Fall 2009

A matrix-based approach to searching colored paths in a weighted colored multidigraph

Haiyan Xu

Kevin W. Li Dr.
University of Windsor

D. Marc Kilgour

Keith W. Hipel .

Recommended Citation

Xu, Haiyan; Li, Kevin W. Dr.; Kilgour, D. Marc; and Hipel, Keith W. .. (2009). A matrix-based approach to searching colored paths in a weighted colored multidigraph. *Applied Mathematics and Computation*, 215 (1), 353-366.
http://scholar.uwindsor.ca/odettepub/64

A Matrix-based Approach to Searching Colored Paths in a Weighted Colored Multidigraph

Haiyan Xu[a], Kevin W. Li[b], D. Marc Kilgour[c], and Keith W. Hipel[a*]

[a] *Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada*
[b] *Odette School of Business, University of Windsor, Windsor, Ontario, Canada*
[c] *Department of Mathematics, Wilfrid Laurier University, Waterloo, Ontario, Canada*

Abstract

An algebraic approach to finding all edge-weighted-colored paths within a weighted colored multidigraph is developed. Generally, the adjacency matrix represents a simple digraph and determines all paths between any two vertices, and is not readily extendable to colored multidigraphs. To bridge the gap, a conversion function is proposed to transform the original problem of searching edge-colored paths in a colored multidigraph to a standard problem of finding paths in a simple digraph. Moreover, edge weights can be used to represent some preference attribute. Its potentially wide realm of applicability is illustrated by a case study: status quo analysis in the graph model for conflict resolution. The explicit matrix function is more convenient than other graphical representations for computer implementation and for adapting to other applications. Additionally, the algebraic approach reveals the relationship between a colored multidigraph and a simple digraph, thereby providing new insights into algebraic graph theory.

Key words: Edge-weighted-colored multidigraph; Edge-colored paths; Adjacency matrix; Decision making; Status quo analysis, Graph model for conflict resolution.

1 Introduction

It is well-known that matrices can efficiently describe adjacency of vertices, and incidence of arcs and vertices in a graph, thereby permitting tracking of paths between any two vertices [11]. Matrices possess various algebraic properties, which can be exploited to develop improved algorithms for solving a variety of problems in a graph. As such, extensive research has been conducted to design effective algorithms and efficient search

[*] Corresponding author. E-mail:kwhipel@uwaterloo.ca; Fax:1-519-7464791 (K.W. Hipel).

procedures by exploring relationships between matrices and paths [12,14,25]. The purpose of the colored path searching problem is to find all edge-colored paths between any two vertices in a given colored multidigraph. The traditional approach of employing adjacency matrix for searching paths is furnished in a simple digraph. For general graph classes, searching for particular paths, such as Hamilton paths [1,24], Euler paths, and shortest path routing between two vertices [26], can be solved efficiently, but there exist very limited algorithms to search colored paths for colored multidigraphs.

The capability of searching edge-weighted-colored paths in a weighted colored multidigraph can have many benefits. For instance, Section 4 illustrates how this capability can be conveniently applied to solve an open problem of status quo analysis in conflict resolution. A strategic conflict is a situation in which two or more decision-makers (DMs) have to make independent choices in face of differing preferences about possible outcomes for the DMs. Among the formal methodologies that handle strategic conflict, the Graph Model for Conflict Resolution [7,18] provides a remarkable combination of simplicity and flexibility. As a post-stability analysis in the graph model, status quo analysis examines whether predicted equilibria (or potential resolutions) are reachable from the status quo or the initial state by tracing the moves and countermoves among DMs. Although decision support systems for basic stability analysis [7] and the group analytic network process [17] are available, the status quo analysis algorithms developed in [21,22] have not been implemented as a practical decision support system. In addition, the existing methodology [21,22] does not track all aspects of conflict evolution from the status quo state to a particular outcome. Some research [15] is related to the graph model analysis, but the proposed approach in this paper investigates the relation between the graph model and algebraic graph theory.

An important restriction of a graph model is that no DM can move twice in succession along any path [7]. Hence, a graph model can be conveniently treated as an edge-weighted-colored multidigraph in which each arc represents a legal unilateral move, distinct colors refer to different DMs, and the weight along the arc identifies some preference attribute. Thus, tracing the evolution of a conflict in status quo analysis is converted to searching all colored paths with some preference structure such as simple preference [7], uncertain preference [19], or strength of preference [13]. Therefore, the proposed procedure developed in this paper includes the main results in [21,22] as a special case. The proposed method can be employed for transportation networks. For instance, because of the accelerating globalization trend, a major logistic challenge is to design a reliable, efficient, and economical systems for moving merchandise within a multi-modal transportation network. Due to diverse geography and weather conditions, cost and time constraints, as well as other factors, chartered companies may have to switch their transport mode when passing through a transfer station. In order to design a competitive transportation system, one must analyze all possible paths from any initial station to a destination to make the best choice. This transportation problem can be conveniently modeled as a problem of finding colored paths and the shortest colored path in a weighted colored multidigraph.

Although many approaches and algorithms for coloring vertices and edges have been developed in graph theory and computer science [3], the edge-colored graph research here differs from previous work in that it is not concerned with how to color edges. Instead, the fundamental problem is to search edge-colored paths in a given colored multidigraph. This research is also different from the well-known network analysis problem of finding paths between two vertices due to the additional color restriction feature that is not present in these problems. Therefore, it is difficult to use existing methods or algorithms directly, including genetic algorithm [4] and neural network [26], to find the shortest colored path. In this paper, an adjacency matrix of an undirected line graph is extended to an adjacency matrix of a colored line digraph, thereby providing new insights into algebraic graph theory [11]. Based on the matrix thus designed, a conversion function is proposed to transform a colored multidigraph to a simple digraph so that the original complex problem of searching edge-colored paths in a colored multidigraph is converted to a standard problem of finding paths in a simple digraph with no color constraints.

Additionally, due to the nature of the explicit algebraic expressions, the proposed method is more effective, convenient, flexible, and extendable than existing approaches in terms of the underlying graphs for carrying out path searching with various constraints, and is general enough to allow for many practical applications. [27] and [28] have shown advantages of using algebraic approaches to calculate potential resolutions and track conflict evolution. However, the proposed method [28] is based on the adjacency matrix to search state-by-state paths. If a graph model contains multiple arcs between the same two states controlled by different DMs, the adjacency matrix would be unable to track all aspects of conflict evolution from the status quo state. It is well known that the incidence matrix can represent multidigraphs if all edges are labeled. The proposed algebraic approach starts with developing a unique edge-labeling rule for colored multidigraphs, and then devises a conversion function based on the incidence matrix.

The rest of the paper is organized as follows. In Section 2, several important definitions in graph theory are reviewed. The proposed approach and main results are presented in Section 3. Section 4 demonstrates how the proposed matrix method can be applied by using a case study of the status quo analysis of the Gisborne Lake conflict under uncertain preference [8]. Some comments and insights are furnished in Section 5.

2 Preliminary definitions and extended definitions in the algebraic graph theory

A multidigraph [5] $G = (V, A, \psi)$ is a set of vertices (nodes) V and a multiset of oriented edges (arcs) A with $\psi : A \to V \times V$. If $a \in A$ such that $\psi(a) = (u, v)$, then we say that a has initial vertex u and terminal vertex v. A multidigraph may contain $a, b \in A$ such that $a \neq b$ and $\psi(a) = \psi(b)$, in which case a and b are said to be multiple arcs.

Let $m = |V|$ denote the number of vertices and $l = |A|$ be the number of edges in a multidigraph G. If there exists $a \in A$ such that $\psi(a) = (u, v)$, then u is said to be *adjacent* to v and (u, v) is said to be incident from u and incident to v. Hence, (u, v) is called in-incident to v and out-incident to u. When G is drawn, it is common to represent the direction of an edge with an arrowhead. We generally assume loop-free graphs; i.e., for any $a \in A$, if $\psi(a) = (u, v)$, then $u \neq v$. It should be pointed out that a simple digraph is a directed graph without multiple edges.

Definition 1 *For a multidigraph $G = (V, A, \psi)$, edge $a \in A$ and edge $b \in A$* ***are consecutive (in the order*** *ab***)*** *iff $\psi(a) = (u, v)$ and $\psi(b) = (v, s)$, where $u, v, s \in V$.*

Definition 2 *For a multidigraph $G = (V, A, \psi)$, the* ***line digraph*** *$L(G) = (A, LA)$ of G is a simple digraph in which vertex set is A and oriented edge set is expressed as LA=$\{d = (a, b) \in A \times A$: a and b are consecutive (in the order ab)$\}$.*

Definition 3 *For a multidigraph $G = (V, A, \psi)$, a* ***path*** *from vertex $u \in V$ to vertex $s \in V$ is a sequence of vertices in G starting with u and ending with s, such that consecutive vertices are adjacent.*

Note that in this paper a path may contain the same vertex more than once [2]. The length of a path is the number of edges therein. Definitions 2 and 3 are adapted from [11].

Definition 4 *A* ***colored multidigraph*** *(V, A, N, ψ, c) is a multidigraph (V, A, ψ) and a set of colors N, and a function $c : A \to N$ such that $c(a) \in N$ is the color of $a \in A$, provided that multiple edges of (V, A, ψ) are assigned different colors , i.e., if $a \neq b$, but $\psi(a) = \psi(b)$, then $c(a) \neq c(b)$.*

If $a \in A$ such that $\psi(a) = (u, v)$ and $c(a) = i$ for $i \in N$, then a can be written as $a = d_i(u, v)$. The line digraph of $G = (V, A, N, \psi, c)$, $L(G)$, is a simple digraph and each vertex in $L(G)$ corresponds to an edge in the multidigraph G. Hence, coloring edges in G is equivalent to assigning colors to vertices in $L(G)$.

Definition 5 *For a colored multidigraph $G = (V, A, N, \psi, c)$, the* ***reduced line digraph*** *$L_r(G) = (A, LA_r)$ of G is a simple vertex-colored digraph with vertex set A and edge set LA_r=$\{d = (a, b) \in A \times A$: a and b are consecutive (in the order ab) and $c(a) \neq c(b)\}$.*

Definition 6 *A* ***weighted colored multidigraph*** *(V, A, N, ψ, c, w) is a colored multidigraph (V, A, N, ψ, c) together with a map $w : A \to \mathbb{R}_0^+$ (the set of non-negative real numbers).*

Thus an arc $a \in A$, $a = d_i(u, v)$, carries a weight $w(a)$, representing some attribute of the move from node u to node v along the arc a, which is assigned color i. A network, for instance, is a multidigraph with weighted edges. A weighted edge-colored path is defined as follows:

Definition 7 *For a weighted colored multidigraph (V, A, N, ψ, c, w), an* ***edge-weighted-colored path*** *is a path in the multidigraph (V, A, ψ) in which each constituent edge carries a weight $w(a) \geq 0$ and any two consecutive edges have different colors.*

Definition 8 *For a weighted colored multidigraph (V, A, N, ψ, c, w), the* ***shortest colored path between two vertices*** *is the colored path that minimizes the sum of the weights of its constituent edges.*

In this paper, a colored multidigraph (V, A, N, ψ, c) is a unit weighted colored multidigraph if $w(u, v) = 1$ for any $a \in A$ such that $\psi(a) = (u, v)$. Many well-known algorithms have been developed to solve the shortest path problems in digraphs, such as Dijkstra's algorithm [6] and Johnson's algorithm [16]. Some other algorithms are available for searching for all paths in undirected graphs, such as the algorithm presented by Migliore et al [23]. Because these algorithms are not based on algebraic representations, it is not easy to extend them to the case of finding colored paths.

Let w_a denote the weight of edge a. Then the weight matrix of a weighted colored multidigraph (V, A, N, ψ, c, w) is defined as follows:

Definition 9 *The* ***weight matrix*** *W is an $l \times l$ diagonal matrix such as its (k, k) entry $W(k, k) = w_k$, where w_k denotes the weight of arc a_k.*

A weighted line digraph $L^{(W)}(G) = (A, LA, w)$ is a set of vertices A together with a set of oriented edges LA, and a map $w : A \to \mathbb{R}_0^+$.

In traditional graph coloring problems, such as vertex coloring and edge coloring, colors are assigned to vertices or edges such that adjacent vertices or consecutive edges have different colors, and the number of colors needed is minimized [5]. In this paper, the edge-weighted-colored graph problem is not concerned with coloring edges, but it aims at searching edge-weighted-colored paths in a given colored multidigraph.

Important matrices associated with a digraph include the adjacency matrix and the incidence matrix [11].

Definition 10 *For a multidigraph (V, A, ψ), the* ***adjacency matrix*** *is the $m \times m$ matrix J with (u, v) entry*

$$J(u, v) = \begin{cases} 1 & \text{if } (u, v) \in A, \\ 0 & \text{otherwise}, \end{cases}$$

where $u, v \in V$.

The incidence matrix B can be extended to the weighted incidence matrix.

Definition 11 *For a weighted multidigraph (V, A, ψ, w), the* ***weighted incidence ma-***

trix *is the* $m \times l$ *matrix* $B^{(W)}$ *with* (v, a) *entry*

$$B^{(W)}(v,a) = \begin{cases} -w_a & \textit{if } a = (v,x) \textit{ for some } x \in V, \\ w_a & \textit{if } a = (x,v) \textit{ for some } x \in V, \\ 0 & \textit{otherwise,} \end{cases}$$

where $v \in V$*,* $a \in A$*, and* $w(a) = w_a$*.*

According to the positive entries and negative entries, the weighted incidence matrix can be separated into weighted in-incidence and weighted out-incidence matrices.

Definition 12 *For a weighted multidigraph* (V, A, ψ, w)*, the* ***weighted in-incidence matrix*** $B_{in}^{(W)}$ *and the* ***weighted out-incidence matrix*** $B_{out}^{(W)}$ *are* $m \times l$ *matrices with* (v, a) *entries*

$$B_{in}^{(W)}(v,a) = \begin{cases} w_a & \textit{if } a = (x,v) \textit{ for some } x \in V, \\ 0 & \textit{otherwise,} \end{cases}$$

and

$$B_{out}^{(W)}(v,a) = \begin{cases} -w_a & \textit{if } a = (v,x) \textit{ for some } x \in V, \\ 0 & \textit{otherwise,} \end{cases}$$

where $v \in V$*,* $a \in A$*, and* $w(a) = w_a$*.*

It is obvious that $B_{in}^{(W)} = (B^{(W)} + |B^{(W)}|)/2$ and $B_{out}^{(W)} = (B^{(W)} - |B^{(W)}|)/2$, where $|B^{(W)}|$ denotes the matrix in which each entry equals the absolute value of the corresponding entry of $B^{(W)}$. Let I denote the identity matrix. If $W = I$, then $B^{(W)} = B$, $B_{in}^{(W)} = B_{in}$, and $B_{out}^{(W)} = B_{out}$.

Definition 13 *For two* $m \times m$ *matrices* M *and* Q*, the* ***Hadamard product*** *for the two matrices is the* $m \times m$ *matrix* $H = M \circ Q$ *with* (s, q) *entry*

$$H(s,q) = M(s,q) \cdot Q(s,q).$$

3 An algebraic approach to searching colored paths

3.1 A Rule of Priority to label colored arcs

A colored multidigraph may contain several arcs with the same initial and terminal vertices, but each arc in this case must be assigned a different color. To work with the set of all arcs, we must label them carefully. Assuming that all colors and nodes are pre-numbered. Therefore, the vertex set V and the color set N in $G = (V, A, N, \psi, c)$ are numbered as $V = \{1, 2, \cdots, m\}$ and $N = \{1, 2, \cdots, n\}$, respectively. Let c_i denote the cardinality of arc set assigned color i, i.e., $c_i = |A_i|$, where $A_i = \{x \in A : c(x) = i\}$ for each $i \in N$.

To label the arcs in a colored multidigraph $G = (V, A, N, \psi, c)$, set $\varepsilon_0 = 0$ and $\varepsilon_i = \sum\limits_{j=1}^{i} c_j$ for $i \in N$, and note that $l = \varepsilon_n = \sum\limits_{i=1}^{n} c_i$ is the cardinality of A in G. The arcs, $a_1, a_2, \ldots, a_l$, will be labeled according to the color order; within each color, according to the sequence of initial nodes; and within each color and initial node, according to the sequence of terminal nodes. The ordering, referred to as the *Rule of Priority*, has the following properties:

(1) If $\varepsilon_{i-1} < k \leq \varepsilon_i$, then $c(a_k) = i$, i.e., a_k has color i;
(2) For $k < h$, if a_k and a_h both have color i for some $i \in N$, and if $\psi(a_k) = (v_x, v_y)$ and $\psi(a_h) = (v_z, v_w)$, then $x \leq z$ and, if $x = z$, then $y < w$.

If all arcs in a colored multidigraph have been labeled according to the Rule of Priority, then the index of an arc uniquely determines its color. Therefore, $A_i = \{a_{\varepsilon_{i-1}+1}, \ldots, a_{\varepsilon_i}\}$, where A_i denotes the set of arcs with color i.

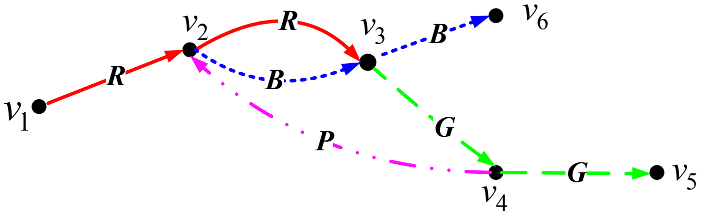
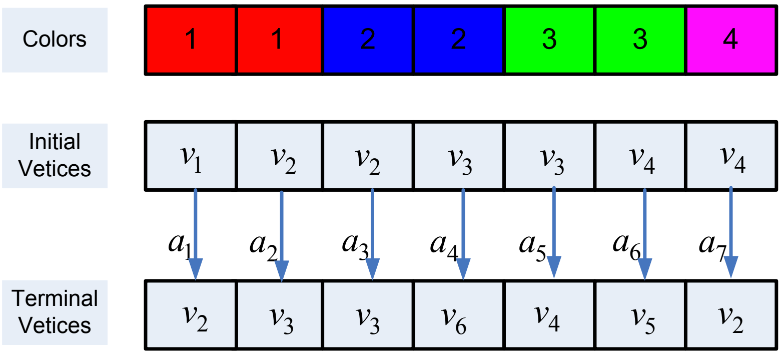


Fig. 1. **A colored multidigraph** G**.**

Example 1 *Fig. 1 shows a colored multidigraph $G = (V, A, N, \psi, c)$. The labels on the arcs of the graph indicate that the corresponding arcs are colored in red, blue, green, and pink, respectively. Assume that the vertex set $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$. According to the Rule of Priority, label all edges to determine the edge-labeled graph.*

First number red 1, blue 2, green 3, and pink 4 so that $N = \{1, 2, 3, 4\}$. The cardinalities of the arc sets A_1, A_2, A_3, and A_4 are 2, 2, 2, and 1, respectively. Then, according to



Colors	1	1	2	2	3	3	4
Initial Vetices	v_1	v_2	v_2	v_3	v_3	v_4	v_4
	a_1	a_2	a_3	a_4	a_5	a_6	a_7
Terminal Vetices	v_2	v_3	v_3	v_6	v_4	v_5	v_2

Fig. 2. **Labeling edges for the graph G.**

the Rule of Priority, the process to label all colored edges is presented in Fig. 2. Obviously, $a_1 = d_1(v_1, v_2); a_2 = d_1(v_2, v_3); a_3 = d_2(v_2, v_3); a_4 = d_2(v_3, v_6); a_5 = d_3(v_3, v_4)$; $a_6 = d_3(v_4, v_5)$; and $a_7 = d_4(v_4, v_2)$. Therefore, the edge labeled graph is expressed as $\langle V, \{A_i, i \in N\}\rangle$, where $A_1 = \{a_1, a_2\}$, $A_2 = \{a_3, a_4\}$, $A_3 = \{a_5, a_6\}$, and $A_4 = \{a_7\}$.

3.2 Extended matrices to searching edge-weighted-colored paths

For a weighted colored multidigraph $G = (V, A, N, \psi, c, w)$, it is obvious from Definitions 2 and 10 that the **adjacency matrix of the line graph** of G is the $l \times l$ matrix LJ with (a, b) entry

$$LJ(a,b) = \begin{cases} 1 & \text{if edges } a \text{ and } b \text{ are consecutive in order } ab \text{ in the graph } G, \\ 0 & \textit{otherwise.} \end{cases}$$

Definition 14 *For a weighted colored multidigraph $G = (V, A, N, \psi, c, w)$, the* ***weighted adjacency matrix*** $LJ^{(W)}$ ***of weighted line digraph*** $L^{(W)}(G)$ *is the $l \times l$ matrix with (a, b) entry*

$$LJ^{(W)}(a,b) = \begin{cases} w_a \cdot w_b & \textit{if edges } a \textit{ and } b \textit{ are consecutive in order } ab \textit{ in the graph } G, \\ 0 & \textit{otherwise.} \end{cases}$$

Now let W be a weight matrix and let $L^{(W)}(G)$ denote the weighted line digraph of G. The following theorem is obtained based on Definition 12, on the weighted in-incidence and out-incidence matrices $B_{in}^{(W)}$ and $B_{out}^{(W)}$, and Definition 14, on the weighted adjacency matrix $LJ^{(W)}$ of the digraph $L^{(W)}(G)$.

Theorem 1 *For a weighted colored multidigraph $G = (V, A, N, \psi, c, w)$, the weighted adjacency matrix $LJ^{(W)}$ of line digraph $L^{(W)}(G)$ satisfies $LJ^{(W)} = -[(B_{in}^{(W)})^T \cdot (B_{out}^{(W)})]$.*

Proof: Let $M = -[(B_{in}^{(W)})^T \cdot (B_{out}^{(W)})]$. Any (k, h) entry of matrix M can be expressed as $M(k, h) = e_k^T \cdot M \cdot e_h = -[(B_{in}^{(W)}) \cdot e_k]^T \cdot [(B_{out}^{(W)}) \cdot e_h]$, where e_k^T denotes the transpose of the k^{th} standard basis vector of the l-dimensional Euclidean space.

The q^{th} nonzero element of the row vector $e_k^T \cdot (B_{in}^{(W)})^T$ is equal to the weight w_k of edge $a_k = d_i(s, q)$ for some $s \in V$. Similarly, the q^{th} nonzero element of the column vector $-(B_{out}^{(W)}) \cdot e_h$ is equal to the weight w_h of edge $a_h = d_j(q, r)$ for some $r \in V$. Hence, $M(k, h) = w_k \cdot w_h \neq 0$ iff a_k and a_h are consecutive from a_k to a_h (See Fig. 3). Then, by Definition 14, $-(B_{in}^{(W)} \cdot (B_{out}^{(W)}) = LJ^{(W)}$. □

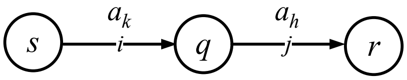


Fig. 3. a_k and a_h are consecutive in order $a_k a_h$.

Let $T_1(B^{(W)}) = -(B_{in}^{(W)})^T \cdot (B_{out}^{(W)}) = LJ^{(W)}$ denote a conversion function. The conversion function, $T_1(B^{(W)})$, maps the weighted incidence matrix $B^{(W)}$ of the graph G to the weighted adjacency matrix $LJ^{(W)}$ of the weighted line digraph of G. It shows that this conversion function transforms the original edge-weighted-colored multidigraph G to a simple vertex-weighted-colored line digraph $L(G)$. Obviously, when $W = I$, $LJ = -(B_{in})^T \cdot (B_{out})$. This matrix captures the adjacency relation between pairs of consecutive edges without considering the color(s) of the consecutive edges. Another conversion function is thus presented next to transform the original problem of searching edge-colored paths in a colored multidigraph to the standard problem of finding paths in a simple digraph without color constraints.

3.3 A conversion function for finding colored paths

Recall that c_i denote the cardinality of the arc set in color i and let E_{c_i} denote a $c_i \times c_i$ matrix with each entry being set to 1 for $i = 1, 2, \cdots, n$. Then, D is defined as the following block diagonal matrix

$$D = \begin{pmatrix} E_{c_1} & 0 & \cdots & 0 \\ 0 & E_{c_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E_{c_n} \end{pmatrix}. \tag{1}$$

It is obvious that this matrix D encodes the color scheme in the graph G, where the dimension of each diagonal block E_{c_i} depends on the number of edges in color i. More specifically, recall that $\varepsilon_i = \sum_{j=1}^{i} c_j$ for $1 \leq i \leq n$. According to the Rule of Priority for labeling edges, for any $a_k \in A$ and $\varepsilon_{i-1} < k \leq \varepsilon_i$, the edge a_k has color i. Hence, for any $a_k, a_h \in A$, if there exists $1 \leq i \leq n$ such that $k, h \in (\varepsilon_{i-1}, \varepsilon_i]$, then edges a_k and a_h have the same color i, and $D(k, h) = 1$. Also, $D(k, h) = 0$ iff edges a_k and a_h have different colors.

Let $LJ^{(W)}$ denote the weighted adjacency matrix of line digraph $L^{(W)}(G)$. To search all edge-weighted-colored paths, the reduced matrix of matrix $LJ^{(W)}$ is defined as follows.

Definition 15 *For a weighted colored multidigraph $G = (V, A, N, \psi, c, w)$, the* ***reduced matrix*** $LJ_r^{(W)}$ ***of matrix*** $LJ^{(W)}$ *is the $l \times l$ matrix with (a, b) entry*

$$LJ_r^{(W)}(a,b) = \begin{cases} w_a \cdot w_b & \text{if edges } a \text{ and } b \text{ are consecutive in order } ab \\ & \quad \text{and have different colors in the graph } G, \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

The conversion function can now be obtained in matrix form by the following theorem.

Theorem 2 *For the weighted colored multidigraph $G = (V, A, N, \psi, c, w)$, let E_l be the $l \times l$ matrix with each entry equal to 1. Then the reduced matrix $LJ_r^{(W)}$ satisfies $LJ_r^{(W)} = LJ^{(W)} \circ (E_l - D)$, where "$\circ$" denotes the Hadamard product.*

Proof: Let $LJ^{(W)}(k, h)$ and $(E_l - D)(k, h)$ denote the (k, h) entries of matrices $LJ^{(W)}$ and $E_l - D$, respectively. Then, $LJ^{(W)}(k, h) \cdot (E_l - D)(k, h) = w_k \cdot w_h \neq 0$ iff $LJ^{(W)}(k, h) = w_k \cdot w_h \neq 0$ and $D(k, h) = 0$. Based on the definitions of matrices $LJ^{(W)}$ and D, $LJ^{(W)}(k, h) \neq 0$ iff edges a_k and a_h are consecutive in order $a_k a_h$. $D(k, h) = 0$ iff edges a_k and a_h have different colors. Obviously, $LJ^{(W)}(k, h) \cdot (E_l - D)(k, h)$ satisfies the statement (2). Therefore, $LJ_r^{(W)} = LJ^{(W)} \circ (E_l - D)$. □

From Theorem 2, $T_2(LJ^{(W)}) = LJ^{(W)} \circ (E_l - D) = LJ_r^{(W)}$. The conversion function, $T_2(LJ^{(W)})$, maps the weighted adjacency matrix $LJ^{(W)}$ of the weighted line digraph of G to its reduced matrix $LJ_r^{(W)}$. It reveals that this conversion function T_2 converts the simple vertex-weighted-colored line digraph $L^{(W)}(G)$ to its reduced subgraph $L_r^{(W)}(G)$, called reduced weighted line digraph, which is a simple digraph with no color constraints.

Theorems 1 and 2 together present a conversion function $F(B^{(W)})$ such that

$$F(B^{(W)}) = [-(B_{in}^{(W)})^T \cdot B_{out}^{(W)}] \circ (E_l - D),$$

where $B_{in}^{(W)} = (B^{(W)} + |B^{(W)}|)/2$ and $B_{out}^{(W)} = (B^{(W)} - |B^{(W)}|)/2$. Therefore, $F(B^{(W)})$ transforms a problem of searching weighted colored paths in an edge-weighted-colored multidigraph to a standard problem of finding paths in a simple digraph with no color constraints. Note that the incident relations between vertices and edges of a graph can uniquely characterize the graph. Therefore, the incidence matrix is treated as the original graph and used for computer implementation.

Example 2 *Fig. 1 shows a colored multidigraph $G = (V, A, N, \psi, c)$. If G is associated with a map $w : A \rightarrow \mathbb{R}_0^+$, then $G = (V, A, N, \psi, c, w)$ is a weighted colored multidigraph. Construct conversion functions to determine the vertex labeled weighted line digraph $L^{(W)}(G)$ and its reduced line digraph $L_r^{(W)}(G)$.*

By Example 1, the colored multidigraph is labeled using the Rule of Priority. It is easy to obtain incident relations between vertices and edges from the graph. Thus, matrices $B_{in}^{(W)}$ and $B_{out}^{(W)}$ are constructed by Definition 12 as follows:

$$B_{in}^{(W)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_1 & 0 & 0 & 0 & 0 & 0 & w_7 \\ 0 & w_2 & w_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_6 & 0 \\ 0 & 0 & 0 & w_4 & 0 & 0 & 0 \end{pmatrix}, \textit{ and } B_{out}^{(W)} = \begin{pmatrix} -w_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -w_2 & -w_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -w_4 & -w_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -w_6 & -w_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

From Theorems 1 and 2, we obtain that

$$T_1(B^{(W)}) = \begin{pmatrix} 0 & w_1w_2 & w_1w_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_2w_4 & w_2w_5 & 0 & 0 \\ 0 & 0 & 0 & w_3w_4 & w_3w_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_5w_6 & w_5w_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_7w_2 & w_7w_3 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$T_2(LJ^{(W)}) = \begin{pmatrix} 0 & 0 & w_1w_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_2w_4 & w_2w_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_3w_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w_5w_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_7w_2 & w_7w_3 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The weight matrix designed here is convenient, since edge-weighted (0 or 1) can be used to flexibly control any move between any two vertices in G. For instance, if $w_4 = 0$, then the original graph will be reduced to a new graph with no edge a_4. If $W = I$, then the conversion function T_1 transforms the edge-labeled multidigraph portrayed in Fig. 4 (1) to the vertex-labeled line digraph $L(G)$ shown in Fig. 4 (2). Then, the reduced line digraph $L_r(G)$ presented in Fig. 4 (3) for finding colored paths is obtained by using the conversion function T_2. The conversion process is illustrated in Fig. 4.

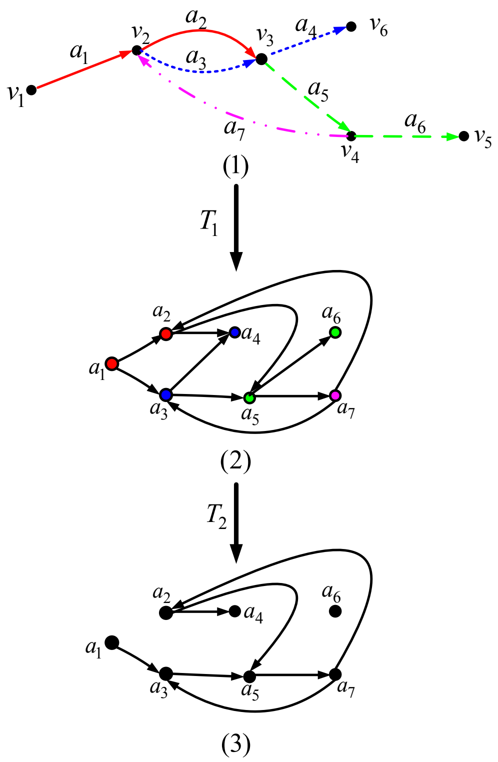


Fig. 4. Transformed graphs of G.

3.4 Computer implementation

Searching colored paths aims to find all edge-colored paths in a given colored multidigraph. Although the shortest path problem in general graph classes has been extensively investigated, searching colored paths in weighted colored multidigraphs is still a novel topic.

Let $A_S = \{a \in A : B_{out}^{(W)}(s,a) \neq 0\}$ and $A_E = \{b \in A : B_{in}^{(W)}(q,b) \neq 0\}$. Here, matrices W, $B_{out}^{(W)}$, and $B_{in}^{(W)}$ have been introduced by Definitions 9 and 12. A_S is the set of arcs starting from vertex s and A_E is the arc set ending at vertex q. The matrix $LJ_r^{(W)}$ provided by Theorem 2 is used to search the edge-weighted-colored paths between any two arcs in a weighted colored multidigraph. Let $P^{(W)}(a,b)$ for $a, b \in A$ denote the edge-weighted-colored paths between two edges a and b. The edge-weighted-colored paths between two vertices s and q for $s, q \in V$ are expressed as $P^{(W)}(s,q)$. A vertex-by-vertex path between any two vertices can be obtained by tracing arc-by-arc paths between two appropriate arcs. Specifically, the paths between s and q can be expressed as $P^{(W)}(s,q) = \bigcup_{a \in A_S,\ b \in A_E} P^{(W)}(a,b)$.

Pseudo code of the proposed algorithm for finding colored paths

```
Step 0: Input the starting arc set $A_S$, the ending arc set $A_E$, and the reduced matrix $LJ_r^{(W)}$.
Step 1: For each arc $a_s \in A_S$ and each arc $a_e \in A_E$, set $a_s$ as the starting arc and $a_e$ as the ending arc. For each pair of $a_s$ and $a_e$, repeat the steps from Step 2 to Step 5.
Step 2: Put $a_s$ into Path-Recorder as the last arc $a_l(1)$ of the first path.
Step 3: In Path-Recorder, for each path $i$, e.g., $P^{(W)}(i)$, check its last arc $a_l(i)$.
    Obtain all the new arcs starting from $a_l(i)$ based on matrix $LJ_r^{(W)}$.
    Case 1: If there is no arc starting from $a_l(i)$, path $P(i)$ ends. Eliminate $P^{(W)}(i)$ from Path-Recorder;
    Case 2: If a new arc has appeared in the path, which means that the path forms a cycle, do not record the new path. If all the new arcs have appeared, eliminate $P^{(W)}(i)$ from Path-Recorder;
    Case 3: If the new arc is the end arc $a_e$, add $a_e$ to the path $P^{(W)}(i)$ to form a new path. Reserve the path into Path-Recorder and set an *end-mark* at the end of the path.
            If all the new arcs are $a_e$, eliminate $P^{(W)}(i)$ from Path-Recorder;
            Otherwise: Add each new arc to path $P^{(W)}(i)$, respectively, to form several new paths.
            Reserve these paths into Path-Recorder, and eliminate the original path $P^{(W)}(i)$ from Path-Recorder.
Step 4: Repeat Step 3 until all the paths in Path-Recorder have the *end-mark* at the end.
Step 5: Output Path-Recorder, which records all paths starting from $a_s$ and ending at $a_e$.
```

The proposed algebraic method is convenient for computer implementation. A pseudo code for the proposed algorithm is presented as follows.

Because the algebraic expressions are explicitly given, the proposed method facilitates the development of improved algorithms to search colored paths and is easy to adapt to new path searching problems. For instance, a transportation network problem of finding the shortest path with specific constraints can be solved by using the conversion function $F(B^{(W)}) = [-(B_{in}^{(W)})^T \cdot B_{out}^{(W)}] \circ M$, where $B^{(W)}$ denotes the original network and matrix M is designed to capture constraint requirements, to transform the original problem to a general shortest path searching problem without the constraints.

Note that in this paper all arcs are distinct on a path but the restriction that all nodes be distinct on a path is relaxed.

The process that converts an edge-colored multidigraph to a simple digraph with no color constraints is presented in Fig. 5.

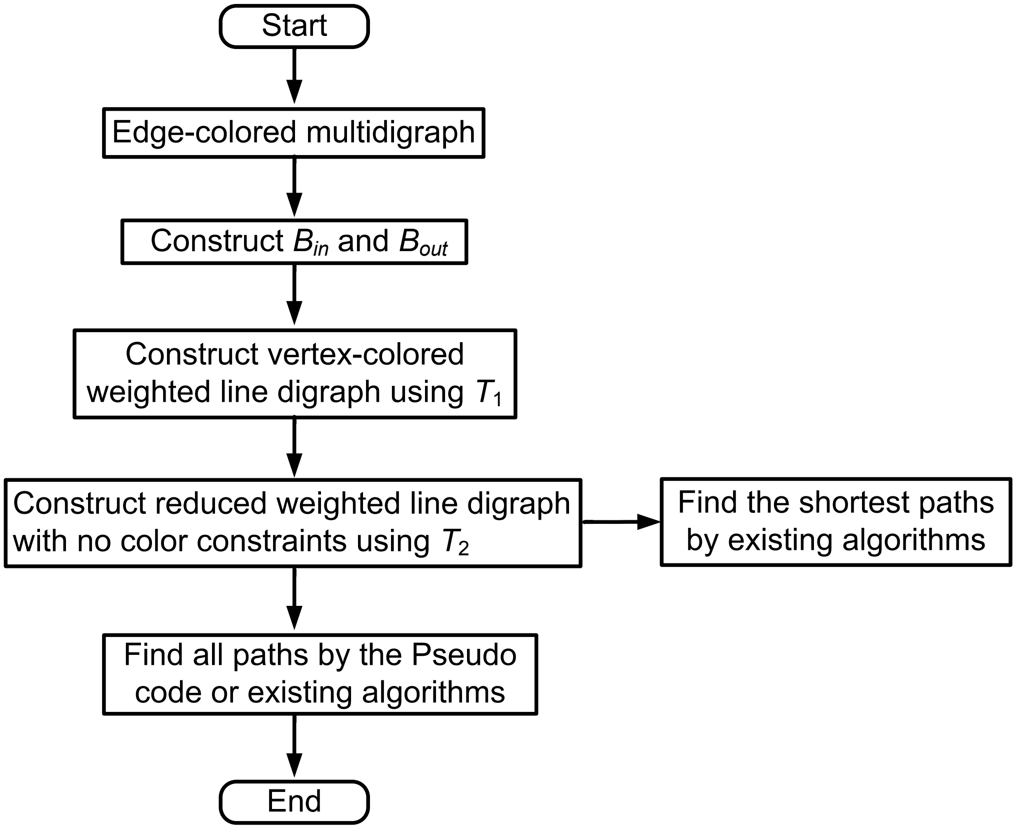

Start
Edge-colored multidigraph
Construct B_{in} and B_{out}
Construct vertex-colored weighted line digraph using T_1
Construct reduced weighted line digraph with no color constraints using T_2
Find the shortest paths by existing algorithms
Find all paths by the Pseudo code or existing algorithms
End

Fig. 5. **The process of finding all colored paths or the shortest colored path**

4 An application: status quo analysis in the graph model for conflict resolution

An application is developed to illustrate how to search colored paths in a weighted colored multidigraph in the context of conflict resolution.

4.1 The graph model for conflict resolution

Definition 16 *A* graph model *is a structure*

$$G\langle N, S, \{\succeq_i, A_i, i \in N\}\rangle$$

where

- N *is a non-empty finite set, called the set of DMs.*
- S *is a non-empty finite set, called the set of states.*
- *For each DM $i \in N$, $\succeq_i$ is a reflexive, transitive, and complete binary relation on S, called i's weak preference.*
- *For each DM i, $A_i \subseteq S \times S$ is DM i's oriented arcs, representing* unilateral moves *by DM i, and $G_i = (S, A_i)$ is i's directed graph.*

Similarly, by the proposed Rule of Priority, the oriented arcs in the graph model are labeled according to the DM order; within each DM, according to the sequence of initial states; and within each DM and initial state, according to the sequence of terminal states.

Two fundamental steps are involved in analyzing a graph model, stability analysis and post-stability (or follow-up) analysis. In stability analysis, each state is examined to determine whether it is stable for each DM (individual stability), and whether it is stable for all DMs (an equilibrium) under appropriate stability definitions (solution concepts) [7]. In the graph model approach, a conflict is conceived to start from the status quo and then pass from state to state according to moves and countermoves controlled by individual DMs, eventually terminate at some state from which no DM is willing to unilaterally move away. As a follow-up analysis, status quo analysis is to determine whether a particular equilibrium is reachable from the status quo and, if so, how to reach it. Thus, in contrast to stability analysis, which identifies states that would be stable if attained, status quo analysis provides a dynamic and forward-looking perspective, identifying states that are attainable, and describing how to reach them [20, 21].

Obviously, DMs' preference information plays a crucial role in any decision analysis. In the original graph model, only a relative preference relation $\succ$ and an indifference relation $\sim$ are available to represent a particular DM's simple preference for one state over another [7]. Furthermore, a preference framework called "strength of preference" that includes two new binary relations, "greatly preferred $\gg$" and "mildly preferred $>$", and the indifference relation was developed by Hamouda et al. [13]. In reality, it is often a challenge to obtain accurate preference information in many situations. Moreover, as [9, 10] discussed, conflicts among the attributes of alternatives can cause preference uncertainty. To incorporate preference uncertainty into the graph model methodology, [19] proposes a new preference structure, in which DMs' preferences are expressed by a triplet of binary relations $\{\succ_i, \sim_i, U_i\}$ on S, where $\{s \succ_i q\}$ indicates that DM i prefers s to q, $s \sim_i q$ means that DM i is indifferent between s and q (or equally prefers s and q), and U_i stands for that DM i's uncertainty about its relative preference between s and q, i.e., sU_iq represents that DM i may prefer state s to q, may prefer q to s, or may be indifferent between s and q.

DM i's *reachable list* from $s \in S$ is the set $R_i(s) = \{q \in S : (s, q) \in A_i\}$, states to which DM i can unilaterally move in one step from state s. The members of $R_i(s)$ are DM i's *unilateral moves (UMs)* from $s \in S$. Similarly, the sets $R_i^+(s) = \{q \in S : (s, q) \in A_i \textit{ and } q \succ_i s\}$ and $R_i^U(s) = \{q \in S : (s, q) \in A_i \textit{ and } qU_is\}$ contain DM i's *unilateral improvements (UIs)* [7] and *unilateral uncertain moves (UUMs)* [22] from state s, respectively. Note that notation $UIUUMs$ denotes unilateral improvements or unilateral uncertain moves.

4.2 Weight matrix representation of preference information

The proposed weight matrix in Section 2 can be used to represent various preference structures. If an edge $a_k = d_i(u, v)$ for $u, v \in S$ and $i \in N$, then the weight matrix with (k, k) entry is defined by

(i) for simple preference,

$$w_{a_k} = \begin{cases} P_w & \text{if } v \succ_i u, \\ E_w & \text{if } u \sim_i v, \\ N_w & \text{if } u \succ_i v, \end{cases} \tag{3}$$

(ii) for preference with uncertainty,

$$w_{a_k} = \begin{cases} P_w & \text{if } v \succ_i u, \\ N_w & \text{if } u \succ_i v, \\ E_w & \text{if } u \sim_i v, \\ U_w & \text{if } uU_iv. \end{cases} \tag{4}$$

Obviously, the preference structure with uncertainty presented by statement (4) expands the simple preference expressed by (3). Therefore, the algebraic approach developed in this paper for analyzing conflict evolution with preference uncertainty includes the main results in [28] as a special case.

By appropriately restricting the element values in a weight matrix, one can trace UMs, UIs, and UIUUMs in a graph model with preference uncertainty.

Definition 17 *Let W denote an $l \times l$ weight matrix. Then*

- *when $P_w = N_w = E_w = U_w = 1$, the weight matrix W is called the UM weight matrix W_{UM};*
- *when $P_w = 1$ and $N_w = E_w = U_w = 0$, the weight matrix W is called the UI weight matrix W_{UI};*
- *when $P_w = U_w = 1$ and $N_w = E_w = 0$, the weight matrix W is called the $UIUUM$ weight matrix W_{UIUUM}.*

Note that if the state set S is treated as a vertex set and DM i's oriented arcs are coded in color i, then a graph model of a conflict is equivalent to a colored multidigraph with appropriate preference relations. By the above discussions, the weight matrix is convenient and flexible to represent preference information in the graph model. Therefore, the graph model is converted to a weighted colored multidigraph. It is natural to use the results of Graph Theory to assist in analyzing of a graph model. Hence, we will hereafter use the same notation as Section 3 to represent a graph model for conflict.

A fundamental problem of status quo analysis can thus be treated as searching all paths from a given initial state to a desirable state within the edge-weighted-colored multidigraph, G. Moreover, in the graph model, a legal path cannot include any DM moving twice in succession. Therefore tracing conflict evolution requires searching all paths in the colored multidigraph that start from the status quo state to some equilibrium and do not contain consecutive arcs in the same color. The existing approaches introduced by Li *et al.* [20–22] provide a limited picture of conflict evolution, and the pseudo-codes have not yet been implemented into a practical decision support system. The existing matrix approach [28] used in status quo analysis for simple preference is based on the adjacency matrix, which is able to reveal state-by-state paths and, hence, cannot handle the case when multiple edges in distinct colors exist between two states. On the other hand, the proposed algebraic approach in this paper is specifically designed to tackle multidigraph and, hence is more capable of and efficient in tracking all aspects of conflict evolution. Let $P^{(W)}(u,v)$ denote the edge-weighted-colored paths between two vertices u and v in a weighted colored multidigraph. Obviously,

(1) if $W = W_{UM}$, the $P^{(W)}(u,v)$ gives all colored paths from u to v where all UMs are allowed, hereafter, denoted by $P_{UM}(u,v)$;
(2) if $W = W_{UI}$, the $P^{(W)}(u,v)$ gives all colored paths from u to v where only UIs are allowed, hereafter, denoted by $P_{UI}(u,v)$;
(3) if $W = W_{UIUUM}$, the $P^{(W)}(u,v)$ gives all colored paths from u to v where only UIUUMs are allowed, hereafter, denoted by $P_{UIUUM}(u,v)$.

4.3 Status quo analysis of the Gisborne conflict

In this subsection, the proposed matrix method is applied to a case study — Status quo analysis of the Gisborne conflict. Lake Gisborne is located near the south coast of a Canadian Atlantic province of Newfoundland and Labrador. In June 1995, a local division of the McCurdy Group of Companies, Canada Wet Incorporated, proposed a project to export bulk water from Lake Gisborne to foreign market. On December 5, 1996, this project was registered with the government of Newfoundland and Labrador. At the time of registration, no policy existed on water export in bulk. However, this proposal immediately aroused considerable opposition from a wide variety of lobby groups. In addition to unpredictable harmful impacts on local environment and First Nations culture, a critical issue is its potential implication of making water a tradeable "commodity" that is thus subject to the rules of WTO (World Trade Organization) and NAFTA (North American Free Trade Agreement). Therefore, if the Lake Gisborne bulk water export project was successfully executed, the water policy in Canada might have to undergo a significant shift as any firm would be able to follow the suit. As such, the Federal Government of Canada sided with the opposing groups by introducing a policy to forbid bulk water export from major drainage basins in Canada. The mounting pressure eventually forced the government of Newfoundland and Labrador to introduce a new bill to ban bulk water export from Newfoundland and Labrador, which effectively terminated the Gisborne water export project. (See details in [8]).

Nevertheless, several support groups remain interested in the project, and the provincial government might restart the project at an appropriate time in the future due to its urgent need for cash. This prospect introduces uncertainty into the preferences of the provincial government for the Gisborne conflict model. This conflict is modeled using three DMs: DM 1, **Federal (Fe)**; DM 2, **Provincial (Pr)**; and DM 3, **Support (Su)**; and a total of three options, as shown in Table 1. The following is a summary of the three DMs and their options [19]:

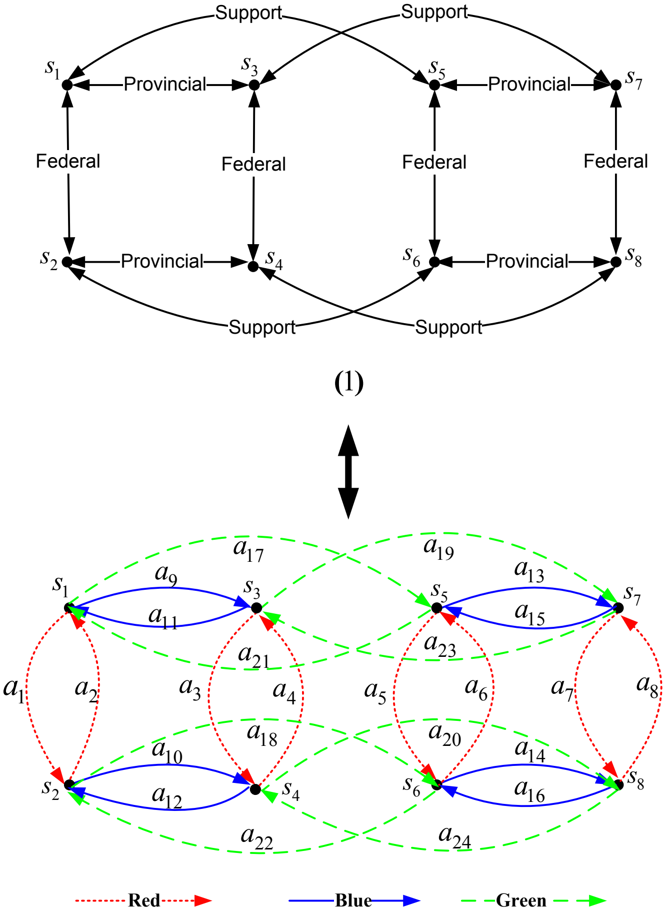
- Federal government of Canada (**Federal**): its option is to continue a Canada-wide accord on the prohibition of bulk water export **(Continue)**,
- Provincial government of Newfoundland and Labrador (**Provincial**): its option is to lift the ban on bulk water export **(Lift)**, and
- Support groups (**Support**): its option is to appeal for continuing the Gisborne project **(Appeal)**.

Table 1
Options and feasible states of the Gisborne conflict

Federal								
1. Continue	N	Y	N	Y	N	Y	N	Y
Provincial								
2. Lift	N	N	Y	Y	N	N	Y	Y
Support								
3. Appeal	N	N	N	N	Y	Y	Y	Y
State number	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8

In the Lake Gisborne conflict model, the three options together determine 8 possible states as listed in Table 1, where a “Y” indicates that an option is selected by the DM controlling it and an “N” means that the option is not chosen. The graph model of the Lake Gisborne conflict is shown in Fig. 6 (1), where the labels on the arcs identify the DMs who control the relevant moves. If DM i’s oriented arcs are coded in color i, then, according to the *Rule of Priority*, Fig. 6 (1) is converted to an edge labeled multidigraph as shown in Fig. 6 (2).

Preference information over the states are given in Table 2, where $\succ$ represents the strict preference relation and is transitive. As shown in Table 2, DM **Federal**’s and DM **Support**’s preference information is modeled to be known completely without any uncertainty, but DM **Provincial**’s preference, on the other hand, is assumed to be partially known as exhibited by its vacillation in the course of this conflict. What is known is that it prefers state s_3 to state s_7, state s_4 to state s_8, state s_1 to state s_5, and



(1)

(2)

Fig. 6. Graph model for the Gisborne conflict.

Table 2

Preference information for the Gisborne conflict

Colors	DMs	Certain preferences
Red	Federal	$s_2 \succ s_6 \succ s_4 \succ s_8 \succ s_1 \succ s_5 \succ s_3 \succ s_7$
Blue	Provincial	$s_3 \succ s_7$, $s_4 \succ s_8$, $s_1 \succ s_5$, $s_2 \succ s_6$, only
Green	Support	$s_3 \succ s_4 \succ s_7 \succ s_8 \succ s_5 \succ s_6 \succ s_1 \succ s_2$

state s_2 to state s_6, but the relative preference across these four groups is uncertain. According to the rule (4), the preference information in Table 2 is applied to weighted edges as given in Table 3.

Table 3

Weights of edges for the labeled graph for the Gisborne conflict

Arc number	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}
Assigned weight	P_w	N_w	P_w	N_w	P_w	N_w	P_w	N_w	U_w	U_w	U_w	U_w	U_w	U_w	U_w	U_w	P_w	P_w	N_w	N_w	N_w	N_w	P_w	P_w

Based on the extended preference structure with uncertainty, Li et al. [19] redefine Nash stability, general metarationality, symmetric metarationality, and sequential stability for graph models with preference uncertainty. According to whether uncertain preferences are deemed as sufficient incentives to motivate the focal DM leaving the current state and credible sanctions to deter the focal DM from doing so, the aforesaid four types of stability are redefined in four different manners and indexed as a, b, c, and d. These four extensions are conceived to depict DMs with distinct risk profiles in face of uncertainty. Li et al. [19] identify states s_4, s_6 and s_8 as equilibria under extension b and d for the Gisborne conflict. Note that for the stability definitions under extensions b and d, the focal DM is conservative in deciding whether to move away from the current state, since it would only move to preferred states (UIs). For details, readers are referred to [19].

In parallel to extensions b and d that predict the three equilibria s_4, s_6, and s_8, we examine the evolution paths P_{UI} (allowing UIs only) from a status quo to the three equilibria. Let the weight matrix W_{UI} be defined according to the information provided in Table 3 in which $N_w = U_w = 0$ and $P_w = 1$. From Theorems 1 and 2, $F(B^{(W_{UI})}) = [-(B_{in}^{(W_{UI})})^T \cdot (B_{out}^{(W_{UI})})] \circ (E_l - D)$ denotes a conversion function that transforms the labeled multidigraph Fig. 7 (1) to the reduced line digraph Fig. 7 (2) that is a simple digraph with no color constraints. Therefore, finding colored paths in Fig. 7 (1) is equivalent to searching paths in Fig. 7 (2). If the status quo is s_1, it is obvious that the equilibria s_4 and s_8 can not be reached by UIs and the equilibrium s_6 is the only equilibrium that is attainable from the status quo. Specifically, the evolutionary paths $P_{UI}(s_1, s_6)$ can be described below:

$$a_1 \longrightarrow a_{18} \Longleftrightarrow s_1 \longrightarrow s_2 \longrightarrow s_6$$
$$a_{17} \longrightarrow a_5 \Longleftrightarrow s_1 \longrightarrow s_5 \longrightarrow s_6$$

But if UIUUMs are allowed, equilibrium s_8 is attainable from the status quo s_1. The weight matrix W_{UIUUM} is defined by setting $N_w = 0$ and $P_w = U_w = 1$. Using conversion matrix $B^{(W_{UIUUM})}$, the labeled graph in Fig. 6 (2) is reduced to Fig. 8 (1) that illustrates the evolution of the graph model for the Gisborne conflict with allowing UIUUMs only. By the conversion function $F(\cdot)$, the colored multidigraph in Fig. 8 (1) is transformed to the reduced line digraph in Fig. 8 (2). Searching colored paths $P_{UIUUM}(s_1, s_8)$ in Fig.

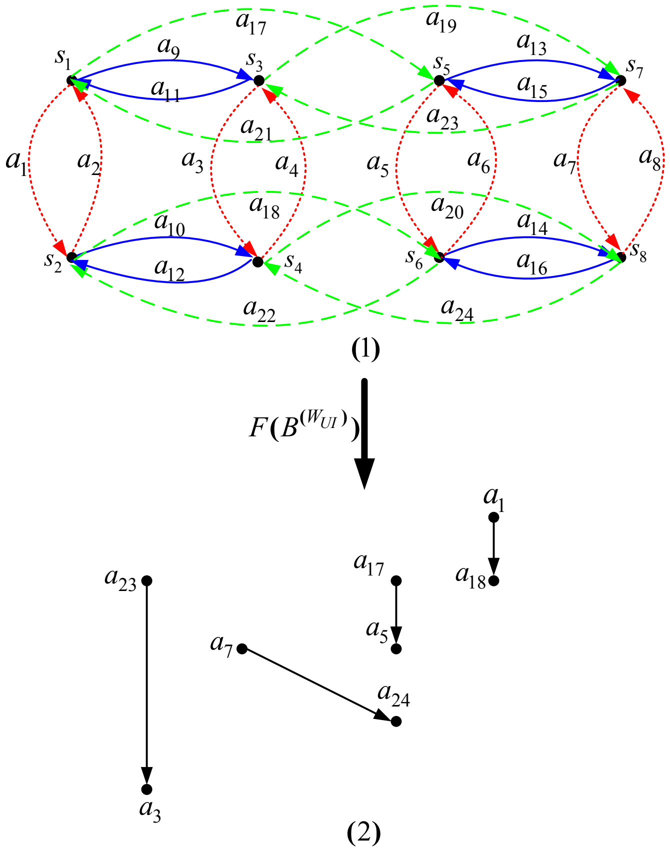


Fig. 7. The conversion graphs for finding the evolutionary UI paths for the Gisborne conflict.

8 (1) is equivalent to finding paths $P_{UIUUM}(a_1, a_{14})$, $P_{UIUUM}(a_1, a_7)$, $P_{UIUUM}(a_9, a_{14})$, $P_{UIUUM}(a_9, a_7)$, $P_{UIUUM}(a_{17}, a_{14})$, and $P_{UIUUM}(a_{17}, a_7)$ in Fig. 8 (2). Therefore, the evolution of the Gisborne conflict with $UIUUM$s from status quo state s_1 to equilibrium s_8 is illustrated as follows:

$a_1 \longrightarrow a_{18} \longrightarrow a_{14}$

$a_9 \longrightarrow a_3 \longrightarrow a_{12} \longrightarrow a_{18} \longrightarrow a_{14}$

$a_{17} \longrightarrow a_5 \longrightarrow a_{14}$

$a_{17} \longrightarrow a_{13} \longrightarrow a_{23} \longrightarrow a_3 \longrightarrow a_{12} \longrightarrow a_{18} \longrightarrow a_{14}$

$a_{17} \longrightarrow a_{13} \longrightarrow a_{23} \longrightarrow a_{11} \longrightarrow a_1 \longrightarrow a_{18} \longrightarrow a_{14}$

$a_{17} \longrightarrow a_{13} \longrightarrow a_7$

After transforming a colored multidigraph to a simple digraph under conversion functions, existing algorithms such as those reported in [23] and [26] can be used to find all paths or search the shortest path.

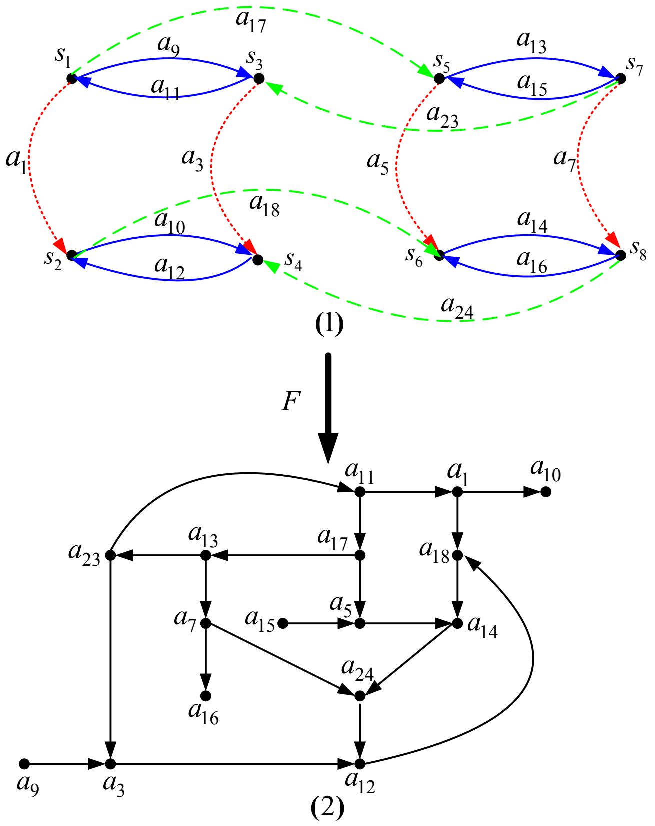


Fig. 8. The conversion graphs for finding the evolutionary $UIUUM$ paths for the Gisborne conflict.

5 Conclusions

This paper proposes a novel algebraic approach to searching colored paths in a weighted colored multidigraph. Specifically, according to a *Rule of Priority*, the weighted colored multidigraph is converted to an edge-labeled multidigraph. With the unique labeling of all colored edges, a conversion function to transform a weighted colored multidigraph to a simple digraph is developed, whereby the problem of searching edge-weighted-colored paths in a weighted colored multidigraph can be achieved by finding paths in a simple digraph with no color constraints. Another contribution of this approach is the weight matrix that is designed to reflect some attribute of edges in a flexible and efficient manner.

This proposed approach is then applied to the status quo analysis in the graph model for conflict resolution to demonstrate how it may be conveniently adapted for practical use. In the graph model, an important restriction is that consecutive moves are not allowed for any DM along any path and, hence, a graph model can be treated as an edge-weighted-colored multidigraph and preference information can be represented by an appropriately designed weight matrix.

The proposed method provides an explicit algebraic expression that facilitates the development of improved algorithms to search all colored paths. The explicit algebraic representation derived in this paper may be adapted for new applications such as transportation networks and status quo analysis with preference strength [13] and hybrid preferences.

References

[1] D. Angluin and L. Valiant, Probabilistic algorithms for Hamiltonian circuits and matchings, J. Comput. Sys. Sci. 18 (1979), 155-190.

[2] F. Buckley and F. Harary, Distance in Graphs, Addison-Wesley, Reading, MA, 1990.

[3] M. Cheng and L. Yin, Transmission scheduling in sensor networks via directed edge coloring, in proceedings of the ICC 2007, Vol. 24-28, 3710-3715.

[4] C. Davies and P. Lingras, Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks, European Journal of Operational Research 144 (2003) 27-38.

[5] R. Dieste, Graph Theory, New York: Springer, 1997.

[6] E. W. Dijkstra, A note on two problems in connexion with graph, In Numerische Mathematik 1 (1959), 269-271.

[7] L. Fang, K.W. Hipel, and D.M. Kilgour, Interactive Decision Making: The Graph Model for Conflict Resolution, New York: Wiley, 1993.

[8] L. Fang, K.W. Hipel, and L. Wang, Gisborne water export conflict study, Proceedings of 3rd International Conference on Water Resources Environment Research 1 (2002), 432-436.

[9] G. W. Fischer, J. Jia, and M. F. Luce, Attribute conflict and preference uncertainty: The randMAU model, Management Science 46 (5) (2000), 669-684.

[10] G. W. Fischer, M. F. Luce, and J. Jia, Attribute conflict and preference uncertainty: Effects on judgment time and error, Management Science 46 (1) (2000), 88-103.

[11] C. Godsil and G. Royle, Algebraic Graph Theory, New York: Springer, 2001.

[12] M. Gondran and M. Minoux, Graphs and Algorithms, New York: Wiley, 1979.

[13] L. Hamouda, D.M. Kilgour, and K.W. Hipel, Strength of preference in graph models for multiple-decision-maker conflicts, Applied Mathematics and Computation 179 (2006) 314-327.

[14] A. J. Hoffman and B. Schiebe, The edge versus path incidence matrix of series-parallel graphs and greedy packing, Discrete Applied Mathematics 113 (2001), 275-284.

[15] T. Inohara, Signed graphs with negative self evaluation and clusterability of graphs, Applied Mathematics and Computation 158 (2004), 477-487.

[16] D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, Journal of the ACM 24(1) (1997), 1-13.

[17] J. K. Levy and K. Taji, Group decision support for hazards planning and emergency management: A Group Analytic Network Process (GANP) approach, Mathematical and Computer Modelling 46 (2007), 906-917.

[18] D.M. Kilgour, K.W. Hipel, and L. Fang, The graph model for conflicts, Automatica 23 (1987), 41-55.

[19] K.W. Li, K.W. Hipel, D.M. Kilgour, and L. Fang, Preference uncertainty in the graph model for conflict resolution, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 34 (2004), 507-520.

[20] K.W. Li, D.M. Kilgour, and K.W. Hipel, Status quo analysis of the Flathead river conflict, Water Resources Research 40 (2004), W05S03.

[21] K.W. Li, D.M. Kilgour, and K.W. Hipel, Status quo analysis in the graph model for conflict resolution, Journal of the Operational Research Society 56 (2005), 699-707.

[22] K.W. Li, K.W. Hipel, D.M. Kilgour, and D.J. Noakes, Integrating uncertain preference into status quo analysis with applications to an environmental conflict, Group Decision and Negotiation 14 (2005), 461-479.

[23] M. Migliore, V. Martorana, and F. Sciortino, An algorithm to find all paths between two nodes in a graph, Journal of Computational Physics 87 (1990), 231-236.

[24] F. Rubin, A search procedure for Hamilton paths and circuits, J. ACM 21 (1974), 576-580.

[25] A. K. Shiny and A. K. Pujari, Computation of prime implicants using matrix and paths, J. Logic Computat. 8(2) (1998), 135-145.

[26] Y. Xia and J. Wang, A discrete-time recurrent neural network for shortest-path routing, IEEE Transactions on Automatic Control, 45(11) (2000), 2129-2134.

[27] H. Xu, K.W. Hipel, and D.M. Kilgour, Matrix representation of solution concepts in multiple-decision-Maker graph models, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 39 (2009), 96-108.

[28] H. Xu, K. W. Li, K. W. Hipel, and D. M. Kilgour, A matrix approach to status quo analysis in the graph model for conflict resolution, accepted by Applied Mathematics and Computation, 2009.

*** Response to Reviewers**

Response to the referee's comments
on the paper AMC-D-09-00695
"**A Matrix-based Approach to Searching Colored Paths in a Weighted Colored Multidigraph**"
by
Haiyan Xu, Kevin W. Li, D. Marc Kilgour, and Keith W. Hipel
Submitted for publication in the
Applied Mathematics and Computation
April 29, 2009

We would like to thank you for carefully reviewing our paper and providing useful comments to improve it. Our revisions, written in response to your comments, are explained below.

- *In Definition 7 on page 5, the authors introduce the concept of ”weighted edge-colored paths.” On the other hand, the authors use the phrase ”edge-weighted-colored paths” in the same page (in the last line in the second paragraph from Definition 9). Also, the title of Section 3.2 on page 8 says ”... edge-weighted-colored paths.” ”weighted edge-colored paths” and ”edge-weighted-colored paths” are the same? If so, use the same phrase. If not, give a definition of ”edge-weighted-colored paths.”*

  Following your thoughtful suggestion, all phrases "weighted edge-colored paths" in the old version have been changed to "edge-weighted-colored paths" in the revised paper. Now the terminology has been used consistently throughout the manuscript.

- *The sentence of Theorem 7 looks ambiguous, because figuring out the assumption of this proposition is difficult. Read again it, and revise it approppreately.*

  As this paper does not have Theorem 7, it is our understanding that the referee actually means the two theorems on pages 8 and 10. Therefore, appropriate modifications have been made for these two theorems to make them more concise and improve their readability. For instance, to simplify the statement of Theorem 1, a new paragraph is added before it is introduced on page 8. To make the statement of Theorem 2 smoother, Definition 15 is added on page 10.