Ann. Acad. Rom. Sci.
Ser. Math. Appl.
ISSN 2066-6594 Vol. 10, No. 2/2018

NEW HEURISTICS FOR MODELING TEMPORAL CONSTRAINTS AND GENERATING PERT NETWORK WITH MINIMUM NUMBER OF DUMMY-ARCS*

Nasser Eddine Mouhoub† Abdelhamid Benhocine‡
Abdelkader Gasmi§

Abstract

In this paper, we study the various types of temporal constraints in project scheduling problem (PSP), then modeling them by using some concepts of line graphs. We apply a new technique for transforming an AoN (Activity-On-Node) network and containing a significant number of arcs with temporal constraints into an AoA (Activity-on-Arrow) network or PERT (Program and Evaluation and Review Technique) network which contains fewer real arcs. Finally, we propose a new technique for constructing, for a given PSP, a PERT network having the minimum number of dummy arcs. The polynomial algorithm regrouping all the techniques and dealing with the existence of transitive arcs is given at the end with an illustrative example.

MSC: 90B35, 90B10, 68R10

keywords: Minimal *AoA* network, PERT graph, Project scheduling problem, Temporal constraints.

*Accepted for publication on May 11, 2018
†mouhoub_n@yahoo.fr Address; Computer science department M'sila University Algeria
‡Address;Mathematics department Setif university Algeria
§gasmi_a@yahoo.fr Address; Computer science department M'sila University Algeria

1 Introduction

In PSP, the precedence constraints between the activities of a project can be represented graphically in two different ways: by assigning the activities either to the nodes or to (a subset of) the arcs of a network. In either case, a directed acyclic network is defined. In AoN network, $G = (V, E)$, the nodes represent the project activities and the arcs represent the immediate precedence relations between the activities. Thus, an AoN network is unique [27]. In AoA networks, one has to often draw some dummy activities in order to satisfy the precedence relationships.

Practitioners of PSP prefer to work with the AoA network because it is easy to read; each activity is represented by an arc. They give number of arguments to justify their choice. This is why according to Fink and Fourier [15], it is more concise. Furthermore, Tsou [36] explain that it is close to the famous Gantt diagram where it is used to represent, in time, the various activities project. According to Cohen and Sadeh [4] 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. Kelley [21] notes that it is advantageous to reduce the length of calculations to build a PERT network having the minimum number of vertices and dummy activities. Krishnamoorty and Deon [22] showed that finding the minimum dummy arcs problem is NP-hard. We focus our objective to introducing temporal constraints in PSP, modeling them by graphs, proposing a new algorithm for generating, for a given PSP, a PERT network starting from the AoN graph and constructing the minimal PERT graph with minimum number of dummy activities using the concepts of line graphs of graphs.

2 Literature

Reasoning with temporal constraints has been a hot research topic for the last fifteen years. According to Dechter [10], the problems involving temporal constraints arise in various areas of computer science such as scheduling, program verification, and parallel computation. Research in common-sense reasoning [15, 32], natural language understanding [1, 18], and planning [24], has identified new types of temporal reasoning problems, specific to Artificial Intelligence applications. Several formalisms for expressing and reasoning about temporal knowledge have been proposed, most notably, Allen's interval algebra [33], linear inequalities (Malik and Binford [23], Dean and McDermott's time map [5]. Each of these representation schemes is supported by a specialized constraint-directed reasoning algorithm. At the same time, extensive research has been carried out over the past years on problems involving general constraints (Montanari [26], Gaschnig [12], Freuder [17], Nebel and Buckert [30], Drakengren and Jonsson [7], Dechter and Pearl [6], yet much of this work has not been extended problems involving temporal constraints. Smith and Pyle [33] presented a new heuristic algorithm in resource constrained project scheduling problem with time windows, Vanhoucke [37] presented a new approach for the treatment of temporal constraints in PSP.

3 Definitions and Notations

Let $G = (V, E)$ be a network. $\Gamma^+(G, i)$ denotes the successors of the vertex i and $\Gamma^-(G, i)$ its predecessors in G. $V(G)$ and $E(G)$ are, respectively, the sets of vertices and arcs of G. If P is a path of G, then $I(G, P)$ and $T(G, P)$ denote the initial vertex and the terminal vertex of P, respectively (P may be an arc).

Let $1, 2, ..., n$ denote the n real activities of a schedule table T which is represented by either a Gantt diagram or an AoN network (G_N) or an AoA network (G_A). Each activity v is represented by a vertex v in G_N and we put the arc (v, u) in G_N if activity v is an immediate predecessor of activity u. There is no activity which succeeds itself, thus G_N and G_A has no circuit. To have one vertex without predecessors and another vertex without successors in G_N and G_A, we add to the schedule table an activity α, with 0 duration, called source, preceding all activities with no predecessors and an activity ω succeeding to all activities with no successors called sink (see [27]). If there is one activity which does not have any predecessor (resp. successor), then let a (resp. x) be this activity.

3.1 Line graph of graph

Let $G = (V, E)$ be a simple or multiple digraph ($|X| > 1$). According to Gross and Yellen [13], a line graph $L(G)$ (also called an adjoint, conjugate, edge-to-vertex dual) of a simple graph G is obtained by associating a vertex with each edge of the graph and connecting two vertices with an edge *iff* the corresponding edges of G have a vertex in common (see figure 1). The line graph of a directed graph G is the directed graph $L(G)$ whose vertex set corresponds to the arc set of G and having an arc directed from an edge e_1 to an edge e_2 if in G, the head of e_1 meets the tail of e_2.

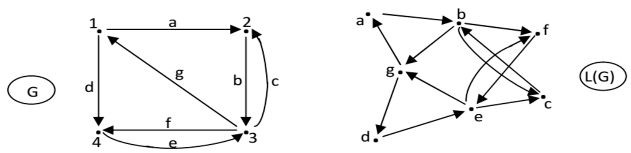

Figure 1: A graph (G) and his line graph L (G).

According to Hemminger et al. [14], if G is a digraph with n vertices $x1, x2, ..., xn$ and $L(G)$ its associated line graph with n' vertices and m' arcs, then: $n' = m$, $m' = \sum_{i=1}^{n} d^-(x_i) \cdot d^+(x_i)$.

Furthermore, the in-respectively out-degree of a vertex $x' = (xi, xj)$ in $L(G)$ are: $d^-(x') = d^-(xi)$, $d^+(x') = d^+(xi)$. The line graphs have been very well studied but we only give, in this section, the outcomes of interest found in [27, 2]:
(a) H is the line graph of a digraph if and only if H does not contain any $'Z'$ configuration.
(b) H is the line graph of a graph G if and only if arcs of H can be partitioned in a complete bipartite $B_i = (X_i, Y_i), i = 1, ..., m$, such that $X_i \bigcap X_j = \emptyset$, and $Y_i \bigcap Y_j = \Phi \forall i \neq j$. The bipartite B_i of H are then in a bijection with the vertices also noted B_i which are neither sources nor sinks, two vertices B_i and B_j of G being connected by an arc from Bi towards B_j if and only if the complete bipartite B_i and B_j of H are such that $Y_i \bigcap X_j \neq \emptyset$, (see figure 2).

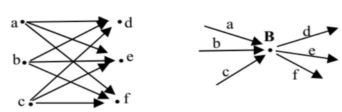

Figure 2: A complete bipartite B of H and the star B of G

(c) H is the line graph of a graph without loops if and only if H does not contain any configuration $'Z'$ or $\triangle$ (see Section 4). Configuration $'Z'$ appears when two nodes have common successors and no common successors or by symmetry when two nodes have common predecessors and no common predecessors (see figure 3).

(d) H is the line graph of a graph if and only if any pair of vertices

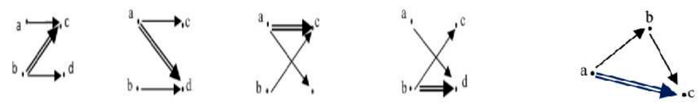

Figure 3: The configuration ”Z”, his forms and the configuration ’△’

having common successors has all their common successors.
(e) H is the line graph of a graph if and only if any pair of vertices having common predecessors has all their common predecessors.

According to Shah et al. [31], H is not the line graph of any directed acyclic graph if is only if there is a pair of nodes having common successors and no common successors or common predecessors and no common predecessors. So, we want to know how to transform H in order to get a new graph which is a line graph of a graph.

4 Generating *AoA* network from a given *AoN* network

4.1 *AoN* network without temporal constraints

We shall concentrate, in this section, on the study of the possibility of transforming the *AoN* network which has a significant number of arcs to an *AoA* network having a reduced number of arcs. How to convert the graph H (which is an *AoN* network without temporal constraints) in order to get a new graph which is the line graph (*AoA* network)? In a previous work [29], we have shown that the difficulty that arises is to know whether H does contain ”Z” configurations or not? If it does not, it is a line graph and the transformation is immediate (as in Fig. 2): Let H be a network. Then G is the line digraph of H, and we write $G \cong L(H)$, if and only if the vertices of G are the arcs of H and (x, y) is an arc of G whenever x and y are arcs of H with $T(x) = I(y)$. A digraph G is the line digraph of a digraph H, or $G \cong L(H)$, if and only if G has no configuration $'Z'$, that is whenever (p, q), (r, q), (r, s) are arcs of G then so is (p, s) and if G has no circuit, then G has no configuration $'\triangle'$. The bare of $'Z'$ is (r, q) and the base of $'\triangle'$ is (p, r) like in figure 4. $(r, q) \in E(G)$ is not a bare of $'Z'$ if and only if: $\forall p \in \Gamma^-(G, q)$, $\forall s \in \Gamma^-(G, r)$, $(p, s) \in E(G)$.

$(p, r) \in E(G)$ is not a base of $'\triangle'$ if and only if $\Gamma^-(G, q) \bigcap \Gamma^-(G, r) = \emptyset$. Consequently, this result is $G_N \in L(G_A)$ if and only if G_N has neither configuration $'Z'$ nor configuration $'\triangle'$: constructing G_A is easy in this

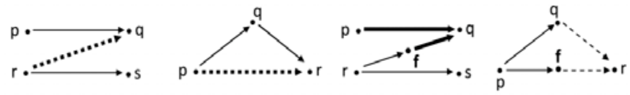

Figure 4: The configurations " Z ", " $\triangle$ ", the introduction of dummy activity f and the partition of completes bipartite in G_N

case since one can use the partition of $E(G_N)$ into bipartite subgraphs [2]. The problem is to construct G_A when G_N has one of the configurations $'Z'$ or $'\triangle'$, that is, how modifying G_N to have a new G_N, satisfying constraints in the scheduling table T and without configuration $'Z'$ or $'\triangle'$. The usual way is to introduce also the dummy activities by subdividing the bare of 'Z' or base of $'\triangle'$ in G_N like in figure 4. The introduction of the dummy arcs aims to eliminate all the $'Z'$ and $'\triangle'$ configurations from the *AoN* network G_N, the constraints remain unchanged. We should recall that the dummy arcs are not necessary in the *AoN* network G_N but are introduced only to build an *AoA* network G_A. For more details on such transformation, the reader can refer to [29].

4.2 *AoN* network with temporal constraints

4.2.1 Definition

The temporal constraint is a time allocation constraint. It comes from imperative management constraints such as the supply availability or time delivery, etc. It specifies the time interval (or semi-interval) during which it is possible to perform an activity. These constraints are often due to availability of stakeholders (human resources): for example a company which produces frames can only intervene between June 15 and August 31 [30].

The temporal constraint affects the modeling of project scheduling and changes. The problem therefore, is to find a way or a technique to normalize the situation and bring it back to the modeling PSP by graphs. In the following, we shall propose an original method that allows us to model the temporal constraints and include them in PSP.

4.2.2 Classification

The class of quantitative temporal constraints has been studied originally by Dechter et al. [7] using Allen's interval algebra. The scope of our article is restricted to those constraint-satisfaction problems that can be stated as follows: modeling the temporal constraints by graphs not by intervals and include them in the search of the AoA graph from a given AoN graph. The precedence relationships typically used in these charts are expressible in terms of the before and meets relation, two of the relations defined in Allen's interval algebra.

The following base relations (Table 1) capture the possible relations between two intervals.

Relation	Symbol	Inverse	Illustration
X before Y	b	bi	X Y
X equal Y	=		X Y
X meets Y	m	mi	X Y
X overlaps Y	o	oi	X Y
X during Y	d	di	X Y
X starts Y	s	si	X Y
X finishes Y	f	fi	X Y

Table 1: Temporal Relations Between Activities

Using this calculus, given facts can be formalized and then used for automatic reasoning. Relations between intervals are formalized as sets of base relations. By graphs, we can classify the most important temporal constraints into eight types and add the latest which is that of the precedence:
(TC1) Activity A starts t time units before the work begins.
(TC2) Activity A can only start t time units after the beginning of the work.
(TC3) Activity B must start t time units after the end of activity A.
(TC4) Activity B starts a fraction of time units a/b after the start of activity A ($a < b$).
(TC5) Activity B must start t time units after the start of activity A ($t < tA$).
(TC6) Activity A must start before time t.
(TC7) Activity A during activity B
(TC8) Activity A equal activity B
(TC9) Activity B must immediately follow activity A.
Figure 5 gives the representation of temporal constraints in an AoN network, and Gantt chart. (f denotes a dummy activity with the duration t).

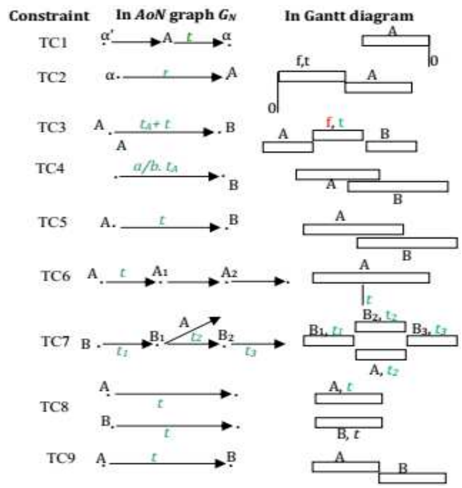

Figure 5: Temporal constraints in G_N network and Gantt diagram

4.2.3 Modeling temporal constraints

In PSP, which is a particularly in an AoN network, incident arcs outside a node (that is to say an activity) have the same value. The presence of temporal constraints in the network AoN violates this property, which makes solving the project scheduling impossible. Calculating dates and critical path research... also become impossible. Modeling by using graphs can solve this problem. We will present in the following a new technique that allows handling such constraints. The following figure (figure 6) gives the unique representation of these constraints in the network AoN. It is clear that the values on the arcs incident out of the node A are different (see the example in figure 8).

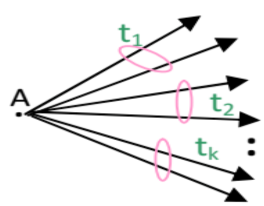

Figure 6: Main temporal constraints in G_N

The representation in G_A (figure 7(a)), consists in gathering several dummy arcs succeeding the real activity A and which have the same value in a single dummy activity. For constraints of type (TC4), (TC5) and (TC9),

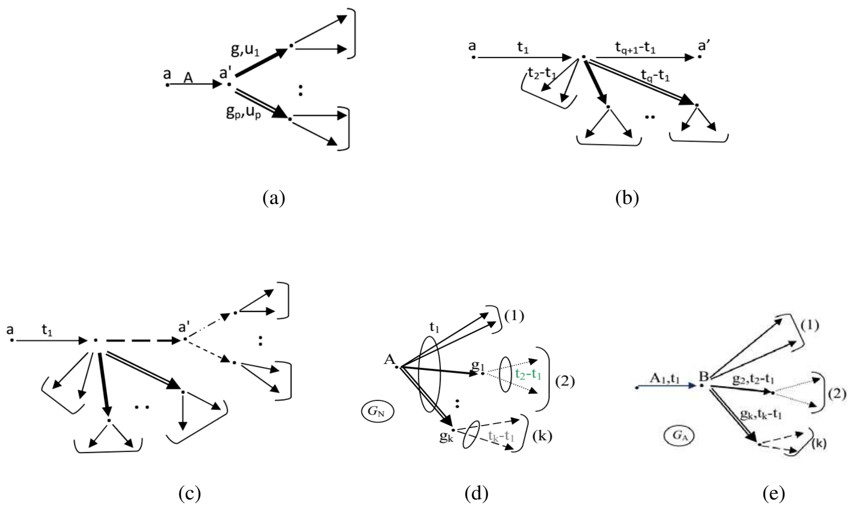

Figure 7: (a) The representation of (TC2), (TC3), (TC7) and (TC8) constraints in AoA network(G_A), (b) Representation of (TC4) (TC5) and TC(9) constraints in G_A, (c) Every type of constraints in G_A, (d) Modification in G_N, (e) In G_A, activity A is subdivided as (A_1, t_1) in G_A. Arcs of the same initial node have the same value

we notice that both start after the beginning of activity A. Representation in *AoA* network(G_A) implies the segmentation of A (or B if TC(9)) into several tasks, in the general case ($A = A_1 + A_2 + ... + A_{q+1}$) or ($B = B_1 + B_2 + B_3$ if (TC9)). We note that the representation of Fig. 7(b) solves this type of constraints.

Finally, we can combine figure 7(a) and figure 7(b) keeping in mind the idea of minimizing dummy arcs. Indeed, to arrive at figure 7(c) we must modify, in G_N, the arcs incident out of a node and which do not have the same value, by introduction of dummy arcs of 0 duration in order to be able to partition it into complete bipartite sub-graphs. The rest of the temporal constraints are also included in the new modeling technique as well as the other types in Table 1.

All these combinations lead us to the changes made in the two networks G_A and G_N respectively (figure 7(d) and figure 7(e)). Correspondence between the representations of temporal constraints in G_N and in G_A is our goal; we modify figure 6. in figure 7(d) and figure 7(c) in figure 7(e).

The introduction of activities $g_1, g_2, , g_k$ of times units $t_2 - t_1, ..., t_k - t_1$, respectively, has the advantage of giving the same value to the arcs of the same initial node in G_N. There is no difficulty to verify that the arcs of the graph (figure 7(d)) are partitioned into a complete bipartite subgraphs and that is the line graph of the graph in figure 7(e).

Let remember that introducing dummy activities f_i in AoA networks, with zero duration, gives the possibility to solve certain situations and raise ambiguities. They do not take in consideration any material or financial mean [27]. For more details, the reader can refer to [4-29]. The dummy activities gi in AoN network is not of zero durations. It is introduced to solve, in G_N, the problem of the presence of temporal constraints in PSP.

In conclusion, the modeling of temporal constraints is to establish a combination involving different types of constraints in G_N in order to find the temporal constraint which is the minimum value of duration and partition each of the other activities in two, the first with the minimum duration and the second with the rest. This partition requires the creation of dummy activities gi for each of the activities. Hence, the transformation to G_A, now becomes easier by applying results of section 3.

To explain these results, consider the following example:
let A be an activity of duration 5 time units. Suppose that:
A precedes B, B_1 and B_2 cannot start 2 time units after the start of activity A, B_3 and B_4 begin only 3 time units after the start of A, B_5 cannot start until A is 3/4 finished, B_6 and B_7 begin only 8 time units after the end of A.
In AoN and AoA networks, let us draw the arcs leaving the node A (figure 8):

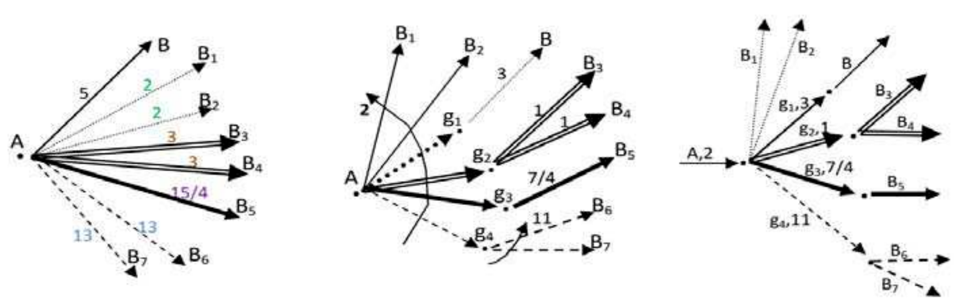

Figure 8: To the left, representation of temporal constraints. Between, subdivision of activity Aas (A_1, g_i) in G_N which the arcs of the same initial node have the same value. To the right, the correspondence in G_A).

4.3 Constructing and generating PERT network

Now, combining the results obtained in previous section with those in Section 4, we can construct an AoA network G_A from the AoN network G_N containing or not temporal constraints as follows:

- Dispose of all, temporal constraints modeled in Fig.7(a).
- Remove the $'Z'$ bares, the $'\triangle'$ bases and make the transformation according to results of Sections 3 and 4.
- Create a new schedule table T' from the initial one T by adding all activities f_i and gi with their predecessors that are created in G_N.

These ideas will be explained in detail in the algorithm(section 6).

Lemma 1. *The number of dummy activities f_i with 0 durations in G_A is equal to the number of $'Z'$ bares and $'\triangle'$ bases in G_N.*

Lemma 2. *If $(i,j) \in E(G_N)$ is neither a temporal constraint, nor a bare of $'Z'$ nor a base of $'\triangle'$, then $T(G_A,i) = T(G_A,j)$.*

Proof. By applying the theorem of Shah [31], G_N can be partitioned into mutually arc-disjoint complete bipartite subgraphs Li, $G_N \cong G_A$ then $T(G_A,i) = T(G_A,j)$ and G_A satisfies schedule table T. □

Lemma 3. *If $(i,j) \in E(G_N)$ is a temporal constraint then $T(G_A,i) \neq T(G_A,j)$.*

Proof. It's clear that $\Gamma^-(G,j_l)l = 1,2,...k$ in T' table is the same in table T (see figure 9)

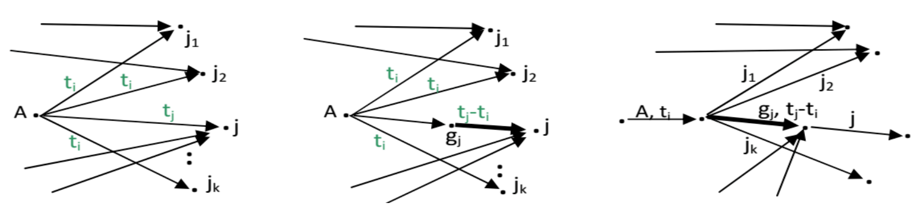

Figure 9: Temporal constraint in G_N, activity A is subdivided as (A, g_1) in G_N and correspondence in G_A

The problem is when g_j appears in G_N and G_A. We know that, $\Gamma^-(G_A,j) = \Gamma^-(G_N,j) \bigcap gj$, $\Gamma^-(G_A,gj) = A$, so $\Gamma^-(G_A,j) \subseteq \Gamma^-(G_N,j)$ then the activity j satisfies schedule table T. □

Lemma 4. *If* $(i, j) \in E(G_N)$ *is a bare of* $'Z'$ *or a base of* $'\triangle'$*, then* $T(G, i) \neq T(G, j)$ *in* G_A *and satisfies schedule table* T*.*

Proof. If $T(G, i) = I(G, j)$ in G_A with schedule table, then

- $\forall p \in \Gamma^-(G_N, j)$, $\forall q \in \Gamma^+(G_N, i)$, $(p, q) \in E(G_N)$, so $(i, j) \in E(G_N)$ is neither a bare of $'Z'$ nor a base of $'\triangle'$, a contradiction. - $(i, j) \in E(G_N)$ is a bare of $'Z'$ or a base of $'\triangle'$ (see Fig. 4.), then

in T: $\Gamma^-(G_N, j) = a, i$,

in T': $\Gamma^-(G_N, j) = a, f$,

f is the unique predecessor of j in G_N, then $\Gamma^-(j) = f$,but $\Gamma^-(f) = i$,then $(i) \in \Gamma^-(G_N, j)$. By applying the theorem of Shah et al. [31], G_N can be partitioned into complete bipartites, and since $G_A \cong G_N$, then G_A satisfies schedule table T'. □

Corollary 1. *Corollary The AoA network* (G_A) *constructed from the AoN network* (G_N) *containing temporal constraints satisfies the initial schedule tables* T' *and* T*.*

5 Constructing the minimal PERT Network

Due to their importance, PERT networks have been drawing many research trials to generate *AoA* networks with minimal number of dummy arcs: Hayes [19] gave a set of approaches to construct an PERT network but does not produce the minimal *AoA* network. He observed that the number of dummy arcs and nodes in the *AoA* network cannot be minimised simultaneously. Dimsdale [9] gave an algorithm for generating a minimal *AoA* network. Fisher et al. [16] proved that Dimsdale's algorithm is false and gave a novel and exact one but with no proof. Cantor and Dimsdale [3] gave, with proofs, an exact algorithm which minimizes the number of vertices in a polynomial time. According to Sterboul and Wertheimer [34], this algorithm seeks rather to minimize the number of vertices and leads to unnecessary long calculations. Krishnamoorty and Deon [22] explained that searching the PERT network with the minimum number of dummy arcs is NP-hard. [20,15], [8,25] proposed PERT network-generating methods but they did not consider the redundancy and other problems brought up by dummy arcs, and therefore, are considerably impaired. Elmaghraby and Herroelen [11] developed a complexity index as a measurement tool for the complexity in activity networks. Kamburowski et al. [19,20] proposed a method that generates minimal complexity-index PERT networks. [4, 28] proposed algorithms of passage from the *AoN* network to an *AoA* network regardless the minimisation of the number of dummy arcs. In previous work [28], we presented an algorithm that builds the minimal PERT graph in terms of dummy activities but two steps are not accurate (the rules 3 and 4 precisely). In this paper we present a new algorithm with the necessary corrections.

5.1 Actions

Let G_A the PERT network obtained in section 4 without temporal constraints. $f_1, f_2, , f_p$ denote the p dummy activities in G_A. It is necessary to represent each real activity i by a black arc, $i = 0, \ldots, n+1$ (with activities α, ω and $g_i, i = 1, 2, \ldots, k$) and each dummy activity by a red arc, this first PERT network G_A satisfies the precedence constraints. The following actions allow us to construct from G_A, a new G_A which is the PERT network having the minimum number of dummy activities and satisfying the schedule table by a decreasing number of arcs and vertices.

1. In G_A, contract the terminals of the real activities (black arcs) having the same successors into one vertex (see figure 10).

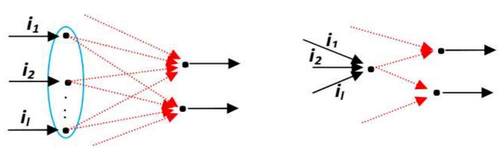

Figure 10: The real arcs having the same successors contracted in a single vertex in G_A

2. In G_A, contract the initials of the real activities (black arcs) having the same predecessors into one vertex (see Fig. 11).

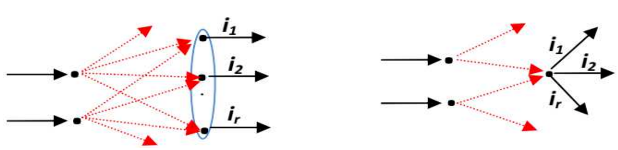

Figure 11: The real arcs having the same predecessors contracted in a single vertex in G_A

3. In G_A, contract the initial vertex ($I(f_j)$) and the terminal vertex ($T(f_j)$) into one vertex if $d^-(T(f_i)) = +1$, with: there is no real arc incident into T(fj)) and there is no real arc incident out of $I(f_j)$ (see figure 12).

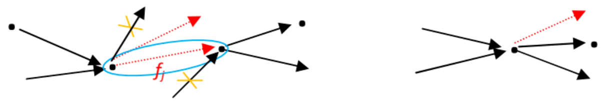

Figure 12: The dummy arc f_j with $d^-(T(f_i)) = +1$, contraction of the vertices $I(f_j)$ and $T(f_j)$ into one vertex in G_A

4. In G_A, contract the initial vertex $(I(f_j))$ and the terminal vertex $(I(f_j))$ into one vertex if $d^+(I(f_j)) = +1$, with: there is no real arc incident into $T(f_j)$ and there is no real arc incident out of $I(f_j)$ (see figure 13). Remove the resulting red loop and repeat this contraction as many times as possible.

Figure 13: To the left, representation of temporal constraints. Between, subdivision of activity Aas (A_1, g_i) in G_N which the arcs of the same initial node have the same value. To the right, the correspondence in G_A).

5. In G_A, if the vertices of which all their successors are included in the set of successors of another vertex, then add to G_A a red arc $(T(G_A, i)$, $T(G_A, j))$ and remove transitive red arcs with initial vertex $T(G_A, i)$. Now, apply this rule to vertices in $\Gamma^+(G_A, T(G_A, j))$ as many times as possible.

6. In G_A, if the vertices of which all their predecessors are included in the set of predecessors of another vertex, then add in G_A a red arc $(I(G_A, j), I(G_A, i))$ and remove transitive red arcs with terminal vertex (G_A, i). Now, apply this rule to vertices in $\Gamma^-(G_A, I(G_A, j))$ as many times as possible.

7. Now apply again, in G_A, if possible, the actions 3 and 4 as many times as possible to obtain a PERT network G_A with minimum number of dummy arcs. 8. If (A, B) is a maximal red bipartite partial subgraph of G_A with $|A| = 2$, $|B| = 2$ and $|A| + |B| = 6$, then delete all its arcs and add a red vertex sf and the red star (A, sf, B). For more details of the actions 5, 6 and 7, the reader can refer to [28].

6 Algorithm

The four steps are applied in a sequential order and the backtracking is unauthorized except if it is mentioned in the step.

BEGIN
STEP 1 :
For every node $v_i(i = 1, 2, \dots, n)$ do
if G_N contains 2 vertices $v_j, v_k \in \Gamma + (v_i)$ and $t(v_j, v_j) \neq t(v_j, v_k)$ then
One pose $t_{min} = min[t(v_i)](i = 1, 2, \dots, n)$
For every vertex $v_j \in \Gamma + (v_i)$ do
if $t(v_j, s_j) > t_{min}$ then
- create dummy node g_j in G_N
- replace the bare (v_j, v_i) in G_N by (v_j, g_j) and (g_j, v_j)
with $t(g_j,\, v_j) = t(v_j, v_i) - t_{min}$ and $t(v_j, g_j) = t_{min}$
STEP 2:
if G_N contains 'Z' configurations then
Identify the Z_i (i = 1, 2, , m)
For every bipartite containing one or more bares of 'Z' do
- Create dummy nodes corresponding to the number of bipartites.
- Regroup the bars having the same initial or the same terminal extremity to the same complete bipartite
- Replace the bare (b_i, c_i) of Z_i in G_N by (b_i, f_i) and (f_i, c_i)
- Create a new schedule table T' from the initial one T by adding all activities f_i and g_i with their predecessors that are created in G_N.
STEP 3
- Identify again the bipartite in G_N
- Represent every bipartite B_i in G_N by a node B_i in G_A.
- Represent every arc such that: an arc is drawn between two nodes B_i and B_j in G_A iff the two bipartite B_i and B_j in G_N are such that and $Y_i \bigcap Y_j = \emptyset$.
STEP 4

1. Contract the terminals of the real activities (black arcs) having the same successors into one vertex. Remove multiple red arcs and repeat this contraction as many times as possible.

2. Contract the initials of the real activities (black arcs) having the same predecessors into one vertex. Remove multiple red arcs and repeat this contraction as many times as possible.

3. Contract the initial vertex $(I(f_j))$ and the terminal vertex $(T(f_j))$ into one vertex if $d^-(T(f_i)) = +1$, with: there is no real arc incident into $T(f_j))$ and there is no real arc incident out of $I(f_j)$. Remove the resulting red loop and repeat this contraction as many times as possible.

4. contract the initial vertex ($I(f_j)$) and the terminal vertex ($T(f_j)$) into one vertex if $d^+(I(f_j)) = +1$, with: there is no real arc incident into $T(f_j)$) and there is no real arc incident out of $I(f_j)$. Remove the resulting red loop and repeat this contraction as many times as possible.

5. If i, $j \in V(G_N)$ with $\Gamma^+(G_N, j) \subset \Gamma^+(G_N, i)$,then add to G_A a red arc $(T(G_A, i), T(G_A, j))$ and remove transitive red arcs with initial vertex $T(G_A, i)$. Apply this action to vertices in $\Gamma^+(G_A, T(G_A, j))$ as many times as possible.

6. If i, $j \in V(G_N)$ with $\Gamma^-(G_N, j) \subset \Gamma^-(G_N, i)$, then add in G_A a red arc $(I(G_A, i), I(G_A, j))$ and remove transitive red arcs with terminal vertex $T(G_A, i)$. Apply this action to vertices in $\Gamma^-(G_A, T(G_A, j))$ as many times as possible.

7. Apply again, in G_A, if possible, the actions 3 and 4 as many times as possible.

8. If (A, B) is a maximal red bipartite partial subgraph of G_A with $|A| > 2$, $|B| > 2$ and $|A| + |B| \geq 6$, then delete all its arcs and add a red vertex sf and the red star (A, sf, B)

END

6.1 Example

Activity	Description	Duration	Predecessor Activity
α	Beginning of project	0	$-$
A	Excavate	12	α
B	Lay the foundation	10	α
C	Put up the rough wall	9	α
D	Put up the roof	6	α

E	Install the exterior plumbing	5	A
F	Install the interior plumbing	4	A, B, C
G	Put up the exterior siding	3	B, C, D
H	Do the exterior painting	6	A, E
I	Do the electrical work	8	E,F
J	Put up the wallboard	2	E, F, G
K	Install the flooring	7	G
L	Install windows	7	H
M	Install the doors	12	H, I
N	Do the interior painting	9	J, K
O	Install the exterior fixtures	11	L, M
P	Install the interior fixtures	13	M, N
ω	End of project	0	O, P

Table: Activity list for the Reliable Construction Co. project.

To illustrate what we taking into account temporal constraints, we consider the following example [35]: The manager of a project will need to arrange for a number of crews to perform the various construction activities at different times. Table 2 shows his list of the various activities. The third column provides important additional information for coordinating the scheduling of the crews.

Temporal constraints are: A starts 2 time units after the beginning of D. C can only start 7 time units after the beginning of the work. D can only start 3 time units after the beginning of the work. E starts 2 time units after the end of A and H starts 5 time units after the end of E. O begins when M is executed to 3/4 and P starts 4 time units after the end of N. K starts 3 time units after the end of G.

Constructing AoN network (G_N) from its schedule table (where all the immediate predecessors of each activity are listed) is a trivial task (see figure 14). The following figures (Figure 15, Figure 16 et Figure 17) represent the successive reductions of dummy arcs in the PERT graph.

6.2 Discussion

The algorithm finishes since the loop is carried out only when there is $'Z'$ configuration and the number of $'Z'$ in G_N is known in advance and finite. The rest of the algorithm is a succession of simple instructions. The complexity of the algorithm is therefore polynomial.

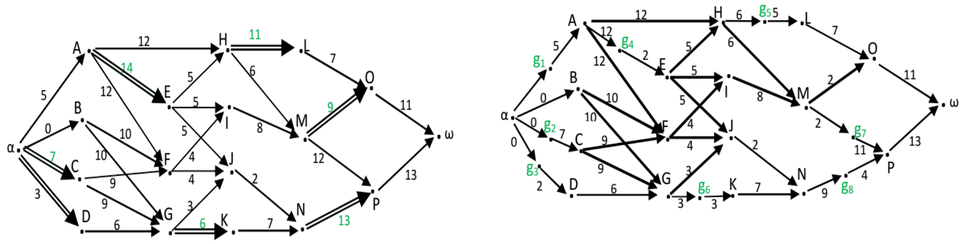

Figure 14: To the left, G_N graph from the schedule table (Table 2.). Edges in bold represent temporal constraints. To the right, G_N network whose edges have the same initial vertex have the same value. The dummy activities from temporal constraints g_i: activities α, A, G, H, M, N are divided in two activities. "Z" bares and $'\triangle'$ base are in bold.

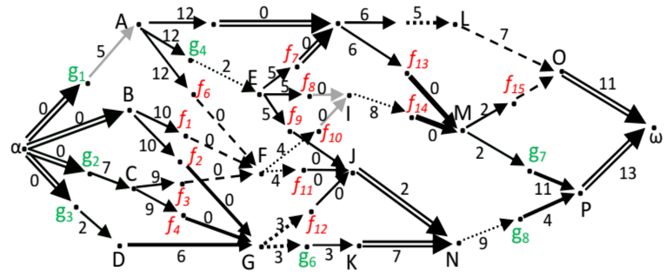

Figure 15: G_N network without "Z" configuration and whose vertices are reorganized into levels. We can verify that the edges can be partitioned into complete bipartite.

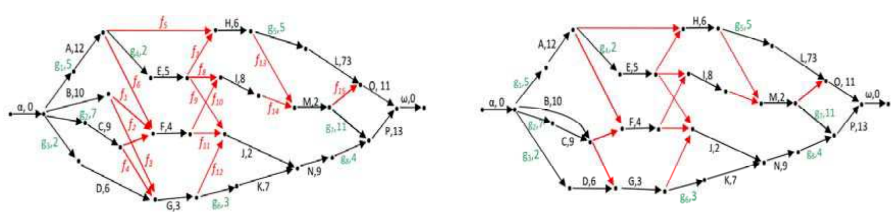

Figure 16: To the left, G_A network of Table 1. The 15 dummy activities f_i have zero duration. To the right G_A network with 13 dummy activities by applying action 1 and 2.

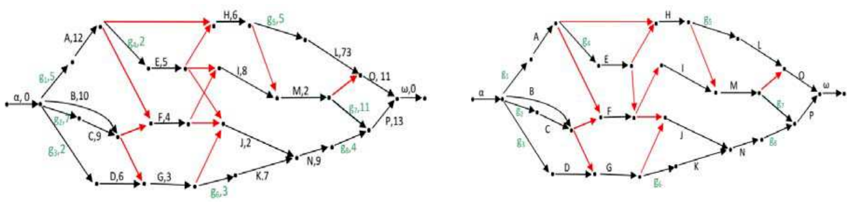

Figure 17: To the left, G_A network with 12 dummy activities by applying action 3 and 4. To the right, G_A network with 11 dummy activities by applying action 5

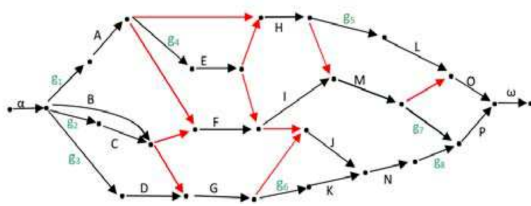

Figure 18: G_A network with 10 dummy activities by applying action 3 and 4. the constructed PERT graph satisfies the constraints of the table 2. The durations of the activities are not mentioned

7 Conclusion

This paper presents a new approaches for constructing PERT networks with temporal constraints and having the minimum number of dummy arcs. Indeed, by applying the four steps in sequential order, the algorithm models temporal constraints by graphs and introduces them in PSP. We have applied this model to a very useful problem for practitioners of project management. That is the shift of the AoN network (easy to draw but difficult in operation), to the AoA network (difficult to draw but widely used among practitioners) while including temporal constraints and using some results of line graphs of graphs. Our work is crowned by a polynomial complexity. the resolution of PSP becomes easier thus the calculation of earliest start and finish times, latest start and finish times, the critical path, free margins, etc. Our new approach is very simple to be applied. It gives a minimal PERT graph by applying eight actions with the total respect of the constraints in schedule table in a very short time. The techniques used in the rules of the algorithm can be exploited in other fields as in the project scheduling with resources constraints ($RCPSP$) and for solving Temporal Constraint

Satisfaction Problem ($TCSP$) in real time and in a dynamic environment.

References

[1] J.F. Allen. Towards a general theory of action and time. *Artif. Intell.* 23(2):123-154, 1984.

[2] A. Benhocine, J.L Fouquet. Pancyclic and panconnected line graphs. *J. Graph Theory* 11(3):385-398, 1987.

[3] D.G. Cantor, B. Dimsdale. On direction-preserving maps of graphs. *J. Comb. Theor.* (6):165-176, 1969.

[4] Y. Cohen, A. Sadeh. A new approach for constructing and generating AOA networks. *J. Eng. Comp and Arch.* 1 (1):1-13, 2007.

[5] T.L Dean, T. L. Mc Dermott. Temporal data base management. *Artif. Intell.* 32:1-55, 1987.

[6] R. Dechter, J.Pearl. Tree clustering for constraint networks. *J. Artif.Intell.* 38(3):353-366, 1989.

[7] R. Dechter. Temporal constraint networks. *Artificial intelligence Journal* 49:61-95., 1991.

[8] E. Demeulmeester, B. Dodin , W. Herroelen. A random activity network generator. *Oper. Res.* 41(5):972-980, 1993.

[9] B. Dimsdale. Construction of minimal project networks. *I.B.M. Syst. J.* 2:24-36, 1963.

[10] T. Drakengren, P. Towards a Complete Classification of Tractability in Allen's Algebra. *IJCAI* 7(1):211-221, 1997.

[11] S.E. Elmaghraby, W.S. Herroelen. On measurement of complexity in activity networks. *Eur. J. Oper. Res.* 5:223-234, 1980.

[12] J. Gashnig. Performance measurement and analysis of certain search algorithms. Tech. Rept. Carnegie-Mellon University, Pittsburg, CMU-CS-79-124, 1979.

[13] L. Gross , J. Yellen. *Graph Theory and Its Applications.* CRC Press, USA, 2006.

[14] R. L. Hemminger, L. W. Beineke . Line graphs and line digraphs in *Selected topics in graph theory* 270-304, 1983.

[15] G. Fink, J. Fourier. *Operational Research and Networks.* Wiley-ISTE, 2008.

[16] A.C. Fisher, J.S. Liebman, G.L.Nemhauser Computer construction of project networks. *Commun. A.C.M.* 11(7):493497 , 1968.

[17] E. C. Freuder. Synthesizing constraint expressions. *Commun. A.C.M.* 21(11):958-965, 1978.

[18] S. Hanks, D.V. McDermott Default reasoning, nonmonotonic logics, and the frame problem. in *Proceedings AAAI* philadephia, 1986.

[19] M. Hayes. *The role of activity precedence relationships in node-orientated networks, in: Project Planning by Network Analysis.* North-Holland Publishing company, 1969.

[20] W. Herroelen, R. Leus. Project scheduling under uncertainty: Survey and research potentials. *European journal of operational research.* 165(2):289-306, 2005.

[21] J.E. Kelley Critical path planning and scheduling-Mathematical basis. *Oper. Res.* 9(3):296-320, 1961.

[22] M.S. Krishnamoorty , N. Deon. Complexity of minimum-Dummy-Activities Problem in a Pert Network. *Networks* 9:189-194, 1979.

[23] J. Malik, J.O.Inford. Reasoning in time and space. *Proceedings IJCAI* 83:343-345, 1983.

[24] D.V. McDermott. A temporal logic for reasoning about process and plan. *Cogn. Sci* 6:101-155, 1983.

[25] D.J. Michael, J.D. Kamburowski, M. Stallmann. On the minimum dummy-arc problem. *Recherche Operationnelle RAIRO* 27(2):153-158, 1996.

[26] U. Montanari. Networks of constraints: fundamental properties and applications to picture processing. *Inf. Sci* 7:95-132, 1974.

[27] N. E.Mouhoub , H. Belouadah, A. Boubetra. Algorithme de construction de graphe PERT partir d'un graphe des potentiels donn. *Studia Univ. Babes Bolyai, Informatica*51(2)3:40-348, 2006.

[28] N. E. Mouhoub , A. Benhocine , H. Belouadah. A new method for constructing a minimal PERT network. *(APM) Applied Mathematical modeling* 35(9):4575-4588, 2011.

[29] N. E. Mouhoub , A. Benhocine. An efficient algorithm for generating AoA networks. *IJCSI* 9(1):114-120, 2012.

[30] B. Nebel, H. J. Buckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of ACM* 42(1):43- 66, 1995.

[31] P. Shah, E. G. Freedman . Bar and Line Graph Comprehension: An Interaction of Top-Down and Bottom-Up Processes. *Topics in cognitive science* 3(3):560-578, 2009.

[32] Shoham Y. *Reasoning about change: Time and causation from the stand point of artificial intelligence, In A new heuristic algorithm in resource constrained project scheduling problem with time windows.* Cambridge, 1969.

[33] T.B. Smith, J.M. Pyle. An effective algorithm for project scheduling with arbitrary temporal constraints. American association for artificial intelligence, 189-198, 1988.

[34] F. Sterboul, D.Wertheimer. Comment construire un graphe PERT minimal. *RAIRO - Operations Research* 15:85-98, 1980.

[35] W. Stevenson, C. Ozgur, A. Nsakanda. *Introduction to management science with spreadsheets.* McGraw-Hill Ryerson Higher Education, 2013.

[36] C-M. Tsou. On the project management scheduling based on agent technology and theory of constraint. *International Journal of Electronic Business Management* 10(4):286-295, 2012.

[37] M. Vanhoucke. *Project management with dynamic scheduling: Baseline scheduling, In Risk analysis and Project control.* Springer, 2005.