Cooperative Task Scheduling for Personal Identity Verification in Networked Systems

Koki Inoue∗, Dai Suzuki†, Toshihiko Kurita∗, Satoshi Imai∗†
∗Fujitsu Laboratories Ltd., Kawasaki, Japan
†Fujitsu Limited, Tokyo, Japan
{inoue.koki, suzuki-dai, kurita, imai.satoshi}@fujitsu.com

Abstract—Several working groups are coping with an ecosystem in which a user manages his/her own digital identity (ID) information among different organizations or companies in a decentralized manner. Accordingly, we developed a platform for trusted ID exchange called IDentiTY eXchange. In the platform, the personal identity verification process will be realized by verifying credentials about users' information issued by other organizations. Through this kind of ID cooperation, users can prove their ID online using the credentials and will no longer need to take procedure for every organization when updating their ID information registered there. To update their ID information among multiple organizations, users have to plan a schedule that represents an order of ID cooperation requests from each organization to other organizations. However, the organizations' policies to identify a user and relationships among the organizations make the scheduling problem a complicated one. In this study, we formulate a scheduling problem in the cooperative network as an integer linear program and propose a heuristic method based on the graph structure. Numerical experiments show that the heuristic approach has feasible scalability for practical use. Finally, we discuss other use cases of the proposed formulation, especially regarding logistics management, such as vehicle scheduling for transporting products from suppliers to consumers.

Index Terms—blockchain network, PIV (personal identity verification), DID (decentralized identity), SSI (self-sovereign identity), logistics, ILP (integer linear programming), graph algorithms

I. INTRODUCTION

Recently, some working groups have been coping with the specification of decentralized identity (DID). The DID technology realizes the concept of self-sovereign identity (SSI), which enables individuals to control their own identity (ID) information under self-management [1], [2] and allows individuals to take ownership of their own digital IDs that can be used across multiple organizations without depending on a certain single organization [3]–[5]. Accordingly, we are developing the IDentiTY eXchange (IDYX) technology as a novel DID mechanism [6]. IDYX contributes to the social implementation of SSI as a platform where users can request others to issue a certificate of their own personal information and then securely exchange it among users and where receivers can verify the certificate associated with DID.

As regards the present self-management of personal ID information, people have too many IDs and thus miss the comprehensive knowledge of their own ID information. A survey in 2017 shows that an average company employee manages as many as 191 IDs and passwords [7]. Thus, individuals should need a system to facilitate the complicated ID management. For example, when a user's information (e.g., address, telephone number, affiliation, and family name) is updated, the user may have trouble taking each procedure for every organization to edit the registered information. Specified organizations must execute complicated personal identity verification (PIV) processes in accordance with the law. The PIV process has several forms, all of which consume time and effort from the users. In ordinary cases, users are required to visit the store to take a know-your-customer (KYC) process or required to mail certain public identification documents (e.g., a copy of residence certificate, driver’s license, and passport). IDYX offers “inter-organization ID cooperation,” which provides a mechanism for participant organizations to identify a user's ID information based on certificates issued by other participants who have authenticated the ID information. The participant organizations can exchange certificates of ID information under the agreement of the users, and they can authenticate the ID information by verifying the certificates on behalf of the PIV process with themselves. As a result, the ID cooperation will reduce the load on users in taking the PIV processes in every organization.

However, a “policy” of each organization, that is, the possibility of users’ ID cooperation based on the trust relationship between organizations, and requirement of the number of certificates issued by other organizations necessary for the authentication of users should be considered. For example, Organization A may deny the ID cooperation with competitor B. Organization C may proceed updating ID information based on a certificate from a single organization, but Organization D may require proofs from two organizations to check the integrity of ID information. Organization E may say that an in-person procedure is mandatory and additional certificates from another organization is preferred. A user cannot easily find an efficient schedule to exchange his/her own ID information among the organizations while satisfying their complicated policies. In this study, we deal with the scheduling problem to complete the user authentication process for all the organizations of interest: To which organizations the users themselves should visit to take PIV process? From which organizations to which organizations the certificates of the users’ ID information should be exchanged? The whole process should be optimized to minimize the cost (e.g., time)
of the procedure taken by the user himself/herself without violating any organization’s policies.

As regards the scheduling problem, several studies have been performed in various fields, such as business process reengineering and manufacturing scheduling. They are divided into two problems: (i) a problem to allocate resources to each task along an inputted workflow and (ii) a problem to obtain a workflow itself. The scope of this study is the scheduling problem classified as (ii), to which researchers have considered similar problems. In [8], the authors optimized the order of task execution in the manufacturing system from the viewpoint of minimizing the cost of manufacturing delay and maximizing the efficiency of machine utilization. However, executing conditions, such as dependencies among tasks, have not been considered. In [9], the authors also addressed the problem of optimizing the order of inspection items in the quality assurance of products. The approach attempts to recognize inferior products in the earlier stage of inspection sequences by preferentially executing severe test items. It considers dependencies between tests, such that some tests should refer to the results of other specific tests. However, the optimization problem does not construct a workflow that completes all the test items. Both existing studies cannot deal with the scheduling problem emphasized in this study. For this problem, a scheduling method is required to consider complex relationships between tasks, such that the identification of the ID information will be completed only when multiple certificates issued by different organizations are obtained. Moreover, it is required to construct a workflow that will complete all the tasks to be executed while satisfying the policies.

In this paper, we propose a method to minimize the whole cost of PIV processes among users and to promote ID cooperation among organizations while satisfying each organization’s policy by formulating the scheduling problem as an integer linear programming (ILP). In addition, we propose a heuristic method to reduce the calculation time for the optimization of the scheduling problem. We expect that the proposed formulation has potential to solve problems in various fields. Finally, we discuss the generalization of the characteristics of the method and determine application prospects in the fields of supply chain management (SCM) [10], especially logistics. The contributions of this study are as follows.

- We define task scheduling problems with complex dependencies between tasks, such as inter-organization ID cooperation.
- We propose an ILP optimization scheme for obtaining a task workflow.
- To reduce the calculation time of the optimization problem, we propose a heuristic method, which is based on a graph algorithm.
- In the numerical experiment, we evaluate the performance of the ILP method and heuristic method.
- We suggest the possibility to apply the proposed formulation in logistics problems.

The remainder of this paper is organized as follows: In Section II, we introduce IDYX. In Section III, we define the scheduling problem in the inter-organization ID cooperation on the IDYX platform. In Section IV, we propose an ILP formulation to calculate the optimal schedule for ID cooperation and propose a heuristic method to reduce the calculation time of the optimization problem. Then, in Section V, we conduct numerical experiments to evaluate the performance of the heuristic method on the basis of the ILP solution as a benchmark. In Section VI, we discuss the applicability of the proposed method for other use cases. Finally, in Section VII, we summarize this paper and discuss future efforts.

II. IDENTITY EXCHANGE (IDYX)

IDYX is an expanded DID technology based on Hyperledger Indy [11], which is a blockchain framework supporting verifiable credentials with a distributed ledger system. It enables participants (organizations and individuals) performing online transactions with each other to evaluate the authenticity of the opponent’s information.

Figure 1 illustrates the abstracts of the IDYX platform. The user can request other participants in the IDYX platform to certify their ID information. The issuer, who plays a role in certifying the user’s ID information, issues the certificate (called credential) of the ID information and signs it with its private key. The issuer places the DID and corresponding public key on the blockchain and then transports the credential to the holder (the user) off-chain. The issuer may also place a credential schema on the blockchain. The schema determines the attributes that are certified in its credentials about the ID information. Then, the holder can generate a verifiable presentation (called proof) consisting of the attributes of his/her ID information certified by the issuer. The holder can then prove the attributes to another participant by sending the proof. The verifier who receives the proof can verify the authenticity of the proof information. The verifier pulls the DID of the issuer from the blockchain to authenticate the issuer and pulls the public key to verify the issuer’s signature on the received proof with it. The blockchain guarantees that the DID and key are never falsified; therefore, the verifier can trust the ID
users trust in the issuing organization. After the PIV phase, each organization updates the ID information based on the received PIV information and presents it to the organization at the end-point. The above-mentioned exchange of the certificate about the ID information is automatically executed in accordance with a rule, which the user has already agreed in advance.

- Receiving and verifying a proof: The organization that receives the proof verifies to confirm the authenticity of the ID information.
- Updating ID information: Each organization has its policy, which defines how many verified proofs must be collected to credit the ID information. The organization updates the ID information when it confirms the consistency among the required number of verified data.

In the case of Fig. 2, the user first takes the PIV process in organizations A and B to update his/her attributes registered there. After the PIV processes, organizations A and B issue credentials on the user’s ID information. The user generates two proofs from each of the credentials and presents them to Organization C. Then, Organization C verifies the two proofs and authenticates the ID information by identifying the attribute information certified by organizations A and B. Now, Organization C has completed updating the registered attributes of the user and thus can issue the credential about the ID information. Then, the user generates the proofs from the credentials issued by organizations A and C and presents them to Organization D. Finally, Organization D authenticates the ID information by verifying the proofs.

III. PROBLEM DEFINITION

In this paper, we address the scheduling problem of constructing a workflow. The obtained workflow is referred as the ID cooperation, as described in Section II. The inputs and output of the scheduling problem are as follows.

- Inputs:
  - Set of organizations to which the user requests update of his/her registered ID information
  - Set of policies that are defined by the individual organizations
- Output:
  - Workflow that describes an order of ID exchanges. It specifies the origin and destination of the ID information (in the format of an array of tuples, such as \{user, organization\} or \{organization, organization\})

The workflow must be a sequence of ID cooperation that satisfies that all the organizations can complete the update process of the ID information without any violation of the organizations’ policies. Different organizations will have different policies. Thus, each organization is assumed to configure and disclose its own policy when participating in the IDXY platform.

The users in the ID cooperation may authenticate themselves to organizations by taking the PIV processes there. They may also exploit other organizations to certify the authenticity of the ID information. In the worst case, the users visit every organization for the respective PIV process. The objective of
this problem is to formulate a workflow that will require the user the minimum PIV processes. In the latter case, the organization requires a greater number of proofs from other organizations than the number defined in its policy. The organization may further require some certificates from other organizations in addition to executing the PIV process. It is assumed to be the case when executing an online electronic KYC, which may be less trustful than the in-person KYC process.

The organizations’ policies also define from/to which organizations they approve to exchange ID information. The certificate of ID information is permitted to be delivered only to the organizations that the issuer approves. Moreover, the verifiers receive proofs only from the issuers they approved. The update process of the ID information in each organization is assumed to be completed just when its requirement for authentication is satisfied. Only the organization that has completed the update process of the ID information can issue the certificate to other organizations.

IV. PROPOSED METHOD

Considering the scheduling problem described in Section III, we propose a method to construct a workflow that minimizes the cost imposed to the user, e.g., time to proceed the PIV process in each organization. We first propose an ILP method, which is classified into a combinatorial optimization problem (NP-hard). Then, we propose a heuristic method based on an original graph structure, which will derive a quasi-optimized solution but drastically reduce the calculation time rather than the ILP method.

A. Optimization method based on ILP

The design variables and parameters used in the ILP formulation are listed in Tables I and II, respectively.

For the organizations that the user selected, let \( c_i \) be the credibility of the ID information necessary for organization \( i \) to complete the authentication of the user, \( f_i \) be the time imposed to the user for accomplishing the PIV process in organization \( i \), and \( g_i \) be the credibility of the ID information that organization \( i \) will obtain by accomplishing the PIV process. We regard \( c_i \) as the number of certificates (issued by other organizations and verified by organization \( i \)) necessary for organization \( i \) to have credibility enough to complete the authentication of the user. Then, \( g_i \) represents the credibility that the PIV process contributes to achieve the requirements for authentication, and thus, considered to be the number of certificates obtained through the PIV process. In most cases, \( g_i = c_i \). In graph \( G(V,E) \), where the organizations are node set \( V \) and the relationships between organizations are edge set \( E \), when organization \( i \) permits to send certificates of ID information to organization \( j \) and organization \( j \) permits to receive the certificate from organization \( i \), let edge \( e_{ij} \) be included in set \( E \). The parameters above are constant values that should be given as inputs.

We describe the progress of ID cooperation in phases. In phase 1, the user himself/herself takes the PIV process in several organizations. Then, in subsequent phases \( p(\geq 2) \), the authentication process in each organization is performed by exchanging the certificates about the ID information among the organizations. \( x_{p,i,j} \), \( y_{p,i} \), and \( z_i \) are binary variables taking \{0, 1\}. Variable \( x_{p,i,j} \) is a variable that represents the presence or absence of edge between organizations \( i \) and \( j \). In the exchange process from organization \( i \) to organization \( j \), organization \( j \) receives and verifies the proof of the ID information generated based on the credential issued by organization \( i \). \( x_{p,i,j} = 1 \) means that the exchange process from organization \( i \) to \( j \) is performed in phase \( p \), and \( x_{p,i,j} = 0 \) means that no ID exchange will be performed between the two organizations in phase \( p \). In case any exchanges from organization \( i \) will be conducted in phase \( p \), organization \( i \) must have completed the update of the ID information in phase \( p - 1 \) or before. Variable \( y_{p,i} \) is a flag representing whether updating the ID information has been completed or not. \( y_{p,i} = 1 \) means that organization \( i \) satisfies the authentication requirement of the ID information in phase \( p \), and \( y_{p,i} = 0 \) means that the update process is still pending in phase \( p \). Organization \( i \) updates the ID information at the end of the phase when it obtains the credibility of \( c_i \) or more through the PIV process with the user and/or ID exchanges with other organizations. Variable \( z_i \) indicates whether or not the user should take the PIV process in organization \( i \). \( z_i = 1 \) means that the user himself/herself visits organization \( i \) to proceed the PIV process in phase \( p \). \( z_i = 0 \) means that the user never performed the PIV process for organization \( i \).

In graph \( G \), let the maximum value among the hop lengths of the longest paths of all the node pairs be \( H \). Then, the ID information from any organization can be shared by any other organizations in not more than \( H \)-steps of ID exchanges. Therefore, we should design the ID cooperation progress for \( H + 1 \) phases, consisting of phase 1 for the PIV process by the user and maximum of \( H \) phases for ID exchanges among the organizations.

The objective function is defined as

\[
\text{minimize } \sum_j f_j z_j + \epsilon \sum_{p,i,j} px_{p,i,j}.
\] (1)

The first term represents the sum of the cost imposed to the user for the PIV process by himself/herself. The second term represents the total number of certificates that are exchanged between organizations. The optimization attempts to determine a solution to minimize these costs.

Then, we explain the following constraints.
TABLE II
PARAMETERS DEFINED IN THE 0-1 ILP FORMULATION

<table>
<thead>
<tr>
<th>Constants</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_i$</td>
<td>Credibility requirement for organization $i$ to authenticate the user</td>
</tr>
<tr>
<td>$f_i$</td>
<td>Cost, e.g., time, imposed to the user for accomplishing the PIV process in organization $i$</td>
</tr>
<tr>
<td>$g_i$</td>
<td>Credibility obtained to organization $i$ through the PIV process, $g_i \leq c_i$</td>
</tr>
<tr>
<td>$E$</td>
<td>$E \ni e_{i,j}$ (for organization pair $i \rightarrow j$ that is allowed to exchange user IDs)</td>
</tr>
<tr>
<td>$H$</td>
<td>Maximum value among the longest hop lengths calculated over all organization pairs, $p \leq H + 1$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Small positive value, e.g., $\epsilon = 10^{-4}$</td>
</tr>
<tr>
<td>$\mathcal{L}$</td>
<td>Large positive value, e.g., $\mathcal{L} = 10^4$</td>
</tr>
</tbody>
</table>

The constraint
\[
\sum_{e_{i,j} \notin E} x_{p,i,j} = 0 \quad (\forall p),
\]
prevents the ID exchange, which is not approved by either of the organizations. The constraint
\[
\sum_p y_{p,i} \geq 1 \quad (\forall i),
\]
ensures that all the organizations completed the update process of the ID information in the final phase or before. The constraint
\[
\sum_i z_i \geq 1,
\]
allows the user to take the PIV process to at least one organization, which is the origin of the ID information. The constraint
\[
x_{1,i,j} = 0 \quad (\forall i,j),
\]
prevents ID exchanges between any organizations in phase 1. Only the PIV processes by the user are proceeded in phase 1. The constraint
\[
\sum_j x_{p+1,i,j} \leq \mathcal{L} y_{p,i} \quad (\forall i,p),
\]
ensures that only organizations that have completed the update of the ID information in previous phases can issue credentials to other organizations.

The constraint
\[
\sum_p x_{p,i,j} \leq 1 \quad (\forall i,j),
\]
prevents any organization pair from taking duplicate exchanges of the ID information. The constraints (8) and (9) configure the flag variable for each organization in each phase. When an organization achieves the credibility requirement for the ID information, the constraint
\[
g_j z_j + \sum_{k \leq p} \sum_i x_{k,i,j} - c_j \leq \mathcal{L} y_{p,j} - 1 \quad (\forall j,p),
\]
sets $y_{p,j} = 1$ in phase $p$ when organization $j$ obtains enough credibility by receiving more certificates than necessary and/or conducting the PIV process. In subsequent phases $p + 1, p + 2, \ldots$ after the completion of the update process, the flag continues as $y_{p+1,j} = 1, y_{p+2,j} = 1, \ldots$. Moreover, while the update process has not yet been performed, the constraint
\[
g_j z_j + \sum_{k \leq p} \sum_i x_{k,i,j} - c_j \leq \mathcal{L} y_{p,j} - 1 \quad (\forall j,p),
\]
sets $y_{p,j} = 0$ in phase $p$ when the credibility requirement of organization $j$ is not achieved.

B. Heuristic method based on the graph structure

We propose a heuristic method, which first transforms the given organizations’ policies into a weighted directed graph. Then, it obtains a partial solution, i.e., a part of workflow, by solving a general shortest-path algorithm, e.g., Dijkstra’s algorithm, whose computational complexity is $O(n^2)$. Next, the method modifies the edge weights or edge itself of the graph, and then the shortest-path algorithm is solved again. The whole workflow is finally constructed by successively appending the partial solutions, which are obtained by the iterations of the operation above.

The proposed heuristic method is described in Algorithm 1. First, the organizations’ policies are given as the input. Matrix $A = \{a[i][j]\}$ is an adjacency matrix, taking the value of $a[i][j] = 1$ when $e_{i,j} \in E$ and $a[i][j] = 0$ when $e_{i,j} \notin E$.

Variables $S$ and $P$ are flags defined for each organization and will be updated at the end of each iteration of the repetitive operations. Flag $S$ represents the state of the update process of the ID information. It evaluated the achievement/non-achievement of the credibility requirement of each organization while referring to the partial solution already obtained. $S[i] = 0$ means that organization $i$ has not completed the update process, $S[i] = 1$ means that it has just completed the update in the previous iteration, and $S[i] = 2$ means that it has completed the update before the previous iteration. Flag $P$ represents the progress when the user proceeds a PIV process in an organization. In the heuristic, it is assumed that the PIV process that generates $g_i$ credibility takes $q_i$ iterations. $P[i] = -1$ means that the PIV process in organization $i$ has been accomplished by the user, $P[i] = 0$ means that it has never been proceeded, and $P[i] \geq 1$ means that it is in progress.

The heuristic algorithm starts by constructing a directed graph with two-dimensional edge weights based on the organizations’ policies. Each node in the graph is representative of each organization, and the extra nodes $s$ and $t$ are included. The weight of each edge is defined by the set of distance ($D$) and capacity ($C$). The capacity $C$ represents the credibility required by organizations for identifying the user. Let $C[i][j]$ be the credibility $q_i$ acquired by organization $i$ through the PIV process by the user, $C[i][j]$ be the credibility acquired by organization $j$ through the ID exchange from organization $i$ to organization $j$ (it is assumed to be 1 for all the organization pairs in this problem definition), and $C[i][t]$ be the credibility requirement $c_i$ necessary for organization $i$ to classify the authentication of the user as accomplished. Distance $D$ represents the cost. Let $D[i][s]$ be the cost $f_i$ for the user
to proceed the PIV process in organization $i$, and $D[i][j]$ be the cost including the ID exchange from organization $i$ to organization $j$ in the workflow. The initial value of $D[i][j]$ is set to a sufficiently large positive value, which means that the ID exchange is not allowed while the source organization $i$ has not completed the update process. When the requirement for the ID exchange between organizations $i$ and $j$ is achieved, the value of $D[i][j]$ is overwritten with an appropriate value, e.g., a small positive value in this study. Let $D[i][t]$ be a small positive value, which means that the edge to terminal $t$ exists, but the cost is negligible. An example of the graph construction is shown in Fig. 3.

In lines 6–8 of Algorithm 1, the costs for the PIV process are discounted, which are applied just once immediately after the construction of the graph. The discount encourages the user to perform the PIV process preferentially in organizations expected to contribute much cost reduction for other organizations. The discount value is based on the cost that should be saved in case the ID exchange from organization $i$ exempts the user from taking the PIV process in organization $j$.

Lines 9–39 show the repetitive iterations. The algorithm is designed to give one of the organizations a credibility of 1 per iteration. In each iteration, the shortest path based on distance $D$ is calculated, and the capacity $C$ of the edges along the shortest path is decremented by 1. In addition, the graph may be modified within the iteration, as described below. A total of $\sum \ c_i$ iteration times are required to complete the update process for all the organizations.

In lines 10–12, the edges between the organizations that have exchanged the ID information in the previous iteration are deleted to prevent duplicate certification. In lines 13–15, the penalty cost to the PIV process is imposed to organization $j$, which accepts the certificate from organization $i$ when organization $i$ has prepared to issue the credential of the ID information. The penalty discourages the user to proceed to the PIV process in organization $j$ because authentication of the proof from other organizations is expected. In lines 16–18, organization $j$ prevents the user to perform the PIV process in organization $j$ when enough number of organizations have completed the update process of the user ID and prepared to issue credentials to organization $j$. In lines 19–21, the path representing the PIV process in organization $i$ will continue to be the shortest path during the $g_i$-time iterations once the PIV process is launched. In lines 22–24, when organization $i$ has just completed the update process, edges are generated along
the paths that represent the ID exchanges from organization \( i \) to other organizations. In line 25, the shortest path from node \( s \) to \( t \) is calculated by referring to the distance \( D \). The shortest path is obtained in the form of a set of node pairs along the path, such as \( \tau = \{(s, i), (i, j), (j, t)\} \), or \( \tau = \{(s, i), (i, t)\} \), which represents the partial solution of the workflow. The former task represents the ID exchange from organizations \( i \) to \( j \), and the latter represents the PIV process in organization \( i \). Then, in lines 26–28, the capacity \( C \) of each edge along the shortest path is decremented by 1, which means that the corresponding organization has obtained the credibility of 1. Finally, flag variables \( P \) and \( S \) are updated in lines 29–38. The iteration above, i.e., modifying the graph and calculating the shortest path, is executed \( \sum_i c_i \) times.

In line 40, after finishing all the iterations above, the partial solutions \( \tau \) are arrayed in the order in which they are obtained. However, identical tasks representing the PIV process for organization \( i \) will appear \( g_i \) times in the workflow, so the function unique eliminates the overlaps. Then, the algorithm outputs the workflow as a confirmed solution.

### V. Numerical Experiments

In this section, we conduct numerical experiments to evaluate the performance of the proposed methods. We first design a generation model of the organizations’ policies, which will be described below. For each number of the organizations, we evaluate the methods with 100 patterns of policies generated by the model. For each pattern, we execute the ILP-based optimization method (ILP) and the graph-based heuristic method (GbH). Then, we evaluate the obtained workflow performance (i.e., the total PIV costs imposed to the user) and the elapsed time for the program to calculate the solution. The machine used in the experiments are specified by Intel Xeon 3.8 GHz 4-cores CPU, 32GB RAM, and 64-bit OS, and the methods are implemented by R language. The package Rsymphony version 0.1-28 [12] is executed as a solver for ILP. Note that we terminate the calculation within up to time_limit = 3600 s, and the feasible solution found is adopted as the solution in this case. The shortest-path calculation in GbH owes the function shortest_paths of the package igraph version 1.2.4.1 [13]. The function is implemented based on Dijkstra’s algorithm; thus, we normalize all the edge weights to be non-negative.

Here, we explain the model for generating the organizations’ policies. The possibility of an ID exchange between organizations is determined based on a directed graph generated by Barabási-Albert (BA) model [14], which is generally used to describe social relationships. For the executing parameter for the BA model, we set the out-degree of newly added nodes as a random number of \( \{2(50\%)\}3(50\%) \). In the graph, each node is regarded as each organization. Then, we interpret the reversed direction of each edge as the direction that the user’s ID certificate is allowed to transmit between two organizations. The required credibility for authentication \( c_i \) is given by a random number of \( \{1(10\%)\}2(80\%)\}3(10\%) \), the credibility of the PIV process \( g_i \) is given by a random number of \( \{\max\{c_i - 1, 1\}\} \{1(10\%)\} \{c_i(90\%)\} \), and the cost (i.e., time imposed to the user) of the PIV process \( f_i \) is given by a random value according to the log-normal distribution \( \Lambda(\mu = 0, \sigma^2 = 1) \).

For each number of nodes, i.e., 10, 20, 30, and 35, the distribution of the total PIV cost imposed to the user is shown in Fig. 4. For a pattern of the organizations’ policies, the optimal solution by ILP is plotted against the \( x \)-axis and that by GbH against the \( y \)-axis. Most patterns are mapped in the vicinity of \( y = x \) indicated by the dashed line, which means that approximate solutions are obtained by GbH. There are some patterns far from the \( y = x \) line, for which GbH degrades its solution performance. The improvement of the drawback is one of our future works. However, some patterns in the area of \( y \geq x \) exist in the experimental results in 35 nodes, for which the solution by ILP is meant to be inferior to that by GbH. In this number of nodes or more, the optimal solution is not necessarily obtained by ILP in a practical time limit, i.e., 1-h calculation in this experiment; instead, only the feasible solution may be obtained. Thus, the solution by GbH may outperform the feasible solution by ILP, so the performance advantage of ILP is not ensured.

Next, we evaluate the program execution time of ILP and GbH. In the several scales for nodes from 5 to 200, the 95% confidence intervals of the elapsed time for executing both methods are shown in Fig. 5. ILP requires more than 1 min in 30 nodes, and the calculation time by ILP greatly increases in 35 nodes and more. In fact, in the experiment of 35 nodes,
there are as many as 17 cases out of 100 patterns in which ILP fails to find the optimal solution within the time_limit (3600 s) but only results in finding the feasible solution. However, ILP can determine optimum solutions within 10 s in the scale of 20 nodes or less. As for GBH, it requires just a few seconds in 100 nodes and approximately 20 s in 200 nodes. Therefore, it is in flexible to flexibly choose which method to be applied according to the targeted scale for nodes, e.g., ILP can be applied for less than 20 nodes and GBH for 20 nodes and more.

VI. DISCUSSION

Here, we discuss a use case of the proposed paradigm other than the ID cooperation scheduling. Our proposed approach is applicable to scheduling problems with the following characteristics:

(i) Executing each task individually is costly.
(ii) In an assist mechanism, such as ID cooperation, completed tasks can help the progress of other tasks with low cost.
(iii) Some tasks may require multiple assists from completed tasks.
(iv) There are dependencies among tasks, so any-to-any assist may not be permitted.

We regard SCM [10] as one of the fields where the proposed methods can be applied. Particularly, the method can be applied for logistics problems, such as a milk run with cross-docks [15]. In the problem, suppliers have to transport their products/resources to a manufacturer’s plant. There are several transportation ways available, as shown in Fig. 6. The following transportation forms (i)–(iii) correspond to the above characteristics (i)–(iii), respectively. (i) Direct shipping: A truck leaves the terminal and goes to a supplier to collect the products, then transports them to the plant, and return to the terminal. (ii) Milk run (cooperative transportation): On the way from a supplier to the plant, a truck visits another supplier and loads the products and then transports them together. A milk run can save the cost of hiring trucks for every supplier in direct shipping. (iii) Consolidation at a cross-dock: Several trucks send products from different suppliers to a cross-dock. After unloading the products, each truck returns to its terminal. Then, in the cross-dock, products sharing the same destination are aggregated to a container, and the cross-dock sends out a truck that carries the container to the plant. When multiple senders utilize the cross-dock, consolidation is an effective way to reduce the costs of transportation. In addition, as for the characteristic (iv), the logistics problem can take route options in consideration. The route options may be defined by business relationships or geographical connection among the suppliers.

Here, we define an optimization problem to minimize the whole transportation cost, which is counted based on the set distance of trucks. The solution determines (1) suppliers to which trucks should go from the terminal and (2) suppliers (including cross-docks) that each truck should visit as a milk run.

As shown in Table III, we additionally define the differential cost as a given parameter. Here, we explain the cost estimation of exploiting milk runs and cross-docks. Let \( f_{i \rightarrow j} \) be the transportation cost from place \( i \), i.e., one of \{supplier, cross-dock, terminal (T), and plant (P)\}, to place \( j \). Then, direct shipping cost \( f_i \) is described as \( f_i = f_{T \rightarrow A} + f_{A \rightarrow i} + f_{P \rightarrow T} \), as an example shown in Fig. 7. Moreover, the cost of a milk run to pick supplier \( j \) following supplier \( i \) is described as \( f_{T \rightarrow A} + f_{i \rightarrow j} + f_{j \rightarrow P} + f_{P \rightarrow T} \). We define the differential cost as \( \Delta f_{i,j} = f_{i \rightarrow j} - f_{A \rightarrow P} + f_{A \rightarrow i} + f_{i \rightarrow j} + f_{j \rightarrow P} + f_{P \rightarrow T} \). The definition is also applied for a successive milk run that goes through more than two suppliers. As shown in Fig. 8, a milk run picking suppliers \( A, B, \) and \( C \) is given by \( f_A + \Delta f_{A,B} + \Delta f_{B,C} \). For consolidation, the differential cost for a truck at supplier \( i \) to utilize cross-dock \( X \) is defined by \( \Delta f_{i,X} = -f_i + f_P - f_{P \rightarrow X} + f_{X \rightarrow T} \). In addition, the cost for the transportation between the cross-dock and plant \( f_X \) is imposed. As shown in Fig. 9, the total cost of both trucks at suppliers \( A \) and \( B \) utilizing cross-dock \( X \) is given by \( f_A + \Delta f_A,X + f_B + \Delta f_B,X + f_X \). Except for the additional cost \( f_X \), the differential costs \( \Delta f_{A,X} \) and \( \Delta f_{B,X} \) can be treated as if visiting cross-dock \( X \) as a milk run.

Here, we introduce the modification of the proposed ILP
Finally, based on the cost estimation, the objective function is accepted that not each cross-dock will be used, so variables $y_{p,i}$ are interpreted in Table IV. The important variables are presented in Table III, cross-docks should be fixed to the origin and destination of terms and plant are fixed to the origin and destination of terminals, and the problem. The important variables are interpreted in Table IV. As for the given parameters in Table III, cross-docks should be fixed to the origin and destination of terminals, and the problem. We first regard the logistics problem presented above. We first regard the logistics problem presented above, we need to evaluate the flag in the last phase.

VII. CONCLUSION

In this paper, as a background of DID, we define the scheduling problem in the inter-organization ID cooperation and propose a mathematical programming method and heuristic method to solve the problem. Through these methods, users can easily control the distribution of their own ID information on the complex web of organizations. Numerical experiments show that the proposed heuristic method is executable in a practical time even with an increased number of nodes where the optimization method may halt. However, the solutions are sometimes degraded from the optimum performance, and there is a room for improvement of the method and parameter adjustment. Furthermore, the methods proposed in this paper are expected to be applied to services or social problems other than the ID cooperation. We also discuss the characteristics of the proposed approach and customize the proposed method to solve the problem of minimizing truck transportation costs in logistics. The findings show the applicability of the method to another use case, but future work is needed to evaluate the performance of the customized method.

REFERENCES