Planning with Sharable Resource Constraints

Philippe Labone and Malik Ghallab
LAAS-CNRS
7, Avenue du Colonel Roche
31077 Toulouse Cedex, France

Abstract

When planning systems deal with realistic domains, they must cope with a large variety of constraints imposed by the environment such as temporal or resource constraints The robustness of the generated plan is a direct consequence of a correct handling of these constraints We argue that increasing the expressiveness of a representation can be achieved without fundamentally affecting the global efficiency of the search This paper presents a temporal planner, LxTeT, which integrates sharable resource management into the process of plan generation In lxTeT, planning operators are described as temporal structures of conditions, effects and sharable resource uses During the search, pending subgoals, protection threats and resource conflicts are detected by three flaw analysis modules The detection of sharable resource conflicts is performed thanks to an efficient clique-search algorithm on a possible intersection graph The control of the search is based on a least-commitment opportunistic strategy Our approach has been implemented, tested and shown to be satisfactory in various application domains

1 Introduction

Classical approaches decompose *plan generation* and resource *management* into two different steps for solving a given planning problem Plan generation consists in selecting and organising a set of operators that will achieve the problem goals, its result is a *plan,* that is a partially ordered set of operators Resource management is then performed on this plan to ensure its correctness with respect to temporal and resource constraints The result, then, is a *schedule* that, if executed, will ensure the goal resolution while satisfying the physical constraints of the environment

The strict separation between *plan generation* and re *source management* raises difficulties such as how to backtrack to the plan generation process, when necessary, or how to generate the plan in an opportunistic way according to resource constraints

This paper develops a representation and an algorithmic approach integrated into a system called lxTeT, for handling abreast and in an opportunistic way plan generation and time/resource management

The following section is a short digest of scheduling and planning literature that influenced our work Section 3 describes a temporal representation of operators and problems The genera] control of the planner is discussed m section 4 whereas section 5 focuses on the resource analysis module We finally presents some experimental results and applicability domains for our system

2 Related Work

2 1 Scheduling

Scheduling systems are essentially concerned with time and resource management ISIS [Fox and Smith, 1984] was one of the first attempt to deal with a *large variety of realistic constraints* for job-shop scheduling Some *operator pre-conditions* were handled to detect inconsistency but nothing was done to enforce their aatiafaction Other systems like OPIS [Smith *et* a/, 1986] or MICRO-BOSS [Sadeh, 1991] introduced the notion of *opportunistic search* that is the ability to dynamically revise the search procedure In their approach, this consists in choosing first to schedule the most constrained operations l e those with the highest contribution to resource bottlenecks Systems like OPAL [Bensana *et al*, 1988] or MASCOT [Erechler *et al*, 1990] work out a *constraint based analysis* to detect some necessary features of the solutions represented by the current partial schedule In these approaches, opportunity of a conflict resolution depends on the number and characteristics, of the different ways to solve it The resource manager of lxTeT has been highly influenced by the latter approaches as an opportunistic search is made necessary by the plan generation aspects the system deals with

2 2 Planning

The algorithmic basis of classical planning have been well clarified, eg in [Chapman, 1987] and [McAllester and Rosenblitt, 1991]

Extending the classical representation to handle temporal constraints like operator durations or earliest latest time, has been an important preoccupation in planning Like FORBIN [Dean *et al ,* 1988] or O-PLAN [Cur

re and Tate, 1991], IxTeT uses time-points as primitives and propagates metric constraints between time-points This has been shown to be less complex [Vilain and Kautz, 1986] than interval based approaches [Allen *et*

1991]

Concerning resource management, SIPE was the first planner to handle consumable and producible resources [Wilkins, 1988] Although no constraint was added to the plan to prevent potential resource conflicts, the search space *was* pruned to avoid some unuseful exploration of over-consumer plans SIPE also implemented the notion of unsharable re-usab)e resources

Because theu management is inherently untractable, *shamble rtsourcts* has rarely been integrated into a planning system The HSTS framework [Muscetolla, 1993] addresses this problem however, it over-constrains the current partial plan as sharable resources are handled through a total-ordered time-line O-PLAN handles sharable resources by using a criterion based on optimistic and pessimistic resource profiles [Drabble and Tate, 1994] to prune the search space but these profiles seem not to be sufficient to ensure the soundness of the search with respect to resource conflicts

This paper presents a *sound, complete* and *efficient least-commitment* based approach to manage sharable resources during the planning process The approach redes on various techniques and ideas from *classical planning, scheduling, constraint satisfaction problems* and *graph theory* It develops an original procedure for managing efficiently and jointly temporal and resource constraints It has been integrated with a sophisticated control into the IxTeT temporal planner

3 Representation

3 1 Time and Variables

For algorithmic complexity reasons, the IxTeT *ttmc-map manager* [Ghallab and Alaoui, 1989, Ghallab and Vidal, 1995] relies on time-points as the elementary primitives Time-points are seen as symbolic variables on which temporal constraints can be posted We handle both *sym bohc constraints* (precedence, simultaneity) and numeric *constraints* expressed as bounded intervals *[l~,l⁺]* on the temporal distance between time-points Disjunctions of these constraints are not directly handled by the time-map manager but through the control module The time map *manager* propagates constraints to ensure the global consistency of the network and answers queries about the relative position of time-points

A *variable constraint* manager handles atemporal variables ranging over finite Bets and propagates domain restriction, *equality,* and *unequality constraints*[1]

3 2 Description of World and Change

The world state is described through a set of *multi-valued state attributes* A state attribute is a k-ary state variable which can take only one value at a time

Actually, as propagating variable uneqnakties is an untractable problem in itself, we *jstt* work out a local consistency for this type of constraints

IxTeT is based on a reified logic formalism where state attributes are temporally qualified by the predicates *hold* and *event*

- an assertion *hold(att(xi,*) v,(ti,t2)) asserts the persistence of the value of attribute *att(x$_{it}$)* to *v* for each *t tl < t < i2 Assertions* allow the expression of operator conditions as well as of causal hnks
- *event(att(xi,*) *{vi,vj),t)* states that an instantaneous change of value of *att(x\,*) from *vl* to *v2* occured at time t *Events* allow the description of state change in operators as well as of expected changes of the world in the initial problem description

3 3 Description of Resource Uses

We define a resource as *any substance or set of objects whose cost or availability induces constraints on the actions that use them*

A resource can be a single item with a unit capacity (an *unsharable resource),* or an aggregate resource that can be shared simultaneously between different actions seeing that its maximal capacity is not exceeded Resources are gathered together into *resource types* two resources belong to the same type if they can be indifferently used

The resource availability profiles and their uses by different operators are described b> means of three predicates

- *use{typ{r)* ff$_r$ (*i»*a)) means that an integer quantity *q* of resource r of type *typ* will be used between time-points *t* and *ty* Notice it is just a borrowing of a part of the resource,
- *con8ume(typ(r) g,t)* states that a quantity y of resource *r* will be consumed at time t, l e that the availability of r will decrease at *t*
- *produce(iyp{r) q,t)* represents a production of resource at time *t*

This paper focuses on the management of *sharable re sources* (i e whose capacity exceed 1) As these resources are often equivalent resources for which we don't need to explicitly manage a final allocation, we first make the hypothesis that all the resources of a given type are *completely aggregated* so that the only information binded to a resource type is its maximal capacity Furthermore, we will only consider the *borrowing* of resources (predicate *use)* We will extend these hypotheses at the end of the paper

3 4 Planning Operators

Planning operators are described through a *hierarchy* of *tasks* A task is a temporal structure composed of

- a set of sub-tasks,
- a set of events describing the changes of the world induced by the task,
- a set of assertions on state attributes to express the required conditions or some causal links between task events,
- a set of resource uses, and

- a set of temporal and instantiation constraints binding the different time-points and variables of the task

Tasks are *deterministic* operators without ramification effects We assume that all conditions and changes linked to the task are explicitly defined into its hierarchic decomposition Notice that this hierarchy is but a user programming facility to build complex tasks on the basis of simpler ones, a compilation procedure is worked out before planning to translate tasks into IxTeT structures by expanding the events, assertions, resource uses and constraints of sub-tasks up to the top level task IxTeT also uses a control hierarchy but it is not directly linked to this representation hierarchy [Garcia and Laborie, 1994]

Below is described the task "INCUBATE" for planning in the context of a spatial station where biological experiments are carried out[2]

```
resource POWER() { capacity = 100, }

attribute POSITION(?Elt) {
  ?Elt in {Culture1  Culture2, Tube1     },
  ?value in {Stock, Wb, Incub    }, }

task INCUBATE(?Elt, ?T)(start end) {
  hold(TEMPERATURE() ?T, (start end))
  hold(POSITION(?Elt) Incub, (start,end)),
  event(INCUBATED(?Elt,?T) (no yes), end)
  use(INCUBATOR() 1, (start,end)),
  use(POWER() 8  (start,end)),
  (end - start) in [00 09 00,00 10 00]  }
```

3.5 Initial Plan

The initial plan $\mathcal{P}_{init}$ is a special task that describes the problem scenario, that is

- the initial values for the set of instantiated attributes,
- the expected changes on some state attributes that will not be controlled by the planner,
- the expected availability profile of resources, and
- the goals that must be achieved

The first two points are represented thanks to a set of *explained events*, the availability profiles with a set of *uses* and the goals as a set of *assertions* to explain IxTeT allows a total flexibility on the expression of temporal constraints between these elements

As many classical planners, IxTeT explores a search tree of partial plans whose root is the initial problem and branches are operators or constraints inserted on the current partial plan

4 Control

4.1 Flaws and Resolvers

Our planner relies on the use of causal links In the following, we extend the vocabulary defined in [McAllester and Rosenblitt, 1991] to the IxTeT temporal formalism

Unexplained propositions a temporal proposition $hold(att(x_1, \ldots, x_n) \ v, (t, t^*))$ or $event(att(x_1, \ldots, x_n) \ (v, v^*), t)$ will be said *explained* iff

1 there is an *establisher* $event(att(x'_1, \ldots, x'_n) \ (v_*, v'), t')$ such that necessarily $(t' \leq t)$, $(v' = v)$, $(x'_1 = x_1)$, and $(x'_n = x_n)$, and
2 if $(t' < t)$, there is a *causal-link* between the establisher and the proposition $hold(att(x_1, \ldots, x_n) \ v, (t', t))$

In IxTeT, establishers are naturally represented by events and causal links by assertions The *sub goal analysis module* analyses the unexplained propositions and create two kind of resolvers *causal links* (establishment by an event already in $\mathcal{P}$) and *tasks* insertion

Threats are possibly conflicting propositions, ie either

1 a pair $< e, cl >$ where $e = event(att(x'_1, \ldots, x'_n) \ (v_*, v^*), t')$ and $cl = hold(att(x_1, \ldots, x_n) \ v, (t_1, t_2))$ such that the current plan $\mathcal{P}$ does not contain one of the constraints $\{(t' < t_1), (t' = t_1 \text{ and } v = v^*), (t_2 < t'), (t' = t_2 \text{ and } v = v_*), (x_1 \neq x'_1), \ldots, (x_n \neq x'_n)\}$ or
2 a pair $< h, cl >$ where $h = hold(att(x'_1, \ldots, x'_n) \ v', (t_*, t^*))$ and $cl = hold(att(x_1 \ldots, x_n) \ v, (t_1, t_2))$ such that $\mathcal{P}$ does not contain one of the constraints $\{(t^* < t_1), (t_2 < t_*), (x_1 \neq x'_1), \ldots, (x_n \neq x'_n), (x_1 = x'_1 \text{ and } \ldots \text{ and } x_n = x'_n \text{ and } v = v')\}$

For each possible threat, the *threat analysis module* computes a disjunction of resolvers composed of the above temporal or variable constraints

Resource conflicts for a type *typ* of resource with maximal capacity $Q(typ)$ are sets $< u_1, \ldots, u_k >$ such that, if $u_i = use(typ() \ q_i, (t_i^-, t_i^+))$

1 $\mathcal{P}$ does not contain one of the constraints $\{(t_i^+ < t_j^-)_{(i \neq j)}\}$, and
2 $\sum_{i=1}^{n} q_i > Q(typ)$

Resolvers of a resource conflict consist in a disjunction of temporal constraints and eventually, variable unequalities if different resources of the same type are allowed This is discussed in section 5

Solution plan criterion

We define a *flaw* as being either an unexplained proposition, a threat or a resource conflict

A partial plan $\mathcal{P}$ will be a solution plan iff it verifies the following conditions

1 *there is no flaw,*
2 *the temporal constraint network is consistent, and*
3 *the variable binding network is consistent*

An essential condition for a real integration of resource management and plan generation is the existence of a *homogeneous representation* on which the system can rely to continually justify the choices it has to make between these two activities The later paragraph points out that, although their *origin* differ, resource conflicts resolvers are of the same *nature* as plan generation ones (pending subgoals and threats) either temporal constraints, variable constraints or task insertion It is finally the representation of the current plan as a set of flaws and

[2] A complete modeling for this domain has been defined within the framework of the PADRE project for the COLUMBUS European Space Lab, financed by the French Minister for Education and Scientific Research, in collaboration with the companies IXI and Matra Marconi Space France

resolvers that will constitute a homogeneous representation where the control module chooses a flaw independently of its bejng a resource conflict, a threat or a pending subgoal This idea meets the one developed in WATPLAN [Yang, 1992] where the flaw/resolver representation is seen as a constraint satisfaction problem (CSP) [Tsang, 1993] whose variables are represented by flaws, and their possible values by resolvers

4 2 Choosing a Resolver to Solve a Flaw

In IxTeT, the explicit representation of temporal metric constraints and finite domain variables permits an *estimation of the commitment* induced by a constraint posting

Indeed, as a partial plan *V* represents a set of possible instantiations *INST(V),* we estimate the commitment of posting a constraint *p* on *V* as

$$commit(\mathcal{P}, \rho) = 1 - \frac{card(INST(\mathcal{P} \cup \{\rho\}))}{card(INST(\mathcal{P}))}$$

that is the ratio of instantiations eliminated by the posting of ρ

$commit(\mathcal{P}, \rho) = 0$ iff ρ is redundant with constraints already posted in $\mathcal{P}$, $commit(\mathcal{P}, \rho) = 1$ iff ρ is inconsistent with $\mathcal{P}$

Thus, if t and t' are two time points such that numeric constraints propagation on the temporal network of $\mathcal{P}$ states that $(t' - t) \in [d_{min}, d_{max}]$, we will estimate the commitment of posting $(t' < t)$ on $\mathcal{P}$ by supposing, in $INST(\mathcal{P})$, a uniform repartition of the value of variable $(t' - t)$ into $[d_{min}, d_{max}]$, so that

$$commit(\mathcal{P}, (t' < t)) = \frac{min(d_{max}, 0) - min(d_{min}, 0)}{d_{max} - d_{min}}$$

Variable constraint as well as causal link commitment are estimated through functions of the same kind

Determining which task is the most relevant for a pending goal and how a task insertion will further constrain the partial plan is a complex problem Such a procedure called *feasibility* is described in [Ghallab and Laruelle, 1994] and corresponds to a look-ahead that ignores threats and resource conflicts Its result is a commitment value attached to a given task insertion

Given a flaw and according to these definitions of commitment, LxTeT always chooses to first insert the least-commitment resolvers

4 3 Choosing next Flaw to Solve

Flaws are chosen in an opportunistic way to reduce the chances of backtracking A function *K(Φ)* measures the easiness to make a choice between the different resolvers of flaw V {p1, *,pt}* according to our least-commitment strategy It estimates how far the best resolver p_{min} is "ahead" of the other resolvers

$$\frac{1}{K(\phi)} = \sum_{i=1}^{k} \frac{1}{1 + commit(\mathcal{P}, \rho_i) - commit(\mathcal{P}, \rho_{min})}$$

The control module always choose to solve first the flaw ϕ that maximises $K(\phi)$ Notice that $0 < K(\phi) \leq 1$ $K(\phi) = 1$ iff ϕ has a unique resolver, we say in this case that Φ is a *deterministic flaw* as it does not complicate the search tree Function K permits to solve in priority those flaws with the smallest number of resolvers, this being modulated by the value of resolvers commitment

4 4 Search Tree Exploration

The search is controlled by a near-admissible A_t algonthm [Ghallab and AUard, 1983] which provides a trade-off between the efficiency *of* the search and the quality of the solution in terms of flexibility The estimate / of a partial plan *V* at a given node is a combination of *g,* the commitment along the path leading from Pmit to *V,* and *h,* a combination *of* the minimal commitment for each remaining flaw *Φ* in *V* Assuming that each remaining flaw in *V* can be independently solved bj its least-commitment resolver, / is inversely proportional to the maximal flexibility[3] of solution plans contained in *V*

5 Resource Management

Resource conflicts are detected and solved through minima/ *critical sets* [Erschler *et at,* 1990] Minimal critical sets (MCS) are detected as particular cliques on a specific representation of the temporal constraints the *possible intersection graph* (PIG)

5 1 Possible Intersection Graphs

The possible intersection graph (PIG) associated to a resource type *typ* on a partial plan $\mathcal{P}$ is a non-oriented graph $\mathcal{G}_{typ}(U, E)$ whose vertices U are the resource propositions $\{u_1, \ldots, u_n\}$ for resource *typ* and edges E the pairs $\{u_i, u_j\}$ such that u_i and u_j may possibly intersect in some temporal instantiation of $\mathcal{P}$ (i e $\mathcal{P}$ does not contain one of the constraint $\{(t_i^+ < t_j^-), (t_j^+ < t_i^-)\}$)

PIGs, being the complementary of comparability graphs, are a special kind of *weakly triangulated graphs* which are themselves a sub-class of *perfect graphs* Moreover, when all the time-points are instantiated, the PIG becomes a *triangulated graph*[4]

5 2 Minimal Critical Sets

A set $\phi = < u_1, \ldots, u_k >$ is a MCS of partial plan $\mathcal{P}$ iff

1 ϕ is a resource conflict, and
2 $\forall \phi' \subset \phi$, ϕ' is not a resource conflict

According to these definitions, MCS are exactly the *minimal over-consumer cliques* of $\mathcal{G}_{typ}$ e g on figure 1, $< u_3, u_4, u_7 >$ is a MCS Because of the least-commitment strategy, the smallest MCSs are the most interesting ones

Our MCS detection algorithm consists in finding some maximal cliques of $\mathcal{G}_{typ}$ and splitting them into MCS

Although the maximum clique problem for weakly triangulated graphs can be solved in polynomial time [Hayward *et al*, 1989], these algorithms are very inefficient

[3] *Flexibility* of & solution plan *V,* being defined, as a measure proportional to *card(INST(V$_i$))*

[4] A triangulated or chordal graph u a graph which does not have chordless cycle with length more than 4 A weakly triangulated graph IB a graph which docs not have choidlesa cycle with length more than 5 or its complement A chord is an edge joining to non-consecu tive vertices in a cycle

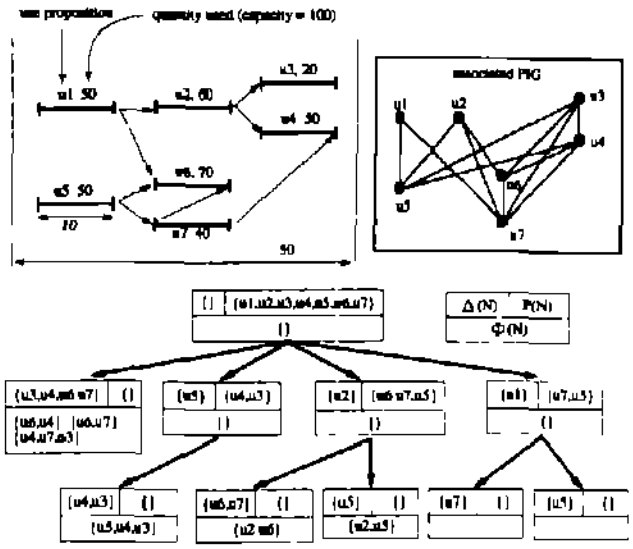

Figure 1 *a PIG and a MCS search tree*

in average and furthermore, they would only give some MCS and not necessarily the most interesting ones

On the other hand, it can be shown that if n is the size of $\mathcal{G}_{typ}$, the maximal number of MCS may reach $n^{1/2}\ 2^n$ making a complete search untractable

Our approach makes a compromise between the quality of the found MCS and the complexity of such a search

We develop a search tree whose nodes N are a triplet $< \Delta(N), P(N), \Phi(N) >$ Node N represents a current clique

$$C(N) = \bigcup_{M \in fathers(N) \cup N} \Delta(M)$$

that we try to enlarge in the following nodes More precisely,

- $\Delta(N) \subset U$ is the contribution of node N to the enlargement of the current clique,
- $P(N) \subset U$ is the pool of candidate propositions to enlarge the current clique in the following nodes, and
- $\Phi(N) \subset 2^{P(N)}$ is exactly those MCS which have at least one proposition in each $\Delta(M)$ for $M \in fathers(N) \cup N$

Developping node N consists in

1 partitioning $P(N)$ into sets $\Delta(N_i)$ where N_i are node N's sons,

2 computing nodes N_i's pools $P(N_i)$, and

3 computing $\Phi(N_i)$, the MCSs attached to nodes N_i

Points 1 and 2 are processed simultaneously through a vertex ordering of $P(N)$ similar to the one defined in [Gavril, 1972] for triangulated graphs A son N_i can be processed in less than $|\Delta(N_i)|\ |P(N)|$ steps Point 3 is achieved through an enumeration method in $O(2^k)$ where k is bounded by the size of $\mathcal{G}_{typ}$ maximum clique

See fig 1 for an example of MCS search tree The search tree is explored by a *breadth first search* that allows to detect first the smallest MCSs that are the most interesting ones according to a least-commitment strategy For each node of the global search tree, we can select all or a given subset of the smallest MCS of the current partial plan The algorithm is further detailed in [Laborie, 1994] where we show this MCS's search procedure to be *sound, complete* and *systematic*

5 3 Minimising MCS's resolvers

To solve a MCS $\phi =< u_1, \ldots, u_n >$, it is sufficient to post one of the constraints $\{(t_i^+ < t_j^-)_{(i,j)\in[1\ n]\times[1\ n],i\neq j}\}$ Thus a MCS of size n has at most $n\ (n-1)$ resolvers

To find the smallest disjunction of resolvers that would be *necessary* and *sufficient* to solve ϕ, we introduce the notion of *equivalence* between two disjunctions of precedence constraints on a partial plan $\mathcal{P}$ $\delta = \{\rho_1, \ldots, \rho_n\}$ and $\delta' = \{\rho'_1, \ldots, \rho'_{n'}\}$ will be said *equivalent on* $\mathcal{P}$ iff each precedence constraint ρ_i of δ implies one of the precedence constraint ρ'_j of δ' on $\mathcal{P}$ and vice versa It can be shown that, given a disjunction δ and a partial plan $\mathcal{P}$, there exists a unique disjunction δ_{min} such that δ and δ_{min} are equivalent on $\mathcal{P}$ and $card(\delta_{min})$ is minimal

Applying this result to the disjunction of resolvers of a MCS ϕ leads to a *minimisation procedure* that runs in $O(n^3)$ (if n is the size of the disjunction) and that consists in removing those resolvers ρ that would imply another resolver ρ' of ϕ The obvious interest of this minimisation is to prune the global search tree of some over-constraining branches

e g , for the MCS $\phi =< u_3, u_4, u_7 >$ in figure 1, we have the following disjunction of resolvers $\delta = \{(t_{11} < t_8), (t_9 < t_{10}), (t_{11} < t_2), (t_3 < t_{10}), (t_3 < t_8)\}$, but, as $(t_9 < t_{10})$ implies $(t_3 < t_{10})$ on $\mathcal{P}$, resolver $(t_9 < t_{10})$ will not be considered

5 4 Integration Planning/Scheduling

Figure 2 shows how the resource analysis module is integrated into the planning system At each node of the global search tree, the resource analysis module selects a certain number of smallest MCSs (§5 2) and computes *their minimal resolvers* (§5 3) *whereas the other* analysis modules focus on threats and pending subgoals (§4 1) The control module, then, can choose a flaw (§4 3) and one of its resolver (§4 2) according to the least-commitment strategy

6 Experimental Results

6 1 MCS Search Procedure

Curves in figure 3 show the performances in the worst case of the MCS search procedure described in §5 2 Times along the Y-axis are given in ms on a Sun Sparc station 10 Along the X-axis is the number of resource proposition for a given type of resource Parameter *size* is the size of the smallest MCS of the analysed partial plan, *nb* is the number of smallest MCS we look for Each point is an average over 50 of the more unfavourable randomly generated problems in terms of temporal constraint density

We can notice the quasi-linear behaviour in average The running time is not fundamentally increased when we look for more than one smallest MCS In practice

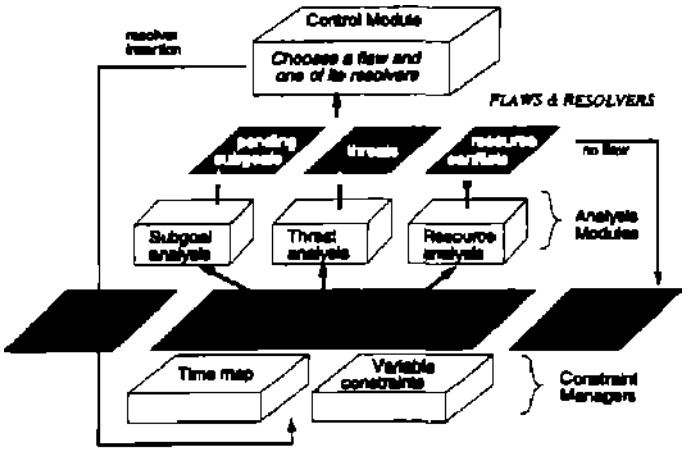

Figure 2 *the IxTeT Planner Architecture*

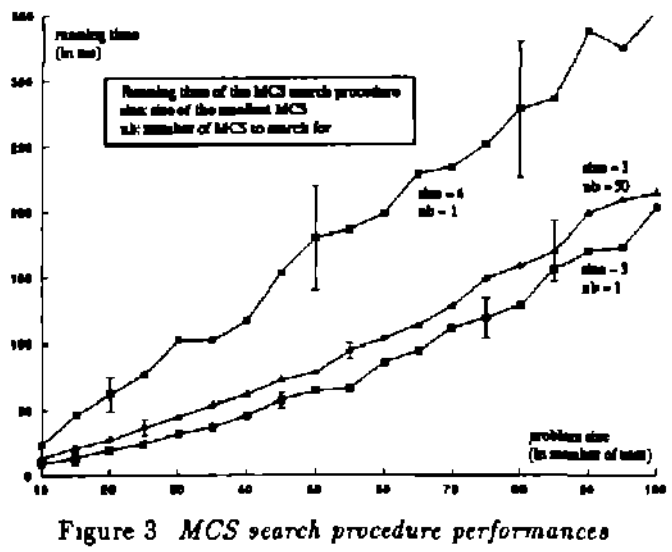

Figure 3 *MCS search procedure performances*

we have about 20 to 50 resource propositions of a given type, *size* is in the order of 3

6 2 Scheduling problems

To compare it to other schedulers, our approach has been tested on some pure scheduling problems, such as the scheduling of equipment compartments integration for the Anane 4 rocket launcher ThiB problem uses 6 sharable resource types (mainly, skilled manpower and test systems) for a total of about 150 resource propositions Because of operations earliest start times and latest completion time, the problem is globally very constrained IxTeT finds a solution in about 5s without backtracking, this time should, for example, be compared to the 1 or 2 mn required by MICRO-BOSS [Sadeh, 1991] to solve problems of the same size

8 3 Integrating Planning and Scheduling

Figure 4 gives a part of the solution plan for planning biological experiments in the COLUMBUS space lab domain which is described by means of 9 state attributes, 11 type of resources and 11 task operators

The uppei part of the IxTeT plan table shows the selected tasks, the lower part, the cumulative use of re-

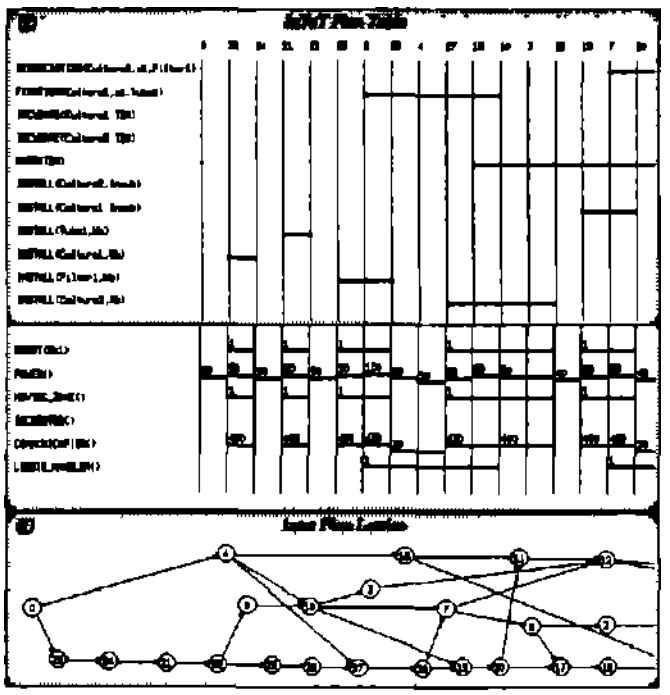

Figure 4 *a Solution Plan*

Figure 4 *a Solution Plan*

sources The horizontal axis refers to the time points in the plan, linearised only for the purpose of an easy display of barcharts The actual precedence constraints of the solution plan are given in the IxTeT plan lattice It should be noticed how our least-commitment approach leads to a highly parallel and flexible solution plan as we try to never over-constrain it

The IxTeT Planner has been tested on many planning problems involving resources (house building, room finishing problem, task-level control for a team of 3 mobile robots, etc) whose generated solution plan contains between 20 to 30 tasks and requires less than 1mn of running time For example, planning for a planetary exploration of Mars with 2 rovers requires about Imn of running time for a three day long plan involving more than 20 tasks, 200 temporal propositions and 100 time-points The plan was generated while taking into account visibility windows between the rovers and an orbiter, sun-light, communication channels and energy constraints The problem was initially stated as a set of data concerning the planet that were to be received by the orbiter within 3 days

6 4 Applicability Domains

Many fields may be interested in such an integrated approach of plan generation and resource management They can be classified according to their needs in terms of plan generation Our approach allows for a great flexibility to express implicit conditions and effects of operations instead of explicit and hard-coded precedence constraints between them as it is traditionally done to describe problems in *factory production* or *project management* Furthermore, domains as *spatial station operations planning* or *management of test systems* need, in addition of resource management, facilities to generate

reconfiguration operations of some sub-ays terns that depend on their last state These operations, because of their duration or because of their use of resources for themselves, may affect the resource management process Finally, activities like *autonomous robots planning,* where plan generation play a major part must take resources into account to produce robust plans

Our approach has been highly motivated by a pre-analysis of these various domains

7 Conclusion and Future Work

IxTeT, is a temporal planner that integrates sharable resource management into the plan generation process This paper shows how this integration is achieved through the crossing of various techniques and ideas from *classical planning, scheduling, graph theory* and *constraint satisfaction problems*

The performances of the resource analysis module allows IxTeT to tackle realistic problems involving sharable resources while the underlying algorithmic ensures the completeness of the search

Future work will mainly consist in improving the expressiveness of the planner We have already integrated producible and *consumable* resources In the same way that consumption of resources can easily be seen as a borrowing over a temporal half-line *[t,* +oo[, we consider the production of resource as an atemporal increase of the resource capacity combined with a borrowing of the resource over] — *oo,t]* This allows to manage producible resources with the possibility of automatically inserting resource production tasks without modifying our MCS detection procedure

References

[Allen *et al*, 1991] J F Allen, H A Kautz, R N Pelavin, and J D Tenenberg *Reasoning about Plans* Morgan Kaufmann Pub ,1991

[Bensana *et al*, 1988] E Bensana, G Bel, and D Dubois OPAL a multi- know ledge- based system for industrial job-ehop scheduling *International Journal of Production Research,* 26{5) 795-819, 1988

[Chapman, 1987] D Chapman Planning for Conjunc tive Goals *Artificial Intelligence,* 32 333-377, 1987

[Curne and Tate, 1991] K Curne and A Tate O-plan the open planning architecture *Artificial Intelligence,* 52 49-86,1991

[Dean *et al*, 1988] T Dean, R J Firby, and D Miller Hierarchical Planning Involving Deadlines, Travel Time and Resources In *Comput Inielhg* , pages 381-398, 1988

[Drabble and Tate, 1994] B Drabble and A Tate The use of optimistic and pessimistic resource profiles to inform search in an activity based planner In *Proceedings AIPS 94,* pages 243-248, 1994

[Erechler *ei al* , 1990] J Erechler, P Lopez, and C Thunot Temporal reasoning under resource constraints Application to task scheduling In *Advances* in *Support Systems Research,* pages 189-194 George E Lasker and Rob bin R Hough (eds), 1990

[Fox and Smith, 1984] MS Fox and S F Smith ISIS a knowledge-based system for factory scheduling *Expert Systems,* 1(1)25-49,1984

[Garcia and Labone, 1994] F Garcia and P Labone Hierarchisation of the search space in temporal planning Technical report, LAAS/CNRS, 1994 Submit ted to EWSP-95

[Gavnl, 1972] F Gavril Algorithms for maximum coloring, maximum clique, minimum covering by cliques and maximum independant set of a chorda! graph *SIAM J Comput ,* 1 180-187, 1972

[Ghallab and Alaoui, 1989] M Ghallab and A Mounir Alaoui Managing Efficiently Temporal Relations Through Indexed Spanning Trees In *Proceedings IJCAI-89,* 1989

[Ghallab and Allard, 1983] M Ghallab and D G Allard *A,* an efficient near admissible heuristic search algorithm In *Proceedings IJCAI 83,* 1983

[Ghallab and Laruelle, 1994] M Ghallab and H Laruelle Representation and Control in Ixtet, a Temporal Planner In *Proceedings AIPS-94,* pages 61-67, 1994

[Ghallab and Vidal, 1995] M GhalUb and T Vidal Focusing on a Sub-graph for Managing Efficiently Numerical Temporal Constraints In *Proceedings FLAIRS-95 (to appear),* 1995

[Hayward *et* af, 1989] R Hayward CT Hoang, and F Maffray Optimizing Weakly Triangulated Graphs *Graphs and Combinatorics,* 5(4) 339-350, 1989

[Labone, 1994] P Labone Planifier avec des contraintes de ressources Technical Report 94077, LAAS-CNRS, Toulouse (France) 1994

[McAlleater and Rosenbhtt, 1991] D McAllester and D Rosenbhtt Systematic nonlinear planning In *Proceedings AAAI 91,* pages 634-639 1991

[Muscetolla, 1993] N Muscetolla HSTS Integrating Planning and Scheduling In *Intelligent Scheduling* Morgan Kaufmann, March 1993

[Sadeh, 199l] N Sadeh *Look-Ahead Techniques for Micro Opportunistic Job Shop Scheduling* PhD thesis, Carnegie Mellon University, march 1991

[Smith *et al,* 1986] S F Smith, MS Fox, and PS Ow Constructing and Maintaining Detailed Production Plans Investigations into the Development of Knowledge-Based Factory *AI Magazine,* 7(4) 45-61, 1986

[Tseng, 1993] E Tsang *Foundations of constraint sat isfaction* Academic Press 1993

[Vilain and Kautz, 1986] M Vilain and H Kautz Constraint propagation algorithms for temporal reasoning In *Proceedings AAAI-86,* pages 377-382, 1986

[Wilkins, 1988] D E Wilkins *Practical Planning* Morgan Kaufmann, San Mateo, CA, 1988

[Yang, 1992] Q Yang A theory of conflict resolution in planning *Artificial Intelligence,* 58 361-392, 1992