# CS227: Assignment 3

The third assignment involves implementing and evaluating **scheduling algorithms**, and writing a report on your experiments.

**Algorithm:** Implement a basic job shop scheduling algorithm with the following features:

1. Convert the basic constraints of the job shop scheduling problem (including precedence constraints between operations, release time constraints, and due time constraints) into a simple temporal problem and find earliest and latest start times for each operation.
2. Use a chronological backtrack search procedure to find a solution that satisfies all the resource constraints. Use constraint-based analysis to detect orderings that are ruled out (corresponding to cases 1, 2, and 3 in the notes), and use the *temporal slack*-based ordering heuristics for cases where either ordering is allowed (case 4 in the notes). Evaluate both the *Slack* and *Bslack* heuristics.
3. Include a discussion of what you would need to do to use some sort of dependency-directed backtracking. You are not required to implement any dependency-directed backtracking scheme, but you are free to do so.
4. Use constraint propagation at each search node (rather than redoing *bellman-ford* each time). You may do constraint propagation either in the distance graph or in the *constraint graph*.

**Problems for evaluation:** The problems for evaluation will be the Sadeh test suite. They can be retrieved from class Coursework site. There are 60 distinct problems in this set. Each problem has 50 operations, numbered from 0-49, and 5 resources, numbered 0-4. Allocate at most 10 minutes to solve each problem. The files contain constraint specifications of the following form:

| | |
|---|---|
| `length(0) = 10` | Duration of task 0 is 10. |
| `needs(16,1)` | Operation 16 uses resource 1. |
| `before(49,9)` | Operation 49 must be before operation 9. |
| `release(17) = 0` | Operation 17 can start at or after time 0. |
| `due(0) = 149` | Operation 0 must be done no later than time 149. |

**Report:** Your report should contain descriptions of the algorithms you are evaluating, the optimizations you may have used to make the algorithms run fast, and how you might augment your algorithms to use dependency-directed backtracking. The report should contain the results of running the experiments and a discussion of your conclusions. Assignments will be graded on the description of the algorithms and optimizations used, the raw results, and the analysis of your results.

**Submission:** The report can be submitted electronically, in class, or directly to the TA. Submit source code electronically as a single .tgz or .zip file that unpacks into its own directory. Please include a small README file describing how to build and run your code. Clearly identify all members of the group both on the report, and in the electronic submission. Send electronic submissions to cs227-submit@lists.stanford.edu.

Assignments may be done in groups of 2-3 students. You may choose any programming language for implementation purposes, though we recommend either C or C++ for maximum efficiency.

Assignments are due by noon on **21 May.**