Summer School of the
European Network
of Excellence in
AI Planning

Planning & Scheduling with Time & Resources

Malik Ghallab

LAAS - CNRS, Toulouse

Sept.26, 00

http://planet.dfki.de

Planning & Scheduling with Time & Resources

Outline

- **Motivations**
- Representations of time, resources and actions
- Time management
- Resource management
- Planning & scheduling
- Conclusion

Presentation coverage

- HTN techniques:
  - clearly relevant, but not covered here

  => Dana Nau's talk
- Formal techniques:
  - briefly mentioned

  => Paolo Traverso's talk
- Mathematical programming techniques: widely used in resource allocation and scheduling, more and more investigated in planning
- CSP-based techniques: the main focus of this talk

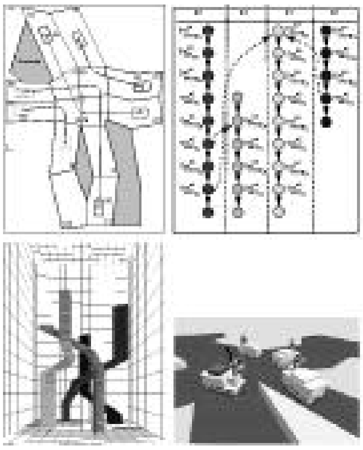
No Blocks world here!

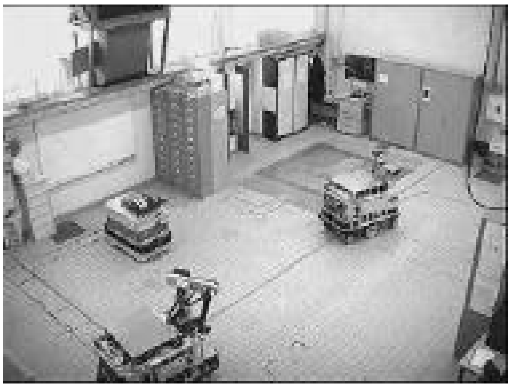
Motivations

Planning for autonomous robots

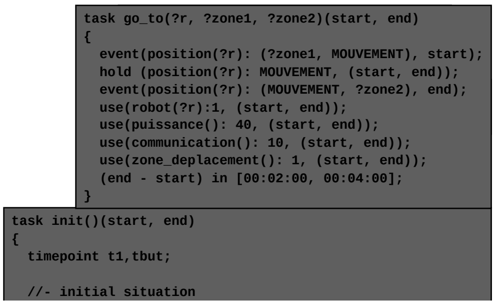
Planning object manipulation tasks

Distributed Planning for multi-robots cooperation



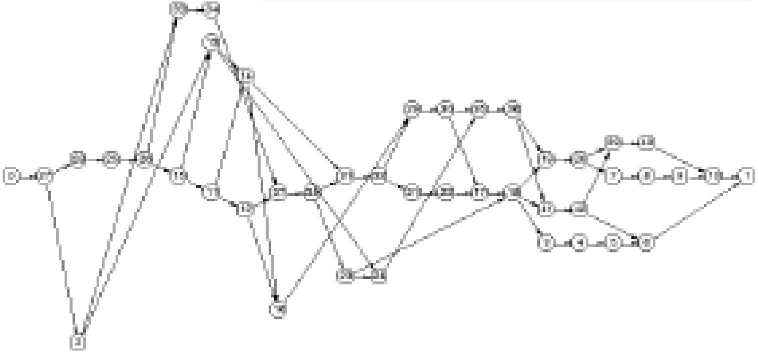


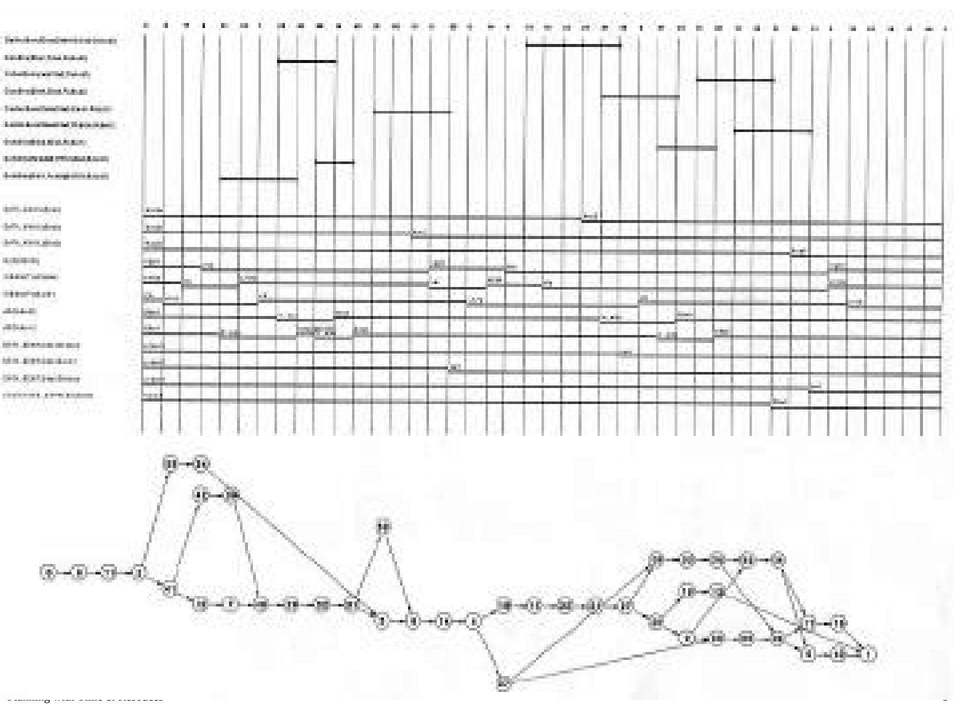
Task planning requirements

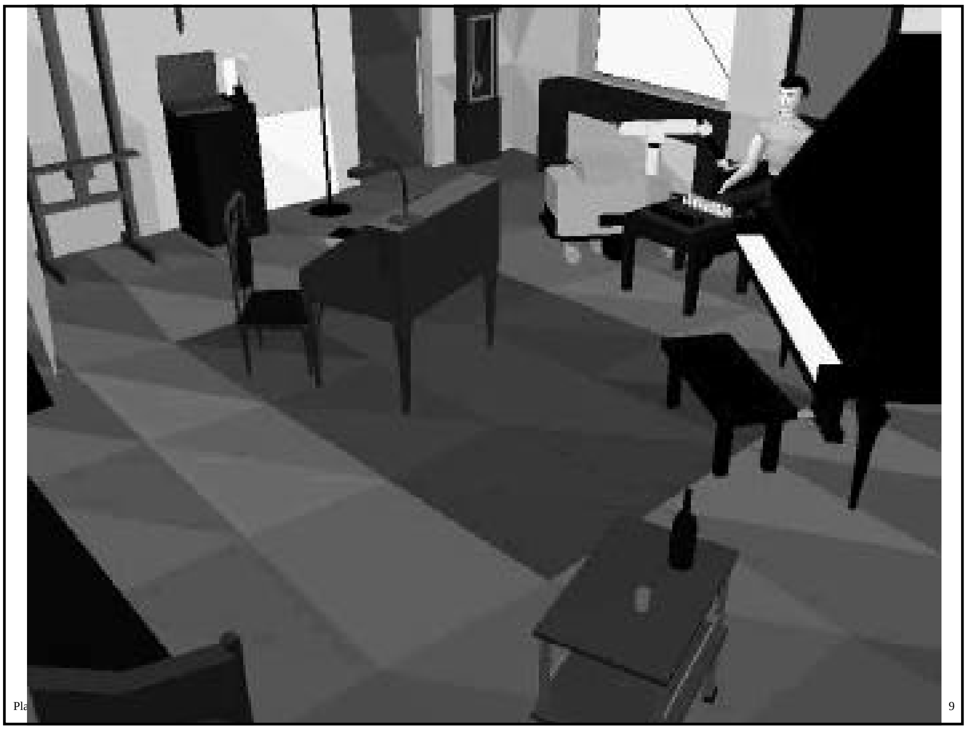


```
task go_to(?r, ?zone1, ?zone2)(start, end)
{
  event(position(?r): (?zone1, MOUVEMENT), start);
  hold (position(?r): MOUVEMENT, (start, end));
  event(position(?r): (MOUVEMENT, ?zone2), end);
  use(robot(?r):1, (start, end));
  use(puissance(): 40, (start, end));
  use(communication(): 10, (start, end));
  use(zone_deplacement(): 1, (start, end));
  (end - start) in [00:02:00, 00:04:00];
}
```

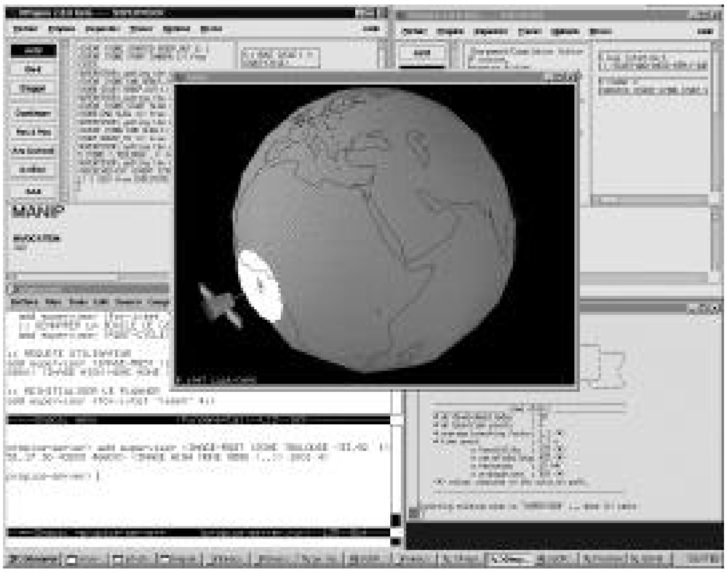
```
task init()(start, end)
{
  timepoint t1,tbut;

  //- initial situation
```

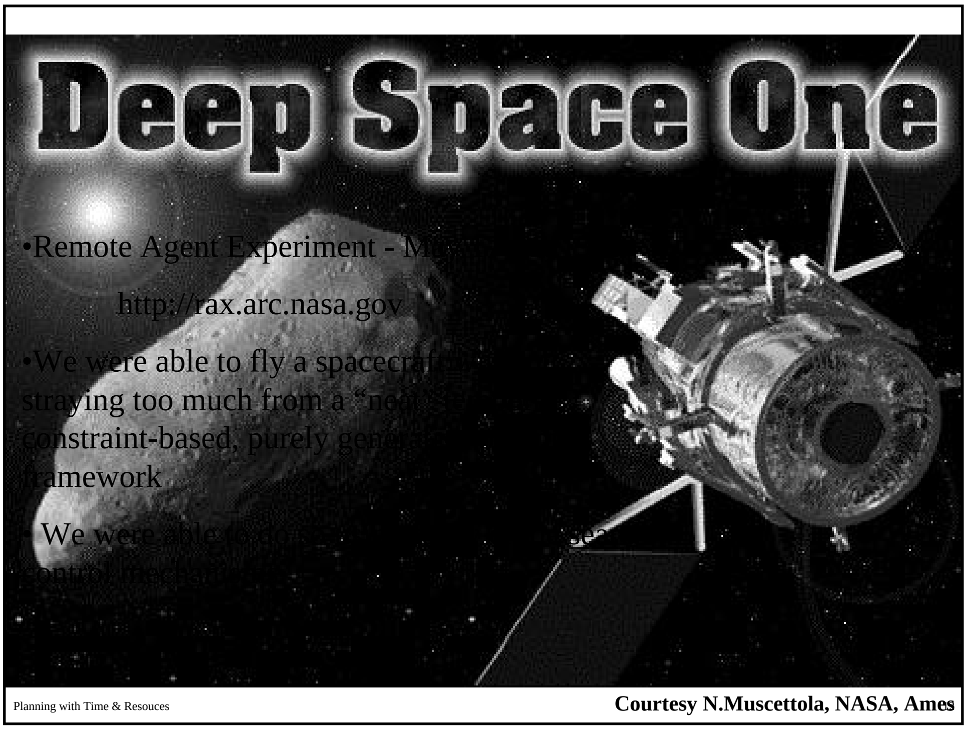






PROBA, an autonomous observation satellite





•Remote Agent Experiment - Ma

http://rax.arc.nasa.gov

•We were able to fly a spacecraft straying too much from a "neat constraint-based, purely gene framework

• We were able to do sea control mecha

Courtesy N.Muscettola, NASA, Ames

Motivation for time in planning

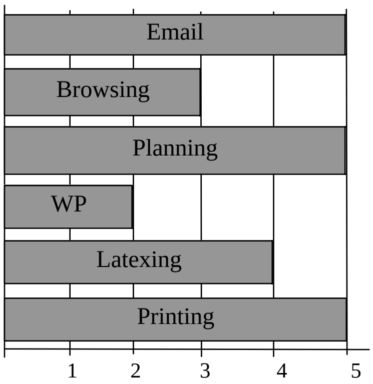
- Time is needed in planning:
  - Planning is the synthesis of a trajectory, a *future* course of actions with *predicted* outcome
  - It is developed inherently with respect to time
  - There are no planning domains without time
  - They are just domains where the restrictive assumptions of classical planning may be acceptable:
    - Actions as instantaneous transitions between states
    - No external dynamics
    - Goals or utilities as desirable states

Application domains mentioned so far in this school

- Scheduling problems
- Autonomous agents
- Software module integrators
- Interactive decision support
- Plan-based interfaces
- Integrated product and process design
- Evacuation operations

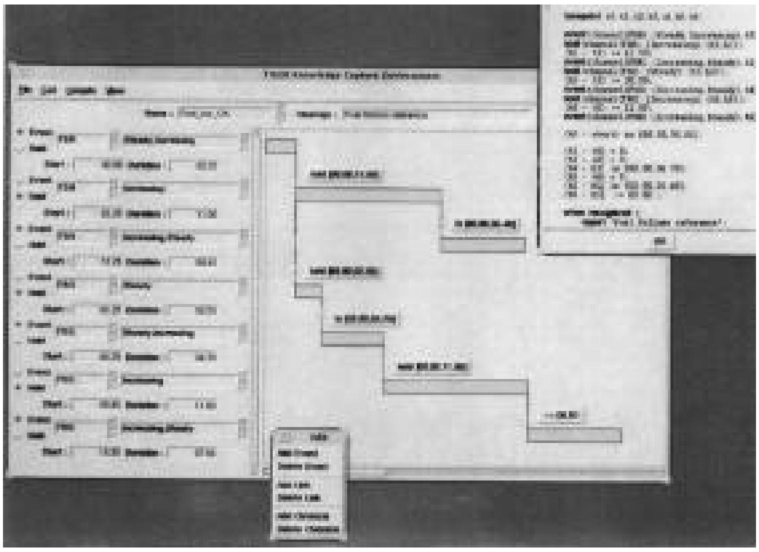
```
 (:action A
     :parameters (?w ?x ?y ?z)
     :precondition (and (P ?w ?x) (Q ?x ?y)
                          (R ?y ?z))
     :effect (and (Q ?x ?z) (not (Q ?z ?y))))
(:action B
     :parameters (?x ?y)
     :precondition (and (Q ?x ?y) (S ?x ?y))
     :effect (and (T ?x)))
(:action C
     :parameters (?x ?y)
     :precondition (and (U ?x) (V ?y))
     :effect (and (P ?x ?y) (not (U ?x))))
(:action D
     :parameters (?x ?y)
     :precondition (and (P ?x ?y) (T ?x))
     :effect (and (U ?x) (not (P ?x ?y))))
```

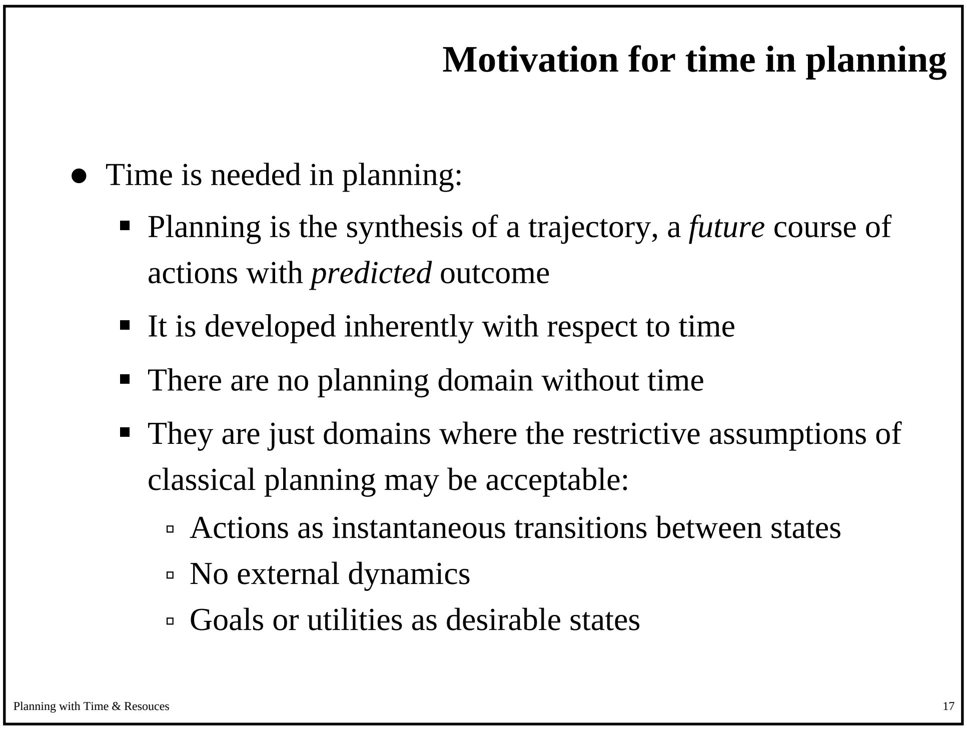
```
(:objects a b c d e f g  h i j k l m n o)

(:init  (Q e k) (Q f k) (Q g k) (Q h k)
     (Q i k) (Q j k) (R k l) (R l m)
     (R m n) (R n o) (S e o) (S f l)
     (S g o) (S h l)  (S i n) (S j o) (U a)
     (U b) (U c) (U d) (V e) (V f) (V g)
     (V h) (V i) (V j))

(:goal (and (T e) (T f) (T g)  (T h) (T i)
     (T j))))
```



Courtesy Derek Long

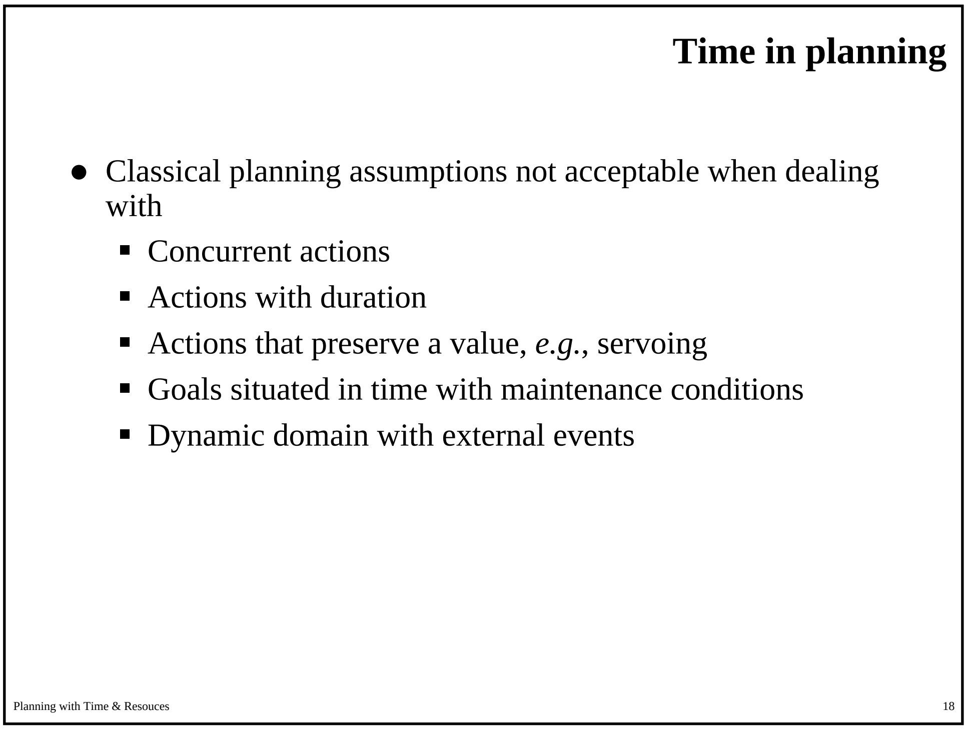
Graphical programming interface





Motivation for time in planning

- Time is needed in planning:
  - Planning is the synthesis of a trajectory, a *future* course of actions with *predicted* outcome
  - It is developed inherently with respect to time
  - There are no planning domain without time
  - They are just domains where the restrictive assumptions of classical planning may be acceptable:
    - Actions as instantaneous transitions between states
    - No external dynamics
    - Goals or utilities as desirable states



Time in planning

- Classical planning assumptions not acceptable when dealing with
  - Concurrent actions
  - Actions with duration
  - Actions that preserve a value, *e.g.*, servoing
  - Goals situated in time with maintenance conditions
  - Dynamic domain with external events

Time for planning

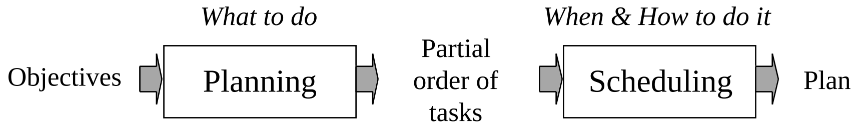
- Time is convenient for planning
  - Time is a peculiar resource :
    - Flows independently of action
    - Is equally available for all actors or processes (parallelism)
    - Time is mathematically structured:
      transitive asymmetric relation
    - It is non reversible
    - It orders causality : *causes precede effects*
  - Requires a representation specific to time, but domain independent, which allows a general temporal reasoning scheme

Resources

- Resources are needed in planning
  - Actions require resources
  - Actions affect resources in a relative way,
    as opposed to absolute change in state propositions
    e.g. *painting a wall with a brush*
    effect : #dry brushes reduced by 1
    wall painted
  - Expressing sharable resources as propositions introduces unnecessary combinatorics
    e.g. naming all equivalent tools
  - Consumable resources require specific handling

Scheduling *vs.* planning

Classical decomposition:



	Planning	Scheduling
KR	*Condition/effect operators*	*Time and resources*
Emphasis	Feasibility	Optimization

Scheduling *vs.* planning

Decomposition Planning - Scheduling:

- Not convenient when interaction planning / scheduling
- No crisp border-line between planning and scheduling
  - When desired tasks are known, it may not be feasible to specify them at a detailed level
    Tasks may appear as goals that need to be planned for
    => Hierarchy of levels from missions to primitive actions
  - Interactive planning: requires a good management of this hierarchy, main decisions and choices left to user, detailed planning, assessment and evaluation automated
  - Controlling autonomous systems:
    an integrated problem, no clear benefit in decomposing it

Planning & Scheduling with Time & Resources

Outline

- Motivations
- **Representations of time, resources and actions**
- Time management
- Resource management
- Planning & scheduling
- Conclusion

Representations of time, resources and actions

- A problem well represented is half solve
- Main representations for planning
  - Time : time-nets, algebraic and geometric representations
  - Resources : ontology, algebraic representations
  - Actions: formal and operator based representations

Representing time: an example

A robotized manufacturing cell has subsystems for

Feeding parts (***F***)
Assembly (***A***)
Inspection (***I***) and
Unloading (***U***)

- parts feeding is done *before or during* inspection or assembly
- unloading is done *after* assembly
- inspection cannot proceed *while* assembly or unloading are performed

How can it be organized ?

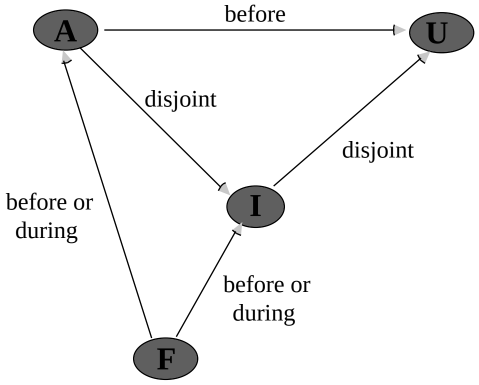
Example

At lunch break I would like

to feed myself (***F***)
to meet Aphrodite (***A***)
to read Irene's letter (***I***) and
to phone to Ursula (***U***)

- I can have lunch *before or during* my meeting with Aphrodite, or while reading Irene's letter
- I want to phone to Ursula *after* meeting Aphrodite
- I cannot read Irene's letter while meeting Aphrodite or while talking on the phone with Ursula

How should I plan my lunch break ?



- Activities spanning over temporal intervals
- Disjunction of symbolic relations constraining these intervals
- Network of binary constraints

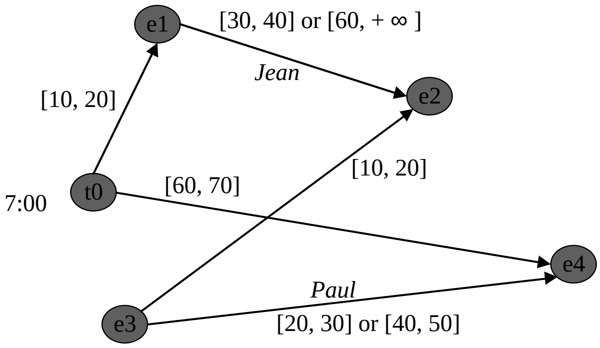
Example

Jean goes to work	by car (30 to 40') or by bus (at least 60')
Paul takes	his bike (40 to 50') or his motor-bike (20 to 30')

Today :

- ▫ Jean has to leave home between 7:10 and 7:20
- ▫ Paul should arrive at work between 8:00 and 8:10
- ▫ Jean has to arrive 10 to 20' after Paul leaves home

- ▪ Is there a coordinated plan for them ?
- ▪ When Paul has to leave home ?
- ▪ Can he use his bike ?
- ▪ What if Jean's car is broken ?
- ▪ Can Jean and Paul meet on their way ?



- time-points representing events
- approximate numerical duration for activities

  [20 ≤ bike commuting ≤ 30] ; [60 ≤ bus commuting]
- precedence constraints between events: [e2 after e3] by [10 to 20]
- temporal localization within absolute reference frame
- disjunction of constraints between events

Temporal networks

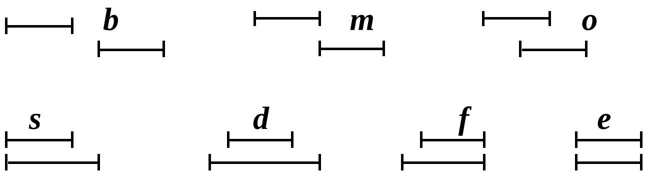
- Nodes as time-tokens: either points or intervals
- Arc as disjunction of primitive relations on points or intervals
  - Set B of primitive relations
  - Constraints between 2 tokens: a subset of B

    $r \quad 2^B$
- Operations and structure over 2^B

Temporal relations

- Points : B = {< , = , > }
  "<" : transitive and asymmetric relation

- Intervals : B = {b, m, o, s, f, d, e, b', m', o', s', f', d'}



Temporal relations

- 2^B set of compound relations :

t1 *different from* t2 ≡ (t1 < t2)   (t1 > t2)

u *disjoint from* v ≡ b(u, v)   b'(u, v)

u *while* v ≡ s(u,v)   d(u,v)   f(u,v)

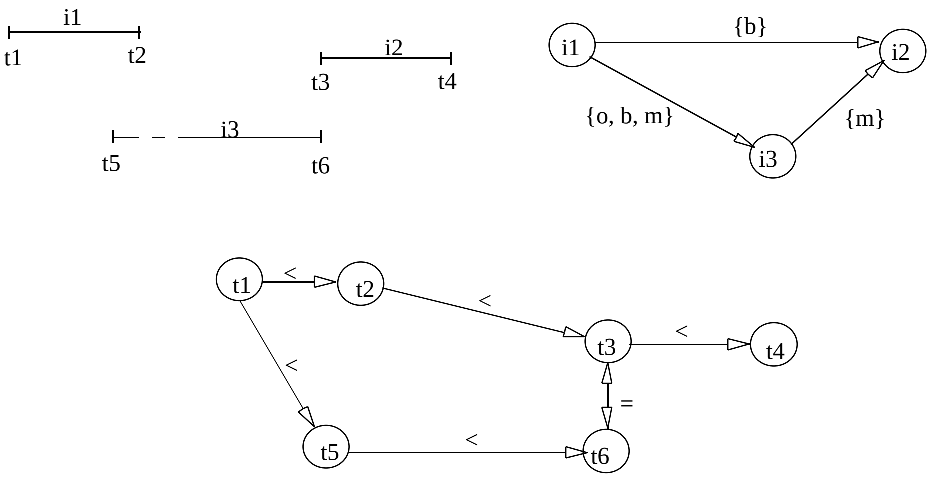
disjoint ≡ {b, b'}

while ≡ {s, d, f}

intersect ≡ {o, s, d, f, e, f', d', s', o'}

synchronize ≡ {m, m'}

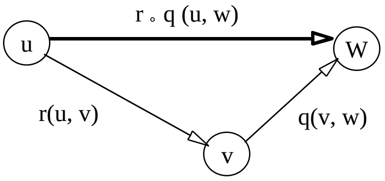
Constraint network



Calculus over 2^B

- inverse of r r' = B - r
- inclusion r q
- union r q
- intersection $r \cap q$
- composition $r \circ q$

if r(u, v) and q(v, w) then $r \circ q$ (u, w)



(2^B, ∩ , ◦) is a semi-ring

(2^B, , ◦) is an algebra

Interval algebra

Point algebra

Composition of temporal relations

- Time-points
  - $\{<\} \circ \{<\} = \{<\}$
- Intervals
  - $\{m\} \circ \{f\} \quad = \{o, s, d\}$
  - $\{m, o'\} \circ \{s, f\} \quad = (m \circ s) \quad (m \circ f) \quad (o' \circ s) \quad (o' \circ f)$

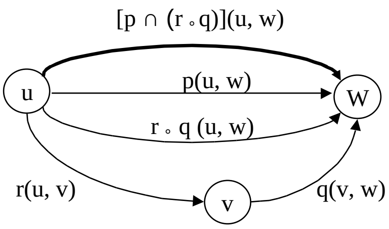
    $= \{m\} \quad \{o, d, s\} \quad \{o', d, f\} \quad \{o'\}$

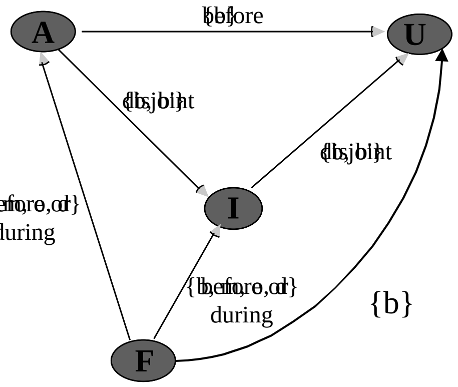
    $= \{m, o, d, s, o', f\}$

Inferred relations

if $p(u,w)$, $q(v,w)$ and $r(u,v)$

then $[p \cap (r \circ q)]\,(u, v)$





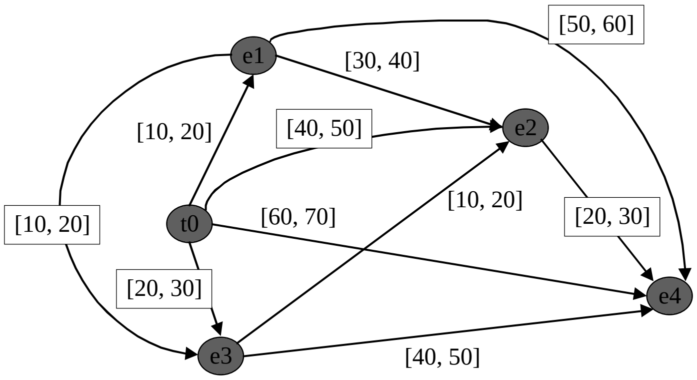
F—>A—>U : {b, m, o, d} ∘ {b} = {b}

Networks of numerical constraints

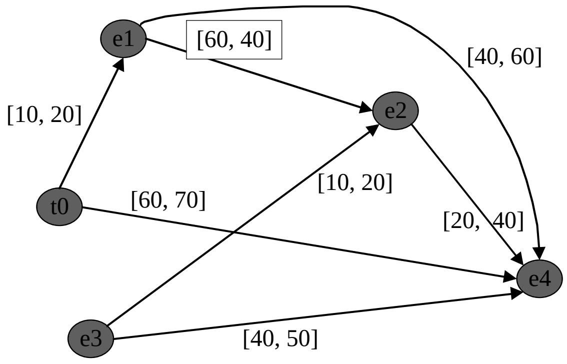
- Simple temporal network (STP) : w/o disjunction
  elementary relation r(t , t') : $\lambda \leq t' - t \leq \mu$
  bounds on time distance
  - $r = [\lambda ; \mu]$ with $\lambda \leq \mu$
  - $r' = [- \mu ; - \lambda]$
  - $r \circ q = [\ \Sigma \lambda_i ; \Sigma \mu_i]$
  - $r \cap q = [\max\{\lambda_i\}; \min\{\mu_i\}]$
- Complex networks with disjunction (TCSP)
  - $(p \quad q) \circ r = (p \quad r) \circ (q \quad r)$
  - $(p \quad q) \cap r = (p \quad r) \cap (q \quad r)$

But for $R= r_1 \quad \ldots \quad r_i$; $P= p_1 \quad \ldots \quad p_j$; $Q= q_1 \quad \ldots \quad q_k$

- $(P \cap Q) \circ R \neq (P \cap R) \circ (Q \cap R)$



e2, e3, e4 : [40-20, 50-10] [0, 30]= [20, 30]
t0, e4, e2 : [60-30, 70-20] [40, 60]= [40, 50]
t0, e2, e3 : [40-20, 50-10] [10, 30]= [20, 30]
e1, e2, e3 : [30-20, 40-10] [-10, 20]= [10, 20]
e1, e2, e4 : [30+20, 40+30] [40, 60] = [50, 60]



e1, t0, e4 : [40, 60]
e2, e3, e4 : [20, 40]
e1, e4, e2 : [40-40, 60-20] [60, + ∞]= [60, 40]

Contradicts $\lambda \le \mu$ => inconsistent network

Other representations of time

Interval i --> start-point, end-point, duration, latency w.r.t. other intervals j, k, ...

$\{s_i, f_i, d_i, w_{ij}, w_{ik}, ...\}$

Uncertainty windows : [lower bound, upper bound]

$\{ [\underline{s}_i, S_i], [\underline{f}_i, F_i], [\underline{d}_i, D_i], [\underline{w}_{ij}, W_{ij}], ... \}$

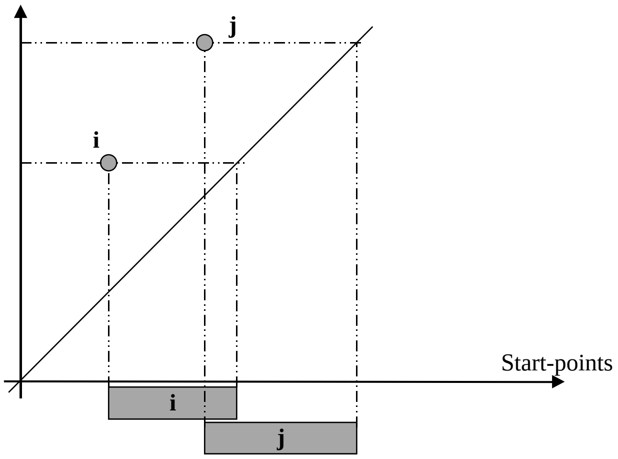
Constraints: $f_i = s_i + d_i$ with $\underline{s}_i \leq s_i \leq S_i$

$s_j = f_i + w_{ij}$

$\underline{f}_i \leq f_i \leq F_i$

$\underline{d}_i \leq d_i \leq D_i$

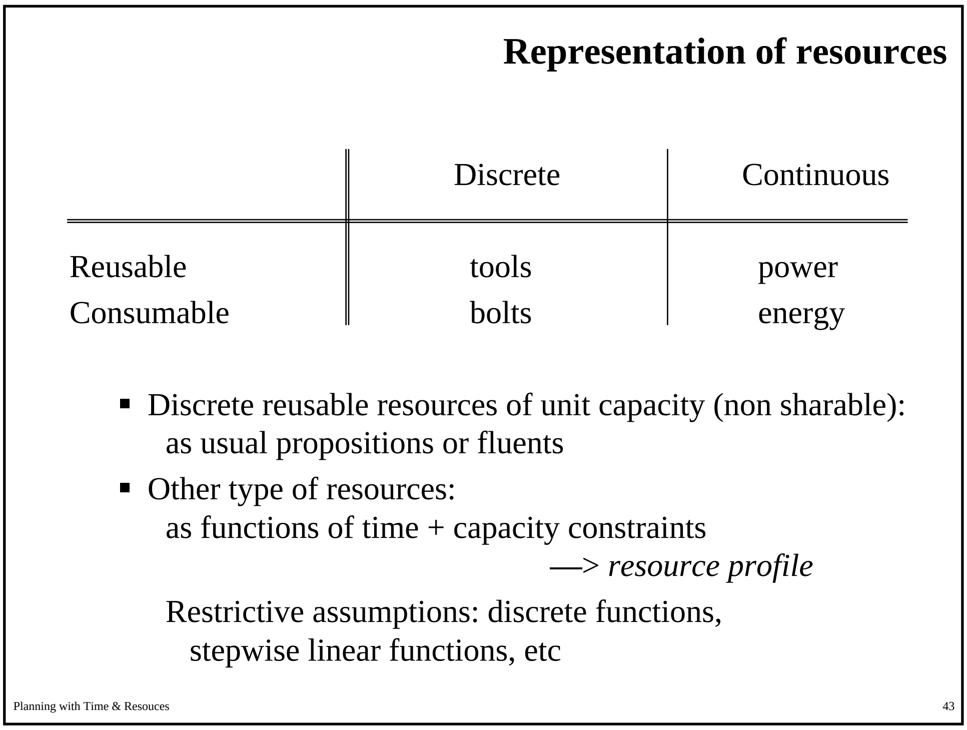
$\underline{w}_{ij} \leq w_{ij} \leq W_{ij}$



Interval (si, fi) : a point in 2D space (start-point, end-point)

Partition of the plan with respect to interval i

if uncertainty windows: 2D domains ([$\underline{s}$i, Si], [$\underline{f}$i, Fi])

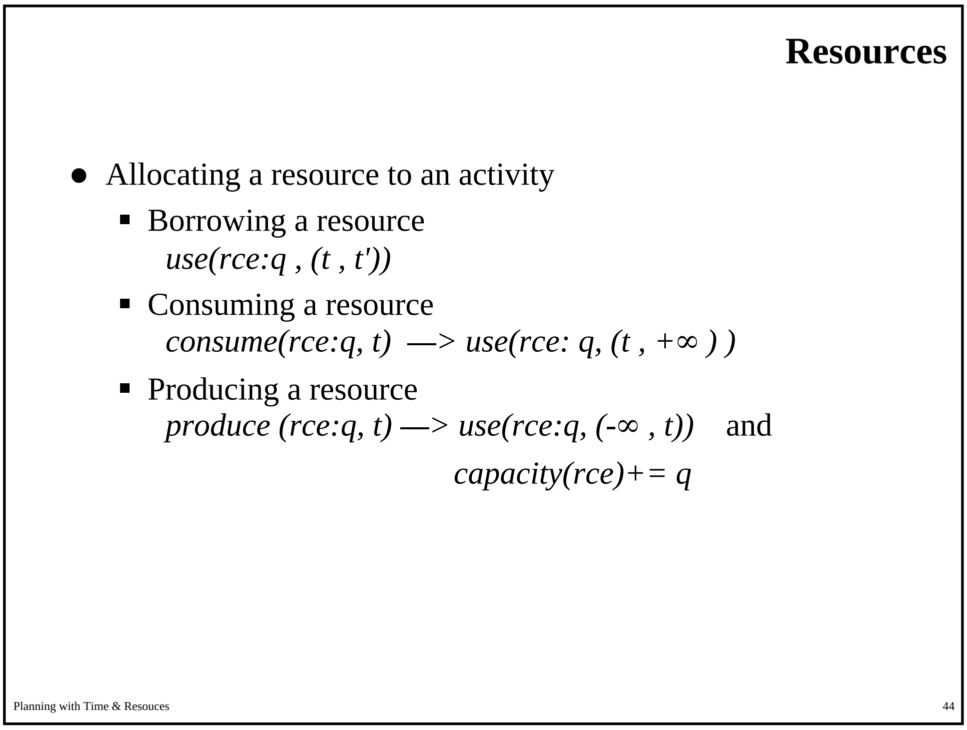


Representation of resources

	Discrete	Continuous
Reusable	tools	power
Consumable	bolts	energy

- Discrete reusable resources of unit capacity (non sharable):
  as usual propositions or fluents
- Other type of resources:
  as functions of time + capacity constraints
  —> *resource profile*

  Restrictive assumptions: discrete functions,
  stepwise linear functions, etc



Resources

- Allocating a resource to an activity
  - Borrowing a resource
    use(rce:q , (t , t'))
  - Consuming a resource
    consume(rce:q, t) —> use(rce: q, (t , +∞))
  - Producing a resource
    produce (rce:q, t) —> use(rce:q, (-∞ , t)) and
    capacity(rce)+= q

Action representations

- Formal representations
  - Modal logic
  - Linear logic
  - Adaptations of classical logic
    - Situation calculus
    - Event calculus
    - Reified logic
- Operator-based representations
- HTN representations
- Functional representations

Formal representations

Temporal logic **:** Time as a modality

Temporal operators : F (future) et P (past) :

$F\phi$: will be true at least once ϕ ;

$P\phi$: has been true at least once ϕ

Operators defined from F and P:

$G\phi \equiv \neg F \neg \phi$: ϕ will always be true

$H\phi \equiv \neg P \neg \phi$: ϕ has always been true

Distance information $F(n)\phi$: ϕ will be true in ***n*** units of time

$P(n)\phi$: ϕ has been true ***n*** units ago

Relative localization : $S(a, \phi)$: ϕ has been true since ***a***

$U(a, \phi)$: ϕ will become true until a

Event calculus

- Idea : focus on local events

  act(e1, exit).
  actor(e1, jean).
  source(e1, home).
  time(e1, 7.30).

CHICA

- Express in Horn clauses
  - Domain axioms concerning described relations and events

    initiates(E, into(X, Y)):- act(E, enter), actor(E, X),
    destination(E, Y).

    terminates(E, into(X,Y)) :- act(E, exit), actor(E, X),
    source(E, Y).
  - General axioms relative to time
- Use logic and constraint programming
  with an abduction based approach to planning

Event calculus

- General axioms:

start(after(e, p), e). *e------>*

start(before(e1, p), init(before(e1, p)).

start(before(e1, p), e2) :- equal(after(e2, p), before(e1, p)).

equal(after(e2, p), before(e1, p)) :-
hold(after(e2, p)), hold(before(e1, p)),
precede(e2, e1), not(broken(e2, p, e1).

e2------->
<--------- e1

broken(e2, p, e1) :- hold(before(e, q)), exclusive(p, q),
precede(e2, e), precede(e, e1).

broken(e2, p, e1) :- hold(after(e, q)), exclusive(p, q),
precede(e2, e), precede(e, e1).

Event calculus

stop(before(e, p), e).

stop(after(e1, p), e2) :- equal(after(e1, p), before(e2, p))

precede(e1, e2) :- time(e1, t1), time(e2, t2), t1 < t2 .

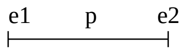
hold(before(e, p)) :- terminates(e, p).

hold(after(e, p)) :- initiates(e, p).

holdAt(p ,t) :- hold(after(e, p)), in(t, after(e, p)).

holdAt(p ,t) :- hold(before(e, p)), in(t, before(e, p)).

in(t, p) :- start(p, e1), stop(p, e2),
time(e1, t1), time(e2, t2), t1 < t, t < t2



Reified logic

- Reify : to name a formula into an object that can be transformed and explicitly manipulated
- Domain relations: as terms, temporally qualified

$Hold(on(a, b), intvl7)$

$\forall\, x, y, t1, \exists\, t2, \forall t3, t4\ :$

$[(t3 \leq t1) \wedge (t2 \leq t4) \wedge$

$Hold\ (position(robot, x); t1) \wedge$

$type(?route, trajectory(x, y)) \wedge$

$Hold(feasible(?route); (t3\ .\ t4)) \wedge$

$Hold(traverses(robot, ?route); (t1\ .\ t2))\ ]$

$Hold(position(robot), y); t2)$

Reified logic

- Planning operator

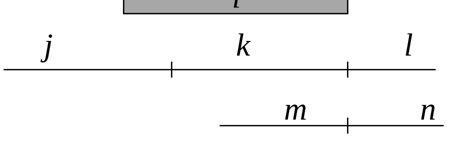
$\forall i, a, b, e \quad putton\ (a, b, e, i) =>$

TRIPS

$\exists\, j, k, l, m, n \quad clear(a, j) \wedge holding(a, k)$

$clear(a, l) \wedge clear(b, m) \wedge on(a, b, n)$

$o(j, i) \wedge f(k, i) \wedge m(j, k) \wedge m(i, l) \wedge \{f, f'\}(i, m) \wedge m(i, n)$



Reified logic

- Domain constraints:

  clear(b, t1) ∧ on(a, b, t2) => disjoint(t1, t2)

  clear(a, i) ∧ holding(a, j) => disjoint(i, j)

- Given a domain model W and a goal G, a plan is a set of assumptions A1, …, An such that

  W $\models$ *A1 ∧ … ∧ An => G* and

  A1 ∧ … ∧ An consistent

- Limited capability for reasoning about the future: cannot predict external events or plan to change them, but can construct plans that take them into account

Operator-based representations

- Extension of Strips operators with temporal information: partial order planning does not require instantaneous state transitions

Deviser

(stack action
((holding x)(clear y) —> ((clear x)(on x y))
(duration (funct x y)))

State change at the end of the action
(if not :decomposition into several actions with specific constructs: *initiate* and *consecutive)*

(goals ((window after 15) (duration 20) (on a b) (on b c))

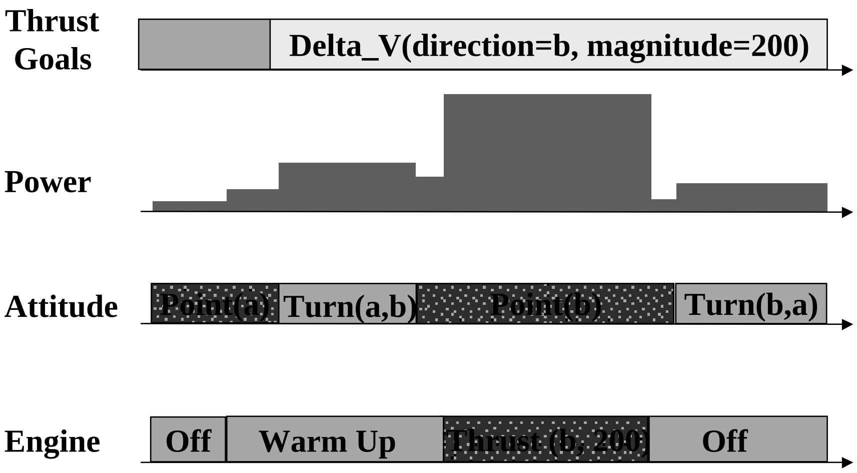
(alarm event (context (alarm set t)) nil —> ((alarm sounding))
(window at t))

Operator-based representations

Zeno

```
(define (operator fly)
    :parameters (m l)
    :resources ((plane m))
    :at-time (s e)
    :pre (:and (:neq m l) (at s plane m)
              (:forall (time t "[s,  e]") (> (fuel t plane) 0)))
    :effect (:and (= (- e s)(/ (dist m l) 650))
                   (:influence s e (fuel plane) (- (/ 650 (mpg plane))))
                   (:forall o (:when (:forall (time t "[s , e]")(in t o)))
                                  (:and (at e o l)
                                        (:forall (time t "[s , e]")(:not (at t o m))))))
```

Parallel Timelines

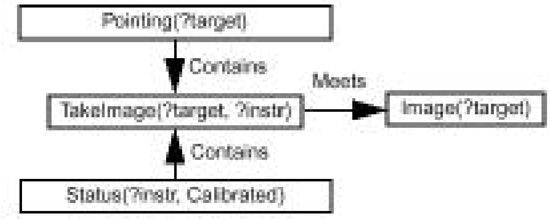


Courtesy N.Muscettola, NASA, Ames

RAX/PS

TakeImage (?target, ?instrument)

contained-by	Status(?instrument, Calibrated)
contained-by	Pointing(?target)
meets	Image(?target)



Turn (?target)	met-by	Pointing(?direction)
	meets	Pointing(?target)

Calibrate (?instrument)		
	met-by	Status(?instrument, On)
	contained-by	CalibrationTarget(?target)
	contained-by	Pointing(?target)
	meets	Status(?instrument, Calibrated)

Chronicle Representation

IxTeT

- Time : linearly ordered discrete set of instants
- Multi-valued domain attributes
  - ***Rigid*** attributes:

    connected(room1, room2); situated(printer1, room3)
  - ***Flexible*** attributes: fluents and resources
    - ***Contingent*** fluents

      day-light; delivery(material)
    - ***Controllable*** fluents, ranging over discrete values, set by actions

      location(?robot) ∈ SITES
    - ***Resources***: constant, real values, relatively changed by actions

      bricks(?storage) ∈ [0, 100]

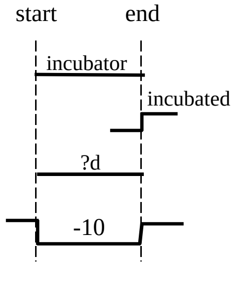
Chronicle Representation

- Predicates: temporally qualified expressions
  - ***Events*** : instantaneous change of the value of a fluent
    event(f(x): (a, b), t)
  - ***Assertions*** : persistence of the value of a fluent over an interval
    hold (f(x): a,   (t1, t2))
  - ***Resource*** predicates
    use (r(x): q, (t, t'))
    consume(r(x): q, (t, t'))
    produce(r(x):q, (t, t'))
- Constraints
  - ***Temporal*** constraints
    $t < t'$; $t - t' \in [dmin, dmax]$
  - ***Atemporal*** constraints
    $x = y$; $x \neq y$; $x \in D$; $(x \in D) \Rightarrow (y \in D')$

Planning operators

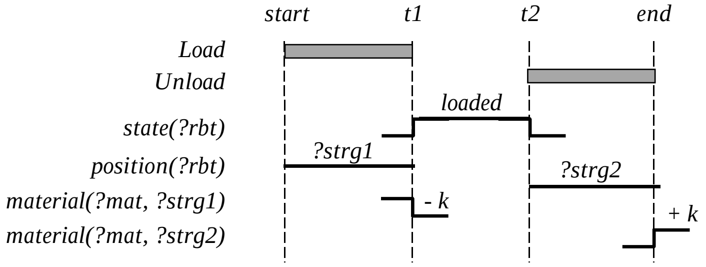
- Conjunction of
  - Predicates : assertions (*hold*), events and resource predicates
  - Subtasks
  - Temporal and atemporal constraints
  - Conditional expressions

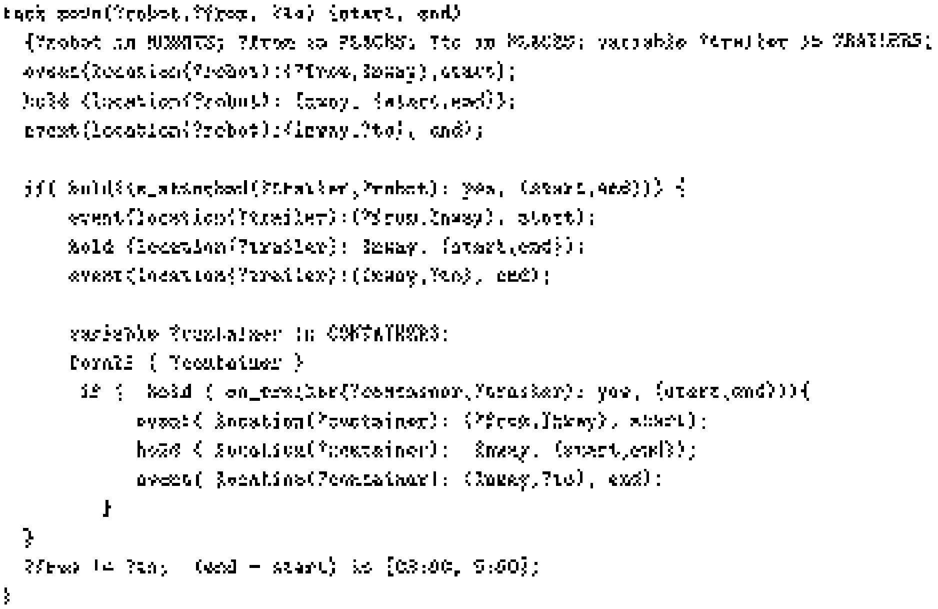
```
task Incubate (?elt, ?d) {
    hold(position(?elt): incubator, (start, end))
    event(state(?elt):(?s, incubated), end)
    hold(temp(incubat): ?d, (start, end))
    use(power: 10, (start, end))
    (end-start) in [9., 10.]
}
```



A planning operator

task Transport-material (?mat, ?q, ?strg1, ?strg2, ?rbt) {
  timepoint t1, t2
  task Load (?mat, ?q, ?strg1) (start, t1));
  task Unload(?mat, ?q, ?strg2) (t2, end));
  hold (state(?robot) : loaded, (t1, t2));
  ?strg1 ≠ ?strg2 ; ?rbt in ROBOTS
  ?t1 < ?t2 ;   end - start in [1., 2.];
}





```
task move(?robot,?from, ?to) (start, end)
  {?robot in ROBOTS; ?from in PLACES; ?to in PLACES; variable ?trailer in TRAILERS;
  event(location(?robot):(?from,away),start);
  hold (location(?robot): away, (start,end));
  event(location(?robot):(away,?to), end);

  if( hold(is_attached(?trailer,?robot): yes, (start,end))) {
      event(location(?trailer):(?from,away), start);
      hold (location(?trailer): away, (start,end));
      event(location(?trailer):((away,?to), end);

      variable ?container in CONTAINERS;
      forall { ?container }
       if (  hold ( on_trailer(?container,?trailer): yes, (start,end))){
            event( location(?container): (?from,away), start);
            hold ( location(?container):  away, (start,end));
            event( location(?container): (away,?to), end);
         }
  }
  ?from != ?to;  (end - start) in [03:00, 5:00];
}
```

Problem description

- Domain description

  Rigid attributes, fluents, resources, constants and domain constraints

- Problem description: ***input chronicle***
  - ***Explained*** expressions on fluents and resources
    - Initial facts
    - Expected evolution
      events and assertions on contingent and controllable fluents
    - Resource availability profiles
  - ***Unexplained*** expressions on fluents (goals)
  - Temporal and atemporal constraints

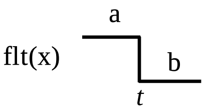
Features of the representation

- No explicit distinction between preconditions and effects

  event(f(x): (a, b), t) may express both:

  Precond

  Effect



- No explicit distinction between activity and assertion
- Handles contingent events
- Flexible *wrt* time and atemporal variables, including resources

- ■ Deterministic representation
- ■ Assumes a complete description

Planning & Scheduling with Time & Resources

Outline

- Motivations
- Representations of time, resources and actions
- **Time management**
- Resource management
- Planning & scheduling
- Conclusion

Time Map Manager

Tasks of a TMM: managing time-tokens of a temporal data-base

Tokens(f)= {i_1, ..., i_k} : temporal qualifications of fluent f

Constraints between time-tokens

- Elementary queries and updating
  - ***Positioning*** two 2 tokens
  - ***Updating*** : adding/removing tokens or constraints while maintaining the consistency of the constraint network

TMM

- Complex queries: temporal properties, e.g.

Simultaneity of two properties:

$$hold(p{:}a;\ t,\ t') \wedge hold(q{:}b;\ t,\ t')$$

Precedence between two events :

$$event(p{:}a,\ b,\ t) \wedge event(q{:}c,\ d;\ t') \wedge (\ t' < t)$$

Complex queries :

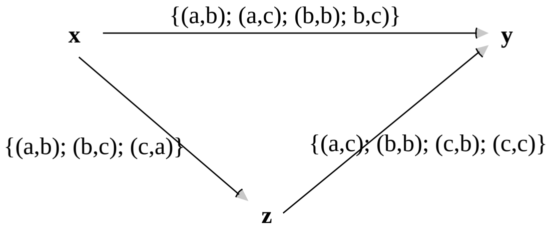
$$hold\ (position(robot){:}\ a;\ t,\ t1) \wedge$$
$$type(?route,\ trajectory(a,\ b)) \wedge$$
$$duration(?route,\ \delta) \wedge$$
$$hold(state(?route){:}\ feasible;\ t3\ ,\ t4) \wedge$$
$$(t3 \leq t1) \wedge (t4 - t1) \leq \delta$$

Basic elementary task of a TMM: managing temporal constraints

Managing temporal constraints: CSP

Example :

x, y, z {a, b, c}



Resolution

$$M_{12} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \qquad M_{13} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad M_{23} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

$M_{ij} \leftarrow M_{ij}\quad [M_{ik} \cdot M_{kj}]$ with $M_{ji} = M_{ij}^T$

CSP

Path consistency algorithm

```
Repeat until all $M_{ij}$ stabilize
    Repeat for $1 \le i, j, k \le n$
        $M_{ij} \leftarrow M_{ij} \quad [M_{ik} \cdot M_{kj}]$
end
```

$M_{12}= \begin{pmatrix} 0\,1\,1 \\ 0\,0\,1 \\ 0\,0\,0 \end{pmatrix} \qquad M_{13} = \begin{pmatrix} 0\,1\,0 \\ 0\,0\,1 \\ 0\,0\,0 \end{pmatrix} \qquad M_{23}= \begin{pmatrix} 0\,0\,0 \\ 0\,1\,0 \\ 0\,1\,1 \end{pmatrix}$

hence : x ={a, b}, y =z ={b, c}

- Algorithm terminates (for discrete domains) with a path-consistent net
- If distributivity property (· distributes over) then algorithm complete: any path-consistent net is consistent

Path consistency for temporal nets

Results and algorithms extended to temporal constraint networks:

Algorithm PC1

```
for    $1 \le i, j, k \le n$   ,  i, j, k distinct
    $r_{ij} \leftarrow r_{ij} \quad [r_{ik} \circ r_{kj}]$
end
```

Incremental path consistency

PC2

```
until Update  is empty do
        Remove an element $r_{ij}$ from Update
        For $1 \leq k \leq n$ , $k \neq i, j$
                Update ← Update     Modify(i, k, $r_{ij} \circ r_{jk}$)
                Update ← Update     Modify(j, k, $r_{ji} \circ r_{ik}$)
end
```

Modify (i, j, r)

```
        r ← r    $r_{ij}$
        if $r = \emptyset$ then Exit (Inconsistent)
        if $r \neq r_{ij}$ then do  $r_{ij}$ ← r
                                   return($r_{ij}$)
                     else return(nil)
```

Properties of Path consistency

- PC keeps a complete graph
  - Positioning two tokens : trivial task
  - Updating:
    - Adding : incremental propagation
    - Removal : re-compute everything from input constraints
  - Complexity : time in $O(n^3)$, space in $O(n^2)$
  - Completeness:
    - PC not complete for interval algebra —> tractable subsets
    - PC complete for point algebra and for STP

Can it be improved ?

Time lattice approach

Representation :

- At the user level: point and restricted interval algebra
- At the system level:point algebra

Network organization : a compromise between

- maintaining a complete graph
- maintaining only input constraints

Primitives used : **($u \leq v$)** and **($u \neq v$)**

Consistent network :

no closed loop through 2 nodes u ≠ v
collapsing closed loops of equal tokens : acyclic graph

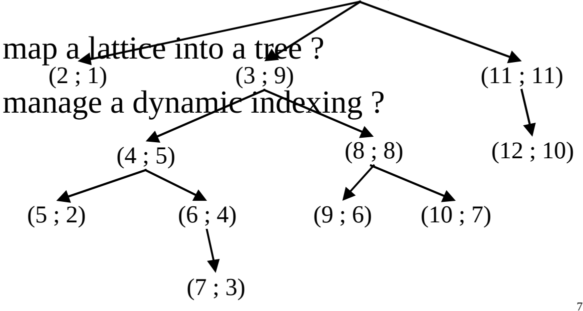
adding a root => rooted acyclic graph or time lattice

- Consistency checking: maintaining an acyclic graph
- Positioning 2 tokens : path search in network

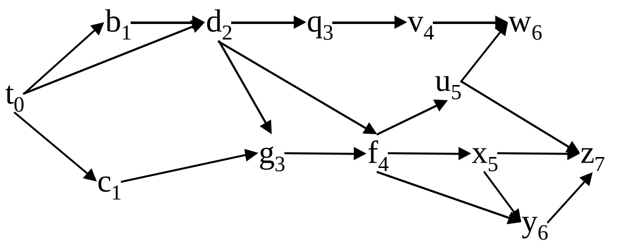
u is before v : iff   path from u to v

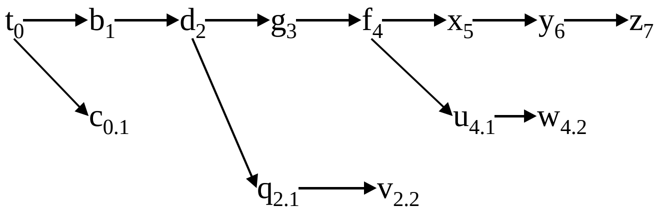
Ancestral information in indexed tree: in O(1)
e.g. if pre (x) < pre (y) and post(x) > post(y)
then y descendant of x

Problems: - how to map a lattice into a tree ?
- how to manage a dynamic indexing ?



Time lattice





Spanning tree

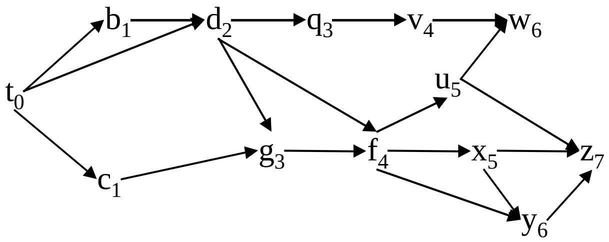
If $I(u)=(i_1\ i_2\ ...\ i_k)$ and $I(v)=(j_1\ j_2\ ...\ j_h)$ then :

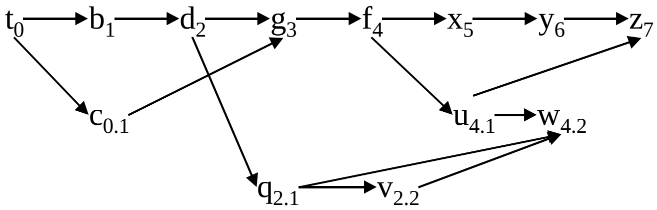
$v \quad s^*(u)$ iff $k \leq h$, $(i_1\ i_2\ ...\ i_{k-1}) = (j_1\ j_2\ .j_{k-1})$ and $i_k \leq j_k$ (i)

Positioning *u* and *v* in spanning tree

- if rank(u) = rank(v) then u and v are not related in Tree;
- if rank(u) < rank(v) :
  - either condition (i) holds : v is a descendant of u
  - or u and v are not related in Tree;
- if rank(u) > rank(v) : switch u and v then check condition (i)

Time lattice





Spanning tree + residual arcs

Positioning of two points

u *before* v ⇔ (rank(u) < rank(v)) [(v s*(u))
(v a(u))
(w a(u) | w before v)]

Compare (u, v)

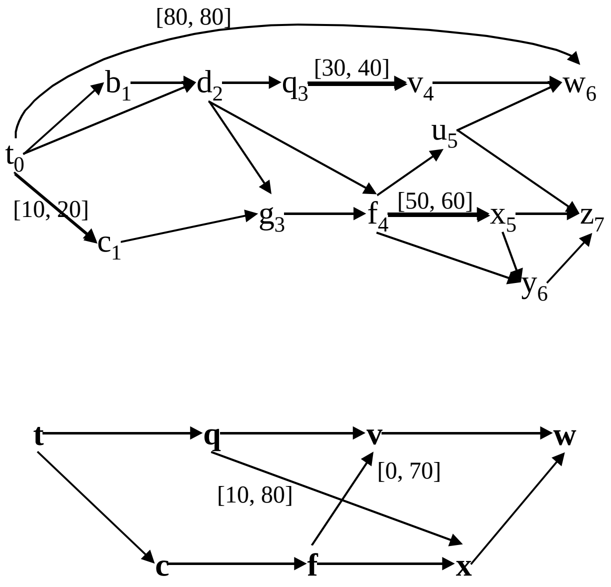
```
if  rank(u) = rank(v)  then  return(nil)
else if r(u) > r(v)  then  Locate (v, u)
      else      Locate (u, v)
```

Locate (u, v)

```
if [v     s*(u) or v     a(u)] then return (u before v)
else for some w     a(u)
      if [r(w) < r(v) and  Locate(w, v) returns (w before v)]
            then return (u before v)
            else return nil
```

Handling numerical constraints

- Managing the time lattice by path search with indexed spanning tree
- Managing numerically constrained sub-lattice by PC algorithm
- Constraints induced in the sub-lattice : filtered by dominance relations, then added back into the lattice
- Important benefit if proportion of numerical constraints is low



Planning & Scheduling with Time & Resources

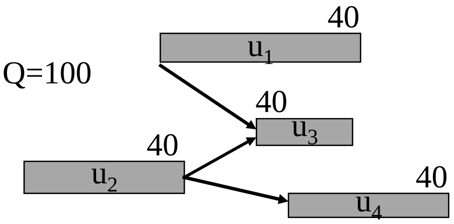
Outline

- Motivations
- Representations of time, resources and actions
- Time management
- Resource management
- Planning & scheduling
- Conclusion

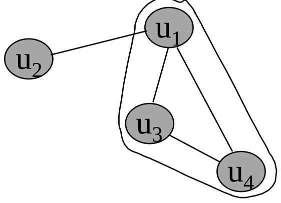
Constraint-based approach

- Resource allocation u_i : use(rce(x_i):q_i, (t_i, t_i'))
  q_i : constant, real value
- Critical set for a resource : a subset of allocations U= {u_i}
  - Over-consuming : $\sum q_i > Q$
  - Corresponding intervals may possibly overlap
- Minimal critical set: a critical set U minimal for set inclusion
- Solving a minimal critical set
  - Allocation constraints $x_i \neq x_j$ or
  - Scheduling constraints $t'_i < t_j$ or
  - Resource production

Resource conflicts

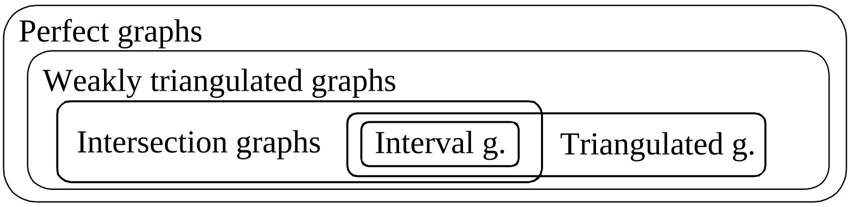


Intersection graph



Minimal critical set :

minimal over-consuming clique of intersection graph



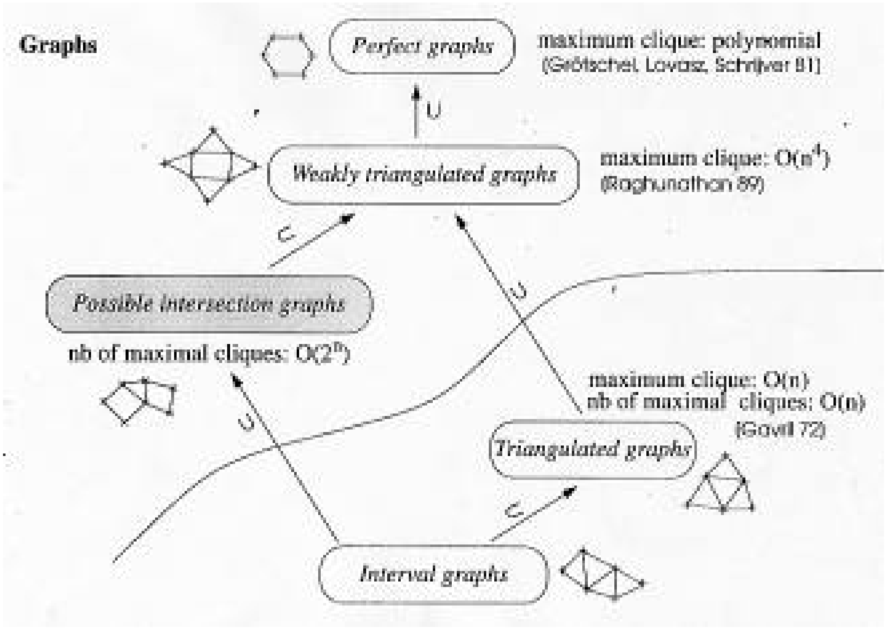
Perfect graphs

Weakly triangulated graphs

Intersection graphs

Interval g.

Triangulated g.



Graphs

Perfect graphs

maximum clique: polynomial (Grötschel, Lovasz, Schrijver 81)

Weakly triangulated graphs

maximum clique: O(n^4) (Raghunathan 89)

Possible intersection graphs

nb of maximal cliques: O(2^n)

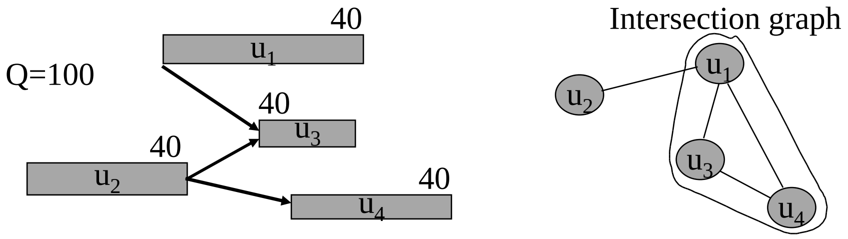
maximum clique: O(n)
nb of maximal cliques: O(n)
(Gavril 72)

Triangulated graphs

Interval graphs

Solving resource conflict

- Finding minimal critical sets: an efficient backtrack search algorithm for intersection graph, in $O(n^k)$, k: size of MCS
- Simplifying resolvers by dominance relation



$\{u_1^+ < u_3^-, u_1^+ < u_4^-, u_3^+ < u_4^-, u_4^+ < u_3^-, u_4^+ < u_1^-,$
$\text{produce}(q>20) \ll u_4^-, \text{produce}(q>20) < u_3^-, \text{produce}(q>20) < u_1^-\}$

Mathematical programming approach

- Required for more complex patterns of resource use
  e.g. energy-consumption= f(speed, distance)

- Testing the consistency of set of resource allocations
  - Linear function: Gaussien elimination and Simplex
  - Non-linear functions: postpone checking till some variables are instantiated

  - In **Zeno**
  - Linear programming mixed with SAT LPSAT

mixed integer programming

Planning & Scheduling with Time & Resources

Outline

- Motivations
- Representations of time, resources and actions
- Time management
- Resource management
- Planning & scheduling
- Conclusion

Partial-order causal link planning

Least commitment regression planning
search space : <P, agenda> where P : <A, L, C>

Zeno

```
POCL(<P , agenda>)
    if C inconsistent then return failure
    if agenda= Ø then return P
    do
        remove a goal g from G
        if g not primitive then reduce g and update P
        else    if g is metric then post g
                else chose a provider for g, add causal link,
                        and resolve constraint
    end
```

POCL

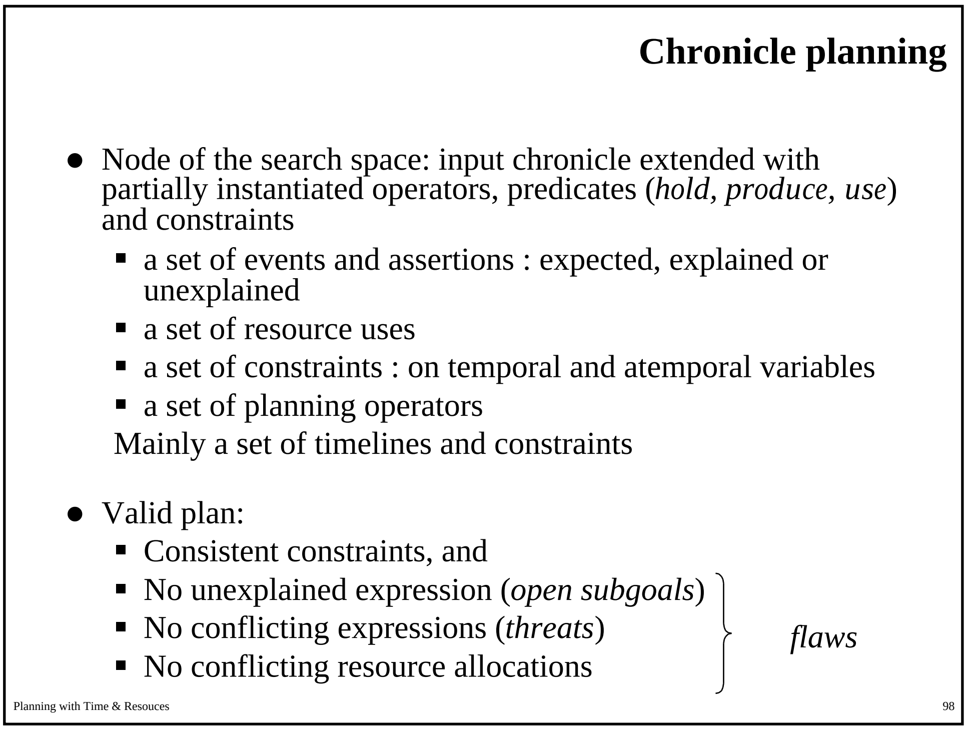
- Goal reduction: - non deterministic choice for disjunction
  - interval splitting
- Posting metric constraints:
  at interval end-points (piecewise linearity assumption)
- Constraints handling:
  - Codesignations: maintaining equivalence classes
  - Linear equations: Gaussien elimination
  - Linear inequalities: Simplex
  - Non-linear equations: delayed until linearized by variable instantiation
  - Temporal constraints on time-points: Warshall transitive closure

Constraint-based interval planning

- A slight shift from POCL: instead of partial plans, a dynamic CSP on a set A of temporally qualified assertions

RAX/PS

```
CBI(A, C)
    if C inconsistent then return failure
    if all a     A have causal explanation then return(A, C)
    do :
        select a with no causal explanation
        either choose a'     A such that (a',c) explains a and
                return CBI(A-{a}, C    {c})
        or choose and operator O =(A', C') that explains a and
                return CBI(A-{a}     A', C     C')
    end
```



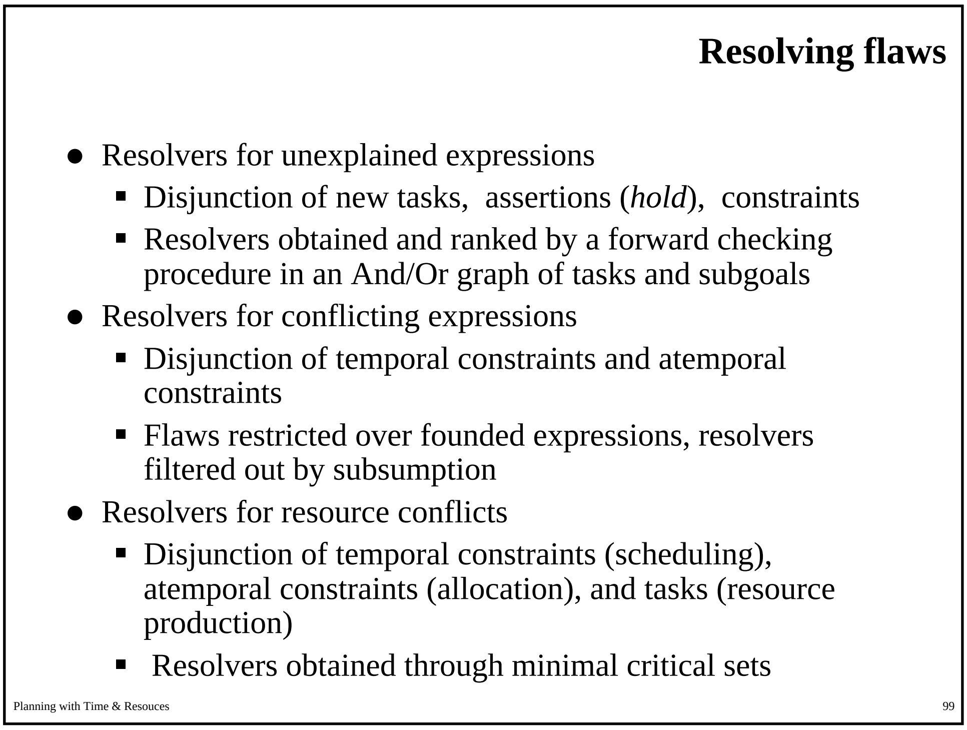
Chronicle planning

- Node of the search space: input chronicle extended with partially instantiated operators, predicates (*hold, produce, use*) and constraints
  - a set of events and assertions : expected, explained or unexplained
  - a set of resource uses
  - a set of constraints : on temporal and atemporal variables
  - a set of planning operators

  Mainly a set of timelines and constraints

- Valid plan:
  - Consistent constraints, and
  - No unexplained expression (*open subgoals*)
  - No conflicting expressions (*threats*)
  - No conflicting resource allocations

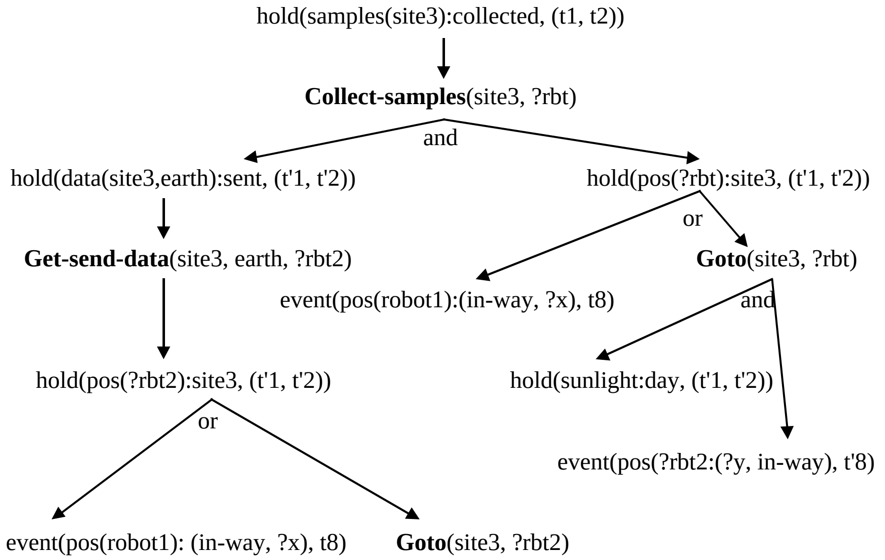
flaws



Resolving flaws

- Resolvers for unexplained expressions
  - Disjunction of new tasks, assertions (*hold*), constraints
  - Resolvers obtained and ranked by a forward checking procedure in an And/Or graph of tasks and subgoals
- Resolvers for conflicting expressions
  - Disjunction of temporal constraints and atemporal constraints
  - Flaws restricted over founded expressions, resolvers filtered out by subsumption
- Resolvers for resource conflicts
  - Disjunction of temporal constraints (scheduling), atemporal constraints (allocation), and tasks (resource production)
  - Resolvers obtained through minimal critical sets

Goal decomposition and task evaluation



hold(samples(site3):collected, (t1, t2))

Collect-samples(site3, ?rbt)

and

hold(data(site3,earth):sent, (t'1, t'2))

hold(pos(?rbt):site3, (t'1, t'2))

or

Get-send-data(site3, earth, ?rbt2)

Goto(site3, ?rbt)

event(pos(robot1):(in-way, ?x), t8)

and

hold(pos(?rbt2):site3, (t'1, t'2))

hold(sunlight:day, (t'1, t'2))

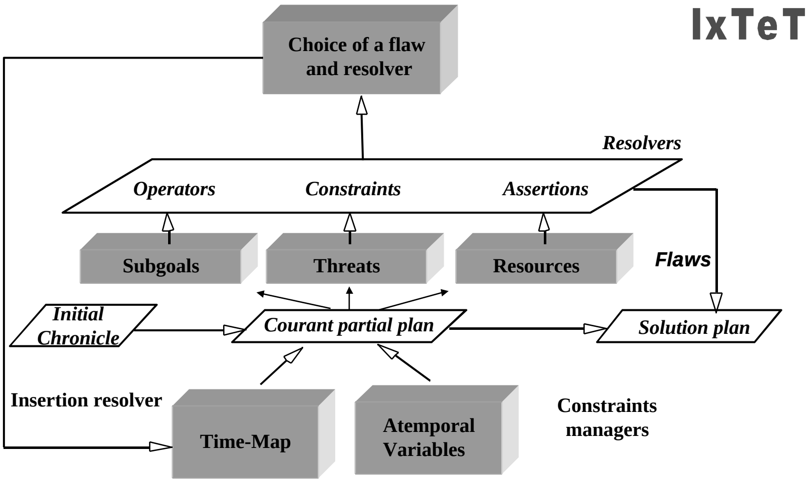
or

event(pos(?rbt2:(?y, in-way), t'8)

event(pos(robot1): (in-way, ?x), t8)

Goto(site3, ?rbt2)

IxTeT



Choice of a flaw and resolver

Resolvers

Operators ***Constraints*** ***Assertions***

Subgoals **Threats** **Resources**

Flaws

Initial Chronicle

Courant partial plan

Solution plan

Insertion resolver

Time-Map

Atemporal Variables

Constraints managers

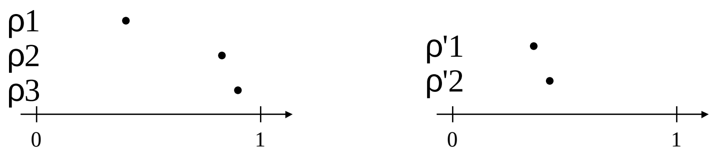
Search control

- Estimate of the commitment due to a resolver

  $\text{Commit}(P, \rho) = 1 - [|\text{completion}(P + \rho)| \,/\, |\text{completion}(P)|]$

- Choice of the next flaw to address: the one with the fewest competing resolvers

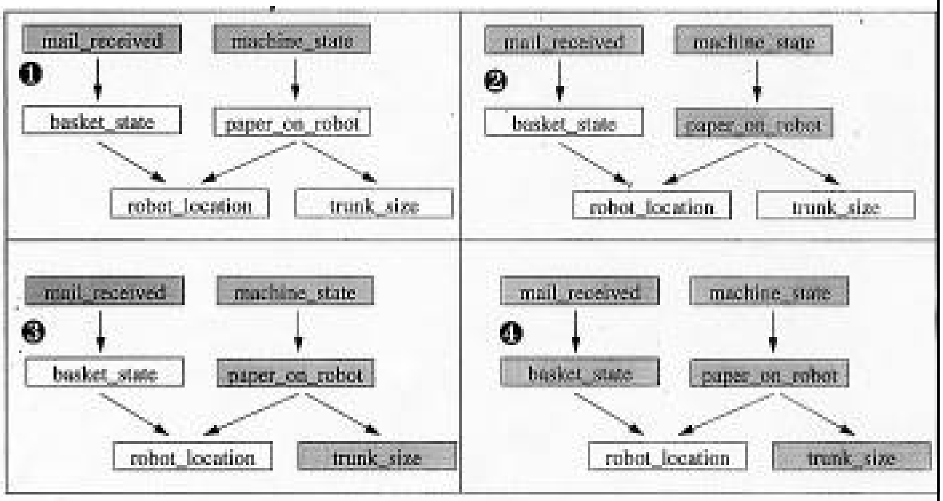
  $1/\text{Opp}(P, \varphi) = \sum_{\rho} [1/(1+ \text{Commit}(P, \rho) - \text{Commit}(P, \rho_{min})]$



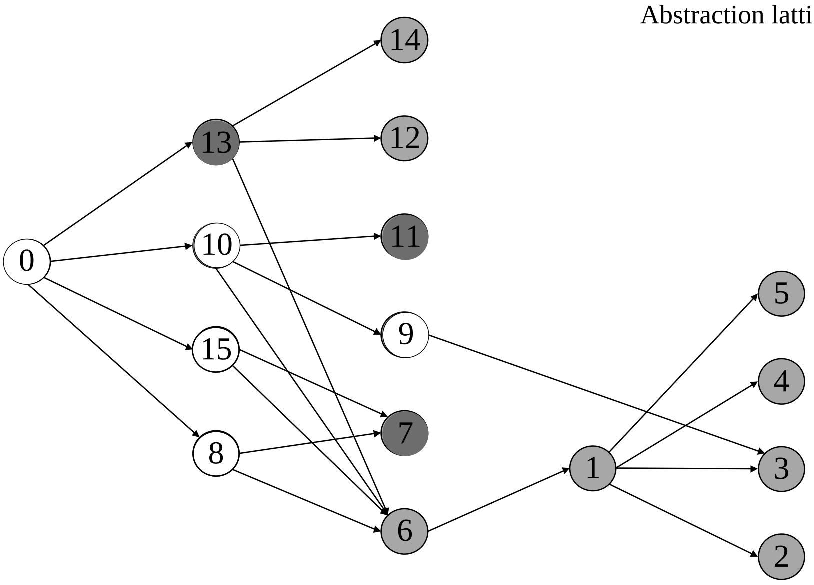
Domain hierarchy

Hierarchy in the search space

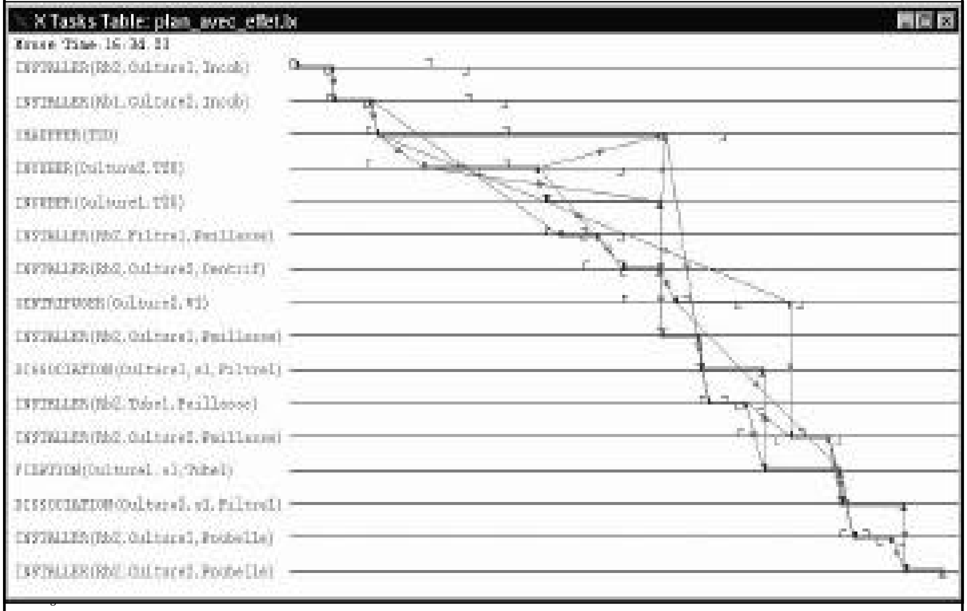
- Hierarchy node : a class of fluent types and corresponding flaws
- Partial order of nodes with the *ordered monotonicity property* obtained by preprocessing:
  solving a flaw on a fluent does lead to a new flaw on a preceding fluent
- Dynamic hierarchy: opportunistic *topological sort* of the partial order guides the search for a flawless plan

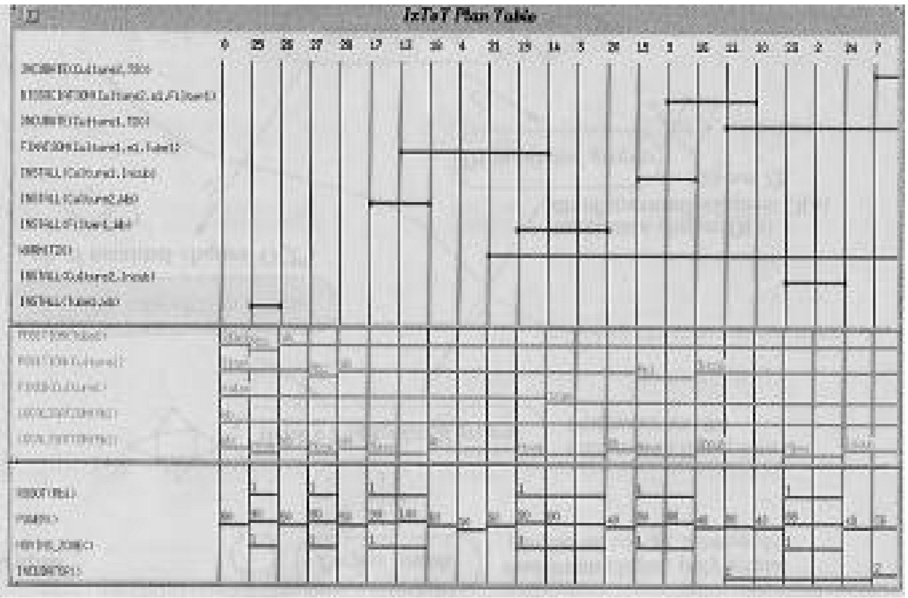


Abstraction lattice



Examples


X Tasks Table: plan_avec_effet.ix
Erase Time 16:34:33
INSTALLER(Rb2,Culture1,Incub)
INSTALLER(Rb1,Culture2,Incub)
CHAUFFER(T20)
INFUSER(Culture2,T20)
INFUSER(Culture1,T20)
INSTALLER(Rb2,Filtre1,Paillasse)
INSTALLER(Rb2,Culture2,Centrif)
CENTRIFUGER(Culture2,V1)
INSTALLER(Rb2,Culture1,Paillasse)
DISSOCIATION(Culture1,s1,Filtre1)
INSTALLER(Rb2,Tube1,Paillasse)
INSTALLER(Rb2,Culture2,Paillasse)
FILTRATION(Culture1,s1,Tube1)
DISSOCIATION(Culture2,s1,Filtre1)
INSTALLER(Rb2,Culture1,Poubelle)
INSTALLER(Rb2,Culture2,Poubelle)


IxTeT Plan Table
0 25 26 27 28 17 12 18 4 21 19 14 5 20 15 9 16 11 10 23 2 24 7
INCUBATE(Culture2,T20)
DISSOCIATION(Culture2,s1,Filtre1)
INCUBATE(Culture1,T20)
FILTRATION(Culture1,s1,Tube1)
INSTALL(Culture1,Incub)
INSTALL(Culture2,Ab)
INSTALL(Filtre1,Ab)
WARM(T20)
INSTALL(Culture2,Incub)
INSTALL(Tube1,Ab)

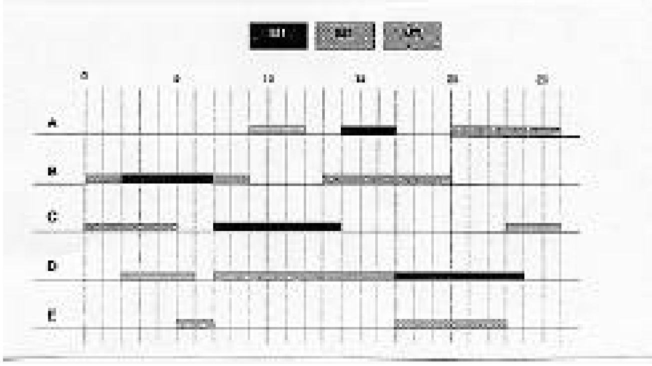
A : M2(durée=3), M1(durée=3), M3(durée=6) ou bien
M2(durée=3), M3(durée=6), M1(durée=3)

B : M2(durée=2), M1(durée=5), M2(durée=2), M3(durée=7) ou bien
M2(durée=2), M3(durée=7), M2(durée=2), M1(durée=5)

C : M1(durée=7), M3(durée=5), M2(durée=3) ou bien
M3(durée=5), M1(durée=7), M2(durée=3)

D : M2(durée=4), M3(durée=6), M1(durée=7), M2(durée=4) ou bien
M2(durée=4), M3(durée=6), M2(durée=4), M1(durée=7)

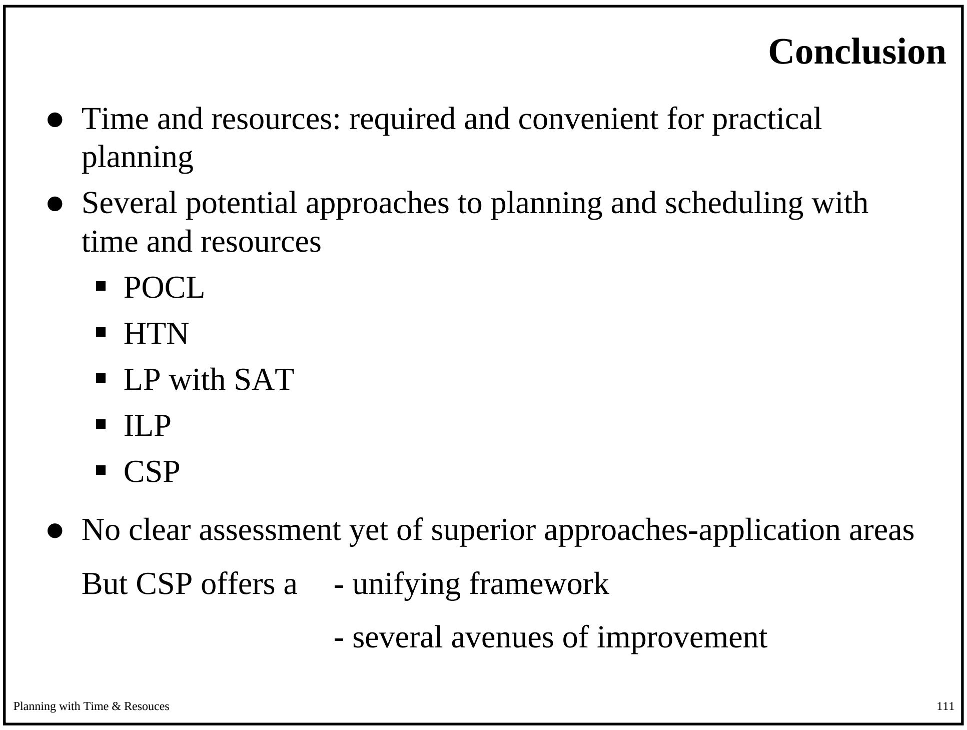
E : M2(durée=6), M3(durée=2) ou bien
M3(durée=2), M2(durée=6)



Planning & Scheduling with Time & Resources

Outline

- Motivations
- Representations of time, resources and actions
- Time management
- Resource management
- Planning & scheduling
- Conclusion



Conclusion

- Time and resources: required and convenient for practical planning
- Several potential approaches to planning and scheduling with time and resources
  - POCL
  - HTN
  - LP with SAT
  - ILP
  - CSP
- No clear assessment yet of superior approaches-application areas

  But CSP offers a - unifying framework

  - several avenues of improvement

Bibliography on
Planning and scheduling with time and resources

Temporal reasoning

Allen J. 1982, Maintaining Knowledge About Temporal Intervals. Comm ACM, 26(11) : 832-843

Allen J. 1984, Towards a general theory of action and time. Artificial Intelligence 23(2), 123–154.

Borchardt G. 1985, Event calculus, Proc. 9th IJCAI

Brusoni V., 1997, Temporal reasoning in data and knowledge bases. PhD thesis, Univ. Di Torino

Brusoni V., Console L. and Terenziani P. 1995, On the omputational complexity of querying bounds on differences constraints. Artificial Intelligence, 74(2):367-379

Dean T.L. 1984, Managing Time Maps,   Proc. of. Canadian Society for Computational Studies of Intelligence (CSCSI 84)

Dean T.L. 1985, Temporal Reasoning Involving Counterfactuals and Disjunctions,   Proc. 9th IJCAI 1985,   1060-1062.

Dechter, R, Meiri, I, and Pearl, J. 1991. Temporal constraint networks. Artificial Intelligence 49, 61-95.

Gerevini A. and Schubert L., 1995, Efficient algorithms for qualitative reasoning about time. Artificial Intelligence, 74(2):207-248

Gerevini A., Schubert L. and Schaeffer S. 1993, Temporal reasoning in timegraph. SIGART Bull 4(3):21-25.

Ghallab M. and Mounir Alaoui A. 1989, Managing Efficiently Temporal Relations through Indexed Spanning Trees. Proc. 11th IJCAI

Kowalski R. and Sergot M.1986, A logic based calculus of events, New Generation Computing 4: 67-95,

Ladkin P. and Reinfeld A. 1992, Effective solution of qualitave constraint problems. Artificial Intelligence, 57:105-124.

Ladkin P.1986, Time Representation: A Taxonomy of Interval Relations,   Proc. AAAI, 360-366.

Ligozat G. 1990, Weak representations of interval algebras. Proc. AAAI, 715-720

Ligozat G. 1991, On generalized interval calculi. Proc. AAAI, 234-240

M. Ghallab and T. Vidal. Focusing on a Sub-graph for Managing Efficiently Numerical Temporal Constraints. In Proceedings FLAIRS-95, 1995.

M. Ghallab and T. Vidal. 1995, Temporal constraint in planning: Free or not free ? In FLAIRS Workshop on Constraint Reasonning.

Malik J. and Binford T.O. 1983, Reasoning in Time and Space,   Proc. 8th IJCAI 1983, 343-345

Meiri I. 1990, Faster constraint satisfaction algorithms for temporal reasoning. Tech. Report R-151, UCLA Cognitive System Lab.

Nebel B., and Bürckert H.J. 1995, Reasoning about temporal relations : a maximal tractable subclass of Allen’s interval algebra, J. ACM, 42(1) : 43-66

Rit J. 1986 Propagating temporal constraints for scheduling, Proc. AAAI, 383-388.

Shoham Y. 1987, Temporal Logics in AI: Semantical and Ontological Considerations, Artificial Intelligence, 33, 89-104

Tsang E. 1987, Time Structure for AI, Proc 10th IJCAI, 456-461.

Vidal T., Ghallab M. 1996, Constraint-based temporal management in planning : the IxTeT approach. Proc. 12th European Conference on AI (ECAI'96)

Vilain M. 1982, A System for Reasoning About Time, Proc AAAI.

Vilain M. and Kautz H. 1986, Constraint Propagation Algorithms for Temporal Reasoning, Proc AAAI, 377-382

Vilain M. Kautz H. and van Beek P.G. 1989, Constraint Propagation Algorithms for Temporal Reasoning : a revised report. In Weld and de Kleer (eds.), Reading in qualitative reasoning about physical systems, Morgan Kaufmann

Scheduling

Beck, J, and Fox, M. 1998. A generic framework for constraint-directed search and scheduling. AI Magazine 19(4), 101–130.

Brucker, P. 1998. Scheduling Algorithms, second edition. Springer-Verlag.

Cesta A., Oddi A., Smith S.F., 1998, Profile-based algorithms to solve multiple capacitated metric scheduling problems. Proc. 4th AIPS

Cesta A., Oddi A., Smith S.F., 2000, Greedy algorithms for multi-capacitated metric scheduling problems. In Recent advances in AI Planning, Biundo and Fox (eds.), LNAI 1809, 213-225

Crawford, J, and Baker, A. 1994. Experimental results on the application of satisfiability algorithms to scheduling problems. Proc. 12th National Conf. on AI, 1092–1097.

Crawford, J. 1996. An approach to resource constrained project scheduling. Proc. 1996 AI and Manufacturing Research Planning Workshop. Albuquerque, NM, AAAI Press.

Dixon, H, and Ginsberg, M. 2000. Combining satisfiability techniques from AI and OR. The Knowledge Engineering Review 15(1).

Drummond, M, Bresina, J, and Swanson, K. 1994. Just-In-Case scheduling. Proc. 12th National Conf. on AI, 1098–1104.

Fox, M, 1994. ISIS: a retrospective. In Intelligent Scheduling, Zweben and Fox (eds.). Morgan Kaufmann., 3–28.

Liatsos V. and Richards B., 2000, Scaleability in planning. In Recent advances in AI Planning, Biundo and Fox (eds.), LNAI 1809, 49-61

Minton, S, Johnston, M, Phillips, A, and Laird S. 1992. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. Artificial Intelligence 58, 161–205.

Pinedo, M. 1995. Scheduling Theory, Algorithms and Systems. Prentice Hall.

Pinson, E. 1995. The job shop scheduling problem: a concise survey and some recent developments. In P. Chrétienne, E. Coffman, J. Lenstra and Z. Liu (eds) Scheduling Theory and its Applications. Wiley.

Plaunt, C, Jónsson, A, and Frank, J. 1999. Satellite tele-communications scheduling as dynamic constraint satisfaction. Proc. 5th Int. Symposium on AI, Robotics and Automation in Space, 277–284.

Smith, S. and Cheng, C. 1993. Slack-based heuristics for constraint satisfaction scheduling. Proc. 11th Nat. Conf. on AI, 139–144.

Wah, B, and Shang, Y. 1997. Discrete Lagrangian based search for solving max-SAT problems, Proc. 15th Int. Joint Conf. on AI, 378–383.

Zweben, M, and Fox, M. 1994. Intelligent Scheduling. Morgan Kaufmann.

Planning with time and resources

Albers P., Ghallab M. 1997, Context dependent effects in temporal planning. In Recent advances on AI planning, Steel and Alami (Eds.), LNCS 1348, Springer, 1-12

Allen, J, and Koomen, J. 1983. Planning using a temporal world model. Proc. 8th Int. Joint Conf. on AI, 741–747.

Allen, J, Hendler, J, and Tate, A. 1990. Readings in Planning. Morgan Kaufmann.

Currie K. and Tate A., 1991, O-Plan, the open planning architecture. Artificial Intelligence, 52(1): 49 - 86

Dean, T, and Kambhampati, S. 1996. Planning and scheduling. In CRC Handbook of Computer Science and Engineering.

Drabble, B, and Tate, A. 1994. The use of optimistic and pessimistic resource profiles to inform search in an activity based planner. Proc. 2nd Int. Conf. on AI Planning Systems, 243–248

Drabble, B. 1993. Excalibur: a program for planning and reasoning with processes. Artificial Intelligence 62, 1– 40.

Eshghi K. 1988, Abductive planning with the event calculus. Proc. 5th Int. Conf. On Logic Programming, 562-579

Ferguson, G, Allen, J, and Miller, B. 1996. TRAINS-95: towards a mixed-initiative planning assistant. Proc. 3rd Int. Conf. on AI Planning Systems, 70–77.

Ghallab, M, and Laruelle, H. 1994. Representation and control in IxTeT, a temporal planner. Proc. 2nd Int. Conf. on AI Planning Systems, 61–67.

Gout J., Ghallab M. 1998, A task planner for the computer-aided design of a space-lander robot. Proc. IEEE International Conference on Robotics and Automation (ICRA'98), 605-610

Jónsson, A, Morris, P, Muscettola, N, Rajan, K, and Smith, B. 2000. Planning in interplanetary space: theory and practice. Proc. 5nd Int. Conf. on AI Planning & Scheduling.

Kautz, H, and Walser, J. 1999. State-space planning by integer optimization. The Knowledge Engineering Review 15(1).

Koehler, J. 1998. Planning under resource constraints. Proc. 13th European Conf. on AI . 489–493

Laborie P., Ghallab M. 1995, IxTeT : an integrated approach for plan generation and scheduling. 4th IEEE Symposium on Emerging Technologies and Factory Automation (ETFA'95), 485-495

Laborie, P, and Ghallab, M. 1995. Planning with sharable resource constraints. Proc. 14th Int. Joint Conf. on AI, 1643–1649.

Lingard A.R. and Richard E.B., 1998, Planning parallel actions, Artificial Intelligence, 99 (2): 261-324

Missiaen L. 1991, Localized abductive planning with the event calculus. PhD thesis, Univ. Of Leuven

Muscettola, N, Nayak, P, Pell, B, and Williams, B. 1998, Remote Agent: to boldly go where no AI system has gone before, Artificial Intelligence 103, 5–48.

Muscettola, N. 1994. HSTS: integrating planning and scheduling. In Intelligent scheduling, Zwenben and Fox (eds.), Morgan Kaufmann, 169–212.

Nau, D, Gupta, S, and Regli, W. 1995. Artificial Intelligence planning versus manufacturing-operation planning: a case study. Proc. 14th Int. Joint Conf. on AI, 1670–1676.

Penberthy, J. and Weld, D. 1994. Temporal planning with continuous change. Proc. 12th National Conf. on AI, 1010–1015.

Smith D.E., Frank J. and Jonsson A.K., 2000, Bridging the gap between planning and scheduling. Knowledge Engineering Review, 15(1).

Smith, D, and Weld, D. 1999. Temporal planning with mutual exclusion reasoning. Proc. 16th Int. Joint Conf. on AI, 326–333.

Tate, A, Drabble, B, and Kirby, R. 1994. O-Plan2: an open architecture for command, planning, and control. 213–239.

Tsang E.P.K. 1986, Plan generation in a temporal frame. Proc. ECAI, 643-657

Tsang E.P.K. 1987, TLP – a temporal planner. In Advances in AI, Hallam and Mellish (eds.), Springer, 63-78

Vere, S. 1983. Planning in time: windows and durations for activities and goals, Pattern Analysis and Machine Intelligence 5(3), 246–267.

Vossen, T, Ball, M, Lotem, A, and Nau, D. 1999. On the use of integer programming models in AI planning. The Knowledge Engineering Review 15(1).

Wilkins D.E. 1988, Practical Planning: Extending the Classical AI Planning Paradigm. Morgan-Kauffmann.

Wolfman, S, and Weld, D. 1999. Combining linear programming and satisfiability solving for resource planning. The Knowledge Engineering Review 15(1).