Un cadre général intégrant les approches pour ordonnancer sous incertitudes

Julien Bidot[1,2]

[1]ILOG S. A., France

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

23 novembre 2005

Plan

- Rappels en planification de tâches et ordonnancement
- Planification de tâches et ordonnancement sous incertitudes
- Classification des techniques et systèmes sous incertitudes
- Domaine d'application
- Generation and Execution Model
- Software Prototype
- Conclusions and Future Work

Plan

Planification de tâches classique

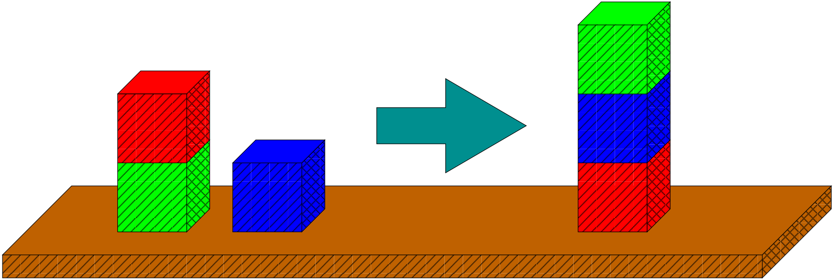
Problème :

- un ensemble d'opérateurs (un ensemble d'actions)
- un état initial
- un (plusieurs) état(s) but(s)
- environnement d'exécution déterministe

Objectif : sélectionner et organiser des actions dans le temps afin d'atteindre l'état but en partant de l'état initial tout en optimisant un critère ; par exemple, minimisation du nombre d'actions à effectuer

Planification de tâches classique (suite)

Exemple : monde des cubes



Ordonnancement classique

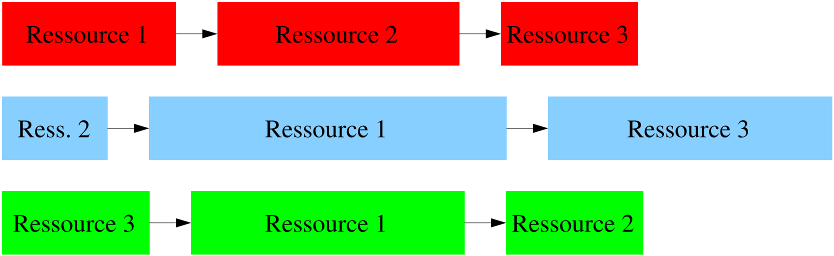
Problème :

- un ensemble d'opérations :
  - durées, ressources requises
  - relations temporelles (par ex. : précédences)
- un ensemble de ressources à capacités finies
- environnement d'exécution déterministe

But : assigner les ressources aux opérations et placer les opérations dans le temps en optimisant un critère ; par example, minimisation du retard maximum par rapport à une échéance

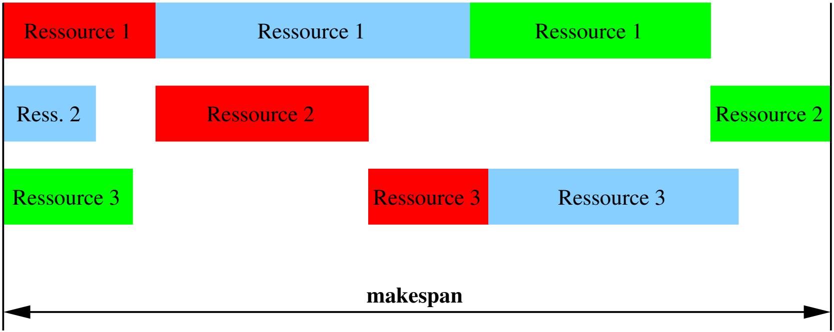
Ordonnancement classique (suite)

Exemple : problème d'ordonnancement à cheminements multiples (JSSP) avec minimisation du makespan



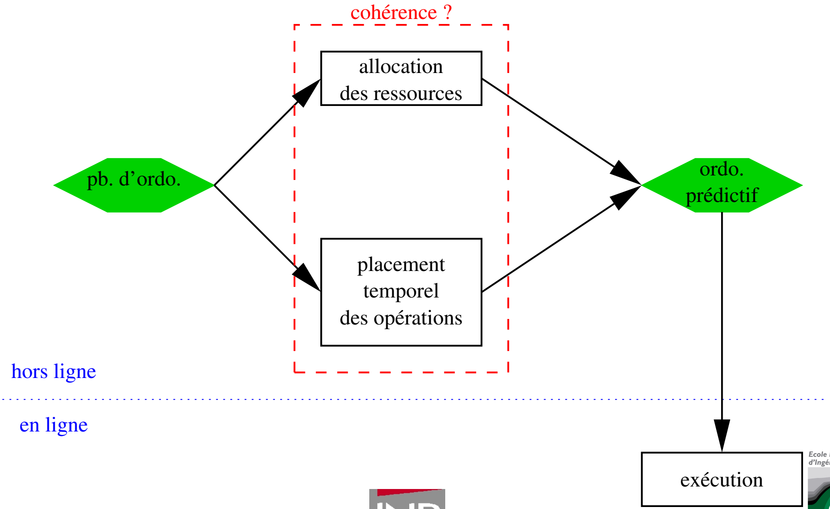
Ordonnancement classique (fin)

Exemple : solution optimale du JSSP présenté précédemment



Ordonnancement sans incertitudes

Schéma général de génération et d'exécution



Plan

Rappels en planification de tâches et ordonnancement

Planification de tâches et ordonnancement sous incertitudes

Classification des techniques et systèmes sous incertitudes

Domaine d'application

Generation and Execution Model

Software Prototype

Conclusions and Future Work

Changing the rules of business™

INP TOULOUSE

Ecole Nationale d'Ingénieurs de Tarbes

Laboratoire Génie de Production

Incertitudes en planification de tâches

Sources d'incertitudes : imprécision ou incertitude ?

- effets imprécis des actions à effectuer
- état du monde partiellement observable ou non observable
- états du monde incertains indépendamment des actions : observations nécessaires
- nouveaux états buts ajoutés pendant l'exécution

Incertitudes en ordonnancement

- durées des opérations ou délais entre plusieurs opérations
- dates de début au plus tôt des opérations
- fiabilités des ressources (pannes, capacités, absence, etc.)
- quantités de ressources requises
- nouvelle opération à effectuer
- etc.

Tentatives de définitions

Propriétés d'une solution :

- ordo. stable = ordo. prédit hors ligne (avant exécution) et exécuté sans modifications
- ordo. robuste = quoiqu'il arrive en ligne (pendant l'exécution), garantie que la qualité de la solution ne se dégrade pas

Caractéristique d'une solution : ordo. flexible = des choix sont laissés, l'arbitrage se faisant en ligne (plusieurs types de flexibilité) [Billaut et al., 2005]

Plan

Approche réactive

raisonnement en ligne = raisonnement en même temps que le processus s'exécute ; règles de priorité (dispatching) locales ou globales (état de l'art [Pinedo, 1995])

ordo. réactif

Approche prédictive-réactive

raisonnement hors ligne statique (avant d'exécuter l'ordo.)
et
modifications en ligne si nécessaire

Approche proactive

raisonnement hors ligne
avec prise en compte des incertitudes
[Davenport and Beck, 2000] et [Herroelen and Leus, 2002]

Classification des techniques et systèmes sous incertitudes (suite)

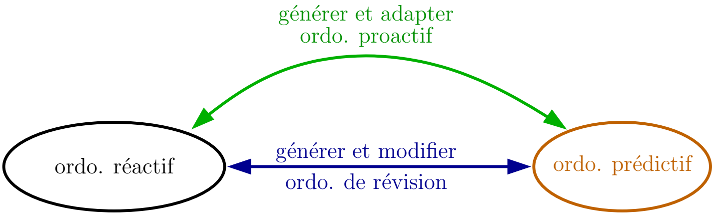
Motivations pour modifier et étendre la terminologie :

- décisions remises en question en ligne ou non : comment on modifie la solution en ligne
- décisons prises sur tout l'horizon du problème ou seulement sur une partie
- ambiguïté de l'approche réactive

→ techniques de révision et progressive

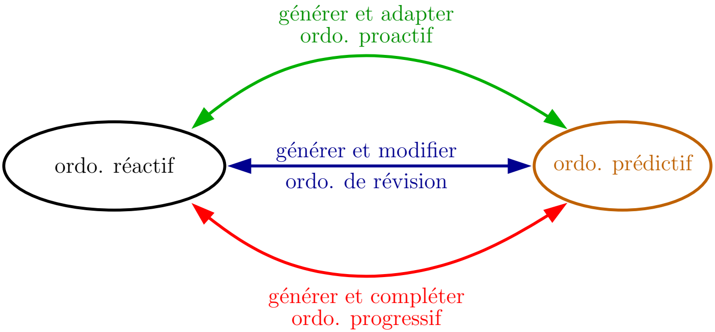
Approche de révision

raisonnement en ligne pouvant remettre en question tout ou partie des décisions → replanification ou réordonnancement, Micro Boss [Sadeh et al., 1993], OPIS [Smith, 1994]



Approche progressive

prise de décisions sur un horizon glissant sans remises en cause des décisions prises, plan de production [McKay et al., 1988], projet MARTHA [Vidal et al., 1996]



Classification des techniques et des systèmes sous incertitudes (fin)

Exemples mixtes :

- méthode proactive et de révision : recouvrement maximal avec β-*Robustness* [Daniels and Carrillo, 1997] et ordo. flou [Dubois et al., 1993] ; planification de tâches [Washington et al., 2000]
- méthode proactive-réactive : modifications des durées des opérations critiques [Davenport et al., 2001], opérations partionnées et séquencées hors ligne [Wu et al., 1999]
- méthode progressive et de révision : planification de tâches continue [Chien et al., 2000] et [Estlin et al., 2000]

Plan

Rappels en planification de tâches et ordonnancement

Planification de tâches et ordonnancement sous incertitudes

Classification des techniques et systèmes sous incertitudes

Domaine d'application

Generation and Execution Model

Software Prototype

Conclusions and Future Work

Changing the rules of business™

TOULOUSE

Ecole Nationale d'Ingénieurs de Tarbes

Laboratoire Génie de Production

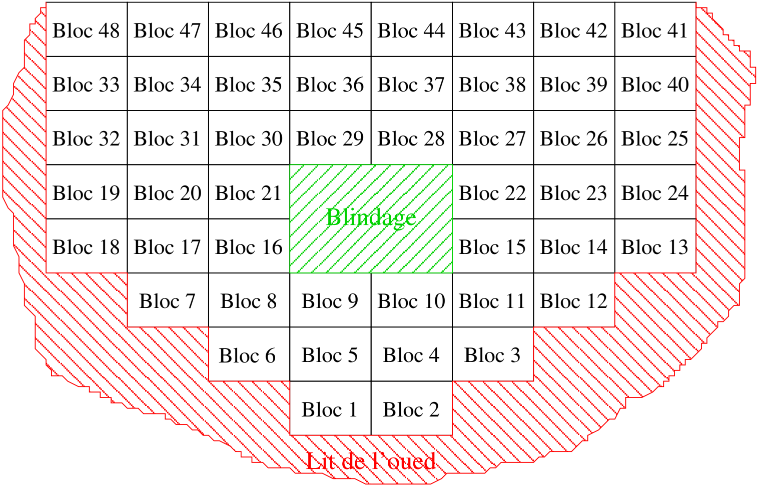
Notre domaine d'application

Gestion de projets : ingénierie civile (construction de barrages)

1. routes
2. habitations en préfabriqué
3. tunnel de dérivation
4. fondations du barrage
5. assèchement du lit de l'oued
6. canal d'assèchement
7. tunnel d'observation
8. mur du barrage et batardeaux
9. fermeture du tunnel de dérivation

Notre domaine d'application (suite)

Mur en béton (coupe transversale) :



Notre domaine d'application (fin)

Incertitudes :

- découverte de mauvaises conditions géologiques
- retard des fournisseurs (blindage, matières premières, etc.)
- occurrence des crues (statistiques)

Coûts :

- achats ou locations (matières ou machines), salaires
- pénalités de retard

But : assurer un coût total max. inférieur à une borne

Plan

Changing the rules of business™

Ecole Nationale d'Ingénieurs de Tarbes

Laboratoire Génie de Production

Constraint satisfaction approach

constraint model of a scheduling problem:

- a set of variables associated with domains
- constraints = a set of logical and/or arithmetic relationships between variables
- each constraint associated with a propagation algorithm that can reduce the domains of the variables involved (initial and then incremental propagation); an empty domain (after propagation) $\rightarrow$ the satisfaction problem is inconsistent

Constraint satisfaction approach

constraint model of a scheduling problem:

- a set of variables associated with domains
- constraints = a set of logical and/or arithmetic relationships between variables
- each constraint associated with a propagation algorithm that can reduce the domains of the variables involved (initial and then incremental propagation); an empty domain (after propagation) $\rightarrow$ the satisfaction problem is inconsistent

Example (constraint propagation)

1. 2 integer variables x and y, each with domain $\{1; 2; 3\}$
2. posting the following constraint: $x > y$
3. after propagation: $x \in \{2; 3\}$ and $y \in \{1; 2\}$

Constraint satisfaction approach (cont'd)

Example (scheduling problem)

- start, processing, and end times of operations = variables
- resource requirements = variables
- optimization criteria = 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, we have to sequence o_k and o_l: $start_k \geq end_l \vee start_l \geq end_k$ (decision made during search)

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 progressive) 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

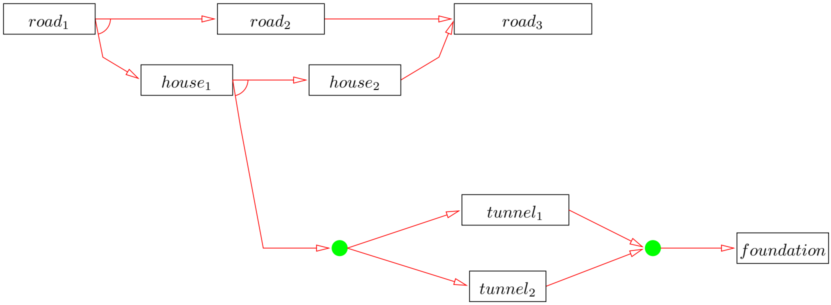
Toy example of a dam project:

- 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

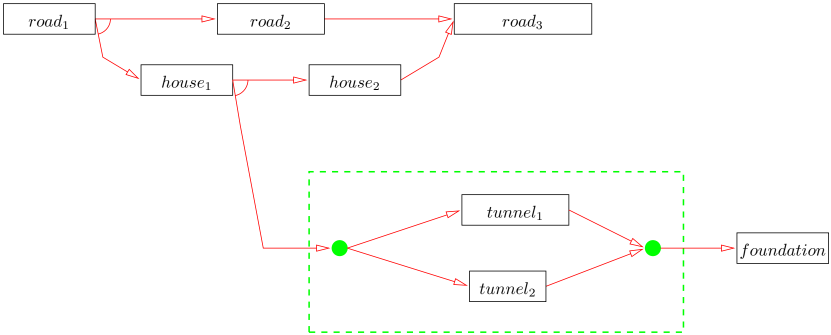
Expressivity of our Model (cont'd)

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



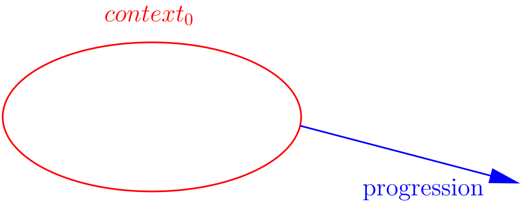
Expressivity of our Model (cont'd)

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 ($context_1$)

Changing the rules of business

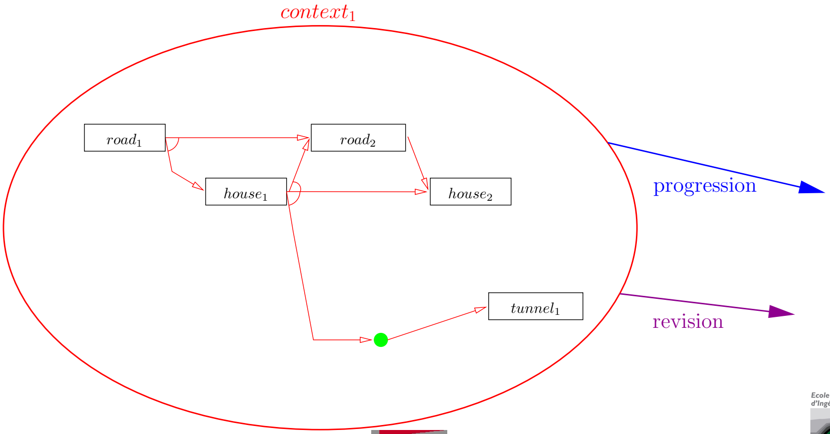
TOULOUSE

Ecole Nationale d'Ingénieurs de Tarbes

Laboratoire Génie de Production

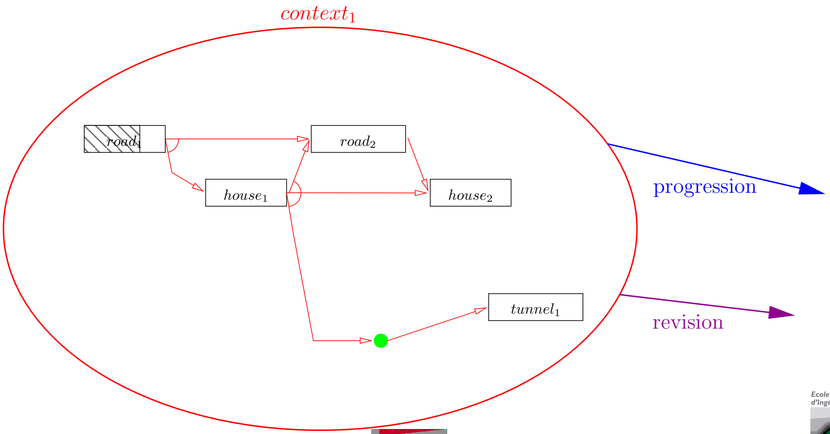
Expressivity of our Model (cont'd)

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*$_1$:

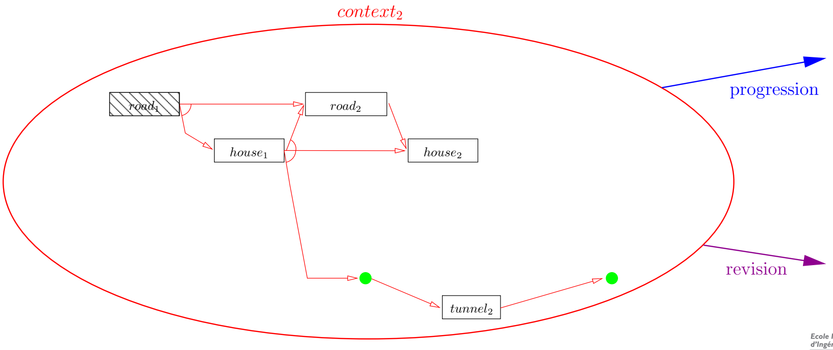


Expressivity of our Model (cont'd)

Model when $road_1$ finished: $tunnel_2$ chosen based on geological conditions for the tunnel $\rightarrow$ revision 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 $\rightarrow$ revision transition activated and a new context generated (*context*$_2$)

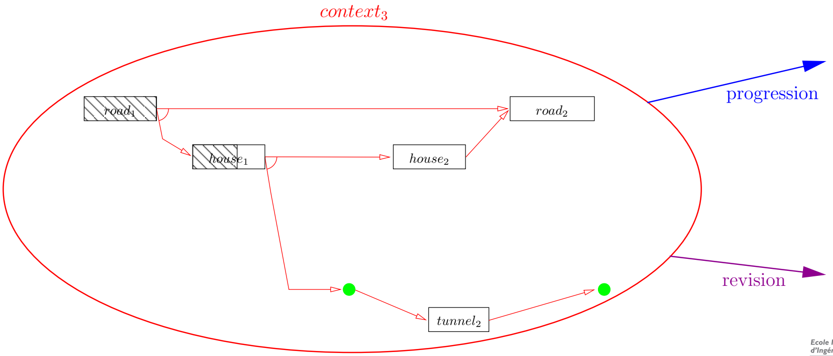


Expressivity of our Model (cont'd)

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 (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*$_3$)

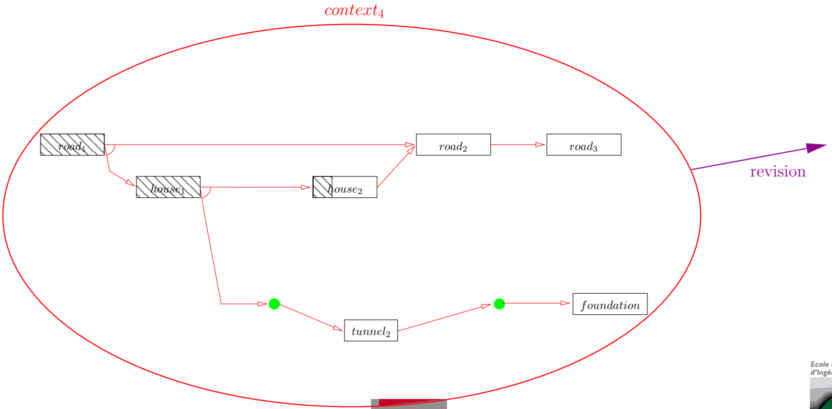


Expressivity of our Model (end)

Model when *house*$_2$ executing: need to select new operations $\longrightarrow$ progression transition activated and a new context generated (*context*$_4$)

Expressivity of our Model (end)

Model when *house*$_2$ executing: need to select new operations $\rightarrow$ progression transition activated and a new context generated (*context*$_4$)



Plan

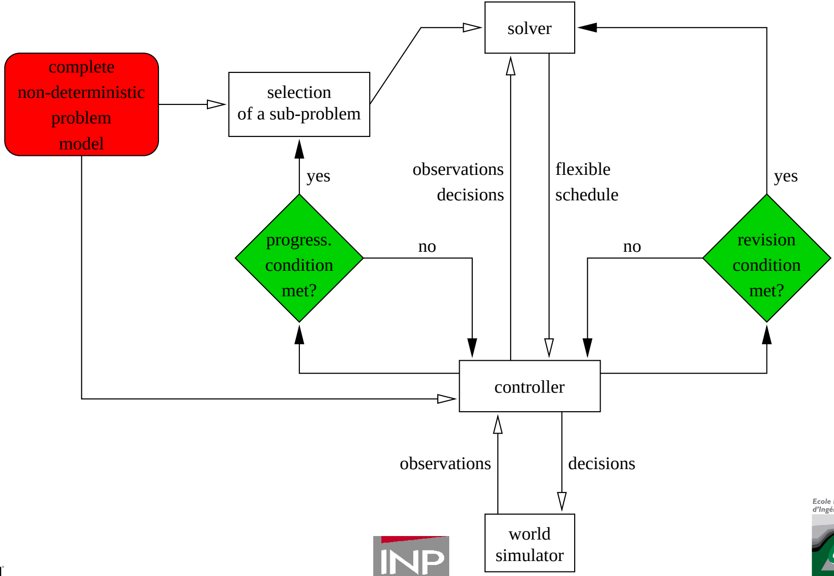
Ecole Nationale d'Ingénieurs de Tarbes

Laboratoire Génie de Production

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 on the basis of a precedence graph for estimating the schedule quality: on-line maintenance of operation start and end times simulating execution
  - execution algorithm = start operations as soon as possible (makespan or tardiness minimization)

Architecture



First Version of our Experimental System

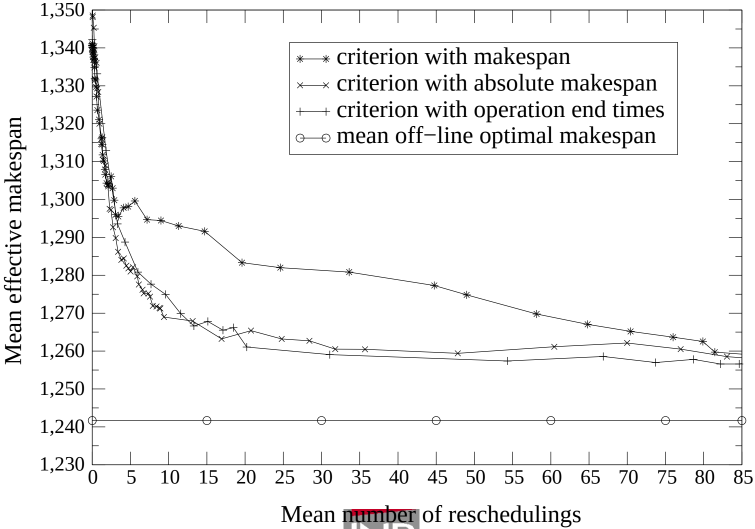
Mixed approach: proaction and revision

- minimization of expected makespan
- an indicative predictive schedule generated off line (tree search with mean operation durations and branch-and-bound procedure → proaction)
- rescheduling of still non executed operations when revision criterion verified → 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]

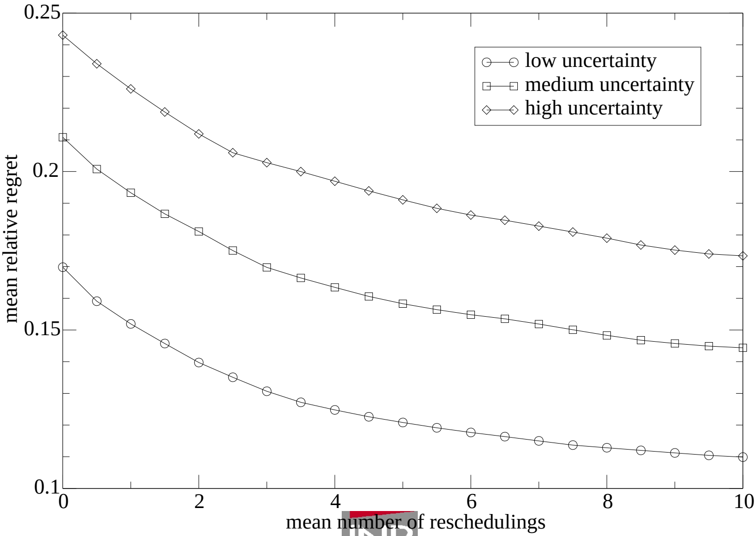
First Version of our Experimental System (cont'd)

Experiments with 3 revision criteria (la11):



First Version of our Experimental System (end)

Impact of uncertainty level on makespan (10 problem instances):



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 the effective global cost $< K^{\max}$
- large-scale problem instances (a few thousands of operations)
- resolution technique:
  - iterative sampling and then Large Neighborhood Search (Slice-Based Search) 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)

Plan

Conclusions

Contributions:

- a new way of classifying resolution methods and systems used in task planning and scheduling under uncertainty
- a theoretical model representing how to interleave schedule generation and execution
- new scheduling problems with probabilistic data
- experimental study conducted with a software prototype in C++ using ILOG Solver and Scheduler libraries

General remarks:

- using standard optimization techniques with simulation = a powerful approach to tackle a large area of complex problems: simulation to both evaluate solution quality and monitor conditions during execution
- promising experimental results (1$^{\text{st}}$ version of our prototype)
- more time needed to finish implementation and run systematic experiments

Future Work

- finish the implementation of the 2nd version of our prototype: using simulation during search to select solutions
- extend our prototype to generate and execute a conditional schedule
- compare our results (quality/computational effort) with other scheduling techniques, such as dispatching
- integrate a task planning procedure in our generation and execution model

Plan

Appendix
- Revision Criteria
- Progression Conditions

Appendix Outline

Revision Criteria

- revCrit$_1$:

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

- revCrit$_2$:

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

- 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

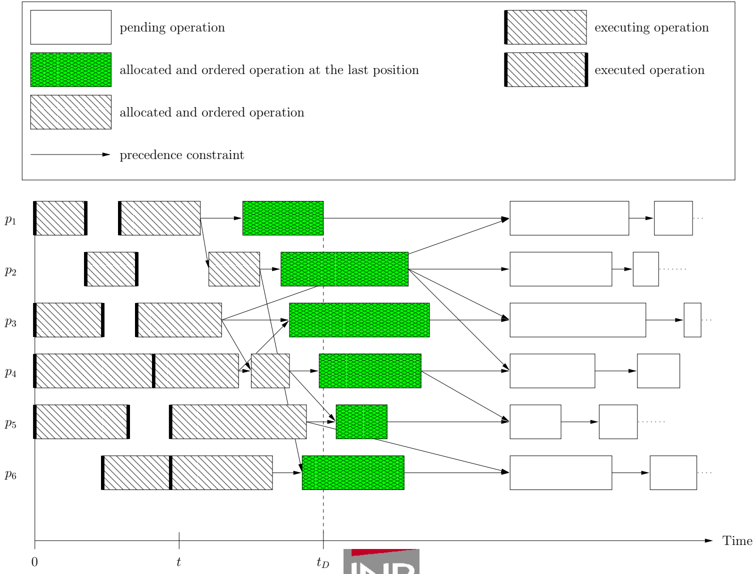
Progression Conditions

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

t: current time

O_{toBeExe}: set of the operations to execute

Progression Conditions (end)



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. and Beck, J. C. (2000).
A survey of techniques for scheduling with uncertainty.
Unpublished manuscript available at
http://www.eil.utoronto.ca/profiles/chris/chris.papers.html.

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/.

Herroelen, W. S. and Leus, R. (2002).
Project scheduling under uncertainty–survey and research potentials.
In *Proceedings of the Eighth International Workshop on Project Management and Scheduling (PMS'02)*, Valencia, Spain.

McKay, K. N., Safayeni, F. R., and Buzacott, J. A. (1988).
Job-shop scheduling theory: What is relevant?
INTERFACES, 18(4):84–90.

Pinedo, M. L. (1995).
Scheduling: Theory, Algorithms and Systems.
Prentice Hall.

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.