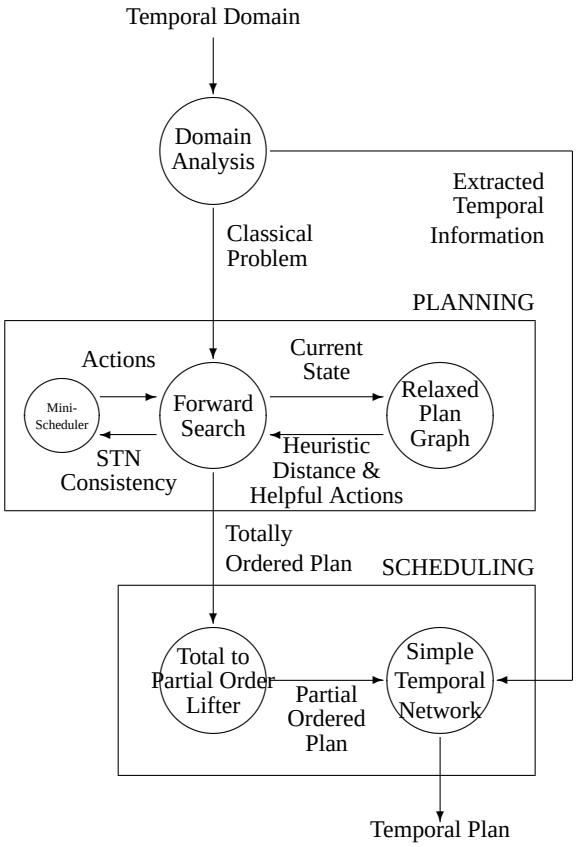

Figure 1. Architecture Overview of CRIKEY

2 UNSCHEDULABLE PLANS

Whilst CRIKEY can perform competitively against other state of the art technology (see [3] for some results), the approach of separating the planning and scheduling has a major flaw. In the domains of IPC'02, all the solution plans could be sequentialised (the actions performed in a strictly ordered manner). That is to say that actions *could* happen in parallel if they didn't interfere, but that, however, was a choice of the planner, and not enforced by the domain. However, it is possible to have domains where some actions *must* happen in parallel.

Take for example the Match Domain (Figure 2) as described in PDDL2.1. In this domain, the goal is to mend fuses. To mend a fuse (which takes 5 time units) you must have a hand free (i.e. you can

only mend one fuse at once) and there must be light (provided by striking a match that lasts 8 time units). It should be obvious that in a problem instance where there are two fuses to fix, two matches are needed. If the problem is completely separated, CRIKEY will not realise this and try only to use one match. Here an unschedulable plan is produced and the temporal planner fails. In this case the sub-problems of planning and scheduling are more tightly coupled and are interdependent. The planning part critically effects the scheduling problem, and so in this case, the two cannot be separated.

```
(:durative-action LIGHT_MATCH
   :parameters (?match - match)
   :duration (= ?duration 8)
   :condition (and (over all (light))
      (at start (unused ?match)))
   :effect (and (at start (light))
      (at start (not (unused ?match)))
      (at end (not (light)))))

(:durative-action MEND_FUSE
   :parameters (?fuse - fuse)
   :duration (= ?duration 5)
   :condition (at start (handfree))
      (and (over all (light)))
   :effect (and
      (at start (not (handfree)))
      (at end (mended ?fuse))
      (at end (handfree)))))
```

Figure 2. Two Actions in the Match Domain

A more in-depth look at where planning and scheduling interact in temporal planning (with particular reference to PDDL2.1) can be found in [4]. A very brief summary follows. Interactions between scheduling and planning occur where there are durative actions (called *content* actions) that must be executed in the time that another sequence of actions (called *envelope* actions) execute. In these cases, the minimum length of time for the content actions must be less (or fit into) the maximum total length of time for the envelope actions. If this is not the case, then an un-schedulable plan is produced.

2.1 Mini-Scheduler

Where the two problems interact, (i.e. where there are envelope actions), some of the logical and temporal reasoning must be done together to make sure that the plan is schedulable (i.e. that there is enough time to execute the content actions within the time of the envelope actions). This is achieved by attaching a mini-scheduler for these cases onto the planner.

When an envelope action is placed in the plan, a new mini-scheduler is associated with it. This mini-scheduler consists of a Simple Temporal Network, a set of content actions (initially empty) and a set of orderings between these actions. The mini-schedulers use the same algorithms as the main scheduling part of CRIKEY. Any (content) actions which are now considered, must be checked against this mini-scheduler to ensure that if they must go in the envelope, that the STN is consistent (that is to say that there is enough time to execute the action). If not, then the action is not considered applicable, and that branch is removed from the search space. When the envelope's end effects are placed in the plan, the mini-scheduler is then discarded.

In the case of the match actions, assuming it is part of a bigger domain, CRIKEY will search forward ignoring temporal information. When it comes to put in the light match action (an envelope), it will instantiate a new mini-scheduler. It will then test to see if a fix fuse action need go in this mini-scheduler, and if so, if it is consistent. Indeed, it fits, so the action is applicable and choosen for the plan. It will then test the second fix fuse action. This is not consistent with mini-scheduler (there is not enough time left to fix it before the match runs out), so cannot be put in the plan. (If the fuses could be fixed in parallel, then this second action would be consistent and so chosen). The mini-scheduler would be closed and the end effects of the light match action applied. CRIKEY would then go on to either light a second match (and so start a new mini-scheduler) or solve another part of the problem. In this way a schedulable plan is produced.

3 CONCLUSIONS AND FURTHER WORK

CRIKEY shows that it is possible to separate out temporal and logical reasoning, whilst combining them where necessary. CRIKEY performs the reasoning necessary to do this. It also demonstrates that it is possible to perform a series of relaxations in order to find a plan, and tighten these relaxations where necessary. It is hoped that these themes will be explored further in the near future. The scheduler is of particular interest since at the moment it only performs a greedy search. It could perform some search to improve the quality of the plan, depending on the metric of the original problem. Also, it is thought that the metric reasoning and logical reasoning can also be separated out in the scheduler, particually since currently CRIKEY cannot accurately schedule domains where the resource consumption or production levels are dependent on the state. Further research is intended to tighten the relaxed plan graph heuristic such that when this fails (i.e. when a plateaux is found), rather then searching a way out of the space, a better (but more costly) heuristic is used.

CRIKEY participated in the 4th International Planning Competition, so results from that are hoped to provide an understanding of it potential. However, the domains used were limited in the complexity in which the planning and scheduling interacted. Further testing will be done with other temporal planners, on domains where there is a close coupling between the two parts of the problem.

REFERENCES

[1] S. Edlekamp and M. Helmert, 'On the implementation of mips', in *Proceedings from the 4th Artificial Intelligence Planning and Scheduling (AIPS), Workshop on Decision-Theoretic Planning*, pp. 18–25. Brekenridge, Colorado:AAAI-Press, (2000).

[2] M. Fox and D. Long, 'The third international planning competition: Temoral and metric planning', in *Proceedings from the 6th International Conference on Artificial Intelligence Planning and Scheduling (AIPS'02)*, pp. 115–117, (2002).

[3] K. Halsey, 'Temporal planning with a non-temporal planner', in *Doctoral Consortium at the 13th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 48–52, (June 2003).

[4] Keith Halsey, Derek Long, and Maria Fox, 'Isolating where planning and scheduling interact', in *Proceedings from the 22nd UK Planning and Scheduling Special Interest Group (PlanSIG'03)*, pp. 104–114, (Dec 2003).

[5] Jörg Hoffmann, 'Extending FF to numerical state variables', in *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-02)*, pp. 571–575, Lyon, France, (July 2002).

[6] Manuela M. Veloso, M. Alicia Pérez, and Jaime G. Carbonell, 'Nonlinear planning with parallel resource allocation', in *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, pp. 207–212. Morgan Kaufmann, (nov 1990).