Workshop Scheduling a Scheduling Competition
ICAPS 2007, Providence, USA, September 22 – 26, 2007

Towards the Benchmarks for Scheduling Problems

Sanja Petrovic

Automated **S**cheduling, Optimis**a**tion and **P**lanning Group (ASAP)
School of Computer Science
University of Nottingham, UK

Issues to be discussed:

1. Types of Scheduling Problems
2. Formal Representation of Scheduling Problems
3. Evaluation of Scheduling Algorithms
4. Randomly Generated Instances versus Real-world Problems
5. Conclusions to be Drawn After the Competition

1. Types of Scheduling Problems

Production scheduling
Employee scheduling
University timetabling
...?

- Tailored made algorithms versus general algorithms?
- How to rank the algorithms?
- Static and/or dynamic problems?

Existing benchmark problems:

Production scheduling

Generated by Demirkol et al. 1996 (~10000 data sets)
http://cobweb.ecn.purdue.edu/~uzsoy/ResearchGroup/Index.html

Generated by Taillard, 1993 (260 data sets)
http://people.brunel.ac.uk/~mastjjb/jeb/orlib/jobshopinfo.html

Nurse rostering

http://www.cs.nott.ac.uk/~tec/NRP
13 data sets

University timetabling

Course timetabling: 1st International Timetabling Competition
Examination timetabling: http://www.cs.nott.ac.uk/~yxb/TTdata

2. Formal Representation of Scheduling Problems

Production scheduling

- precedence constraints
- release dates
- due dates
- machine availability
- etc.

Employee Scheduling

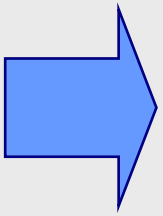
- cover
- maximum number of days
- minimum number of days
- number of weekends
- preferred working shifts
- etc.

University timetabling

- no person in more than one place
- exams of each student should be equally spread
- etc.

3. Evaluation of the Scheduling Algorithms

Multicriteria problem
Efficiency and effectiveness
Flexibility and extensibility
Learning capabilities



Single criterion
Quality of the generated solution given CPU time

- Need for a software for calculating the value of the objective function(s)
- Need for new instances to check the performance of the algorithm(s)

4. Randomly Generated Instances versus Real-world Problems

- Randomly generated benchmark problems
  enable the control of the properties of the problem.

- Real-world problems
  - can be of structure that is difficult to capture
  - confidential
  - complex and may require synergy of algorithms
  - fraught with uncertainties
  - usually require rescheduling

5. Conclusions to be Drawn After the Competition

- No definite conclusion about the superiority of a single algorithm should be drawn.

- The algorithm developers should use the benchmark data sets to analyse which type of problems their algorithms can handle well, and to compare their results with the results obtained by using other algorithms.