Read Me document for datasets for Task-Based formulation of Vessel Crew Scheduling problem

Alex Leggate

November 12, 2015

This document explains the accompanying datasets, which have been used to produce the test results given in our paper "Models and Formulations for Crew Scheduling in Off-Shore Supply Vessels". There are 240 datasets, each using a different combination of the generating parameters which were used.

1 Parameters used

Each dataset contains detail of which parameters were used in the data generation process (described in the paper). The six parameters given are:

- `Probability employee ill given previously fit`: this is probability q as described in the paper.
- `Probability employee ill given previously ill`: this is probability p as described in the paper.
- `Time-reduction of probabilities`: this is indicates whether the time-reduction factor $r(d)$ was used to generate the data (YES/NO).
- `Disruption factor for near tasks`: This is the value of factor K_N, used to modify costs within the first four weeks of the planning period.
- `Disruption factor for more distant tasks`: This is the value of factor K_L, used to modify costs occurring in the longer term.
- `Agency crew penalty`: This is the value of factor K_{AG}, used to modify the costs of agency crew.

2 Data

Each data file contains the same data items:

- `Max_OF` is the initial value of the cost limit Λ, used in the change-minimization problem.
- `Max_Sol` is a parameter which could be used for setting an iteration limit in the change-minimization problem.
- `DAYS_TO_PLAN` is the length T of the planning period, in days.
- `JOBS_TO_PLAN` is the size of the task set J, equal to $|J| = n_W$.
- `LINKED_SETS` is a now rejected experimental parameter, set equal to zero in all datasets.
- `PROJECT_NUMBER` is the size of project set K.
- `EMP` gives the contents of the regular employee set E_R.
- `GUARANTEED` gives the contents of the set of fixed contract employees G.
- `start_time` gives the starting time s_j of each task $j \in J$.
- `duration` gives the duration d_j of each task $j \in J$.
- `linked_matrix` indicates membership of the experimental 'linked' sets; as this was rejected, this is simply a vector of zeros for all datasets.
- `min_rest` gives the minimum required duration of a rest period (in *weeks*) ρ_i for each employee $i \in E_R$.
- `max_work` gives the maximum permitted number of consecutive working *weeks* ω_i for each employee $i \in E_R$.
- `eligible` gives the eligibility matrix, indicating whether or not (1 or 0, respectively) employee i can perform task j. Data is given such that each row represents a different employee $i \in E_R$ and each column represents a different task $j \in J$. This data has no explicit place in the formulation of the problem; instead, it is used to determine whether an allocation decision variable z_{ij} should be defined for an employee for a given task.
- `AG_eligible` similarly gives the eligibility of agency employees $i = m + 1$ for each task $j \in J$.
- `work_resource_zero` gives the values of W_{i0}, the number of consecutive *weeks* for which employee $i \in E_R$ has worked immediately prior to the start of the planning period (i.e. at time $t = 0$).
- `rest_resource_zero` indicates whether or not (1 or 0) employee $i \in E_R$ requires to be allocated a rest period at the start of the planning period (i.e. at time $t = 0$).

- `initial` shows the allocations in the current schedule of employees $i \in E_R$ (rows) to each task $j \in J$ (columns), which is denoted as x^*_{ij} in the formulation of the problem. Here, $x^*_{ij} = 1$ if employee i is allocated to task j in the current schedule; and equals zero otherwise.

- `AG_initial` similarly shows the allocations in the current schedule of agency crew ($i = m + 1$) to each task $j \in J$, represented as $x^*_{m+1,j}$ in the formulation.

- `change_cost` gives the cost c'_{ij} of changing the assignment of employee $i \in E_R$ with respect to task $j \in J$. As with the eligibility matrix, rows represent employees while columns represent roles.

- `AG_change_cost` similarly gives the cost $c'_{m+1,j}$ of changing an agency employee's assignment with respect to each task $j \in J$.

- `under_rate` is the effective rate of pay per day μ_i for employee $i \in G$, which can be considered an *under-time* rate in the event that the employee works less than their 'guaranteed days' in the year.

- `over_rate` is daily rate ϕ_i at which overtime is paid to employee $i \in G$, in the event that the employee works more than their 'guaranteed days' in the year.

- `current_excess` is the expected additional undertime or overtime cost Ω_i for each employee $i \in G$ under the current schedule, as defined in the `initial` matrix $\mathbf{X}^*$.

- `g_days` is the number of working days g_i which are guaranteed to each employee $i \in G$. Employee i will receive pay for at least this number of days regardless of how much work they are assigned over the year; and will be paid extra if they work more than this amount.

- `exp_worktime` is the expected number of days $\bar{W}_i$ that employee $i \in G$ will work during the year *outwith* the current planning period.

- `project_matrix` indicates whether or not (1 or 0) each task $j \in J$ belongs to each project $k \in K$, which each row representing a different project. Note that if $K = \emptyset$ (i.e. if $|K| = 0$, as given by `PROJECT_NUMBER` above), then this data item will be given as a single row of zeros.

- `experience` gives the experience score e_{ij} of each employee $i \in E_R$ (rows) with respect to each task $j \in J$ (columns).

- `AG_experience` similarly gives the experience score $e_{m+1,j}$ of agency crew $i = m + 1$ with respect to each task $j \in J$.

- `min_experience` gives the minimum total experience ϵ_k required across all tasks in each project $k \in K$. Note that if $K = \emptyset$, then this is represented as a zero vector of length 1.

3 Problem formulation

The data above is designed to be applicable in both the cost-minimizing and change-minimizing recovery-type formulations of the Vessel Crew Scheudling model given in our paper. We also give these formulations here.

3.1 Additional definitions

In addition to the data provided above, the formulations also require the following:

Sets

N - set of dummy *rest* tasks; $|N| = n_R$.

$J \cup N$ - combined set of working and rest tasks; $|J \cup N| = n_W + n_R = n$.

C_γ - sets of tasks which overlap in time; $C_\gamma \subseteq J \cup N \; \forall \gamma \in \Gamma = \{1, 2, \ldots, \gamma_{max}\}$, where $0 \leq \gamma_{max} \leq \frac{1}{2}n(n+1)$. Rather than considering every combination of tasks, we can define these sets efficiently to reduce the total number of constraints (see 3.4).

B - set of ordered indices of all tasks in $J \cup N$; $s_{b-1} \leq s_b$ for $b \in B, b \geq 2$.

Parameters

w_b - *work resource* value of task b;

$$w_b = \begin{cases} d_b & \text{for } b \in J \\ -\max\limits_{\forall i \in E} \omega_i & \text{for } b \in N \end{cases}$$

r_b - *rest resource* value of task b;

$$r_b = \begin{cases} 1 & \text{for } b \in J \\ -1 & \text{for } b \in N \end{cases}$$

Decision Variables

y_{ij} - 1 if there is a change to employee i's schedule w.r.t. task j; 0 otherwise.

z_{ij} - 1 employee i is allocated to task j in the new schedule; 0 otherwise.

W_{ib} - accumulated *work resource* value for employee i once all tasks up to and including the task b have been considered.

R_{ib} - corresponding accumulated *rest resource* value.

u_i - number of days under the guaranteed days for employee $i \in G$.

o_i - number of days over the guaranteed days for employee $i \in G$.

ψ_i^u - 1 if the undertime value for employee i is non-negative; 0 otherwise (used in the change-minimization problem only).

ψ_i^o - 1 if the overtime value for employee i is non-negative; 0 otherwise (used in the change-minimization problem only).

3.2 Recovery-type formulation

Our cost-minimizing recovery-type formulation (denoted as $RF1$) is therefore:

$$\min \sum_{\forall i,j} c'_{ij} y_{ij} + \sum_{i \in G} (\mu_i u_i + \phi_i o_i - \Omega_i) \tag{1}$$

subject to:

$$\sum_{i=1}^{m+1} z_{ij} = 1 \qquad \forall j \in J \tag{2}$$

$$\sum_{j \in C_\gamma} z_{ij} \leq 1 \qquad \forall i \in \{1, \ldots, m\}, \gamma \in \Gamma \tag{3}$$

$$\sum_{i=1}^{m+1} \sum_{\forall j \in P_k} e_{ij} z_{ij} \geq \epsilon_k \qquad \forall k \in K \tag{4}$$

$$W_{i,b-1} + z_{ib} w_b \leq W_{ib} \qquad \forall b \in B, i \in E_R \tag{5}$$

$$W_{ib} \leq \omega_i \qquad \forall b \in B, i \in E_R \tag{6}$$

$$R_{i,b-1} + z_{ib} r_b \leq R_{ib} \qquad \forall b \in B, i \in E_R \tag{7}$$

$$u_i \geq g_i - \left(\bar{W}_i + \sum_{j \in J} d_j z_{ij} \right) \qquad \forall i \in G \tag{8}$$

$$o_i \geq \left(\bar{W}_i + \sum_{j \in J} d_j z_{ij} \right) - g_i \qquad \forall i \in G \tag{9}$$

$$z_{ij} = x^*_{ij} + (1 - 2x^*_{ij}) y_{ij} \qquad \forall i, j \text{ s.t. } y_{ij} \text{ is defined} \tag{10}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i, j \text{ s.t. } y_{ij} \text{ is defined} \tag{11}$$

$$R_{ib} \in \{0, 1\} \qquad \forall b \in B, i \in E_R \tag{12}$$

$$W_{ib} \geq 0 \qquad \forall b \in B, i \in E_R \tag{13}$$

$$u_i, o_i \geq 0 \qquad \forall i \in G \tag{14}$$

3.3 Alternative Recovery-type formulation

We then adapt our change-minimizing recovery-type formulation (denoted as $RF2$) from the above formulation of $RF1$. We must change the objective function to

$$\min \sum_{\forall i,j} y_{ij} \tag{15}$$

and add the old objective as a constraint:

$$\sum_{\forall i,j} c'_{ij} y_{ij} + \sum_{i \in G} (\mu_i u_i + \phi_i o_i - \Omega_i) \leq \Lambda \tag{16}$$

Then, we replace (8) with the following constraints:

$$u_i \leq g_i - \left(\bar{W}_i + \sum_{j \in J} d_j z_{ij} \right) + M\psi_i^u \quad \forall i \in G \tag{17}$$

$$u_i \leq M \left(1 - \psi_i^u\right) \quad \forall i \in G \tag{18}$$

and similarly, use the following constraints to replace (9):

$$o_i \geq \left(\bar{W}_i + \sum_{j \in J} d_j z_{ij} \right) - g_i + M\psi_i^o \quad \forall i \in G \tag{19}$$

$$o_i \leq M \left(1 - \psi_i^o\right) \quad \forall i \in G \tag{20}$$

and to ensure u_i and o_i are not both set to non-zero for the same i, we add the constraint:

$$\psi_i^u + \psi_i^o \leq 1 \quad \forall i \in G \tag{21}$$

Then, our formulation, denoted as $RF2$, can be stated as $\min_{y,R,W,u,o,\psi^u,\psi^o}$ $(\sum_{\forall i,j} y_{ij} | (y, R, W, u, o, \psi^u, \psi^o) \in X^{RF2})$ where $X^{RF2} = \{(y, R, W, u, o, \psi^u, \psi^o) |$ $(2) - (7), (10) - (14), (16) - (21), \psi^u, \psi^o \in \{0, 1\}^{|G|}\}$.

3.4 Algorithm to define sets C_γ

As stated above, it is possible to define sets of overlapping tasks such that we do not need to consider every possible overlapping pair. In order to do this, we first define an ordered list of time points $t_p, p = 0, 1, 2, \ldots, p_{max}$ such that t_0 is the starting time of the first task in J, each task start time s_j and task end time $s_j + d_j$ equate to a t_p for all $j \in J$, and $t_p > t_{p-1}$ $\forall p$. Algorithm 1 describes briefly a way to enumerate these constraints.

Algorithm 1 Algorithm defining overlapping task sets C_γ

calculate all time points t_p for the set of tasks $J \cup N$
set counters $\alpha = 0$ and $\beta = 0$
set property *constraint required* to be false, and let the calculation set $C^c = \emptyset$
while $\alpha \leq p_{max}$ **do**
  if $t_\alpha = s_j + d_j$ for some $j \in J \cup N$ **then**
    if *constraint required* is true **then**
      set $\beta = \beta + 1$
      set $C_\beta = C^c$
      reset *constraint required* to false
    end if
    for all $j \in J \cup N$ such that $t_\alpha = s_j + d_j$, remove j from set C^c
  end if
  if $t_\alpha = s_j$ for some $j \in J \cup N$ **then**
    set *constraint required* to be true
    for all $j \in J \cup N$ such that $t_\alpha = s_j$, add j to the set C^c
  end if
  if $\alpha = p_{max}$ **then**
    set $\gamma_{max} = \beta$
    set $\Gamma = \{1, 2, \ldots, \gamma_{max}\}$
  end if
  set $\alpha = \alpha + 1$
end while