Lower bounds for the Multi-Skill Project Scheduling Problem with hierarchical levels of skills

Odile Bellenguez
Emmanuel Néron
Laboratoire d'Informatique de l'Université de Tours,
64 av Jean Portalis,
37200 Tours, France
e-mail: odile.bellenguez@etu.univ-tours.fr
e-mail: neron@univ-tours.fr

In a Multi-skill Project Scheduling Problem (MSPSP), there is a project, composed of activities, to realize. Each activity has to be done by a subset of staff members, chosen for their abilities to satisfy the needs of activities. In our problem, each person has different skill(s), and a level of ability for each of his skills. It is particularly adapted to modelize what happen in a software development society, where employees could be analyst, programmer, DB architect and so on... The goal is to minimize the project duration. This paper introduces in details this problem, and lower bounds, usefull to evaluate the minimum duration of a project.

1 Project scheduling with hierarchical levels of skills

A project, composed of n activities, has to be to scheduled in a minimum time. There are classical precedence relationships between the activities, so an activity cannot start before the end of all its predecessors. Precedence constraints define G(X,U), an activity-on-node graph. To be processed, those activities need some resources having specific skill. In our case resources are staff members. For every person, we know his unavailability period(s) and his level of ability, like in [6]. For every skill, a person have only one level, thus he can performed activities requiering this skill at a level lower than or equal to his level of ability. His level can be equal to 0, if he does not master the skill. Let us define:

- A_i, $i \in \{0..n\}$, the set of activities. A_0 and A_n are dummy activities, which modelize the start and the end of the project,
- S_k^l, $k \in \{0..K\}$, $l \in \{0..L\}$, the set of skills. k is the number of the skill and l the level,

- $b^l_{i,k}$, $i \in \{0..n\}$, $k \in \{0..K\}$, $l \in \{0..L_{max}\}$, the number of persons, able to do S_k at a level at least equal to l, requiered to execute A_i,

- P_m, $m \in \{0..M\}$, the staff members,

- $A(P_m, t)$, $m \in \{0..M\}$, $t \in \{0..T_{max}\}$, equals to 1 if P_m is available at time t, 0 otherwise,

- $A(P_m, t_1, t_2) = \sum_{t=t_1}^{t_2-1} A(P_m, t)$, $m \in \{0..M\}$, the total time P_m is available between t_1 and t_2

- $S_{m,k} = l$, $m \in \{0..M\}$, $k \in \{0..K\}$, the level l at which P_m is able to do S_k. To simplify the model, we will use the notation $S^{l'}_{m,k} = 1, \forall l' \leq l$, if P_m is able to do S_k at level l or lower, $S^{l'}_{m,k} = 0$ otherwise.

For each activity, we use the precedence graph to compute its release date r_i, and thus r_n is a first lower bound of the project duration (critical path). Then, if we fix D, a deadline of the project, we can compute the deadline $\tilde{d}_i(D)$ of each activity. If we are able to prove that at least one activity cannot be processed within its time window $[r_i, \tilde{d}_i(D)]$ then D+1 is a valid lower bound of the project duration. Solving this problem is clearly $\mathcal{NP}$-Hard, as the well-known Resource Constrained Project Scheduling Problem (RCPSP) can be seen as a special case of the MSPSP with hierarchical level.

2 Graph of compatibility

We assume that D is fixed and $\tilde{d}_i(D)$ are computed, and we try to prove that the project duration cannot be smaller than D. The first way to evaluate our project duration is based on anti-chains, like in [4]. So, we create a graph G' = (X,U') where nodes are activities, and we link two activities A_i and A_j, (i,j) $\in$ U', if they can be in progress at the same time. To make two activities be in progress at the same time, there must exist a not empty intersection between $[r_i, \tilde{d}_i]$ and $[r_j, \tilde{d}_j]$, and at least one feasible subset of staff members available on $[r_i, \tilde{d}_i] \bigcap [r_j, \tilde{d}_j]$ which is able to satisfy all the needs of A_i and A_j simultaneously. This last item is verified by solving the corresponding assignment problem, which can be solved using a max-flow formulation presented in the figure 1 (see [5] for more details).

We search for a maximum-weighted independant set in G'(X,U'), where weights are durations of activities, because this set is a maximum set of activities that cannot be in progress at the same time, so its weight provides a lower bound of the project duration. But finding the maximum independant set is a $\mathcal{NP}$-Hard problem. Two lower bounds have been tested:

- Heuristic: greedy algorithm that takes all the nodes of the critical path, and then add successively all the independant maximum-weighted nodes, until we cannot add any node.

- Exact: MIP formulation solved using CPLEX 8.0.

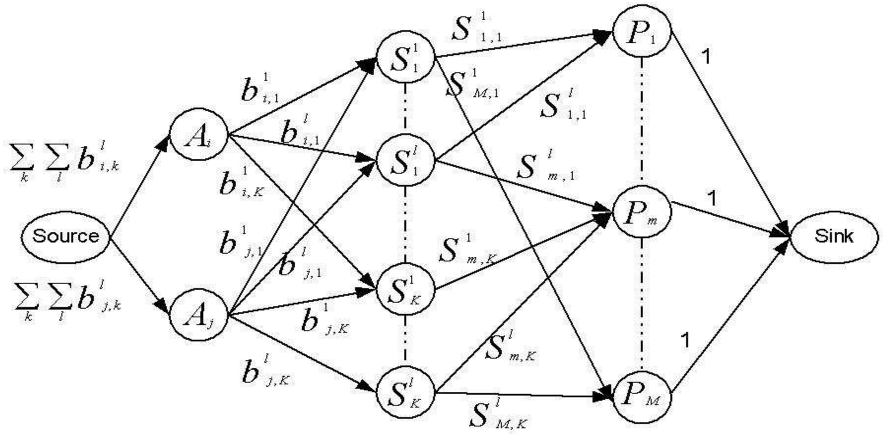

Figure 1: Needs of A_i and A_j to satisfy if they can be in progress at the same time

3 Energetic reasoning based lower bound

The third lower bound we propose is based on energetic reasoning, inspired from [1], [3]. We assume that D is fixed, and compute $\tilde{d}_i(D)$, in order to prove that project duration cannot be smaller than D. Then, we compute all $[t_1, t_2]$, $t_1 \in \{r_i, r_i{+}p_i, \tilde{d}_i(D){-}p_i, \forall i \in \{1..n\}\}$, $t_2 \in \{r_i{+}p_i, \tilde{d}_i(D){-}p_i, \tilde{d}_i(D), \forall i \in \{1..n\}\}$, where p_i is the activity's duration. On those $[t_1, t_2]$ we check if there are enough resources to satisfy at least the mandatory part of each activity, which is the minimum part of an activity that must be processed on the interval. The mandatory part of A_i between t_1 and t_2, $w(i, t_1, t_2)$, is:
$w(i, t_1, t_2) = min(max(0, r_i + p_i - t_1), max(0, t_2 - (\tilde{d}_i(D) - p_i)), p_i, t_2 - t_1)$.

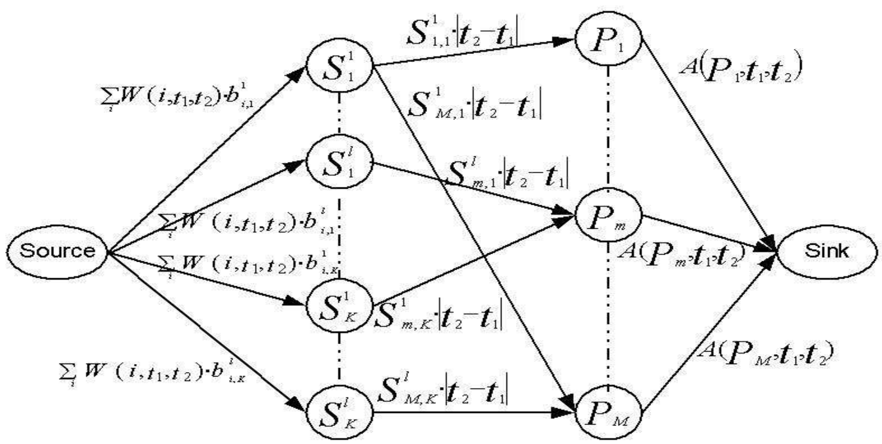

Figure 2: Needs of mandatory parts to satisfy on $[t_1, t_2]$

If it does not exist a feasible assignment of staff members to mandatory parts of the activities according to employees' skills constraint, we know that there is no feasible schedule that respect $\tilde{d_i}(D)$, thus D+1 is a valid lower bound for the project duration.

Once angain we solve the corresponding assignment problem, presented by the graph 2, where we search for a maximum flow. If there exists at least one interval where we cannot satisfy the activities, i.e. the maximum flow is lower than the sum of the needs of all the mandatory parts on the interval then we can conclude that there is no solution that respect $\tilde{d_i}(D)$, so D+1 is a valid lower bound.

4 Conclusion

The problem introduced here is a first attempt to modelize problems where staff members have different skills and different ability levels. The lower bounds we propose will be used both to evaluate efficiency of heuristic methods [2] and improve existing Branch and Bound Methods. When we use this lower bound in a tabu search the gap between it and the final makespan we get is around 4.2%, on 185 pseudo-randmoly generated instances. Moreover, for 70 instances the lower bound is equal to the final makespan found.

References

[1] Baptiste Ph., Le Pape C. and Nuijten W., *Satisfiability Tests and Time Bound Adjustments for Cumulative Scheduling Problems*. Annals of Operational Research 92(1999): 305-333.

[2] Bellenguez O. and Néron E., *Méthodes approchées pour le probleme de gestion de projet multi-compétence*, Ecole d'Automne de Recherche Opérationnelle, Tours, France(2003).

[3] Lopez P., Erschler J. and Esquirol P., *Ordonnancement de Tâches sous Contraintes : une Approche Energétique*, R.A.I.R.O.-A.P.I.I., 26(1992): 453-481.

[4] Mingozzi A., Maniezzo V., Ricciardelli S. and Bianco L., *An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation.*, Management Science, 44-5(1998): 714-729.

[5] Néron, E., *Lower Bounds for the Multi-Skill Project Scheduling Problem*, 8th International Workshop on Project Management and Scheduling , Valence, Espagne(2002): pp. 274-277.

[6] Toroslu, I., *Personnel assignment problem with hierarcical ordering constraints*, computers and industrial engineering(2003): 45, 493-510.