Un cadre général intégrant les approches pour ordonnancer sous incertitudes

A General Framework Integrating Techniques for Scheduling under Uncertainty

Julien Bidot[1,2]

[1]ILOG S. A., France

[2]Laboratoire Génie de Production
Ecole Nationale d'Ingénieurs de Tarbes, France

28 novembre 2005

Cadre de la thèse

- Planification et ordonnancement classiques = problèmes difficiles (NP-complets) avec environnement d'exécution déterministe
- En pratique : environnement d'exécution non déterministe !
- Pour faire face à l'incertitude :
  - produire une solution protégée contre les perturbations
  - produire une solution sur un horizon partiel glissant
  - modifier la solution courante pendant l'exécution si nécessaire
- Nos objectifs :
  - clairement distinguer les techniques de résolution et les systèmes utilisés pour ordonnancer et planifier sous incertitudes
  - développer un prototype informatique intégrant ces différentes techniques de résolution, en utilisant les composants ILOG Solver et Scheduler

Plan

Basics in Task Planning and Scheduling

Classification of Techniques and Systems under Uncertainty

Generation and Execution Model

Software Prototype

Conclusions et perspectives

Plan

Task Planning and Scheduling

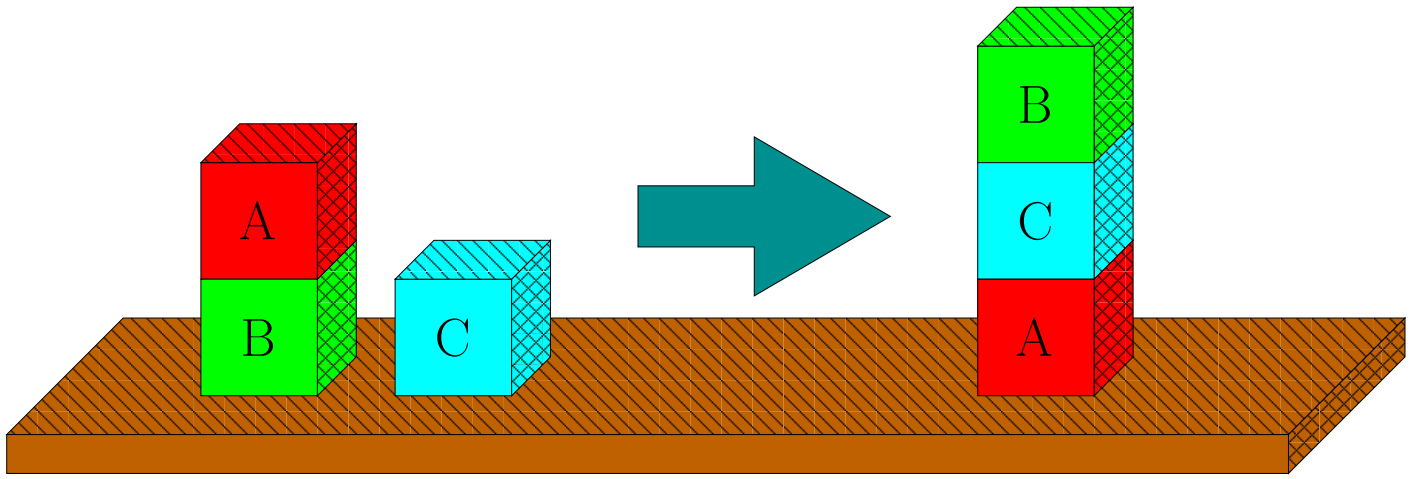
- task planning: *select* and *organize* actions in *time* with respect to time requirements and resources to reach the goal state(s) from the initial state
- scheduling: *set* operations in *time* with respect to time requirements and resources

Example in Task Planning

Blocks world

Operator:

- `Move`(?X, ?Z, ?Y):
  - Preconditions: `Free`(?Y), `Free`(?X), `On`(?X, ?Z)
  - Effects: `On`(?X, ?Y), ¬ `Free`(?Y), `Free`(?Z)

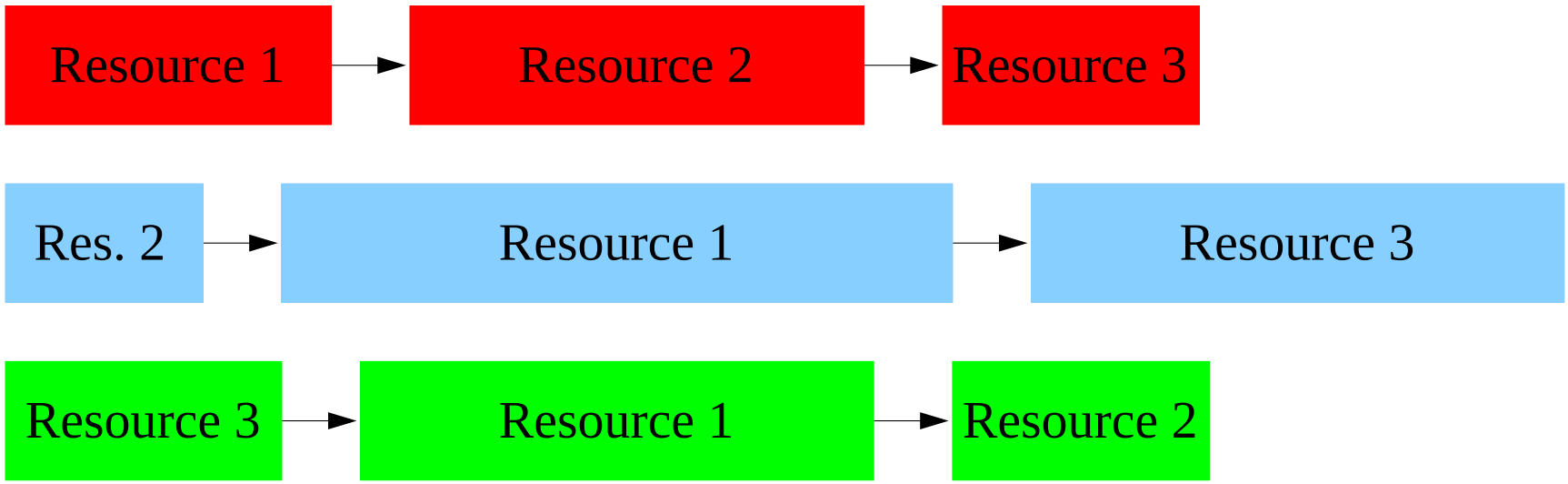


Free(A)
Free(C)
¬Free(B)
On(A, B)

Free(B)
¬Free(C)
¬Free(A)
On(B, C)
On(C, A)

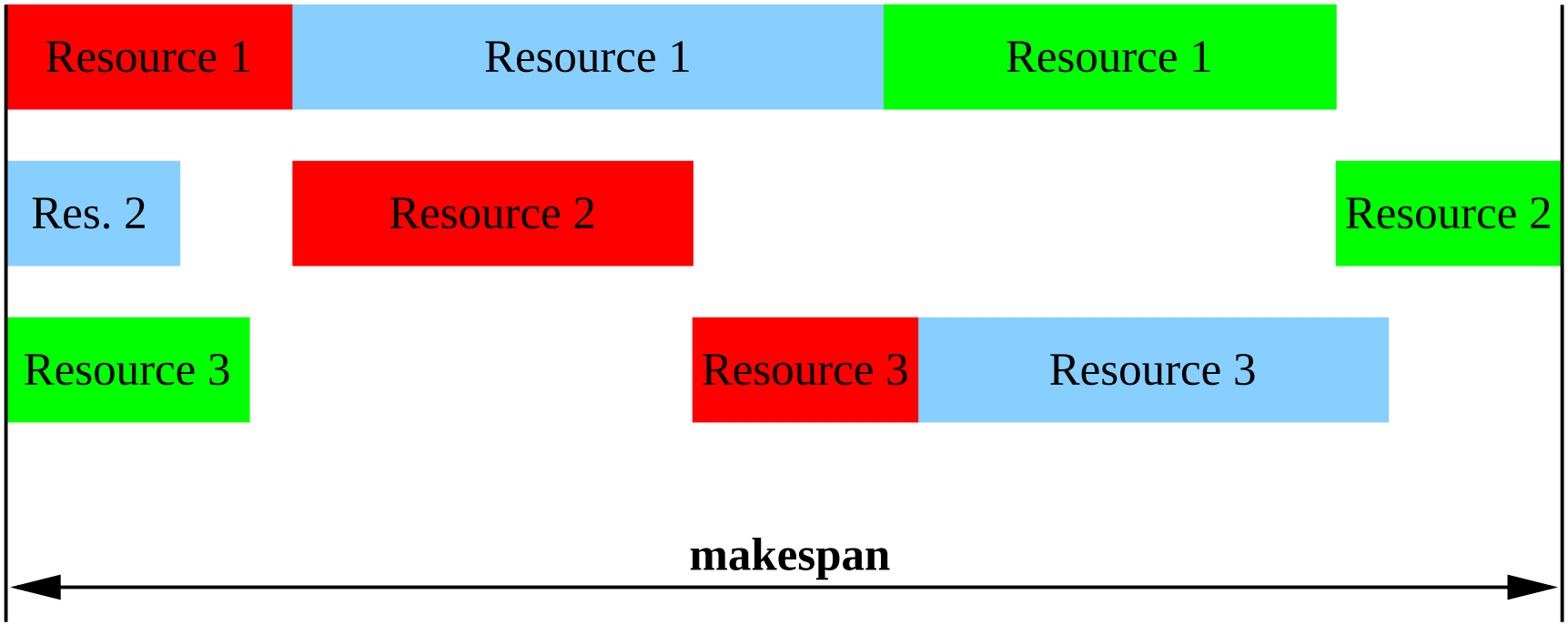
Example in Scheduling

Job-Shop Scheduling Problem (JSSP) with makespan minimization



Example in Scheduling (end)

Optimal solution of the preceding JSSP



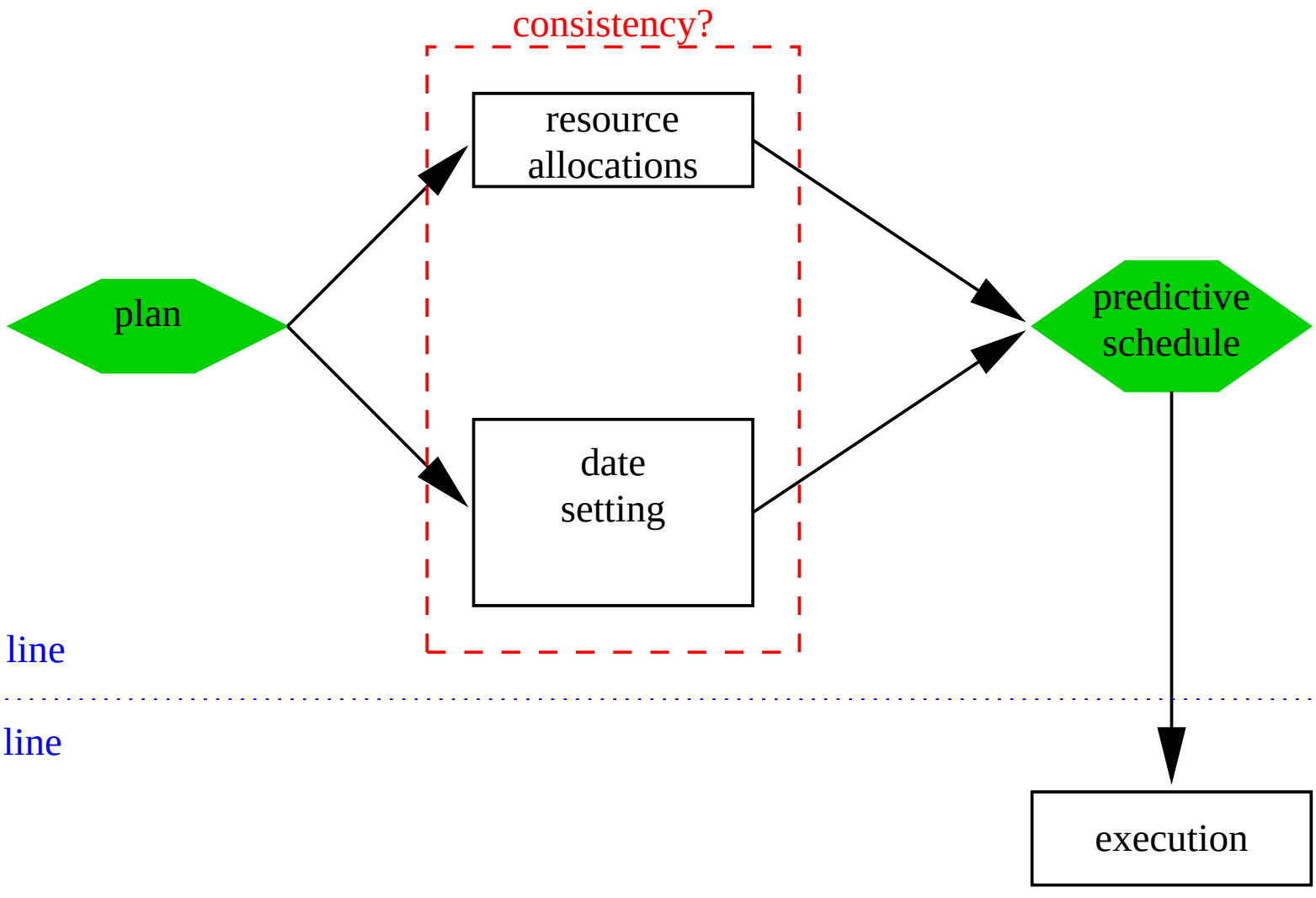
Constraint Satisfaction Approach for Scheduling

Example

- start, processing, and end times of operations = variables
- optimization criterion = expression with variables
- some constraints posted; for example, operation o_i executed after the end time of operation o_j: $start_i \geq end_j$; operation o_k and operation o_l require the same unary resource: resource constraints posted, that is, sequence o_k and o_l: $start_k \geq end_l \vee start_l \geq end_k$ (decision made during search)

[Nuijten, 1994]

Scheduling without Uncertainty



Uncertainty Sources in Task Planning

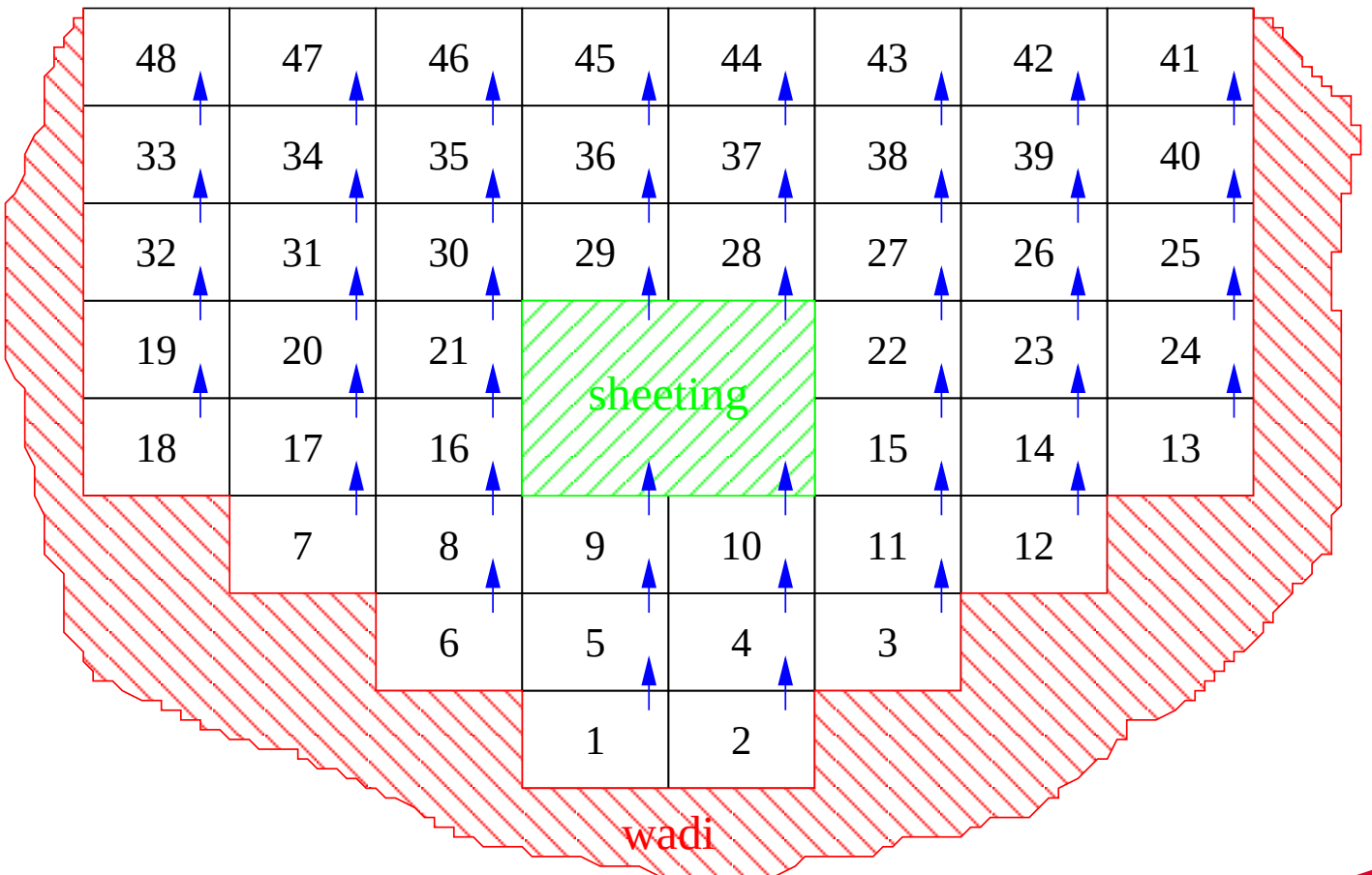
- imprecise action effects
- partially observable or non-observable world states
- uncertain world states independently of actions $\longrightarrow$ observations
- new goal states during execution

Uncertainty Sources in Scheduling

Our application domain: project management

Dam construction: a set of tasks to perform with time and resource requirements

Dam wall in concrete (view from the front side):



Uncertainty Sources in Scheduling (end)

When constructing a dam:

- imprecise task durations
- imprecise quantity of required workers
- uncertain resources (breakdowns, absenteeism)
- observation of worse or better geological conditions on the building site than predicted
- late or early deliveries of subcontractors
- water flooding occurrences (statistics)

Costs:

- buying and renting machines and material, salaries
- tardiness costs

Objective: try to minimize the probability that the total cost greater than the initial negotiated budget

Definitions

Properties of a solution:

- stable schedule = schedule generated off line (before execution) and executed without modifications
- robust schedule = whatever happens on line (during execution), guaranty that solution quality does not derive

Characteristic of a solution: flexible solution = a subset of decisions done off line, the rest being done on line (different types of flexibility)
[Billaut et al., 2005]

Plan

Proactive Approach

Off-line reasoning using knowledge about uncertainty (stability)
More or less rigid solution (flexibility)
One or more solutions generated
controllability [Morris et al., 2001]

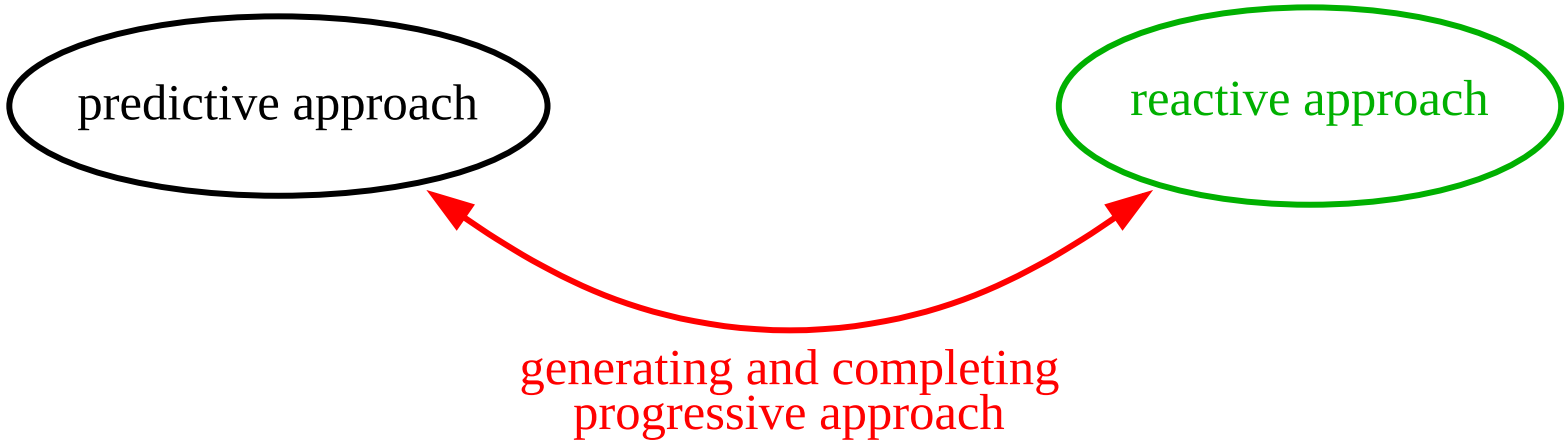
Revision Approach

On-line reasoning that changes a subset of decisions (instability) =
predictive-reactive approach $\longrightarrow$ replanning or rescheduling
Micro Boss [Sadeh et al., 1993], OPIS [Smith, 1994]

Progressive Approach

A subset of decisions made on a short-time, gliding horizon: decisions not changed (stability), production plan
[McKay et al., 1988], MARTHA project [Vidal et al., 1996]

- very short horizon = reactive approach (priority rules, dispatching rules)
- full horizon = predictive approach



Examples of Mixed Approaches

- proactive-revision techniques: *β-Robustness* [Daniels and Carrillo, 1997] and fuzzy scheduling [Dubois et al., 1993]; task planning of robots [Washington et al., 2000]
- proactive-reactive approaches: slack-based techniques [Davenport et al., 2001], partitioning and sequencing of operations off line [Wu et al., 1999]
- progressive-revision approaches: continuous task planning (space exploration) [Chien et al., 2000] and [Estlin et al., 2000]

Comparisons of Techniques

	On-line memory need	On-line CPU need	Quality optimization criterion	Stability
Revision	Average	High	Very high	Low
Proactive continuous	Low	Low	High	High
Proactive discrete	Very high	Very low	High	Average
Progressive	Very low	Average	Average	Very High

Plan

Unification of Scheduling Techniques under Uncertainty

Proposal of a theoretical model for generating and executing schedules: dynamic by nature (representation changes over time)
Model = automaton defined as follows:

- states = execution contexts = schedules more or less flexible (mutually exclusive sets of operations), each represented by a constraint network and associated with an execution algorithm (precise temporal setting or selection of an alternative)
- generation transitions to generate and change contexts when meeting some conditions in the current context; each generation (revision or progression) condition associated with a generation algorithm

[Vidal et al., 2003]

Unification of Scheduling Techniques under Uncertainty (cont'd)

More details:

- each variable either controllable (instantiated by an execution or generation algorithm) or uncontrollable (instantiated by Nature)
- each generation transition associated with a (revision or progression) condition involving variables

Expressivity of our Model

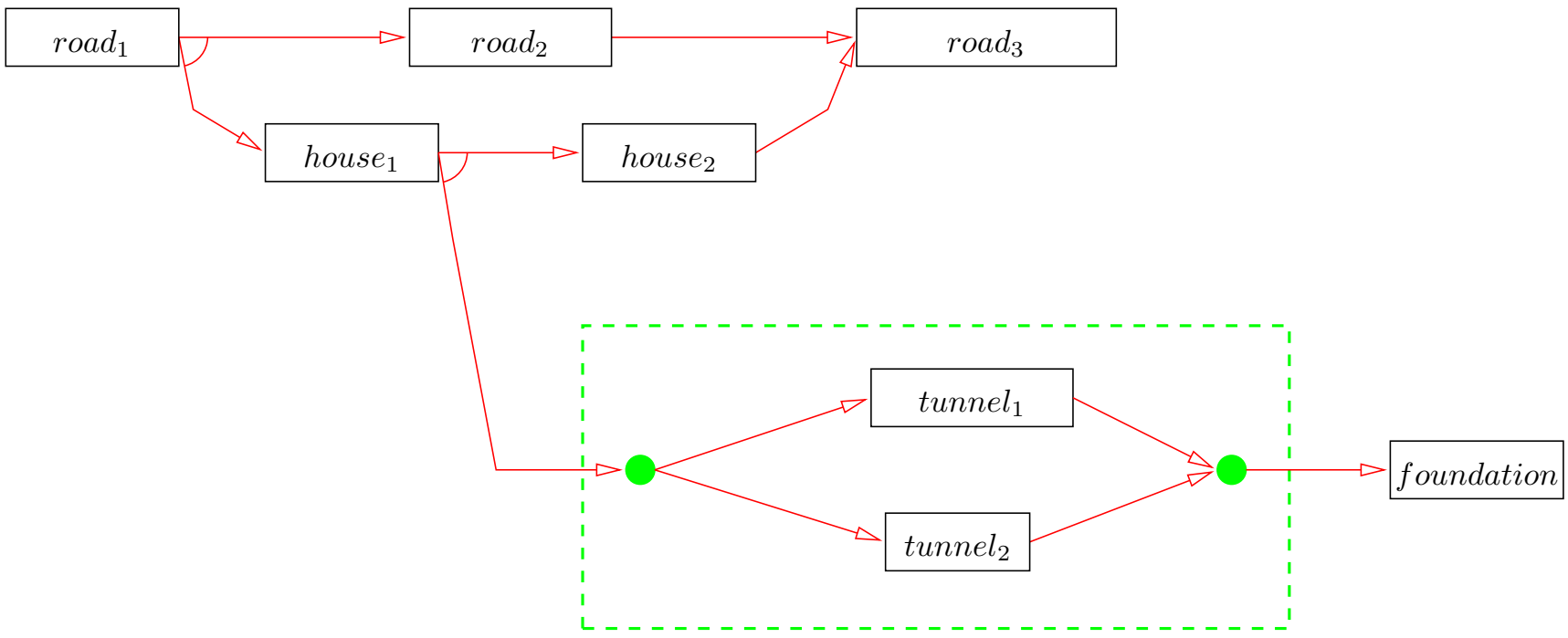
Toy example of a dam project:

- 2 trucks
- operations with imprecise processing times: construction of roads ($road_1$, $road_2$, and $road_3$), houses ($house_1$ and $house_2$), a tunnel ($tunnel_1$ and $tunnel_2$), foundations ($foundation$)
- precedence constraints between operations
- resource constraints: each operation requires one of two trucks
- 2 alternative ways of digging the tunnel depending on geological conditions observed at the end of $road_1$
- a due date and a tardiness-cost function

Objective: minimize expected tardiness costs

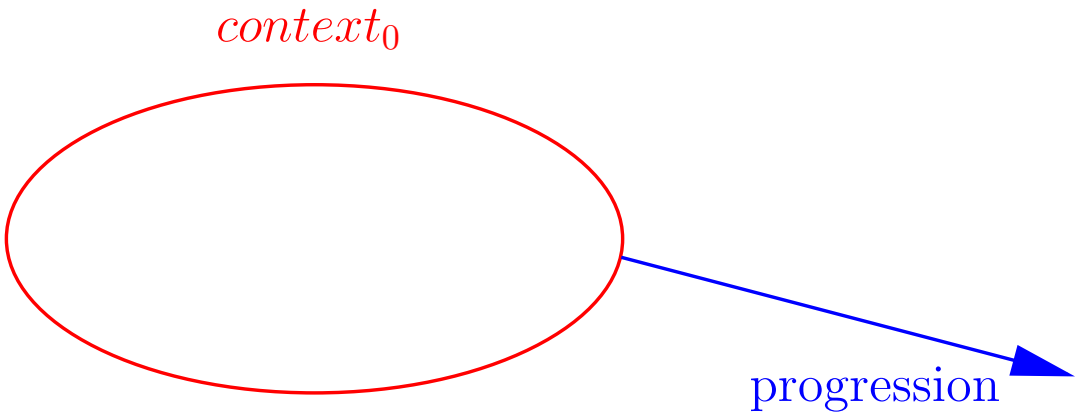
Expressivity of our Model (cont'd)

Representation of the preceding problem as an AND/OR precedence graph:



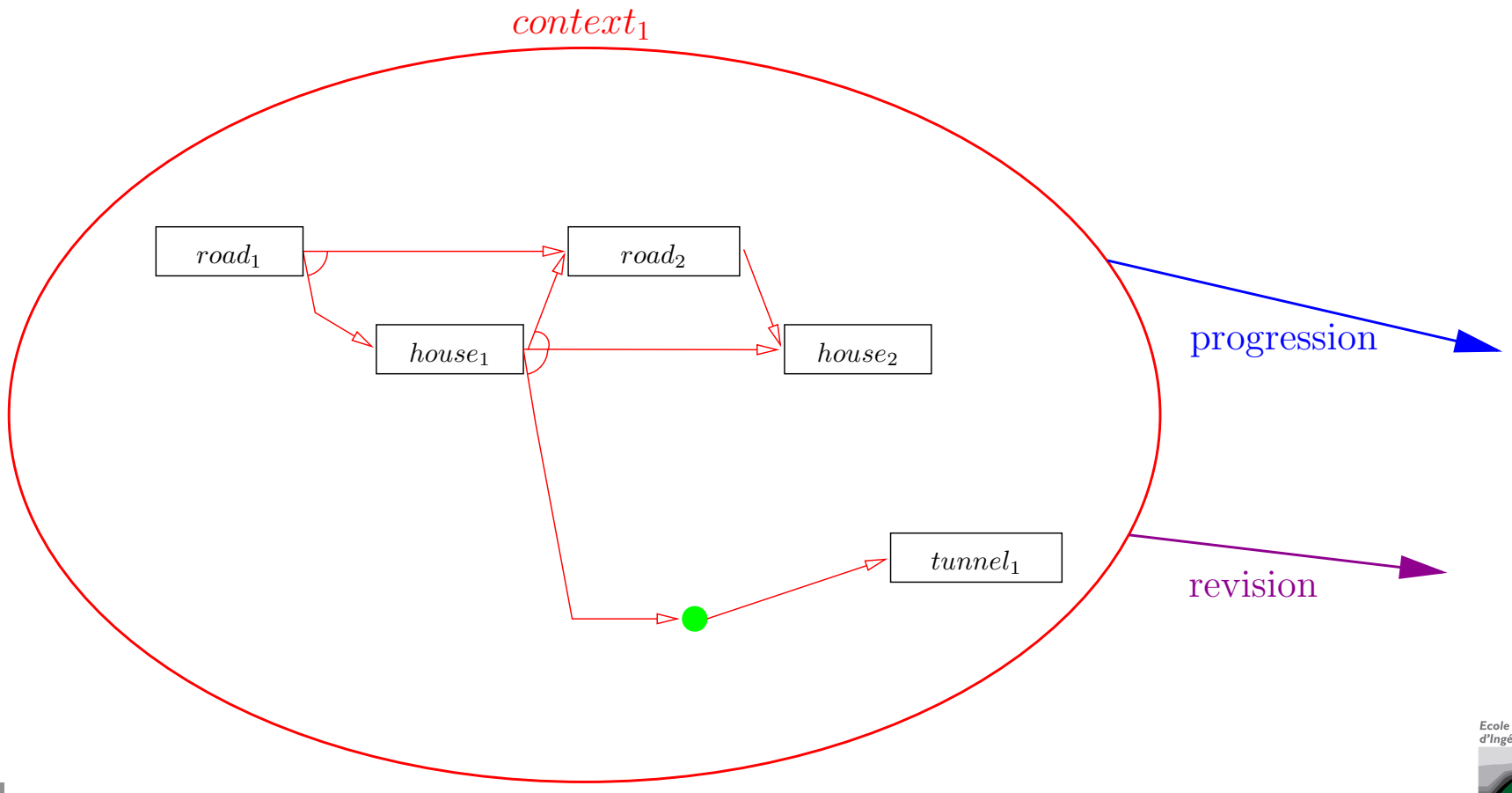
Expressivity of our Model (cont'd)

Model before starting execution: 2 template transitions (revision and progression) and 1 empty context ($context_0$)



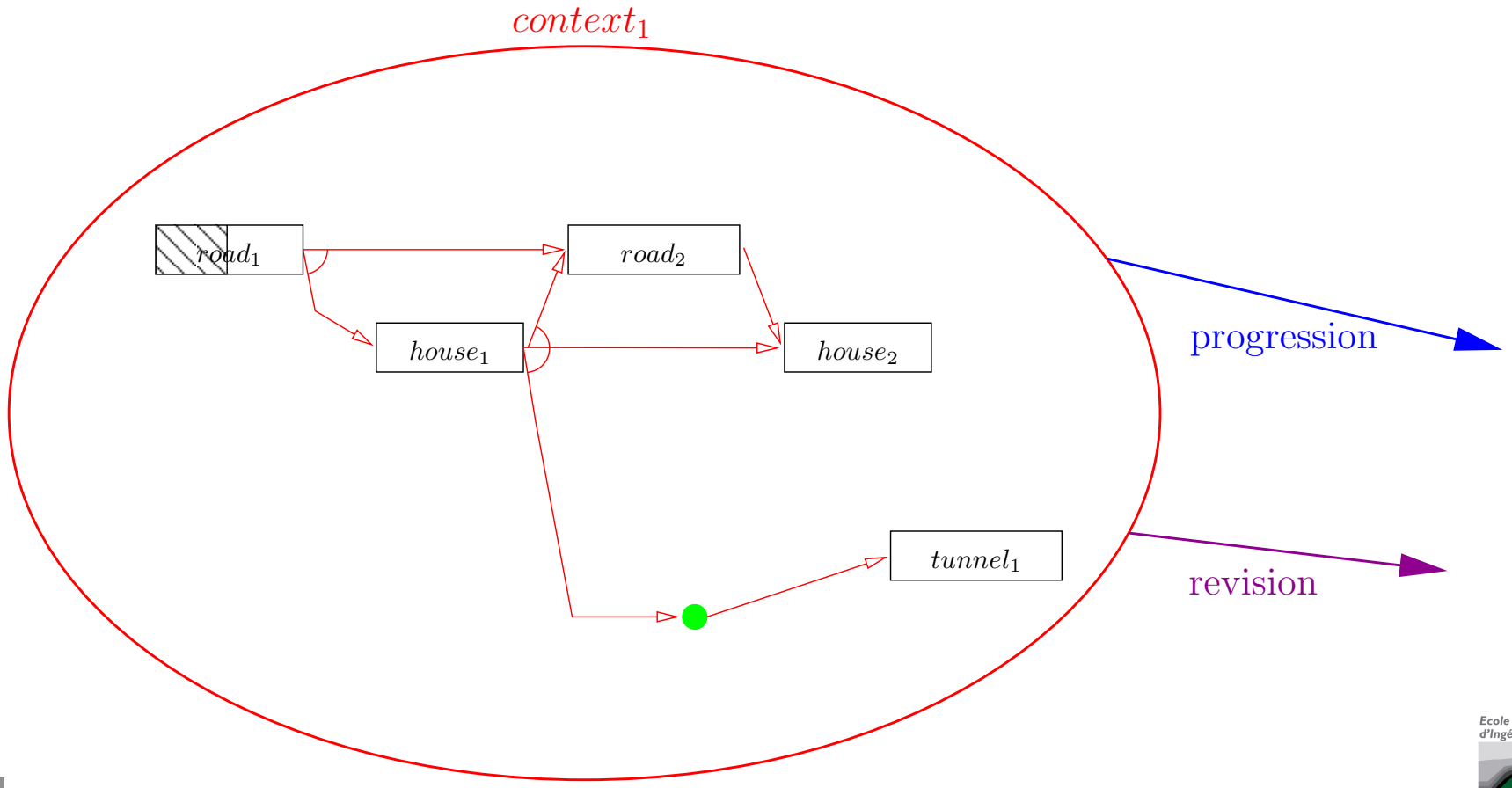
Expressivity of our Model (cont'd)

Model before starting execution: the progression condition met ($context_0$ is empty) $\rightarrow$ a new context generated ($context_1$)



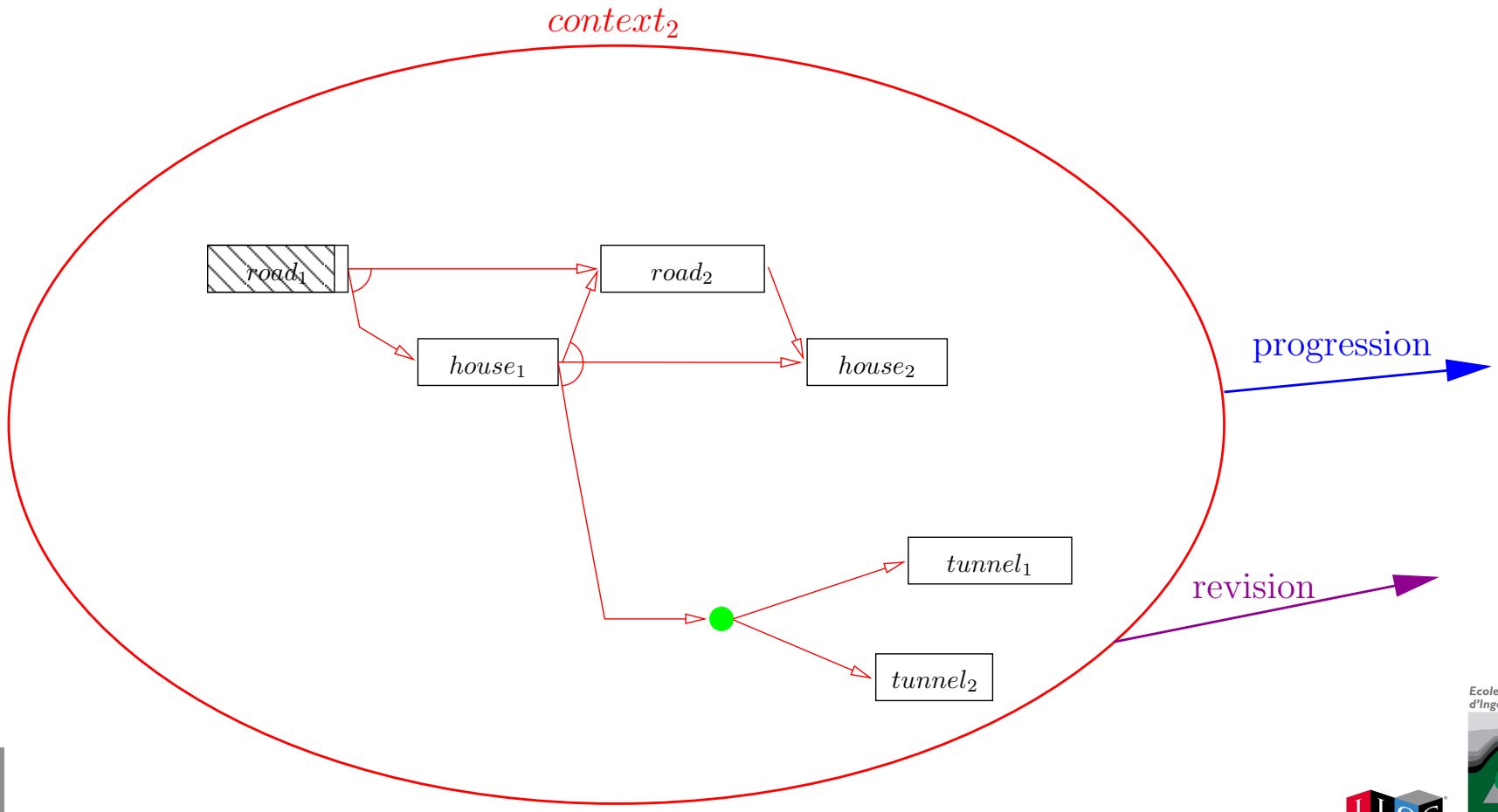
Expressivity of our Model (cont'd)

Model during execution
of $road_1$:



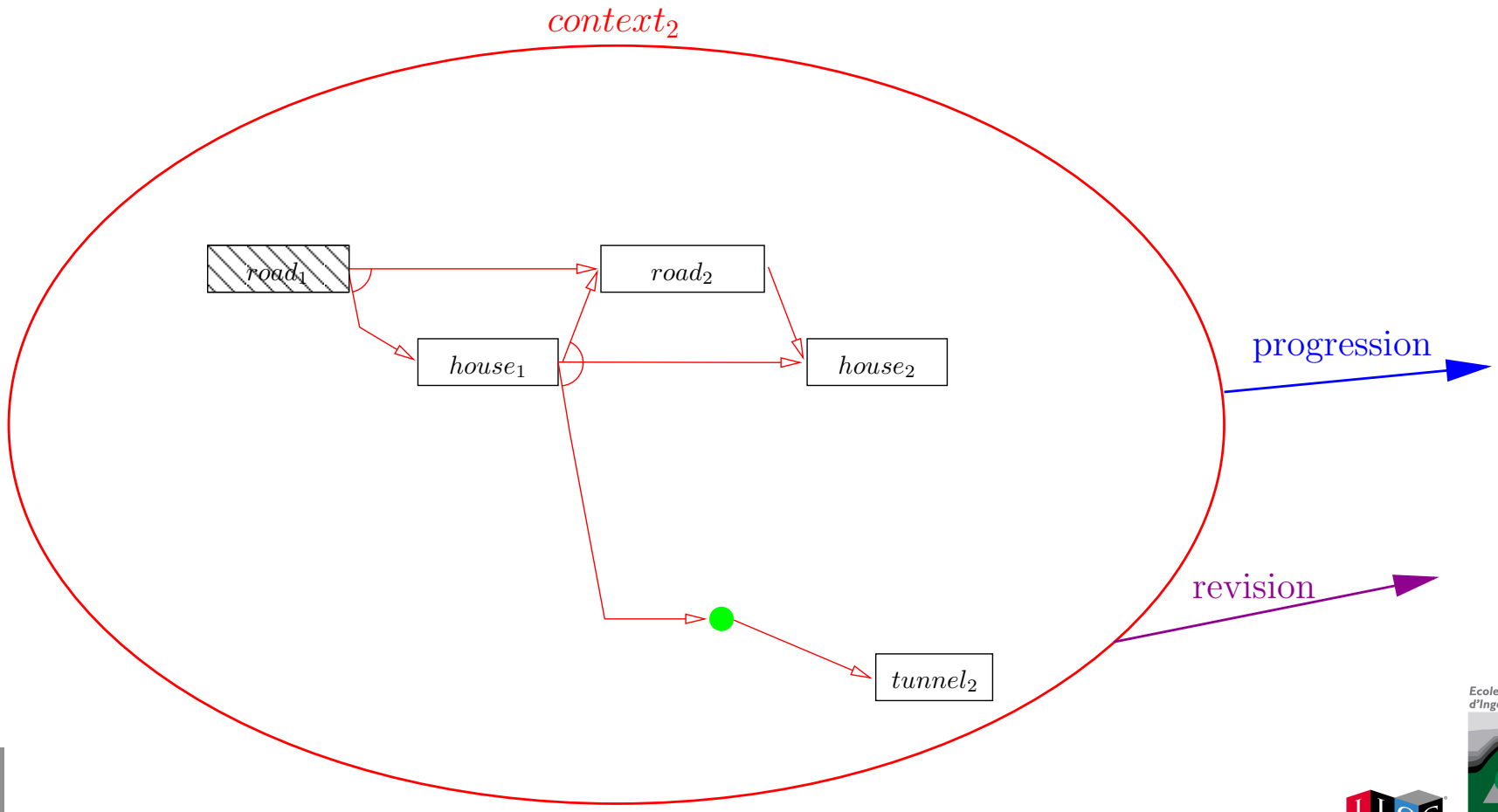
Expressivity of our Model (cont'd)

Model during execution of $road_1$:
Probability of $tunnel_2$ $\uparrow$, probability of $tunnel_1$ $\downarrow$: progression transition activated and a new context generated ($context_2$)



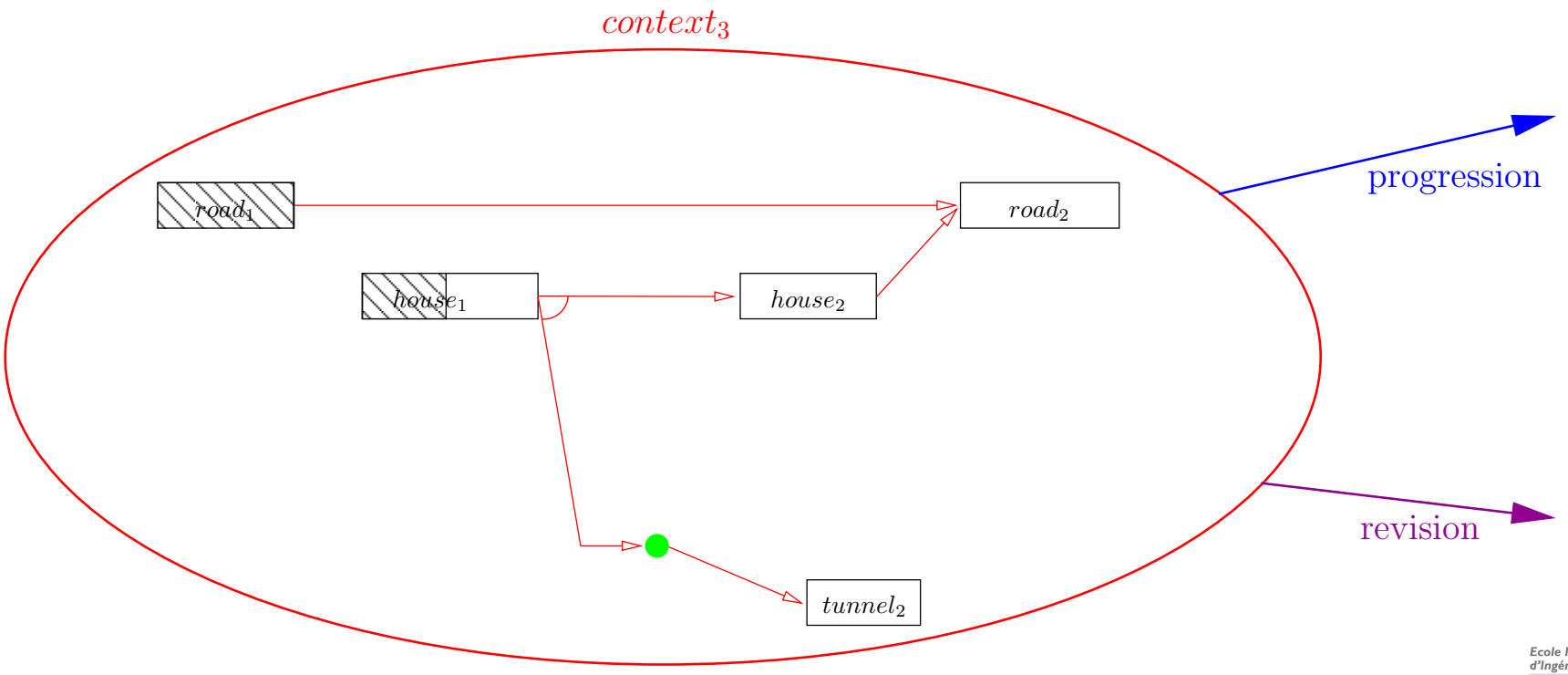
Expressivity of our Model (cont'd)

Model when $road_1$ finished:
$tunnel_2$ chosen based on geological conditions for the tunnel



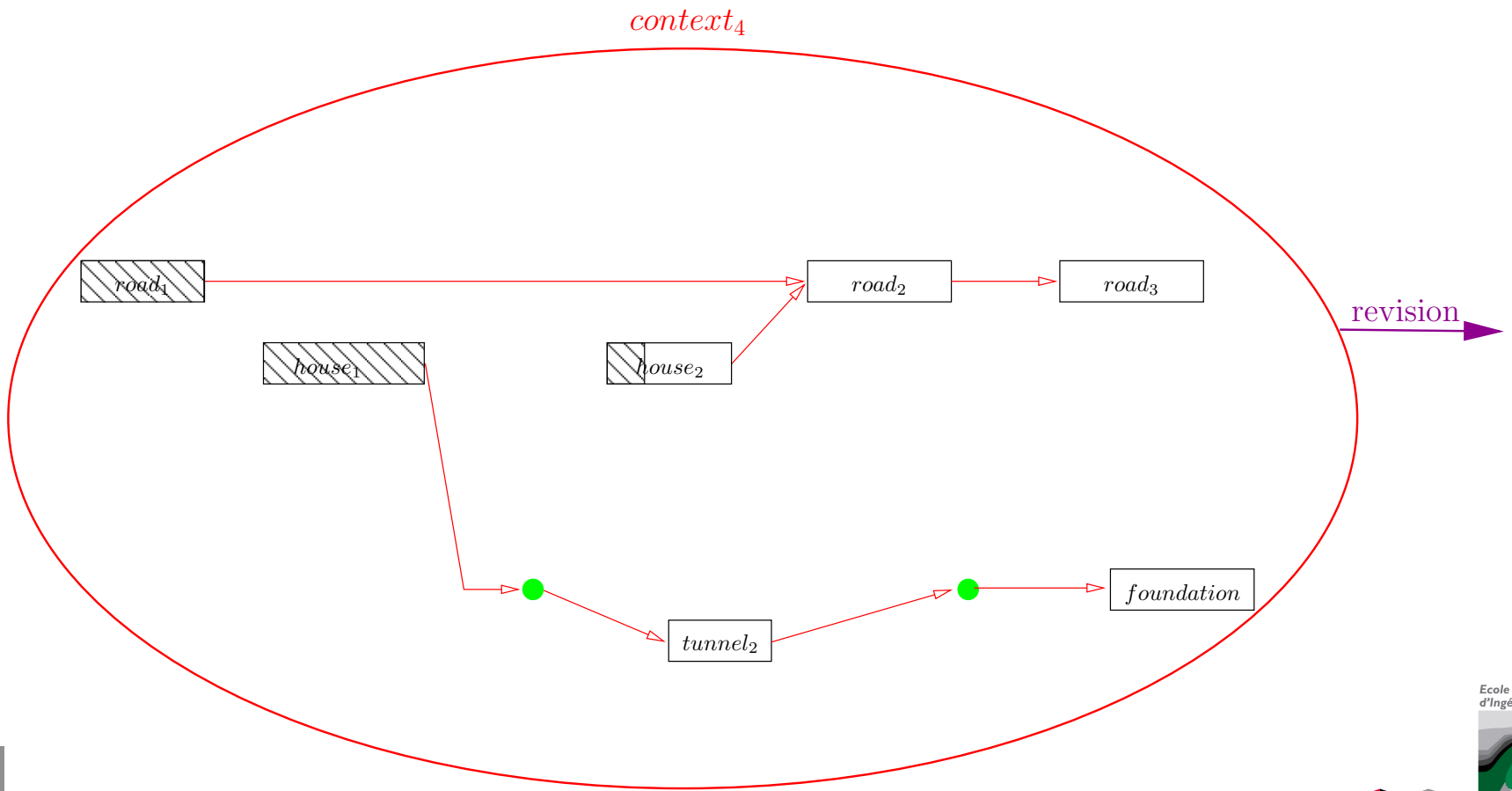
Expressivity of our Model (cont'd)

Model when $house_1$ executing: expected tardiness costs increase too much ($house_2$) $\longrightarrow$ revision transition activated and a new context generated ($context_3$)



Expressivity of our Model (end)

Model when $house_2$ executing: select new operations (anticipation) $\longrightarrow$ progression transition activated and a new context generated ($context_4$)

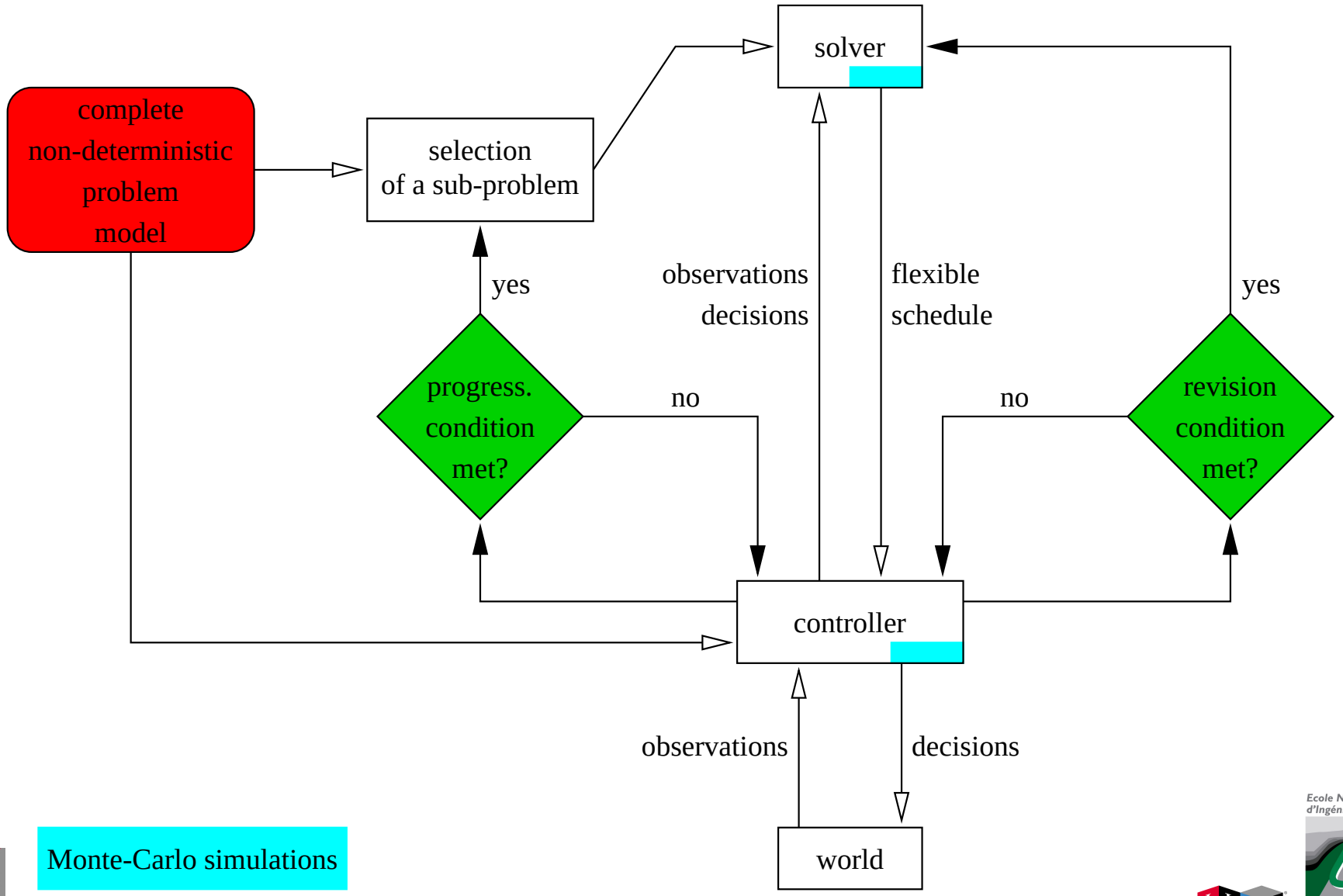


Plan

Experimental Study

- scheduling problems with probabilistic data
- one optimization criterion
- decision-making using:
  - constraint solvers: ILOG Solver and Scheduler
  - heuristics and metaheuristics
  - Monte-Carlo simulation: estimating the schedule quality during search + on-line maintenance of distributions of operation start and end times
  - execution algorithm = start operations as soon as possible (makespan or tardiness minimization)

Architecture



Monte-Carlo simulations

First Version of our Experimental System

Revision approach

- JSSP with minimization of expected makespan
- an indicative predictive schedule generated off line (tree search with mean operation durations and branch-and-bound procedure $\longrightarrow$ proaction)
- rescheduling of still non executed operations when revision criterion verified $\longrightarrow$ generation of a new indicative predictive schedule
- to indirectly set the number of reschedulings: sensibility
- several revision criteria investigated alternatively
- experiments with 100-operation problem instances
- impact of uncertainty level on quality

[Bidot et al., 2003]

First Version of our Experimental System (cont'd)

Revision criteria:

- $\mathrm{revCrit}_1$:

$$M_{\mathrm{exp}} > \frac{M_{\mathrm{ind}}}{\varsigma},$$

- $\mathrm{revCrit}_2$:

$$|M_{\mathrm{exp}} - M_{\mathrm{ind}}| > \frac{D}{\varsigma},$$

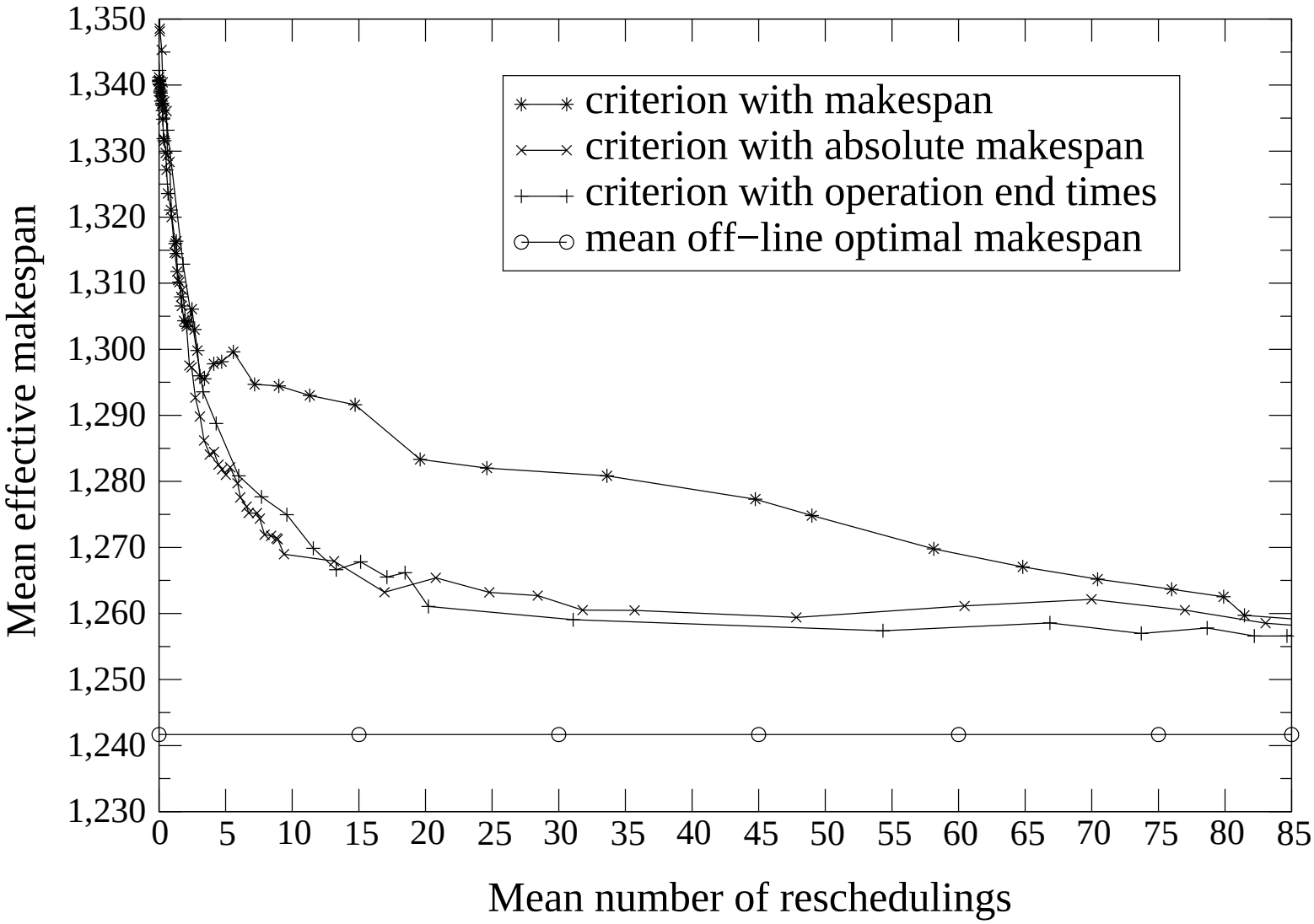
- $\mathrm{revCrit}_3$:

$$\frac{\sum_{O_{\mathrm{new}}} |end_{\mathrm{exp}} - end_{\mathrm{ind}}|}{nbNewOp} > \frac{D}{\varsigma},$$

M: makespan
D: mean of the mean operation durations
end: operation end time
ς: sensitivity
O_{new}: set of $nbNewOp$ operations that were executing the last time we rescheduled

First Version of our Experimental System (end)

Experiments (la11):



Second Version of our Experimental System

Mixed approach: revision, proaction, and progression

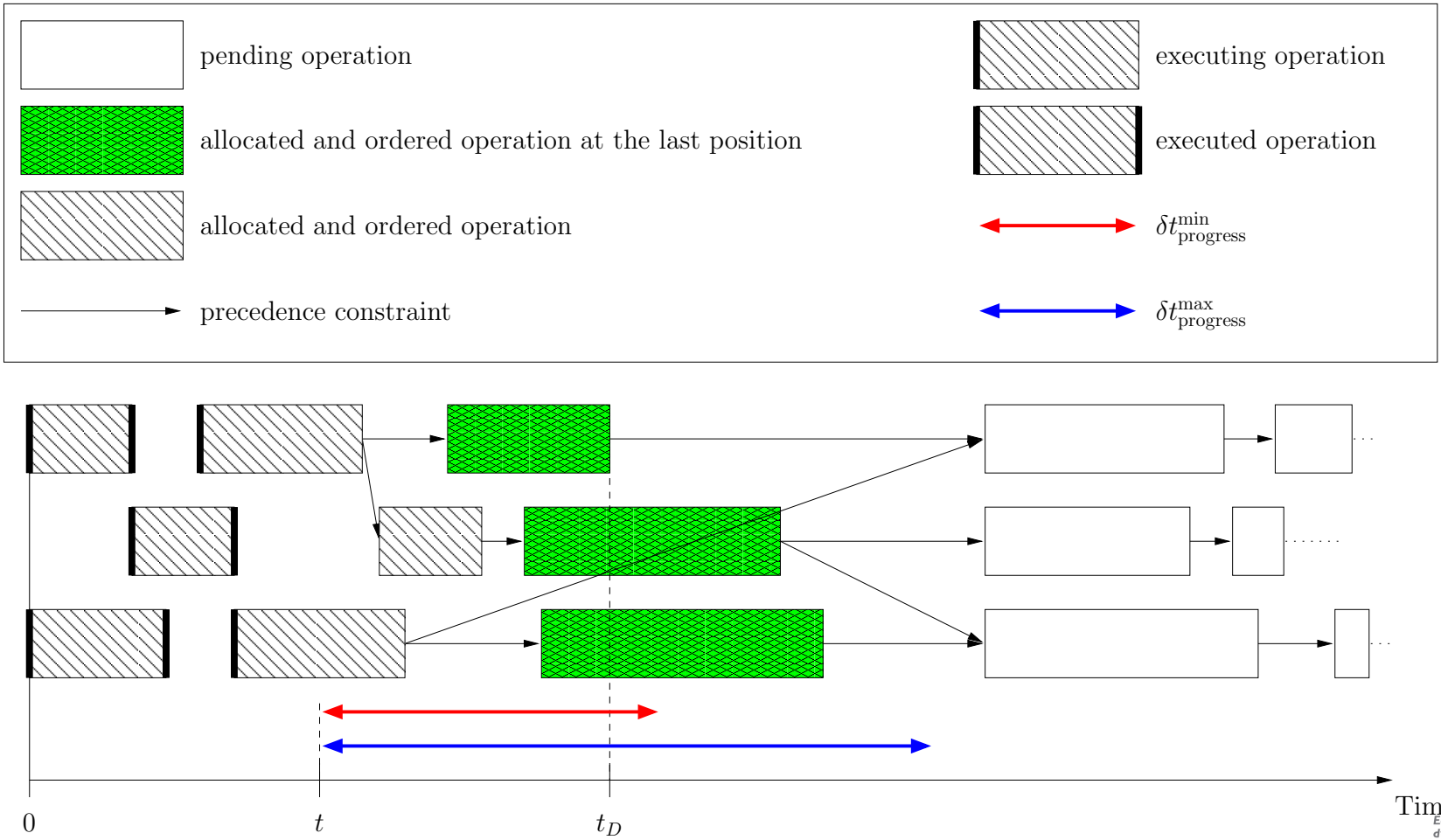
- extended JSSP: machines may break down (imprecise start times and durations), alternative resources, a due date associated with each job, allocation and tardiness costs, a maximal global cost $K^{\max}$
- objective: maximization of the probability that $K^{\text{eff}} < K^{\max}$
- large-scale problem instances (a few thousands of operations)
- solving technique:
  - fast random restart and Large Neighborhood Search alternatively with mean operation durations extended w.r.t. resource breakdown distributions, allocation and sequencing decisions made, then simulation of flexible schedules (proaction)
  - execution monitoring using different conditions:
    - sensibility (revision)
    - decisions made on a gliding horizon (progression)

Progression Conditions

Temporal conditions: $\delta t^{\min}_{\text{progress}}$ and $\delta t^{\max}_{\text{progress}}$
Uncertainty conditions (standard deviations): $\sigma t^{\min}_{\text{progress}}$ and $\sigma t^{\max}_{\text{progress}}$
t: current time
O_{toBeExe}: set of the operations to execute

- For starting selection:
  - $t_{\text{D}} - t \leq \delta t^{\min}_{\text{progress}}$, where $t_{\text{D}} = \min_{\forall p_i \in P}(\max_{\forall o_{ij} \in O_{\text{toBeExe}}}(end^{\exp}_{ij}))$ or
  - $\min_{\forall p_i \in P}(\max_{\forall o_{ij} \in O_{\text{toBeExe}}}(\sigma(end^{\exp}_{ij}))) \leq \sigma t^{\min}_{\text{progress}}$
- For stopping selection: operation o_{ij} of process plan p_i not selected when
  - $end_{ij-1} \geq t + \delta t^{\max}_{\text{progress}}$ and
  - $\sigma(end_{ij-1}) \geq \sigma t^{\max}_{\text{progress}}$

Progression Conditions (end)



Plan

Contributions

- une nouvelle classification des techniques de résolution et des systèmes pour planifier et ordonnancer sous incertitudes
- un modèle théorique intégrant notre classification et représentant la génération et l'exécution d'ordonnancements dans un environnement non déterministe
- nouveaux problèmes d'ordonnancement avec des données probabilistes
- étude expérimentale menée grâce à un prototype logiciel en C++ utilisant ILOG Solver et Scheduler

Remarques générales

- marier des techniques d'optimisation classiques avec de la simulation = une approche puissante pour s'attaquer à des problèmes complexes (simulation pour évaluer la qualité des solutions et pour contrôler le système de prise de décisions en surveillant l'exécution des solutions)
- résultats expérimentaux prometteurs (1^{re} version du prototype)
- plus de temps pour finir l'implémentation et lancer une campagne de tests plus complète

Perspectives

- finir l'implémentation de la 2^{e} version de notre prototype (génération des sous-problèmes déterministes, heuristiques et utilisation de la simulation pendant la recherche pour sélectionner des solutions)
- étendre notre prototype pour générer et exécuter des ordonnancements conditionnels (sous-groupes d'opérations mutuellement exclusifs)
- comparer nos résultats expérimentaux en termes de qualité/effort de calcul avec d'autres approches (*dispatching*, réordonnancements périodiques...)
- intégrer dans notre modèle théorique une procédure de type planification de tâches (raisonnement causal avec préconditions et effets des actions)


Bidot, J., Laborie, P., Beck, J. C., and Vidal, T. (2003).
Using simulation for execution monitoring and on-line rescheduling with uncertain durations.
In *Working Notes of the ICAPS'03 Workshop on Plan Execution*, Trento, Italy.


Billaut, J.-C., Moukrim, A., and Sanlaville, E., editors (2005).
Flexibilité et robustesse en ordonnancement.
Hermès.


Chien, S. A., Knight, R., Stechert, A., Sherwood, R., and Rapideau, G. (2000).
Using iterative repair to improve the responsiveness of planning and scheduling.
In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS)*, pages 300–307, Breckenridge, Colorado, United States of America.


Daniels, R. L. and Carrillo, J. E. (1997).
Beta-robust scheduling for single-machine systems with uncertain processing times.

IIE Transaction, 29:977–985.


Davenport, A. J., Gefflot, C., and Beck, J. C. (2001).
Slack-based techniques for building robust schedules.
In *Proceedings of the Sixth European Conference on Planning (ECP)*, Toledo, Spain.


Dubois, D., Fargier, H., and Prade, H. (1993).
The use of fuzzy constraints in job-shop scheduling.
In *Working Notes of the IJCAI'93 Workshop on Knowledge-based Production Planning, Scheduling, and Control*, pages 101–112, Chambéry, France.


Estlin, T., Rabideau, G., Mutz, D., and Chien, S. A. (2000).
Using continuous planning techniques to coordinate multiple rovers.
Electronic Transactions on Artificial Intelligence, 4, Section A:45–57.
http://www.ep.liu.se/ea/cis/2000/016/.


McKay, K. N., Safayeni, F. R., and Buzacott, J. A. (1988).

Job-shop scheduling theory: What is relevant?
INTERFACES, 18(4):84–90.


Morris, P. H., Muscettola, N., and Vidal, T. (2001).
Dynamic control of plans with temporal uncertainty.
In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 494–502, Seattle, Washington, United States of America.


Nuijten, W. P. M. (1994).
Time- and Resource-constrained Scheduling. A Constraint Satisfaction Approach.
Ph.D. dissertation, Technische Universiteit Eindhoven, Eindhoven, Netherlands.


Sadeh, N. M., Otsuka, S., and Schnelbach, R. (1993).
Predictive and reactive scheduling with the Micro-Boss production scheduling and control system.
In *Working Notes of the IJCAI'93 Workshop on Knowledge-based Production Planning, Scheduling, and Control*, Chambéry, France.


Smith, S. F. (1994).
OPIS: A methodology and architecture for reactive scheduling.
In Zweben, M. and Fox, M. S., editors, *Intelligent Scheduling*, pages 29–66. Morgan Kaufmann, San Francisco.


Vidal, T., Beck, J. C., and Bidot, J. (2003).
Vers un modèle intégrant les diverses approches d'ordonnancement sous incertitudes.
In *Proceedings of the Workshop "Risque" of Plate-forme de l'Association Française pour l'Intelligence Artificielle*, Laval, France.


Vidal, T., Ghallab, M., and Alami, R. (1996).
Incremental mission allocation to a large team of robots.
In *Proceedings of the Third International Conference on Artificial Intelligence Planning and Scheduling (AIPS)*, Edinburgh, Scotland.


Washington, R., Golden, K., and Bresina, J. L. (2000).
Plan execution, monitoring, and adaptation for planetary rovers.

Electronic Transactions on Artificial Intelligence, 4, Section A:3–21.
http://www.ep.liu.se/ej/etai/2000/004/.

Wu, S. D., Byeon, E.-S., and Storer, R. H. (1999).
A graph-theoretic decomposition of the job-shop scheduling problem to achieve scheduling robustness.
Operations Research, 47:113–123.