Minimum Equivalent Precedence Relation Systems

Kin Cheong Sou

Abstract—In this paper two related simplification problems for systems of linear inequalities describing precedence relation systems are considered. Given a precedence relation system, the first problem seeks a minimum equivalent subset of the precedence relations (i.e., inequalities) which has the same solution set as that of the original system. The second problem is similar to the first one, but the minimum equivalent system need not be a subset of the original system. This paper shows that the first problem is NP-hard. However, a sufficient condition is derived under which the first problem is solvable in polynomial-time. In addition, a decomposition of the first problem into independent tractable and intractable subproblems is derived. The second problem is shown to be solvable in polynomial-time, with a full parameterization of all solutions described. The results in this paper generalize those in [2], [3] for unweighted directed graphs. The relation between weighted directed graphs and precedence relations problem and transitive reduction problem, which are applicable to unweighted directed graphs.

I. INTRODUCTION

A. Problem statement

In this paper we consider precedence relation systems with \( n \) variables of the form:

\[
x_i - x_j \leq c_{ij}, \quad (i, j) \in E,
\]

where \( c_{ij} \in \mathbb{R} \) and \( E \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\} \) are given, and \( x_1, x_2, \ldots, x_n \) are the variables. \( E \) is the index set of all precedence relations in (1). Let \( c(E) \in \mathbb{R}^{|E|} \) be the vector of edge weights such that if the \( k \)th element of \( E \) is \((i, j)\), then \( c_k = c_{ij} \). Also, let \( V = \{1, 2, \ldots, n\} \) denote the set of variable indices. Then, system (1) is described by the triple \((V, E, c(E))\).

This paper considers two related problems regarding the simplification of precedence relation systems.

The first problem – maximum index set of redundant relations problem: we seek a maximum (cardinality) index set of redundant relations, with the definition of an index set of redundant relations given by:

Definition 1 (Index set of redundant relations): Let \((V, E, c(E))\) be a precedence relation system, then \( R \subseteq E \) is called an index set of redundant relations of \((V, E, c(E))\) if

\[
x \text{ satisfies } (V, E, c(E)) \iff x \text{ satisfies } (V, (E \setminus R), c(E \setminus R)).
\]

In this paper, two precedence relation systems are equivalent (with symbol \( \equiv \)) if they have the same solution set. Therefore, condition (2) can be stated as \((V, (E \setminus R), c(E \setminus R)) \equiv (V, E, c(E))\). In its minimization form, the first problem can be posed as finding a minimum (cardinality) subset \( E' \subseteq E \) (i.e., \( E' = E \setminus R \)) such that \((V, E', c(E')) \equiv (V, E, c(E))\).

The second problem – equivalent reduction problem: we seek an equivalent reduction of a given precedence relation system \((V, E, c(E))\) as follows:

Definition 2 (Equivalent reduction): Let \((V, E, c(E))\) be a precedence relation system. A precedence relation system \((V, E', c'(E'))\) with \( E' \subseteq V \times V \) and \( c'(E') \in \mathbb{R}^{|E'|} \) is an equivalent reduction of \((V, E, c(E))\) if it satisfies the following two conditions:

1. a. \( x \in \mathbb{R}^n \) satisfies \((V, E, c(E))\) iff \( x \) satisfies \((V, E', c'(E'))\).
2. b. w.r.t. property 2.a. \( E' \) has the minimum cardinality.

The maximum index set of redundant relations problem (i.e., the first problem) is a restriction of the equivalent reduction problem (i.e., the second problem). In the search of the minimum equivalent system of precedence relations for the first problem (in its minimization version) we are restricted to a subset of (1), whereas in the second problem there is no such restriction. In addition, according to definition the values in \( c' \) in the second problem need not be the same as \( c \) as in the first problem.

B. Summary of main contributions and previous works

The main contributions of this paper are as follows:

1. We derive a sufficient condition under which the maximum index set of redundant relations problem has a unique solution and is polynomial-time solvable. In addition, we show that in general the maximum index set of redundant relations problem is NP-hard.
2. We show that the maximum index set of redundant relations problem can be decomposed into a finite number of independent subproblems, one of which being polynomial-time solvable and the rest NP-hard.
3. Based on the decomposition, we provide a parameterization of all solutions to the maximum index set of redundant relations problem.
4. We provide a complete parameterization of all solutions to the equivalent reduction problem. The parameterization suggests a procedure that can find any solution to this problem in polynomial time.

Due to space limitation, in this paper all statements are presented without proof. The proofs can be found in [1].

In essence, the results in this paper are generalizations of those in [2], [3]. The generalization is in the sense that the results in [2], [3] pertain unweighted directed graphs, while the results in this paper pertain weighted directed graphs. The relation between weighted directed graphs and precedence relations problem and transitive reduction problem, which are applicable to unweighted directed graphs.
inequality systems will be explained in the sequel. The main finding in this paper is that even in the more general weighted directed graph setting, the complexity and decomposition results for the graph versions of the main problems are analogous to the results for the restricted unweighted cases in [2], [3]. In comparison with methods to simplify general sets of linear inequalities (e.g., [4], [5]), this paper proposes specialized and time-efficient algorithms for the special setting of precedence relation inequalities in (1). Note also that the results in this paper pertain to minimum equivalent precedence relation inequalities, instead of minimal representations of general inequality systems (e.g., [6]).

C. Application motivations

The precedence relation system in (1) arises in many application domains (e.g., [7]–[10]). For instance, [11] considered the following nonconvex resource allocation problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} f_i(x_i) \\
\text{subject to} & \quad x_i - x_j \leq c_{ij}, \quad \forall (i,j) \in E \\
& \quad \sum_{i=1}^{n} g_i(x_i) \leq b,
\end{align*}
\]

where \( f_i, g_i \) are arbitrary given functions, \( E \subseteq V^2 \) with \( V = \{1, \ldots, n\} \), and \( C \) is a given finite set. [11] showed that the computation cost to solve (3) increases exponentially with the cardinality of the minimum feedback vertex set of the undirected version of \((V,E)\). By solving the precedence relation system simplification problems in this paper, and replacing \((V,E)\) with an equivalent \((V,E')\) with \(|E'| < |E|\), it is possible to substantially reduce the time to solve (3).

D. Organization

Section II defines the precedence graph associated with (1). It also establishes the equivalence between the algebraic notion of index set of redundant relations with a precedence graph-based concept to be defined as redundant edge set. Section III develops the complexity and decomposition results for the problem of finding the maximum redundant edge set. Section IV focuses on equivalent reduction problem. It characterizes the set of all equivalent reductions, and provides a polynomial-time algorithm to obtain the reductions.

II. PRELIMINARIES

A precedence graph \( (V,E,c(E)) \) is an edge weighted directed graph with node set, edge set and vector of edge weights being \( V \), \( E \) and \( c(E) \) respectively. In addition, a precedence graph corresponds to a precedence relation system sharing the same triple \((V,E,c(E))\) as follows:

| \( V \) | node set | \( E \) | edge set | \( c_{ij} \) for \((i,j) \in E\) | edge weight | \( x_i - x_j \leq c_{ij} \) |
|---|---|---|---|---|---|---|

A walk from \( u \in V \) to \( v \in V \) is defined as \((u = i_0,i_1,\ldots,i_m = v)\) where \((i_k,i_{k+1}) \in E\) for \( k = 0,\ldots,m-1 \) and the traversed nodes \( i_0,\ldots,i_m \) are not necessarily distinct. A closed walk is a walk \((i_0,i_1,\ldots,i_m = v)\) such that \( i_0 = i_m \). A (simple) path from \( u \) to \( v \) is a walk such that all traversed nodes are distinct. A cycle is a closed walk where \( i_0 = i_m \) and all other traversed nodes are distinct. The weight of a walk (e.g., path, cycle) is the sum of the weights of all traversed edges, with the edge weight added as many times as an edge is traversed. This paper allows degenerate path \((u)\) for \( u \in V \) (i.e., a single node), and its weight is zero. The following symbols will be used throughout the paper. For any two nodes \( u \) and \( v \), the symbol \( p_{uv} \) is used to denote a path from \( u \) to \( v \). The weight of a path \( p_{uv} \) is denoted \( c_{p_{uv}} \). Similarly, \( w_{uv} \) denotes a walk from \( u \) to \( v \) with weight \( c_{w_{uv}} \). The symbol \( u \rightarrow v \) means “from \( u \) to \( v \)”. The symbol \( u \rightsquigarrow v \) means an edge from \( u \) to \( v \).

Because of the associated precedence inequalities, a precedence graph satisfies the following property:

Lemma 1: Let \( G = (V,E,c(E)) \) be a precedence graph. If the corresponding precedence inequality system has at least one solution, then the weights of all closed walks (e.g., cycles) in \( G \) are nonnegative.

In this paper, we make the following assumptions on precedence graphs:

Assumption 1:

1.a All precedence inequality systems have at least one solution. Hence, no precedence graph contains any negative weight closed walks according to Lemma 1.
1.b There is no parallel edges between nodes in any graph.
1.c There is no self-loop in any graph.

Assumption 1.a states that we consider only consistent precedence inequality systems. Assumptions 1.b and 1.c are natural since we search for minimum precedence graphs.

The algebraic object of index set of redundant relations defined in (2) can be reinterpreted as a precedence graph based object of redundant edge set to be defined in Definition 3.

Definition 3 (Redundant edge set): For an edge weighted directed graph \( G = (V,E,c(E)) \), an edge subset \( R \subseteq E \) is called a redundant edge set of \( G \) if either \( R = \emptyset \) or when \( R \neq \emptyset \), it holds that

\[ c_{p_{uv}} \leq c_{w_{uv}} \]

for every \((u,v) \in R \) with weight \( c_{w_{uv}} \), there exists a path \( p_{uv} \) in \((V,E \setminus R, c(E \setminus R)) \) from \( u \) to \( v \) such that \( c_{p_{uv}} \leq c_{w_{uv}} \).

The following statement formalizes equivalence between index set of redundant relations and redundant edge set.

Lemma 2: Let \( G = (V,E,c(E)) \) denote a precedence relation system and its precedence graph. Then \( R \subseteq E \) is an index set of redundant relations of \( G \) (Definition 1) if and only if it is a redundant edge set of \( G \) (Definition 3).

In view of Lemma 2, the algebraic problem of finding the maximum index set of redundant relations in (1) can be posed as the graph problem of finding the maximum (cardinality) redundant edge set in the precedence graph associated with (1), with the definition of redundant edge set given in (4).
III. MAXIMUM REDUNDANT EDGE SET

A. Complexity of maximum redundant edge set problem

The following concept and statement determine if a given precedence graph has nonempty redundant edge sets:

Definition 4 (Redundant edge): For an edge weighted directed graph \( G = (V, E, c(E)) \), an edge \((i,j) \in E\) is called a redundant edge of \( G \) if the singleton \( \{(i,j)\} \) is a redundant edge set of \( G \) (defined in (4)).

Lemma 3: Let \( G = (V, E, c(E)) \) be an edge weighted directed graph. Then, there exists a nonempty redundant edge set in \( G \) if and only if there exists a redundant edge in \( G \).

In addition, if there is a redundant edge, then the maximum weight of the walk among all walks in \( G \) which goes from \( i \) to \( j \). Note that \( d_{ij} = 0 \) for all \( i \in V \), and this is attained by the single-node path \( (i) \) since \( G \) does not have any negative weight closed walks.

Definition 5 (Equivalence relation \( \sim \)): Let \( G = (V, E, c(E)) \) denote an edge weighted directed graph without negative weight closed walks. Let relation \( \sim \) be defined in Definition 6. We define the following:

7.a \( K \) denotes the number of equivalence classes in \( V \) defined by relation \( \sim \).

7.b For \( k \in \{1, 2, \ldots, K\} \), \([v_k]\subseteq V \) denotes the equivalence class containing \( v_k \), where \( v_k \) is the (arbitrarily) designated representing node the equivalence class.

7.c For \( k \in \{1, 2, \ldots, K\} \), we denote

\[ E_k := (i, j) \in E \mid c_{ij} > d_{ij} \]  

In fact, edges in \( E_k \) are redundant because their weights are “greater than necessary”. This motivates the use of superscript “r” in (5).

Let \( E_k \subseteq E_k \) be defined as

\[ E_k := (i, j) \in E_k \mid c_{ij} = d_{ij} \]  

That is, \( E_k \subseteq E_k \) since \( c_{ij} \geq d_{ij} \) for all \((i,j) \in E \). This motivates the superscript “c”, since \( E_k \) is the complement of \( E_k \).

7.d For \( i \in \{1, 2, \ldots, K\}, j \in \{1, 2, \ldots, K\}, i \neq j \) we denote

\[ E_{ij} := \{(u,v) \in E \mid u \in [v_i], v \in [v_j]\} \]

That is, \( E_{ij} \) denotes the set of edges connecting two nodes inside an equivalence class \([v_i]\).

Definition 6 (Minimum walk weight): Let \( G = (V, E, c(E)) \) be an edge weighted directed graph without negative weight closed walks. For \( i \in V, j \in V \), define \( d_{ij} \) to be the minimum weight of the walk among all walks in \( G \) which goes from \( i \) to \( j \). Note that \( d_{ii} = 0 \) for all \( i \in V \), and this is attained by the single-node path \( (i) \) since \( G \) does not have any negative weight closed walks.

Definition 7 (Equivalence classes induced by relation \( \sim \)): Let \( G = (V, E, c(E)) \) denote an edge weighted directed graph without negative weight closed walks. Let relation \( \sim \) be defined in Definition 5. Let \( d_{ij} \), the minimum walk weight in \( G \), be defined in Definition 6. We define the following:

- \( E_k := (i, j) \in E_k \mid c_{ij} > d_{ij} \)
- \( E_k := (i, j) \in E_k \mid c_{ij} = d_{ij} \)

B. Decomposition of maximum redundant edge set problem

We first introduce an equivalence class in the node set for weight graphs, generalizing strongly connected component for unweighted graphs. Then, we define an auxiliary graph induced by the equivalence class partitioning called condensation. After that, we establish the fact that the maximum redundant edge set problem can be decomposed into \( K + 1 \) independent subproblems, where \( K \) is the number of equivalence classes. Theorem 3 summarizes the main result.

Theorem 1: Let \( G := (V, E, c(E)) \) be an edge weighted directed graph. Assume that all cycles in \( G \) have positive weights. Then the maximum redundant edge set is unique. In addition, if there is a redundant edge, then the maximum redundant edge set is also a redundant edge set.

The first main result of the paper concerns a sufficient condition to guarantee, among other properties, the polynomial-time solvability of the maximum redundant edge set problem.

Theorem 2: Let \( G \) be an edge weighted directed graph. The problem of finding the maximum redundant edge set of \( G \) is NP-hard.

In summary, Assumption 1.a states that a precedence graph \( G \) cannot have negative cycles. If in addition there is no zero-weight cycles, Theorem 1 states that the maximum redundant edge set of \( G \) is unique, and it is the set of all redundant edges. On the other hand, in the more general case where zero-weight cycles are not prohibited, Fig. 1 indicates that the maximum redundant edge set need not be unique. In addition, Theorem 2 states that the maximum redundant edge set problem is NP-hard in general.

![Fig. 1: A 3-node graph with zero weight cycles and two maximum redundant edge sets (i.e., \{\{1,2\}\} and \{\{1,3\}\}).](image)
7. Collecting all inter-equivalence class edges, we denote
\begin{align*}
E_0 & := \bigcup_{1 \leq i \neq j \leq K} E_{ij} \\
E_0' & := \bigcup_{1 \leq i \neq j \leq K} E_{ij}' \\
E_0'' & := \bigcup_{1 \leq i \neq j \leq K} \{(v_{ij}', v_{ij}'')\}
\end{align*}
where  \(1 \leq i \neq j \leq K\) is shorthand for \(\{(i, j) \mid 1 \leq i, j \leq K, i \neq j\}\). It holds that \(E_0'' \subseteq E_0' \subseteq E_0\).

Definition 8 (Condensation): Let \(G = (V, E, c(E))\) denote an edge weighted directed graph without negative weight closed walks. Let other symbols involved be defined in Definitions 5, 6, 7 in the context of \(G\). We define the condensation of \(G\), denoted \(\tilde{G} := (\tilde{V}, E_0, \tilde{c}(E_0))\) as follows:

- The set of all nodes of \(\tilde{G}\) is \(\tilde{V} := \{v_1, v_2, \ldots, v_K\}\).
- In \(\tilde{G}\), there is an edge \((v_i, v_j) \in E_0\) with \(i \neq j\) if and only if in \(G\) the set \(E_{ij} \neq \emptyset\) (see Definition 7.d for \(E_{ij}\)).
- For any \((v_i, v_j) \in E_0\), the edge weight \(\tilde{c}(v_{ij})\) is defined as
  \[ \tilde{c}(v_{ij}) := \min_{(u, v) \in E_{ij}} d_{uv} + c_{uv} + d_{wj}, \]
  where \(d_{uv}\) and \(d_{wj}\) are defined in Definition 6.
- According to the definition of \(v_{ij}'\) and \(v_{ij}''\) in Definition 7.d, it holds that
  \[ \tilde{c}(v_{ij}) = d_{v_{ij}'} + c_{ij} + d_{v_{ij}''}. \]
- The vector of edge weights is denoted \(\tilde{c}(E_0)\).

Fig. 2 illustrates an example of two equivalence classes defined by relation \(\sim\). The condensation of the graph in Fig. 2 is illustrated in Fig. 3.

Fig. 2: The number on each edge denotes the weight of the corresponding edge. The first equivalence class is \([1] = \{1\}\), \([2, 3, 4, 5]\) is the other equivalence class \([2]\). The intra-equivalence class edge sets are \(E_1 = \emptyset\), \(E_2 = \{(2, 5), (3, 2), (3, 4), (4, 2), (5, 3)\}\) and \(E_2' = \{(3, 2)\}\). The inter-equivalence class edge sets are \(E_{12} = \{(1, 2)\}\) and \(E_{21} = \{(3, 1)\}\). The representing edges for \(E_{12}\) and \(E_{21}\) are, respectively, \((v_{12}', v_{12}'') = (1, 2)\) and \((v_{21}', v_{21}'') = (3, 1)\). In this example, \(E_0 = E_0' = E_0'' = \{(1, 2), (3, 1)\}\).

Fig. 3: The condensation of the graph in Fig. 2. There are two edges in the condensation: \((v_1, v_2) = (1, 2)\) and \((v_2, v_3) = (2, 1)\) (i.e., \(E_0 = \{(1, 2), (2, 1)\}\). Their weights are \(c_{12} = d_{v_1 v_2} + c_{v_2 v_1} + d_{v_1 v_2} = 0 + 1 + 0 = 1\), and \(c_{21} = d_{v_2 v_1} + c_{v_2 v_1} + d_{v_2 v_1} = -2 + 2 + 0 = 0\).

For \(k \in \{1, 2, \ldots, K\}\), \([v_k]\) denotes the equivalence class defined in Definition 7.b.
- For \(k \in \{1, 2, \ldots, K\}\), \(E_k'\) and \(E_k''\) are defined in Definition 7.c.
- For \(1 \leq i \neq j \leq K\), \(E_{ij}'\) and \(E_{ij}''\) are defined in Definition 7.d.
- \(\tilde{G}\) is the condensation of \(G\) defined in Definition 8.
Every maximum redundant edge set of \(G\) can be parameterized by
\[ R_0^* \cup \bigcup_{k=1}^{K} (E_k' \cup R_k^*), \]
where for \(k \in \{1, 2, \ldots, K\}\), \(R_k^*\) is a maximum redundant edge set of the subgraph \(([v_k], E_k', c(E_k'))\). In addition, \(R_0^*\) is
\[ R_0 := \bigcup_{1 \leq i \neq j \leq K} R_{ij}', \]
and \(R_0^*\) is the maximum redundant edge set of \(\tilde{G}\), the condensation of \(G\).

As a result of Theorem 3, the graph in Fig. 2 with its maximum redundant edge set removed is illustrated in Fig. 4.

Fig. 4: As a result of Theorem 3, the maximum redundant edge set of the graph in Fig. 2 is \(R^* = \{(3, 2)\}\) (in general the maximum redundant edge set need not be unique). In the parameterization in (11), \(K = 2\), \(E_1' = \emptyset\) and \(E_2' = \{(3, 2)\}\). \(R_0^* = \emptyset\), and by inspection \(R_2^* = \emptyset\) since the subgraph \(([2], E_2' = \{(2, 5), (3, 3), (3, 4), (4, 2)\}, c(E_2'))\) is a zero-weight cycle. From Fig. 3, the maximum redundant edge set of the condensation \(\tilde{G}\) is \(R_0^* = \emptyset\). Hence, from (13) \(R_{12}^* = R_{21}^* = \emptyset\) (since, for instance, \(E_{12} = E_{12}' = \{(1, 2)\}\)

C. Computation for maximum redundant edge set
To compute the quantities in the statement of Theorem 3, the first step is the identification of the equivalence classes...
\([v_1], [v_2], \ldots, [v_k]\) defined by relation \(\sim\). The following statement and algorithm are useful in the identification:

**Lemma 4:** Let \(G = (V, E, c(E))\) be an edge weighted directed graph without negative weight closed walks. For \(i, j \in V\), let \(d_{ij}\) be the minimum walk weight defined in Definition 6. Then, \(i \sim j\) if and only if \(d_{ij} + d_{ji} = 0\). ■

**Algorithm 1:**

1. Solve the all-pair shortest path problem for all source/destination pairs in \(G\). Let \(d_{ij}\) denote the shortest path distance from \(i\) to \(j\).
2. For each pair of \(1 \leq i \neq j \leq n\), declare \(i \sim j\) if and only if \(d_{ij} + d_{ji} = 0\). Build an undirected graph \((V, E_\sim)\) such that edge \((i, j) \in E_\sim\) if and only if \(i \sim j\) and \(i \neq j\).
3. The equivalence classes defined by relation \(\sim\) are the connected components of \((V, E_\sim)\).

The first step of Algorithm 1 can be computed using Floyd-Warshall algorithm in \(O(|V|^3)\) time. The second step requires \(O(|V|^2)\) time. The third step requires \(O(|V| + |E_\sim|) = O(|V|^2)\) time. Hence, Algorithm 1 requires \(O(|V|^2)\) time. Once the equivalence classes have been identified, the computation involved in Definition 7 and Definition 8 requires \(O(|E|)\) time and \(O(K^2)\) time respectively.

The next step in utilizing Theorem 3 is to compute the maximum redundant edge set \(R_0^\ast\) for condensation \(\tilde{G}\). The following statement is vital for the computation:

**Lemma 5:** Let \(G = (V, E, c(E))\) be an edge weighted directed graph without negative weight closed walks. Let \(\tilde{G} = (\tilde{V}, \tilde{E}_0, \tilde{c}(E_0))\) be the condensation of \(G\) in Definition 8. Then, the weights of all cycles in \(\tilde{G}\) are positive. ■

As a consequence of Lemma 5, Theorem 1 applies and it states that \(R_0^\ast\) is the collection of all redundant edges (see Definition 4) of condensation \(\tilde{G}\). The following algorithm identifies the redundant edges in \(\tilde{G}\) (cf. Lemma 7 in [1]):

**Algorithm 2:**

1. Solve the all-pair shortest path problem for all source/destination pairs in \(\tilde{G} = (\tilde{V}, \tilde{E}_0, \tilde{c}(E_0))\). Let \(d_{ij}\) denote the shortest path distance from \(i\) to \(j\).
2. An edge \((i, j) \in E_0\) is declared a redundant edge iff

\[
\min_{(i, k) \in E_0, k \neq i, k \neq j} \{\tilde{c}_{ik} + \tilde{d}_{kj}\} \leq \tilde{c}_{ij},
\]

(14)

Algorithm 2 requires \(O(K^3)\) time (with \(K \leq |V|\)). The last but most demanding part in utilizing Theorem 3 is to compute \(R_k^\ast\) for \(k = 1, 2, \ldots, K\) in (11). The complexity for solving for \(R_k^\ast\) can be argued as follows: the maximum redundant edge set problem on subgraph \([v_k], E^k_\ast, c(E^k_\ast)\) generalizes the minimum equivalent graph problem [2] on unweighted directed graph \([v_k], E^k_\ast\). In addition, the latter problem is known to be NP-hard (e.g., [12]). Hence, solving for \(R_k^\ast\) is NP-hard. On the other hand, it turns out that the subproblem for finding a maximum redundant edge set in \([v_k], E^k_\ast, c(E^k_\ast)\) is no harder than the (NP-hard) minimum equivalent graph problem for undirected graph \([v_k], E_k^\ast\). The following statement provides the rationale:

**Lemma 6:** Let \(G = (V, E, c(E))\) be an edge weighted directed graph without negative weight closed walks. For \(k \in \{1, 2, \ldots, K\}\), let \([v_k]\) be the equivalence class defined by relation \(\sim\) (Definition 7.b), and \(E_k^\ast\) be defined in Definition 7.c. Let \(R_k \subseteq E_k^\ast\) be given. Then, the following two statements are equivalent:

6.a \(R_k\) is a redundant edge set of \(([v_k], E_k^\ast, c(E_k^\ast))\).
6.b \(([v_k], E_k^\ast)\) and \(([v_k], E_k^\ast, c(E_k^\ast))\) have the same reachability (i.e., there is a walk \(i \sim j\) in \(([v_k], E_k^\ast)\) if and only if there is a walk \(i \sim j\) in \(([v_k], E_k^\ast, c(E_k^\ast))\). ■

In essence, \(R_k^\ast\) can be obtained by solving the minimum equivalent graph problem on unweighted graph \(([v_k], E_k^\ast)\) obtained from \(([v_k], E_k^\ast, c(E_k^\ast))\) simply by ignoring the edge weights. The details of the algorithms for the unweighted problem can be found in, for example, [2], [13], [14].

**IV. RESULTS ON EQUIVALENT REDUCTION PROBLEM**

This section presents a full parameterization of the set of all equivalent reductions of any precedence relation system. Contrary to the problem of finding the maximum redundant edge set which is NP-hard, every equivalent reduction can be computed in polynomial-time.

Because of the correspondence between a precedence relation system and its precedence graph, in this section we shall extend the notion of equivalent reduction in Definition 2 to precedence graphs. Given a precedence graph \(G\), we call another precedence graph \(G'\) equivalent to \(G\), with notation \(G' \equiv G\), if the precedence relation systems corresponding to \(G\) and \(G'\) are equivalent (i.e., they have the same solution set). Consequently, an equivalent reduction of a precedence graph \(G\) is a precedence graph \(G' \equiv G\) such that \(G'\) has the minimum possible number of edges.

Analogous to the maximum redundant edge set problem, an equivalent reduction can be decomposed into \(K + 1\) components. The following statement summarizes the decomposition result related to equivalent reduction:

**Theorem 4:** Let \(G = (V, E, c(E))\) be a precedence graph. Let the following be defined in the context of \(G\):

- \(K\) is the number of equivalence classes induced by relation \(\sim\) in Definition 7.a.
- For \(k \in \{1, 2, \ldots, K\}\), \([v_k]\) denotes the equivalence class defined in Definition 7.b, with \(v_k\) being the representing node of \([v_k]\).
- For \(i, j \in V\), \(d_{ij}\) is the minimum walk weight defined in Definition 6.
- For \(1 \leq i \neq j \leq K\), \((v'_i, v'_j)\) are defined in Definition 7.d.
- \(\tilde{G}\) is the condensation of \(G\) defined in Definition 8, in accordance with designation of representing nodes \(v_i\)'s.

Then, every equivalent reduction of \(G\) (defined in Definition 2) can be parameterized by

\[
(V, E^{\ast\prime}, c^{\ast\prime}(E^{\ast\prime})),
\]

(15)

In (15), the edge set \(E^{\ast\prime}\) is parameterized by

\[
E^{\ast\prime} = E_0^\ast \cup \bigcup_{k=1}^{K} E_k^\ast,
\]

(16)

where for \(k \in \{1, \ldots, K\}\), \(E_k^\ast = \emptyset\) if \(|[v_k]| = 1\), otherwise \(E_k^\ast\) contains \(|[v_k]|\) edges forming a zero weight directed
cycle traversing all nodes in \([v_k]\). In addition, \(E''\) can be decomposed into
\[
E'' = \bigcup_{1 \leq i \neq j \leq k} E''_{ij},
\]
where
\[
E''_{ij} = \begin{cases} 
(u, v) \in [v_i] \times [v_j], & \text{if } E_{ij} \neq \emptyset \text{ and } (v_i, v_j) \notin \tilde{R}_0, \\
\emptyset, & \text{if } E_{ij} = \emptyset \text{ or } (v_i, v_j) \in \tilde{R}_0,
\end{cases}
\]
and \(\tilde{R}_0\) is the maximum redundant edge set of \(G\), the condensation of \(G\). In (15), the edge weights \(c''(E'')\) are defined by
\[
c_{uv}'' = \begin{cases} 
d_{uv} + c_{uv} + d_{uv}, & \text{if } (u, v) \in [v_i] \times [v_j], \ i \neq j, \\
d_{uv}, & \text{if } (u, v) \in [v_i] \times [v_i].
\end{cases}
\]

The main difference between the maximum redundant edge set parameterization in Theorem 1 and the equivalent reduction parameterization in Theorem 4 lies in the treatment of the edges inside equivalence classes. For Theorem 1 NP-hard minimum equivalent graph problem needs to be solved for each equivalence class, whereas for Theorem 4 each equivalence class is replaced with a Hamiltonian cycle which is easy to construct (see (18) for the formula for the edge weights). Note also in (17) if two different equivalence classes are to be connected, there is no specific requirement on which two nodes from these two classes are connected, as long as the edge weight satisfies (18). Fig. 5 shows two equivalent reductions of the example graph in Fig. 2.

V. Conclusions

The maximum index set of redundant relations problem (i.e., maximum redundant edge set problem) is a generalization of the minimum equivalent graph problem. Similarly, the equivalent reduction problem is a generalization of the transitive reduction problem. Nevertheless, the generalizations are shown to possess analogous computation properties of the respective restrictions. The maximum redundant edge set problem is NP-hard in general, and it is solvable in polynomial time if the graph does not have zero-weight cycles. This is analogous to the statement that the minimum equivalent graph problem is NP-hard in general, and it is solvable in polynomial time if the graph is acyclic. In addition, the decomposition of the maximum redundant edge set problem based on the equivalence classes defined by the “on-a-zero-weight-closed-walk” relation is analogous to the decomposition of the minimum equivalent graph problem based on strongly connected components. Further, in the decomposition, the subproblems dealing with the edges between equivalence classes are both solvable in polynomial time. The subproblem within an equivalence class for maximum redundant edge set problem is in fact equivalent to the (NP-hard) minimum equivalent graph problem within the corresponding equivalence class, with the implication that all available exact or inexact algorithms for minimum equivalent graph problem can be utilized. Finally, analogous results also hold between the equivalent reduction problem and the transitive reduction problem. The structure of decompositions of the solutions are analogous, and both problems can be solved in polynomial-time using similar shortest path calculations.

REFERENCES

[1] K. C. Sou, “Minimum equivalent precedence relation systems,” 2015. [Online]. Available: http://arxiv.org/abs/1503.04753
[2] D. M. Moyles and G. L. Thompson, “An algorithm for finding a minimum equivalent graph of a digraph,” J. ACM, vol. 16, no. 3, pp. 455–460, Jul. 1969. [Online]. Available: http://doi.acm.org/10.1145/321526.321534
[3] A. Aho, M. Garey, and J. Ullman, “The transitive reduction of a directed graph,” SIAM Journal on Computing, vol. 1, no. 2, pp. 131–137, 1972.
[4] J. Telgen, “Identifying redundant constraints and implicit equalities in systems of linear constraints,” Management Science, vol. 29, no. 10, pp. 1209–1222, 1983.
[5] K. Fukuda, B. Gartner, and M. Szedlak, “Combinatorial redundancy detection,” 2014. [Online]. Available: http://arxiv.org/pdf/1412.1241.pdf
[6] K. Clarkson, “More output-sensitive geometric algorithms,” in Foundations of Computer Science, 1994 Proceedings, 35th Annual Symposium on, Nov 1994, pp. 695–702.
[7] K. C. Sou, J. Weimer, H. Sandberg, and K.H. Johansson, “Scheduling smart home appliances using mixed integer linear programming,” in Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, dec. 2011, pp. 5144–5149.
[8] K.C. Sou, M. Kördel, J. Wu, H. Sandberg, and K.H. Johansson, “Energy and co2 efficient scheduling of smart home appliances,” in European Control Conference, 2013.
[9] C. Chekuri and R. Motwani, “Precedence constrained scheduling to minimize sum of weighted completion times on a single machine,” Discrete Applied Mathematics, vol. 98, no. 1–2, pp. 29–38, 1999.
[10] Y. Li and V. Stojanović, “Yield-driven iterative robust circuit optimization algorithm,” in Proceedings of the 46th Annual Design Automation Conference. ACM, 2009, pp. 599–604.
[11] K. C. Sou, H. Sandberg, and K. H. Johansson, “Nonserial dynamic programming with applications in smart home appliances scheduling – part II: Nonserial dynamic programming,” in European Control Conference, 2014.
[12] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, NY, USA: W. H. Freeman & Co., 1990.
[13] H. T. Hsu, “An algorithm for finding a minimal equivalent graph of a digraph,” J. ACM, vol. 22, no. 1, pp. 11–16, Jan. 1975.
[14] S. Khuller, B. Raghavavari, and N. Young, “Approximating the minimum equivalent digraph,” SIAM Journal on Computing, vol. 24, no. 4, pp. 859–872, 1995.