NICTA Victoria Laboratories
Department of Computer Science and Software Engineering
The University of Melbourne
433-637 Constraint Programming
Second Semester, 2010

Search

Search straegies

Consider the following variation of the job-shop scheduling problem with the Boolean variables t that control which task is before which, exposed.

Experiment with different searching strategies to solve this. Try to find the optimal solution in the least choices possible, or failing to find the optimal solution try to find the best solution within 100000 fails. Find a robust strategy for all examples.

You should experiment with

- Variables to label: s or b
- Variables order for labelling
- Variable selection strategy
- Value selection strategy
- Restarts/limits
- Autonomous search style strategies
- Perhaps, precomputation of bottlenecks in the problem or strategies that change depending on input data.

jobshop.mzn

```
int: jobs;                                    % no of jobs
int: tasks;                                   % no of tasks per job
array [1..jobs,1..tasks] of int: d;           % task durations
int: total = sum(i in 1..jobs, j in 1..tasks)
             (d[i,j]);                        % total duration

array [1..jobs,1..tasks] of var 0..total: s;  % start times
var 0..total: end;                            % total end time
array [1..tasks, 1..jobs, 1..jobs] of var bool: b; % relative orders
                  % b[j,i,k] holds if task[i,j] is before task[k,j]

constraint forall(i in 1..jobs) (
               forall(j in 1..tasks-1)
                   (s[i,j] + d[i,j] <= s[i,j+1]) /\
               s[i,tasks] + d[i,tasks] <= end       );
constraint forall(j in 1..tasks) (
               forall(i,k in 1..jobs where i < k) (
           (b[j,i,k] -> s[i,j] + d[i,j] <= s[k,j]) /\
                   ((not b[j,i,k]) -> s[k,j] + d[k,j] <= s[i,j])  ) );
constraint forall(j in 1..tasks) ( % unused Booleans are false
               forall(i,k in 1..jobs where i >= k)(b[j,i,k] = false) );
```

```
ann: search;
solve :: search minimize end;

output ["end = ", show(end), "\n"] ++
       [ show(s[i,j]) ++ " " ++
         if j == tasks then "\n" else "" endif |
         i in 1..jobs, j in 1..tasks ];
```

Some sample data files are

jobshop.dzn

```
jobs = 5;
tasks = 5;
d = [| 1, 4, 5, 3, 6
     | 3, 2, 7, 1, 2
     | 4, 4, 4, 4, 4
     | 1, 1, 1, 6, 8
     | 7, 3, 2, 2, 1 |];
```

jobshopi.dzn

```
jobs = 5;
tasks = 5;
d = [| 1, 3, 4, 1, 7
     | 4, 2, 4, 1, 3
     | 5, 7, 4, 1, 2
     | 3, 1, 4, 6, 2
     | 6, 2, 4, 8, 1 |];
```

jobshop4.dzn

```
jobs = 4;
tasks = 4;
d = [| 1, 4, 5, 6
     | 3, 7, 1, 2
     | 4, 4, 4, 4
     | 7, 2, 2, 1 |];
```

jobshop6.dzn

```
jobs = 6;
tasks = 6;
d = [| 1, 4, 5, 3, 6, 3
     | 3, 2, 7, 1, 2, 4
     | 4, 7, 2, 6, 1, 4
     | 4, 4, 4, 4, 4, 4
     | 1, 1, 1, 6, 8, 3
     | 7, 3, 2, 2, 1, 5 |];
```

jobshop-taskdom.dzn

```
jobs = 5;
tasks = 5;
d = [| 1, 4, 5, 3, 6
     | 3, 2, 7, 1, 2
     | 4, 4, 8, 4, 4
     | 1, 1, 6, 6, 8
     | 7, 3, 8, 2, 1 |];
```

jobshop-jobdom.dzn

```
jobs = 5;
tasks = 5;
d = [| 1, 4, 5, 3, 6
     | 3, 2, 7, 1, 2
     | 6, 9, 7, 8, 9
     | 1, 1, 1, 6, 8
     | 7, 3, 2, 2, 1 |];
```

jobshop5-10.dzn

```
jobs = 5;
tasks = 10;
d = [| 8, 5, 1, 4, 5, 3, 6, 3, 6, 3
     | 3, 2, 7, 1, 2, 4, 7, 8, 5, 6
     | 4, 7, 2, 9, 3, 7, 8, 6, 1, 4
     | 4, 4, 4, 4, 4, 4, 4, 4, 4, 4
     | 1, 7, 1, 6, 1, 6, 8, 8, 3, 6 |];
```

The files are all available from the resources page. Optimal schedules for the data files are: `jobshop.dzn`: 30, `jobshopi.dzn`: 33, `jobshop4.dzn`: 23, `jobshop6.dzn`: 41, `jobshop-taskdom.dzn`: 39, `jobshop-jobdom.dzn`: 44 `jobshop5-10.dzn`: 68