REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
Université Abdelhamid Ibn Badis Mostaganem
Laboratoire de Mathématiques Pures et Appliquées

1^{er} *Workshop*
en
Contrôle et Optimisation
Mostaganem 18-19 et 20 Décembre 2011

Titre : The constraints duration in project scheduling

Nom Prénom: Mouhoub Nasser Eddine
E-mail : mouhoub_n@yahoo.fr

Keywords: Line graph of graph, project scheduling, AoN directed acyclic graph, AoA directed acyclic graph, PERT network.

Abstract : A project is an enterprise consisting of several activities which are to be carried out in some specific order. They can be represented graphically in two different ways, either by assigning the activities to the nodes 'AoN' directed acyclic graph (dag) or to the arcs 'AoA dag'. In this paper, a new algorithm is proposed for constructing, for a given project scheduling problem, an Activity-on-Arc dag starting from the Activity-on-Node dag using the concepts of line graphs of graphs.

The relation $\prec$ can be represented graphically in two different ways, either by assigning the activities to the nodes or to (a subset of) the arcs. In either case a *directed acyclic graph* (*dag*) is defined. In an *activity on node* (*AoN*) dag, each activity corresponds one-to-one with a node, and we say that $u \prec v$ is *represented* if there is a directed path of arcs leading from v's node to u's node. Thus, an *AoN* dag is unique except for possible transitive arcs. In an *activity on arc* (*AoA*) dag or (PERT network), each activity v corresponds to an arc, where parallel arcs that share the same start and terminal nodes are permitted. We say $u \prec v$ is represented in an *AoA* dag if there is a path from t_u the terminal node of the arc for u, to s_v, the start node of the arc for v (the path is empty if $t_u = s_v$). Experts prefer to work with the PERT network rather than the *AoN* dag because it is more concise, it is close to the famous Gantt diagram and the structure of the PERT network is much more suitable for certain analytical techniques and optimization formulations. However, the major disadvantage of this method is in the existence of dummy arcs. Their number is likely to be significantly high especially if the size of the network is too large. The *AoA* dag is not unique. By using the concepts of line graph of graph, we build an *AoA* dag starting from an *AoN* dag. This idea is transformed into an algorithm using graph theory. Because of the facility of use of *AoA* dag, we must concentrate our efforts on the study of the possibility of transforming the *AoN* dag (a significant number of arcs) to *AoA* dag (a reduced number of arcs). So we want to know how to transform H (which is an *AoN* dag) in order to get a new graph which is the line graph (*AoA* dag). The difficulty which arises is to know if H does contain "Z" configurations or not (see fig. 1. (a))? If it does not contain "Z" it is a line graph and the transformation is immediate. Before turning to this transformation, we must identify all arcs containing constraint duration. They should be studied in groups of two. It is also important to introduce new tasks and to subdivide certain others in a way that the arcs starting from the same node must have the same value. We partition arcs of the *AoN* dag in a complete bipartite $B_i = (X_i, Y_i)$. In the *AoA* dag which we wants to build, each B_i is represented by a node still noted B_i and will be the center of the star like in fig.2

But if it contains "Z", we have to eliminate the bar from each "Z" preserving the constraints of succession. It is in this case that one must modify it in order to transform it into a directed acyclic graph associated by preserving the constraints of anteriorities. We introduce, in the *AoN* dag, a dummy arc f in every "Z" (fig. 1(b)). The introduction of the dummy arcs aims at eliminating all "Z" configurations from the *AoN* dag, the constraints remains unchanged. We make now the necessary transformations as in the precedent case. It is advantageous to reduce the length of calculations to build an *AoA* dag having the minimum number of nodes and dummy activities. We then pose the problem of looking for "Z" in the

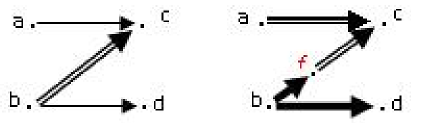

Fig. 1. « Z » configuration in AoN dag and his transformation.

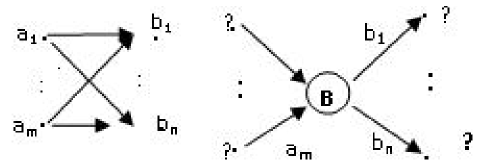

Fig. 2 The complete bipartite B in the line graph and the node B of the corresponding AoA dag.

AoN dag. The technique consists to eliminate several "Z" at the same time by regrouping bars having the initial end or the final end in the same bipartite complete.