Lecture 29: Last lecture

- Scheduling: Heuristics from the disjunctive graph
- Scheduling: Computational complexity, approximability
- Research problems, topics for theses etc.

Lower bounds of schedule cost

Given a disjunctive graph, define for an operation o

- head(o): time necessarily needed before processing o

  Highest duration of a directed path that ends in o

- tail(o): time necessarily needed after processing o

  Highest duration of a directed path that starts from o

Lower bounds of schedule cost (cont'd)

Define for a set S of operations

- the shortest head $H(S) = \min_{o \in S} \text{head}(o)$
- the shortest tail $T(S) = \min_{o \in S} \text{tail}(o)$
- the sum of processing times $P(S) = \sum_{o \in S} d(o)$

Lower bounds of schedule cost (cont'd)

Given a set of operations S on one machine, $H(S)+P(S)+T(S)$ is a lower bound on the cost of the schedule:

- Operations S cannot overlap because they are on the same machine: at least time $P(S)$ is needed for processing S.
- If from operations in S the one with the smallest head is performed first, at least time $H(S)$ is needed before S.
- If from operations in S the one with the smallest tail is performed last, at least time $T(S)$ is needed after S.

Lower bounds of schedule cost (cont'd)

Let O_m be the set of operations on machine m.

Now a lower bound on the cost of schedule is

$$\max_{m \in M} \left(\max_{S \subseteq O_m} H(S) + P(S) + T(S) \right)$$

In other words, we compute the lower bounds on all sets S of operations that are computed on one machine.

Algorithms for scheduling: local search

- Idea: two schedules are neighbors if one can be obtained from the other by a small modification (to its graph).
- Modifications:
  - reverse an arrow, or
  - reorder consecutive operations in the graph (preserving their locations in their respective jobs.)

  Modifications must preserve acyclicity

Algorithms for scheduling: local search

Finding good schedules proceeds as follows:

1. Start from an randomly chosen schedule.

2. Go from the current schedule to a neighboring schedule (if the neighboring schedule is sufficiently good.)

3. Algorithms: simulated annealing, tabu search, ...

Computational intractability of scheduling

Optimal solutions for job shop scheduling can be found polynomial time if

- number of jobs is 2,
- number of machines is 2, all jobs have 1 or 2 operations, or
- number of machines is 2, all operations have duration 1.

In all cases the problem obtained by incrementing the number of machines, jobs, operations or durations by 1, is NP-hard.

Approximability of job shop scheduling

THEOREM (Williamson et al. 1993) Deciding if there is a schedule of length 4 is NP-complete.

COROLLARY There is no polynomial-time algorithm that finds schedules of length $< \frac{5}{4}$ from optimal (unless P=NP.)

PROOF SKETCH: A schedule of length 4 exists if and only if p-approximation algorithm with $p < \frac{5}{4}$ finds a schedule of length 4. (Schedule of length 5 would be more than p from the optimal.)

THEOREM (Shmoys et al. 1994) There is a poly-time algorithm that produces schedules of length $\frac{\log^2 m}{\log \log m}$ times the optimal.

Open research problems: deterministic planning

- progression vs. regression
- systematic vs. local search algorithms
- optimal plans: heuristic search vs. propositional logic

Topics for theses: deterministic planning

1. Experimental: progression vs. regression

   Compare two planners (same search algorithm and distance heuristic), one using progression, the other regression.

2. Experimental/theoretical: phase transitions in planning

   Probability of plan existence on average can be predicted from quantitative properties, e.g. $\frac{\text{number of operators}}{\text{number of state variables}}$.

   Analyze the connection between plan existence and difficulty of planning both analytically and experimentally.

Open research problems: conditional planning (w/wo probabilities)

- What kind of algorithms are there?

- Algorithmic improvements: Symmetry reduction, partial-order reduction, ...

- Heuristics for guiding algorithms for partial observability

Topics for theses: conditional planning

1. Theoretical/experimental: symmetry reduction for conditional planning

2. Theoretical/experimental: efficient algorithms for planning with partial observability

   Efficient updates of value functions, better representations of value functions, ...

3.

Topics for theses: other areas

1. Literature review (as a Studienarbeit): algorithms for crew scheduling, flight timetabling, fleet assignment, or job shop scheduling