A greedy MIP-based heuristic approach to the problem of scheduling technicians and interventions for telecommunications

S. Balev et O. Gaci

Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes
25 rue Ph. Lebon BP 540, 76058 Le Havre Cedex, France
{Stefan.Balev, Omar.Gaci}@univ-lehavre.fr

1 Introduction

This paper deals with the problem of scheduling technicians and interventions for telecommunications. This optimization problem is proposed by France Telecom and it is the subject of the ROADEF Challenge 2007 [1]. This is a real life scheduling problem with many constraints of various types, the most important of which are:

- team constraints – technician teams are formed each day and should not change during the day;
- competence constraints – each intervention has competence requirements and can only be done by a sufficiently competent team;
- precedence constraints – an intervention cannot start before its predecessors are finished.

At first glance the problem looks like a multi-dimensional bin packing problem, but it is much more difficult because of the above mentioned constraints and the classical bin packing algorithms are not applicable to it.

This paper presents a heuristic scheduling algorithm based on iterative solving of mixed integer programming (MIP) models [2]. Each model forms a team and assigns interventions to it. Due to a limited number of pages, we do not give a formal description of the problem, the reader is referred to [3] for such description. In this paper we use the same notations as in [3]. The only difference is that we use i to index the interventions and d to index the domains (and not I and i as in [3]).

2 Scheduling algorithm

Theoretically, it is possible to formulate a MIP model which constructs a schedule of minimal score subject to the problem's constraints. But this kind of model will contain so many variables and constraints that it will not be solvable even for problem instances of small size. That is why we chose to construct the schedule day by day. Each day we select a subset of interventions that are not scheduled in the previous days and that can be scheduled using the available technicians for the day. This local vision has obvious disadvantages, but at least the problems that we need to solve for each day are easier.

Our first approach was to formulate a MIP model for each day. This model constructs teams using the available technicians for the day and assigns to each team interventions. It maximizes the total cost of the scheduled interventions. The cost of each intervention depends on its priority, its execution time and its competence requirements. The iterative solving of this model works well for the small instances of set A, but for the bigger instances from set B the size of the model becomes enormous and it cannot produce solutions in reasonable time.

That is why we decided to go one more level down and to construct the schedule for each day team by team. Once again we use a MIP model to select a subset of available technicians and to assign interventions to this subset. This sacrifice is payed by the fact that the obtained MIP models are very simple. In this way we can construct a complete schedule very quickly (time of order 0.01 s for the smallest instances up to 2-3 min for the biggest ones on the test machine). Introducing a random factor, we can construct many different schedules and retain the best of them.

2.1 Team scheduling model

Suppose that we have constructed a schedule for days $1, \ldots, j-1$ and that some teams are already formed on day j. Now we want to form the next team for day j and assign interventions to it. Let $\mathcal{T}$ be the set of available technicians (working on day j and not participating in the previously formed teams for this day). Let $\mathcal{I}$ be the set of available interventions (not hired and not scheduled until now). We also suppose that $\mathcal{I}$ contains only interventions that can be done by the technicians in $\mathcal{T}$. If $\mathcal{I} = \emptyset$, nothing more can be done on this day and we can skip to day $j+1$. That is why we consider the case $\mathcal{I} \neq \emptyset$. Let $\mathcal{T}(d,n) = \{t \in \mathcal{T} \ : \ C(t,d) \geq n\}$ be the set of available technicians of level at least n in domain d.

We introduce binary variables x_t, $t \in \mathcal{T}$ and y_i, $i \in \mathcal{I}$. The variable x_t is 1 iff technician t is selected in the next team and y_i is 1 iff intervention i is assigned to this team. The new team is formed according to the solution of the following model.

$$\text{Minimize} \quad \sum_{t \in \mathcal{T}} \alpha_t x_t - \sum_{i \in \mathcal{I}} \beta_i y_i \tag{1}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} T(i) y_i \leq 120 \tag{2}$$

$$\sum_{t \in \mathcal{T}(d,n)} x_t \geq R(i,d,n) y_i \qquad \forall i \in \mathcal{I}, d, n \tag{3}$$

$$x_t \in \{0,1\} \qquad \forall t \in \mathcal{T} \tag{4}$$

$$y_i \in \{0,1\} \qquad \forall i \in \mathcal{I} \tag{5}$$

The constraint (2) says that the total execution time of the selected interventions must not exceed the working time of the team. The constraints (3) ensure that the team has at least the required number of technicians in each competence domain and level for each selected intervention.

The goal of the objective function (1) is to minimize the idle resources in the team. The first term represents the resources used and the second one the work done. There are different ways to choose the coefficients α_t and β_i. This choice is crucial since it determines the quality of the obtained solution.

For example, let $\alpha_t = 120$ and $\beta_i = T(i) \times H(i)$, where $H(i) = \max_d\{R(i,d,1)\}$ is a lower bound on the number of technicians needed to do intervention i. Figure 1 explains the meaning of this objective function. Consider the second team composed by technicians 2, 3, 4 and 5. The area of the team's rectangle is $120 \times 4 = 480$. The team executes interventions 18, 19, 16, 11 and 5. Each intervention is represented by a gray rectangle of width $T(i)$ and height $H(i)$. The total area of the rectangles of these interventions is $60 \times 4 + 15 \times 4 + 15 \times 1 + 15 \times 1 + 15 \times 4 = 390$. The objective function seeks to minimize the empty (white) area in the team's rectangle. For the second team this area is $480 - 390 = 90$, while for the first and the third teams it is 0.

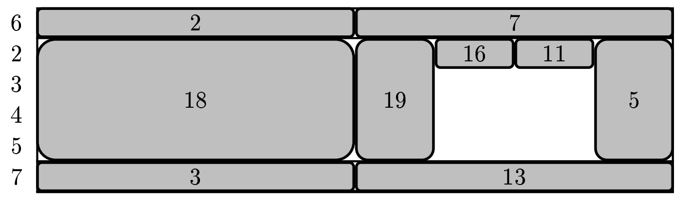

Fig. 1. Example of scheduling of day 1 for instance 3 from set A

Of course in this example the objective coefficients are too simple to work well. We have to multiply the β_i coefficients by weight factors depending on the priority of each intervention. In this way the interventions of higher priority will be selected earlier. Furthermore, the model may select a team which is too competent for the selected interventions and leave less competent technicians for the next teams in the same day. After several experiments we retained the following objective coefficients:

$$\alpha_t = 120 \times \left(1 + \frac{\sum_d C(t,d) - 1}{D \times N - 1}\right) \times \max_{i \in \mathcal{I}}\{\text{PrioCoef}(i)\}, \quad \beta_i = T(i) \times H(i) \times \text{PrioCoef}(i) \quad (6)$$

where $\text{PrioCoef}(i) = 10^{6-2\times\text{Prio}(i)}$, D is the number of domains and N is the number of levels. In this way an intervention of priority p is 100 times more expensive than the same intervention of priority $p+1$ and a technician having full competences in all domains costs twice as much as a technician having level 1 in one domain and level 0 in all other domains. These coefficients seem to work better than the other variants we tested, but a more detailed experimental study is needed in order to determine appropriate values of α_t and β_i. A finer estimation of the costs taking into account the competences might give better results.

A small fault of our model is that it might have a trivial zero solution. To avoid this trivial solution we choose a random intervention $i^* \in \mathcal{I}$ among these of highest priority and fix the y_{i^*} variable to one. The side effect of this random choice is that we can generate different schedules and retain the best of them.

2.2 Assigning starting times to the interventions

The iterative solving of the MIP model from the previous section produces teams for day j and affects interventions to each team. Now we have to determine the starting time of each intervention scheduled on day j. We use the following MIP model to find the starting times:

$$\begin{aligned}
\text{Minimize} \quad & 28\tau_1 + 14\tau_2 + 4\tau_3 + \tau_4 && && (7)\\
\text{subject to} \quad & T(k) \le s_i - s_k + 120 z_{ik} \le 120 - T(i) && \forall i,k \;:\; d(i) = d(k) = j,\ a(i) = a(k) && (8)\\
& s_k + T(k) \le s_i && \forall i,k \;:\; d(i) = d(k) = j,\ k \in \text{Pred}(i) && (9)\\
& s_i + T(i) \le \tau_{\text{Prio}(i)} && \forall i \;:\; d(i) = j && (10)\\
& \tau_p \le \tau_4 && p = 1,2,3 && (11)\\
& 0 \le s_i \le 120 - T(i) && \forall i \;:\; d(i) = j && (12)\\
& z_{ik} \in \{0,1\} && \forall i,k \;:\; d(i) = d(k) = j,\ a(i) = a(k) && (13)
\end{aligned}$$

In this model the continuous variable s_i is the start time of intervention i. Constraints (8) ensure that interventions affected to the same team do not overlap. When the binary switch variable z_{ik} iz zero, (8) becomes $s_k + T(k) \le s_i$ and when $z_{ik} = 1$ it transforms to $s_i + T(i) \le s_k$. Constraints (9) ensure that i is not started before its predecessors are finished. The continuous variables τ_p, $p = 1,2,3$ represent the end time of the last intervention of priority p and τ_4 is the end time of the last scheduled intervention for day j. Of course τ_p has some meaning only if there are no more unscheduled interventions of priority p. Otherwise we can set the objective function coefficient of τ_p to zero.

2.3 Precedence constraints

The method described in the previous sections generates schedules that may violate some precedence constraints. In this section we describe a procedure that checks and repairs such violations. This procedure is applied at the end of each day and ensures that the obtained schedule respects the precedence constraints.

We define the chain length of an intervention in the following way

$$\text{ChainLen}(i) = \begin{cases} 0 & \text{if } i \text{ is scheduled before day } j \\ T(i) + \max\{\text{ChainLen}(k) \;:\; k \in \text{Pred}(i)\} & \text{otherwise} \end{cases} \quad (14)$$

It is clear that if $\text{ChainLen}(i) > 120$ then i cannot be scheduled on day j. We can use this simple test to exclude from the set $\mathcal{I}$ all the interventions with too long chains of unscheduled predecessors before starting the scheduling of day j. Another simple thing to do is to encourage the selection of interventions with successors by the model (1)-(5). We do this by multiplying β_i by two for all i having successors of the same priority. Although these precautions, some precedence constraints may still be violated after constructing the schedule for day j. The possible problems are:

– Some intervention i is scheduled on day j but some $k \in \text{Pred}(i)$ is not scheduled yet.
– All interventions scheduled on day j have their predecessors scheduled, but some of their predecessors are also scheduled on day j and the problem (7)-(13) is infeasible.

A radical way to avoid these problems is to exclude from the set $\mathcal{I}$ the interventions that have unscheduled predecessors. This means that two interventions in precedence relation cannot be scheduled on the same day. This solution is too restrictive and produces quite sparse schedules. Instead of it we use the following iterative algorithm:

1. Exclude from $\mathcal{I}$ the interventions with chain length more than 120. Multiply by 2 the β-coefficients of the interventions that have unscheduled predecessors of the same priority.
2. Construct a schedule for day j by iteratively solving the problem (1)-(5).
3. If there are interventions scheduled on day j with unscheduled predecessors, pick one of them randomly, exclude it from $\mathcal{I}$ and go to step 2.
4. Solve the problem (7)-(13). If it is feasible, stop.
5. Pick random intervention scheduled on day j which have predecessor scheduled on day j. Exclude it from $\mathcal{I}$. Go to step 2.

This approach slows down the scheduling algorithm because the same day may be scheduled many times, but it is less restrictive than the obvious solution of excluding all the interventions with unscheduled predecessors.

2.4 Hired interventions

Some interventions can be subcontracted to external companies. The constraint to respect is that the total cost of such hired interventions must not exceed the abandon budget A. In our algorithm the choice of hired interventions is as simple as solving the following knapsack problem:

$$\text{Maximize} \quad \sum_i \beta_i y_i \tag{15}$$

$$\text{subject to} \quad \sum_i a_i y_i \leq A \tag{16}$$

$$y_i \in \{0,1\} \qquad \forall i \tag{17}$$

where a_i is the abandon cost of intervention i and the coefficients β_i are the same as in (6). The binary variable y_i is one iff the intervention i is abandoned. Only the interventions that have no successors are considered. This knapsack problem is solved before the scheduling algorithm starts. The selected interventions are supposed scheduled on day -1 and are not considered further. We have chosen this approach because it is easy to implement but this is not at all the best thing to do. A more intelligent approach would consist in dynamically selecting the interventions to hire in the day when the last intervention of priority p is scheduled ($p = 1, \ldots, 4$).

3 Experimental results

Our algorithm is implemented in C. To solve the MIP models we use CPLEX 10 callable library [4]. The experiments presented in this section were executed on the test machine provided by France Telecom and ILOG. We used the test instances provided by the organizers of the ROADEF Challenge. For each instance we executed 10 runs. The time of each run was limited to 20 minutes. Note that this is real time (not CPU time) and the machine was available to all participants in the challenge during the experiments. Table 1 presents summary results of these runs. For each instance we give the best and the worst score obtained, as well as the average and the standard deviation of the score over the 10 runs.

For comparison we also give the reference scores provided by the organizers. We can see that in 18 out of 20 cases even the worst score obtained by our program is better than the reference score. Of course, the reference scores are just examples of valid scores, not of "good" scores, but for the moment this is the only base of comparison we have.

We can also observe the robust behavior of our algorithm. Although this is a randomized algorithm, the results obtained at different runs are close (see the standard deviation line). For

Table 1. Summary results of 10 runs (the time of each run is 20 min)

set A	1	2	3	4	5	6	7	8	9	10
ref	2490	4755	15840	14880	41220	30090	38580	26820	35600	51720
min	2625	4755	13560	13620	33480	21915	32220	20100	28740	40140
max	2625	4755	13560	13620	33480	24255	32220	22020	30420	40140
avg	2625	4755	13560	13620	33480	23397	32220	20790	29513	40140
std.dev	0	0	0	0	0	824	0	569	711	0
set B	1	2	3	4	5	6	7	8	9	10
ref	69960	34065	34095	50340	150360	47595	56940	51720	44640	61560
min	44160	18180	19245	43905	119340	32850	40260	37920	33360	42960
max	45600	21555	20700	69105	123240	35115	41700	40080	35040	46320
avg	44904	19827	20199	52907	121332	33513	40944	39372	33996	44544
stddev	568	1275	548	9492	1343	897	609	1007	614	1185

7 instances, we always obtain the same result. We explain this robustness by the fact that our algorithm generates many schedules per run and gives the best of them as solution. The only exception is instance 4 from data set B. The big standard deviation is due to 3 outliers: 57210 and 69105 (2 occurrences). A possible explication of this fact is that the test machine was used by other participants during these runs.

4 Conclusion

This paper presents a step by step heuristic algorithm for the problem of scheduling technicians and interventions for telecommunications. At each step we form a team using the available technicians and assign interventions to this team. Our algorithm does not seek to minimize directly the score function. Instead, we try to keep the technicians as busy as possible, hoping that in this way the interventions will be done earlier. The idea of our algorithm is well resumed by the proverb "Never put off till tomorrow what can be done today". The main disadvantage of this greedy approach is that bad choices in the earlier days may lead to idle resources in the later days. The advantage is that the schedules are constructed quickly and with random choices, so we can generate several schedules and choose the best of them.

This paper presents a preliminary version of our work. There are a lot of things to be improved in our algorithm. A better choice of the objective coefficients of the team scheduling model can be done. The precedence constraints can be managed in better way in order to avoid rescheduling of the same day and to accelerate the algorithm. Some improvement procedures can be added for the days containing the last intervention of each priority level. Hired interventions may be selected dynamically, during the scheduling and not before it, etc.

We would like to thank the organizers of the ROADEF Challenge 2007 for proposing this interesting and exciting industrial optimization problem. We have been working on it with a lot of pleasure. Special thanks to Van-Dat Cung and Anne-Marie Bustos for the technical help and for answering our (sometimes stupid) questions.

References

1. Challenge ROADEF 2007 http://gilco.inpg.fr/ChallengeROADEF2007/.
2. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley (1988)
3. Dutot, P.-F., Laugier, A., Bustos, A.-M.: Technicians and interventions scheduling for telecommunications (2006) http://gilco.inpg.fr/ChallengeROADEF2007/en/sujet/sujet2.pdf.
4. ILOG: ILOG CPLEX 10.0 User's Manual. (2006)