INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 00 (2017) 1–30
DOI: 10.1111/itor.12392

Information supply chain optimization with bandwidth limitations

Michael J. Hirsch[a] and Héctor Ortiz-Peña[b]

[a]*ISEA TEK, LLC, 620 Wymore Road, Suite 260, Maitland, FL 32751, USA*
[b]*CUBRC, Inc., 4455 Genesee Street, Suite 106, Buffalo, NY 14225, USA*
E-mail: mhirsch@iseatek.com [Hirsch]; hector.ortiz-pena@cubrc.org [Ortiz-Peña]

Received 27 July 2015; received in revised form 29 November 2016; accepted 10 December 2016

Abstract

Workflow management systems allow for visibility, control, and automation of many business processes. Recently, nonbusiness domains have taken an interest in the management of workflows, and the optimal assignment and scheduling of workflow tasks to users across a network. This research aims at developing a rigorous mathematical programming formulation of the workflow optimization problem. The resulting formulation is nonlinear, but a linearized version is produced. Three heuristics are developed to find solutions efficiently. Computational experiments are presented and analyzed, comparing solutions to the original linearized formulation with the three heuristics.

Keywords: workflow optimization; information supply chain; heuristics

1. Introduction

A workflow management system (WfMS) allows, in general, the control and assessment of the tasks (or activities) associated with a business process, defined in a workflow. A workflow is a model of a process, consisting of a set of tasks, users, roles, and a control flow that captures the interdependencies among tasks. The control flow can be defined explicitly by indicating precedence relationships among the tasks, or indirectly by the information requirements (e.g., documents, messages, etc.) in order to perform the tasks. WfMS has emerged as an important technology for automating business processes, drawing increasing attention from researchers.

One key aspect on WfMS is a good assignment and scheduling of tasks to users in a workflow. The research addressed here is concerned with information workflows subject to bandwidth limitations on the communication network. Information is generated by some tasks (i.e., produced as output) and consumed by other tasks (i.e., needed as input). There are potentially multiple information workflows, with multiple tasks, that need to be assigned to users. Users can take on certain roles, and tasks can only be performed by users with certain roles. The tasks themselves can have precedence relationships. Overall, the goal is to minimize the time at which the last task gets completed, that is, to minimize the makespan.

Joshi (2003) discussed the problem of workflow scheduling aiming to achieve cost reduction through an optimal assignment and scheduling of workflows. Each workflow was characterized by a unique due date and tardiness penalty. The problem was formulated as a mixed-integer linear program (MILP). The model assumed that the dependencies and precedence relationships among the workflows were deterministic and unique. Tasks were not preemptive and the processing times and due dates were deterministic and known. Users could assume several roles but perform only one task at a time. The total cost component that the model minimized considered two elements: *processing cost* and *tardiness penalty cost*. A branch-and-price approach was proposed to solve the problem. An algorithm based on simulated annealing was evaluated by Kuipers (2003) for the optimal selection, scheduling and direction of strips (i.e., a set of tasks for optical, radar, or infrared photographs of the earth) to be acquired by a satellite. Neither of these algorithms considered the flow of information in the problem formulation.

Schurgers et al. (2002) considered the flow of information on the management of sensor networks. The authors evaluated the reduction on the latency of getting information from different nodes in the network while conserving energy by turning off the sensors' transmission and reception capabilities, but still being able to monitor a region for events of interest (e.g., forest fire, intrusion detection, etc.). Latency was defined as the time between a node alerting another node about the availability of information, to the time when both nodes had turned on and were ready to transmit/receive the information. Yick et al. (2008) presented a comprehensive review of literature in the area of wireless sensor networking. The main concern in most of the evaluated research was on the topology of the network regarding its application due to limited bandwidth, the flow of information across the network, and then limited lifetime of a sensor.

In this paper, the assignment and scheduling of tasks to users, and the information flow among the users, is formulated as a single mixed-integer nonlinear program (MINLP). The flow of information considered the required information by certain tasks (as input), the information produced by tasks (as output), and precedence relationships between tasks. It can be shown that this optimization problem, as a variant of the vehicle routing problem, is *NP-hard*, requiring heuristic strategies for finding good-quality solutions efficiently as the problem size grows. In addition to linearizing the MINLP, three heuristics were developed, all based on the concept of problem decomposition and the identified subproblems from the MINLP formulation: task assignment, scheduling, and bandwidth allocation (BW) problems. The linearized formulation and the three heuristics were stressed on multiple randomly generated Monte-Carlo scenarios considering different combinations of tasks, users, and artifacts (i.e., pieces of information required to complete a task). ILOG CPLEX (IBM, Armonk, NY) was used as the mechanism for finding the solution to the MILP formulation and each subproblem in the heuristics. Metrics on the solution quality and time to find a solution were analyzed and compared for each approach.

The remainder of this paper is organized as follows. Section 2 derives the MINLP formulation for the information workflow problem. In Section 3, a detailed description of the three heuristic approaches is provided. A computational study and analysis is presented in Section 4, comparing all four approaches (solving the linearized version of the MINLP as well as the three heuristics). Finally, conclusions and future research are discussed in Section 5.

2. Mathematical formulation

The overall problem addressed here is to assign tasks occurring on multiple information workflows to users, and flow information among the users, from tasks that produce information as output to tasks that require the information as input. Each task can only be assigned to a user if the user is able to perform the task (as seen by a nonempty intersection of the set of user roles and the set of allowable roles for the task). For example, the task might be to make a decision regarding mission planning of a unmanned vehicle, which can only be done by a commander or an intelligence analyst, and a particular user is someone who has expertise as a imagery analyst and an intelligence analyst, so there is overlap in the roles and the task can be assigned to this user. One thing to note is that users in this context could be humans or computer software systems. Users have processing times to accomplish tasks. There are precedence relationships between the tasks (e.g., Task k must be completed before Task m can start). In addition, tasks might require certain information as input before they can begin, and tasks might generate information as output when they are completed. If a task needs a certain piece of information as input (e.g., α), there needs to be an assignment of the transference of α from a user assigned a task producing α as output to the user assigned the task requiring α as input. It is assumed that each user has a finite amount of uplink and downlink bandwidth with which to send and receive information. Additionally, if a user is assigned multiple tasks, the user cannot work on those tasks concurrently, nor can the user stop a task before completion. It is important to note that the term “user” is being used in a general sense. A user could represent a human performing a task, as well as an automated system performing a task.

Figures 1 and 2 present an example scenario. In this scenario, there are two workflows, with multiple tasks. The arrows in each workflow denote task precedence. For example, on Workflow

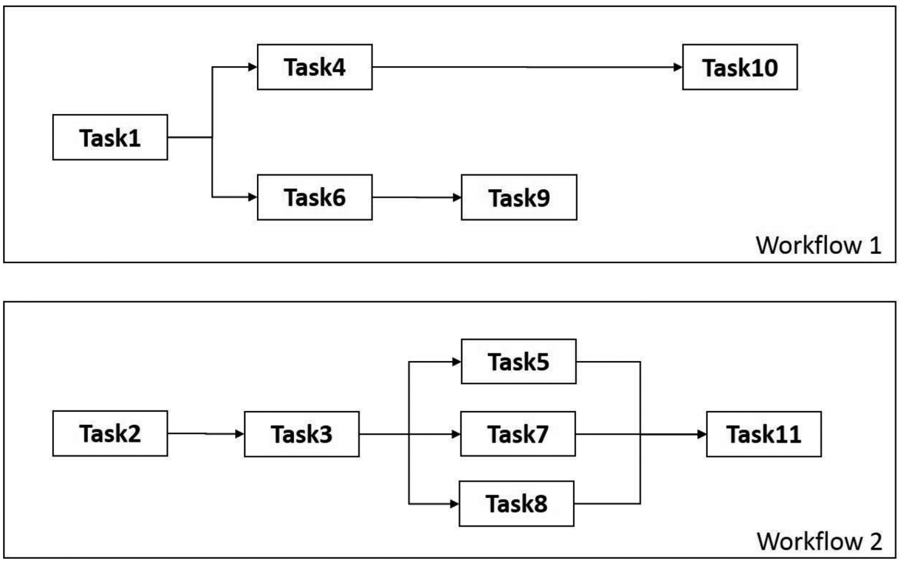

Fig. 1. Initial information workflows.

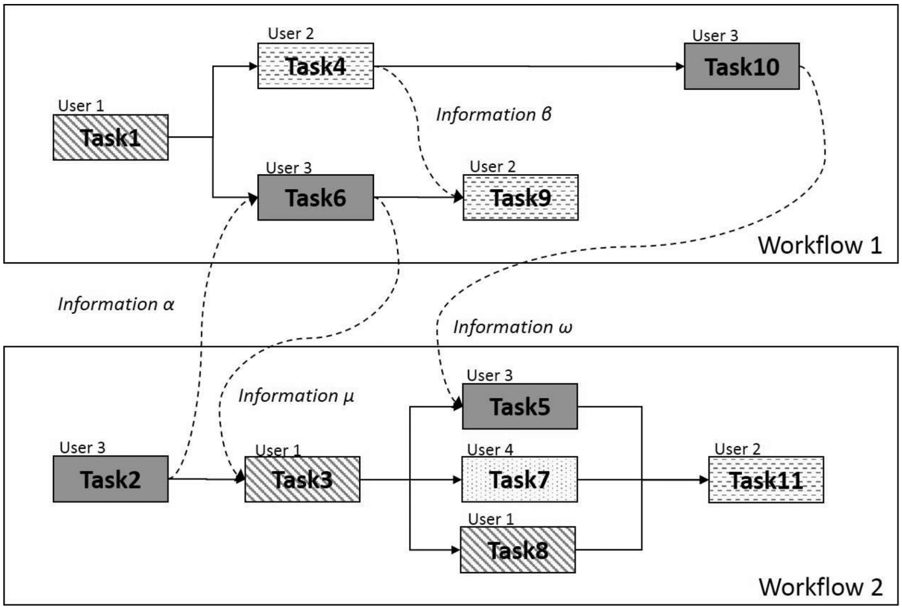

Fig. 2. Assignment of users to workflow tasks, and information flow among users.

1, Task 9 cannot start until Task 6 has completed, while on Workflow 2, Task 11 cannot start until all of Tasks 5, 7, and 8 have completed. The goal is to assign the tasks on the workflows to users, schedule the tasks assigned to each user, and determine the appropriate information transfers that need to take place among the users, and when those information transformations need to be performed. Figure 1 displays the initial workflows, while Fig. 2 shows the resultant assignment. In this figure, the tasks are coded by the user assignments and the information flows were detailed. In this example scenario, *User* 1 is first assigned to perform *Task* 1 (on workflow 1). When *Task* 1 is completed and *User* 1 receives information μ from *User* 3, then *User* 1 can begin its next task, *Task* 3 (on workflow 2). Once complete with *Task* 3, *User* 1 can begin *Task* 8 (on workflow 2). This workflow assignment results in the schedule displayed in Fig. 3. In this schedule, it is clear that there are some idle times for some of the users. This is due to when they are either (a) waiting for another task to be completed by another user (e.g., *User* 2 waiting for *Tasks* 5, 7, and 8 to complete before starting *Task* 11) or (b) waiting on information needed for their next task (e.g., *User* 1 waiting for *Artifact* μ as input before starting *Task* 3). While this is a simplified example, in general it is possible for a task to produce multiple artifacts as output, and a task can require multiple artifacts as input. The mathematical formulation can handle these generalities.

2.1. Parameters

This section discusses all of the parameters needed for the mathematical formulation.

$\mathcal{T}$ defines the number of activities (indexed by i);
$\mathcal{P}$ defines the number of users (indexed by j);

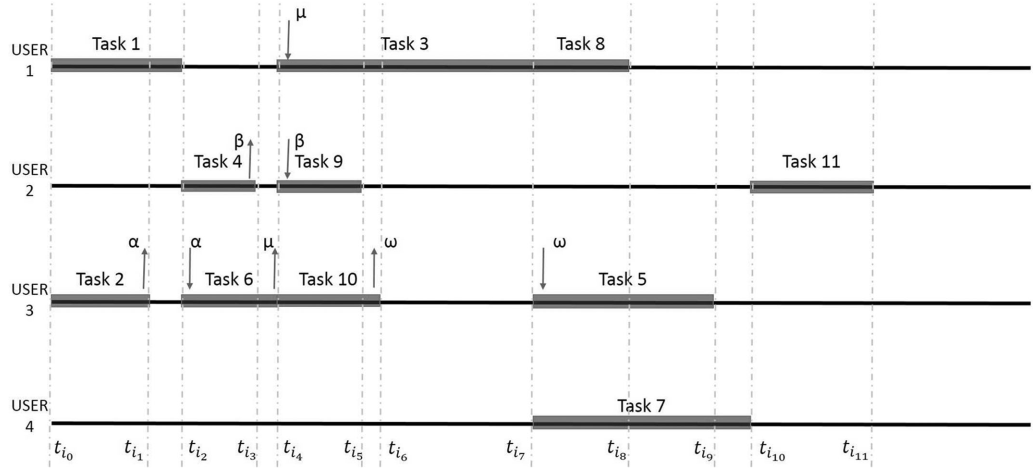

Fig. 3. Resultant user/task schedules for the two information workflows. Captured in this schedule are the sending of information from a user (upward arrow, upon completion of a task) and the receiving of information by a user (downward arrow, at start of a task). Note that there are some idle times for the users, when they are either (a) waiting for another task to be completed by another user (i.e., task precedence constraint) or (b) waiting on information needed for their next task.

ρ_{ij} defines the processing time of activity i by user j;
$\mathcal{Q}$ defines the number of possible roles (indexed by q);
$\mathcal{N}$ defines the number of possible inputs/outputs of all activities on all workflows, called artifacts (indexed by n);
R_j is a binary vector of length $\mathcal{Q}$, where $R_{jq} = \begin{cases} 1 & \text{if user } j \text{ can perform role } q \\ 0 & \text{otherwise} \end{cases}$;

$\bar{R}_i$ is a binary vector of length $\mathcal{Q}$, where $\bar{R}_{iq} = \begin{cases} 1 & \text{if activity } i \text{ can be performed by role } q \\ 0 & \text{otherwise} \end{cases}$;

I_i is a binary vector of length $\mathcal{N}$, where $I_{in} = \begin{cases} 1 & \text{if activity } i \text{ requires artifact } n \\ 0 & \text{otherwise} \end{cases}$;

O_i is a binary vector of length $\mathcal{N}$, where $O_{in} = \begin{cases} 1 & \text{if activity } i \text{ produces artifact } n \\ 0 & \text{otherwise} \end{cases}$;

V_i is a binary vector of length $\mathcal{T}$, where $V_{i\hat{\imath}} = \begin{cases} 1 & \text{if activity } \hat{\imath} \text{ must finish before activity } i \text{ can start} \\ 0 & \text{otherwise} \end{cases}$;

ω_n is the size (e.g., number of bytes) of artifact n;
d_j defines the downlink bandwidth (e.g., bytes per time period) of user j. The downlink bandwidth specifies how much information user j can receive each time period;
u_j defines the uplink bandwidth (e.g., bytes per time period) of user j. The uplink bandwidth specifies how much information user j can send each time period;
Δ is the number of time periods (indexed by k). *Note*: The starting time associated with time period k is denoted t_k. We assume that time is discretized into Δ time periods, of equal length;
$\mathcal{H}$ is a large enough constant number.

2.2. Decision variables

This section discusses all of the decision variables needed for the mathematical formulation.

$y_{ij} = \begin{cases} 1 & \text{if activity } i \text{ is assigned to user } j \\ 0 & \text{otherwise} \end{cases};$

$z_{ij\ell} = \begin{cases} 1 & \text{if } \ell\text{th activity of user } j \text{ is activity } i \\ 0 & \text{otherwise} \end{cases};$

$\hat{z}_{j\ell} = \begin{cases} 1 & \text{if } \exists\, i \text{ s.t. } z_{ij\ell} = 1 \\ 0 & \text{otherwise} \end{cases}.$

Note: This is used to enforce the ℓth activity of user j not being assigned if the $(\ell - 1)$th activity of user j is not assigned;

$\hat{S}_i$ is start time of activity i;

$\hat{E}_i$ is completion time of activity i;

$S_{j\ell}$ is start time of ℓth activity of user j;

$E_{j\ell}$ is completion of ℓth activity of user j;

$\Phi_{\hat{j}jn} = \begin{cases} 1 & \text{if user } \hat{j} \text{ is assigned to send artifact } n \text{ to user } j \\ 0 & \text{otherwise} \end{cases};$

$F_{\hat{j}jnk}$ is the proportion of artifact n user $\hat{j}$ has sent to user j by time period k;

$r_{\hat{j}n}$ is the time at which user $\hat{j}$ ends activity with artifact n as output;

$\bar{r}_{jn}$ is the time at which user j completely receives artifact n as input;

$x_{\hat{j}jnk}$ is the proportion of artifact n user $\hat{j}$ sent to user j during time period k;

$\varphi_{i\hat{j}n} = \begin{cases} 1 & \text{if user } \hat{j} \text{ is assigned an activity } i \text{ that will produce artifact } n \\ 0 & \text{otherwise} \end{cases}$

$g_{\hat{j}n} = \begin{cases} 1 & \text{if none of the tasks assigned to user } \hat{j} \text{ produce artifact } n \text{ as output} \\ 0 & \text{otherwise} \end{cases};$

$b_{\hat{j}nk} = \begin{cases} 1 & \text{if } t_k \geq r_{\hat{j}n}, \text{ that is, if time-step } t_k \text{ is after user } \hat{j} \text{ produces artifact } n \\ 0 & \text{otherwise} \end{cases};$

$c_{\hat{j}jnk} = \begin{cases} 1 & \text{if } F_{\hat{j}jnk} = 1, \text{ that is, if user } \hat{j} \text{ has completed the transfer of artifact } n \text{ to user } \\ & j \text{ by time-step } t_k \\ 0 & \text{otherwise} \end{cases}.$

2.3. Nonlinear formulation

The resultant MINLP formulation is given as follows (where, in all constraints to follow, $i, \hat{\imath} \in \{1, \ldots, \mathcal{T}\}$, $i \neq \hat{\imath}$, $j, \hat{j} \in \{1, \ldots, \mathcal{P}\}$, and $\ell, \hat{\ell} \in \{1, \ldots, \mathcal{T}\}$):

$$F = \min[\max_i \{\hat{E}_i\}] \tag{1}$$

s.t.

$$y_{ij} \leq \sum_{r=1}^{\mathcal{R}} R_{jr}\bar{R}_{ir} \quad \forall i, j \tag{2}$$

$$\sum_{j=1}^{\mathcal{P}} y_{ij} = 1 \quad \forall i \tag{3}$$

$$\sum_{\ell=1}^{\mathcal{T}} z_{ij\ell} = y_{ij} \quad \forall i, j \tag{4}$$

$$\hat{S}_i = \sum_{j=1}^{\mathcal{P}} \sum_{\ell=1}^{\mathcal{T}} z_{ij\ell} S_{j\ell} \quad \forall i \tag{5}$$

$$\hat{E}_i = \sum_{j=1}^{\mathcal{P}} \sum_{\ell=1}^{\mathcal{T}} z_{ij\ell} E_{j\ell} \quad \forall i \tag{6}$$

$$E_{j\ell} = S_{j\ell} + \sum_{i=1}^{\mathcal{T}} \rho_{ij} z_{ij\ell} \quad \forall \ell, j \tag{7}$$

$$S_{j0} = 0,\ E_{j0} = 0 \quad \forall j \tag{8}$$

$$S_{j\ell} \geq E_{j,\ell-1} \quad \forall \ell, j \tag{9}$$

$$\hat{S}_i \geq V_{i\hat{\imath}} \hat{E}_{\hat{\imath}} \quad \forall i, \hat{\imath} \tag{10}$$

$$S_{j\ell} \geq I_{in} z_{ij\ell} \Phi_{\hat{\jmath}jn} z_{\hat{\imath}\hat{\jmath}\hat{\ell}} O_{\hat{\imath}n} E_{\hat{\jmath}\hat{\ell}} \quad \forall n, \ell, \hat{\ell}, j, \hat{\jmath}, i, \hat{\imath} \tag{11}$$

$$\sum_{\hat{\jmath}=1}^{\mathcal{P}} \Phi_{\hat{\jmath}jn} \geq I_{in} y_{ij} \quad \forall n, i, j \tag{12}$$

$$\Phi_{\hat{\jmath}jn} \leq \sum_{\hat{\imath}=1}^{\mathcal{T}} O_{\hat{\imath}n} y_{\hat{\imath}\hat{\jmath}} \quad \forall n, j, \hat{\jmath} \tag{13}$$

$$\Phi_{\hat{j}jn} \leq \sum_{i=1}^{\mathcal{T}} I_{in} y_{ij} \quad \forall n, j, \hat{j} \tag{14}$$

$$\sum_{\hat{j}=1}^{\mathcal{P}} \Phi_{\hat{j}jn} \leq 1 \quad \forall n, j \tag{15}$$

$$\hat{z}_{j\ell} = \sum_{i=1}^{\mathcal{T}} z_{ij\ell} \quad \forall j, \ell \tag{16}$$

$$\hat{z}_{j\ell} \geq \hat{z}_{j,\ell+1} \quad \forall \ell \in \{1, \ldots, \mathcal{T}-1\}, j \tag{17}$$

$$\sum_{i=1}^{T} z_{ij\ell} \leq 1 \quad \forall j, \ell \tag{18}$$

$$F_{\hat{j}jnk} \leq \Phi_{\hat{j}jn} \quad \forall n, k, \hat{j}, j \tag{19}$$

$$r_{\hat{j}n} \geq \sum_{i=1}^{\mathcal{T}} \hat{E}_i \varphi_{i\hat{j}n} \quad \forall n, \hat{j} \tag{20}$$

$$g_{\hat{j}n} + \sum_{i=1}^{\mathcal{T}} \varphi_{i\hat{j}n} = 1 \quad \forall n, \hat{j} \tag{21}$$

$$g_{\hat{j}n} \leq 1 - O_{in} y_{i\hat{j}} \quad \forall n, \hat{j}, i \tag{22}$$

$$g_{\hat{j}n} \geq 1 - \sum_{i=1}^{\mathcal{T}} O_{in} y_{i\hat{j}} \quad \forall n, \hat{j} \tag{23}$$

$$F_{\hat{j}jnk} \leq b_{\hat{j}nk} \quad \forall n, k, \hat{j}, j \tag{24}$$

$$b_{\hat{j}nk}\mathcal{H} < t_{k-1} - r_{\hat{j}n} + \mathcal{H} \quad \forall n, k, \hat{j} \tag{25}$$

$$b_{\hat{j}nk}\mathcal{H} \geq t_{k-1} - r_{\hat{j}n} \quad \forall n, k, \hat{j} \tag{26}$$

$$\hat{S}_i \geq I_{in} y_{ij} \bar{r}_{jn} \quad \forall n, j, i \tag{27}$$

$$c_{\hat{j}jnk} \leq F_{\hat{j}jnk} \quad \forall n, k, \hat{j}, j \tag{28}$$

$$c_{\hat{j}jnk} \geq F_{\hat{j}jnk} - 0.99999 \quad \forall n, k, \hat{j}, j \tag{29}$$

$$\bar{r}_{jn} \geq \left(1 - c_{\hat{j}jnk}\right) \Phi_{\hat{j}jn} t_{k+1} \quad \forall n, k, \hat{j}, j \tag{30}$$

$$x_{\hat{j}jnk} = F_{\hat{j}jnk} - F_{\hat{j}jn,k-1} \quad \forall n, k, \hat{j}, j \tag{31}$$

$$\sum_{\substack{j=1 \\ j \neq \hat{j}}}^{\mathcal{P}} \sum_{n=1}^{\mathcal{N}} x_{\hat{j}jnk} \omega_n \leq u_{\hat{j}} \quad \forall k, \hat{j} \tag{32}$$

$$\sum_{\hat{j}=1_{\hat{j} \neq j}}^{\mathcal{P}} \sum_{n=1}^{\mathcal{N}} x_{\hat{j}jnk} \omega_n \leq d_j \quad \forall k, j \tag{33}$$

$$F_{\hat{j}jn0} = 0 \quad \forall n, \hat{j}, j \tag{34}$$

$$\hat{S}_i, \hat{E}_i, S_{j\ell}, E_{j\ell} \in [0, \mathcal{H}] \quad \forall \ell, j, i \tag{35}$$

$$x_{\hat{j}jnk}, r_{\hat{j}n}, \bar{r}_{jn} \in [0, \mathcal{H}] \quad \forall n, k, \ell, \hat{j}, j, i \tag{36}$$

$$\hat{z}_{j\ell}, y_{ij}, z_{ij\ell}, \Phi_{\hat{j}jn} \in \{0, 1\} \quad \forall n, \ell, \hat{j}, j, i \tag{37}$$

$$b_{\hat{j}nk}, c_{\hat{j}jnk}, \varphi_{i\hat{j}n}, g_{\hat{j}n} \in \{0, 1\} \quad \forall n, k, \ell, \hat{j}, j, i \tag{38}$$

2.4. Interpretation of formulation

The objective function (1) minimizes the maximum time at which an activity is completed.
Constraints (2) only permit user j to perform activity i if there exists overlap between the roles user j can assume (R_{jr}) and the roles necessary to fill activity i ($\bar{R}_{ir}$).
Constraints (3) enforce that each activity must be assigned exactly one user.

Constraints (4) specify that one of the ℓ activities of user j must be activity i, if and only if activity i is assigned to user j.

Constraints (5) relate the start time of activity i ($\hat{S}_i$) to the start time of the ℓth activity of user j ($S_{j\ell}$).

Constraints (6) relate the end time of activity i ($\hat{E}_i$) to the end time of the ℓth activity of user j ($E_{j\ell}$).

Constraints (7), for user j, relate the end time of the ℓth activity ($E_{j\ell}$) to the start time of the ℓth activity ($S_{j\ell}$) and the processing time of the activity (ρ_{ij}).

Constraints (8) provide initial values for variables needed to make Constraints (9) consistent.

Constraints (9) relate, for user j, the starting time of the ℓth ($S_{j\ell}$) to the ending time of the $(\ell-1)$th activity ($E_{j,\ell-1}$).

Constraints (10) enforce that the starting time of activity i ($\hat{S}_i$) must occur after activity $\hat{\imath}$ is completed ($\hat{E}_{\hat{\imath}}$), if there exists a precedence relationship between activities i and $\hat{\imath}$ (i.e., if $V_{i\hat{\imath}} = 1$).

Constraints (11) enforce that the starting time of the ℓth activity of user j ($S_{j\ell}$) must occur after the $\hat{\ell}$th activity of user $\hat{\jmath}$ ($E_{\hat{\jmath}\hat{\ell}}$) is completed, if activity i is the ℓth activity of user j (i.e., $z_{ij\ell} = 1$), activity $\hat{\imath}$ is the $\hat{\ell}$th activity of user $\hat{\jmath}$ (i.e., $z_{\hat{\imath}\hat{\jmath}\hat{\ell}} = 1$), activity i needs as input artifact n (i.e., $I_{in} = 1$), activity $\hat{\imath}$ produces artifact n as output (i.e., $O_{\hat{\imath}n} = 1$), and user $\hat{\jmath}$ is assigned to send artifact n to user j (i.e., $\Phi_{\hat{\jmath}jn} = 1$);

Constraints (12) will force at least one user to send artifact n to user j if user j needs it as input for an assigned task. *Note*: This constraint, by itself, does not verify that the user assigned to send artifact n to user j does in fact have a task that produces artifact n as output.

Constraints (13) will force $\Phi_{\hat{\jmath}jn}$ to be 0 if user $\hat{\jmath}$ is not assigned a task that produces artifact n as output.

Constraints (14) will force $\Phi_{\hat{\jmath}jn}$ to be 0 if user j is not assigned a task that needs artifact n as input.

Constraints (15) allow at most one user to send artifact n to user j.

Constraints (16) and (17) allow the $(\ell+1)$th activity of user j ($\hat{z}_{j,\ell+1}$) to be assigned an activity only if the ℓth activity of user j is also assigned.

Constraints (18) permit at most one activity to be assigned as the ℓth activity of user j.

Constraints (19) force $F_{\hat{\jmath}jnk}$ to be 0 when user $\hat{\jmath}$ is not assigned to send artifact n to user j.

Constraints (20)–(23) determine the time at which user $\hat{\jmath}$ ends the activity with artifact n produced as output ($r_{\hat{\jmath}n}$).

Constraints (24) force $F_{\hat{\jmath}jnk}$ to be 0 if by time period k, user $\hat{\jmath}$ has not completed activity with artifact n as output.

Constraints (25) force $b_{\hat{\jmath}nk}$ to be 0 when $t_k < r_{\hat{\jmath}n}$, and allows $b_{\hat{\jmath}nk} \in \{0, 1\}$ otherwise.

Constraints (26) force $b_{\hat{\jmath}nk}$ to be 1 when $t_k \geq r_{\hat{\jmath}n}$ and allows $b_{\hat{\jmath}nk} \in \{0, 1\}$ otherwise.

Constraints (27) determine the minimum time at which activity i can start.

Constraints (28) force $c_{\hat{\jmath}jnk}$ to be 0 if user $\hat{\jmath}$ has not finished sending artifact n to user j, and allows $c_{\hat{\jmath}jnk} \in \{0, 1\}$ otherwise.

Constraints (29) force $c_{\hat{\jmath}jnk}$ to be 1 if user $\hat{\jmath}$ has completely sent artifact n to user j, and allows $c_{\hat{\jmath}jnk} \in \{0, 1\}$ otherwise.

Constraints (30) force $\bar{r}_{jn}$ to be no smaller than t_{k+1} when user $\hat{\jmath}$ has not completely sent artifact n to user j by time t_k (*Note*: If $\Phi_{\hat{\jmath}jn} = 0$, then $\bar{r}_{jn} \geq 0$).

Constraints (31) determine the proportion of artifact n sent by user $\hat{\jmath}$ to user j during time period k, $x_{\hat{\jmath}jnk}$.

Constraints (32) enforce that user $\hat{j}$, during time period k, does not utilize more uplink bandwidth than allocated ($u_{\hat{j}}$).
Constraints (33) enforce that user j, during time period k, does not utilize more downlink bandwidth that allocated (d_j).
Constraints (34) define initial time-step values on the $F_{\hat{j}jnk}$ variables.
Constraints (35)–(38) prescribe domain restrictions on all of the decision variables.

Note: From the parameter section, Section 2.1, $\mathcal{H}$ is defined to be a "large-enough" constant. Determining large enough is in most cases more of an art than a science. However, there are some simple methods to determine analytically appropriate values for these types of constants. Considering first just the processing time, then the worst-case scenario would be to have the activities done serially, one after the other, with the user assigned to each activity being the user that would take the most amount of time to perform that activity. Hence, an appropriate value for $\mathcal{H}$ would be

$$\mathcal{H} = \sum_{i=1}^{\mathcal{T}} \max_{j=1,\ldots,\mathcal{P}} \left[\rho_{ij}\right]. \tag{39}$$

Now considering bandwidth limitations into our derivation for $\mathcal{H}$, let $\tilde{f}$ be the smallest number of bytes that can be sent in one time period and let $\tilde{\omega}$ be the size of the largest artifact (in bytes), that is,

$$\tilde{f} = \min[\min_j\{d_j\}, \min_j\{u_j\}] \tag{40}$$

$$\tilde{\omega} = \max_n\{\omega_n\}. \tag{41}$$

$\mathcal{H}$ can therefore be defined as in Equation (42), where the first term is just concerned with the worst-case task processing and the second term captures the number of time periods necessary to send the largest artifact, using the smallest bandwidth, multiplied by the number of artifacts, and then multiplied by one less than the number of users, as the worst case would be if one user needed to send all of the artifacts to all of the other users:

$$\mathcal{H} = \sum_{i=1}^{\mathcal{T}} \max_{j=1,\ldots,\mathcal{P}} [\rho_{ij}] + (\tilde{\omega}/\tilde{f})n(\mathcal{P} - 1). \tag{42}$$

3. Solution methodologies

The formulation derived in Section 2.3 is a nonlinear mixed-integer programming problem. However, it can be linearized, and the resulting MILP is presented in the Appendix (Section A.1). As such, the MILP formulation can be solved using a number of commercial software packages (e.g., ILOG CPLEX, LINDO [Lindo Systems Inc., Chicago, IL], etc.). It can be shown that this optimization problem is *NP-hard*, as a variant of the vehicle routing problem, which necessitate heuristic strategies for finding good-quality solutions efficiently (Garey and Johnson, 1979; Ausiello et al., 1999) when the problem sizes are of sufficient size.

At its most basic, the MILP formulation consists of three coupled problems: an assignment problem, a scheduling problem, and a BW problem. Based on the concept of problem decomposition, we have developed three innovative heuristics. Each of the three heuristics approximates the original problem by decoupling at least one of the coupled problems.

3.1. Decomposition heuristic A

- Solve the assignment problem.
  - Minimize (43), subject to the constraints (2) and (3). This has the effect of load balancing the tasks across the users:

$$F_1 = \min\left[\max_j\left(\sum_{i=1}^{\mathcal{T}} \rho_{ij} y_{ij}\right)\right]. \tag{43}$$

- Solve the coupled scheduling/BW problem, with the assignment solution as input.
  - Using the assignment variables y_{ij} as input parameters, minimize (1), subject to constraints (4)–(38).

3.2. Decomposition heuristic B

- Solve the coupled assignment/scheduling problem;
  - Minimize (1), subject to (2)–(18), (35), (37). This solves the problem ignoring bandwidth limitations, effectively assuming unlimited and instantaneous bandwidth. Thus, assignment of artifact transference between users is determined here; but the time at which this transference occurs is not computed here.
- Solve the BW problem, with assignment and scheduling solution as input.
  - Using the assignment variables y_{ij}, $\hat{z}_{j\ell}$, $z_{ij\ell}$, and $\Phi_{\hat{j}jn}$ as input parameters, minimize (1) subject to (5)–(11), (19)–(36), (38).

3.3. Decomposition heuristic C

- Solve the assignment problem;
  - Minimize (43), subject to the constraints (2) and (3).
- Solve the scheduling problem, with assignment solution as input;

Table 1
Values used for the number of tasks, users, and artifacts over the numerical experiments

Parameter	Values
$\mathcal{T}$	5, 10
$\mathcal{P}$	2, 3, 5, 10
$\mathcal{N}$	3, 5, 10

  - Using the assignment variables y_{ij} as input parameters, minimize (1), subject to constraints (4)–(18), (35), (37).

- Solve the BW problem, with assignment and scheduling solution as input.
  - Using the assignment variables y_{ij}, $\hat{z}_{j\ell}$, $z_{ij\ell}$, and $\Phi_{\hat{j}jn}$ as input parameters, minimize (1) subject to (5)–(11), (19)–(36), (38).

4. Computational experiments

In this section, we compare the solution quality and efficiency of finding solutions to the full formulation, as well as the three decomposition heuristics, for a variety of test scenarios.

4.1. Test environment

The full formulation and the decomposition heuristic were solved using version 12.2.0.0 of `CPLEX` ILOG CPLEX (IBM, Armonk, NY). `Matlab R2011b (7.13.0.564)` Matlab was used to generate all of the Monte-Carlo scenarios, create the mixed-integer linear programming formulation file, call `CPLEX`, and analyze the solution results from `CPLEX` (MathWorks, Natick, MA). All experiments were conducted on a Dell Latitude E5420 with an Intel(R) Core(TM) i5-2540M CPU with 2.60 GHz and 8.00 GB RAM.

4.2. Experimental results

To test the four different approaches (solving the linearized version of the MINLP with CPLEX, as well as the three decomposition heuristics), we simulated multiple workflow processes, varying the number of users, $\mathcal{P}$, the number of tasks, $\mathcal{T}$, and the number of artifacts, $\mathcal{N}$, as displayed in Table 1.

For each combination of number of tasks, users, and artifacts, 10 Monte-Carlo scenarios were created randomly. For each task, the processing time was determined through a random draw from a uniform distribution, $U(1, 30)$, of time units. For each user, the uplink and downlink bandwidths were determined through a random draw from a uniform distribution, $U(1, 3)$, of size units per time unit. For each artifact, the size was determined through a random draw from a uniform distribution, $U(5, 60)$, of size units. Each of the four approaches was given five minutes to solve each scenario.

We note that for the three decomposition heuristics, we did not allocate a certain portion of the five minutes to each of the decomposition steps, rather five minutes was given to the decomposition as a whole. This means that the first subproblem was allowed up to the full five minutes to find the best solution possible (ideally optimal within a tolerance of ϵ). The next subproblem was given any remaining time to find the best solution possible (ideally optimal within a tolerance of ϵ), and so on until the last subproblem was given any remaining time to find the best solution possible (again, ideally optimal within the tolerance).

Metrics to analyze the four approaches include solution quality and time to find a solution. In addition, we also compute the number of constraints and decision variables for each of the four approaches. From Hirsch et al. (2010), the optimality gap can be defined as $GAP = |f(x_\mu) - f(x^*)|$, where x_μ is the best solution found by approach μ, x^* is the optimal solution, and f is the objective function defined in Equation (1). It can then be stated that the solution found is sufficiently close to the optimal solution if

$$GAP \leq \begin{cases} \epsilon & \text{if } f(x^*) = 0 \\ \epsilon \cdot |f(x^*)| & \text{if } f(x^*) \neq 0, \end{cases} \tag{44}$$

where $\epsilon = 0.001$.

From the experiments, it was seen that for small problem sizes CPLEX is able to solve the full formulation to optimality. Also, Decomposition B appears to be doing better than Decompositions A and C on these small-sized problems. As the problem size increased, CPLEX was no longer able to solve the full formulation to optimality, and the quality of Decomposition B solution decreased with respect to that of Decompositions A and C. Eventually, the full formulation and Decomposition B were not able to find solutions in CPLEX, while Decompositions A and C were still able to find solutions, comparable to each other. These detailed results for each individual Monte-Carlo scenario are presented in Tables A1–A6 (Section A.2 in the Appendix).

As adapted from Hirsch et al. (2014), Table 2 summarizes the results over all the Monte-Carlo scenarios. For each of the four approaches, OSF% (where OSF is optimal solution found) is defined as the percentage of Monte-Carlo scenarios for which the approach found a solution satisfying Equation (44) (based on the number of times the optimal solution is known from the full formulation approach), and BSF% (where BSF is best solution found) is defined as the percentage of Monte-Carlo scenarios for which the approach found a solution satisfying Equation (44) with respect to the best solution found over all four approaches. As is made clear in this table, for smaller sized problems, Decomposition B performs much better than do Decompositions A and C. However, for larger sized problems, solutions are not able to be found for Decomposition B, while Decompositions A and C do find feasible solutions within the given time bounds. This is due to how the full formulation is decoupled. Both Decompositions A and C first solve an assignment problem, and then use the results of that assignment problem to solve the remaining problems. This appears to reduce the complexity of the formulation significantly, as is evidenced by the results in the table.

Table 3 presents results concerned with CPLEX solving the full formulation approach (i.e., solving the complete MILP). For each combination of number of tasks, artifacts, and users, this table displays the percentage of Monte-Carlo scenarios where CPLEX found the optimal solution within the time bound (column 4), the percentage of time that CPLEX was not able to find even a feasible solution within the time bound (column 6), and for those scenarios where a feasible, but not optimal solution was found, the average solution GAP (gap between the best solution found

Table 2
Comparison of optimal solution found (OSF) and best solution found (BSF) metrics over the different approaches

No. of tasks	No. of artifacts	No. of users	Metric	FF	DA	DB	DC
5	3	2	OSF	100	50	100	50
			BSF	100	50	100	50
5	3	3	OSF	100	10	40	10
			BSF	100	10	40	10
5	3	5	OSF	70	10	40	10
			BSF	80	10	60	10
5	3	10	OSF	0	0	0	0
			BSF	0	40	80	40
5	5	2	OSF	100	90	90	90
			BSF	100	90	90	90
5	5	3	OSF	80	0	40	0
			BSF	100	0	60	0
5	5	5	OSF	10	0	10	0
			BSF	30	20	90	20
5	5	10	OSF	0	0	0	0
			BSF	0	50	60	50
5	10	2	OSF	80	50	50	50
			BSF	100	50	60	50
5	10	3	OSF	40	20	40	20
			BSF	60	40	80	30
5	10	5	OSF	0	0	0	0
			BSF	0	20	90	20
5	10	10	OSF	0	0	0	0
			BSF	0	0	0	0
10	3	2	OSF	0	0	0	0
			BSF	0	90	0	80
10	3	3	OSF	0	0	0	0
			BSF	0	100	0	100
10	3	5	OSF	0	0	0	0
			BSF	0	100	0	100
10	3	10	OSF	0	0	0	0
			BSF	0	100	0	100
10	5	2	OSF	10	10	10	10
			BSF	10	100	10	70
10	5	3	OSF	0	0	0	0
			BSF	0	100	0	90
10	5	5	OSF	0	0	0	0
			BSF	0	100	0	90
10	5	10	OSF	0	0	0	0
			BSF	0	90	0	90
10	10	2	OSF	0	0	0	0
			BSF	0	80	10	100
10	10	3	OSF	0	0	0	0
			BSF	0	70	0	80
10	10	5	OSF	0	0	0	0
			BSF	0	80	0	80
10	10	10	OSF	0	0	0	0
			BSF	0	0	0	0

FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

Table 3
Solution quality analysis for the full formulation using CPLEX

No. of tasks	No. of artifacts	No. of users	% Time optimal solution found	Mean nonoptimal solution GAP (%)	% Time feasible solution not found
5	3	2	100		0
5	3	3	100		0
5	3	5	70	95	0
5	3	10	0		100
5	5	2	100		0
5	5	3	80	67	0
5	5	5	10	83	20
5	5	10	0		100
5	10	2	80	82	0
5	10	3	40	95	40
5	10	5	0		100
5	10	10	0		100
10	3	2	0	98	70
10	3	3	0		100
10	3	5	0		100
10	3	10	0		100
10	5	2	10	98	80
10	5	3	0		100
10	5	5	0		100
10	5	10	0		100
10	10	2	0	100	80
10	10	3	0		100
10	10	5	0		100
10	10	10	0		100

Column 4 displays the percentage of Monte-Carlo scenarios which found the optimal solution, column 5 displays the mean GAP for feasible, but not optimal solutions, and column 6 displays the percentage of Monte-Carlo scenarios for which CPLEX was not able to find a feasible solution within the time bounds.

and the optimal solution) is computed (column 5). As can be seen in this table, for smaller sized problems, CPLEX is able to solve the problem to optimality. However, as the problems grow in size, even when CPLEX is able to find a feasible solution within the time bounds, the GAP is quite large, meaning the feasible solution found is far from optimal. And for many of the scenarios, CPLEX was unable to even find a feasible solution within the time bounds.

For a given $\{\mathcal{P}, \mathcal{T}, \mathcal{N}\}$ combination, Table 4 presents the average time (over the 10 Monte-Carlo scenarios) each approach took to run. The times are given in seconds. It is easy to see that Decompositions A and C take significantly less time than does Decomposition B and the full formulation. This is again due to how the problem was decomposed.

Lastly, Tables 5–7 present the number of constraints and decision variables for each of the four approaches. For the three decomposition approaches, the table presents the total number of constraints and decision variables, as well as the number for each component. It is clear to see, by looking at Tables 4–7, that by first considering just the assignment problem, as is done by decomposition approaches A and C, while the overall problem formulation is not considerably different from the full formulation, CPLEX is able to find solutions much more quickly.

Table 4
Average time (in seconds) to find solution

No. of tasks	No. of users	No. of artifacts	FF	DA	DB	DC
5	2	3	7.558	0.871	3.582	0.436
5	2	5	6.936	1.643	2.078	1.033
5	2	10	103.740	10.042	11.106	6.962
5	3	3	67.298	1.538	10.172	1.286
5	3	5	122.240	4.253	9.989	3.530
5	3	10	227.490	18.077	37.465	27.050
5	5	3	162.320	2.441	26.476	3.051
5	5	5	277.310	8.834	43.899	16.519
5	5	10	*T*	69.104	213.760	152.920
5	10	3	*T*	18.673	179.420	39.031
5	10	5	*T*	72.329	284.590	142.100
5	10	10	*T*	*T*	*T*	*T*
10	2	3	300.090	163.920	220.170	106.220
10	2	5	286.110	169.050	197.760	124.620
10	2	10	300.440	229.280	230.650	221.330
10	3	3	*T*	80.840	*T*	77.456
10	3	5	*T*	120.280	*T*	98.879
10	3	10	*T*	254.150	*T*	128.800
10	5	3	*T*	60.596	*T*	59.691
10	5	5	*T*	103.460	*T*	69.934
10	5	10	*T*	227.590	*T*	172.520
10	10	3	*T*	89.147	*T*	78.652
10	10	5	*T*	206.170	*T*	277.460
10	10	10	*T*	*T*	*T*	*T*

A value of '*T*' is listed if none of the 10 Monte-Carlo scenarios resulted in a solution being found by the approach within the time bounds of five minutes. FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

5. Concluding remarks

This research has been focused on workflow optimization, incorporating bandwidth limitations on the system. A rigorous mathematical model for the problem has been derived, which resulted in an MINLP. A linearized version of the formulation is presented in Section 2. Since the resultant optimization problem is *NP-hard*, three heuristics have been developed based on the concept of problem decomposition. The resulting MILP formulation and three heuristics were stressed on a number of scenarios, using CPLEX as the mechanism for finding a solution. Each approach was given a total of five minutes to find a solution. For very small problems, the full formulation can be solved to optimality within the time limit, and hence outperforms the heuristics. One of the decomposition heuristics (Decomposition B) performs close to the full formulation on the small-sized problems. However, for moderately sized problems, the full formulation cannot be solved to optimality (there exist numerous cases where even a feasible solution cannot be found within the time limit). For these cases, Decomposition A seems to perform slightly better than Decomposition C, and both outperform Decomposition B. In addition, both take significantly less time than does Decomposition B. However, as can be seen in the computational results, none of the heuristics are

Table 5
Number of constraints broken out by each component of each approach

No. of tasks	No. of artifacts	No. of users	FF	DA	DB	DC
5	3	2		21 : 49,690	47,171 : 48,942	21 : 47,156 : 48,942
			49,705	49,711	96,113	96,119
5	3	3		26 : 113,022	106,379 : 110,757	26 : 106,359 : 110,757
			113,042	113,048	217,136	217,142
5	3	5		36 : 324,346	298,331 : 314,511	36 : 298,301 : 314,511
			324,376	324,382	612,842	612,848
5	3	10		61 : 1,417,906	1,228,171 : 1,341,686	61 : 1,228,116 : 1,341,686
			1,417,961	1,417,967	2,569,857	2,569,863
5	5	2		21 : 71,386	67,611 : 70,198	21 : 67,596 : 70,198
			71,401	71,407	137,809	137,815
5	5	3		26 : 163,236	152,801 : 159,549	26 : 152,781 : 159,549
			163,256	163,262	312,350	312,356
5	5	5		36 : 472,036	429,741 : 455,791	36 : 429,711 : 455,791
			472,066	472,072	885,532	885,538
5	5	10		61 : 2,092,786	1,778,691 : 1,966,046	61 : 1,778,636 : 1,966,046
			2,092,841	2,092,847	3,744,737	3,744,743
5	10	2		21 : 125,626	118,711 : 123,338	21 : 118,696 : 123,338
			125,641	125,647	242,049	242,055
5	10	3		26 : 288,771	268,856 : 281,529	26 : 268,836 : 281,529
			288,791	288,797	550,385	550,391
5	10	5		36 : 841,261	758,266 : 808,991	36 : 758,236 : 808,991
			841,291	841,297	1,567,257	1,567,263
5	10	10		61 : 3,779,986	3,154,991 : 3,526,946	61 : 3,154,936 : 3,526,946
			3,780,041	3,780,047	6,681,937	6,681,943
10	3	2		41 : 839,515	832,361 : 836,747	41 : 832,331 : 836,747
			839,545	839,556	1,669,108	1,669,119
10	3	3		51 : 1,894,342	1,873,994 : 1,885,672	51 : 1,873,954 : 1,885,672
			1,894,382	1,894,393	3,759,666	3,759,677
10	3	5		71 : 5,302,156	5,217,296 : 5,263,646	71 : 5,217,236 : 5,263,646
			5,302,216	5,302,227	10,480,942	10,480,953
10	3	10		121 : 21,654,691	21,009,761 : 21,352,371	121 : 21,009,651 : 21,352,371
			21,654,801	21,654,812	42,362,132	42,362,143
10	5	2		41 : 1,205,031	1,194,021 : 1,200,603	41 : 1,193,991 : 1,200,603
			1,205,061	1,205,072	2,394,624	2,394,635
10	5	3		51 : 2,722,036	2,689,496 : 2,707,864	51 : 2,689,456 : 2,707,864
			2,722,076	2,722,087	5,397,360	5,397,371
10	5	5		71 : 7,631,646	7,492,506 : 7,567,926	71 : 7,492,446 : 7,567,926
			7,631,706	7,631,717	15,060,432	15,060,443
10	5	10		121 : 31,280,671	30,210,381 : 30,777,731	121 : 30,210,271 : 30,777,731
			31,280,781	31,280,792	60,988,112	60,988,123
10	10	2		41 : 2,118,821	2,098,171 : 2,110,243	41 : 2,098,141 : 2,110,243
			2,118,851	2,118,862	4,208,414	4,208,425
10	10	3		51 : 4,791,271	4,728,251 : 4,763,344	51 : 4,728,211 : 4,763,344
			4,791,311	4,791,322	9,491,595	9,491,606
10	10	5		71 : 13,455,371	13,180,531 : 13,328,626	71 : 13,180,471 : 13,328,626

Continued

Table 5
Continued

No. of tasks	No. of artifacts	No. of users	FF	DA	DB	DC
			13,455,431	13,455,442	26,509,157	26,509,168
10	10	10		121 : 55,345,621	53,211,931 : 54,341,131	121 : 53,211,821 : 54,341,131
			55,345,731	55,345,742	107,553,062	107,553,073

For each combination of tasks, artifacts, and users, the first row represents the number of constraints for each component of the decomposition approaches, separated by a ":", and the second row represents the sum over all components. FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

Table 6
Number of decision variables broken out by each component of each approach

No. of tasks	No. of artifacts	No. of users	FF	DA	DB	DC
5	3	2	12,345	1 : 12,345	8,133 : 12,345	1 : 8,133 : 12,345
			6,282	10 : 6,272	2,082 : 6,200	10 : 2,072 : 6,200
5	3	3	28,832	1 : 28,832	18,194 : 28,832	1 : 18,194 : 28,832
			13,497	15 : 13,482	4,632 : 13,365	15 : 4,617 : 13,365
5	3	5	86,496	1 : 86,496	50,316 : 86,496	1 : 50,316 : 86,496
			38,100	25 : 38,075	12,750 : 37,850	25 : 12,725 : 37,850
5	3	10	412,481	1 : 412,481	200,621 : 412,481	1 : 200,621 : 412,481
			177,850	50 : 177,800	50,650 : 177,200	50 : 50,600 : 177,200
5	5	2	19,753	1 : 19,753	12,133 : 19,753	1 : 12,133 : 19,753
			9,690	10 : 9,680	2,090 : 9,600	10 : 2,080 : 9,600
5	5	3	47,624	1 : 47,624	27,194 : 47,624	1 : 27,194 : 47,624
			21,675	15 : 21,660	4,650 : 21,525	15 : 4,635 : 21,525
5	5	5	150,616	1 : 150,616	75,316 : 150,616	1 : 75,316 : 150,616
			65,550	25 : 65,525	12,800 : 65,250	25 : 12,775 : 65,250
5	5	10	788,721	1 : 788,721	300,621 : 788,721	1 : 300,621 : 788,721
			343,850	50 : 343,800	50,850 : 343,000	50 : 50,800 : 343,000
5	10	2	40,373	1 : 40,373	22,133 : 40,373	1 : 22,133 : 40,373
			20,310	10 : 20,300	2,110 : 20,200	10 : 2,100 : 20,200
5	10	3	104,054	1 : 104,054	49,694 : 104,054	1 : 49,694 : 104,054
			49,995	15 : 49,980	4,695 : 49,800	15 : 4,680 : 49,800
5	10	5	363,416	1 : 363,416	137,816 : 363,416	1 : 137,816 : 363,416
			170,925	25 : 170,900	12,925 : 170,500	25 : 12,900 : 170,500
5	10	10	2,201,821	1 : 2,201,821	550,621 : 2,201,821	1 : 550,621 : 2,201,821
			1,042,350	50 : 1,042,300	51,350 : 1,041,000	50 : 51,300 : 1,041,000

For the number of tasks equal to 5, and each combination of artifacts and users, the first row represents the number of continuous variables in each component of each approach and the second row represents the number of binary decision variables in each component of each approach. Note that ":" separates the number of decision variables within each component of an approach. FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

able to find a feasible solution when the number of tasks is 5 or 10, and the number of users and artifacts are set to 10, which is by no means a large problem. Future research will consider problem sizes that more closely mimic real-world situations, which will require more complex heuristics. Greedy randomized adaptive search procedures (GRASP) (Hirsch et al., 2010) is a promising

Table 7
Number of decision variables broken out by each component of each approach

No. of tasks	No. of artifacts	No. of users	FF	DA	DB	DC
10	3	2	152,335	1 : 152,335	144,463 : 152,335	1 : 144,463 : 152,335
			44,112	20 : 44,092	36,252 : 43,860	20 : 36,232 : 43,860
10	3	3	343,512	1 : 343,512	324,684 : 343,512	1 : 324,684 : 343,512
			97,092	30 : 97,062	81,387 : 96,705	30 : 81,357 : 96,705
10	3	5	959,956	1 : 959,956	901,126 : 959,956	1 : 901,126 : 959,956
			266,925	50 : 266,875	225,675 : 266,250	50 : 225,625 : 266,250
10	3	10	3,904,391	1 : 3,904,391	3,602,231 : 3,904,391	1 : 3,602,231 : 3,904,391
			1,083,000	100 : 1,082,900	901,500 : 1,081,500	100 : 901,400 : 1,081,500
10	5	2	230,183	1 : 230,183	216,463 : 230,183	1 : 216,463 : 230,183
			49,960	20 : 49,940	36,260 : 49,700	20 : 36,240 : 49,700
10	5	3	520,764	1 : 520,764	486,684 : 520,764	1 : 486,684 : 520,764
			109,830	30 : 109,800	81,405 : 109,425	30 : 81,375 : 109,425
10	5	5	1,464,176	1 : 1,464,176	1,351,126 : 1,464,176	1 : 1,351,126 : 1,464,176
			304,975	50 : 304,925	225,725 : 304,250	50 : 225,675 : 304,250
10	5	10	6,040,831	1 : 6,040,831	5,402,231 : 6,040,831	1 : 5,402,231 : 6,040,831
			1,285,200	100 : 1,285,100	901,700 : 1,283,500	100 : 901,600 : 1,283,500
10	10	2	426,903	1 : 426,903	396,463 : 426,903	1 : 396,463 : 426,903
			66,680	20 : 66,660	36,280 : 66,400	20 : 36,260 : 66,400
10	10	3	973,344	1 : 973,344	891,684 : 973,344	1 : 891,684 : 973,344
			149,550	30 : 149,520	81,450 : 149,100	30 : 81,420 : 149,100
10	10	5	2,777,226	1 : 2,777,226	2,476,126 : 2,777,226	1 : 2,476,126 : 2,777,226
			436,850	50 : 436,800	225,850 : 436,000	50 : 225,800 : 436,000
10	10	10	11,854,431	1 : 11,854,431	9,902,231 : 11,854,431	1 : 9,902,231 : 11,854,431
			2,074,200	100 : 2,074,100	902,200 : 2,072,000	100 : 902,100 : 2,072,000

For the number of tasks equal to 10, and each combination of artifacts and users, the first row represents the number of continuous variables in each component of each approach, and the second row represents the number of binary decision variables in each component of each approach. Note that ":" separate the number of decision variables within each component of an approach. FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

heuristic that will be explored to solve the full formulation, as well as incorporating GRASP into the decomposition strategies.

References

Ausiello, G., Protasi, M., Marchetti-Spaccamela, A., Gambosi, G., Crescenzi, P., Kann, V., 1999. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag, New York.

Garey, M.R., Johnson, D.S., 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., New York.

Hirsch, M.J., Pardalos, P.M., Resende, M.G.C., 2010. Speeding up continuous GRASP. *European Journal of Operational Research* 205, 3, 507–521.

Hirsch, M.J., Schroeder, D., Maggiar, A., Dolinskaya, I., 2014. Multi-depot vessel routing problem in a direction dependent wavefield. *Journal of Combinatorial Optimization* 28, 1, 38–57.

Joshi, A., 2003. Reactive scheduling in workflow management systems: a branch-and-price approach. Master's thesis, University at Buffalo, Buffalo, NY.

Kuipers, E.J., 2003. An algorithm for selecting and timetabling requests for an earth observation satellite. Bulletin de la Société Francaise de Recherche Opérationnelle et d'Aide à la Décision 7–10.

Schurgers, C., Tsiatsis, V., Ganeriwal, S., Srivastava, M., 2002. Topology management for sensor networks: exploiting latency and density. Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Lausanne, Switzerland, pp. 135–145.

Yick, J., Mukherjee, B., Ghosal, D., 2008. Wireless sensor network survey. *Computer Networks: The International Journal of Computer and Telecommunications Networking* 52, 12, 2292–2330.

Appendix

A.1. Linearized formulation

In this section, we present the linearization of the MINLP (Equations (1)–(38)) presented in Section 2. We can linearize the MINLP formulation, defined in Equations (1)–(38), as follows. To begin, we can replace the objective function, Equation (1), with ξ and adding the additional constraints:

$$\xi \geq \hat{E}_i \quad \forall i \tag{A1}$$

$$\xi \in [0, \mathcal{H}]. \tag{A2}$$

We replace Equation (5) with

$$\hat{S}_i = \sum_{j=1}^{\mathcal{P}} \sum_{\ell=1}^{\mathcal{T}} \mu_{ij\ell} \tag{A3}$$

and adding the additional constraints

$$\mu_{ij\ell} \leq S_{j\ell} \quad \forall i, j, \ell \tag{A4}$$

$$\mu_{ij\ell} \geq S_{j\ell} - (1 - z_{ij\ell})\mathcal{H} \quad \forall i, j, \ell \tag{A5}$$

$$\mu_{ij\ell} \leq \mathcal{H} z_{ij\ell} \quad \forall i, j, \ell \tag{A6}$$

$$\mu_{ij\ell} \geq 0 \quad \forall i, j, \ell. \tag{A7}$$

We replace Equation (6) with

$$\hat{E}_i = \sum_{j=1}^{\mathcal{P}} \sum_{\ell=1}^{\mathcal{T}} \nu_{ij\ell} \quad \forall i \tag{A8}$$

and adding the additional constraints

$$\nu_{ij\ell} \leq E_{j\ell} \quad \forall i, j, \ell \tag{A9}$$

$$\nu_{ij\ell} \geq E_{j\ell} - \left(1 - z_{ij\ell}\right)\mathcal{H} \quad \forall i, j, \ell \tag{A10}$$

$$\nu_{ij\ell} \leq \mathcal{H} z_{ij\ell} \quad \forall i, j, \ell \tag{A11}$$

$$\nu_{ij\ell} \geq 0 \quad \forall i, j, \ell. \tag{A12}$$

We replace Equation (11) with

$$S_{j\ell} \geq I_{in} O_{\hat{\imath}n} \delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \quad \forall n, \ell, \hat{\ell}, j, \hat{\jmath}, i, \hat{\imath} \tag{A13}$$

and add the constraints

$$\alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \leq z_{ij\ell} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell} \tag{A14}$$

$$\alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \leq z_{\hat{\imath}\hat{\jmath}\hat{\ell}} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell} \tag{A15}$$

$$\alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \geq z_{ij\ell} + z_{\hat{\imath}\hat{\jmath}\hat{\ell}} - 1 \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell} \tag{A16}$$

$$\gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \leq E_{\hat{\jmath}\hat{\ell}} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell} \tag{A17}$$

$$\gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \geq E_{\hat{\jmath}\hat{\ell}} - (1 - \alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}})\mathcal{H} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell} \tag{A18}$$

$$\gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \leq \mathcal{H} \alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell} \tag{A19}$$

$$\gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \geq 0 \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell} \tag{A20}$$

$$\delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \leq \gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, n \tag{A21}$$

$$\delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \geq \gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} - \left(1 - \Phi_{\hat{\jmath}jn}\right)\mathcal{H} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, n \tag{A22}$$

$$\delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \leq \Phi_{\hat{\jmath}jn} \mathcal{H} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, n \tag{A23}$$

$$\delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \geq 0 \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, n \tag{A24}$$

$$\alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \in \{0, 1\} \quad \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}. \tag{A25}$$

We replace Equation (20) with

$$r_{\hat{j}n} \geq \sum_{i=1}^{\mathcal{T}} \psi_{i\hat{j}n} \quad \forall n, \hat{j} \tag{A26}$$

and add the additional constraints

$$\psi_{i\hat{j}n} \leq \hat{E}_i \quad \forall i, \hat{j}, n \tag{A27}$$

$$\psi_{i\hat{j}n} \leq \varphi_{i\hat{j}n}\mathcal{H} \quad \forall i, \hat{j}, n \tag{A28}$$

$$\psi_{i\hat{j}n} \geq \hat{E}_i - \left(1 - \varphi_{i\hat{j}n}\right)\mathcal{H} \quad \forall i, \hat{j}, n \tag{A29}$$

$$\psi_{i\hat{j}n} \geq 0 \quad \forall i, \hat{j}, n. \tag{A30}$$

Similarly, we replace Equation (27) with

$$\hat{S}_i \geq I_{in}\kappa_{ijn} \quad \forall n, j, i \tag{A31}$$

and the additional constraints

$$\kappa_{ijn} \leq \bar{r}_{jn} \quad \forall i, j, n \tag{A32}$$

$$\kappa_{ijn} \leq y_{ij}\mathcal{H} \quad \forall i, j, n \tag{A33}$$

$$\kappa_{ijn} \geq \bar{r}_{jn} - \left(1 - y_{ij}\right)\mathcal{H} \quad \forall i, j, n \tag{A34}$$

$$\kappa_{ijn} \geq 0 \quad \forall i, j, n. \tag{A35}$$

Finally, we replace Equation (30) with

$$\bar{r}_{jn} \geq \Phi_{\hat{j}jn}t_{k+1} - \lambda_{\hat{j}jnk}t_{k+1} \quad \forall n, k, \hat{j}, j \tag{A36}$$

and add the constraints

$$\lambda_{\hat{j}jnk} \leq c_{\hat{j}jnk} \quad \forall n, k, \hat{j}, j \tag{A37}$$

$$\lambda_{\hat{j}jnk} \leq \Phi_{\hat{j}jn} \quad \forall n, k, \hat{j}, j \tag{A38}$$

$$\lambda_{\hat{j}jnk} \geq c_{\hat{j}jnk} + \Phi_{\hat{j}jn} - 1 \quad \forall n, k, \hat{j}, j \tag{A39}$$

$$\lambda_{\hat{j}jnk} \in \{0, 1\} \quad \forall n, k, \hat{j}, j. \tag{A40}$$

Therefore, the MINLP has been transformed into the problem of minimizing ξ subject to the constraints (2)–(4), (7)–(10), (12)–(19), (21)–(26), (28), (29), (31)–(38), and (A1)–(A40). The resulting formulation is an MILP and can be solved using a number of commercial software packages.

A.2. Detailed results

Tables A1–A6 present the results for each Monte-Carlo scenario, partitioned by number of tasks and number of artifacts. Columns 1 and 2 list the number of users and the Monte-Carlo scenario instance. Columns 3–6 list the objective solution value found for the full formulation (FF), Decomposition A (DA), Decomposition B (DB), and Decomposition C (DC) approaches, respectively. For those problem instances where the FF approach took less than five minutes to run (and hence verifiably found the optimal solution) a decomposition approach that found a solution satisfying the optimality gap is presented in boldface type. For all problem instances, when a decomposition approach found a solution satisfying the optimality gap (with the optimal solution substituted with the best solution over all approaches) the decomposition approach solution value is presented with an asterisk. As can be seen from these tables, for small problem sizes, CPLEX is able to solve the FF to optimality. For these small-sized problems, Decomposition B appears to be doing better than Decompositions A and C. However, as the problem size increases, CPLEX is no longer able to solve the FF to optimality, and the quality of the Decomposition B solution decreases with respect to that of Decompositions A and C. Eventually the FF and Decomposition B are not able to find solutions in CPLEX, while Decompositions A and C are still able to find solutions. When the problem size reaches the largest combination of parameters considered, shown in the last portion of Table A6, none of the approaches is able to find a solution because of memory allocation issues. The memory allocation issues were internal to Matlab being able to actually create the linear program, but would be expected eventually in any programming language, as a result of the large size of the linear programs to model the information workflow problem. Tables 5–7 give a better understanding of the size of the linear program for the FF as well as the three decomposition heuristics. Lastly, it is noteworthy that there exist cases where CPLEX returns very large solution values for the FF approach (e.g., instance 3 for five users in Table A1, instance 6 for two users in Table A6, etc.). These are not typos. In these specific instances, CPLEX is able to find a feasible solution for the FF, however these solutions are of very poor quality as compared to the heuristic decomposition solutions.

Table A1
Comparison between approaches, with number of tasks fixed to 5 and number of artifacts fixed to 3

No. of users	Instance no.	FF	DA	DB	DC
2	1	**43***	46	**43***	46
2	2	**40***	52	**40***	52
2	3	**27***	**27***	**27***	**27***
2	4	**62***	**62***	**62***	**62***
2	5	**72***	79	**72***	79
2	6	**57***	**57***	**57***	**57***
2	7	**65***	117	**65***	117
2	8	**67***	**67***	**67***	**67***
2	9	**84***	87	**84***	87
2	10	**40***	**40***	**40***	**40***
3	1	**29***	58	58	58
3	2	**33***	60	**33***	60
3	3	**34***	39	39	39
3	4	**25***	30	30	30
3	5	**58***	65	**58***	65
3	6	**33***	58	34	58
3	7	**58***	**58***	**58***	**58***
3	8	**34***	60	40	60
3	9	**34***	45	45	45
3	10	**78***	88	**78***	88
5	1	**58***	89	**58***	89
5	2	29*	39	33	39
5	3	1297	44	25*	44
5	4	**68***	82	84	82
5	5	**34***	54	**34***	54
5	6	**44***	55	**44***	55
5	7	83	83	61*	83
5	8	**22***	40	37	40
5	9	**26***	**26***	**26***	**26***
5	10	**22***	27	27	27
10	1	F	63	60*	63
10	2	F	40	39*	40
10	3	F	53*	53*	53*
10	4	F	25*	F/M	25*
10	5	F	71	10*	71
10	6	F	71	46*	71
10	7	F	23*	23*	23*
10	8	F	45*	55	45*
10	9	F	33	23*	33
10	10	F	60	56*	60

Boldface entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the optimal solution. "∗" entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the best solution found. "F" entries denote where CPLEX was not able to find a feasible solution for a given approach, "M" entries denote a memory allocation issue in writing the CPLEX input file for a given approach, and "F/M" entries denote when one of the components of a decomposition approach resulted in a "F" or "M." FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

Table A2
Comparison between approaches, with number of tasks fixed to 5 and number of artifacts fixed to 3

No. of users	Instance no.	FF	DA	DB	DC
2	1	**75***	**75***	**75***	**75***
2	2	**72***	**72***	**72***	**72***
2	3	**90***	**90***	**90***	**90***
2	4	**70***	**70***	**70***	**70***
2	5	**70***	**70***	**70***	**70***
2	6	**65***	87	87	87
2	7	**80***	**80***	**80***	**80***
2	8	**58***	**58***	**58***	**58***
2	9	**68***	**68***	**68***	**68***
2	10	**93***	**93***	**93***	**93***
3	1	**42***	71	**42***	71
3	2	32*	33	32*	33
3	3	**64***	75	**64***	75
3	4	48*	57	48*	57
3	5	**56***	73	73	73
3	6	**25***	36	**25***	36
3	7	**70***	72	**70***	72
3	8	**40***	50	50	54
3	9	**40***	66	65	66
3	10	**49***	51	51	51
5	1	1523	68	30*	68
5	2	28	30	19*	30
5	3	62*	85	62*	85
5	4	F	45	32*	45
5	5	**29***	67	**29***	67
5	6	51	42*	42*	42*
5	7	67	46*	46*	46*
5	8	70	71	65*	71
5	9	53*	74	79	79
5	10	F	76	69*	80
10	1	F	32*	32*	32*
10	2	F	27*	F/M	27*
10	3	F	59*	F/M	59*
10	4	F	45	25*	45
10	5	F	86	57*	86
10	6	F	41*	53	41*
10	7	F	50	32*	50
10	8	F	55	53*	55
10	9	F	43*	F/M	43*
10	10	F	64	20*	64

Boldface entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the optimal solution. "$*$" entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the best solution found. "F" entries denote where CPLEX was not able to find a feasible solution for a given approach, "M" entries denote a memory allocation issue in writing the CPLEX input file for a given approach, and "F/M" entries denote when one of the components of a decomposition approach resulted in an "F" or "M." FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

Table A3
Comparison between approaches, with number of tasks fixed to 5 and number of artifacts fixed to 3

No. of users	Instance no.	FF	DA	DB	DC
2	1	49*	81	49*	81
2	2	**88***	**88***	**88***	**88***
2	3	**73***	111	101	111
2	4	**35***	**35***	**35***	**35***
2	5	**60***	**60***	**60***	**60***
2	6	53*	74	78	74
2	7	**38***	**38***	**38***	**38***
2	8	**72***	112	112	112
2	9	**35***	80	80	80
2	10	**73***	**73***	**73***	**73***
3	1	**80***	90	**80***	90
3	2	F	86	65*	86
3	3	F	64*	64*	64*
3	4	F	81	59*	81
3	5	F	67*	67*	74
3	6	**91***	**91***	**91***	**91***
3	7	74*	76	76	76
3	8	**67***	75	**67***	75
3	9	**65***	**65***	**65***	**65***
3	10	42*	52	52	52
5	1	F	131	56*	131
5	2	F	68	32*	68
5	3	F	97	47*	97
5	4	F	69	34*	69
5	5	F	104*	104*	104*
5	6	F	65	45*	65
5	7	F	71	70*	71
5	8	F	44*	77	44*
5	9	F	115	101*	115
5	10	F	127	121*	127
10	1	M	M	M	M
10	2	M	M	M	M
10	3	M	M	M	M
10	4	F	F	F	F
10	5	M	M	M	M
10	6	M	M	F/M	M
10	7	M	M	M	M
10	8	M	M	M	M
10	9	M	M	M	M
10	10	F	F	F/M	F

Boldface entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the optimal solution. "∗" entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the best solution found. "F" entries denote where CPLEX was not able to find a feasible solution for a given approach, "M" entries denote a memory allocation issue in writing the CPLEX input file for a given approach, and "F/M" entries denote when one of the components of a decomposition approach resulted in an "F" or "M." FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

Table A4
Comparison between approaches, with number of tasks fixed to 5 and number of artifacts fixed to 3

No. of users	Instance no.	FF	DA	DB	DC
2	1	F	117*	F/M	117*
2	2	F	60	F/M	54*
2	3	371	71*	116	105
2	4	F	82*	F/M	82*
2	5	145	74*	F/M	74*
2	6	F	81*	F/M	81*
2	7	F	59*	F/M	59*
2	8	115	60*	94	88
2	9	F	101*	F/M	101*
2	10	F	80*	F/M	80*
3	1	F	143*	F/M	143*
3	2	F	86*	F/M	86*
3	3	F	40*	F/M	40*
3	4	F	68*	F/M	68*
3	5	F	54*	F/M	54*
3	6	F	109*	F/M	109*
3	7	F	109*	F/M	109*
3	8	F	61*	F/M	61*
3	9	F	86*	F/M	86*
3	10	F	77*	F/M	77*
5	1	F	53*	F/M	53*
5	2	F	52*	F/M	52*
5	3	F	56*	F/M	56*
5	4	F	47*	F/M	47*
5	5	F	69*	F/M	69*
5	6	F	60*	F/M	60*
5	7	F	60*	F/M	60*
5	8	F	50*	F/M	50*
5	9	F	85*	F/M	85*
5	10	F	102*	F/M	102*
10	1	F	43*	F/M	43*
10	2	F	40*	F/M	40*
10	3	F	95*	F/M	95*
10	4	F	31*	F/M	31*
10	5	F	55*	F/M	55*
10	6	F	140*	F/M	140*
10	7	F	65*	F/M	65*
10	8	F	47*	F/M	47*
10	9	F	24*	F/M	24*
10	10	F	50*	F/M	50*

Boldface entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the optimal solution. "$*$" entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the best solution found. "F" entries denote where CPLEX was not able to find a feasible solution for a given approach, "M" entries denote a memory allocation issue in writing the CPLEX input file for a given approach, and "F/M" entries denote when one of the components of a decomposition approach resulted in an "F" or "M." FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

Table A5
Comparison between approaches, with number of tasks fixed to 5 and number of artifacts fixed to 3

No. of users	Instance no.	FF	DA	DB	DC
2	1	139	100*	F/M	100*
2	2	F	60*	F/M	60*
2	3	F	74*	F/M	74*
2	4	F	64*	F/M	65
2	5	F	96*	F/M	96*
2	6	**122***	**122***	**122***	**122***
2	7	F	108*	F/M	108*
2	8	F	72*	F/M	130
2	9	F	100*	F/M	100*
2	10	F	65*	F/M	F/M
3	1	F	51*	F/M	51*
3	2	F	50*	F/M	50*
3	3	F	54*	F/M	54*
3	4	F	111*	F/M	111*
3	5	F	86*	F/M	86*
3	6	F	51*	F/M	51*
3	7	F	71*	F/M	71*
3	8	F	53*	F/M	53*
3	9	F	52*	F/M	64
3	10	F	113*	F/M	113*
5	1	F	93*	F/M	93*
5	2	F	86*	F/M	86*
5	3	F	40*	F/M	40*
5	4	F	98*	F/M	98*
5	5	F	38*	F/M	40
5	6	F	61*	F/M	61*
5	7	F	86*	F/M	86*
5	8	F	126*	F/M	126*
5	9	F	96*	F/M	96*
5	10	F	59*	F/M	59*
10	1	F	59*	F/M	59*
10	2	F	68*	F/M	68*
10	3	F	139*	F/M	139*
10	4	F	60*	F/M	60*
10	5	F	74*	F/M	74*
10	6	F	40*	F/M	40*
10	7	F	57*	F/M	352
10	8	F	48*	F/M	48*
10	9	F	106*	F/M	106*
10	10	F	F	F/M	5670*

Boldface entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the optimal solution.* entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the best solution found. 'F' entries denote where CPLEX was not able to find a feasible solution for a given approach, 'M' entries denote a memory allocation issue in writing the CPLEX input file for a given approach, and 'F/M' entries denote when one of the components of a decomposition approach resulted in a 'F' or 'M'. FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.

Table A6
Comparison between approaches, with number of tasks fixed to 5 and number of artifacts fixed to 3

No. of users	Instance no.	FF	DA	DB	DC
2	1	F	125*	F/M	125*
2	2	F	110*	F/M	110*
2	3	F	115*	F/M	115*
2	4	F	91*	F/M	91*
2	5	1277	94*	95	94*
2	6	1509	152	118	90*
2	7	F	119*	F/M	119*
2	8	F	115*	115*	115*
2	9	F	385	F/M	105*
2	10	F	107*	F/M	107*
3	1	F	84*	F/M	84*
3	2	F	53*	F/M	53*
3	3	F	96*	F/M	96*
3	4	F	97*	F/M	97*
3	5	F	100	F/M	69*
3	6	F	56*	F/M	57
3	7	F	62*	F/M	64
3	8	F	2525	F/M	68*
3	9	F	243	F/M	80*
3	10	F	82*	F/M	82*
5	1	F	48*	F/M	48*
5	2	F	4980	F/M	52*
5	3	F	44*	F/M	44*
5	4	F	50*	F/M	50*
5	5	F	65*	F/M	66
5	6	F	77*	F/M	77*
5	7	F	56*	F/M	66
5	8	F	F	F/M	109*
5	9	F	72*	F/M	72*
5	10	F	61*	F/M	61*
10	1	M	M	M	M
10	2	M	M	M	M
10	3	M	M	M	M
10	4	M	M	M	M
10	5	M	M	M	M
10	6	M	M	M	M
10	7	M	M	M	M
10	8	M	M	M	M
10	9	M	M	M	M
10	10	M	M	M	M

[a]Boldface entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the optimal solution. "$*$" entries denote those solutions satisfying the optimality GAP (Equation (44)) with respect to the best solution found. "F" entries denote where CPLEX was not able to find a feasible solution for a given approach, "M" entries denote a memory allocation issue in writing the CPLEX input file for a given approach, and "F/M" entries denote when one of the components of a decomposition approach resulted in an "F" or "M." FF, full formulation; DA, Decomposition A; DB, Decomposition B; DC, Decomposition C.