

MODELS FOR ADJACENT RESOURCE SCHEDULING

C.W. Duin and H.J. van der Sluis

Operations Research Group

Faculty of Economics and Econometrics

University of Amsterdam

Abstract:

We will formulate and compare models for an interesting relatively new problem, that of adjacent resource scheduling.

As an example, consider an airport where passengers check in for their flight, joining waiting lines before one or more desks; at the desk the luggage is checked and so forth. To smooth these operations the airport maintains a clear order in the waiting lines: a number $n(j)$ of *adjacent* desks is to be assigned exclusively during a fixed time-interval $I(j)$ to flight j . For each flight in a given time horizon, one seeks a feasible assignment to adjacent desks and the objective is to minimize the total number of involved desks.

We can formulate the problem more formally. A ‘bank’ of adjacent resource units $r=1,2,\dots,R$ is to handle jobs $j=1,2,\dots,J$ with starting period $a(j)$ and completion period $b(j)$ within a planning horizon with periods $t=1,2,\dots,T$. Each resource unit can

be assigned to at most one job per period. During an interval $I(j)=[a(j),b(j)]$ (of $b(j)-a(j)+1$ time periods in $[1,T]$), each job j is to be assigned an invariant interval of $n(j)$ resource units within $[1,R]$. The problem is to find the minimal number of resources R needed to complete all jobs.

On one hand, this problem can be compared to two-dimensional strip-packing where shifting is allowed in only one dimension, on the other hand it is like a reservation problem with additional adjacency constraints. In a more general problem variant the number $n(j)$ varies over the time interval.

Firstly, we present two integer linear programs for these problems: a disjunctive model with r_{jt} representing the highest resource index for job j in period t as decision variable and an all-binary model with variables y_{jtr} equal to 1 if r is the highest resource index used by job j in period t . These models are trimmed to respectively a polynomial size and a pseudo-polynomial size. The latter is extended with a set of valid inequalities, and can then solve instances of moderate size.

Using the all-binary model, instances with 96 jobs and a planning horizon of 48 periods and constant $n(j)$ are solved within a few minutes on a 400 MHz Pentium II, while the disjunctive model was able to solve only a few instances within reasonable time.

Secondly, we apply Constraint Programming (CP) and experiment with different variable selection and value assignment strategies. A straightforward disjunctive CP-model either solves instances with constant $n(j)$ within seconds or not at all.

Finally, we compare the results of these different approaches for instances with $n(j)$ varying over the time interval $I(j)$.

Key words: integer programming, constraint programming