A Unified Framework for Integer Programming Formulation of Graph Matching Problems

Bahram Alidaee[1,*], Haibo Wang[2] and Hugh Sloan[1]

[1]School of Business Administration, The University of Mississippi, University, MS 38677, USA

[2]Sanchez School of Business, Texas A&M International University, Laredo, TX 78041, USA

Abstract: Graph theory has been a powerful tool in solving difficult and complex problems arising in all disciplines. In particular, graph matching is a classical problem in pattern analysis with enormous applications. Many graph problems have been formulated as a mathematical program then solved using exact, heuristic and/or approximated-guaranteed procedures. On the other hand, graph theory has been a powerful tool in visualizing and understanding of complex mathematical programming problems, especially integer programs. Formulating a graph problem as a *natural* integer program (IP) is often a challenging task. However, an IP formulation of the problem has many advantages. Several researchers have noted the need for natural IP formulation of graph theoretic problems. The aim of the present study is to provide a unified framework for IP formulation of graph matching problems. Although there are many surveys on graph matching problems, however, none is concerned with IP formulation. This paper is the first to provide a comprehensive IP formulation for such problems. The framework includes variety of graph optimization problems in the literature. While these problems have been studied by different research communities, however, the framework presented here helps to bring efforts from different disciplines to tackle such diverse and complex problems. We hope the present study can significantly help to simplify some of difficult problems arising in practice, especially in pattern analysis.

Keywords: Combinatorial optimization, graph matching, integer programming, quadratic assignment problem.

1. INTRODUCTION

Graph theory has been a powerful tool in solving difficult and complex problems that arise in different disciplines. In particular, *graph matching* has enormous applications. Recent surveys of graph theoretic problems are presented in [1-9]. Most graph optimization problems are NP-hard. In fact, all problems discussed in this paper are known to be NP-hard, except for the *graph isomorphism* and the *Golomb Ruler* problems whose complexity statuses are still open, [10,11].

Over the years, many graph and combinatorial problems have been formulated as an optimization problem and have been solved by using exact, heuristic, or approximation-guaranteed procedures [12-16]. Similarly, graph theory has been a powerful tool in visualizing and understanding complex optimization problems, (see for example, [17-19] for integer programming, [20-23] for image analysis, [24-26] for bioinformatics, and [27] for the social sciences).

Since the graph theory and optimization are so closely related, the theoretical and algorithmic developments from one area have been successfully used to address difficult and complex problems in the other discipline. Formulating graph problems as *natural* integer programs (IP) is often a challenging task. However, to tackle such difficulty researchers often use an *indirect* IP formulation of the problem and use it as an upper (lower) bound to design algorithms. A series of *indirect* linear programs for seven graph theoretic problems is presented in [28] where it is noted that "the formulation is indirect in the sense that it captures certain properties of an optimal (integer) solution, yet an integral solution (to the formulation) does not provide an integral solution to the original problem, as opposed to standard linear programming relaxation" (Page 588). A *natural* integer programming formulation of a graph problem has several advantages: (1) it is equivalent to the graph problem;(2) its relaxation often serves as a good upper (lower) bound for the graph problem; and (3) it is often used in designing approximation-guaranteed and heuristic algorithms. A unified approach based on graph matching and using mathematical programming formulations provides avenues for better understanding of relationships between these complex problems. In connection to metric labeling problems, Kleinberg and Tardos in [29] presented *natural* IP formulations for special cases and gave approximation-guaranteed results. They pointed out that "the difficulty of the general case has to do with the absence of a *natural* integer programming formulation for the problem," [30]. In that spirit, Chekuri, Khanna, Naor and Zosin in [30] presented a *natural* IP formulation for the general case with an approximation-guaranteed algorithm. Several other papers, [30-33], have pointed out the need for *natural* IP formulations of

[*]Address correspondence to this author at the School of Business Administration, The University of Mississippi, University, MS 38677, USA; Tel: 662-915-5466; E-mail: balidaee@bus.olemiss.edu

graph problems. An early IP formulation of a matching problem is given in [34]. Luttamazuzi, Pelsmajer, Shen and Yang in [33] suggested a *natural* IP formulation for three graph optimization problems, including *Bandwidth* minimization. A different *natural* IP formulation of Bandwidth minimization is also introduced in [35]. The first approach to a *natural* IP formulations of a *linear arrangement* problem is given in [36,37]. In [38], a *natural* IP formulation of a graceful labeling problem is discussed. A *natural* IP formulation for multi-layer crossing minimization is proposed in [39]. The *natural* IP formulations for the Golomb Ruler problem are presented in [10,40] and in [41], a *natural* IP formulation for a *contact map problem* is given.

In Part I of this paper, we provide a unified approach based on *graph matching* that allows *natural* integer program formulations for many graph theoretic problems. The relationship between these problems and the quadratic assignment problem (QAP) is pointed out. The approaches include an IP formulation for a variety of problems including the list given below. IP formulations for problems 1-12are presented in Part II, while IP formulations for problems 13-18 are presented in Part I. For some of these problems, we do not know of any other *natural* IP formulations available in the literature, for some other problems our formulations are different from those that appear in the literature and provide alternative approaches.

1. Several variations of traveling salesman problem,
2. Bandwidth problem,
3. Linear arrangement problem,
4. Profile minimization in graphs,
5. Minimum cut linear arrangement,
6. Graph and subgraph isomorphism,
7. Largest common subgraph problem,
8. Maximum subgraph matching,
9. Graph coloring and related problems: graph homomorphism, list coloring, optimum cost chromatic partition, sum coloring, graph labeling, *T*-coloring, maximum independent set and clique, and maximum clique partitioning,
10. Metric labeling variations: embedding a graph in a *d*-dimensional mesh, linear arrangement with *d*-dimensional cost, minimizing storage-time product, and task assignment in distributed computing systems,
11. Golomb ruler problem,
12. Contact map problem,
13. Graph orientation
14. Facility layout problem,
15. Quadratic assignment problem,
16. Fixed spectrum frequency assignment,
17. Interval graph completion,
18. Multi-layer crossing minimization.

PART I

2. GENERAL APPROACHES

Parameters:

*G=(V,E),*an undirected graph,

*G=(V,A),*a directed graph,

V, (|V|=n), set of vertices in a graph *G,*

$E \subseteq V \times V$, set of edges in undirected graph *G* (allowing possible self-loop),

$A \subseteq V \times V$, set of arcs in digraph *G* (no self-loop is allowed),

$a = (u,v) \in A$, an arc directed from *u* toward *v,*

$w_e(w_a)$, weight of an edge $e = \{u,v\} \in E$ (an arc $a = (u,v) \in A$),

$\bar{E}$, set of edges in complement of a graph *G,*

$\bar{A}$ *,* set of edges in complement of a digraph *G,*(i.e., removing all 'arcs' in *A* and adding an 'edge'

between every two nodes where there is no arc in $u \in V$. Thus, if $\{u,v\} \in \bar{A}$ then $(u,v),(v,u) \notin A$),

$N(u) = \{v \in V : \{u,v\} \in E\}$, neighbors of a node $u \in V$ in graph *G,*

$N(u) = \{v \in V : (u,v) \in A\}$ *,* neighbors of a node $u \in V$ in digraph *G,*

$N[u] = N(u) \cup \{u\}$,

$f : V \to V'$, a function mapping *V* in graph *G* to*V'* in graph *G',*

c(u,u'), non-negative cost of assigning a node $u \in V$ to a node $u' \in V'$,

d({u,v},{u',v'}), the *interaction (or communication)* costs (possibly negative) between two edges, *d((u,v),(u',v')),* the *interaction (or communication)* costs (possibly negative) between two arcs.

Decision Variables:

$x_{uu'} = 1$, if a node $u \in V$ is assigned to a node $u' \in V'$, and 0 otherwise,

$y_{ee'} = 1$, if an edge $e \in E$ is assigned to an edge $e' \in E'$, and 0 otherwise,

$y_{aa'} = 1$, if an arc $a \in A$ is assigned to an arc $a' \in A'$, and 0 otherwise,

$y^{\tau}_{ee'} = 1$, if an edge $e \in E$ is assigned to an edge $e' \in E'$, and $d(e,e') = \tau$ for a constant τ; and 0 otherwise,

$y^{\tau}_{aa'} = 1$, if an arc $a \in A$ is assigned to an arc $a' \in A'$, and $d(a,a') = \tau$ for a constant τ; and 0 otherwise,

$K \geq 0$, non negative variable (in some cases it is a given constant),

$K_u \geq 0$ ($K_{u'} \geq 0$), for $u \in V$ ($u' \in V'$),non negative variable.

With no loss of generality, we let elements of *V* and *V'* be integer values, i.e., *V={1,…,n}* and *V'={1,…,n'}.* Notations $u \in V$ and $u' \in V'$ represent nodes in graph (or digraph) G and *G',* respectively. Since the nodes are integer values thus for ease of presentation, we may use them as weight of the nodes. With these notations, for $u, v \in V$ and $u', v' \in V'$, we denote |*u-v*| and |*u'-v'*|, as absolute values of the difference between the numbers representing the nodes. For each problem, we look for a *matching* (possibly many-to-many), $f: V \to V'$, between nodes of two graphs where some conditions need to be satisfied. The imposed conditions depend on the given problem. For most problems, we define a graph *G'=(V',E')*in the case of undirected graph; and a digraph *G'=(V',A')* in the case of directed graph, with the '*desirable'* output characteristics, then we find a function *f* to map nodes of graph *G* to nodes of the defined graph *G'.* Example 1 illustrates this point.

Example 1. Consider the *k*-traveling salesman problem (*k*-TSP). In *k*-TSP we are given a graph *G* (directed or undirected) with a distance between edges (arcs in the case of digraph). We look for a minimum distance tour (circuit) of *k* cities [42,43]. When *k=n*, we have the TSP (see [43,44], for comprehensive surveys, and [45,46] for classification of integer programming formulations). In *k*-TSP, we define a graph *G'=(V',E')* in the form of a cycle, Figure **1** without arrows, and in directed graph *G'=(V',A')* in the form of a circuit, Figure **1**, with *k* nodes and distance one between consecutive cities. Then we look for a subset $S \subseteq V$ with |*S*|=*k,* and a one-to-one function, $f: S \to V'$, that minimizes the total distance on the tour (circuit), i.e., a *subgraph isomorphism* of *G* to *G'* that minimizes total distance traveled. In the case of Bottleneck *k*-TSP, one looks for a tour (circuit) to minimize the maximum distance traveled between any two cities. As we mentioned, here we have a *subgraph isomorphism (SI)* of *G* to *G'.* Given two undirected graphs *G=(V,E)* and *G'=(V',E'),* with $|V| = n \geq |V'| = n'$, a *subgraph isomorphism* of *G* to *G'* is a graph $\overline{G} = (S, L)$ with $S \subseteq V$ and $L \subseteq E$ such that |*S*|=|*V'*|*,* |*L*|=|*E'*| and there exists a one-to-one function $f: V' \to S$ satisfying $\{u', v'\} \in E'$ if and only if $\{f(u'), f(v')\} \in L$. Equivalently, SI asks whether there is a subset of edges and vertices of *G* that is isomorphic to a smaller graph *G'*. A slightly different *k-TSP* may be defined as follows, called *induced k-TSP*. In that the subgraph is an *induced subgraph isomorphism (ISI)*. Given two undirected graphs *G=(V,E)* and *G'=(V',E'),* with $|V| = n \geq |V'| = n'$, an *induced subgraph isomorphism* of *G* to *G'* is a graph $\overline{G} = (S, L)$ with $S \subseteq V$ and $L \subseteq E$ such that |*S*|=|*V'*|, |*L*|=|*E'*| and there exists a one-to-one function $f: V' \to S$ satisfying $\{u', v'\} \in E'$ if and only if $\{f(u'), f(v')\} \in E$. Equivalently, the question is whether we can delete vertices from *G* to obtain a graph isomorphic to *G'*. Note that in SI we do not rule out possibility of existing $\{u, v\} \in E \setminus L$ such that *f(u')=u* and *f(v')=v* for some $\{u', v'\} \in E'$. To illustrate the points, consider graph *G* in Figure **2**, where there are several subgraphs with 4 nodes and 4 edges that each creates a *4-TSP*. For example, 1-2-3-4-1 is a subgraph which is a *4-TSP*, while it is not an *induced subgraph.* Another 4

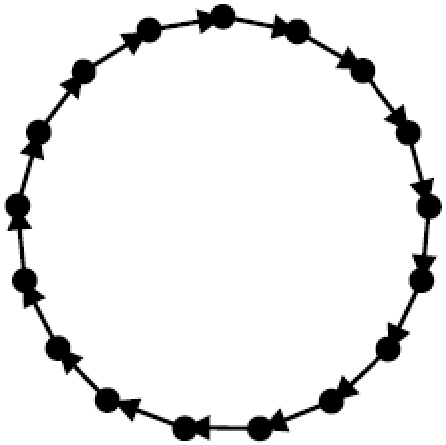

Figure 1: A desired feasible output graph for TSP with k nodes.

nodes 4edges subgraph which is *4-TSP* is 1-5-2-3-1 but it is not *induced subgraph*. However, by deleting node 1 we get an *induced subgraph*2-3-4-5-2 with 4 nodes and 4 edges which creates an *induced 4-TSP*. Integer programming formulation of subgraph isomorphism and induced subgraph isomorphism are presented in Part II.

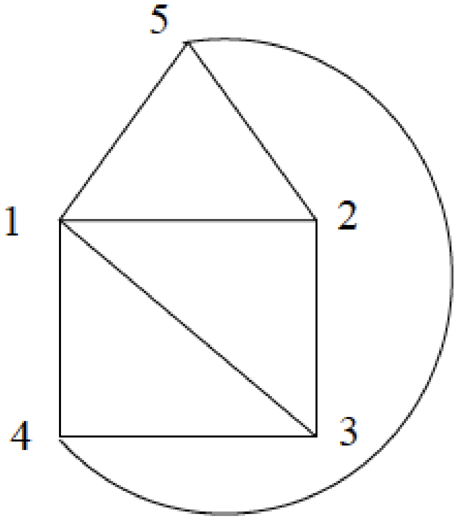

Figure 2: Illustration of subgraph isomorphism (SI) and induced subgraph isomorphism (ISI).

While in some problems, we look for a *feasible* solution, in other problems, we may need to optimize an objective function. The objective functions are in one of the formsP1-P7 given in Table **1**. In the table, objective functions are in a *minimization* format. However, they need to be appropriately modified when formulating a specific problem. Objective functions P1-P7 cover a wide range of combinatorial problems. In the next section, we present several general matching problems where focus is on undirected graphs, but the formulas can easily be modified for directed graphs.

Based on the aforementioned discussion, we present three general but related problems, outputs 1-3. Output 1 is a feasibility problem. Both outputs 2 and 3, optimizes an objective function while satisfying 'some form of feasibility' introduced by output 1. The feasibility in output 1 ensures that nodes of two graphs are assigned to each other while cost of communication between edges of the two graphs must be limited to a pre-specified set of numbers. Output 2 optimizes one of the objective functions P1-P7 while feasibility in output 1 is satisfied. Output 3 optimizes sum of penalties for violating feasibility conditions of output 1. For each output, natural IP formulation is presented. In each case, proofs are available. Due to saving space we do not present the proofs here, however, upon request the proofs are available from the authors. These outputs provide a general framework that within which many problems including those listed in 1-18 with appropriate modifications of the formulations can be considered as special cases of one of these three outputs. IP formulations for problems 1-12 are presented in Part II, while IP formulations for problems 13-18 are presented in Part I.

Table 1: General Objective Functions Used in the Problems

$$P1: \ Min_f \left\{ Max_{u\in V, v\in N(u)} \left\{ c(u,f(u)) + d\left(\{u,v\},\{f(u),f(v)\} \right) \right\} \right\},$$

$$P2: \ Min_f \left\{ \sum_{u\in V} c(u,f(u)) + \sum_{\{u,v\}\in E} d\left(\{u,v\},\{f(u),f(v)\} \right) \right\},$$

$$P3: \ Min_f \left\{ \sum_{u\in V} Max_{v\in N(u)} \left\{ c(u,f(u)) + d\left(\{u,v\},\{f(u),f(v)\} \right) \right\} \right\},$$

$$P4: \ Min_f \left\{ Max_{u'\in V'} \left\{ \sum_{u\in V: f(u)=u'} c(u,f(u)) + \sum_{\substack{\{u,v\}\in E:(f(u)=u' \text{ or} f(v)=u') \\ \text{and} \{f(u),f(v)\}\in E'}} d\left(\{u,v\},\{f(u),f(v)\} \right) \right\} \right\},$$

$$P5: \ Min_f \left\{ \sum_{u'\in V'} Max_{u\in V: f(u)=u'} \left\{ c(u,f(u)) + \sum_{v\in N(u)} d\left(\{u,v\},\{f(u),f(v)\} \right) \right\} \right\},$$

$$P6: \ Min_f \left\{ Max_{u\in V} \left\{ c(u,f(u)) + \sum_{v\in N(u)} d\left(\{u,v\},\{f(u),f(v)\} \right) \right\} \right\},$$

$$P7: \ Min_f \left\{ Max_{i'\in V'} \left\{ \sum_{\{u,v\}\in E, f(u)\le i' < f(v)} \left\{ c(u,f(u)) + c(v,f(v)) + d\left(\{u,v\},\{f(u),f(v)\} \right) \right\} \right\} \right\}.$$

2.1. Three General Matching Problems

Input: Undirected graphs *G=(V,E),* and *G'=(V',E')* with possible self-loop for each vertex in *V'*. A set of *allowable* assignments, $L_u \in V'$, for each $u \in V$. A set of integers, t_e, (possibly empty) for each edge $e \in E$, called the *forbidden set*. A penalty p_e^τ associated with each element $\tau \in t_e$. A cost of *c(u,u')* for assigning a node $u \in V$ to a node $u' \in V'$. A cost of *d(e,e')* for assigning an edge $e \in E$ to an edge $e' \in E'$. In each output 1-3 below, the task is to find a function $f : V \to V'$ such that $f(u) \in L_u$ for $u \in V$, and $\{f(u), f(v)\} \in E'$ whenever $\{u,v\} \in E$,and

Output 1: $d(\{u,v,\},\{f(u),f(v)\}) \notin t_{\{u,v\}}$.

Output 2: $d(\{u,v,\},\{f(u),f(v)\}) \notin t_{\{u,v\}}$, to minimize one of the objectives P1-P7.

Output 3: if $d(\{u,v,\},\{f(u),f(v)\}) = \tau \in t_{\{u,v\}}$ there is a penalty $p_{\{u,v\}}^\tau$. The task is to minimize total penalty.

In the following we will present IP formulation for each output 1-3. We will point out problems from the list that are special cases of the outputs. However, the IP formulations for outputs 1-3 need to be appropriately modified to represent these special cases, presented in Part II.

2.1.1. Formulations for Output 1

The matching problem with output 1 is known as the T-coloring problem [47,48]. An IP formulation of the output 1 is as follows.

$$\sum\nolimits_{u'\in L_u} x_{uu'} = 1, \text{ for } u \in V, \tag{1}$$

$$x_{uu'} + x_{vv'} \le 1, x_{uv'} + x_{vu'} \le 1, \text{ for } \{u,v\} \in E, \\ \text{and } \{u',v'\} \in \overline{E}', u' \ne v', \tag{2}$$

$$x_{uu'} + x_{vv'} \le 1, x_{uv'} + x_{vu'} \le 1, \text{ for } \{u,v\} \in E, \\ \text{and } \{u',v'\} \in E', \text{ and } d(\{u,v\},\{u',v'\}) \in t_{\{u,v\}}. \tag{3}$$

Constraints (1)e nsure node assignments are from allowable sets. Constraints (2) ensure that end points of each edge $\{u,v\} \in E$ can only be assigned to end points of an edge from *E'*. Here, for a node in graph *G'* self-loop is allowed. Thus endpoints of an edge in *E* can be assigned to one node in *V'*. Constraints (3) forbid an edge *e* in *E* be assigned to an edge *e'* in *E'* if communication cost *d(e,e')*is in the set t_e.

Modification of these formulations for directed graphs is given below. In that, constraint (1) has the same effect as was explained. Since a complement of a directed graph is a graph with edges, thus constraints (4) include edges instead of arcs in the complement of graph *A'*. Constraints (4) ensure that end points of each arc $(u,v) \in A$ can only be assigned to end points of an arc from *A'*. Constraints (5) ensure that arcs are correctly assigned to each other. In that, if two arcs *(u,v)* and *(u',v')* are assigned to each other, we must have *u'=f(u)* and *v'=f(v).* Constraints (6) forbid two arcs *a* and *a'* be assigned to each other, if the communication cost *d(a,a')*is in the set t_a.

$$\sum\nolimits_{u'\in L_u} x_{uu'} = 1, \text{ for } u \in V, \tag{1}$$

$$x_{uu'} + x_{vv'} \le 1, x_{uv'} + x_{vu'} \le 1, \text{ for } (u,v) \in A, \\ \text{and } \{u',v'\} \in \overline{A}', \tag{4}$$

$$x_{uv'} + x_{vu'} \le 1, \text{ for } (u,v) \in A, \text{ and } (u',v') \in A', \tag{5}$$

$$x_{uu'} + x_{vv'} \le 1, \text{ for } (u,v) \in A, \text{ and } (u',v') \in A', \\ \text{and } d((u,v),(u',v')) \in t_{(u,v)}. \tag{6}$$

2.1.2 Formulations for Output 2

2.1.2.1. Output 2 and Objective Function P1

The matching problem with output 2 and the objective function P1 is a bottleneck minimization that includes as special case: Bottleneck *k*-TSP, bottleneck *induced k-TSP*, Bandwidth minimization, several variations of graph coloring, graph labeling, and the Golomb Ruler problems. An IP formulation of the problem can be stated as follows. Note that the formulations need to be appropriately modified for special cases. Formulations for these special cases are presented in Part II.

Min K, s.t. (1-3), $K \ge 0$, and

$$x_{uu'} + x_{vv'} \le y_{\{u,v\}\{u',v'\}} + 1, \quad x_{uv'} + x_{vu'} \le y_{\{u,v\}\{u',v'\}} \\ +1, \text{ for } \{u,v\} \in E, \text{ and } \{u',v'\} \in E', \tag{7}$$

$$\sum\nolimits_{u'\in L_u} c(u,u')x_{uu'} + \sum\nolimits_{u'\in L_u} \sum\nolimits_{v'\in L_v \cap N[u']} d(\{u,v\}\{u',v'\})y_{\{u,v\}\{u',v'\}} \\ \le K, \text{ for } u \in V, v \in N(u). \tag{8}$$

Constraints (1-3) ensure feasibility of output 1 is satisfied. Constraints (7) ensure that $y_{ee'} = 1$ if and only if two edges *e* and *e'* are assigned to each other which adds a value of *d(e,e')* to the LHS of (8). If a node $u \in E$ is assigned to a node $u' \in E'$, and if a node $v \in N(u)$ is assigned to $v' \in N[u']$, i.e., a neighboring edge of *u* assigned to a neighbor edge of *u'* then it can add a cost of *c(u,u')+d({u,v},{u',v'})* to the objective function. Since we minimize *K*, the largest cost over all nodes $u \in V$ is being minimized.

Example 2 illustrates the formulation when used in bottleneck *k-TSP*, and bottleneck *induced k-TSP*.

Example 2. Given graph *G=(V,E),* the problem is to find (1) an *induced k-TSP*, and (2) a *k-TSP*, to minimizes the longest distance between any two cities, where $w_{\{u,v\}}$ is a given real number representing distance between two cities *u* and *v*.

In order to give IP formulation, we create a graph *G'=(V',E')* with |*V'*|=*k* nodes similar to graph in Figure **1** without arrows. Then we want to find: (1) an ISI of *G* and *G'* to minimize longest distance between every two cities, and (2) an SI of *G* and *G'* to minimize longest distance between every two cities. Let *c(u,u')=0* for all $u \in V$ and $u' \in V'$. Also let distance between every two cities with an edge in *G'* be equal to one. Define $d(e,e') = w_e w_{e'}$ for all $e \in E$ and $e' \in E'$. In order to give IP for *induced k-TSP*, we need to find subsets $S \subseteq V$ and $L \subseteq E$ such that |*S*|=*k,* and |*L*|=|*E'*|, and a one-to-one function $f : S \to V'$ satisfying $\{u',v'\} \in E'$ if and only if $\{f(u'), f(v')\} \in E$ to minimize longest distance between every two cities in *S.* Here, clearly the objective function is a special case of P1, as follows.

$$P1: \quad Min_f \left\{ Max_{u\in V, v\in N(u)} \left\{ d\left(\{u,v\},\{f(u),f(v)\}\right) \right\} \right\} = \\ Min_f \left\{ Max_{e\in E, e'\in E'} \left\{ d\left(e,e'\right) \right\} \right\}.$$

A modification of the above IP for this special case is as follows. We need constraints (9) to create a subset *S* and a one-to-one function between nodes of *V'* and *S*.

$$\sum\nolimits_{u'\in V'} x_{uu'} \le 1, \ \text{for}\ u \in V, \quad \sum\nolimits_{u\in V} x_{uu'} = 1, \ \text{for}\ u' \in V'. \tag{9}$$

Furthermore, to have *induced subgraph isomorphism* we need constraints (10-11).

$$x_{uu'} + x_{vv'} \le 1, \ x_{uv'} + x_{vu'} \le 1, \ \text{for}\ \{u,v\} \\ \in E, \ \text{and}\ \{u',v'\} \in \overline{E}', \tag{10}$$

$$x_{u'u} + x_{v'v} \le 1, \ x_{v'u} + x_{u'v} \le 1, \ \text{for}\ \{u,v\} \\ \in \overline{E}, \ \text{and}\ \{u',v'\} \in E'. \tag{11}$$

If an edge *{u,v}* is assigned to an edge *{u',v'}* then we have $y_{\{u,v\}\{u',v'\}} = 1$ ensured by constraints (7).

$$x_{uu'} + x_{vv'} \le y_{\{u,v\}\{u',v'\}} + 1, \quad x_{uv'} + x_{vu'} \le y_{\{u,v\}\{u',v'\}} \\ +1, \ \text{for}\ \{u,v\} \in E, \ \text{and}\ \{u',v'\} \in E'. \tag{7}$$

For a given node $u \in V$, consider a neighbor node $v \in V$ (i.e., there is an edge $e = \{u,v\} \in E$). If the edge *e* is assigned to an edge *e'={u',v'}* then we have $d(e,e') = w_e w_{e'} = w_e$. The objective is to minimize the maximum of *d(e,e')* for all $e \in E$ and $e' \in E'$ when assigned to each other, this is done by adding an objective function, *Min K*, and constraints (12).

$$d(\{u,v\}\{u',v'\})y_{\{u,v\}\{u',v'\}} \le K, \quad d(\{u,v\}\{v',u'\})y_{\{u,v\}\{u',v'\}} \\ \le K, \ \text{for}\ \{u,v\} \in E, \ \text{and}\ \{u',v'\} \in E'. \tag{12}$$

Constraints (12) can also be replaced by constraints (13) that are the same as constraints (8). Note that in (13), given an edge *{u,v},*only if it is assigned to an edge *{u',v'}* then we have $y_{\{u,v\}\{u',v'\}} = 1$. Here, given a node *u* and one of its neighbors $v \in N(u)$ (i.e., there is an edge $e = \{u,v\} \in E$), since we already have ensured a one-to-one correspondence between *S* and *V'* by (9) then the left hand side (LHS) of the (13) will be exactly equal to the LHS of (12).

$$\sum\nolimits_{u'\in L_u} \sum\nolimits_{v'\in L_v \cap N[u']} d(\{u,v\}\{u',v'\})y_{\{u,v\}\{u',v'\}} \\ \le K, \ \text{for}\ u \in V, v \in N(u). \tag{13}$$

Now, to appropriately modifying the formulations for *k-TSP*, we only need to remove the constraints (10) from the set of constraints presented for *induced k-TSP*. Note that, different but simpler formulations for *k-TSP* and induced *k-TSP* are presented in Part II.

2.1.2.2. Output 2 and Objective Function P2

The matching problem with output 2 and objective function P2includes as special case: *k*-TSP, *induced k-TSP*, variations of coloring problems, linear arrangement, variations of metric labeling, the largest common subgraph problem, maximum subgraph matching, facility layout, and contact map problem. An IP formulation is as follows. Similar to the case for P1, the formulations for special cases need to be appropriately modified. They are presented in Part II.

$$Min \ \sum\nolimits_{u\in V} \sum\nolimits_{u'\in L_u} c(u,u')x_{uu'} \\ +\frac{1}{2}\sum\nolimits_{u\in V} \sum\nolimits_{v\in N(u)} \sum\nolimits_{u'\in L_u} \sum\nolimits_{v'\in L_v \cap N[u']} d(\{u,v\}\{u',v'\})y_{\{u,v\}\{u',v'\}} \tag{14}$$

s.t. (1-3),(7).

Given a matching between the two graphs the objective function P2 sums up all node assignments and all edge assignments costs and minimizes the total quantity, constraints (1-3) ensure feasibility of assignments. Constraints (7) ensure that communication costs between edges are added to total cost. Note that the objective function (14) has similarity with constraints (8). In constraints (8), for a given node *u* and one of its neighbors *v*(i.e., $e = \{u,v\} \in E$), if assigned respectively to *u'* and *v'*(i.e., $e' = \{u',v'\} \in E'$), then we considered the total cost of *c(u,u')*+*d(e,e')*, and minimize largest of such cost. However, in (14) sum of all such costs is being minimized. Example 3 illustrates the formulation when used in *k-TSP*, and *induced k-TSP*.

Example 3. The problem is similar to problem of example 2, however, the objective is minimization of the total distance travelled.

Here, parameters are defined similar to example 2. However, we minimize the following quantity for all *e* and *e'* assigned to each other.

$$\sum\nolimits_{e\in E} \sum\nolimits_{e'\in E'} d(e,e'). \tag{15}$$

This is exactly the same as P2 given as follows.

$$P2: \ Min_f \left\{ \sum\nolimits_{\{u,v\}\in E} d\big(\{u,v\},\{f(u),f(v)\}\big) \right\}.$$

An IP formulation for *induced k-TSP* is thus as follows.

$$Min \ \sum\nolimits_{e\in E} \sum\nolimits_{e'\in E'} d(e,e')y_{ee'} = \\ \frac{1}{2}\sum\nolimits_{u\in V} \sum\nolimits_{v\in N(u)} \sum\nolimits_{u'\in V'} \sum\nolimits_{v'\in N(u')} d(\{u,v\}\{u',v'\})y_{\{u,v\}\{u',v'\}} \tag{14}$$

s.t. (7),(9-11).

An IP formulation for *k-TSP* is similar except that constraints (10) are removed. Note that, here also different but simpler formulations for *k-TSP* and induced *k-TSP* are presented in Part II.

2.1.2.3. Output 2 and Objective Function P3

The matching problem with output 2 and objective function P3 includes profile minimization problem (PMP) as special case. The PMP is equivalent to the interval graph completion problem, which is to find a super-graph of a graph *G* with as small a number of edges as possible. Given an undirected graph *G=(V,E),* the PMP looks for a one-to-one function $f:V \to \{1,\ldots,|V|\}$ such that $\sum_{u\in V} Max_{v\in N[u]}\{f(u)-f(v)\}$ is minimized. The objective function in PMP is a special case of P3. Example 4 below illustrates this point.

Example 4. Define a complete undirected *G'=(V',E')* with *|V'|={1,…,|V|}*. For each $u\in V$ let $c(u,f(u))=f(u)$, and for each $\{u,v\}\in E$ let $d(\{u,v\},\{f(u),f(v)\})=-Min[f(u),f(v)]$. The objective of PMP is now equal to P3 given as

$$P3:\ Min_f\left\{\sum_{u\in V} Max_{v\in N(u)}\left\{c(u,f(u))+d\left(\{u,v\},\{f(u),f(v)\}\right)\right\}\right\}.$$

IP formulation for PMP is presented in Part II. An IP formulation of output 2 with the objective function P3 is given below. Note that, the objective function P3 has similarity with P1. Given a node $u\in V$ the maximization part of P3 (the quantity in the summation) is the same as maximization in P1. However, P3 sums up such quantity over all nodes *u*.

$Min\ \sum_{u\in V} K_u$, s.t. (1-3),(7), $K_u \geq 0$, and

$$\sum_{u'\in L_u} c(u,u')x_{uu'} + \sum_{u'\in L_u}\sum_{v'\in L_v\cap N[u']} d(\{u,v\}\{u',v'\})y_{\{u,v\}\{u',v'\}} \qquad (16)$$
$$\leq K_u,\ \text{ for } u\in V, v\in N(u).$$

Constraints (1-3) and (7) have the same effect as was explained earlier. For a given node $u\in V$, the LHS in(16)is the same as constraints (8), however, the objective function minimizes sum of all such quantities.

2.1.2.4. Output 2 and Objective Function P4

The matching problem with output 2 and the objective function P4 includes several variations of metric labeling, and task assignment in distributed computing systems (TADCS) as special cases. Here, we are looking for a function (not necessarily one-to-one) $f:V\to V'$ between nodes of two graphs. If anode $u\in V$ is assigned to a node $u'\in V'$ there is a cost of *c(u,u').* Given a node *u'* there is a possibility that several nodes from *V* are assigned to it. If an edge $e\in E$ is assigned to an edge $e'\in E'$ then there is a cost of *d(e,e').*Thus, the total cost of assigning all nodes $u\in V$ to a given node $u'\in V'$ is equal to

$$\sum_{u\in V:f(u)=u'} c(u,f(u)) + \sum_{\substack{\{u,v\}\in E:(f(u)=u' \text{ or} f(v)=u') \\ \text{and}\{f(u),f(v)\}\in E'}} d\left(\{u,v\},\{f(u),f(v)\}\right). \quad (17)$$

The objective is to find a matching function that minimizes the maximum of (17) over all nodes $u'\in V'$ which is exactly P4. Given a node *u'*, the LHS in (18) is the total costs equivalent to (17). The reason that the triple summation is divided by 2 is that we are assuming if $u'\in L_u$ then $u'\in L_v$ for $v\in N(u)$. If this assumption is not satisfied the formulation needs some minor modification. Thus, an IP formulation is as follows.

Min K, s.t. (1-3),(7), $K\geq 0$, and

$$\sum_{u:u'\in L_u} c(u,u')x_{uu'}$$
$$+\frac{1}{2}\sum_{u:u'\in L_u}\sum_{v\in N(u)}\sum_{v'\in L_v\cap N[u']} d(\{u,v\}\{u',v'\})y_{\{u,v\}\{u',v'\}} \qquad (18)$$
$$\leq K,\ \text{for } u'\in V'.$$

A special case of the above formulation under the *task assignment in distributed computing systems (TADCS)* is discussed in [49]. Example 5 illustrates this case.

Example 5. In the TSDCS, we have a set of tasks, *V,* and a set of processors, *V'.* A *task graph G=(V,E)* represents the set of tasks and their communication links shown by the edges of the graph. A *processor graph G'=(V',E')* represents the set of processors and their communication links shown by the edges of the graph. The problem is to find an assignment function $f:V\to V'$ that assigns the tasks to the processors. If task $u\in V$ is assigned to an eligible processor $u'\in V'$, the processing time is *c(u,u').* If two tasks $u,v\in V$ need to communicate (i.e., if there is an edge $\{u,v\}\in E$), they can be assigned to one processor; in that case, there is no communication time between them. In this case we are assuming node *u'* has a self-loop. Two tasks can also be assigned to two different processors if a communication link is possible between them (i.e., if there is an edge $\{u',v'\}\in E'$). In that case, there is a communication time between two tasks measured by *d({u,v},{f(u),f(v)}).* The problem asks for an assignment $f:V\to V'$to minimize the completion time of the last task completed, i.e., the load of the processor with the maximum processing and communication times [49], which is a special case of P4. However, if the problem is to minimize total times (see for example, [50-53] for special cases of such scheduling problem) then we have the objective P2 to be minimized. Several other specific examples of metric labeling with objective functions P2 or P4 are presented in Part II. Prominent applications are in pattern analysis, and image processing, (see for example, [28-30]).

2.1.2.5. Output 2 and Objective Function P5

The matching problem with output 2 and the objective function P5 includes several variations of graph coloring as special cases including a problem of transmission of real-time messages in a metropolitan network or a problem related to dynamic storage allocation, [54,55], and scheduling on a batch machine with job compatibilities, [55-57]. In P5, if a node *u* is assigned to a node *u'* the total costs is a sum of node assignment cost, *c(u,u')*, and edge assignment costs. It is the sum of *d({u,v},{f(u),f(v)})* for all *v* such that $\{u,v\} \in E$ and $\{f(u), f(v)\} \in E'$ which is equal to

$$c(u, f(u)) + \sum_{v \in N(u)} d(\{u,v\},\{f(u), f(v)\}). \quad (19)$$

Among all nodes *u* assigned to a given node *u'* we are looking at the maximum that is equal to

$$Max_{u \in V: f(u)=u'} \left\{ c(u, f(u)) + \sum_{v \in N(u)} d\left(\{u,v\},\{f(u), f(v)\}\right) \right\}. \quad (20)$$

The problem looks for a function $f: V \to V'$ to minimize sum of all such maximum costs over all nodes $u' \in V'$ which is exactly P5.An IP formulation of the problem can be stated as follows.

$Min \sum_{u' \in V'} K_{u'}$, s.t. (1-3),(7), $K_{u'} \geq 0$, for $u' \in V'$, and

$$c(u,u')x_{uu'} + \sum_{v \in N(u), u' \in L_v} d(\{u,v\}\{u',v'\}) y_{\{u,v\}\{u',v'\}} \leq K_{u'}, \text{ for } u' \in V', \text{ and } u \in V \text{ with } u' \in L_u. \quad (21)$$

The LHS in (21) is the cost contributed by node *u* when assigned to a node *u'* plus the cost communicated by each edge neighboring *u* assigned to each edge neighboring *u'*. Since the RHS is the same for all *u* thus the objective function P5 minimizes sum over all such values. Also, note that constraints (1-3) and (7) plays the same role as explained before.

Example 6 illustrates a task assignment problem as a special case of the above problem.

Example 6. The problem is similar to the TSDCS defined in example 5. However, for a given processor *u'* the total cost is defined by (19). Special cases of this problem is scheduling on a batch machine with job compatibilities where communication costs are zero, [55,56,57]. In this special case the problem becomes a *Weighted Vertex Coloring Problem (WVCP)* discussed in more detail in Part II. Note that in TSDCS since a task $u \in V$ must communicate with its neighbors, i.e., with each $v \in N(u)$, thus if *u* is assigned to *u'* and if task *v* is assigned to the processor *u'* there is no communication costs however if it is assigned to a processor $v' \in V'$ with $u' \neq v'$ then the communication cost is *d({u,v},{u',v'})*. Thus the total cost for assigning a task $u \in V$ to a given processor $u' \in V'$ is equal to the LHS of (20). The value of $K_{u'}$ is the largest over all tasks $u \in V$, where the sum is being minimized.

2.1.2.6. Output 2 and Objective Function P6

The matching problem with output 2 and the objective function P6 generalizes the *graph orientation (GO)* problem [58-60], and the edge covering problem [61]. Other names for GO are *edge linking* [62], *link orientation* [63], and *vertex-orderings* [64].

In GO, the problem is to determine a balanced ordering of the vertices of a graph; that is, the neighbors of each vertex *u* are as evenly distributed to the left and right of *u* as possible. The problem has applications in variety of areas including in graph drawing [64], and providing quality of service (QoS) networks [60,63]. The major objectives of various information services on modern networks are to provide high efficiency, good quality, and maximum throughput for the system resources [60,63]. The GO has been proposed for assigning flow orientations over links in a network. Refer to [60,63] for applications, proof of NP-hardness, and several polynomial solvable special cases. Refer to [61] for a recent IP formulation. Sadig, Mozafari and Kashan in [61] mention other applications including the edge covering (EC) problem. The EC problem has applications in locating emergency facilities such as police stations, and road hospitals or an electronic network for locating information center. The main characteristic of these applications is that the customer's demand distributes uniformly in the paths between the location points [61].

The GO can be defined as follows, [60,63]. Given an undirected graph *G=(V,E)*,each node *u* is associated with a weight (cost) α_u, and each edge $\{u,v\} \in E$ is associated with two weights (costs) $w_{(u,v)}$ and $w_{(v,u)}$ where (u,v) is directed from *u* to *v* (an orientation from *u* to *v*, or an out-degree node of *u*) and (v,u) is directed from *v* to *u* (an orientation from *v* to *u*, or an in-degree node of *u*). Thus, each edge is considered as two directed arcs. Given an orientation for all edges, for a node $u \in V$ defined out-degree nodes, *ON(u),* and in-degree nodes, *IN(u)*, as follows

$$ON(u) = \{v : (u,v) \text{ is an out-degree node of } u\},$$

$$IN(u) = \{v : (v,u) \text{ is an in-degree node of } u\}.$$

The problem is to find an orientation for each edge. Given an orientation for a graph, the total cost associated with each node is equal to $\alpha_u + \sum_{v \in ON(u)} w_{(u,v)}$. The GO looks for choosing an orientation for each edge to minimize the largest costs for all node $u \in V$ which is minimization of $Max_{u \in V} \{\alpha_u + \sum_{v \in ON(u)} w_{(u,v)}\}$.

Output 2 with objective function P6, generalizes the GO problem and the EC problem, can be stated as follows. We are given two undirected graphs *G=(V,E)* and *G'=(V',E')*, we want to find a function $f: V \to V'$. In that, if a node *u* is assigned to a node *u'* the cost is the same as (19). Among all nodes $u \in V$ we are looking at the maximum cost which is equal to

$$Max_{u \in V} \left\{ c(u, f(u)) + \sum_{v \in N(u)} d\left(\{u,v\},\{f(u),f(v)\}\right) \right\}. \quad (22)$$

Note that, in (22) for simplicity of presentation we did not include edge orientation, i.e., choosing a direction for each edge. To consider edge orientation minor modification in the formulation is needed. In fact, when edge orientation needs to be considered value of *d({u,v),{f(u),f(v)})*for *{u,v}* depends on directed of the edge, i.e., we have

$$d(\{u,v\},\{f(u),f(v)\}) \neq d(\{v,u\},\{f(v),f(u)\}).$$

The objective is to fund a function *f* to minimize (22) which is exactly P6. Now, an IP formulation of the problem can be stated as follows:

Min K, s.t. (1-3),(7), $K \geq 0$, and

$$\sum_{u' \in L_u} c(u,u')x_{uu'} + \sum_{u' \in L_u} \sum_{v \in N(u)} \sum_{v' \in L_v \cap N[u']} d(\{u,v\}\{u',v'\})y_{\{u,v\}\{u',v'\}} \quad (23)$$
$$\leq K, \text{ for } u \in V.$$

Constraints (1-3) and (7) has the same effect as we described before. The LHS in (23) is representing the cost of assigning a given node $u \in V$ to a node $u' \in V'$ plus cost of all edges neighboring *u,* if assigned to edges in *E'*. The objective function P6 minimizes the largest of such quantity.

2.1.2.7. Output 2 and Objective Function P7

The matching problem with output 2 and the objective function P7 includes the *Minimum Cut Linear Arrangement Problem (MCLAP)*as a special case (see [6] for recent discussion and complexity of special cases).Given an undirected graph *G=(V,E)*, MCLAP looks for smallest value of *K* and a one-to-one function $f: V \to \{1,\ldots,n\}$ such that for all $1 \leq i \leq n$, $|\{\{u,v\} \in E, f(u) \leq i < f(v)\}| \leq K$.IP formulation for MCLAP is presented in Part II. Ithas applications in telecommunication [65], computer aided design, automatic graph drawing, protein engineering, optimization of rail systems [2], and computational logic [66], among others.

In output 2 with objective function P7, we are looking for a function $f: V \to V'$. In that, given a node $i' \in V'$, if endpoints of an edge *{u,v}* in *E* are assigned to end points of an edge *{f(u),f(v)}* in *E'* where $f(u) \leq i' < f(v)$ then the cost is

$$c(u,f(u)) + c(v,f(v)) + d(\{u,v\},\{f(u),f(v)\}).$$

We want to minimize the maximum of such quantity over all nodes $i' \in V'$. An IP formulation can be stated as follows:

Min K, s.t. (1-3),(7), $K \geq 0$, and

$$\sum_{\substack{\{u,v\} \in E: \{u',v'\} \in E', \\ f(u)=u' \leq i' < f(v)=v'}} (c(u,u') + c(v,v') + d(\{u,v\},\{u',v'\}))y_{\{u,v\}\{u',v'\}} \quad (24)$$
$$\leq K, \text{ for } i' \in V'.$$

Constraints (1-3) and (7) have the same effects as described before. The left hand side of (24) represents the cost contributed by all edges $K_{\max} = n$ if *u* and *v* in *V* are respectively assigned to two nodes *u'* and *v'* in *V'* where $\{u',v'\} \in E'$ and such that $u' \leq i' < v'$ for a given node $i' \in V'$. The objective function minimizes this cost over all nodes $i' \in V'$.

2.1.3. Formulations for Output 3

The matching problem with output 3is a generalization of P2. In that, we allow edges to be assigned to each other if the cost of an assignment is in the *forbidden set.* However, this is done with the expense of a penalty, and the task is to minimize the total penalty. It includes the *Fixed Spectrum Frequency Assignment* (FSFA), [67-70], as a special case. Montemanni, Smith and Allen in [68,70] presented an IP formulation of FSFA; however, our formulation given below for the general case when applied to FSFA is different, and is simpler with fewer variables.

$$Min \sum_{u \in V} \sum_{v \in N(u)} \sum_{u' \in L_u} \sum_{v' \in L_v \cap N[u']} \sum_{\tau \in t_{\{u,v\}}} p^{\tau}_{\{u,v\}} y^{\tau}_{\{u,v\}\{u',v'\}}$$

s.t. (1-2), and

$$x_{uu'} + x_{vv'} \leq y^{\tau}_{\{u,v\}\{u',v'\}} + 1, \; x_{uv'} + x_{vu'} \leq y^{\tau}_{\{u,v\}\{u',v'\}} + 1,$$

for

$$\{u,v\} \in E, \{u',v'\} \in E', d(\{u,v\},\{u',v'\}) = \tau \in t_{\{u,v\}}. \quad (25)$$

Constraints (1) and (2) ensure a one-to-one assignment between nodes of *V* and *V'*. Constraints (25) ensure that if two edges are assigned to each other and communication costs is in the forbidden set, then $y^{\tau}_{\{u,v\}\{u',v'\}} = 1$. This adds a penalty to the objective function to be minimized.

Example 7 illustrates the FSFA problem as a special case. The FSFA problem involves the assignment of discrete channels (or frequencies) to the transmitters of a radio network, such as a mobile telephone network. In that assignment, frequency separation is necessary to avoid interference by other transmitters to the signal received from the wanted transmitter at the reception points [68,69,70].

Example 7. The FSFA problem can be represented by a weighted undirected graph *G=(V,E)*. The problem looks for a mapping $f : V \to F$ where each $u \in V$ represents a transmitter of a radio network, and *F={0,1,…,*|*F*|*-1}* is a set of consecutive frequencies available for every vertex (transmitter) in *V*. The set of edges *E* represents those pairs of transmitters for which the assigned frequencies are constrained. The constraint arises because one of the transmitters can interfere with signals received from the other. For each $\{u,v\} \in E$ there is $t_{\{u,v\}} \in N^{+}$. The value of $t_{\{u,v\}}$ represents the highest separation between the frequencies assigned to the transmitter *u* and the one assigned to *v* that generates unacceptable interference. If we indicate with *f(u)* the frequency assigned to transmitter *u*, then if $| f(u) - f(v) | > t_{\{u,v\}}$, the interference involving the two transmitters is acceptable. The larger $| f(u) - f(v) |$ is the better the assignment, i.e., less interference. If however, $| f(u) - f(v) | \leq t_{\{u,v\}}$ there is a cost (penalty) equal to $p_{\{u,v\}} \in N^{+}$. The objective is to find an assignment *f* that minimizes the sum $p_{\{u,v\}}$ over all pairs $\{u,v\} \in E$ for which $| f(u) - f(v) | \leq t_{\{u,v\}}$. It is clear that FSFA is a special case of the above formulation.

3. RELATIONSHIP WITH QUADRATIC ASSIGNMENT PROBLEM

The Quadratic assignment problem (QAP) is one of the most celebrated combinatorial problems with enormous applications [71]. There is a close relationship between the problems studied in this paper and QAP. Several variations of the QAP have been reported in the literature, (see [72] for a comprehensive survey). Those are the QAP minimizing total costs (QAP), the quadratic bottleneck assignment problem (QBAP), the quadratic semi-assignment problem (QSAP), the biquadratic assignment problem (BiQAP), the quadratic 3-dimensional assignment problem (Q3AP), and the multiobjective QAP (mQAP). The QAP and QSAP are special cases of the problem with output 2 and the objective function P2. The QBAP is a special case of the graph matching problem with output 2 and objective function P1. The BiQAP (also known as the *quartic assignment problem* [73,74]), and Q3AP are generalizations of QAP where the interaction is among several nodes of the graph, [72]. To save space, we will not go further into details of these problems. However, we discuss QAP, and BQAP in context of matching two graphs. In all cases proofs are available. Due to saving space we do not present the proofs here, however, upon request the proofs are available from the authors. QSAP is a straightforward variation of QAP.

The next section discusses situations in which there are interactions between several nodes of graphs. Proofs are available upon request from the authors. Then in Part II, specific problems will be discussed. The classic book on complexity theory by Garey and Johnson [75] is a major source for definitions and understanding of these and other related problems that we use. For ease of presentation, if applicable, we refer to the *'problem index number'* addressed in [75].

4. GENERALIZATION TO MULTI-NODE-INTERACTION, AND MULTI-LAYER GRAPH MATCHING PROBLEMS

In the problems that we have discussed so far, interactions could occur only between two nodes of a graph. However, in many realistic situations interactions among more than two nodes in the graph can occur, [73,74]. Furthermore, there are real problems, e.g., multi-layer crossing minimization problem [32], where the nodes of a graph are layered and thus the solution looks for several matching functions at the same time where the solution of each matching affects the whole configuration, [76]. Below we first consider the *interval graph completion (IGC) problem*, for which Even, Nao, Rao and Schieber in [28] presented an *indirect* IP formulation. It has applications in VLSI gate layout problems, [77]. A *natural* IP formulation has been proposed in [78]. However, we propose a simpler *natural* IP formulation.

(GT35) Interval Graph Completion (IGC):

Input: A connected undirected graph *G=(V,E).*

Output: A minimum cardinality set of edges *F* such that $G = (V, E_S)$ is an interval graph and $E_S = E \cup F$.

A graph is an interval graph if and only if there is a linear ordering of the vertices such that if vertex *u* with index *u'* has an edge to vertex *v* with index *v',* where *u'<v',* then every vertex whose index is between *u'* and *v'* also has an edge to vertex *v*, [28]. Define a complete undirected graph *G=(V',E')* where |*V*|=|*V'*| with nodes numbered *1,2,…,*|*V*|. The IGC problem is to find a one-to-one assignment, $f: V \rightarrow V'$ where the condition of an interval graph is satisfied. Below we give an IP formulation of the problem. Here, $y_e = 1$ if an edge $e \in \overline{E}$ is not among minimum cardinality set *F*, and zero otherwise.

$$Min \;\sum\nolimits_{e \in \overline{E}} y_e,$$

s.t.

$$\sum\nolimits_{u' \in V'} x_{uu'} = 1, \text{ for } u \in V, \quad \sum\nolimits_{u \in V} x_{uu'} = 1, \text{ for } u' \in V', \tag{26}$$

$$\begin{gathered} x_{uu'} + x_{vv'} + x_{zz'} \le y_{\{z,v\}} + 2, \text{ for } \{u,v\} \\ \in E, \text{ and } u' < z' < v' \text{ and } \{z,v\} \in \overline{E}, \end{gathered} \tag{27}$$

Constraints (26) ensure a one-to-one assignment between *V* and *V'*. Constraints (27) ensure that for an edge *{u,v}* where *u* is assigned to *u'* and *v* is assigned to *v'* with *u'<v'*, if a node *z* is assigned to a node *z'* with *u'<z'<v'* and if there is no an edge *{z,v}* in *E* then the value of $y_{\{z,v\}}$ must be equal to 1 and thus edge *{z,v}* is added to the set *F*. The objective function minimizes cardinality of the set *F*. The IGC is equivalent to the *profile minimization problem (PMP),* [79]. The PMP is discussed in Part II.

Now, we discuss the problem of *multi-layer crossing minimization* that has many applications including combinatorial geometry, theory of VSLI, and graph drawing, and computational biology [25,80,81,82,83,84]. Nollenberg and Wolff in [85] have recently presented an IP formulation for a special case of this problem.

Given a directed graph *G=(V,A)* the set of nodes *V* is partitioned in to *p* disjoint sets, $V_1, \ldots, V_p$, where $V_l \cap V_k \neq \varnothing$, for *l,k=1,…,p,* $l \neq k$ and $V = \bigcup_{l=1}^{p} V_l$ with $|V_l| = n_l$. Define an arc $(u,v) \in A$ for $u \in V_l$ and $v \in V_{l+1}$, for some *l=1,…,p-1*(where each set of nodes V_l is placed on a straight line, i.e., layer *l*).We look for a permutation of vertices V_l in each layer *l* such that the number of crossing arcs is minimized. Here, the set of arcs is partitioned into $A_1, \ldots, A_{p-1}$ where A_l is the set of arcs between nodes V_l and V_{l+1} for *l=1,…,p-1.*

Multi-layer Crossing Minimization (MLCM):

Input: A multi-layered graph *G=(V,A)* where the set of nodes *V* is partitioned into *p* disjoint sets, $V_1, \ldots, V_p$, for *l=1,…,p.*

Output: A permutation of vertices V_l in each layer *l* such that the number of crossing arcs is minimized.

If *p=2* the problem is known as 2-layer crossing minimization, which is NP-hard [86]. Several IP formulations of the 2-layer problem have been provided, [37,87]. The MLCM problem is NP-hard even if permutation of one layer is fixed, known as *one-sided* crossing minimization [88]. Recently, several other special cases of the MLCM problem have shown to be NP-hard, [89,90]. In most heuristics and/or exact algorithms, the 2-layer version of the problem is extensively used with or without permutation of one side fixed, [32]. Carpano, [91] stated that the difficulty with dealing with the general MLCM involves not only in finding the best permutation for each layer with respect to its adjacent layers but also the effect of this choice on the configuration as a whole. In that spirit, an IP formulation was presented in [32].

Define a one-to-one function $f_l : V_l \rightarrow \{1, \ldots, n_l\}$ for each layer *l*. Let $x_{ui}^l = 1$ if node $u \in V_l$ is assigned to $i \in \{1, \ldots, n_l\}$, and 0 otherwise. Two arcs $(u,u'),(v,v') \in A_l$ (for $u,v \in V_l$, $u',v' \in V_{l+1}$) cross iff either $f_l(u) < f_l(v)$ and $f_{l+1}(v') < f_{l+1}(u')$ or $f_l(v) < f_l(u)$ and $f_{l+1}(u') < f_{l+1}(v')$. Let $y_{(u,u')(v,v')}^l = 1$ iff two arcs *(u,u')* and *(v,v')* in layer *l* cross, and 0 otherwise. Our new IP formulation of the problem is as follows.

$$Min \;\sum\nolimits_{i=1}^{p-1} \sum\nolimits_{(u,u'),(v,v') \in A_l} y_{(u,u')(v,v')}^l$$

$$\begin{gathered} s.t. \;\sum\nolimits_{i=1}^{n_l} x_{ui}^l = 1, \text{ for } l \in \{1, \ldots, p), u \in V_l, \\ \sum\nolimits_{u \in V_l} x_{ui}^l = 1, \text{ for } l \in \{1, \ldots, p), i \in \{1, \ldots, n_l), \end{gathered} \tag{28}$$

$$x_{ui}^l + x_{vj}^l + x_{v'i'}^{l+1} + x_{u'j'}^{l+1} \le y_{(u,u')(v,v')}^l + 3,$$

$$\text{for } l \in \{1, \ldots, p-1\}, (u,u'),(v,v') \in A_l,$$

$$\text{for } i,j \in \{1, \ldots, n_l\}, \text{and } i',j' \in \{1, \ldots, n_{l+1}\}, \text{ and } i < j, i' < j', \tag{29}$$

$$x_{uj}^l + x_{vi}^l + x_{u'i'}^{l+1} + x_{v'j'}^{l+1} \le y_{(u,u')(v,v')}^l + 3,$$

$$\text{for } l \in \{1, \ldots, p-1\}, (u,u'),(v,v') \in A_l,$$

for $i, j \in \{1,\ldots,n_l\}$, and $i', j' \in \{1,\ldots,n_{l+1}\}$, and $i < j, i' < j'$. (30)

Constraints (28) represent a one-to-one assignment between set of layered nodes V_l and $\{1,\ldots,n_l\}$, *l=1,…,p.* Constraints (29-30) ensure a penalty value of 1 is added to the objective function if two edges cross. The objective function therefore minimizes the sum of all crossings.

PART II

Graph theoretic approaches for solving difficult problems have been an important practical and theoretical research area for several decades. Graphs have been applied to almost every discipline of sciences, engineering, and social sciences. Some of the major applications in graph matching are in pattern analysis, social network analysis, homeland security, telecommunications, and manufacturing [10, 27,29,49,53,92-102]. In Part II we present IP formulations for a wide range of difficult problems, problems 1-12 from the list we presented earlier, using the framework presented in the Part I. For some of these problems we have not seen any *natural* IP formulations in the literature and for some others our formulations provide alternative approaches.

5. INTEGER PROGRAMMING FORMULATIONS OF GRAPH PROBLEMS

This section presents *natural* integer programming formulation for problems 1-12 in the list that we presented earlier (refer to the survey papers [2,5,92] for comprehensive discussion and applications). Also, the classic book on complexity theory by Garey and Johnson [75] is a major source for definitions and understanding of these and other related graph problems. We may discuss the *recognition* (*feasibility*) and/or the *optimization* problems separately based on appropriateness and availability of limited space. For simplicity, formulas in Table **2** are used over-and-over when we present the formulations. To save space we do not present proofs here, however, upon request the proofs are available from the authors.

We start the discussion with some variations of the TSP. For a comprehensive survey of variants of TSP refer to [103]. We discuss a variation, known as *k*-TSP, that includes TSP and the Bottleneck TSP as special

Table 2: **Assignment Formulas Used in Formulating Problems in Part II**

$$(a1)\quad \sum_{u'\in V'} x_{uu'} = 1, \ \text{for } u \in V, \qquad\qquad (a2)\quad \sum_{u\in V} x_{uu'} = 1, \ \text{for } u' \in V',$$

$$(b1)\quad \sum_{u'\in V'} x_{uu'} \le 1, \ \text{for } u \in V, \qquad\qquad (b2)\quad \sum_{u\in V} x_{uu'} \le 1, \ \text{for } u' \in V',$$

$$(c1)\quad x_{uu'} + x_{vv'} \le 1,\ x_{uv'} + x_{vu'} \le 1, \ \text{for } \{u,v\} \in E, \text{ and } \{u',v'\} \in \overline{E}', \qquad (c2)\quad x_{u'u} + x_{v'v} \le 1,\ x_{v'u} + x_{u'v} \le 1, \ \text{for } \{u,v\} \in \overline{E}, \text{ and } \{u',v'\} \in E',$$

$$(d1)\quad x_{uu'} + x_{vv'} \le 1,\ x_{uv'} + x_{vu'} \le 1, \ \text{for } (u,v) \in A, \text{ and } (u',v'),(v',u') \notin A', \qquad (d2)\quad x_{uu'} + x_{vv'} \le 1,\ x_{uv'} + x_{vu'} \le 1, \ \text{for } (u',v') \in A', \text{ and } (u,v),(v,u) \notin A$$

$$(e)\quad x_{uv'} + x_{vu'} \le 1 \ \text{for } (u,v) \in A, \text{ and } (u',v') \in A',$$

$$(f)\quad x_{uu'} + x_{vv'} \le y_{\{u,v\}\{u',v'\}} + 1 \ \ x_{uv'} + x_{vu'} \le y_{\{u,v\}\{u',v'\}} + 1, \ \text{for } \{u,v\} \in E, \text{ and } \{u',v'\} \in E',$$

$$(g)\quad x_{uu'} + x_{vv'} \le y_{(u,v)(u',v')} + 1, \ \text{for } (u,v) \in A, \text{ and } (u',v') \in A',$$

$$(h1)\quad x_{uu'} + x_{uv'} \le 1, \ \text{for } u \in V, \text{ and } \{u',v'\} \in E', \qquad\qquad (h2)\quad x_{uu'} + x_{uv'} \le 1, \ \text{for } u \in V, \text{ and } (u',v') \in A',$$

$$(i1)\quad x_{uu'} + x_{vu'} \le 1, \ \text{for } \{u,v\} \in E, \text{ and } u' \in V', \qquad\qquad (i2)\quad x_{uu'} + x_{vu'} \le 1, \ \text{for } (u,v) \in A, \text{ and } u' \in V'.$$

cases, [42,43]. We discuss the *k*-TSP for directed graphs. Formulations can be easily modified for undirected graphs. Our formulation here is different and more general than those in the literature, (see [7,46] for classification of TSP formulations). Also note that in Part I, we gave several new IP formulation of *k-TSP* and *induced k-TSP* for undirected graphs. Similar formulations can be given for directed graphs, however, here in Part II we present different formulas for these problems. To save space our concentration will be on *k-TSP* for directed graphs. With minor modifications the formulation can be extended to *induced k-TSP*.

5.1. Variations of TSP

(ND22, ND23, ND24) Some Variations of TSP:

Input: Directed graph *G=(V,A),* distance $w_{(u,v)}$ for arc $(u,v) \in A$, and positive integers $k \le n$, and*K.*

Output A: Is there a circuit of *G* with *k* nodes?

Output B: Is there a circuit of *G* with *k* nodes with total traveled distance of*K* or less?

Output C: Is there a circuit of *G* with *k* nodes with the longest distance between any two cities less than or equal to *K*?

We first define a *circle graph G'=(V',A')* with *k* nodes similar to Figure **1**. In that, for *u'=1,…,k-1* we have $(k,1),(u',u'+1) \in A'$. The problem looks for an assignment $f : V \to V'$, where each element of *V* may be assigned to an element of *V',* and each element of *V'* must be assigned to an element of *V.* This is a *subgraph isomorphism problem.* If *k=n,* we have the Hamiltonian circuit, i.e., the TSP, and the Bottleneck TSP, respectively, (see [104,105] for recent optimal procedures of TSP and Bottleneck TSP). Formulation of the problem for Output A can be stated as follows:

(a2),(b1), and $x_{uu'} \le \sum_{v\in N(u)} x_{v(u'+1)}$, for $u \in V, u' \in V'\setminus\{k\}$, and $x_{uk} \le \sum_{v\in N(u)} x_{v1}$, for $u \in V$.

Constraints (a2) and (b1), taken from Table **2**, ensure that a node in *V* may be assigned to anode in *V'* and a node in *V'* must be assigned to exactly one node in *V.* The first set of inequalities ensure that if a node *u* is assigned to any node *u'*, except node *k*, in *V'* then a node *v* where *(u,v)* is an arc in *G* must be assigned to node *u'+1*. The last set of inequalities endure that for a node *u* in *V* assigned to node *k* in *V'*, there must be a node *v* in *V* such that *(u,v)* is an arc in *G* and *v* is assigned to node 1.If there is a solution to this formulation then we have found a circuit of *k* nodes in graph *G*, otherwise no such circuit exists in the graph *G*. Formulation of the problem for Output B can be stated as follows.

(a2),(b1),(g) and $x_{uu'} \le \sum_{v\in N(u)} x_{v(u'+1)}$, for $u \in V, u' \in V'\setminus\{k\}$, and $x_{uk} \le \sum_{v\in N(u)} x_{v1}$, for $u \in V$.

$$\sum_{u\in V}\sum_{v\in N(u)}\sum_{u'\in V'\setminus\{k\}}\sum_{u'+1} w_{(u,v)} y_{(u,v)(u',u'+1)} +$$
$$\sum_{u\in V}\sum_{v\in N(u)} w_{(u,v)} y_{(u,v)(k,1)} \le K.$$

Here to the set of feasible solutions in output A, constraints (g), from Table **2**, are added. Constraints (g) ensure that if two arcs *(u,v)* and *(u',v')* are assigned to each other then $y_{(u,v)(u',v')} = 1$. The LHS of the last inequality adds up all distances in created a circuit that needs to be within a bound *K*. Note that the LHS is in the form of P2. Thus, if *Min K* is added to the formulation we find smallest *K* that satisfies output B. Formulation of the problem for Output C can be given as a bottleneck optimization. Feasible solution is the same as for output B, however, the last inequality is replaced by,

$w_{(u,v)} y_{(u,v)(u',v')} \le K$, for $(u,v) \in A$, and $(u',v') \in A'$.

Note that here also the LHS is in the form of P1. Thus, if *Min K* is added to the formulation we find smallest *K* that satisfies output C. Next, we present several linear arrangement problems. Bandwidth optimization on a graph which has variety of applications, [26,106-108], is discussed first. IP formulations have been given in [33,35], however, our formulation differs from theirs.

5.2. Linear Arrangement Problems on Graphs

(GT40) Bandwidth Problem (BP):

Input: Undirected graph *G=(V,E),* and a positive integer $K \le n$.

Output: Is there a linear ordering of *V* with the bandwidth *K* or less, i.e., a one-to-one function $f : V \to \{1,\ldots,n\}$ such that, for all edges $\{u,v\} \in E$, $| f(u) - f(v) | \le K$?

Define an undirected graph *G'=(V',E')* with |*V'*|=*n,* where the nodes are numbered *1,…,n.* For nodes $u',v' \in V'$ define an edge iff $|u' - v'| \le K$. Now, we have $E' = \{\{u',v'\} : u',v' \in V', \text{ and } w_{\{u',v'\}} = |u' - v'| \le K\}$.

The BP feasibility problem can be stated as a problem of finding a one-to-one assignment, $f : V \to V'$, such that for each $\{u,v\} \in E$ we have $\{f(u), f(v)\} \in E'$. An IP formulation derives from satisfying constraints (a1), (a2), and (c1) in Table **2**. In the optimization version of the BP, we look for a one-to-one assignment $f : V \to \{1,\ldots,n\}$ to minimize *K*. In that, *G'* is a complete graph and the objective function is $Min_f\{Max_{\{u,v\}\in E} \,|\, f(u) - f(v)\,|\}$ which is in the form of P1. IP formulation is as follows:

Min K, s.t. (a1),(a2),(f), and $w_{\{u',v'\}}y_{\{u,v\}\{u',v'|} \le K$, for $\{u,v\} \in E$, and $u',v' \in V'$.

Inequalities (f) assign a value of 1 to $\lambda - colorings$ when an edge $\{u,v\} \in E$ is assigned to an edge $\{u',v'\} \in E'$. The last set of inequalities ensures that a value of |*f(u)-f(v)*| is assigned to edge *{u,v}*.

(GT41) Directed Bandwidth Problem (DBP):

Input: Directed graph *G=(V,A),* and a positive integer $K \le n$.

Output: Is there a one-to-one function $f : V \to \{1,\ldots,n\}$ such that, for all arcs $(u,v) \in A$, *f(u)<f(v)* and $f(v) - f(u) \le K$?

Similar to the case of BP define a digraph *G'=(V',A')*. Define an arc between two nodes $u',v' \in V'$ if *v'>u'* and $v' - u' \le K$. Now, we have $A' = \{(u',v') : u',v' \in V', \text{ with } w_{(u',v')} = v' - u' \le K\}$. The DBP feasibility problem can be stated as a problem of finding a one-to-one assignment, $f : V \to V'$, such that for each $(u,v) \in A$ we have $(f(u), f(v)) \in A'$. IP formulation of the DBP derives from satisfying constraints (a1),(a2),(d1), and (e) in Table **2**. In the optimization version of the DBP, the graph *G'* is a complete graph. We look for a one-to-one assignment $f : V \to \{1,\ldots,n\}$ to minimize *K.* The objective function is $Min_f\{Max_{(u,v)\in A}\{f(v) - f(u)\}\}$ which is in the form of P1. IP formulation is as follows:

Min K, s.t. (a1),(a2),(e),(g), and $w_{(u',v')}y_{(u,v)(u',v')} \le K$, for $(u,v) \in A$, and $(u',v') \in A'$.

Next, we discuss the linear arrangement problem (LAP) which has variety of applications, [2,109,110].

(GT42) Linear Arrangement Problem (LAP):

Input: Undirected graph *G=(V,E)* with weight w_e for each $e \in E$, and a positive integer *K.*

Output: Is there a one-to-one function $f : V \to \{1,\ldots,n\}$ such that $\sum_{\{u,v\}\in E} w_{\{u,v\}} \,|\, f(u) - f(v)\,| \le K$?

Define an undirected graph *G'=(V',E')* with |*V'*|=*n,* and nodes numbered *1,…,n.* For two nodes $u',v' \in V'$ let $w_{\{u',v'\}} = |\,u' - v'\,|$. Find a one-to-one assignment $f : V \to V'$ such that $\sum_{\{u,v\}\in E} w_{\{u,v\}} \,|\, f(u) - f(v)\,| \le K$. An IP formulation can be stated as follows which is different from those in [36,37].

(a1),(a2),(f), and $\sum_{\{u,v\}\in E,\{u',v'\}\in E'}(w_{\{u,v\}}w_{\{u',v'\}}y_{\{u,v\}\{u',v'\}}) \le K$.

The optimization of the problem is in the form of P2. We need to add *Min K* to the feasibility problem.

(GT43) Directed Linear Arrangement Problem (DLAP):

Input: Directed graph *G=(V,A)* with weight w_a for each $a \in A$, and a positive integer *K.*

Output: Is there a one-to-one function $f : V \to \{1,\ldots,n\}$ such that *f(u)<f(v)* whenever $(u,v) \in A$ and such that $\sum_{(u,v)\in A} w_{(u,v)}(f(v) - f(u)) \le K$?

Define a complete digraph *G'=(V',A')* where |*V'*|=*n* with nodes numbered *1,…,n.* For $u',v' \in V'$ define an arc *(u',v')* if *v'>u',* and let $w_a = v' - u'$. We want to find a one-to-one assignment $f : V \to V'$ such that for each arc $(u,v) \in A$ if *u* is assigned to node $u' \in V'$ and *v* is assigned to node $v' \in V'$ then there is an arc $(u',v') \in A'$ such that $\sum_{(u,v)\in A} w_{(u,v)}(f(v) - f(u)) \le K$. An IP formulation of the problem is stated as follows.

(a1),(a2),(e),(g) and
$\sum_{(u,v)\in A,(u',v')\in A'}(w_{(u,v)}w_{(u',v')}y_{(u,v)(u',v')}) \le K$.

In the optimization version of the problem, we look for an assignment that minimizes the total cost. Thus, we only need to add *Min K* to the feasibility problem that is in the form of P2. Next, we discuss the *profile minimization problem* which has many applications, [79,111,112]. It is closely related to the BP and LAP. It has been shown in [79] that the PMP is equivalent to the IGC problem (IGC was discussed in Part I).

Profile Minimization Problem (PMP):

Input: Undirected graph *G=(V,E).*

Output: A one-to-one function $f : V \to \{1,\ldots,|V|\}$ such that $K = \sum_{u\in V} Max_{v\in N[u]}\{f(u) - f(v)\}$ is minimized.

Define a complete undirected graph *G'=(V',E')* with |*V'*|=*{1,…,*|*V*|*}.*We want to find a one-to-one assignment $f : V \to V'$ that minimizes the value of *K.* There is a clear similarity between PMP, BP, and LAP. PMP has the elements of both BP and LAP, (see [7]). For $u \in V$ let $c(u, f(u)) = f(u)$, and for $\{u,v\} \in E$ let $d(\{u,v\},\{f(u), f(v)\}) = -Min[f(u), f(v)]$. The objective of PMP is

$$Min_f \{\sum_{u \in V} Max_{v \in N(u)} \{c(u, f(u)) + d(\{u,v\},\{f(u), f(v)\})\}.$$

An IP formulation of PMP can be stated as follows, which is in the form of P3:

$Min_f \sum_{u \in V} K_u$, s.t. (a1),(a2),(f), and

$$\sum_{u' \in V'} c(u,u')x_{uu'} +$$
$$\sum_{u' \in V'} \sum_{v' \in V'} d(\{u,v\},\{u',v'\})y_{\{u,v\}\{u',v'\}} .$$
$$\leq K_u, \text{ for } u \in V, v \in N(u)$$

We next consider the *min-cut linear arrangement problem*, (see [6] for recent discussion and complexity of special cases). It has applications in telecommunication [65], computer aided design, automatic graph drawing, protein engineering, optimization of rail systems [2], and computational logic [66].

(GT44) Minimum Cut Linear Arrangement Problem (MCLAP):

Input: Undirected graph *G=(V,E).*

Output: Find smallest value of *K* and a one-to-one function $f : V \to \{1,\ldots,n\}$ such that for all $1 \leq i \leq n$,

$$|\{\{u,v\} \in E, f(u) \leq i < f(v)\}| \leq K.$$

Define a complete undirected graph *G'=(V',E')* with |*V'*|=*n* and let the nodes be numbered *1,…,n.* In the MCLAP feasibility problem we want to find a one-to-one assignment between nodes graphs *G* and *G'*, such that given any node $i' \in V'$, the cardinality of all edges *{u,v}* in *E* with one end *u* assigned to node $u' \in V'$ ($u' \leq i'$) and the other end *v* assigned to node $v' \in V'$ (*v'>i'*) is within the limit *K.* An IP formulation of the feasibility problem can be given as follows:

(a1),(a2),(f), and $\sum_{\{u,v\} \in E} \sum_{u' \leq i', v' > i'} y_{\{u,v\}\{u',v'\}} \leq K$, for $i' \in V'$

The formulation can be extended to the optimization of MCLAP by adding,

$$Min_f \{Max_{i' \in V'} \{\sum_{\{u,v\} \in E} \sum_{u' \leq i', v' > i'} y_{\{u,v\}\{u',v'\}}\}\}.$$

An IP formulation is derived by adding *Min K* to the feasibility problem. The objective function is in the form of P7. In that, we have *c(u,f(u))=0* for all *u* in *V*, and *d({u,v},{f(u),f(v)})=1* for all *{u,v}* in *E*. In the next section we discuss graph coloring variations. We first consider the *graph* and *subgraph isomorphism problems*. The complexity of the graph isomorphism problem is still open [11], although the subgraph isomorphism problem is known to be NP-hard [75].

5.3. Graph Coloring Variations

(A13, OPEN 1, page 285) Graph Isomorphism (GI):

Input: Undirected graphs *G=(V,E)* and *G'=(V',E'),* with |*V*|=|*V'*|=*n.*

Output: Are *G* and *G'* isomorphic, i.e., is there a one-to-one onto function $f : V \to V'$ such that $\{u,v\} \in E$ if and only if $\{f(u), f(v)\} \in E'$?

An IP formulation of the GI derives from satisfying (a1),(a2),(c1), and (c2) in Table **2**. Next, we discuss *subgraph isomorphismproblem (SI)* and the *induced subgraph isomorphismproblem (ISI)*. A subgraph of a graph *G* is a graph whose vertex set is a subset of that of *G*, and whose edges is a subset of that of *G* restricted to this subset. A subgraph *H* of a graph *G* is said to be *induced* (or full) if for any pair of vertices *u* and *v* of *H*, *{u,v}* is an edge of *H* if and only if *{u,v}* is an edge of *G*. In other words, *H* is an induced subgraph of *G* if it has exactly the edges that appear in *G* over the same vertex set (refer to example 1 for pictorial illustration).

(GT48) Subgraph Isomorphism (SI):

Input: Undirected graphs *G=(V,E)* and *G'=(V',E'),* with $|V| = n \geq |V'| = n'$.

Output: Does *G* contain a subgraph isomorphic to *G',* i.e., a subset $S \subseteq V$ and a subset $L \subseteq E$ such that |*S*|=|*V'*|*,* |*L*|=|*E'*| and there exists a one-to-one function $f : V' \to S$ satisfying $\{u',v'\} \in E'$ if and only if $\{f(u'), f(v')\} \in L$?

Induced Subgraph Isomorphism (ISI):

Input: Undirected graphs *G=(V,E)* and *G'=(V',E'),* with $|V| = n \geq |V'| = n'$.

Output: Does *G* contain a subgraph isomorphic to *G',* i.e., a subset $S \subseteq V$ and a subset $L \subseteq E$ such that |*S*|=|*V'*|*,* |*L*|=|*E'*| and there exists a one-to-one function $f : V' \to S$ satisfying $\{u',v'\} \in E'$ if and only if $\{f(u'), f(v')\} \in E$?

Note that as was discussed in example 1, Part I, in SI we do not rule out possibility of existing $\{u,v\} \in E \setminus L$ such that *f(u')=u* and *f(v')=v* for some $\{u',v'\} \in E'$. An

IP formulation of SI derives from satisfying (a2),(b1), and (c2) in Table **2**, and an IP formulation of ISI derives from satisfying (a2),(b1),(c1), and (c2). All discussion on GI and SI with the undirected graphs can be extended to the directed graphs in straightforward manner. Thus, in order to save space we will not present them here. Below we discuss the largest common subgraph problem (LCS) that has variety of applications, especially in data analysis, [113,114,115]. An IP formulation was presented in [116]. We present a simpler formulation with significantly less variables.

(GT49) Largest Common Subgraph Problem (LCS):

Input: Undirected graphs *G=(V,E)* and *G'=(V',E')*.

Output: Find the largest subsets $L \subseteq E$ and $L' \subseteq E'$ with |*L*|=|*L'*| such that the subgraphs $G_1 = (V, L)$ and $G_1' = (V', L')$ are isomorphic.

Consider an assignment $f : V \to V'$ where a node $u \in V$ can be assigned to only one node $u' \in V'$ and vice versa. Here, two edges, $\{u,v\} \in E$ and $\{u',v'\} \in E'$ are assigned to each other only if we have either $x_{uu'} + x_{vv'} = 1$ or $x_{uv'} + x_{vu'} = 1$, but not both. Let $w_e = w_{e'} = 1$ for all $e \in E$ and $e' \in E'$. For two edges $\{u,v\} \in E$ and $\{u',v'\} \in E'$, we have $w_{\{u,v\}} w_{\{u',v'\}} (x_{uu'} x_{vv'} + x_{uv'} x_{vu'}) = 1$ iffthey are assigned to each other. A quadratic IP formulation of the problem can be stated as follows:

$Max \sum_{u \in V} \sum_{u' \in V'} \sum_{v \in V} \sum_{v' \in V'} w_{\{u,v\}} w_{\{u',v'\}} (x_{uu'} x_{vv'} + x_{uv'} x_{vu'})$, s.t. (b1),(b2).

In order to transfer the quadratic formulation to a linear integer program, we let $y_{\{u,v\}\{u',v'\}} = 1$ if an edge $\{u,v\} \in E$ is assigned to an edge $\{u',v'\} \in E'$ with *u* assigned to *u'* and *v* assigned to *v',* and 0 otherwise. We let $z_{\{u,v\}\{u',v'\}} = 1$ if an edge $\{u,v\} \in E$ is assigned to edge $\{u',v'\} \in E'$ with *u* assigned to *v'* and *v* assigned to *u',* and 0 otherwise. An IP formulation can be stated as follows where objective function is in the form of P2.

$Min \sum_{\{u,v\} \in E, \{u',v'\} \in E'} (y_{\{u,v\}\{u',v'\}} + z_{\{u,v\}\{u',v'\}})$, s.t. (b1),(b2), and

$y_{\{u,v\}\{u',v'\}} \le x_{uu'}, y_{\{u,v\}\{u',v'\}} \le x_{vv'}, z_{\{u,v\}\{u',v'\}} \le x_{uv'}, z_{\{u,v\}\{u',v'\}} \le x_{vu'}$, f or $\{u,v\} \in E, \{u',v'\} \in E'$.

Next, we present maximum subgraph matching (MSM) with variety of applications, [117,118,119].

(GT50) Maximum Subgraph Matching (MSM):

Input: Directed graphs *G=(V,A)* and *G'=(V',A')*.

Output: Find the largest subsets $R \subseteq V \times V'$ such that for all $< u, u' >, < v, v' > \in R$, $(u,v) \in A$ if and only if $(u',v') \in A'$.

Here, we have many-to-many matching. A slightly different version of the problem is that the two generated subgraphs must preserve all arc relations and thus must be *isomorphic*. Authors in [120] originally introduced the MSM problem with no references to isomorphism, stated as, *"find the maximal matches, where a match is a correspondence (many-many relation) between a subgraph H of G and a subgraph H' of G', which preserves the relation."* In the following, first we discuss the *isomorphic* version of the problem (MISM) then we discuss MSM as stated in GT50. Consider an assignment $f : V \to V'$ with the assumption that node $u \in V$ can be assigned to only one node $u' \in V'$, and vice versa. We look for the largest subsets $S \subseteq V$ and $S' \subseteq V'$ with |*S*|=|*S'*| to be assigned to each other and such that for all $u, v \in S$ and $u', v' \in S'$, $(u,v) \in A$ if and only if $(u',v') \in A'$. The two subsets of nodes $S \subseteq V$ and $S' \subseteq V'$, respectively, create *induced* subgraphs in *G* and *G'*.

Maximum Induced Subgraph Matching (MISM):

Input: Directed graphs *G=(V,A)* and *G'=(V',A')*.

Output: Find the largest subsets $S \subseteq V$ and $S' \subseteq V'$ with |*S*|=|*S'*| and a one-to-one function $f : S \to S'$ such that for all $u, v \in S$ and $u', v' \in S'$, $(u,v) \in A$ if and only if $(u',v') \in A'$.

An IP formulation is as follows.

$Max \sum_{u \in V} \sum_{u' \in V'} x_{uu'}$, s.t. (b1),(b2),(d1),(d2),(e).

Now, an IP formulation for the MSM problem, GT50, can be stated as follows.

$Max \sum_{u \in V} \sum_{u' \in V'} x_{uu'}$, s.t. (d1),(d2),(e),(h2),(i2).

The objective functions in the above problems are in the form of P2. In the next several subsections, we consider the graph coloring and several related problems and their generalizations. The relationship among these problems is discussed, and IP formulations are introduced. The problems have enormous applications(see for example, [20,56,121-126]).

(GT4) Graph K-Colorability (GKC):

Input: Undirected graph *G=(V,E),* and an integer $K \leq n$.

Output: Does there exist a function $f : V \rightarrow \{1,\ldots,K\}$ such that $f(u) \neq f(v)$ whenever $\{u,v\} \in E$?

Graph Coloring (GC):

Input: Undirected graph *G=(V,E).*

Output: Find smallest value of *K,* a function $f : V \rightarrow \{1,\ldots,K\}$ such that $f(u) \neq f(v)$ whenever $\{u,v\} \in E$.

A generalization of the GKC is a graph *homomorphism* stated as follows, GT52. Extension to directed graph is also presented.

(GT52) Graph Homomorphism (GH):

Input: Undirected graphs *G=(V,E)* and *G'=(V',E').*

Output: Find a *homomorphism* from *G* to *G',* i.e., find a function $f : V \rightarrow V'$ such that $\{f(u), f(v)\} \in E'$ whenever $\{u,v\} \in E$.

Directed Graph Homomorphism (DGH):

Input: Directed graphs *G=(V,A)* and *G'=(V',A').*

Output: Find a *homomorphism* from *G* to *G',* i.e., find a function $f : V \rightarrow V'$ such that $(f(u), f(v)) \in A'$ whenever $(u,v) \in A$.

An IP formulation of the GH problem derives from satisfying(a1),(c1) and (i1) in Table **2**, and for DGH it derives from satisfying(a1),(d1),(e) and (i2). Consider now the GKC, and define a complete undirected graph *G'=(V',E')* where |*V'*|=*K* with the nodes numbered*1,…,K.* Now, GKC seeks for an assignment, $f : V \rightarrow V'$, such that $\{f(u), f(v)\} \in E'$ whenever $\{u,v\} \in E$. Since *G'* is complete graph this is the same as finding an assignment, $f : V \rightarrow V'$ such that the end points of an edge $\{u,v\} \in E$cannot be assigned simultaneously to one node in *G'.* An IP formulation of GKC drives from satisfying (a1) and (i1) in Table **2**. In the *graph coloring optimization problem (GC),* the goal is to find the smallest number of colors to assign to elements of the set *V.* Given an upper bound for the number of nodes *V'* (e.g., |*V'*|=*n*), similar to GKC define a complete graph *G'.* The IP formulation for GC problem is now stated as follows:

Min K, s.t. (a1),(i1), and $u' x_{uu'} \leq K$, for $u \in V, u' \in V'$.

Consider now a generalization of the coloring problem that we call *maximum (minimum) weight subset coloring problem* (MWSCP). The problem both in maximum and minimum format has many applications. In MWSCP, if node $u \in V$ receives a color $u' \in V'$ (where *V'* is the set of colors), then there is a benefit (or a cost) of *c(u,u')* (refer to [56] for approximation algorithms for special cases of these problems). For a given a subset $S \subseteq V$ and a function $f : S \rightarrow \{1,\ldots,K\}$, define $Z = \sum_{u \in S} c(u, f(u))$ to be weight of *S*. Now, MWSCP can be stated as follows:

Maximum (Minimum) Weight Subset Coloring Problem (MWSCP):

Input: Undirected graph *G=(V,E),* and an integer $K \leq n$.

Output: Find a maximum (or minimum) weight subset $S \subseteq V$, and smallest positive integer $l \leq K$, and a function $f : S \rightarrow \{1,\ldots,l\}$ such that $f(u) \neq f(v)$ whenever $\{u,v\} \in E$ and $u,v \in S$.

In MWSCP, we have a bi-objective optimization problem. We discuss the problem for maximum weight subset. It can be easily modified for minimum case. In MWSCP, we look for a maximum weight subset $S \subseteq V$ that can be colored by *K* colors. Let *Z** be the maximum weight. We then want to find smallest number of colors, *l*, that admits *Z**. Define a complete graph *G'=(V',E')* with *V'={1,…,K}*. Now, IP formulations can be given as follows. The IP (A) is in the form of P2, and (B) in the form of P1.

$Max\ Z = \sum_{u \in V} \sum_{u' \in V'} c(u,u') x_{uu'}$, s.t. (b1),(i1).

To find smallest number of colors, i.e., *strength* of *G*, we solve the following IP.

Min l, s.t. (b1),(i1), and $u' x_{uu'} \leq l$, for $u \in V, u' \in V'$, $l \geq 0$, and

$\sum_{u \in V} \sum_{u' \in V'} c(u,u') x_{uu'} = Z^*$.

Several important combinatorial optimization problems are special cases of MWSCP:

(1) If *K*=1then we have a single color and thus *G'* has a single vertex with no edge. For each node $u \in V$ let w_u be a given weight and define $c(u, f(u)) = w_u$. Now, IP (A) solves the well known *maximum weight independent set* problem.

(2) If *K=n,* then we have *S=V*. Given an assignment $f : V \rightarrow \{1,\ldots,K\}$ that satisfies condition of

MWSCP, for node $u \in V$ let $c(u, f(u)) = f(u)$. Here, weight of *c(u,f(u))* is exactly equal to the integer number of node assigned to *u.* Now if we minimize IP (A) instead of maximizing, then we have solved the *optimum cost chromatic partition (OCCP)* [127,128,129], which has the *sum coloring (SC)* [130] as a special case. Both OCCP and SC problems have numerous applications in scheduling theory, resource allocation, and VSLI design [98]. IP (B) then solves for *strength* of graph *G*. Note that these problems can be considered as special cases of the *Graph Grundy Number* (GGN, GT56, [75]), which is defined on digraphs. Our formulation can easily be modified for GGN (see [93] for survey of results).

Another generalization of the graph coloring is as follows, known as *weighted vertex coloring (WVCP)*. Its special cases have been discussed in the literature and have applications in batch scheduling, transmission of real time messages in metropolitan network, and dynamic storage allocation [54,55]. IP formulation was presented in [57].

Weighted Vertex Coloring Problem (WVCP):

Input: Undirected graph *G=(V,E),* and positive weight w_u for each $u \in V$.

Output: Find a function $f : V \to \{1, \ldots, K\}$ for some $K \le n$ such that $f(u) \ne f(v)$ whenever $\{u, v\} \in E$ and $u, v \in V$, to minimize $Z = \sum_{1 \le u' \le K} Max_{u \in V, f(u) = u'} \{w_u\}$.

Let $K_{\max}$ be an upper bound to the optimal value of *K* (e.g., $K_{\max} = n$), now define a complete graph *G'=(V',E')* where $V' = \{1, \ldots, K_{\max}\}$. The problem is to find a function $Min \sum_{u \in V} \sum_{v \in N(u)} \sum_{u' \in V' \backslash \{k\}} \sum_{u'+1} w_{(u,v)} y_{(u,v)(u',u'+1)}$ $+ \sum_{u \in V} \sum_{v \in N(u)} w_{(u,v)} y_{(u,v)(k,1)}$. that minimizes $w_{(u,v)} y_{(u,v)(u',v')} \le K, \text{ for } (u,v) \in A, \text{ and } (u',v') \in A'$.. Here, the objective function is a special case of P5. An IP formulation of WVCP is presented below. In the special case when all weights are equal to 1,we have the GC problem. However, the IP formulation of the special case of GC is different from the one we presented earlier (see [131] for similar formulation of a special case). In this formulation we have $c(u, f(u)) = w_u$.

$Min \sum_{u' \in V'} K_{u'}$, s.t. (a1),(i1), and $c(u, u') x_{uu'} \le K_{u'}$, for $u \in V, u' \in V'$.

An important graph optimization problem with enormous applications is the *maximum clique partition problem* (MCPP). It has the *graph partitioning problem* (GPP) as a special case. Applications of MCPP and GPP include clustering and pattern analysis(see [21,22,23,101,102,132,133,134,135,136] for recent applications). The problem is stated as follows:

Maximum Clique Partition Problem (MCPP):

Input: Undirected graph *G=(V,E),* weight w_e (a real number) for each $e \in E$.

Output: Find a partition, $V = \{V_1, \ldots, V_k\}$, of vertices into disjoint cliques to maximize

$Z = \sum_{1 \le l \le K} \sum_{u,v \in V_l, \{u,v\} \in E} w_{\{u,v\}}$.

The value of *K* that admits maximum in the problem is called *maximum clique partition number,* $\overline{\chi}_{\max}(G)$. Authors in [137,138] presented an IP formulation of the problem. Our formulation is based on the GC concept and is different from theirs. Let $\overline{G} = (V, \overline{E})$ be the complement of graph *G=(V,E).* Now, the MCPP is equivalent to the problem of partitioning vertices of $\overline{G}$ into *independent* sets (or coloring of vertices) where the sum over all weight of edges, $e \notin \overline{E}$, i.e., between pair of vertices in each independent set is maximized. Let $K_{\max}$ be an upper bound to $\overline{\chi}_{\max}(G)$ (e.g., $K_{\max} = n$). Define a graph *G'=(V',E')* where $V' = \{1, \ldots, K_{\max}\}$ and *E'* contains only self-loop edges for each node of *V'.* We look for a function $f : V \to V'$ such that $f(u) \ne f(v)$ whenever $\{u, v\} \in \overline{E}$ and such that *Z* is maximized. A set of nodes of *G* that are assigned to a node *u'* in *G'* create an independent set in $\overline{G}$, and thus a clique in *G*. An IP formulation can be given as follows, which is in the form of P2.

$Max \sum_{u' \in V'} \sum_{u,v \in V, \{u,v\} \in E} w_{\{u,v\}} y_{\{u,v\}\{u',u'\}}$, s.t. (a1), and

$x_{uu'} + x_{vu'} \le 1$, for $\{u, v\} \in \overline{E}$, and $u' \in V'$,

$y_{\{u,v\}\{u',u'\}} \le x_{uu'}$, $y_{\{u,v\}\{u',u'\}} \le x_{vu'}$, for $\{u, v\} \in E$, and $u' \in V'$.

If (b1) in Table **2** is substituted for (a1) in the IP formulation, and if $K_{\max} = 1$ (i.e., *G'* has a single node), and letting edge weights be equal to 1, then it solves the *maximum weighted clique* problem (MWC), i.e., the *maximum weighted independent set* problem (MWIS) in graph $\overline{G}$. Next, we discuss another interesting generalization of graph coloring known as *graph labeling* (GL) with many applications.

Graph Labeling (GL):

Input: Positive integers*m, k ($m \ge k$)* and λ_{mk} .

Output: Is there an *f(m,k)-labeling* of graph *G=(V,E),* i.e., is there an assignment $f : V \to \{0,1,\ldots,\lambda_{mk}\}$ (coloring of vertices with non-negative integers) such that

(A) $|f(u) - f(v)| \ge m$, if *u* and *v* are adjacent, and

(B) $|f(u) - f(v)| \ge k$, if *u* and *v* are distance two apart.

Input: Positive integers *m* and *k* ($m \ge k$).

Output: Find an *f(m,k)-labeling* of graph *G=(V,E),* i.e., $f : V \to \{0,1,\ldots,\lambda_{mk}\}$ such that (A) and (B) are satisfied and λ_{mk} is the smallest positive integer. The optimal value of λ_{mk} , denoted by $\lambda_{mk} - number$, called the minimum *span* of *G.*

Applications of graph labeling problems are reported in channel assignment [139], radio labeling [140,141,142], and computer networks [97]. Refer to [1,5,9,143] for comprehensive surveys. In [38] a *natural* IP formulation for the special case of the graph labeling problem is presented. Several generalizations of GL have been presented, [144,145]. These models can be considered as special cases of the *T-coloring* problem [47,48,146]. To present an IP formulation of the GL feasibility problem, we first define a complete graph *G'=(V',E')* with $V' = \{0,1,\ldots,\lambda_{mk}\}$ and *E'* as set of edges between all pairs $u',v' \in V'$. For each pair $u',v' \in V'$, let $w_{\{u',v'\}} = |u' - v'|$. In graph *G,* for every pair $u,v \in V$ with a distance 2 apart, we add a dummy edge. Denote the set of dummy edges by E_D . An IP formulation of the feasibility of GL is now stated in the following. In that, the assignment is between the nodes of two graphs, $\overline{G} = (V, E \cup E_D)$ and *G'=(V',E').* Feasible and infeasible assignments are similar to the graph coloring problem that appear above.

(a1), and $x_{uu'} + x_{vv'} \le 1$, and $x_{uv'} + x_{vu'} \le 1$, for $\{u,v\} \in E$, and $\{u',v'\} \in E'$, $w_{\{u',v'\}} < m$,

$x_{uu'} + x_{vv'} \le 1$, and $x_{uv'} + x_{vu'} \le 1$, for $\{u,v\} \in E_D$, and $\{u',v'\} \in E'$, $w_{\{u',v'\}} < k$,

$x_{uu'} + x_{vu'} \le 1$, for $\{u,v\} \in E \cup E_D$, and $u' \in V'$.

The formulation can be extended to GL optimization. Let λ_{mk} be an upper bound to $\lambda_{mk} - number$. Now, an IP formulation of the problem can be stated as follows.

$Min\ \lambda$, The above feasible conditions, $u'x_{uu'} \le \lambda$, for $u \in V, u' \in V'$, and $\lambda \ge 0$.

Here, the last set of inequalities combined with the minimization of the objective function ensure that the optimal value of λ will be equal to $\lambda_{mk} - number$. Next, we discuss the *metric labeling problem* (MLP), which has applications in different areas of engineering, especially in pattern analysis, [147-152]. An IP formulation for special cases was presented in [29,30], and in [150,151] approximation results were presented. Our IP formulation is more general than theirs.

Metric Labeling Problem (MLP):

Input: Undirected graph *G=(V,E)* with a possible self-loop for each node, and weight w_e for each $e \in E$. A set of labels *V'* with a metric distance function $D : V' \times V' \to R$, and a cost function, *c*, assigning a cost of *c(u,u')* when two nodes $u \in V$ and $u' \in V'$ are assigned to each other.

Output: Find an assignment, $f : V \to V'$, to minimize P2 (or P4).

In many applications, we have $f(u) \in L_u$ for a given subset $L_u \in V'$. In general, we have $V' = \bigcup_{u \in V} L_u$. If label $u' \in V'$ is assigned to node $u \in V$, it incurs a cost of *c(u,u'),* and if two labels $u',v' \in V'$ (possibly $u' = v'$)are assigned to end points of an edge $\{u,v\} \in E$, it incurs a cost of $d(\{u,v\},\{u',v'\}) = w_{\{u,v\}} D(u',v')$. Define a graph *G'=(V',E')* where $\{u',v'\} \in E'$ iff $D(u',v') \ne 0$. Originally the MLP was defined for *D(.,.)* as a metric function [28]. Our formulation does not assume that. We look for a function $f : V \to V'$ to minimize P2 (or P4). An IP formulation that minimizes the objective function P2 can be stated as follows.

$$Min\ \sum_{u \in V} \sum_{u' \in L_u} c(u,u') x_{uu'}$$

$$+ \frac{1}{2} \sum_{u \in V} \sum_{v \in N(u)} \sum_{u' \in L_u} \sum_{v' \in L_v \cap N[u']} w_{\{u,v\}} D(u',v') y_{\{u,v\}\{u',v'\}}$$

s.t. (g), and $\sum_{u'\in L_u} x_{uu'} = 1$, for $u \in V$.

An IP formulation of the problem that minimizes the objective function P4 can be stated as follows. Our formulation differs from those in the literature and is more general.

Min K, s.t. (g), and $\sum_{u'\in L_u} x_{uu'} = 1$, for $u \in V$,

$$\sum_{u:u'\in L_u} c(u,u')x_{uu'} + \frac{1}{2}\sum_{u:u'\in L_u}\sum_{v\in N(u)}\sum_{v'\in L_u \cap N[u']} w_{\{u,v\}} D(u',v') y_{\{u,v\}\{u',v'\}} \le K$$

, for $u' \in V'$.

A special case of the above formulation under the *task assignment in distributed computing systems (TADCS)* is discussed in [49] in which a graph matching formulation for the problem is presented and in which is also some discussion of an integer programming formulation. In the TSDCS, we have a set of tasks, *V,* and a set of processors, *V'.* A *task graph G=(V,E)* represents the set of tasks and their communication links shown by the edges of the graph. A *processor graph G'=(V',E')* represents the set of processors and their communication links shown by the edges of the graph. The problem is to find an assignment function $f : V \to V'$ that assigns the tasks to the processors. If task $u \in V$ is assigned to an eligible processor $u' \in V'$, the processing time is *c(u,u').* If two tasks $u, v \in V$ need to communicate (i.e., if there is an edge $\{u,v\} \in E$), they can be assigned to one processor; in that case, there is no communication time between them. Two tasks can also be assigned to two different processors if a communication link is possible between them (i.e., if there is an edge $\{u',v'\} \in E'$). In that case, there is a communication time between two tasks measured by *d({u,v},{f(u),f(v)}).* The problem with the objective P2 asks for an assignment $f : V \to V'$ to minimize the total times (e.g., [50-53]). In a problem with objective P4, we want to minimize the completion time of the last task completed, i.e., load of the processor with the maximum processing and communication times [49]. Other graph optimization problems related to the MLP with objective function P2 are *Graph Embeddings in d-Dimensions, Linear arrangement with d-dimensional cost*, and *Minimizing Storage-Time Product.* The above formulations can be modified for these problems. However, Even, Naor, Rao, and Schieber in [28] presented *indirect* IP formulations for each case.

In the next section, we present some variations of graph crossing optimization. We first discuss the *Golomb Ruler problem (GR)* which has a variety of applications including crystallography, radio-astronomy, and telecommunication [10,153,154,155].

5.4. Graph Crossing Optimization

A GR with *n* marks is defined as a set of *n* integers, called marks, $0 = a_1 < a_2, \ldots, < a_n$, such that the differences $a_j - a_i$, $1 \le i < j \le n$ are all distinct and nonzero. The optimal GR with *n* marks is a GR with the smallest value of a_n, referred to as the *length* of the ruler. The complexity of the problem is still open,[10]. Many good heuristics for large problems are available, [156]. Upper bounds are discussed in [10], and several IP formulations are proposed in [10,40]. We propose a new formulation here.

Golomb Ruler (GR):

Input: Two nonnegative integer numbers *n* and *K* with $n \le K$.

Output: Is there a set of *n* integers $\{a_1, \ldots, a_n\} \subseteq \{0,1,\ldots,K\}$ where $0 = a_1 < a_2, \ldots, < a_n$, such that the differences $a_j - a_i$, $1 \le i < j \le n$ are all distinct?

Input: A nonnegative integer *n.*

Output: A set of *n* integers $0 = a_1 < a_2, \ldots, < a_n$, such that the differences $a_j - a_i$, $1 \le i < j \le n$ are all distinct, and a_n is smallest.

Let *V={1,…,n}* be a set of *n* integers placed linearly on a line from the smallest to the largest. Similarly, let *V'={0,1,…,K}* be a set of *K+1* integers placed linearly from the smallest to the largest on a parallel line to the first line. Define a directed graph *G=(V,A)* where *A={(i,j):i<j,* for *i,j=1,…,n}* with weight $w_a = 1$ for $a \in A$, and define a directed graph *G'=(V',A')* where *A'={(i,j):i<j,* for *i,j=0,1,…,K}* with weight $w_{a'} = 1$ for each $a' \in A'$. Consider an assignment $f : V \to V'$, such that each element of *V* is assigned to one element from *V'* and one element of *V'* may be assigned to one element of *V* (the assignment between nodes $u \in V$ and $u' \in V'$ is called an *alignment*). We look for an assignment $f : V \to V'$, where no two *alignments* cross, and such that *f(u)-f(v)*for all $(u,v) \in A$ are distinct. Each element $u \in V$ receives an integer number $a_u = f(u)$ where $a_u < a_{u+1}$ for *u=1,…,n-1* and differences between these integers are distinct. If we force the smallest element in *V* to be assigned to the first element in *V',* then we have the desired result for the GR feasibility problem.

Below, we state an IP formulation of the feasibility problem.

(a1),(b2),(g), $x_{uv'} + x_{vu'} \leq 1,\ 1 \leq u < v \leq n$, and $1 \leq u' < v' \leq K$,

$\sum_{(u,v)\in A} \sum_{(u',v')\in A', v'-u'=k} y_{(u,v)(u',v')} \leq 1$, for *k=1,…,K*, and $x_{11} = 1$.

Here, we used directed graphs to formulate the problem, however, it is easy to modify the formulation for undirected graphs. In the GR optimization problem, it is assumed that an upper bound*K* for value of a_n is given (e.g., a large enough number). When an element from*V'* is assigned to the largest element of *V,* it carries the length of Golomb Ruler, i.e., a_n, that is minimized. The objective function is in the form of P1.

Min K,* (a1),(b2),(g), $x_{uv'} + x_{vu'} \leq 1,\ 1 \leq u < v \leq n$, and $1 \leq u' < v' \leq n'$,

$\sum_{(u,v)\in A} \sum_{(u',v')\in A', v'-u'=k} y_{(u,v)(u',v')} \leq 1$, for *k=1,…,K*,

$u' x_{nu'} \leq K^*$, for $u' \in V'$, and $x_{11} = 1$.

Next, *Contact Map Problem* (CMP), with applications in computational biology, is discussed (refer to [41,31,91], for recent results). We are given two undirected graphs *G=(V,E)* and *G'=(V',E')* where for each set *V* and *V',* a pre-specified ordering is given. Each set of vertices with their order represents a DNA. An edge in graph *G* (or *G')* represents a *contact* if two end vertices are "close enough", measured by a distance measure, with each other. Assume that the vertices of each graph are lined up in the given order on two straight lines parallel to each other. We look for the assignment between the vertices of the two graphs. An assignment between a vertex $u \in V$ and a vertex $u' \in V'$ is called an *alignment*. If the end points of an edge $\{u,v\} \in E$ are assigned to end points of an edge $\{u',v'\} \in E'$, it is called a *shared contact*. A feasible assignment is obtained when no two *alignments* cross. The objective is to find a feasible assignment between the vertices of two graphs with the maximum number of shared contacts.

Contact Map Problem (CMP):

Input: Two undirected graphs *G=(V,E)* with |*V*|=*n*, and *G'=(V',E')* with |*V'*|=*n',* and pre-specified ordering for the elements of *V,* and the elements of *V'.*

Output: An assignment $f : V \to V'$ with the maximum number of *shared contacts* where no two alignments are crossed.

The DNA similarity problem was first introduced in graph theoretic format in [94]. It was shown that the objective of CMP is to find the largest subgraph isomorphism between two graphs while preserving the pre-assigned sequence structure of the nodes in each graph and avoiding cross alignment. Using the formulas in LCS (discussed earlier), we can formulate the problem as follows. In *G,* let $w_e = 1$ for $e \in E$, and in *G',* let $w_{e'} = 1$ for $e' \in E'$. If two edges $\{u,v\} \in E$ and $\{u',v'\} \in E'$ with *u<v* and *u'<v'* are feasibly assigned together, then we must have $x_{uu'} x_{vv'} = 1$. In that, we have $w_e w_{e'} x_{uu'} x_{vv'} = 1$. Now using LCS results that we discussed earlier an IP formulation of the problem can be stated as follows [41]:

$Max \sum_{\{u,v\}\in E, \{u',v'\}\in E'} y_{\{u,v\}\{(u',v'\}}$, s.t. (b1),(b2), $x_{uv'} + x_{vu'} \leq 1$, for $1 \leq u \leq v \leq n$ and $1 \leq u' \leq v' \leq n'$,

$y_{\{u,v\}\{(u',v'\}} \leq x_{uu'}$, $y_{\{u,v\}\{(u',v'\}} \leq x_{vv'}$, for $\{u,v\} \in E$ and $\{u',v'\} \in E'$.

6. CONCLUSION AND REMARKS

In this paper, we introduced general graph optimization approaches. The general problems include a wide range of graph theoretic problems. Natural integer programming formulations for the general problems, as well as variety of specific matching problems, were presented. Some of these formulations are new and some are alternative to existing ones. Relationship between the general approaches and the quadratic assignment problem were pointed out. Further study should concentrate on applications of the IP formulations in practice. Often real problems have extra constraints that needed to be included in the formulations we presented here. Since these problems are NP-hard and in practice problems are very-large-scale thus heuristics and meta-heuristics should be designed in connection with formulations presented here as solution procedures. Another line of research may be use of the formulations presented here to develop lower or upper bounds in algorithmic development for specific problems.

ACKNOWLEDGEMENT

We thank two reviewers for their comments that greatly helped to improve the paper.

REFERENCES

[1] Calamoneri T. The L(h,k)-Labelling Problem: A Survey and Annotated Bibliography. The Computer Journal 2006; 49(5): 585-608.
http://dx.doi.org/10.1093/comjnl/bxl018

[2] Diaz J, Petit J, Serna M. A Survey of Graph Layout Problems. ACM Computing Surveys 2002; 34(3): 313-356. http://dx.doi.org/10.1145/568522.568523

[3] Gallian J. A Survey: recent results, Conjectures, and Open Problems in Labeling Graphs. Journal of Graph Theory 1989; 13: 491-504. http://dx.doi.org/10.1002/jgt.3190130410

[4] Gallian J. A Guide to the Graph Labeling Zoo. Discrete Applied Mathematics 1994; 49: 213-229. http://dx.doi.org/10.1016/0166-218X(94)90210-0

[5] Gallian J. A Dynamics Survey of Graph Labeling 2010; November 13.

[6] Gurski F. Linear Layouts Measuring Neighbourhoods in Graphs. Discrete Mathematics 2006; 306: 1637-1650. http://dx.doi.org/10.1016/j.disc.2006.03.048

[7] Lai Y-L, Williams K. A Survey of Solved Problems and Applications on Bandwidth, Edgesum, and Profile of Graphs. Journal of Graph Theory 1999; 31: 75-94. http://dx.doi.org/10.1002/(SICI)1097-0118(199906)31:2<75::AID-JGT1>3.0.CO;2-S

[8] Romanowski C, Nagi R, Sudit M. Data Mining in an Engineering Design Environment: OR Applications from Graph Matching. Computers and Operations Research 2006; 33: 3150-3160. http://dx.doi.org/10.1016/j.cor.2005.01.025

[9] Yeh KR. A Survey of Labeling Graphs with a Condition at Distance Two. Discrete Mathematics 2006; 306: 1217-1231. http://dx.doi.org/10.1016/j.disc.2005.11.029

[10] Meyer C, Jaumard B. Equivalence of Some LP-based Lower Bounds for the Golomb Ruler Problem. Discrete Applied Mathematics 2006; 154(1): 120-144. http://dx.doi.org/10.1016/j.dam.2005.07.006

[11] Toran J. On the Hardness of Graph Isomorphism. SIAM J. Comput 2004; 33: 1093-1108. http://dx.doi.org/10.1137/S009753970241096X

[12] Bar-Yehuda R, Bendel K, Freund A, Rawitz D. Local ratio: A Unified Framework for Approximation Algorithms. ACM Computing Surveys 2004; 36(4): 422-463. http://dx.doi.org/10.1145/1041680.1041683

[13] Blum C, Roli A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys 2003; 35(3): 268-308. http://dx.doi.org/10.1145/937503.937505

[14] Chandru V, Rao R. Combinatorial Optimization and Network Algorithms. ACM Computing Surveys 1996; 28(1): 55-58. http://dx.doi.org/10.1145/234313.234341

[15] Galil Z. Efficient Algorithms for Finding Maximum Matching in Graphs. ACM Computing Surveys 1986; 18(1): 23-38. http://dx.doi.org/10.1145/6462.6502

[16] Sofianopoulos S. The Process Allocation Problem: a Survey of the Application of Graph-Theoretic and Integer Programming Approaches. Journal of the Operational research Society 1992; 43(5): 407-413. http://dx.doi.org/10.1057/jors.1992.67

[17] Atamturk A, Nemhauser GL, Savelbergh M. Conflict Graphs in Solving Integer Programming Problems. European Journal of Operational Research 2000; 121: 40-55. http://dx.doi.org/10.1016/S0377-2217(99)00015-6

[18] Christofides N, Benavent E. An Exact Algorithm for the Quadratic Assignment Problem on a Tree. Operations Research 1989; 37: 760-768. http://dx.doi.org/10.1287/opre.37.5.760

[19] Khuller S, Raghavachari B. Graph and Network Algorithms. ACM Computing Surveys 1996; 28(1): 43-45. http://dx.doi.org/10.1145/234313.234334

[20] Gómez D, Montero J, Yáñez J, Poidomani C. A Graph Coloring Approach for Image Segmentation. OMEGA 2007; 35: 173-183. http://dx.doi.org/10.1016/j.omega.2005.05.003

[21] Grady L, Schwartz E. Isoperimetric Graph Partitioning for Image Segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence 2006; 28(3): 469-475. http://dx.doi.org/10.1109/TPAMI.2006.57

[22] Huang Y, Liu Q, Lv F, Gong Y, Mtaxas D. Unsupervised Image Categorization by Hypergraph Partition. IEEE Trans. Pattern Analysis and Machine Intelligence 2011; 33(6): 1266-1273. http://dx.doi.org/10.1109/TPAMI.2011.25

[23] Unsalan C, Boyer KL. A Theoretical and Experimental Investigation of Graph Theoretical Measures for Land Development in Satellite Imagery. IEEE Trans. Pattern Analysis and Machine Intelligence 2005; 27(4): 575-589. http://dx.doi.org/10.1109/TPAMI.2005.65

[24] Chu W, Ghahramani Z, Podtelezhnikov A, Wild D. Bayesian Segmental Models with Multiple Sequence Alignment Profiles for Protein Secondary Structure and Contact Mp Prediction. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2006; 3(2): 99-113.

[25] Venkatachalam B, Apple J, St. John K, Gusfield D. Untangling Tanglegrams: Comparing Trees by Their Drawings. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2010; 7(4): 588-597. http://dx.doi.org/10.1109/TCBB.2010.57

[26] Zhu Q, Adam Z, Choi V, Sankof D. Generalized Gene Adjacencies, Graph Bandwidth, and Clusters in Yeast Evolution. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2009; 6(2): 213-220. http://dx.doi.org/10.1109/TCBB.2008.121

[27] Reid E, Chen H. Mapping the Contemporary Terrorism Research Domain. International Journal of Human-Computer Studies 2007; 65(1): 42-56. http://dx.doi.org/10.1016/j.ijhcs.2006.08.006

[28] Even G, Naor J, Rao S, Schieber B. Divide-and-Conquer Approximation Algorithms via Spreading Metrics. Journal of ACM 2000; 47(4): 585-616. http://dx.doi.org/10.1145/347476.347478

[29] Kleinberg J, Tardos E. Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Field. Journal of ACM 2002; 49: 616-630. http://dx.doi.org/10.1145/585265.585268

[30] Chekuri C, Khanna S, Naor J, Zosin L. A Linear Programming Formulation and Approximation Algorithms for the Metric Labeling Problem. SIAM J Discrete Math 2005; 18: 608-625. http://dx.doi.org/10.1137/S0895480101396937

[31] Greenberg H, Hart W, Lancia G. Opportunities for Combinatorial Optimization in Computational Biology. INFORMS Journal on Computing 2004; 16(3): 211-232. http://dx.doi.org/10.1287/ijoc.1040.0073

[32] Junger M, Lee E, Mutzel P, Odenthal T. A Polyhedral Approach to the Multi-Layer Crossing Number Problem, Springer-Verlag 1997.

[33] Luttamaguzi J, Pelsmajer M, Shen Z, Yang B. Integer Programming Solutions for Several Optimization Problems in Graph Theory, DIMACS Technical Report 2005-02, January 2005.

[34] Almoahamad AH, Duffuaa SO. A Linear Programming Approach for the Weighted Graph Matching Problem," IEEE Trans. Pattern Analysis and Machine Intelligence 1993; 15(5): 522-525. http://dx.doi.org/10.1109/34.211474

[35] Pinana E, Plana I, Campos V, Martri R. GRASP and Path Relinking for the Matrix Bandwidth Minimization. European Journal of Operational Research 2004; 153: 200-210. http://dx.doi.org/10.1016/S0377-2217(02)00715-4

[36] Grotschel M, Junger M, Reinelt GA. A Cutting Plane Algorithm for the Linear Ordering Problem. Operations Research 1984; 32: 1195-1220. http://dx.doi.org/10.1287/opre.32.6.1195

[37] Grotschel M, Junger M, Reinelt GA. Facets of Linear Ordering Polytope. Mathematical Programming 1985; 33: 43-60. http://dx.doi.org/10.1007/BF01582010

[38] Eshghi K, Azimi P. Applications of Mathematical Programming in Graceful Labeling Graphs. Journal of Applied Mathematics 2004; 1: 1-8. http://dx.doi.org/10.1155/S1110757X04310065

[39] Junger M, Mutzel P. 2-Layer Straightline Crossing Minimization: Performance of Exact and Heuristic Algorithms. Journal of Graph Algorithms and Applications 1997; 1(1): 1-25. http://dx.doi.org/10.7155/jgaa.00001

[40] Lorentz R, Nilssen R. Application of Linear Programming to the Optimal Difference Triangle Set problem. IEEE Transactions on Information Theory 1991; 37(5): 1486-1488. http://dx.doi.org/10.1109/18.133274

[41] Caprara A, Lancia G, Carr B, Walenez B, Istrail S. 1001 Optimal PDB Structure Alignments: Integer Programming Methods for Finding the Maximum Contact Map Overlap, Journal of Computational Biology 2004; 11(1): 27-52. http://dx.doi.org/10.1089/106652704773416876

[42] Arora S. Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and Other Geometric Problems. Journal of the ACM 1998; 45(5): 753-782. http://dx.doi.org/10.1145/290179.290180

[43] Hero A, Michel O. Asymptotic Theory of Greedy Approximations to Minimal k-Point Random Graphs. IEEE Transactions on Information Theory 1999; 45(6): 1921-1938. http://dx.doi.org/10.1109/18.782114

[44] Lawler E, Lenstra J, Kan R, Shmoys D. The Traveling Salesman Problem, John-Wiley & Sons, Chichester 1990.

[45] Langevin A, Soumis F, Desrosiers J. Classification of Traveling Salesman Problem Formulations. Operations Research Letters 1990; 9: 127-132. http://dx.doi.org/10.1016/0167-6377(90)90052-7

[46] Orman AJ, Williams HP. A survey of different integer programming formulations of the travelling salesman problem, Springer 2006.

[47] Liu D-F. T-Colorings of Graphs. Discrete Mathematics 1992; 101: 203-212. http://dx.doi.org/10.1016/0012-365X(92)90603-D

[48] Roberts F. T-colorings of Graphs: Recent Results and Open Problems. Discrete Mathematics 1991; 93: 229-245. http://dx.doi.org/10.1016/0012-365X(91)90258-4

[49] Shen C-C, Tsai W-H. A Graph Matching Approach to Optimal Task Assignment in Distributed Computing Systems using a Minimax Criterion. IEEE Transactions on Computers 1985; 34(4): 197-203. http://dx.doi.org/10.1109/TC.1985.1676563

[50] Ernst A, Jiang H, Krishnamoorthy M. Exact Solutions to Task Allocation Problems. Management Science 2006; 52(10): 1634-1646. http://dx.doi.org/10.1287/mnsc.1060.0578

[51] Lewis M, Alidaee B, Kochenberger G. Using xQx to Model and Solve the Uncapacitated Task Allocation Problem. Operations Research Letters 2005; 33(2): 176-182. http://dx.doi.org/10.1016/j.orl.2004.04.014

[52] Menon S. Effective Reformulations for Task Allocation in Distributed Systems with a Large Number of Communicating Tasks. IEEE Transactions on Knowledge and Data Engineering 2004; 12: 1497-1508.

[53] Stone HS. Multiprocessor Scheduling with the Aid of Network Flow Algorithms. IEEE Trans of Software Engineering 1977; 3(1): 93-95. http://dx.doi.org/10.1109/tse.1977.233840

[54] Escoffier B, Monnot J, Paschos V. Weighted Coloring: Future Complexity and Approximability Results. Information Processing Letters 2006; 97: 98-103. http://dx.doi.org/10.1016/j.ipl.2005.09.013

[55] Guan DJ, Zhu X. A Coloring Problem for Weighted Graphs. Information Processing Letters 1997; 61(2): 77-81. http://dx.doi.org/10.1016/S0020-0190(97)00002-1

[56] Hassin R, Monnot J. The Maximum Saving Partition Problem. Operations Research Letters 2005; 33: 242-248. http://dx.doi.org/10.1016/j.orl.2004.07.007

[57] Malaguti E, Monaci M, Toth P. Models and Heuristic Algorithm for a Weighted Vertex Coloring Problem. Journal of Heuristics 2009; 15(5): 503-526. http://dx.doi.org/10.1007/s10732-008-9075-1

[58] Barbarossa S, Scutari G. Decentralized Maximum-Likelihood Estimation for Sensor Networks Composed of Nonlinearly Coupled Dynamical Systems. IEEE Transactions on Signal Processing 2007; 55(7): 3456-3470. http://dx.doi.org/10.1109/TSP.2007.893921

[59] Li Z, Li B, Lau LC. A Constant Bound on Throughput Improvement of Multicast Network Coding in Undirected Networks. IEEE Transactions on Information Theory 2009; 55(3): 1016-1026. http://dx.doi.org/10.1109/TIT.2008.2011516

[60] Yen CK. The Edge-orientation Problem and Some of its Variants on Weighted Graphs, Information Sciences 2006; 176(19): 2791-2816. http://dx.doi.org/10.1016/j.ins.2005.09.001

[61] Sadigh AN, Mozafari M, Kashan AH. A Mixed Integer Linear Program and Tabu Search Approach for the Complementary Edge Covering Problem. Advances in Engineering Software 2010; 41: 762-768. http://dx.doi.org/10.1016/j.advengsoft.2009.12.017

[62] Shih F, Cheng S. Adaptive Mathematical Morphology for Edge Linking. Information Sciences 2004; 167(1-4): 9-21. http://dx.doi.org/10.1016/j.ins.2003.07.020

[63] Yang SJ. Efficient Algorithms to Solve the Link-orientation Problem for Multi-square, Convex-bipartite, and Convex-split Networks. Information Sciences 2005; 171(4): 475-493. http://dx.doi.org/10.1016/j.ins.2004.11.008

[64] Biedl T, Chan T, Gnjali Y, Hajiaghayi MT, Wood D. Balanced vertex-orderings of Graphs. Discrete Applied Mathematics 2005; 148: 27-48. http://dx.doi.org/10.1016/j.dam.2004.12.001

[65] Banerjee S, Mukherjee B, Sarkar D. Heuristic Algorithms for Constructing Optimized Structures of Linear Multihop Lightwave Networks. IEEE Transactions on Communications 1994; 42(2/3/4): 1811-1826.

[66] Mak W-K, Young EFY. Temporal Logic Replication for Dynamically Reconfigurable FPGA Partitioning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 2003; 22(7): 952-959. http://dx.doi.org/10.1109/TCAD.2003.814237

[67] Borndofer R, Eisenblatter A, Grotschel M, Martin A. A. Frequency Assignment in Cellular Phone Networks. Annals of Operations Research 1998; 76: 73-93. http://dx.doi.org/10.1023/A:1018908907763

[68] Montemanni R, Smith DH, Allen S. Lower Bounds for Fixed Spectrum Frequency Assignment. Annals of Operations Research 2001; 107: 237-250. http://dx.doi.org/10.1023/A:1014911401612

[69] Montemanni R, Moon JNJ, Smith DH. An Improved Tabu Search Algorithm for the Fixed-Spectrum Frequency-Assignment Problem. IEEE Transactions on Vehicular Technology 2003; 52(3): 891-901. http://dx.doi.org/10.1109/TVT.2003.810976

[70] Montemanni R, Smith DH, Allen SM. An Improved Algorithm to Determine Lower Bounds for the Fixed Spectrum Frequency Assignment Problem. European Journal of Operational Research 2004; 156: 736-751. http://dx.doi.org/10.1016/S0377-2217(03)00127-9

[71] Pardalos P, Rendl F, Wolkowicz H. The Quadratic Assignment problem: A Survey and Recent Developments,

American Mathematical Society, Providence, Rhode Island, USA 1994.

[72] Loiola EM, De Abreu NM, Boaventura-Netto P, Hahn P, Querido T. A Survey for the Quadratic Assignment Problem. European Journal of Operational Research 2007; 176(2): 657-690. http://dx.doi.org/10.1016/j.ejor.2005.09.032

[73] Chiang W-C, Kouvelis P, Urban T. Incorporating Workflow Interference in Facility Layout Design: The Quartic Assignment Problem. Management Science 2002; 48(4): 584-590. http://dx.doi.org/10.1287/mnsc.48.4.584.206

[74] Chiang W-C, Kouvelis P, Urban T. Single- and Multi-objective Facility Layout with Workflow Interference Considerations. European Journal of Operational Research 2006; 174(3): 1414-1426. http://dx.doi.org/10.1016/j.ejor.2005.03.007

[75] Garey M, Johnson D. Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco 1991.

[76] Vashist A, Kulikowski C, Muchnik I. Ortholog Clustering on a Multipartite Graph, IEEE/ACM Transactions on Computational Biology and Bioinformatics 2007; 4(1): 17-27. http://dx.doi.org/10.1109/TCBB.2007.1004

[77] Guha S. Nested Graph Dissection and Approximation Algorithms. 41st Annual Symposium on Foundation of Computer Science 2000; November 12-14. http://dx.doi.org/10.1109/sfcs.2000.892072

[78] Lopes IC, Val´erio de Carvalho JM. An integer programming model for the Minimum Interval Graph Completion Problem. Electronic Notes in Discrete Mathematics 2010; 36: 583-590. http://dx.doi.org/10.1016/j.endm.2010.05.074

[79] Tsao Y-P, Chang GJ. Profile Minimization on Products of Graphs. Discrete Mathematics 2006; 306: 792-800. http://dx.doi.org/10.1016/j.disc.2006.01.015

[80] Chen H-F, Lee DT. On Crossing Minimization Problem. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 1998; 17(5): 406-418. http://dx.doi.org/10.1109/43.703928

[81] Herman I, Melancon G, Marshall MS. Graph Visualization and Navigation in Information Visualization: A Survey. IEEE Transactions on Visualization and Computer Graphics 2000; 6(1): 24-43. http://dx.doi.org/10.1109/2945.841119

[82] Laguna M, Marti R. GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization. INFORMS Journal on Computing 1999; 11: 44-52. http://dx.doi.org/10.1287/ijoc.11.1.44

[83] Laguna M, Marti R, Valls V. Arc Crossing Minimization in Hierarchical Digraphs with Tabu Search. Computers and Operations Research 1997; 24(12): 1175-1186. http://dx.doi.org/10.1016/S0305-0548(96)00083-4

[84] Miller R, Stout QF. Geometric Algorithms for Digitized Pictures on a Mesh Connected Computer. IEEE Trans. Pattern Anal Machine Intell 1985; 7: 216-228. http://dx.doi.org/10.1109/TPAMI.1985.4767645

[85] Nollenburg M, Wolff A. Drawing and Labeling High-Quality Metro Maps by Mixed-Integer Programming. IEEE Transactions on Visualization and Computer Graphics 2011; 17(5): 626-641. http://dx.doi.org/10.1109/TVCG.2010.81

[86] Garey M, Johnson D. Crossing Number is NP-complete. SIAM J Alg Disc Meth 1983; 4(3): 312-316. http://dx.doi.org/10.1137/0604033

[87] Valls V, Marti R, Lino P. A Tabu Thresholding Algorithm for Arc Crossing Minimization in Bipartite Graphs. Annals of Operations Research 1996; 63: 233-251. http://dx.doi.org/10.1007/BF02125456

[88] Eades P, Whitesides S. Drawings Graphs in Two Layers. Theoretical Computer Science 1994; 131: 361-374. http://dx.doi.org/10.1016/0304-3975(94)90179-1

[89] Finocchi I. Crossing-constrained hierarchical drawings. Journal of Discrete Algorithms 20064(2): 299-312. http://dx.doi.org/10.1016/j.jda.2005.06.001

[90] Zheng L, Song L, Eades P, Crossing Minimization Problems of Drawing Bipartite Graphs in Two Clusters, APVis '05 proceedings of the 2005 Asia-Pacific symposium on Information visualisation-Vo. 45 2005.

[91] Carpano MJ. Automatic Display of Hierarchized Graphs for Computer-Aided Decision Analysis, IEEE Trans. Systems, Man and Cybernetics 1980; 10(11): 705-715. http://dx.doi.org/10.1109/TSMC.1980.4308390

[92] Conte D, Foggia P, Sansone C, Vento M. Thirty Years of Graph Matching in Pattern Recognition. International Journal of Pattern Recognition and Artificial Intelligence 2004; 18(3): 265-298.

[93] Erdos P, Hedetniemi S, Laskar R, Prins G. On the Equality of the Partial Grundy and Upper Ochromatic Numbers of Graphs. Discrete Mathematics 2003; 272: 53-64. http://dx.doi.org/10.1016/S0012-365X(03)00184-5

[94] Goldman D, Istrail S, Papadimitriou C. Algorithm Aspects of Protein Structure Similarity, IEEE Computer Society Press) 1999; pp. 512-521.

[95] Griggs JR, Yeh R. Labeling Graphs with a Condition at Distance 2. SIAM J Discrete Math 1992; 5: 586-595. http://dx.doi.org/10.1137/0405048

[96] Holmgren A. Using Graph Models to Analyze the Vulnerability of Electric Power Networks. Risk Analysis 2006; 26(4): 955-969. http://dx.doi.org/10.1111/j.1539-6924.2006.00791.x

[97] Jin XT, Yeh R. Graph Distance-Dependent Labeling Related to Code Assignment in Computer Networks. Naval Research Logistics 2004; 51: 1-8.

[98] Malafiejski M. Sum coloring of graphs. In: M. Kubale, Editor, Graph colorings, American Mathematical Society 2004; pp, 55-66. http://dx.doi.org/10.1090/conm/352/06372

[99] Sohn J. Evaluating the Significance of Highway Network Links Under the Flood Damage: An Accessibility Approach. Transportation Research Part A 2006; 40: 491-506. http://dx.doi.org/10.1016/j.tra.2005.08.006

[100] Voth D. TIA Program Researches Terrorism Patterns. IEEE Intelligent Systems 2003; 18(4): 4-6. http://dx.doi.org/10.1109/MIS.2003.1217620

[101] Wang H, Alidaee B, Glover F, Kochenberger G. Solving Group Technology Problems via Clique Partitioning. International Journal of Flexible Manufacturing Systems 2007; 18(2): 77-97. http://dx.doi.org/10.1007/s10696-006-9011-3

[102] Wang H, Obremski T, Alidaee B, Kochenberger G. Clique Partitioning for Clustering: A Comparison with K-means and Latent Class Analysis. Communications in Statistics - Simulation and Computation 2008; 37(1): 1-13. http://dx.doi.org/10.1080/03610910701723559

[103] Guttin G, Punnen A. The Traveling Salesman Problem and Its Variations. Kluwer Academic Publishers, Dordrecht 2002.

[104] Vairaktarakis G. On Gilmore-Gomory's Open Question for the Bottleneck TSP. Operations Research Letters 2003; 31: 483-491. http://dx.doi.org/10.1016/S0167-6377(03)00050-6

[105] Vairaktarakis G. Simple Algorithms for Gilmore-Gomory's Traveling Salesman and Related Problems. Journal of Scheduling 2003; 6: 499-520. http://dx.doi.org/10.1023/A:1026200209386

[106] Bhusnurmath A, Taylor CJ. Graph Cuts via Norm Minimization. IEEE Transaction on Pattern Analysis and Machine Intelligence 2008; 30(10): 1866-1871. http://dx.doi.org/10.1109/TPAMI.2008.82

[107] Chai SM, Chiricescu S, Essick R, Lucas B, May P, Moat K, Norris JM, Schuette M, López-Lagunas A. Streaming Processors for Next-Generation Mobile Imaging Applications. IEEE Communications Magazine 2005; 43(12): 81-89. http://dx.doi.org/10.1109/MCOM.2005.1561924

[108] Klemm RP. WebCompanion: A Friendly Client-Side Web Prefetching Agent. IEEE Transactions on Knowledge and Data Engineering 1999; 11(4): 577-594. http://dx.doi.org/10.1109/69.790809

[109] Charon I, Hudrey O. An Updated Survey on the Linear Ordering Problem for Weighted or Unweighted Tournaments. Ann Oper Res 2010; 175: 107-158. http://dx.doi.org/10.1007/s10479-009-0648-7

[110] Haralick RM, Kartus J. Arrangements, homomorphisms, and discrete relaxation. IEEE Trans Syst Man Cybern SMC 1978; 8: 600-612. http://dx.doi.org/10.1109/TSMC.1978.4310036

[111] Guan DJ, Williams K. Profile Minimization on Triangulated Triangles. Discrete Mathematics 2003; 260: 69-76. http://dx.doi.org/10.1016/S0012-365X(02)00450-8

[112] Lai Y-L, Chang G. On the Profile of the Corona of Two Graphs. Information Processing Letters 2004; 89: 287-292. http://dx.doi.org/10.1016/j.ipl.2003.12.004

[113] Cunningham P. A Taxonomy of Similarity Mechanisms for Case-Based Reasoning. IEEE Transactions on Knowledge and data Engineering 2009; 21(11): 1532-1543. http://dx.doi.org/10.1109/TKDE.2008.227

[114] Kimelman D, Kimelman M, Mandelin D, Yellin DM. Bayesian Approaches to Matching Architectural Diagrams. IEEE Transactions on Software Engineering 2010; 36(2): 348-274. http://dx.doi.org/10.1109/TSE.2009.56

[115] Messmer BT, Bunke H. Efficient Subgraph Isomorphism Detection: A Decomposition Approach. IEEE Transactions on Knowledge and data Engineering 2000; 12(2): 307-323. http://dx.doi.org/10.1109/69.842269

[116] Wong JL, Kourshanfar F, Potkonjak M. Flexible ASIC: shared masking for multiple media processors, Proceedings of 42nd Design Automation Conference, June 13-17 2005; pp. 909-914. http://dx.doi.org/10.1145/1065579.1065818

[117] Horaud R, Sossa H. Polyhedral Object Recognition by Indexing. Pattern Recognition 1995; 28(12): 1855-1870. http://dx.doi.org/10.1016/0031-3203(95)00048-8

[118] Mateus D, Cuzzolin F, Horaud R, Boyer E. Articulated Shape Matching Using Locally Linear Embedding and Orthogonal Alignment. IEEE 11th International Conference on Computer Vision 2007. ICCV 2007; pp. 1-8. http://dx.doi.org/10.1109/iccv.2007.4409180

[119] Medaglia L, Fang SC. A Genetic-based Framework for Solving (Multi-criteria) Weighted Matching Problems. European Journal of Operational Research 2003; 149: 77-101. http://dx.doi.org/10.1016/S0377-2217(02)00484-8

[120] Barrow HG, Burstall RM. Subgraph Isomorphism, Matching Relational Structures and Maximal Cliques. Information Processing Letters 1976; 4: 83-84. http://dx.doi.org/10.1016/0020-0190(76)90049-1

[121] Al-Kofahi Y, Lassoued W, Lee W, Roysam B. Improved Automatic Detection and Segmentation of Cell Nuclei in Histopathology Images. IEEE Transactions on Biomedical Engineering 2010; 54(4): 841-852. http://dx.doi.org/10.1109/TBME.2009.2035102

[122] Chalopin J, Paulusma D. Graph Labelings Derived from Models in Distributed Computing: A Complete Complexity Classification. Networks 2011; pp. 1-25. http://dx.doi.org/10.1002/net.20432

[123] Haralick RM, and Shapiro L. The Consistent Labeling Problem: Part I. IEEE Trans Pattern Anal Machine Intell PAMI 1979; 1(2): 173-184. http://dx.doi.org/10.1109/TPAMI.1979.4766903

[124] Haralick RM, Shapiro L. The Consistent Labeling Problem: Part II. IEEE Trans Pattern Anal Machine Intell PAMI 1980; 2(3): 193-203. http://dx.doi.org/10.1109/TPAMI.1980.4767007

[125] Hausner A. A new clustering algorithm for coordinate-free data. Pattern Recognition 2010; 43: 1306-1319. http://dx.doi.org/10.1016/j.patcog.2009.10.018

[126] Karpovsky MG, Levitin LB. Data Verification and Reconciliation With Generalized Error-Control Codes. IEEE Transactions on Information Theory 2003; 49(7): 1788-1793. http://dx.doi.org/10.1109/TIT.2003.813498

[127] Kroon LG, Sen A, Deng H, Roy A. The Optimal Cost Chromatic Partition Problem for Trees and Interval Graphs, WG '96 Proceedings of the 22nd International Workshop on Graph-Theoretic Concepts in Computer Science, Springer-Verlag, London, UK 1997.

[128] Nicoloso S. Sum Coloring and Interval Graphs: A Tight Upper Bound for the Minimum Number of Colors. Discrete Mathematics 2004; 280: 251-257. http://dx.doi.org/10.1016/j.disc.2003.06.015

[129] Supowit KJ. Finding a Maximum Planar Subset of a Set of Nets in Channel. IEEE Trans Comput Aided Design 1987; 6(1): 93-94. http://dx.doi.org/10.1109/TCAD.1987.1270250

[130] Kubicka E. The Chromatic Sum of Graph, Western Michigan University, Kalamozoo 1989.

[131] Butenko S, Festa P, Pardalos P. On the Chromatic Number of Graphs. Journal of Optimization Theory and Applications 2001; 109(1): 69-83. http://dx.doi.org/10.1023/A:1017557620292

[132] Kochenberger G, Glover F, Alidaee B, Wang H. Clustering of Microarray Data via Clique Partitioning. Journal of Combinatorial Optimization 2005; 10: 77-92. http://dx.doi.org/10.1007/s10878-005-1861-1

[133] Lin L, Liu X, Zhu S-C. Layered Graph Matching with Composite Cluster Sampling. IEEE Trans. Pattern Analysis and Machine Intelligence 2010; 32(8): 1426-1442. http://dx.doi.org/10.1109/TPAMI.2009.150

[134] Santos JM, de Sa´ JM, Alexandre LA. LEGClust—A Clustering Algorithm Based on Layered Entropic Subgraphs. IEEE Trans. Pattern Analysis and Machine Intelligence 2008; 30(1): 62-75. http://dx.doi.org/10.1109/TPAMI.2007.1142

[135] Soundararajan P, Sarkar S. An In-Depth Study of Graph Partitioning Measures for Perceptual Organization. IEEE Trans. Pattern Analysis and Machine Intelligence 2003; 25(6): 642-660. http://dx.doi.org/10.1109/TPAMI.2003.1201817

[136] Yoo JS, Shekar S. A Joinless Approach for Mining Spatial Colocation Patterns. IEEE Transactions on Knowledge and Data Engineering 2006; 18(10): 1323-1337. http://dx.doi.org/10.1109/TKDE.2006.150

[137] Grotschel M, Wakabayashi Y. A Cutting Plane Algorithm for a Clustering Problem. Mathematical Programming 1989; 45: 59-96. http://dx.doi.org/10.1007/BF01589097

[138] Grotschel M, Wakabayashi Y. Facets of the Clique Partitioning Polytope. Mathematical Programming 1990; 47: 367-387. http://dx.doi.org/10.1007/BF01580870

[139] Hale W. Frequency Assignment: Theory and Application. Proc IEEE 1980; 68: 13-22. http://dx.doi.org/10.1109/PROC.1980.11899

[140] Bodlaender H, Broersma H, Fomin F, Pyatkin A, Woeginger G. Radio Labeling with Preassigned Frequencies. SIAM J Optim 2004; 15: 1-16.
http://dx.doi.org/10.1137/S1052623402410181

[141] Bodlaender H, Kloks H, Tan R, Van Leeuwen J. Approximations for of Graphs. The Computer Journal 2004; 47(2): 193-204.
http://dx.doi.org/10.1093/comjnl/47.2.193

[142] Erwin D, George J, Mauro D. On Labeling the Vertices of Products of Complete Graphs with Distance Constraints. Naval Research Logistics 2003; 50: 1-6.

[143] Griggs JR, Král D. Graph labellings with variable weights, a survey. Discrete Applied Mathematics 2009; 157: 2646-2658.

[144] Fiala J, Kral D, Skrekovski R. A Brooks-type Theorem for the Generalized List T-coloring. SIAM journal of Discrete Methods 2005; 19(3): 588-609.

[145] Kral D. The Channel Assignment Problem with Variable Weights. SIAM Journal of Discrete Methods 2006; 20(3): 690-704.
http://dx.doi.org/10.1137/040619636

[146] Graf A. Distance Graphs and the T-coloring Problem. Discrete Mathematics 1999; 196: 153-166.
http://dx.doi.org/10.1016/S0012-365X(98)00199-X

[147] Boykov Y, Vekseler O, Zabih R. Fast Approximate Energy Minimization via Graph Cuts. IEEE Trans. Pattern Analysis and Machine Intelligence 2001; 23(11): 1222-1239.
http://dx.doi.org/10.1109/34.969114

[148] Jiang H, Drew MS, Li Z-N. Matching by Linear Programming and Successive Convexification. IEEE Transactions on Pattern Analysis and Machine Intelligence 2007; 29(6): 959-975.
http://dx.doi.org/10.1109/TPAMI.2007.1048

[149] Jiang H, Yu SX, Martin DR. Linear Scale and Rotation Invariant Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 2011; 33(7): 1339-1355.
http://dx.doi.org/10.1109/TPAMI.2010.212

[150] Komodakis N, Tziritas G. Approximate Labeling via Graph Cuts Based on Linear Programming. IEEE Transactions on Pattern Analysis and Machine Intelligence 2007; 29(8): 1436-1453.
http://dx.doi.org/10.1109/TPAMI.2007.1061

[151] Komodakis N, Paragios N, Tziritas G. MRF Energy Minimization and Beyond via Dual Decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2011; 33(3): 531-552.
http://dx.doi.org/10.1109/TPAMI.2010.108

[152] Werner T. A Linear Programming Approach to Max-Sum Problem: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 2007; 29(7): 1165-1179.
http://dx.doi.org/10.1109/TPAMI.2007.1036

[153] Blackburn SR, Etzion T, Martin KM, Paterson MB. Two-Dimensional Patterns With Distinct Differences—Constructions, Bounds, and Maximal Anticodes. IEEE Transactions on Information Theory 2010; 56(3): 1216-1229.
http://dx.doi.org/10.1109/TIT.2009.2039046

[154] Blackburn SR, Etzion T, Martin KM, Paterson MB. Distinct Difference Configurations: Multihop Paths and Key Predistribution in Sensor Networks. IEEE Transactions on Information Theory 2010; 56(8): 3961-3972.
http://dx.doi.org/10.1109/TIT.2010.2050794

[155] Dollas A, Rankin W, McCracken D. A New Algorithm for Golomb Ruler Derivation and Proof of the 19 Mark Ruler. IEEE Trans. Inform. Theory 199844(1): 379-382.
http://dx.doi.org/10.1109/18.651068

[156] Galinier P, Jaumard N. A Tabu Search Algorithms for Difference Triangle Sets and Golomb Rulers. Computers and Operations Research 2006; 33: 955-970.
http://dx.doi.org/10.1016/j.cor.2004.08.006

Received on 13-05-2015 Accepted on 09-06-2015 Published on 30-07-2015