A new linear program for QoS-aware web service composition based on complex workflow

Virginie Gabrel, Maude Manouvrier, Cecile Murat

Laboratoire d'Analyses et Modélisation de Systèmes pour l'Aide à la Décision
UMR 7243

CAHIER DU LAMSADE

339

Mai 2013

A new linear program for QoS-aware web service composition based on complex workflow

V. Gabrel, M. Manouvrier, C. Murat

A new linear program for QoS-aware web service composition based on complex workflow

Virginie Gabrel, Maude Manouvrier, and Cécile Murat
PSL, Université Paris Dauphine, LAMSADE CNRS UMR 7243, F-75016 Paris, France
Email: {gabrel,manouvrier,murat}@lamsade.dauphine.fr

***Abstract*—In this article, we propose a new model based on mixed linear programming to determine a composite web service (structured by a workflow) minimizing a QoS measure while satisfying some QoS constraints. The proposed mixed linear program is solved using a standard solver (CPLEX). Our experiments show that big-size instances can be exactly solved. To the best of our knowledge, it is the first time that a linear program with a polynomial number of variables and constraints is proposed for QoS-driven web service composition based on complex workflow with interlaced patterns.**

I. Web service composition

Web services (WS) are the most famous implementation of service oriented architectures (SOA) [1] allowing the construction and the sharing of independant and autonomous softwares [2]. WS composition consists in combining several existing WS into a composite one, which becomes a value-added process [3]. The WS composition problem aims at identifying a set of existing WS such that the composition of those WS can satisfy the user's functional and non-functional requirements [4]. To differentiate several WS having the same functionality, QoS criteria can be used to select the "best" WS satisfying the users' requirements.

The following sub-sections describe our context: the workflow structure of the composite WS and the QoS criteria. Then, some related works are studied. Section II presents and justifies our new mixed linear programming model. Section III presents some very promising experimental results that show the relevance of our model. Finally, Section IV concludes.

A. Workflow

A workflow describes how to combine the functionalities of different WS in order to satisfy the user. In a workflow, an activity represents a set of WS sharing a same functionality, and a pattern represents temporal dependency between different activities. In this article, we consider three patterns: sequence, parallel (AND) and exclusive choice (XOR). Figure 1 represents these workflow patterns, based on YAWL model [5], where Ai is an activity. The sequential pattern, see Figure 1.(a), indicates that $A1$ must be executed before $A2$. The XOR and AND patterns start with a split and finish with a join. In AND pattern, see Figure 1.(b), all activities $A1, \ldots, Ak$ have to be executed in parallel. For XOR pattern, see Figure 1.(c), only one activity among $A1$ to Ak has to be executed.

In the following, we consider general complex workflows in which these patterns can be recursively concatened and interlaced (an example of such a workflow is given in Figure 3). A pattern can be decomposed on several branches (each branch being a workflow containing several activities linked by patterns). For XOR pattern (see Figure 2), one and only one branch must be selected. For AND pattern, all branches must be performed.

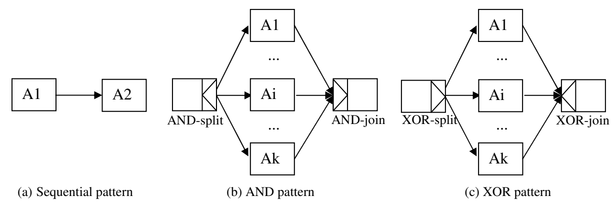

Fig. 1. Workflow patterns

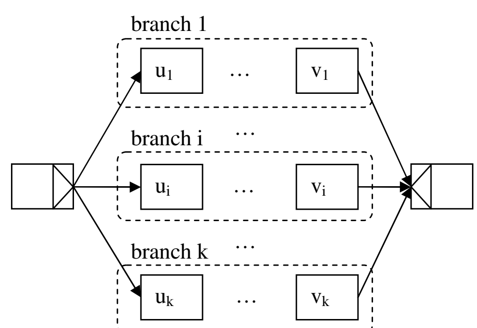

Fig. 2. Complex XOR pattern with interlaced patterns

The first vertex u_i of branch i can be an activity or a split, the last vertex v_i can be an activity or a join.

In a complex workflow, an end-to-end route is a path going from the first vertex of the workflow to the last one containing all branches of each AND pattern belonging to the path and, containing exactly one branch of each XOR pattern belonging to the path.

Therefore, the composition problem is simultaneously the problem of:
- selecting an end-to-end route,
- and choosing a WS for performing each activity belonging to the selected end-to-end route.

Several WS, with similar functionality, permit to perform the same activity. These WS may have different Quality of

Service (QoS) levels. These QoS criteria represent the non-functional properties of web services.

B. QoS criteria

In a large majority of studies (e.g. [6], [7]), the QoS criteria are: execution price, execution duration/response time, reputation, reliability (the probability that its execution performs successfully) and availability. The way to compute the score of a composite WS depends on the type of the QoS criterion and on the pattern. For some criteria, the score of a composite web service is the sum of the web services' values belonging to the composition (e.g. price), for others, the score is based on a max operator (e.g. time for WS executed in parallel), otherwise, the score is the multiplication of the web services' values belonging to the composition (e.g reliability).

In the following, we consider that each WS is evaluated on three criteria:

- the first criterion is the cost of a WS j denoted c_j,
- the second criterion is the duration of a WS j denoted d_j,
- the third is the reliability of a WS j, denoted p_j.

We choose these three criteria because they represent different ways of score computing: the first criteria is a sum-type measure to be minimized, the second is a max-type measure to be minimized and the third one is a multiplication-type measure to be maximized.

C. Related works on QoS-aware web service composition

Several recent articles propose different algorithms for QoS-driven WS composition based on workflow (see [8], [9] for recent surveys). The QoS criteria are used to define an overall QoS score and some constraints to satisfy (e.g. the response time has to be less than 5 seconds). The constraints, representing the user's requirements, are expressed in terms of upper or lower bounds on the QoS scores.

The problem difficulty depends on the type of the considered workflow. When the input workflow only contains sequential patterns, the problem is represented as a Multiple Choice Knapsack Problem [10]. In [11], a heuristic is proposed to select a web service for each activity so that the utility provided by the QoS attributes of the composition is maximised subject to constraints defined by the user.

When several patterns are considered in the input workflow, determining the composite WS that maximizes the QoS criteria subject to QoS constraints, becomes a more complex problem. One of the difficulties comes from the XOR pattern which implies a choice between the activities to be done.

This difficulty no longer exists (in [12] and [13]) when the considered workflow includes the probability for executing each XOR branch. The problem remains then to select one WS per activity (some activities being chosen with a known probability). Thus, the QoS criteria cannot be exactly computed but an average-case analysis is performed (authors propose also to perform a best-case and a worst-case analysis). In this context, the problem can be represented with an integer linear program. In [14] the authors extend their model to new patterns. To address scalability issues, they propose an approximate algorithm.

Without considering probability for executing each XOR branch, the problem of WS selection concerns both the selection of the activities (included in XOR branch) and the selection of one WS by chosen activity. In [15], the proposed model contains as many constraints as end-to-end routes in the workflow. This model is used to solve some instances but, as the number of end-to-end routes exponentially grows with the workflow size, some large-size instances may not be formulated and even less solved. Thus, authors propose to find near-optimal solutions in polynomial time with an approximate algorithm. In [7], authors also claim that all end-to-end routes of the workflow must be generated in order to find the best composite WS. So, they propose to decompose the problem into two steps: in the first one, an end-to-end route is chosen and, in the second one, they propose a MIP formulation to select one WS per activity belonging to the chosen route in order to maximize QoS criteria. With such a decomposition, the obtained solution is not necessarily optimal.

The real challenge is to include into the same optimization problem the selection of activities and the selection of WS for performing each activity structured by a complex workflow.

This work follows the study presented in [16]. In [16], we have proposed a 0-1 linear model for representing the composition problem based on a complex workflow satisfying transactional properties (without QoS constraints).

We propose an extension of the model presented in [16] for exactly solving the QoS-aware WS composition problem by simultaneously considering the choice of activities and web services. In this new version, we address complex workflows with global QoS constraints, the objective being to minimize a sum-type criterion. More precisely, our objective is to minimize the cost criterion and to satisfy two QoS constraints concerning execution time and reliability. Our model is very interesting since it avoids the enumeration of all end-to-end routes and the number of variables and constraints is a linear function of the workflow size.

In the section II, we present our model by introducing data, decision variables and constraints. The section III is devoted to experiments showing reasonable execution times.

II. Mathematical programming model

In this section, we present our mixed linear programming-based model for determining a composite WS (structured by a complex workflow) minimizing a QoS measure and satisfying QoS constraints.

A. Data

The inputs of our model are:

1) a workflow which is a graph $G = (V, E)$, describing the execution order of a set A of n activities and the patterns (sequence, AND and XOR) combining them,
2) for each activity A_i, a list W_i of $|W_i|$ web services which can perform activity A_i (we denote by $W = \cup_{i=1}^{n} W_i$ the set of WS with total cardinality $|W|$),

3) for each WS j, its values on each criteria represented by a vector (c_j, d_j, p_j).

As usually, $\forall v \in V$, $\Gamma^+(v)$ represents the set of direct successors of vertex v and $\Gamma^-(v)$ the set of direct predecessors of vertex v.

The vertices of G represent the activities and the AND and XOR patterns of the workflow. Each AND (resp. XOR) pattern is represented by two vertices u and v, where u represents the split of the pattern and v represents the join. Figure 3 gives an example of such a graph with the corresponding numbered vertices. For each vertex v which represents a join of AND or XOR pattern, let s be a function which returns the vertex number corresponding to the split. For example, in Figure 3, $s(12) = 7$.

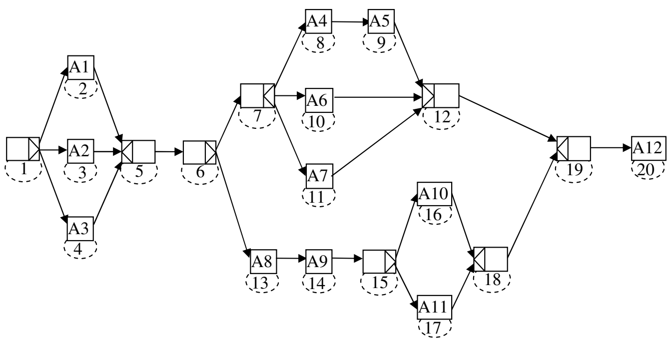

Fig. 3. Example of graph associated with a workflow

Let us recall that a composite WS is associated to an end-to-end route. For example, in Figure 3 the candidate end-to-end route $\{1, 3, 5, 6, 7, 8, 9, 10, 11, 12, 19, 20\}$ implies that the activities $A_2, A_4, A_5, A_6, A_7, A_{12}$ must be executed (one WS per activity must be chosen). In this solution, A_6, A_7 and (A_4, A_5) can be performed in parallel.

The set V of vertices can be partitioned into 5 subsets: V_A, V^S_{AND}, V^J_{AND}, V^S_{XOR} and V^J_{XOR}.

- V_A is the set of vertices corresponding to the activities. Given $v \in V_A$, let $a(v)$ be the indice of the activity in the workflow associated with v. By convention, $a(v) = 0$ if v is not associated with an activity and we define an empty set of WS for this fictitious activity ($W_0 = \emptyset$). For example vertex 4 is associated with activity A_3 then $a(4) = 3$,
- V^S_{AND} (resp. V^S_{XOR}) is the set of vertices corresponding to the split of AND (resp. XOR) patterns,
- V^J_{AND} (resp. V^J_{XOR}) is the set of vertices corresponding to the join of AND (resp. XOR) patterns.

The set E of arcs can be partitioned into 3 subsets: E_{SEQ}, E^S and E^J.

- E_{SEQ} is the set of arcs (u, v) corresponding to sequence patterns, where $u \in V_A \cup V^J_{AND} \cup V^J_{XOR}$ and $v \in V_A \cup V^S_{AND} \cup V^S_{XOR}$,
- $E^S = E^S_{AND} \cup E^S_{XOR}$ is the set of arcs (describing split operator) which initial extremity belongs to $V^S_{AND} \cup V^S_{XOR}$,
- $E^J = E^J_{AND} \cup E^J_{XOR}$ is the set of arcs (describing join operator) which final extremity belongs to $V^J_{AND} \cup V^J_{XOR}$.

B. Decision variables for selecting activities and WS

As previously explained, our WS composition problem is to select simultaneously the activities to be performed and one WS for each chosen activity. For the selection of activities, we introduce 3 types of variables:

1) $\forall v \in V_A$, x_v is a binary variable equal to 1 if the activity with index $a(v)$ is performed,
2) $\forall v \in V^J_{AND} \cup V^S_{AND}$, x_v is a binary variable equal to 1 if the join (or split) operator of the AND pattern represented by v is successful (meaning that all activities inside the AND pattern are performed) and 0 otherwise,
3) $\forall v \in V^J_{XOR} \cup V^S_{XOR}$, x_v is a binary variable equal to 1 if the join (or split) operator of the XOR pattern represented by v is successful (meaning that only one branch inside the XOR pattern is performed) and 0 otherwise.

Any end-to-end route is represented by a 0-1 vector $x = (x_v)_{v=1,\ldots,|V|}$. For the selection of WS, we introduce the following variables: $\forall A_i \in A$ and $\forall j \in W_i$, w_{ij} is a binary variable equal to 1 if the activity A_i is performed by the WS j and 0 otherwise.

C. Constraints for selection

We have to modelize the fact that each selected activity must be realized by exactly one WS and no WS must be retained for non-selected activities.

(C1) (WS selection) For each vertex v belonging to V_A, we have to select at most one WS for the activity $a(v)$:

$$x_v = \sum_{j \in W_{a(v)}} w_{a(v)j}$$

Moreover, the WS composition satisfies the user's functional requirement if the last operation has been successfully completed.

(C2) (Ending activity or operator) For the last vertex of indice $|V|$, we have:

$$x_{|V|} = 1$$

This constraint can be seen as an objective to achieve.

The constraints induced by the workflow, presented in the following section, describe the patterns (and the corresponding activities) that must be selected in order to satisfy constraint (C2).

D. Constraints induced by the workflow

We can represent the workflow constraints in a linear form.

(C3) (Sequential pattern) For each arc $(u, v) \in E_{SEQ}$, if the pattern is selected then $x_u = x_v = 1$, if it's not the case $x_u = x_v = 0$. So we have:

$$x_u = x_v$$

(C4) (AND pattern) For each vertex $v \in V^J_{AND}$, the corresponding AND join pattern v is performed if and only if the last vertex u of each branch is selected:

$$x_v = x_u \ \forall u \in \Gamma^-(v)$$

(C5) (XOR pattern) For each vertex $v \in V^J_{XOR}$, if $x_v = 1$ then only one branch has to be selected and, if $x_v = 0$, then none branch is selected. Considering the last vertex u of each branch, these two cases are represented by:

$$x_v = \sum_{u \in \Gamma^-(v)} x_u$$

(C6) (SPLIT-JOIN) For each AND (resp. XOR) pattern associated with vertices u and v (with $u = s(v)$), we have:

$$x_v = x_u$$

Example 1: Applied to the example of Figure 3, we have the following data: $V_A = \{2, 3, 4, 8, 9, 10, 11, 13, 14, 16, 17, 20\}$, $V^S_{AND} = \{7\}$, $V^J_{AND} = \{12\}$, $V^S_{XOR} = \{1, 6, 15\}$ and $V^J_{XOR} = \{5, 18, 19\}$. Values of functions a are represented in Table I (only for v such that $a(v) \neq 0$) and s in Table II (only for v such that $s(v) \neq 0$).

TABLE I
FUNCTION a APPLIED TO THE EXAMPLE

v	2	3	4	8	9	10	11	13	14	16	17	20
$a(v)$	1	2	3	4	5	6	7	8	9	10	11	12

TABLE II
FUNCTION s APPLIED TO THE EXAMPLE

v	5	12	18	19
$s(v)$	1	7	15	6

Then, we have the following constraints induced by the workflow:

$$\begin{aligned}
(C2):&\quad x_{20} = 1\\
(C3):&\quad x_6 = x_5 \qquad x_9 = x_8 \qquad x_{14} = x_{13}\\
&\quad x_{15} = x_{14} \qquad x_{20} = x_{19}\\
(C4):&\quad x_{12} = x_9 \qquad x_{12} = x_{10} \qquad x_{12} = x_{11}\\
(C5):&\quad x_5 = x_2 + x_3 + x_4\\
&\quad x_{19} = x_{12} + x_{18} \qquad x_{18} = x_{16} + x_{17}\\
(C6):&\quad x_5 = x_1 \qquad x_{19} = x_6 \qquad x_{12} = x_7 \qquad x_{18} = x_{15}.
\end{aligned}$$

E. The Qos measures

Concerning the value of a composite WS w represented by the w_{ij} variables, each criterion induces a specific analysis. For the first criterion which is a sum-type criterion, we have :

$$c(w) = \sum_{i=1}^{n} \sum_{j \in W_i} c_{ij} w_{ij}$$

For the second criterion, which is a max-type criterion, the analysis is much complex since some WS can be executed in parallel inside a same AND pattern. Indeed, in order to compute the execution time of a given AND pattern, we have to compute the execution time of each branch of the pattern (which is the sum of the execution time of the WS chosen for each activity belonging to the branch). Then, the execution time of the considering AND pattern is equal to the maximal branch execution time.

In our model, we introduce the following additional variables : t_v represents the starting time of the vertex v, $\forall v \in V$. The ending time of vertex v is obtained by adding to t_v its execution time depending on the type of v:

- the execution time of a vertex representing an activity $a(v)$ is computed as follows : $\sum_{j \in W_{a(v)}} d_{a(v)j} w_{a(v)j}$ (when the activity $a(v)$ is not performed, all variables $w_{a(v)j}$ are equal to 0 and the execution time is 0),
- the execution time of a vertex representing a pattern is equal to 0.

We have to introduce a set of constraints in order to compute the starting time variables.

To begin, we define the starting time of the first vertex of the workflow :

(T0) $t_1 = 0$.

To force $t_v = 0$ when the vertex v is not selected, we introduce the following constraint:

(T1) $\forall v \in V \setminus \{1\},\ \ t_v \leq M x_v$, with M a big constant which does not limit the t values. We propose to take M equal to $M = \sum_{i \in A} \max_{j \in W_i} d_{ij}$

For the sequential patterns, we introduce the following constraint :

(T2) $\forall (u, v) \in E_{SEQ}\ \ t_v \geq t_u + \sum_{j \in W_{a(u)}} d_{a(u)j} w_{a(u)j}$ with the convention that $a(u) = 0$ and $W_0 = \emptyset$ when u is a pattern. Thus, when u represents a pattern, the inequality becomes $t_v \geq t_u$.

Example 2: In the example of Figure 3, for each activity we consider three WS having durations given in Table III.

TABLE III
WS DURATIONS FOR THE EXAMPLE

v	2	3	4	8	9	10	11	13	14	16	17	20
$Ai = a(v)$	1	2	3	4	5	6	7	8	9	10	11	12
$d_{i1} = d_{a(v)1}$	4	3	4	1	1	2	3	2	3	8	7	3
$d_{i2} = d_{a(v)2}$	3	4	4	1	2	1	4	3	4	9	6	5
$d_{i3} = d_{a(v)3}$	5	2	5	2	3	3	2	1	2	7	8	6

For arc $(8, 9)$, the constraint (T2) is:

$$t_9 \geq t_8 + w_{a(8)1} + w_{a(8)2} + 2w_{a(8)3}$$

For arc $(5, 6)$, constraint (T2) becomes: $t_6 \geq t_5$.

For the AND patterns, we introduce the following constraints :

(T3) the first vertex v of each branch can begin when the split represented by u is performed,

$$\forall u \in V^S_{AND},\ \forall v \in \Gamma^+(u),\ \ t_v \geq t_u$$

(T4) the join pattern v can be performed when the last activity of each branch is ended,

$$\forall v \in V^J_{AND},\ \forall u \in \Gamma^-(v),\ \ t_v \geq t_u + \sum_{j \in W_{a(u)}} d_{a(u)j} w_{a(u)j}$$

Example 3: In our example, only $u = 7$ is a split AND pattern, then (T3) induces the constraints:
$t_8 \geq t_7 \quad t_{10} \geq t_7 \quad t_{11} \geq t_7$
The vertex 12 is the only join AND pattern, then (T4) induces the constraints:
$t_{12} \geq t_9 + w_{a(9)1} + 2w_{a(9)2} + 3w_{a(9)3}$
$t_{12} \geq t_{10} + 2w_{a(10)1} + w_{a(10)2} + 3w_{a(10)3}$
$t_{12} \geq t_{11} + 3w_{a(11)1} + 4w_{a(11)2} + 2w_{a(11)3}$

For a split XOR pattern u, only one branch is executed. Then, for the first activity v of each branch, we have to satisfy the non-linear constraint:

$$\forall v \in \Gamma^+(u),\ \ t_v x_v \geq t_u$$

From (C5), only one branch is selected. Let us denote v' the first activity of the selected branch. We have: $\sum_{v \in \Gamma^+(u)} t_v x_v = t_{v'} x_{v'}$. Constraint (T1) implies that $t_v = 0$, $\forall v \neq v'$ thus the previous expression can be linearized as follows: $\sum_{v \in \Gamma^+(u)} t_v x_v = \sum_{v \in \Gamma^+(u)} t_v$.
For the XOR patterns, we introduce the following constraints:

(T5) the starting time of the first vertex of the selected branch can be obtained as follows,

$$\forall u \in V^S_{XOR} \quad \sum_{v \in \Gamma^+(u)} t_v \geq t_u$$

(T6) for the ending time of the patterns,

$$\forall v \in V^J_{XOR}\ \ t_v \geq \sum_{u \in \Gamma^-(v)} (t_u + \sum_{j \in W_{a(u)}} d_{a(u)j} w_{a(u)j})$$

Example 4: Let us consider the vertex $u = 6$ which is a split XOR pattern. Constraint (T5) gives: $t_7 + t_{13} \geq t_6$
The join vertex associated being $v = 19$, constraint (T6) gives: $t_{19} \geq t_{12} + t_{18}$

Finally, we have to introduce a global constraint to limit the total duration time to T. Since we set $t_1 = 0$, the total execution time of the composite WS is given by:
$t_{|V|} + \sum_{j \in W_{a(|V|)}} d_{a(|V|)j} w_{a(|V|)j}$
Consequently, the global constraint on time is:

(Q1) $t_{|V|} + \sum_{j \in W_{a(|V|)}} d_{a(|V|)j} w_{a(|V|)j} \leq T$

For the third criterion, which is a multiplicative-type criterion, we apply the same approach as in [7]. The global reliability of a composite WS w, denoted $r(w)$, is given by
$r(w) = \prod_{(i=1,\dots,n,\ j \in W_i):w_{ij}=1} p_{ij}$

If we want to guarantee a reliability of probability P for the selected composition, then we have to add: $r(w) \geq P$. The linearization of this expression is natural by considering the log function:
$\log(r(w)) = \sum_{(i=1,\dots,n,\ j \in W_i):w_{ij}=1} \log(p_{ij}) \geq log(P)$
Since $\sum_{(i=1,\dots,n,\ j \in W_i):w_{ij}=1} \log(p_{ij}) = \sum_{i=1}^{n} \sum_{j \in W_i} log(p_{ij}) w_{ij}$
the global constraint on reliability is:

(Q2) $\sum_{i=1}^{n} \sum_{j \in W_i} log(p_{ij}) w_{ij} \geq log(P)$

The constraint (Q1) represents the fact that the execution time is limited by T. The constraint (Q2) represents the fact that the reliability must be at least equal to P. Parameters T and P are given by the user.

The following section presents the final model.

F. The model

Our purpose is to minimize the cost criterion and to satisfy two QoS constraints concerning the execution time and the reliability.

$$\left\{\begin{array}{lr}
\min \sum_{i=1}^{n} \sum_{j \in W_i} c_{ij} w_{ij} & \\
x_v = \sum_{j \in W_{a(v)}} w_{a(v)j} \qquad \forall v \in V_A & (C1)\\
x_{|V|} = 1 & (C2)\\
x_u = x_v \qquad \forall (u,v) \in E_{SEQ} & (C3)\\
x_u = x_v \qquad \forall (u,v) \in E^J_{AND} & (C4)\\
\sum_{u \in \Gamma^-(v)} x_u = x_v \qquad \forall v \in V^J_{XOR} & (C5)\\
x_u = x_v \qquad \forall v \in V^J_{AND} \cup V^J_{XOR} \text{ with } u = s(v) & (C6)\\
t_1 = 0 & (T0)\\
t_u \leq M x_u \qquad \forall u \in V \setminus \{1\} & (T1)\\
t_v \geq t_u + \sum_{j \in W_{a(u)}} d_{a(u)j} w_{a(u)j} \qquad \forall (u,v) \in E_{SEQ} & (T2)\\
t_v \geq t_u \qquad \forall (u,v) \in E^S_{AND} & (T3)\\
t_v \geq t_u + \sum_{j \in W_{a(u)}} d_{a(u)j} w_{a(u)j} \qquad \forall (u,v) \in E^J_{AND} & (T4)\\
\sum_{v \in \Gamma^+(u)} t_v \geq t_u \qquad \forall u \in V^S_{XOR} & (T5)\\
t_v \geq \sum_{u \in \Gamma^-(v)} (t_u + \sum_{j \in W_{a(u)}} d_{a(u)j} w_{a(u)j})\ \ \forall v \in V^J_{XOR} & (T6)\\
t_{|V|} + \sum_{j \in W_{a(|V|)}} d_{a(|V|)j} w_{a(|V|)j} \leq T & (Q1)\\
\sum_{i=1}^{n} \sum_{j \in W_i} \log(p_{ij}) w_{ij} \geq log(P) & (Q2)\\
w_{ij} \in \{0,1\} \qquad \forall i = 1,\dots,n\ \forall j = 1,\dots,|W_i| & \\
x_v \in \{0,1\} \qquad \forall v = 1,\dots,|V| & \\
t_v \geq 0 \qquad \forall v = 1,\dots,|V| &
\end{array}\right.$$

Example 5: For the example of Figure 3 and Table III, we associate the following costs: let us recall that 3 WS are associated to each activity $i = 1,\dots,12$, $c_{i1} = 4$, $c_{i2} = 3$ and $c_{i3} = 2$. To simplify, we assume that the reliability of each

WS is equal to 1. In order to determine the less expensive composite WS, we have to solve our linear program by setting $M = T = \sum_{i=1}^{12} \max_{j \in \{1,2,3\}} d_{ij} = 56$.

An optimal solution is given by the following end-to-end route $\{1, 3, 5, 6, 13, 14, 15, 17, 18, 19, 20\}$ for which the third WS of each activity $A_2, A_8, A_9, A_{11}, A_{12}$ is selected. The cost of the optimal solution is 10.

If the response time T has to be less than 12 then the optimal solution corresponds to another end-to-end route $\{1, 3, 5, 6, 7, 8, 9, 10, 11, 12, 19, 20\}$ for which the third WS of each activity $A_2, A_4, A_6, A_7, A_{12}$ is selected, and the second WS of activity A_5 is performed.

The number of binary variables is equal to $|V| + |W|$, plus $|V|$ real and non negative variables which gives $O(|W|)$ variables. The number of constraints is equal to $2n+3+2|E_{SEQ}|+3|E^J_{AND}|+3|V^S_{XOR}|+|V^J|$ which gives $O(|E|)$ constraints. Indeed, we have n constraints for $(C1)$, $|E_{SEQ}|$ constraints for $(C3)$ and $(T2)$, $|E^J_{AND}| = |E^S_{AND}|$ constraints for $(C4)$, $(T3)$ and $(T4)$, $|V^J_{XOR}| = |V^S_{XOR}|$ for $(C5)$, $(T5)$ and $(T6)$ and $|V^J|$ pour $(C6)$.

Let us underline that our model can represent several variants of the composition problem: minimize the execution time subject to a budget constraint, maximize an aggregate score subject to several QoS constraints... In practice, it is of interest to know the best value we can obtain on each criterion. For that, by changing the objective function of our model, we can minimize total execution time (without any QoS constraint) or minimize total cost (without time constraint).

Example 6: For the previous example, by minimizing $t_{20} + 3w_{12,1} + 5w_{12,2} + 6w_{12,3}$ and removing $(Q1)$ and $(Q2)$, we obtain 7 which is the minimal duration of a composite WS (with a total cost equals to 19).

III. Experimentations

The model was solved using CPLEX solver, and the experiments were carried out on a Dell PC with Intel (R) Core TM i7-2760, with 2,4 Ghz processor and 8 Go RAM, under Windows 7 and Java 7. For testing purpose, we randomly generated workflows with interlaced XOR and AND patterns, by varying the number of activities (from 10 to 200), and by randomly varying the number of AND and of XOR patterns (representing at most 30% of the number of activities in the workflow) as well as their position in the workflow. The WS registery was generated by randomly assigning QoS values to each candidate WS (a value between 10 and 50 for the price and a value between 50 and 200 ms for the duration). For a given number of activities and a given number of WS per activity, 10 instances are randomly generated in order to compute average value.

In order to analyse the impact of one QoS constraint on computation time and on optimal solution value, we choose in a first step to relax the reliability constraint $Q2$: the reliability of each WS is equal to 1 and $P = 1$.

The Table IV reports the average size of considered linear programs. The number of constraints only depends on the workflow. As expected, the number of variables linearly increases with the number of WS per activity.

TABLE IV
Linear program size

n	Number of constraints	Number of variables for 10 WS	50 WS	100 WS	200 WS
10	55	131	531	1031	2031
50	269	648	2648	5148	10148
100	514	1285	5285	10285	20285

In the following experiments, for a given workflow and WS registery, we begin by determining the composite WS with minimal duration time, denoted T^*. We recall that the constraint $Q1$ is relaxed when the time T increases, inducing an expansion of the feasible solution set.

In order to analyse the impact of the constraint $Q1$ on computation time, we first consider 50-activities workflows with 100 WS per activity. In Figure 4, we report the computation time when T increases from T^* to $3T^*$. The Figure 4 clearly shows that "easy" cases (with computation times less than 9 ms) appear for high values of T (greater than $1.5T^*$). The computation times are higher (around 17 ms) when T varies from $1.1T^*$ to $1.3T^*$.

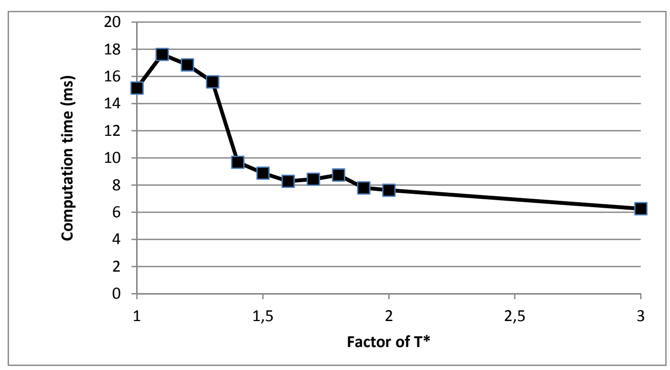

Fig. 4. Computation time in function of the QoS time constraint

In order to analyse the influence of the number of WS on the computation time, we first consider 50-activities workflows with an increasing number of WS (from 50 to 500 per activity). Then, we compute minimal cost composite WS in two cases: in the first case, the total duration mustn't exceed $1.1T^*$ and, in the second case, the duration musn't exceed $1.5T^*$ (the QoS constraint $Q1$ is less restrictive). The results are presented in the Figure 5.

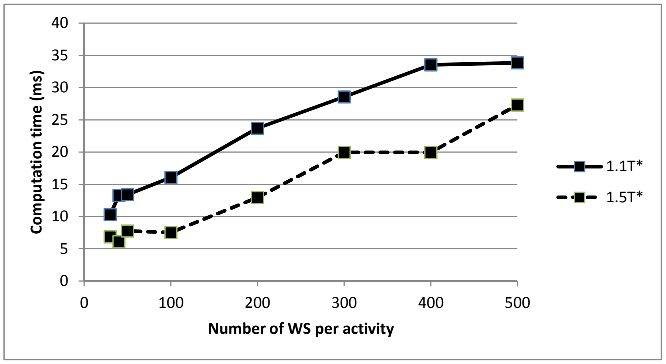

Fig. 5. Computation time for 50-activities workflows

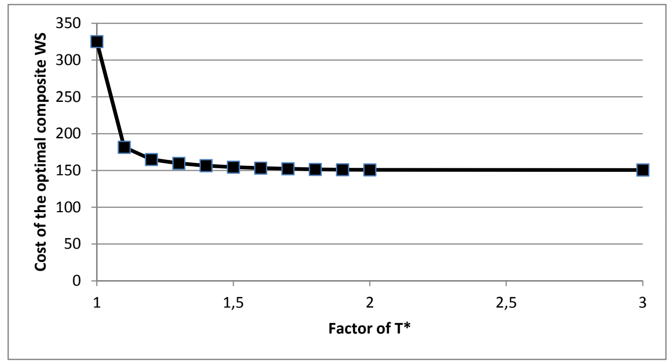

Fig. 7. Cost of the optimal composite WS in function of T

As previously noted, for a given number of WS, the optimal solution is found more rapidly when the constraint on time is "relaxed" (equals to $1.5T^*$). For realistic instances, around 50 WS per activity, the average computation time (obtained on 10 workflows) is less than 7 ms when T equals $1.5T^*$, and less than 13 ms when T equals $1.1T^*$. These experimental results are very promising. Moreover, it appears that the execution time linearly increases with the number of WS.

Afterwards, we consider different workflows with an increasing number of activities (with 100 WS per activity). The Figure 6 presents the evolution of computation times for activities' number varying from 10 to 200.

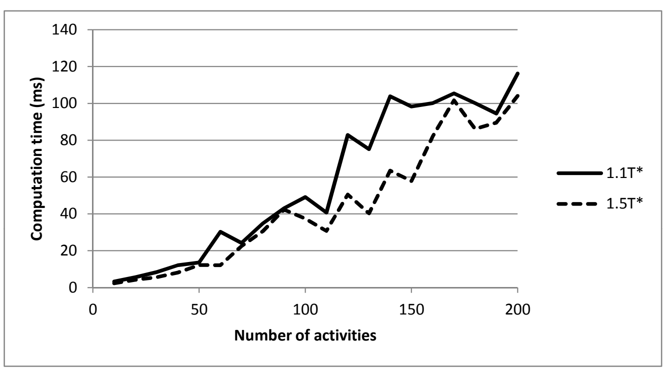

Fig. 6. Computation time for 100 WS per activity

We note that the computation time linearly increases with the number of activities. Moreover, the computation time is of the same order for the two considered values of T.

The Figure 7 shows the link between the cost of the optimal composite WS and the time constraint for 50-activities workflows with 100 WS per activity.

In Figure 7, it clearly appears that the cost of the optimal composite WS rapidly decreases when the QoS time constraint is relaxed. More precisely, when T increases from T^* to $1.2T^*$, the cost of the optimal solution is divided by 2. From T equals to $1.5T^*$, the minimal cost of the optimal composite WS is reached in a very large majority of cases.

Finally we re-introduce the reliability constraint $Q2$ to our model. The score of each WS on the reliability criterion is randomly chosen in the range $[0, 1]$ and we set $P = 0.5$. Our experiments, reported in Figure 8, show that the linear program is more difficult to solve with two QoS constraints. Indeed, the computation time exponentially increases with the number of WS. However, it is still reasonable for realistic instances (an average of 100 ms for a 50-activities workflow with 50 WS per activity).

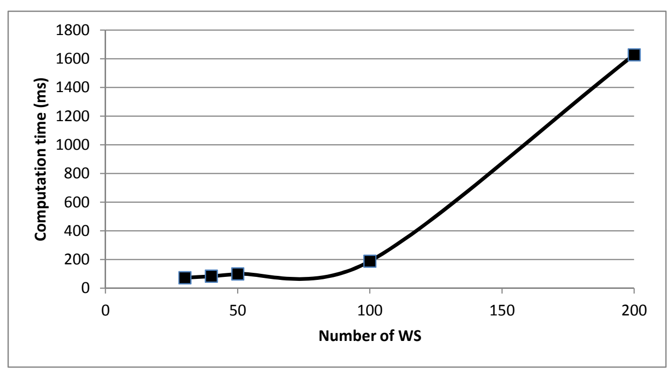

Fig. 8. Computation time for 50-activities workflow with $T = 1.5T^*$ and $P = 0.5$

These computational experiments are very promising. With the proposed model, the WS composition problem with a sum-type criterion to optimize and a single QoS constraint is tractable even for large size instances. However, the same problem with an additional QoS constraint is much more difficult to solve for large size instances with a standard solver like Cplex.

IV. Conclusion

In this article, we present a mixed linear program for determining a composite WS minimizing a sum-type criterion subject to QoS constraints. This model allows to simultaneously optimize the activities and the chosen WS per activity in a workflow including interlaced XOR and AND patterns. Compared with graph-based approaches [15], we do not have to enumerate all the potential end-to-end routes in the workflow. Consequently, the number of variables and constraints of our proposed model are polynomial in function of the number of WS and patterns. Extensive computational experiments on random workflows and WS registries show that our approach is very promising for exactly solving the QoS-aware composition problem.

However, the computation time is more important when several QoS constraints are considered. It takes several seconds to find the optimal solution for big size instances and this may be too long for real time computation. In this case, it would be useful to propose approximate algorithms to find solutions very rapidly. With our linear programming approach, we are able to determine the optimal solution, to compute the distance between the optimal solution and the approximate one and thus to obtain an experimental approximation ratio.

Finally, web services are suppose to be independent in our study. We would like to extend our model in order to take into account the correlation between web services (generating compatibility or exclusion constraints, or QoS modifications). This will be the subject of future research.

ACKNOWLEDGMENT

This work is supported by the *Action de Recherche Fondamentale en Recherche Opérationnelle (ARFRO)* of the French Operational Research Group (GDR RO).

REFERENCES

[1] T. Erl, *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall, 2005.

[2] J. E. Haddad, M. Manouvrier, and M. Rukoz, "TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition," *IEEE Trans. on Services Computing*, vol. 3, no. 1, pp. 73–85, 2010.

[3] Y. Cardinale, J. E. Haddad, M. Manouvrier, and M. Rukoz, *Handbook of Research on Non-Functional Properties for Service-oriented Systems : Future Directions*. IGI Global, 2011, ch. Transactional-aware Web Service Composition: A Survey.

[4] L. Cui, S. Kumara, and D. Lee, "Scenario analysis of web service composition based on multi-criteria mathematical goal programming," *Service Science*, 2011 (to appear).

[5] "Yawl: Yet another workflow language," retrieved Oct. 11, 2011, from http://www.yawlfoundation.org/.

[6] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for qos-based web service composition," in *Proc. of the 19th Int. Conf. on World wide web (WWW)*, 2010, pp. 11–20.

[7] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.

[8] A. Strunk, "Qos-aware service composition: A survey," in *Proceedings of the 2010 Eighth IEEE European Conference on Web Services*, ser. ECOWS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 67–74.

[9] J. E. Haddad, "Optimization techniques for qos-aware workflow realization in web services context," in *Resource Discovery Lecture Notes in Computer Science Volume 6799*, Z. Lacroix and M. Vidal, Eds., 2012, pp. 134–149.

[10] T. Yu and K.-J. Lin, "Service selection algorithms for web services with end-to-end qos constraints," *Information Systems and e-Business Management*, vol. 3, pp. 103–126, 2005.

[11] R. Berbner, M. Spahn, N. Repp, O. Heckmann, and R. Steinmetz, "Heuristics for QoS-aware Web Service Composition," in *ICWS '06: Proceedings of the IEEE International Conference on Web Services*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 72–82.

[12] D. Schuller, J. Eckert, A. Miede, S. Schulte, and R. Steinmetz, "Qos-aware service composition for complex workflows," in *Proceedings of the Fifth International Conference on Internet and Web Applications and Services (ICIW 2010), Barcelona, Spain*. IEEE, May 2010, pp. 333–338.

[13] D. Schuller, A. Miede, J. Eckert, U. Lampe, A. Papageorgiou, and R. Steinmetz, "Qos-based optimization of service compositions for complex workflows." in *ICSOC*, ser. Lecture Notes in Computer Science, P. P. Maglio, M. Weske, J. Yang, and M. Fantinato, Eds., vol. 6470, 2010, pp. 641–648.

[14] D. Schuller, A. Polyvyanyy, L. Garca-Bauelos, and S.Schulte, "Optimization of complex qos-aware service compositions," in *Service-Oriented Computing*, ser. Lecture Notes in Computer Science, G. Kappel, Z. Maamar, and H. Motahari-Nezhad, Eds. Springer Berlin Heidelberg, 2011, vol. 7084, pp. 452–466.

[15] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end qos constraints," *ACM Transactions on the Web*, vol. 1, pp. 1–26, 2007.

[16] V. Gabrel, M. Manouvrier, I. Megdiche, and C. Murat, "A new 0-1 linear program for qos and transactional-aware web service composition," in *Proceedings IEEE Symposium on Computers and Communications (ISCC)*, 2012, pp. 845–850.