Dynamic Scheduling

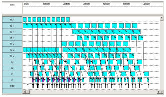
Hana Rudová

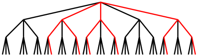
Fakulty of Informatics, Masaryk University
Brno, Czech Republic

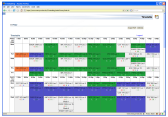
http://www.fi.muni.cz/~hanka

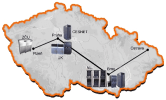
Dynamic scheduling

1. Problem description
   - Static problem
   - Dynamic problem
   - Minimal perturbation problem
2. Solving methods
   - Dispatching rules
   - Iterative forward search
   - Local search
   - Branch & Bound
3. Applications
   - University course timetabling
   - Scheduling in Grid environment





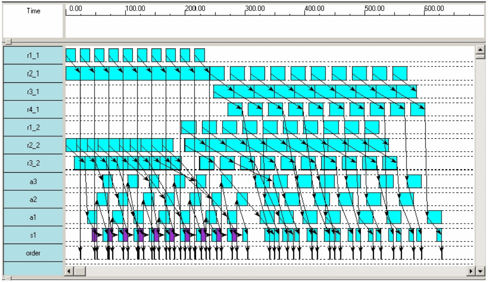




Basic terminology: Scheduling and CSP

Scheduling

- optimal resource allocation of activities over the time



Visopt ShopFloor System

Constraint Satisfaction Problem

- domain variables, domains, constraints = relations
- solution: variable assignment satisfying all constraints
- objective function + optimization

Scheduling problem as a CSP

Domain variables

- allocation of activity A in time and space
- time assignment: start(A), p(A), end(A)
- space allocation: resource(A)

Domains

- release time and due date for time variables
- alternative resources for space variables

Constraints: resource constraints, temporal constraints, ...

Objective function

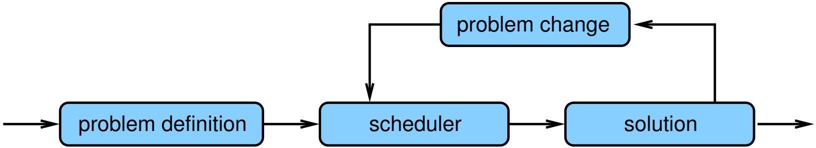
- resource usage, preferences over time and resources, ...

Dynamic scheduling

- Problem change during the solving process
- Examples:
  - unscheduled resource breakdown
  - new jobs appear during solving phase
  - shorter/longer processing time than expected

Dynamic scheduling

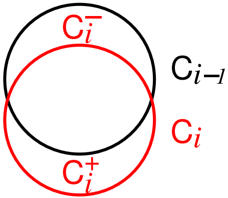
- Problem change during the solving process
- Examples:
  - unscheduled resource breakdown
  - new jobs appear during solving phase
  - shorter/longer processing time than expected



Dynamic CSP

- Dynamic constraint satisfaction problem:
  sequence $\{P_0, P_1, \ldots, P_n\}$ where
  - each P_i is a CSP given by the set of constraints C_i
  - C_i^+ set of new constraints
  - C_i^- set of removed constraints

  and it holds
  - $C_i^- \subseteq C_{i-1}$
  - $C_i = C_i^+ \cup C_{i-1} \backslash C_i^-$



- Note:
  - domain changes can be encoded into constraints

Minimal perturbation problem

- New solution with minimal changes/perturbation from original solution
    - solution published
    - changes undesirable wrt. user
    - try to avoid avalanche effect
- Example
    - school timetable
    - flight assignment
    - ...

Minimal perturbation problem

- New solution with minimal changes/perturbation from original solution
  - solution published
  - changes undesirable wrt. user
  - try to avoid avalanche effect
- Example
  - school timetable
  - flight assignment
  - ...
- Minimal perturbation problem: $(P, \alpha, C^+, C^-, \delta)$
  - P is a CSP with the set of constraints C
  - α solution of P
  - C^+, C^- sets of new and removed constraints
  - δ distance function over two CSP solutions
- Solution of minimal perturbation problem
  - CSP solution β of the problem with $C \backslash C^- \cup C^+$ with minimal $\delta(\beta, \alpha)$

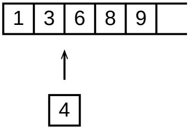
Dispatching rules

Dispatching rules

- new job assigned to the resource wrt. rule
- rule = longest job/job with smallest due date/... earlier
- combination of several rules

Good for problems with

- high dynamics
  - subsequently appearing jobs
  - uncertain processing time
  - resource breakdowns
- the need for a prompt response

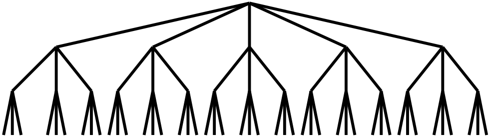


Not helpful for (complex) optimization

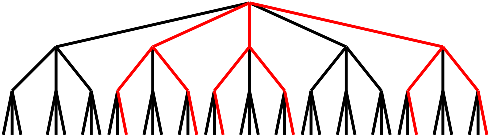
- minimal perturbation problem
- Quality of Servise (QoS) requirements

Search in the solution space

- Trivial enumeration



- CSP solving: search + constraint propagation
  - propagation =
    removal of inconsistent values from variable domains



Iterative forward search (IFS) for CSP

- Constraint propagation
- Constructive method
- Partial consistent assignment

- Standard forward step
  - selection of variables and their values if possible
- "Backward step"
  1. if impossible to assign selected variables without constraint violation
  2. finding conflicting variables and removal of their assignment
  3. assignment of selected variables

- Consequence
  - IFS is not a tree search method
  - principles of the local search
- Important principle
  - value selection
  - conflict statistics (memory for the number of conflicts)

Timetable

PHYS 112 (268)	7:30a	8:00a	8:30a	9:00a
Mon			ENGR 126R Lec 1 4, 54, 0	
Tue	OLS 252 Lec 1 15, 1, 3		PHYS 272 Lec 1 0, 17, 0	
Wed			ENGR 126R Lec 1 4, 54, 0	

Conflict statistics

- Suppose: selection of value a of variable A
  we must remove assignment of b for variable B, i.e.

$$[A = a \rightarrow \neg\ B = b]$$

- We can remember during computation

$$\begin{aligned} A = a \quad \Rightarrow \quad & 3 \times \neg\ B = b, \quad 4 \times \neg\ B = c \\ & 2 \times \neg\ C = a, \quad 120 \times \neg\ D = d \end{aligned}$$

Conflict statistics

- Suppose: selection of value a of variable A
  we must remove assignment of b for variable B, i.e.

$$[A = a \rightarrow \neg\ B = b]$$

- We can remember during computation

$$\begin{array}{rcll} A = a & \Rightarrow & 3 \times \neg\ B = b, & 4 \times \neg\ B = c \\ & & 2 \times \neg\ C = a, & 120 \times \neg\ D = d \end{array}$$

- Value selection
  - selection of values with smallest conflicts
- Aging
  - values remebered for some last iterations
  - older conflicts with smaller weight

IFS for dynamic CSP

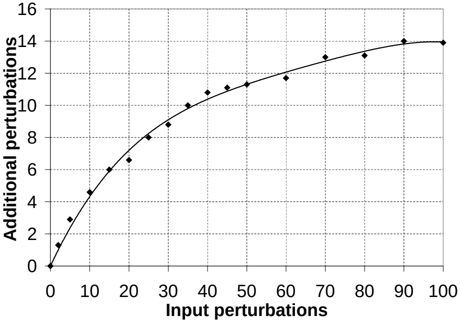
CSP: search goals

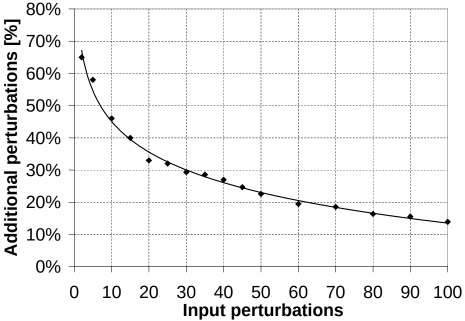
1. search for assignment of all values
2. optimization of the objective function
   - preferred time assignment, preferred resource allocation, ...

Minimal perturbation problem: extension

3. minimization of the number of different values
4. original values preferred
   - during value selection supported
     minimal difference in the number of differently assigned values

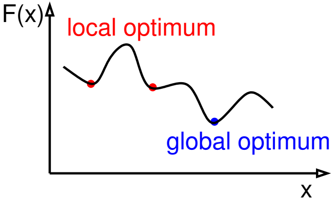
IFS: experiments





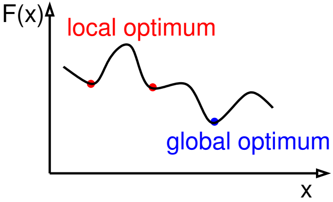
Local search LS

- Local changes/repairs of complete inconsistent assignment
- Initial assignment (trivially: random)
- Local changes
    - exchange of two jobs,
      job moved to different resource, ...



Local search LS

- Local changes/repairs of complete inconsistent assignment
- Initial assignment (trivially: random)
- Local changes
  - exchange of two jobs,
    job moved to different resource, ...



- CSP: search goals
  1. improvements in objective function
  2. search for consistent assignment
- Dynamic CSP: extension
  3. solution of original problem as initial assignment
  4. natural extension
  5. good time complexity
- Minimal perturbation problem: extension
  6. minimization of the different values
  7. no optimality guarantees

Branch & Bound (BB)

Branch & Bound search

- tree search method
- search for optimal solution
- maintained lower bound
- space with quality worse than lower bound pruned

Branch & Bound (BB)

Branch & Bound search

- tree search method
- search for optimal solution
- maintained lower bound
- space with quality worse than lower bound pruned

Iterative forward search, local search

- large scale problems
- no guarantee of solution quality

Branch & Bound

- not applicable for large scale problems
- can be used for local optimization

BB: dynamic CSP

We have: solution of original problem + new changes

and want "improve it"

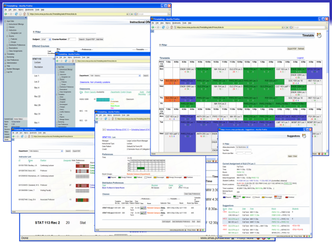
Selection of variable x with improper assingment

Application of BB with at most n changes

- reasonable n is rather small
- reasonable search space size
- optimality guarantee within this search space
- solution out of this space not reasonable wrt. x anyway

1. Problem description
   - Static problem
   - Dynamic problem
   - Minimal perturbation problem
2. Solving methods
   - Dispatching rules
   - Iterative forward search
   - Local search
   - Branch & Bound
3. Applications
   - University course timetabling
   - Scheduling in Grid environment

University course timetabling

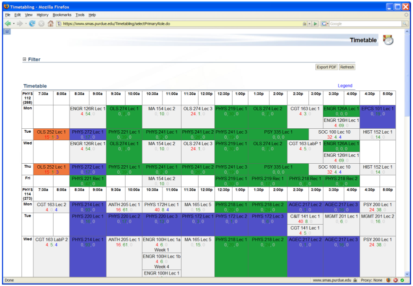


PURDUE UNIVERSITY

- Timetabling at Purdue University
- Large scale decentralized problem
  - 2 central problems: large lecture rooms, computer laboratories
  - about 70 problems for particular administrative domains
- State of the project
  - 2001-2005: 3 projects between Masaryk and Purdue University
  - spring 2005: first used large lecture room timetabling
  - autumn 2007: first used system for the whole university
- Timetabling steps
  1. centralized solving of large lecture rooms
  2. solving of particular departmental problems
  3. centralized solving of computer laboratories problem
  4. centralized changes in the timetable

Static timetabling problems

- Large lecture room problem
  - 800 courses x 2 meetings, 55 classrooms
  - individual course enrollment for about 28.000 students
  - preferences on time and space, distances between classrooms,...
  - many exception handled
- Other problems: both automated solving and data input
- Algorithm: Iterative forward search



Minimal perturbation problem

Solving methods

- Iterative forward search to react on user requests
- Branch and Bound for local schedule optimization

Suggestions

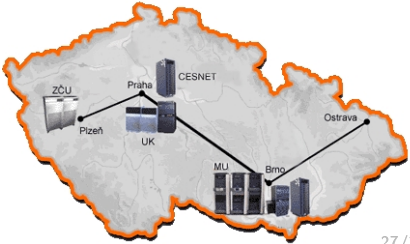
Score	Class	Date	Time	Room	Students
+15.2	POL 101 Lec 3	Full Term	TTh 12:00p → TTh 7:30a	BRNG 2280	+11
+31.7	POL 101 Lec 3	Full Term	TTh 12:00p → TTh 10:30a	BRNG 2280	+36 (h+3)
	HIST 342 Lec 1	Full Term	TTh 10:30a → TTh 1:30p	BRNG 2280 → BRNG 2290	
+36.6	POL 101 Lec 3	Full Term	TTh 12:00p → TTh 10:30a	BRNG 2280	+36 (h+4)
	HIST 342 Lec 1	Full Term	TTh 10:30a → TTh 7:30a	BRNG 2280	
+44.1	POL 101 Lec 3	Full Term	TTh 12:00p → TTh 10:30a	BRNG 2280	+34 (h+2)
	HIST 342 Lec 1	Full Term	TTh 10:30a → TTh 3:00p	BRNG 2280 → BRNG 2290	
	OBHR 330 Lec 4	Full Term	TTh 3:00p	BRNG 2290 → LWSN B155	

(all 1571 possibilities up to 3 changes were considered, top 4 of 17 suggestions displayed)

Search Deeper

Job Scheduling in Grid environment

- Distributed heterogeneous resources
  - clusters, supercomputers, desktops, special instruments
  - autonomy, robustness
- Type of jobs
  - high-throughput (parameter studies), paralell, distributed
- Scheduling on Grids
  - scheduling of computational jobs to resources
  - scheduling of data transfers and data processing
- Examples
  - several clusters over Czech Republic (*META Center*)
    hundreds clusters over Europe (EGEE)
  - several supercomputers(Deisa)
  - SETI@Home



Grid scheduling with Alea simulator

Problems with QoS requirements

- synthetic problems proposed by the group at ISTI CNR (Italy)
- heterogenous resources (frequency, number of processors)
- dynamic problems
  - jobs arriving over the time
  - known job processing time
  - no resource changes
- due dates, sw licences
- resource usage, job slowdown

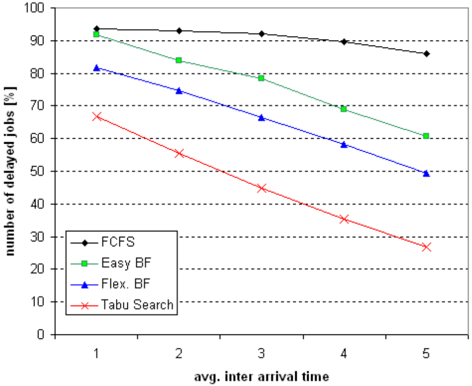
Grid simulator Alea

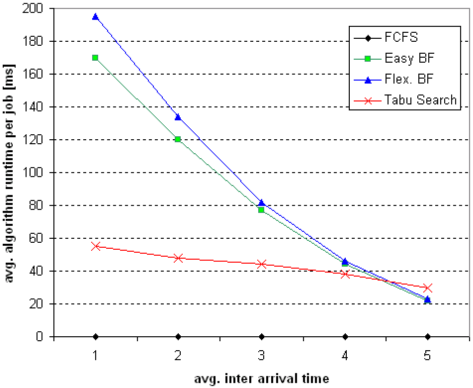
- extension of simulation tool GridSim (Java)
- centralized scheduler
- modular system: different problems and algorithms
- static and dynamic scheduling
- `http://www.fi.muni.cz/~klusacek/alea`

Local search for dynamic grid scheduling

- Dispatching rules
  - common use in the theory and practice
  - used for generation of the initial solution
- Local search
  - applied to static grid scheduling problems - very slow
    grid community very sceptical to
    the effective use of local search!
  - local search still not applied to dynamic grid scheduling
- Goal
  - application of local search to dynamic grid scheduling problems
  - generally
    - parallel jobs, QoS, interactive jobs, advance reservation, ...
    - application of advanced AI/OR algorithms
- Current results
  - synthetic problems with QoS requirements
  - improvements in optimization results
  - very good running time

Local search: experiments





Comparison of grid scheduling algorithms

Algorithms in Grid simulators

- SimGrid, GridSim, GSSIM , Alea
- development and testing of new algorithms
- comparison of algorithms

Algorithms in production systems

- PBSPro, SGE, Maui, Moab
- in simulators: approximated with FIFO (with backfilling)
- hard to reimplement
  - many rules, features, bugs
  - closed source, algorithms not published

Goal: comparing our and production algorithms

Simulator with Production Scheduling Algorithms

PBSPro production system

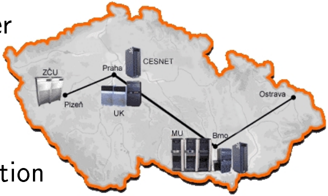
Magrathea tool

- for management of virtual machines
- developed at CESNET and Masaryk University

Simulator with virtual machines (VMs)

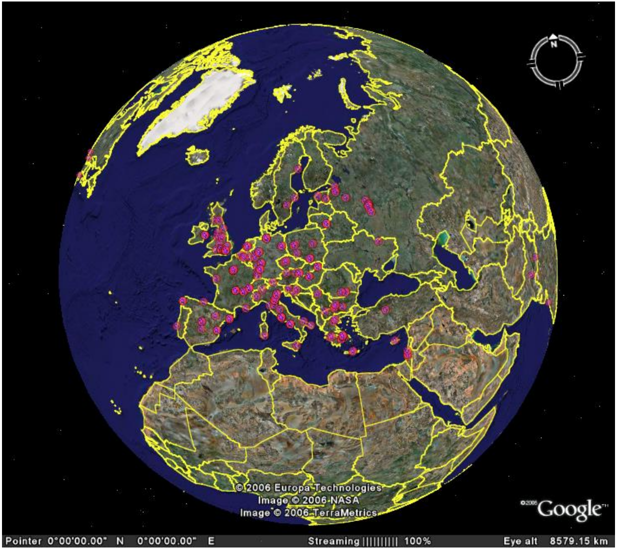
- running on 16 core AMD machine
- several VMs running on a single computer
- VMs representing 300 nodes

- jobs submitted directly to PBS in VMs
- sleep jobs with no cpu/memory consumption



- real workloads from Czech Grid *META Center* . . . 2005-2007
- run time reduction: 1 month $\rightarrow$ 1.5 hours, 1 year $\rightarrow$ $\leq$ 1 day

Motivation for future work



egee
Enabling Grids for E-sciencE

Now:
48 countries
243 clusters
44.000 CPU
daily over
100.000 jobs

Next year:
expected 10x
job increase