Information about Problem Instances

General Information

In this directory we store vehicle routing/job shop scheduling (VRP/JSP) problem instances that we generated as part of the PRAS project. The archive files contain the problem instances.

The ideology of the big problem class reformulation study between vehicle routing and shop scheduling is this. By hypothesis, we identify key problem parameters that may be responsible for noticeable difference in performance of the same standard algorithm applied to a VRP and a job shop instance separately. Then we generate sets of mixed VRP-JSP problems so that in each set the key parameters are fixed and the number of instances in a set is constant (100). We start from geterating sets of pure VRP and JSP problems. Then we vary one key parameter at a time gradually transforming VRPs into JSPs and vice versa. Each time we vary a parameter, we generate a new set of instances.

The files whose names start with 'Direct' correspond to the VRP → JSP reformulation study and contain VRP instances (pure and transformed).

The files whose names start with 'Symmetric' correspond to the JSP → VRP symmetric study and contain JSP instances (again, pure and transformed).

The following section has details about the structure of each data file.

All VRPs have 1 depot, 100 customers and 25 vehicles. Each visit has to be done within a time window. All JSPs are of size 10 x 10.

Problem Specification

Our VRP-JSP problem instances have a unified format that allows us to represent the whole spectrum of problems with varied parameters across the VRP and JSP domains. The specification is displayed below.

```
1.  # instance filename
2.  # generation parameters (random generator seeds etc.)
    ...
3.  OptCriterion
4.  n m
5.  V
6.  Origin Horizon
7.  XDepot YDepot
8.  id Dur Demand PostCode1 PostCode2 EST LFT X Y res1 res2 .. -1
    ...
9.  0    TT01 ... TT0n
    ...  ...  ... ...
    TT10 0    ... TTin
    ...  ...  ... ...
    TTn0 TTn1 ... TTnn
```

```
10. resourceID Capacity
11. ...
12. Diff nDiff
    ...
13. Same nSame
    ...
14. Precedence nPrecedence
    ...
```

The specification has several sections. In the first section, comments, each line starts with a hash symbol ('#'). Here we record the instance name and problem generator parameters that were used to produce an instance.

In the general information section (lines 3 - 7) we specify the optimization criterion, the number of resources (vehicle in the VRPs and machines in the JSPs) n and the number of operations m (customer visits or activities). The optimization criterion 'VRP' corresponds to the minimization of total travel time. The optimization criterion 'JSP' means the minimization of makespan.

Parameter V (line 5) is the speed of vehicles for the VRPs and speed of transitions between activities in the JSPs. Line 6 shows the start and end time of the working day. Line 7 specifies the x,y-coordinates of the depot in the VRPs (0's for the JSPs).

The operations section (starting with line 8) describes each operation in a separate line as follows: visit ID, duration in minutes, demand for resource capacity (in all our instances it equals 1), alphanumeric postal codes of the respective customer (if it is a JSP, we ignore it), earliest start and latest finish times, x,y-coordinates of the customer (0's for JSPs) and finally the list of resources allowed to perform this operation followed by '- 1'.

In the transition time matrix section (starting with line 9) we explicitly display the $n+1 \times n+1$ transition time matrix with the first row corresponding to the depot. Clearly, for the pure JSPs every element of this matrix is zero. Even though an explicit transition time matrix is redundant for pure VRPs because we can always compute distances having coordinates of locations; we need it to be able to represent non-pure VRPs with transition times independent of distance as well as JSPs with (zero) transition times.

In the resources section (line 10) we list all the resources with their capacities measured as the number of operations they can perform.

Finally, lines 12 - 14 belong to the side constraints section. Each side constraint occupies a separate line and involves a pair of operations $\{o_i, o_j\}$ and states that the operations must be performed by different resources ($Diff$ side constraints), by the same resource ($Same$ constraints) or says that the operation o_i should be completed before o_j starts on the same or different resource. $nDiff$, $nSame$ and $nPrecedence$ show how many constraints of each type we have.