Job Shop Problem Generation

Kevin Sim, Emma Hart (k.sim, e.hart (at) napier.ac.uk)
Edinburgh Napier University, UK

Two sets of problems are generated comprising of 500 and 200 instances that can be used with two different objective fitness functions ($\sum_{j=1}^{n} \omega_j T_j$ and C_{max}) effectively making 1400 problems in total.

Problems are generated using all combinations of machines m and jobs n where $(m) \in [5, 10, 15, 20, 25]$ and $n \in [10, 25, 50, 100]$. For each of the 20 parameter combinations 25 problems are generated for the training set and 10 for the test set. Each problem is considered using both fitness metrics, doubling the number of training and test instances.

The processing time for an operation is selected randomly from a uniform distribution following [3] using Equation 1.

$$p_{i,j,k} = U[m/2, m2] \tag{1}$$

The release dates are drawn randomly from one of two distributions as in [3] using Equation 2 depending on the number of jobs in the problem instance.

$$r_i = \begin{cases} U[0,20] & \text{if } n < 50 \\ U[0,40] & \text{if } n \geq 50 \end{cases} \tag{2}$$

Due dates are defined as in [1] using Equation 3. The term c is fixed at 1.3, the average of the values used in [2] which investigated the difficulty of relatively "tight" problems generate with $c = 1.4$ and $c = 1.2$.

$$d_i = r_i + c \times \sum_{j=1}^{n} p_{ij} \tag{3}$$

Job weights are selected using the 4:2:1 rule taken from [4] which is informed by research suggesting that 20% of a company's customers are the most important, 60 % are of of average importance and 20% are less important. Consequently the first 20% of jobs in an instance are assigned a weight of 4, the next 60% receive a weight of 2 and the remaining 20% of an instances jobs are given a weight of 1.

Acknowledgements

Funded by **EPSRC: EP/J021628/1** and supported by Edinburgh Napier University's Institute for Informatics and Digital Innovation (IIDI)

References

[1] K. R. Baker. Sequencing rules and due-date assignments in a job shop. *Management Science*, 30(9):1093–1104, 1984.

[2] M. Singer and M. Pinedo. A computational study of branch and bound techniques for minimizing the total weighted tardiness in job shops. *IIE Transactions*, 30(2):109–118, 1998.

[3] J. C. Tay and N. B. Ho. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers & Industrial Engineering*, 54(3):453–473, 2008.

[4] H. Zhou, W. Cheung, and L. C. Leung. Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm. *European Journal of Operational Research*, 194(3):637 – 649, 2009.