Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

A Mathematical Optimization Model for Solving Minimum Ordering Problem with Constraint Analysis and some Generalizations

Samira Rezaei*[1] and Amin Ghodousian†[2]

[1]Department of Algorithms and Computation, University of Tehran.
[2]University of Tehran, College of Engineering, Faculty of Engineering Science.

ABSTRACT

In this paper, a mathematical method is proposed to formulate a generalized ordering problem. This model is formed as a linear optimization model in which some variables are binary. The constraints of the problem have been analyzed with the emphasis on the assessment of their importance in the formulation. On the one hand, these constraints enforce conditions on an arbitrary subgraph and then give sufficient conditions for feasibility, on the other hand, they provide a natural way to generalize the applied aspects of the model without increasing the number of the binary variables.

ARTICLE INFO

Article history:
Received 30, May 2015
Received in revised form 18, january 2016
Accepted 30 February 2016
Available on line 01, April 2016

Keyword: linear programming, integer programming, minimum ordering.

AMS subject Classification: 65K05, 90C35.

1 Introduction

linear ordering problem (LOP) arises in a variety of practical applications and generalizes prerequisite problems such as the Minimum Linear Arrangement, Min Sum Set Cover, Minimum Latency Set Cover, and Multiple Intents Ranking.

*Email: samirarezaei@ut.ac.ir
†Corresponding author: A. Ghodousian. Email: a.ghodousian@ut.ac.ir

The objective of the class of LOPs is to minimize different functionals by finding a suitable permutation of the graph vertices commonly used for general graphs or matrices [1,18]. Due to its interest in practice, it has received considerable attention and consequently a variety of approaches have been proposed to the solution of these NP-Hard problems[19]. Moreover, Exact algorithms presented differ generally from each other confronting with binary variables constraints. In [28], an algorithm was proposed using a LP-relaxation of the LOP for the lower bound. A branch and cut algorithm have been studied in [24] and authors in [34] investigated a combined interior point/cutting plane algorithm. Efforts have been made [2, 21-23] to obtain improvements on the exact methods with some success, where considerably large instances have been solved to optimality for example the work done on the Traveling Salesman Problem. The idea is to relax the integer constraints in the formulation of the problem and solve it as a continuous problem. Although it is not possible to think to obtain all facets associated with an NP-hard problem, the relaxation was solved [35] by using the known facets. In doing so, if the optimal solution is not found, a branch and bound procedure can be used to finish up the problem. State of the art exact algorithms can solve fairly large instances from specific instance classes up to a few hundred columns and rows, although they fail on much smaller instances of other classes.

The LOP was also tackled by a number of heuristic algorithms including constructive algorithms such as BeckerâŁ™s greedy algorithm [6], local search algorithms an example of which is the CK heuristic [9], metaheuristic approaches such as elite tabu search and scatter search [7,8,31] and iterated local search (ILS) algorithms [12,41]. Many heuristic algorithms were developed in order to achieve near optimal solution. Among the most successful are spectral sequencing (SS) [13], optimally oriented decomposition tree [4], multilevel based [30,26] and simulated annealing [36]. SS approach, one of the most popular methods, consists of ordering the graph vertices according to the sorted coordinates of the second eigenvector of the graph Laplacian. The heuristic argumentation of SS is based on the fact that the continuous version of the minimum 2-sum problem can be solved to the optimum by this method [10]. In practice, for the (discrete) minimum 2-sum, it was shown [39] that the direct application of SS does not achieve satisfactory results, while the lower bounds based on SS are very far from the best known ordering costs [39]. In [27,43,13] better results were obtained using different approximated SS (by calculating the second eigenvector less precisely).

Furthermore, many approximating approaches have been suggested [37, 14, 29, 33, 32, 25, 38, 3, 34, 17, 40, 16] to find the solution to the problems (or a lower or upper bound for an optimal solution). Seymour [42] was the first to propose a directed graph decomposition divide-and-conquer approach for the minimum feedback arc set problem. Even et al. [15] extended the recursive decomposition technique used by Seymour for minimum containing interval graph problems.

Finally, there has been recent interest in studying minimization problems with submodular cost functions. However, almost all of these problems turn out to be quite intractable and have large polynomial lower bounds. There are some exceptions such as the submodular vertex cover problem [28,20] and the submodular multiway partition [11]. However, many interesting approximating algorithms and improvements [3,17,40] were studied in min sum set cover, which is a special instance of LOP with a supermodular cost function. Authors in [3] provided a small improvement (with a rather technically involved analysis) by obtaining a 1.99995 factor approximation, and raised the question of further improving the bound [3]. Then a theorem can provide a substantial improvement, with an alternative rounding and a simpler analysis, in giving a 1.79 factor approximation.

This paper is an attempt to propose a generalized model of the LOPs can be considered as second-order. Second-order ordering problem is a problem, we assume, in which the ordering induced by an optimal path is restricted to the number of pre-assumed arrangement constraints on nodes. Such conditions are commonly considered by some management requirements as new conditions on a network. More precisely, we would like to find a spanning path in a network, that optimizes an objective function (calssical ordering problem) provided that some special criterion on nodes or/and edges are satisfied; for example, prespecified positions of some special nodes on the path (some nodes are assumed to have fixed known positions in an optimal ordering of nodes), the priority of some nodeâŁ™s positions (some nodes have to be visited before or after other nodes), admissible travel-time interval in the path in where some nodes have to be visited.

In section 2, first we characterize a mathematical modeling of LOP and analyze the significance of role playing the constraints in the formulation of the problem. These constraints are added step by step to the problem and convert an arbitary subgraph into one with directed cycle (if any) and then a spanning directed out-tree rooted at a node will be presented and finally a spanning path (a feasible solution for the LOP) will be put forward. The model is formed as a linear programming problem in which some variables are binary. This optimization model can find the exact solution to the problem, and having a linear structure, it can be well implemented whenever desired by softwares such as matlab, lingo, and maple. Although binary variables, generally, cause some computational difficulties, we can overcome this problem desirably using well-known approaches in the field of integer programming such as cutting plane methods or interior point algorithms. However, the linear structure of the problem is useful both in theoretical point of view (using powerful methods such as simplex as well as a special kind of cutting plane technique [5]) and approximating implementation (applying simplex algorithm where all variables are continuous between 0 and 1).

Section 3 illustrates the application of the proposed model on a network. In closing section, we present several applicability of the model handling some other constraints in different applied problems. This section has been divided into three subsections including theoretical, computational, and applied results. By way of conclusion, we will suggest that the model can cover more generalized and complicated problems and it can also be reduced to classical ordering problem in its special case. These generalizations do not increase the number of binary variables and keep this cardinality twice the number of the edges in a graph.

2 Problem characterization

In this section, we formulate a problem that finds a directed spanning path with minimum cost and analyze the importance of each constraint that must be included in the formulation to characterize one of such these paths.

Consider an undirected connected graph $G = (N, E)$ consisting of a set N of n nodes ($|N| = n$)

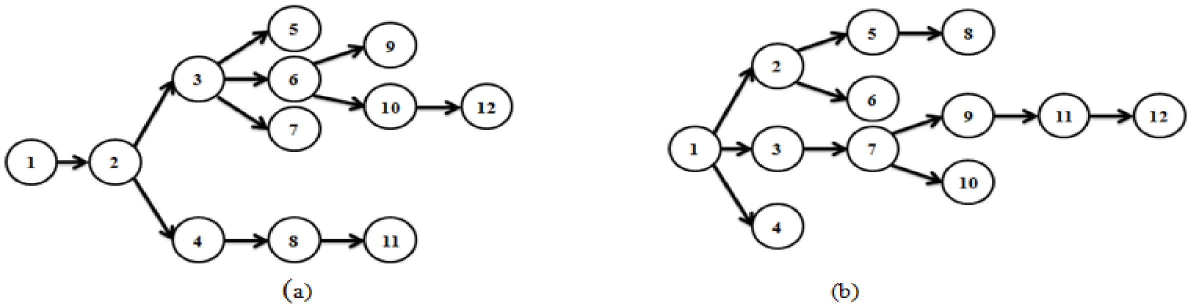


Figure 1: (a), (b) are directed out-tree rooted at node 1

whose elements (nodes) have been numbered from 1 to n and a set E of m edges ($|E| = m$). We refer to an undirected edge joining the node pair i and j as $\{i, j\}$ and a directed one from node i to node j as (i, j). Edge (i, j) is an outgoing edge of node i and an incoming edge of node j. Each directed edge (i, j) has an associated cost coefficient t_{ij} (in general, t_{ij} may not be equal to t_{ji}). Also, let P denote the set of all spanning directed paths in G (the paths under consideration in the ordering problems).
Since any solution of the feasible region in the ordering problem (set P) is directed (actually, directed spanning path), it is reasonable to work with a directed network as the network under consideration. So, we have the following definition.

Definition 2.1. *Let $\bar{E} = \{(i, j)$ and $(j, i) : \{i, j\} \in E\}$ and $\bar{G} = (N, \bar{E})$.*

By definition1, each undirected edge $\{i, j\} \in E$ is replaced by two directed edges (i, j) and $(j, i) \in \bar{E}$, and $\bar{G}$ is the network obtained from G by this transformation. We recall some terminologies in network flows studied, for example, in combinatorial optimization and then prove a theorem (Theorem 1) that will be of use later.

Definition 2.2. *A tree is said to be a directed out-tree rooted at a node if the unique path in the tree from that node to every other node is a directed path (figure1). We show the unique directed path from node i_1 to node i_k by notation $i_1 - i_2 - \cdots - i_k$ and let $|p_0|$ denote the length of p_0 (the number of edges in p_0) i.e, $p_0 = |\{(i, j) \in \bar{E} : (i, j) \in p_0\}|$.*
Also, we say that node i is the predecessore of node j in path p_0, if $(i, j) \in p_0$

Definition 2.3. *Let T (including n nodes numbered from 1 to n) be a directed out-tree rooted at a node $i_1 \in \{1, \cdots, n\}$. If L_i denotes the lable value of node i, we set, $L_{i_1} = 1$ and $L_j = 1 + L_i$ for each $(i, j) \in T$.*

Lemma 2.4. *Let T and label values $L_i(1 = 1, \cdots, n)$ be defined as in definitions 2 and 3. Suppose that k_s denotes the set of nodes having the same lable value equal to s i.e. $k_s = \{i \in N : L_i = s\}$. Then,*

a) If $i_1 - i_2 - \cdots - i_k$ is the unique directed path p_0 in T from i_1 to i_k, then $L_{i_k} = 1 + |p_0|$.

b) $k_1 = \{i_1\}$

c) $1 \leq L_i \leq n, \forall i \in N.$

d) If $k_s = \emptyset$*, then* $k_t = \emptyset$ *for each* $t \geq s$.

e) $\sum_{s=1}^{n} |k_s| = n$ *(*$|k_s|$ *denotes the cardinality of* k_s*).*

f) $\sum_{s=1}^{n} |k_s|.s = \sum_{i=1}^{n} L_i$

Proof. a) The existence of p_0 is resulted from definition 2. By definition 3, we have $L_{i_k} = 1 + L_{i_{k-1}} = 2 + L_{i_{k-2}} = \cdots = (k-1) + L_{i_1} = |p_0| + 1$. So $L_{i_k} = 1 + |p_0|$. If $i_1 = i_k$, p_0 is a trivial path consisting of one node i_1 and no edge. Hence, $|p_0| = 0$ and we have $L_{i_1} = 1$.

b) By definition 3, $L_{i_1} = 1$ that implies $i_1 \in k_1$. By part (a), for each $i_k \in N$, $L_{i_k} = 1 + |p_0|$. If $i_1 \neq i_k$, then $|p_0| > 0$ that implies $L_{i_k} \geq 2$. Thus, $i_k \notin k_1$.

c) For each $i_k \in N$, $L_{i_k} = |p_0| + 1$. Since p_0 has at most n nodes, then $p_0 \leq n-1$. Thus, $1 \leq L_{i_k} = 1 + |p_0| \leq n$ (if $i_k = i_1$, $|p_0| = 0$ and $L_{i_1} = 1$).

d) Since, $k_s = \emptyset$, we have from part (b), $s \neq 1$. Hence, $t \geq s > 1$. By contradiction, suppose that $i \in k_t$ (i.e. $L_i = t$) and p_0 is the unique directed path from i_1 to i. If, j_1 denotes the predecessor of node i in p_0, definition 3 implies $L_i = 1 + L_{j_1}$. Then, $L_{j_1} = t-1$. Similarly, for a node j_2, the predecessor of node j_1, we have $L_{j_1} = 1 + L_{j_2}$ that implies $L_{j_2} = t-2$. By repeating, there is a node j_{t-s} with $L_{j_{t-s}} = s$, so $j_{t-s} \in k_s$ that contradicts $K_s = \emptyset$

e) Since $k_s \subseteq N (s = 1, \cdots, n)$, we have $\bigcup_{s=1}^{n} k_s \subseteq N$. Also, for each $i \in N$, we have $i \in k_{L_i}$ ($1 \leq L_i \leq n$, from part (c)) that implies $N \subseteq \bigcup_{s=1}^{n} k_s$. Thus, $\bigcup_{s=1}^{n} k_s = N$. From definitions 2 and 3, if $s \neq t$, then $k_s \cap k_t = \emptyset$. Therefore, $\sum_{s=1}^{n} |k_s| = |N| = n$.

f) From the proof of part (e), $N = \bigcup_{s=1}^{n} k_s$ and $k_s \cap k_t = \emptyset$, if $s \neq t$. Hence,

$$\sum_{i=1}^{n} L_i = \sum_{s=1}^{n} \sum_{i \in k_s} L_i = \sum_{s=1}^{n} (\sum_{i \in k_s} s) = \sum_{s=1}^{n} (S. \sum_{i \in k_s} 1) = \sum_{s=1}^{n} S.|k_s|$$

.

□

$\square$

Lemma 2.5. *Let T and label values $L_i(i = 1, \cdots, n)$ be defined as in definitions 2 and 3. Suppose that $A = \sum\limits_{s=1}^{n} \varphi(k_s)$ where*

$$\varphi(k_s) = \begin{cases} \sum\limits_{i=1}^{|k_1|} i & s=1 \\ \sum\limits_{i=1}^{|k_s|} (|k_1| + |k_2| + \cdots + |k_{s-1}| + i) & k_s \neq \emptyset, s \neq 1 \\ 0 & k_s = \emptyset \end{cases}$$

Then,

a) $A = \frac{n(n+1)}{2}$

b) $\sum_{i=1}^{n} L_i \leq A$ in which equality holds iff $|k_s| = 1$

Proof. a) From lemma 1 (part (b)), $k_1 \neq \emptyset$. We set $k_{n+1} = \emptyset$ (it is justifiable because $|N| = n$). Then, without loss of generality, suppose that $t+1(t \in \{1, \cdots, n\})$ is the least index such that $k_{t+1} = \emptyset$. So, we have from lemma 1 (part(d)), $k_{t+1} = k_{t+2} = \cdots = k_n = \emptyset$.
Let $B_1 = \sum\limits_{i=1}^{t} |k_i|$ and $B_2 = \sum\limits_{i=t+1}^{n} |k_i|$. Therefore, $B_2 = 0$ and

A$= \sum_{s=1}^{n} \varphi(k_s) = \sum_{s=1}^{t} \varphi(k_s) =$

$\sum_{i=1}^{|k_1|} i + \sum_{i=1}^{|k_2|} (|k_1| + i) + \cdots \sum_{i=1}^{|k_t|} (|k_1| + \cdots + |k_{t-1}| + i)$

$= 1 + 2 + \cdots + \sum_{i=1}^{t} |k_i|$

$= \sum_{i=1}^{B_1} i = \sum_{i=1}^{B_1+B_2} i$

in which the last equality hold from lemma1 (part(e)).Thus , $A = \frac{n(n+1)}{2}$

b) Consider $\varphi(k_s)$ (s^{th} term in A). At first, suppose that $k_s \neq \emptyset$ and $s > 1$. By definition, $\varphi(k_s) = \sum\limits_{i=1}^{|k_s|} (|k_1| + \cdots + |k_{s-1}| + i)$. From lemma 1 (part (d)), we have $k_i \neq \emptyset$, $\forall i \in \{1, 2, \cdots, s-1\}$that implies $|k_i| \geq 1$, $\forall i \in \{1, 2, \cdots, s-1\}$. Hence, $s - 1 \leq |k_1| + \cdots + |k_{s-1}|$ that implies

$$s \leq |k_1| + \cdots + |k_{s-1}| + i, \forall i \in \{1, 2, \cdots, |k_s|\} \tag{*1}$$

and (*1) is an equality iff $|k_1| = \cdots = |k_{s-1}| = i = 1$. By summing inequalities (*1) for $i = 1, 2, ..., |k_s|$

$$|k_s|.S \leq \sum_{i=1}^{|k_s|} (|k_1| + \cdots + |k_{s-1}| + i) = \varphi(k_s) \tag{*2}$$

Since each side in (*1) is non-negative, (*2) is an equality iff (*1) is an equality for each $i \in \{1, 2, \cdots |k_s|\}$ iff $|k_1| = \cdots = |k_{s-1}| = |k_s| = 1$. In special cases, if $k_s = \emptyset$, then $|k_s| = \varphi(k_s) = 0$, that converts (*2) into trivial equality $0 = 0$. Also, if $s = 1$, we have, $1 = |k_1| = \sum_{i=1}^{|k_1|} i = \varphi(k_1)$ in which $|k_1| = 1$, from lemma 1 (part (b)). Therefore, (*2) is true for each $\varphi(k_s)$, $s = 1, 2, \cdots, n$. By summing inequalities (*2) for $s = 1, 2, \cdots, n$, we have $\sum_{s=1}^{n} |k_s|.s \leq \sum_{s=1}^{n} \varphi(k_s) = A$ in which the equality holds iff for each $s \in \{1, \cdots, n\}$, (*2) is an equality iff for each $s \in \{1, \cdots, n\}$, either $k_s = \emptyset$ or $|k_1| = \cdots = |k_s| = 1$ □

□

Theorem 2.6. *Let T and label values $L_i (i = 1, \cdots, n)$ be defined as in definitions 2 and 3. T is a directed path (from i_1 as the beginning node) iff $\sum\limits_{i=1}^{n} L_i = \frac{n(n+1)}{2}$.*

Proof. At first, we note from definition 2 that if T is a path, it is a directed path. However, if T is a directed path, the result follows directly by definition 3. Conversely, suppose that the equality holds. On the contrary, if T is not a path, there must exist a node p in T having more than one outgoing edge, say $(p, j_1), \cdots, (p, j_k) \in T$ (node 3 in figure 1.a, for example). Thus, by definition 3, we have $L_{j_1} = \cdots = L_{j_k} = 1 + L_p$. let $L_{j_1} = \cdots = L_{j_k} = q$. Since $|k_q| \geq 2$, lemma 2 (part(b)) implies $\sum\limits_{i=1}^{n} L_i < \frac{n(n+1)}{n}$. This contradiction completes the proof. □ □

Definition 4 below introduces the variables of our problem.

Definition 2.7. *Let W be a subgragh of $\bar{G}$ denoted by $W \subseteq \bar{G}$. Set,*

$$x_{ij} = \begin{cases} 1 & \textit{if } (i,j) \in W \\ 0 & \textit{otherwise} \end{cases}$$

Equivalently, every variable assignment $x_{ij} \in \{0, 1\}$ characterizes a subgraph of $\bar{G}$. We write $\{i, j\} \in W$, if $(i, j) \in W$ or $(j, i) \in W$. The following lemma gives a useful equivalence.

Lemma 2.8. *Let $T \subseteq \bar{G}$ be a directed out-tree rooted at a node $i_1 \in \{1, \cdots, n\}$. We set, $L_{i_1} = 1$ and*

$$L_j - L_i \leq (x_{ij} - x_{ji}) + (n-1)(1 - x_{ij} - x_{ji}) \tag{1}$$

$$L_j - L_i \geq (x_{ij} - x_{ji}) + (n-1)(x_{ij} + x_{ji} - 1) \tag{2}$$

Then, values $L_i (i = 1, \cdots, n)$ are the same as in definition 3.

Proof. Since $1 \leq L_i \leq n, \forall i \in N$ (lemma 1, part (c)) we have, $-(n-1) \leq L_j - L_i \leq n - 1$, $\forall i, j \in N$. By definition 4, $x_{ij} = x_{ji} = 0$ iff $(i, j), (j, i) \notin T$. In this case, (1) and (2) imply trivial inequalities $-(n-1) \leq L_j - L_i \leq n-1$. By definition 4, $(i, j) \in T$ (and then $(j, i) \notin T$) iff $x_{ij} = 1$ and $x_{ji} = 0$. From (1) and (2), we have $L_j - L_i = 1$ that implies $L_j = 1 + L_i$. Similarily, definition 4 implies $(j, i) \in T$ (and then $(i, j) \notin T$) iff $x_{ji} = 1$ and $x_{ij} = 0$. Thus, we have from (1) and (2), $L_j - L_i = -1$ that means $L_i = 1 + L_j$. Finally, from definition 3, it is impossible that $(i, j) \in T$ and $(j, i) \in T$. However, if we set $x_{ij} = x_{ji} = 1$, then (1) , (2) are converted to $(n-1) \leq L_j - L_i \leq -(n-1)$, that is not possible. □ □

For several reasons, we need to extend sets N and $\bar{E}$ by adding an artificial node s and n artificial directed edges $(s,i), i = 1, \cdots, n$ (from s to every original node) to $\bar{G}$. Some identities of this new structure are demonstrated by lemma 4 and theorem 2 below required to prove the main theorem of the paper. Other properties of G^* will be expressed later in the conclusion section. At first, we present this new structure formally as follows:

Definition 2.9. *Let* $G^* = \{N \cup \{s\}, E^*\}$ *where* $E^* = \bar{E} \cup \{(s,i) : i = 1, \cdots, n\}$.

Lemma 2.10. *Suppose that* $W \subseteq G^*$ *satisfies the following equalities:*

$$\sum_{j \in A(i) \cup \{s\}} x_{ji} = 1, i = 1, \cdots, n \tag{3}$$

Where $A(i) = \{j \in N : \{i,j\} \in E\}$ *and* $x_{si} = 1 (x_{si} = 0)$ *iff* $(s,i) \in W ((s,i) \notin W), i = 1, \cdots, n$. *Then,*

a) each $i \in N$ *is in* W *(i.e.* W *is spanning subgraph of* G^**).*

b) for each cycle $C \subseteq W$, $s \notin C$.

c) each cycle $C \subseteq W$ *is directed.*

Proof. a) Let $i \in N$ be an arbitrary node. By (3), there exists the unique node $j \in A(i) \cup \{s\}$ such that $x_{ji} = 1$. Now definition 4 implies $(j,i) \in W$, that means $i \in W$.

b) By contradiction, suppose that $s \in C$. Let C be as a sequence of nodes $i_k - i_{k-1} - \cdots - i_1 - s - j_1 - j_2 - \cdots - j_k - i_k$. From definition 5, $(j_1, s) \notin C \subseteq G^*$. Thus, we must have $(s, j_1) \in C$. From (3), $(j_1, j_2) \in C$. Or else, if $(j_2, j_1) \in C$. then j_1 has two incoming edges (s, j_1) and (j_2, j_1), and then definition 4 implies $x_{sj_1} = x_{j_2 j_1} = 1$,that contradicts (3) for $i = j_1$. Using this argument repeatedly, it follows that $(j_2, j_3), (j_3 . j_4), \cdots, (j_{k-1}, j_k)$ and $(j_k, i_k) \in C$. Similarly, $(s, i_1) \in C$ and we have, by repeating the preceding argument,

$$(i_1, i_2), (i_2, i_3), \cdots, (i_{k-2}, i_{k-1})$$

and $(i_{k-1}, i_k) \in C$. Thus, node i_k has two incoming edges (j_k, i_k) and (i_{k-1}, i_k) that contradicts (3).

c) Let C be as $i_1 - i_2 - \cdots - i_r - i_1$ ($s \notin C$, from part (b)). Without loss of generality, let $(i_1, i_2) \in C$. From (3) and definition 4, each node $i \in N$ must have exactly one incoming edge. Thus, since $(i_1, i_2) \in C$, we have $(i_2, i_3) \in C$. Using (3) repeatedly, we have $(i_3, i_4), \cdots, (i_{r-1}, i_r)$ and $(i_r, i_1) \in C$. This proves that C is a directed cycle. □

□

Theorem 2.11. *Let $W \subseteq G^*$ and variables $L_i(i = s, 1, \cdots, n)$ be arbitrary (none of them is pre-specified). If W satisfies (1),(2) and (3), then W is a spanning directed out-tree rooted at node s.*

Proof. At first, we show that W has no cycle. By contradiction, suppose that a cycle $C \subseteq W$. According to lemma 4 (part (c)), C is directed. Hence, C is a directed path $j_1 - j_2 - \cdots - j_r$ from j_1 to j_r together with (j_r, j_1). Therefore, (1) and (2) imply that $L_{j_1} < L_{j_r}$ (because of the path) and $L_{j_r} < L_{j_1}$ (because $(j_r, j_1) \in C$). By this contradiction, the result follows. Now, since W has at least n nodes as well as n edges (from lemma 4, part (a), and (3)), it must have $n+1$ nodes in the first place (i.e. $s \in W$), and then exactly n edges in the second place (or else, if each of both cases is not true, W has a cycle). Therefore, W is spanning by the former statement and a tree by the latter. Finally, suppose, by contradiction, that the path $s - j_1 - j_2 - \cdots - j_k$ in W is not directed. From the definition of G^*, we have $(s, j_1) \in W$. If $(j_2, j_1) \in W$, then, j_1 contradicts (3). Hence, $(j_1, j_2) \in W$. In the same manner, we have, $(j_{r-1}, j_r) \in W$ for $r = 3, 4, \cdots, k$. But, this contradicts, that the path is not directed. Therefore, the unique path in W from s to every other node is directed. This completes the proof. □

□

Corollary 2.12. *Let $W = (N_W, E_W)$ be a subgraph of G^* and satisfy (1),(2),(3) and following constraints:*

$$\sum_{j=1}^{n} x_{sj} = 1, \ \ L_s = 0 \tag{4}$$

If $x_{si_1} = 1$ for some $i_1 \in \{1, \cdots, n\}$, then $\bar{W} = (N_W - \{s\}, E_W - \{(s, i_1)\})$ is a spanning directed out-tree rooted at node i_1 in $\bar{G}$ with $L_{i_1} = 1$

Proof. From theorem 2, W is a spanning directed out-tree rooted at node s. Hence, W has $n+1$ nodes ($N_W = N \cup \{s\}$) and n edges. From (4), node i_1 is only node connected to s with $L_{i_1} = 1 + L_s = 1$. Thus, by deleting node s and edge $(s, i_1), \bar{W} = (N, E_W - \{(s, i_1)\})$ is still a connected graph consisting of n nodes and $n-1$ edges. Therefore, $\bar{W}$ is a spanning tree in $\bar{G}$. On the other hand, for an arbitary node $i \in N$, there exists the unique directed path $s - i_1 - i_2 - \cdots - i$ in W from s to i. Thus, by removing node s and edge $(s, i_1), i_1 - i_2 - \cdots - i$ is the unique directed path in $\bar{W}$ from i_1 to an arbitrary node $i \in N$. Therefore, $\bar{W}$ is a directed out-tree rooted at node i_1. This completes the proof. □

□The following theorem characterizes the feasible solution set of the ordering problem.

Theorem 2.13. *Let $W \subseteq G^*$. If W satisfies (1),(2),(3),(4) and $\sum\limits_{i=1}^{n} L_i = \frac{n(n+1)}{2}$, then $\bar{W} \in P$*

Proof. The result follows from corollary1 and Theorem1. □

□

Finally, the objective function of the proposed model is considered as $\sum\limits_{j=1}^{n} \sum\limits_{i=1}^{n} t_{ij} x_{ij}$ for each $(i, j) \in \bar{E}$, where the cost coefficients of the artificial edges t_{sj}, $j = 1, \cdots, n$, are equal to 0. From theorem 3, the model of the problem is written as follows:

$$
\begin{aligned}
min \quad & \sum_{j=1}^{n}\sum_{i=1}^{n} t_{ij}x_{ij} \\
& \text{for} \quad i, j = s, 1, \cdots, n \\
& L_j - L_i \leq (x_{ij} - x_{ji}) + (n-1)(1 - x_{ij} - x_{ji}) \\
& L_j - L_i \geq (x_{ij} - x_{ji}) + (n-1)(x_{ij} + x_{ji} - 1) \\
& \sum_{j \in A(i) \cup \{s\}} x_{ji} = 1 \qquad\qquad , i = 1, \cdots, n \\
& \sum_{j=1}^{n} x_{sj} = 1, L_s = 0 \\
& \sum_{i=1}^{n} L_i = \frac{n(n+1)}{2} \\
& x_{ij} \in \{0, 1\} \quad i, j = s, 1, ..., n
\end{aligned} \tag{5}
$$

Remark.

a) It is not necessary, to add a restriction on the outgoing edges of nodes in the problem i.e. $\sum_j x_{ij} = 1$, $i = s, 1, \cdots, n$. This constraint is automatically satisfied, as shown in theorems 2 and 3.

b) By theorem 3, the labels of the nodes in a path are ordered from 1 to n. We can find the beginning node of a path simply in two ways; (I) as the node connected to node s (node i_1, say, for which $x_{si_1} = 1$) or (II) as the node with a label value equal to 1. The end node of a path is the node with a label value equal to n. Other labels also show the position of the interior nodes in a path.

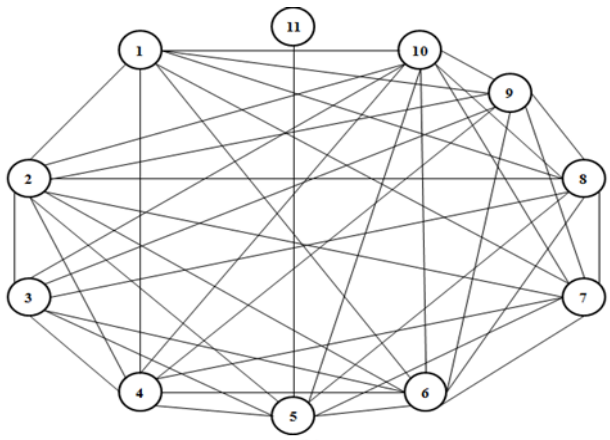
c) Similar to variables L_i, we can define variables $p_i (i = s, 1, \cdots, n)$ as follows:

$$
\begin{aligned}
& p_s = 0 \\
& p_j - p_i \leq t_{ij}(x_{ij} - x_{ji}) + (n-1)(1 - x_{ij} - x_{ji}) \\
& p_j - p_i \geq t_{ij}(x_{ij} - x_{ji}) + (n-1)(x_{ij} + x_{ji} - 1)
\end{aligned}
$$

It is easy to verify that p_i gives actually the time consumed for traveling a path from its beginning node to node i.

$$M = \begin{bmatrix} 0 & 5 & 0 & 1 & 0 & 1 & 3 & 1 & 2 & 1 & 0 \\ 5 & 0 & 1 & 1 & 1 & 2 & 1 & 2 & 1 & 1 & 0 \\ 0 & 1 & 0 & 7 & 5 & 2 & 0 & 1 & 2 & 1 & 0 \\ 1 & 1 & 7 & 0 & 2 & 6 & 2 & 0 & 9 & 2 & 0 \\ 0 & 1 & 5 & 2 & 0 & 8 & 1 & 9 & 0 & 2 & 1 \\ 1 & 2 & 2 & 6 & 8 & 0 & 2 & 2 & 2 & 3 & 0 \\ 3 & 1 & 0 & 2 & 1 & 2 & 0 & 1 & 7 & 1 & 0 \\ 1 & 2 & 1 & 0 & 9 & 2 & 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 2 & 9 & 0 & 2 & 7 & 1 & 0 & 2 & 0 \\ 1 & 1 & 1 & 2 & 2 & 3 & 1 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a)



(b)

Figure 2: (a) node-node adjacency matrix $M = (m_i)_{11\times 11}$ in which $m_{ij} = t_{ij}$, (b) undirected graph $G = (N, E)$

3 Numerical example

Figure 2.a depicts an undirected graph $G = (N, E)$ with 11 nodes and 41 edges. In figure 2.b, $M = (m_{ij})_{11\times 11}$ shows the node-node adjacency matrix of G where $m_{ij} = 0$, if $\{i, j\} \notin E$ and $m_{ij} = t_{ij} \in E$ (t_{ij} is the cost coefficient of edge $\{i, j\}$). Figure 3 depics the minimum spanning directed path (that is from node 2 to node 11) resulted from theorem 3 with the objective function value 12 and minimum spanning directed out-tree (that is rooted at node 11) characterised by theorem 2 with the objective function value 10 (in figure 3.(b), since the result is a tree, not a path, then the objective function value is not equal to p_{11}). L_i 's are the positions of nodes in the optimal path. $L_2 = 1$ introduces node 2 as the beginning node of the optimum path and $L_{11} = 11$ says that node 11 is the end node. Also, for example, $L_7 = 9$ shows that node 7 is 9^{th} node in the path and $p_8 = 3$ gives the time of travel from the beginning (node 2) to node 8. Furthermore, we have $p_2 = 0$ and $p_{11} = 12$ (total time needed for traveling all of the path)

4 Conclusion

In this section, we present the properties of the proposed model in three subsections.

4.1 Theoretical aspects

We investigated the constraints of the problem with emphasis on the assessment of their importance in the formulation. These constraints, step by step, confined the domain of the subgraphs in the main network to find a directed spanning path as a feasible solution.

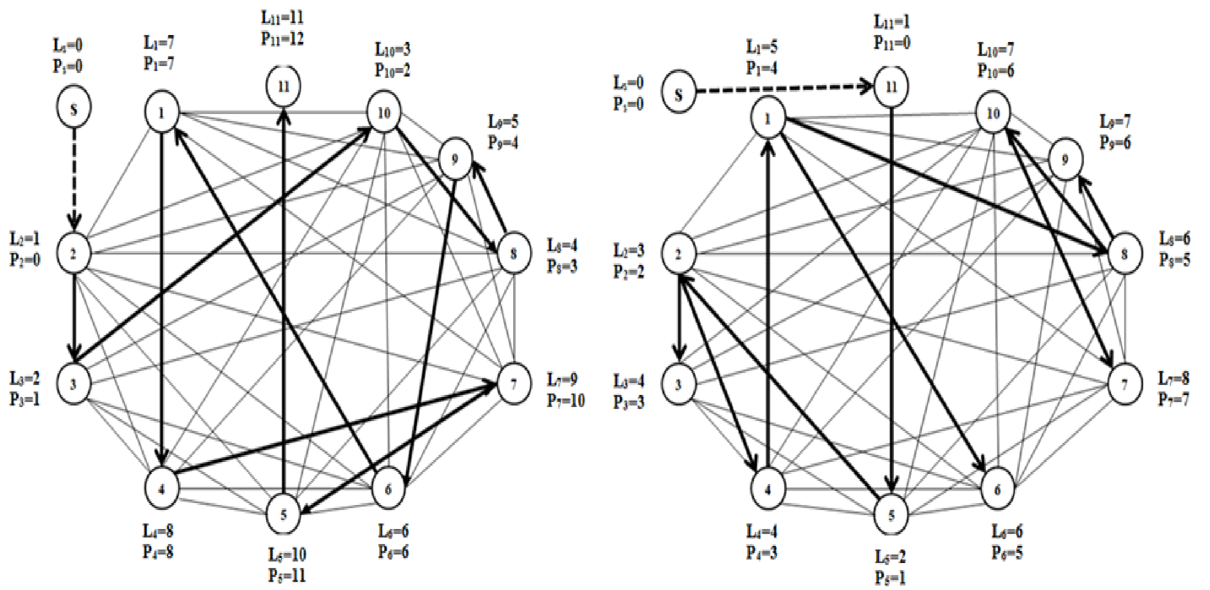


Figure 3: (a) Minimum spanning directed path (from node 2 to node 11), (b) Minimum spanning directed out-tree (rooted at node 11)

The properties proved by lemma 4 and theorem 2 can be used in problems in which other optimal structures such as cycles and trees instead of spanning paths are interested to be found.
Similar to node s, one can add another artificial node t (with $L_t = n + 1$, if necessary) and n directed edges $(j, t), j = 1, \cdots, n$ for other special applications.

4.2 Computational aspects

The model has $2m + n$ binary variables:
$2m$ variables x_{ij} for each directed link and n variables x_{sj} in order to construct of G^*. Variables L_i and P_i are automatically attained as integer. However, there are several ways to handle such these situations as stated in the introduction.
The model has $2m + n + 2$ constraints:
$2m$ constraints by (1) and (2) to find a desired tree ($\bar{G}$ has at least m cycles for each pair of x_{ij} and x_{ji}).
n constraints by (3) to find a spanning solution (n nodes must be included).
One constraint by (4) to find a beginning node (each node may play the role of the beginning node in an optimal path).
One constraint by (5) to find a path.

4.3 Applied aspects

Some problems that can be formulated as this model are presented as follows:

Solving the ordering problem by assuming pre-specified ordering on some nodes in the optimal path. For example $L_4 = 6, L_5 = 1, L_6 = 2$
Solving the ordering problem in which some nodes have to be visited before or after (locally) some others. For example $L_4 \leq L_6, L_4 \leq L_{10} \leq L_2$
Solving the ordering problem in which the orders of nodes satisfy an special formula (rule). For example, $P_1 \leq P_3 + 4$, $P_{11} = P_3 + P_5$, $P_2 = 2k + 1$ $(k \in N)$, $L_3 = 2k$, $L_4 + L_{10} = L_2$
Solving the ordering problem in which some nodes have to be visited after or before a certain time or in a certain period of time. For example $P_1 \leq 5$, $P_3 \geq 4$, $7 \leq P_2 \leq 9$.
Solving the ordering problem in which some nodes have to be visited befor or after (temporally) some other nodes. For example $P_3 \geq P_4$
Finding the shortest path emanating from a certain node i_0: we set initially, $L_{i_0} = 1$, in problem (5)
Finding the shortest path terminating at the certain node i_0: we set initially, $L_{i_0} = n$, in problem (5)
Finding the shortest path from a certain node i_0 to a certain node i_1 such that node i_2 is in the middle of the path. For example, if n is an even number, we set $L_{i_0} = 1$, $L_{i_1} = n$, $L_{i_2} = \frac{n}{2}$.
Finding the shortest path with certain length or cost equal to R_0 from node i_0 to node i_1. We add the following constraints in problem (5):

$$\sum_j \sum_i c_{ij} x_{ij} = R_0$$
$$L_{i_0} = 1, \;\; L_{i_1} = n$$

5 Acknowledgement

This work was supported by Tehran University. Our special thanks go to the University of Tehran, College of Engineering and Department of Engineering Science for providing all the necessary facilities available to us for successfully conducting this research.

References

[1] Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi., Complexity and approximation . Springer-Verlag, Berlin, 1999.

[2] Bachem, A. Grötschel, M. : New aspects of Polyhedral Theory, in Korte, B (ed): Modern Applied Mathematics: Optimization and Operations Research, North-Holland Publishing Co. Amsterdam - New York -. Oxford, 51-106, 1982.

[3] Barenholz, U. Feige, U., Peleg, D.: Improved approximation for min-sum vertex cover. (2006)

[4] Bar-Yehuda, R. Even, G., Feldman, J., and Naor, J. Computing an optimal orientation of a balanced decomposition tree for linear arrangement problems. Journal of Graph Algorithms and Applications 5, 4, 1-27. 2001.

[5] Bazarra, M. S. Jarvis, J., Sherali, H. D., Linear programming and network flows, Third Edition, Published Online: 15 AUG 2011.

[6] Becker, O. : Das Helmstädtersche Reihenfolgeproblem - die Effizienz verschiedener Näherungsverfahren, in Computer Uses in the Social Science, Wien, January 1967.

[7] Campos, V. Glover, F., Laguna, M. and Martí, R.: An experimental evaluation of a scatter search for the linear ordering problem , J. Global Optim. 21(4), 397-414, 2001.

[8] Campos, V. Laguna, M. and Martí, R.: Scatter search for the linear ordering problem, in D. Corne, M. Dorigo and F. Glover (eds), New Ideas in Optimization, McGraw-Hill, London, UK, pp. 331-339. 1999.

[9] Chanas, S.and Kobylanski, P.: A new heuristic algorithm solving the linear ordering problem , Comput. Optim. Appl. 6 , 191-205, 1996.

[10] Charikar, M. Hajiaghayi, T., Karlo, H., and Rao, S. B., 22 spreading metrics for ordering problems. To appear, SODA, 2006.

[11] Chekuri, C., Ene, A.: Approximation algorithms for submodular multiway parti- tion. In: Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science. FOCS '11, Washington, DC, USA, IEEE Computer Society .807-816 . 2011.

[12] Congram, R. K.: Polynomially searchable exponential neighbourhoods for sequencing problems in combinatorial optimisation, PhD thesis, University of Southampton, Faculty of Mathematical Studies, UK, 2000.

[13] Corso, G. M. D. and Romani, F. Heuristic spectral techniques for the reduction of bandwidth and work-bound of sparse matrices. Numerical Algorithms 28, 1-4 (Dec.), 117-136. 2001a.

[14] Devanur, N.R., Khot, S.A., Saket, R., Vishnoi, N.K.: Integrality gaps for sparsest cut and minimum linear arrangement problems. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing. STOC '06, New York, NY, USA, ACM . 537-546, 2006.

[15] Even, J. Naor, S. Rao, and B. Schieber. âŁœDivide-and-conquer approximation algorithms via spreading metricsâŁž. In 36th IEEE Symposium on Foundations of Computer Science,pages 62âŁ“71, 1995.

[16] Feige, U., Lee, J.R.: An improved approximation ratio for the minimum linear arrangement problem. Inf. Process. Lett. 101 . 26-29. (January 2007).

[17] Feige, U., Lov-asz, L., Tetali, P. Approximating min sum set cover. Algorithmica 40 . 219-234 . 2004.

[18] Garey, M. R., Johnson, D. S. Computers and Intractability. A guide to the theory of NP-completeness. Freeman and Company, 1979.

[19] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. Theoretical Computer Science, 1:237-267, 1976.

[20] Goel, G., Karande, C., Tripathi, P., Wang, L.: Approximability of combinatorial problems with multi-agent submodular cost functions. In: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science. FOCS '09, Washington, DC, USA, IEEE Computer Society . 755-764. 2009.

[21] Grötschel, M. Padberg, MW: Polyhedral aspects of the Traveling Salesman Problem I. Theory, European Institute for Advanced Studies in Management. W. Germany. 1983a.

[22] Grötschel, M. Padberg, MW: Polyhedral aspects of the Traveling Salesman Problem II. Computation., European Institute for Advanced Studies in Management. W. Germany. 1983b.

[23] Grötschel, M: Approaches to Hard Combinatorial Optimization Problem, in Korte, B (ed) : Modern Applied Mathematics: Optimization and Operations Research, North-Holland Publishing Co. Amsterdam - New York -. Oxford, 437-575, 1982.

[24] Grötschel, M., Jünger, M. and Reinelt, G.: A cutting plane algorithm for the linear ordering problem, Oper. Res. 32(6), 1195-1220, 1984.

[25] Hansen, M., Approximation algorithms for geometric embeddings in the plane with applications to parallel processing problems, in Proceedings of the 30th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, pp.604-609, 1989.

[26] Hu, Y. F. and Scott, J. A. A multilevel algorithm for wavefront reduction. SIAM J. Sci. Comput. 23, 4, 1352-1375. 2001.

[27] Ilya Safro. Dorit Ron. Achi Brandt : Multilevel Algorithms for Linear Ordering Problems, The Weizmann Institute of Science . 2009.

[28] Iwata, S., Nagano, K.: Submodular function minimization under covering con- straints. In: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science. FOCS '09, Washington, DC, USA, IEEE Computer Society . 671-680. 2009.

[29] Juvan, M. Mohar, B., Optimal linear labeling and eigenvalues of graphs. Discrete Applied Mathematics, 36(2):153-168, 1992.

[30] Koren, Y. and Harel, D. Multi-scale algorithm for the linear arrangement problem. Proceedings of 28th Inter. Workshop on Graph-Theoretic Concepts. 2002.

[31] Laguna, M., Martí, R. Campos, V.: Intensification and diversification with elite tabu search solutions for the linear ordering problem, Comput. Oper. Res. 26(12) , 1217-1230. 1999.

[32] Leighton, F. T. Rao, S. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms, J. ACM, 46, pp. 787-832, 1999.

[33] Liu, W. Vannelli, A. Generating lower bounds for the linear arrangement problem. Discrete Applied Mathematics, 59:137-151, 1995.

[34] Mitchell, J. E. and Borchers, B.: Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm, in H. L. Frenk, K. Roos, T. Terlaky and S. Zhang (eds), High Performance Optimization, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 349-366. 2000.

[35] Mushi, A.R : The Linear Ordering Problem: An Algorithm for the optimal solution, Mathematics Department, University of Dar es salaam. African Journal of Science and Technology (AJST) Science and Engineering Series Vol. 6, No. 1, pp. 51-64.

[36] Petit, J. Experiments on the minimum linear arrangement problem. ACM Journal of Experimental Algorithmics, 8. 2003.

[37] Rao, S. Richa, A. W., New approximation techniques for some linear ordering problems. SIAM J. Comput. 34(2):388-404, 2004.

[38] Ravi, R. Agrawal, A. Klein, P., Ordering problems approximated: Single-processor scheduling and interval graph completion, in Proceedings of the 18th International Colloquium on Automata, Languages, and Programming, Lecture Notes in Comput. Sci. 510, Springer-Verlag, Berlin, pp. 751-762, 1991.

[39] Safronetz, D., R. Lindsay, B. Hjelle, R. A. Medina, K. Mirowsky-Garcia, and M. A. Drebot. Use of IgG avidity to indirectly monitor epizootic transmission of Sin Nombre virus in deer mice (Peromyscus maniculatus). American Journal of Tropical Medicine and Hygiene 75: 1135-1139 . 2006.

[40] Satoru Iwata (Research Institute for Mathematical Sciences, Kyoto University), Prasad Tetali (School of Mathematics, Georgia Institute of Technology) , and Pushkar Tripathi (School of Computer Science, Georgia Institute of Technology) : Approximating Minimum Linear Ordering Problems. 2013

[41] Schiavinotto, T. and Stützle, T.: Search space analysis of the linear ordering problem, in G. R. Raidl et al. (eds), Applications of Evolutionary Computing, Lecture Notes in Comput. Sci. 2611, Springer-Verlag, Berlin, pp. 322-333. 2003.

[42] Seymour, P. D. Packing directed circuits fractionally, Combinatorica, 15, pp. 281-288, 1995.

[43] Yehuda Koren. David Harel : A Multi-Scale Algorithm for the Linear Arrangement Problem, Dept. of Computer Science and Applied Mathematics The Weizmann Institute of Science, Rehovot, Israel , yehuda,harel, wisdom. weizmann.ac.il. 2002.