A Branch-and-Bound Algorithm for the Maximum Weight Perfect Matching Problem with Conflicting Edge Pairs

Temel Öncan*
Endüstri Mühendisliği Bölümü
Galatasaray Üniversitesi
İstanbul, TÜRKİYE
ytoncan@gsu.edu.tr

M. Hakan Akyüz
Endüstri Mühendisliği Bölümü
Galatasaray Üniversitesi
İstanbul, TÜRKİYE
mhakyuz@gsu.edu.tr

İ. Kuban Altınel
Endüstri Mühendisliği Bölümü
Boğaziçi Üniversitesi
İstanbul, TÜRKİYE
altinel@boun.edu.tr

ABSTRACT

This paper introduces a branch-and-bound (B&B) algorithm for the maximum weight perfect matching problem with conflicting edge pairs which is an $\mathcal{NP}$-hard problem. The proposed B&B algorithm is based on the relaxation obtained by removing the cardinality restriction on the feasible matchings and uses a non-dichotomized branching rule considering exposed vertices in a relaxed optimum solution. We have performed extensive computational experiments on randomly generated test instances and compared the proposed B&B algorithm with two Binary Integer Linear Programming models solved with an off-the-shelf commercial solver. According to our experiments, we have observed that the proposed B&B algorithm yields promising performance.

KEYWORDS

Integer Programming, Maximum Weight Perfect Matching, Branch-and-Bound, Conflicts

1 INTRODUCTION

The well-known *Maximum Weight Perfect Matching Problem* (MWPMP) consists of finding a perfect matching with maximum total weight [9]. The MWPMP is known to be polynomially solvable and it has several applications in scheduling, facility location and workforce planning [1]. In this work, we address an extension of the MWPMP with additional conflicting edge pair constraints. The so-called conflict constraints are also referred to as *the exclusionary side constraints* or *the disjunctive constraints*. Hence, the extended problem is named as the *Maximum Weight Perfect Matching Problem with Conflicting Edge Pairs* (MWPMC) which deals with determining a maximum weight perfect matching such that no two conflicting edges are in the solution at the same time, namely a maximum weight conflict free perfect matching. The MWPMC is known to be $\mathcal{NP}$-hard [7].

As a practical application of the MWPMC, we can mention the case arising in logistics, where toxic chemical substances and foods are prohibited to be stored in the neighbor locations. In a potential extension of the ordinary Symmetric Traveling Salesman Problem (STSP) there can be an incompatibility relation between the edges incident with vertices: some of them may not be selected if a particular edge is in the tour and a tour can consist of only compatible edges. This scenario is possible due to security reasons during the routing of an important person. Recall that a tour for a salesperson is a connected spanning subgraph in which all points have degree 2. If we drop the connectedness requirement, i.e. the subtour elimination constraints, the STSP turns into the determination of optimum 2-factors. The 2-factor problem is a natural extension of the perfect matching problem and in fact the determination of an optimal 2-factor reduces to the determination of an optimal perfect matching. In other words, the MWPMC can be viewed as a relaxation of the mentioned STSP extension.

In the literature, several combinatorial optimization problems with conflict constraints have been addressed. Among them we can mention, the minimum spanning tree problem with conflict constraints [6, 7, 16, 22, 24], the shortest path problem with conflict constraints [7], the transportation problem with exclusionary side constraints [10, 13, 23], the knapsack problem with conflict constraints [3, 4, 19], the bin packing problem with conflict constraints [5, 12, 21], the maximum flow problem under conflict and forcing constraints [20] and the minimum cost non-crossing flow problem on layered networks [2].

For all we know, the only work addressing the MWPMC is performed by Darmann et al. [7] where they provide complexity results of this problem and discuss its approximation hardness. As a special case, the MWPMC on bipartite graphs has been considered by Öncan et al. [16] and Öncan and Altınel [15]. In [16], the authors have introduced some complexity results as well as polynomially solvable cases. They have also proposed heuristics and lower bounding procedures. Recently, Öncan and Altınel [15] have developed two branch-and-bound (B&B) algorithms with dichotomized branching rules for the MWPMC in bipartite graphs.

The motivation of this work is to devise an exact solution approach, namely a specially tailored B&B algorithm, for the MWPMC in general graphs. Two Binary Integer Linear Programming (BILP) formulations are also proposed for the MWPMC. Computational experiments are performed on randomly generated test instances in order to compare the performance of the proposed B&B algorithm with the ones of the BILP formulations solved with CPLEX Mixed-Integer Linear Programming (MILP) solver. We have observed that the proposed B&B algorithm yields an outstanding performance for most of the cases.

In the next section, we introduce some definitions which are used throughout the paper and present two BILP formulations for the MWPMC. Then, in Section 3, we give the outline of the new B&B algorithm. Section 4 is where we report the experimental results. Finally, concluding remarks are discussed in Section 5.

2 TWO BINARY INTEGER LINEAR PROGRAMMING FORMULATIONS

Let $G = (V(G), E(G))$ be a graph, where $V(G)$ and $E(G)$ stand for the set of vertices and edges, respectively. We associate non-negative weights w_e for all edges $e \in E(G)$ and let $\delta_G(v)$ denote the subset of edges incident with vertex $v \in V(G)$. Then, the degree of vertex $v \in V(G)$ is defined as $d_G(v) = |\delta_G(v)|$ where

*Corresponding author

ISBN 978-3-89318-079-0 on OpenProceedings, ISSN 2510-7437

DOI: 10.5441/002/inoc.2019.07

$|\cdot|$ stands for the cardinality of a set. Besides, the *complement* of G is defined as the graph $\overline{G} = (V(\overline{G}), E(\overline{G}))$ where $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{\{u, v\} \notin E(G) : u, v \in V(G), u \neq v\}$.

A *stable set* (independent vertex set) of G is any subset $S \subseteq V(G)$ such that no two vertices in S are adjacent. The *Maximum Cardinality Stable Set Problem* (MCSP) consists of finding a stable set with a maximum number of vertices. This number is so-called as the stability number of G which is designated as $\alpha(G)$. When we associate weights to all vertices in $V(G)$, for every subset $S \subseteq V(G)$ the weight $w(S)$ is computed as the sum of the weights of vertices in S. The *Maximum Weight Stable Set Problem* (MWSP) tries to find a stable set S of G with maximum weight $w(S)$ which is represented as $\alpha_w(G)$. Both MCSP and MWSP are well-known $\mathcal{NP}$-complete combinatorial optimization problems [11].

A *matching* (independent edge set) $M = (V(M), E(M))$ of G is defined as a subset of $E(G)$ where no two edges share the same vertex. A *perfect matching* stands for a matching such that each vertex of $V(G)$ is incident with exactly one of the edges in the matching [8, 14]. The weight of a matching, i.e. $w(E(M))$, is calculated as the sum of the edge weights in the matching. Formally speaking, MWPMP tries to find a perfect matching M of G with maximum $w(E(M))$.

A *clique* $K = (V(K), E(K))$ is a complete subgraph of G. A clique is maximal if no other vertex $v \in V(G) \setminus V(K)$ is adjacent to all vertices in $V(K)$. Clearly, each clique in G corresponds to a stable set in $\overline{G}$. Therefore, the MCSP and the MWSP defined on G are equivalent to the *Maximum Clique Problem* (MCP) and the *Maximum Weight Clique Problem* (MWCP) on $\overline{G}$, respectively.

Now we are at the stage to present two BILP formulations for the MWPMC. Given a set of conflicting edges with each edge $e \in E(G)$, the MWPMC tries to find a perfect matching M such that no two conflicting edges e and f are allowed to be in $E(M)$. The conflicting edge pairs can be represented with a conflict graph $C = (V(C), E(C))$ where $V(C) \equiv E(G)$ and each conflicting edge pair corresponds to an edge in $E(C)$. The set of conflicting edges with edge $e \in E(G)$ is denoted as $\delta_C(e)$ and the degree of vertex e in the conflict graph C is $|\delta_C(e)| = d_C(e)$. In other words, the set of edges incident with vertex e is represented with $\delta_C(e) \subseteq E(C)$ in the conflict graph. Note that, when $f \in \delta_C(e)$ then $e \in \delta_C(f)$ and for two edges $e, f \in E(G)$ such that $\{e, f\} \in E(C)$.

Let the binary decision variable x_e be equal to 1 if and only if edge $e \in E(G)$ is in the perfect matching. Recall that w_e represents non-negative weight for edge $e \in E(G)$. Then we can formulate the MWPMC as follows:

$$\max z = \sum_{\{e\} \in E(G)} w_e x_e \tag{1}$$

subject to

$$\sum_{e \in \delta_G(v)} x_e = 1 \quad \text{for } v \in V(G) \tag{2}$$

$$x_e + x_f \leq 1 \quad \text{for } e \in E(G); f \in \delta_C(e) \tag{3}$$

$$x_e \in \{0, 1\} \quad \text{for } e \in E(G) \tag{4}$$

The objective function (1) is to minimize the weight of the perfect matching, i.e. $w(E(M))$. Constraints (2) enforce that every vertex is connected to exactly one of the edges in the solution. Constraints (3) obviate the conflicting edge pairs to be in the perfect matching. Constraints (4) are for the binary restrictions on the decision variables.

Notice that when we aggregate constraints (3) for all $f \in \delta_C(e)$ we obtain the following equivalent inequalities:

$$\sum_{f \in \delta_C(e)} x_f + d_C(e) x_e \leq d_C(e) \quad \text{for } e \in E(G) \tag{5}$$

We will call the formulation which consists of (1)-(4), as *STRONG* and the formulation including (1), (2), (4) and (5) as *WEAK*. Note that, when we define P_S and P_W as the polytopes corresponding to feasible solution sets of the Linear Programming (LP) relaxation of the *STRONG* and *WEAK* formulations respectively, then $P_S \subseteq P_W$ holds.

3 A BRANCH-AND-BOUND ALGORITHM FOR THE MWPMC

The proposed B&B algorithm employs maximum weight matching with conflicting edge pair (MWMC) relaxation of the MW-PMC, which is obtained when we replace the equality signs '=' in constraints (2) with inequalities '≤'. Hence, at each node of the B&B tree, including the root node, we solve the MWMC relaxation of the MWPMC.

During the exploration of the B&B tree, the solution of the MWMC relaxation yields a conflict free matching which is not necessarily perfect, and hence there may exist a set of exposed vertices in the relaxed optimal solution. Hence, given an exposed vertex $v \in V(G)$, subproblems of the B&B tree are generated by enforcing one by one the edges incident to v to be in the solution. Note that the B&B tree is not necessarily a binary tree.

All B&B nodes but the root node, are characterized by a set of edges. The ones which must be included in the solution and the edges that must be excluded from the solution. The edges in the former set are called as *included edges* and the edges in the latter one are named as *excluded edges*. The remaining edges of $E(G)$ are the *free edges*. Broadly speaking, during the run of the algorithm, at each node of the B&B tree, we consider a set of free edges and enforce them to be included in the solution. Meanwhile, we prune a B&B node either by comparing its upper bound value with the best known lower bound value or by making sure that the current node can not provide a feasible solution, i.e. a conflict free perfect matching. A formal outline of the proposed B&B algorithm is depicted with Algorithm 1.

Now we will discuss the details of the B&B algorithm. To this end, we will introduce some additional notation. Let t be the B&B node index and let $I^{(t)}$ and $X^{(t)}$ stand for the subsets of edges which must be included to and excluded from a conflict free perfect matching at node t of the B&B tree, respectively. Then the subproblem at node t is denoted by $MWPMC^{(t)}$ which is the MWPMC solved on the subgraph $G^{(t)} = (V(G^{(t)}), E(G^{(t)}))$ of the original graph G, with the vertex set $V(G^{(t)})$ obtained by deleting the vertices incident with the edges in $I^{(t)}$ and the edge set $E(G^{(t)}) = E(G) \setminus \{I^{(t)} \cup X^{(t)}\}$. For an upper bound on the $MWPMC^{(t)}$ we solve its maximum weight matching with conflicting edge pair relaxation, namely $MWMC^{(t)}$.

Let us define *extended conflict graph* $C^{(t)} = (V(C^{(t)}), E(C^{(t)}))$, at node t of the B&B tree, corresponding to $G^{(t)}$, where $V(C^{(t)})$ and $E(C^{(t)})$ are the set of vertices and edges of $C^{(t)}$, respectively. Vertices of the extended conflict graph $C^{(t)}$ correspond to the edges of $G^{(t)}$, i.e. $E(G^{(t)})$. The weights associated with the edges in $E(G^{(t)})$ are the weights of the vertices in $C^{(t)}$. On the other hand, the edges of the conflict graph represent the set of conflicting edge pairs and the set of incident edge pairs in $G^{(t)}$. Furthermore, we define the complement of $C^{(t)}$ as $\overline{C^{(t)}}$.

3.1 Initialization

At the initialization of the proposed B&B algorithm, we set $t = 0$ and we start with the original graph $G^{(0)} = G$ and its corresponding conflict graph $C^{(0)}$. The best known lower bound $\underline{z}$ is set to $-\infty$. The set of active problem list is initialized with $MWPMC^{(0)}$ and initially, both of the edge subsets $I^{(0)}$ and $X^{(0)}$ are empty.

3.2 Lower Bound

First of all, we check whether $\left|E(G^{(t)})\right| < \frac{|V(G)|}{2} - \left|I^{(t)}\right|$ holds in order to guarantee that the graph $G^{(t)}$ has the potential to yield a perfect matching. Otherwise we prune node t.

Next, we perform another check at the lower bounding step right after finding the maximum cardinality conflict free matching on $G^{(t)}$ which is denoted with $S^{(t)}$. Let $\alpha(C^{(t)})$ be the size of a maximum cardinality stable set $S^{(t)}$ on $C^{(t)}$. In case $\alpha(C^{(t)}) = \frac{|V(G)|}{2} - \left|I^{(t)}\right|$ holds then we again prune node t since we have a feasible solution for the $MWPMC^{(t)}$, i.e. a conflict free perfect matching, which consists of $I^{(t)} \cup S^{(t)}$. Here, this feasible solution gives us a chance to update the best known lower bound $\underline{z}$. After a lower bound is calculated we proceed to perform the upper bound computation. The determination of $\alpha(C^{(t)})$ requires the solution of the NP-hard MCSP at every node of the B&B tree. However, the use of exact value helps fathoming a larger number of nodes, which can balance the increase in the computational cost.

3.3 Upper Bound

At each node t, we compute an upper bound for the $MWPMC^{(t)}$ by solving the subproblem $MWMC^{(t)}$. In case the $MWMC^{(t)}$ has an optimum solution with weight $z^{(t)}$ then we have a maximum weight conflict free matching $M^{(t)}$ with value $z^{(t)}$. Hence, we determine an upper bound value at node t as $\overline{z}^{(t)}=z^{(t)} + \sum_{e \in I^{(t)}} c_e$. In case the $MWMC^{(t)}$ has no solution then we set $\overline{z}^{(t)} = -\infty$ in order to prune current node t. $MWMC^{(t)}$ is actually equivalent to the solution of the NP-hard MWSP on $C^{(t)}$. Hence, the determination of an upper bound at every node of the B&B tree has similar negative effect on the overall computational cost of the B&B algorithm. Again similarly, this can be balanced by the increasing ability to fathom more nodes because of the tightness of the upper bound.

3.4 Pruning

At this stage we either prune the current active node t or proceed to the division operation. Actually, we consider three cases. First, we check whether the current upper bound value is less than the best known lower bound value, i.e. $\overline{z}^{(t)} \leq \underline{z}$ holds. In such case the current active node is not taken into further consideration and fathomed. In the second case, the solution obtained in the upper bounding procedure, i.e. matching $M^{(t)}$ which is obtained by solving $MWMC^{(t)}$ on $G^{(t)}$, yields a conflict free perfect matching together with the edge subset $I^{(t)}$. Then, we have a chance to update the best known lower bound value. For the remaining case, we have at least one exposed vertex which will be considered in the division operation.

3.5 Branching Rule for Division

We perform branching operation considering the selected exposed vertex $v \in V(G^{(t)})$. Note that this operation does not necessarily outputs a dichotomized B&B tree. Recall that, at each node t of the B&B tree, $I^{(t)}$ and $X^{(t)}$ stand for the set of edges which must be included to and excluded from a conflict free perfect matching, respectively. Hence, given $M^{(t)}$ which is the maximum weight conflict free matching obtained by solving $MWMC^{(t)}$ on $G^{(t)}$ and $v \in V(G^{(t)})$ be an $M^{(t)}$ exposed vertex, we create $d^{(t)} = d_{G^{(t)}}(v)$ new subproblems by enforcing one by one each edge e_i incident to v, i.e. $e_i \in \delta_{G^{(t)}}(v)$, to be in the solution. Therefore, we generate subproblems with the following characterizations:

$$\left\{\begin{array}{l} I^{(ti)} = I^{(t)} \cup \{e_i\} \\ X^{(ti)} = X^{(t)} \cup \delta_C(e_i) \\ E(G^{(ti)}) = E(G) \setminus \{I^{(ti)} \cup X^{(ti)}\} \end{array}\right\} \text{ for } i = 1, \ldots, d^{(t)} \quad (6)$$

3.6 Stable Sets on Extended Conflict Graph

During the run of the B&B algorithm, we try to find a maximum cardinality conflict free matching and a maximum weight conflict free matching on $G^{(t)}$ in order to compute lower and upper bound values for the $MWPMC^{(t)}$, respectively. To compute a lower bound for the $MWPMC^{(t)}$, we solve the $MCSP$ on the extended conflict graph $C^{(t)}$ and find a stable set $S^{(t)}$ with the stability number $\alpha(C^{(t)})$. For that purpose, we solve the MCP on the complement of $C^{(t)}$, namely $\overline{C^{(t)}}$ by running the exact algorithm by Östergård [18]. On the other hand, to find an upper bound for the $MWPMC^{(t)}$ we solve the subproblem $MWMC^{(t)}$ by transforming it into an equivalent MWCP on $\overline{C^{(t)}}$. For that purpose, we employ the MWCP algorithm by Östergård [17].

4 COMPUTATIONAL EXPERIMENTS

We have performed the computational experiments in order to compare the performance of the proposed B&B algorithm with two BILP formulations solved by the state-of-the-art MILP solver CPLEX 12.7.0. All computations are performed on an HPE SRV DL380 GEN9 Server with a 2.20 GHz E5-2650v4 Processor and 192 GB RAM operating within Windows Server 2016 environment. To the best of our knowledge, there is no standard test library for MWMCP hence, we have generated random test instances.

4.1 Test Instances

In Table 1 we report the properties of the randomly generated instances. The first column includes the name of the instance sets where each of which contains 5 randomly generated test problems. The number following the letter "N" stands for the number of vertices of the corresponding instance set. Besides, a suffix is added to represent the density of the graph G. For example, a suffix of "H" is used to represent high edge density of the graph generated for the test instance.

In Table 1 configurations are presented in the columns two to six. The second column gives the number of vertices in G, i.e. $|V(G)|$, for each instance set. In our test bed, $|V(G)|$ changes from 36 to 58 vertices. The third column incorporates the number of edges in G, i.e. $|E(G)|$ which varies from 126 to 770 edges. The fourth column includes the number of conflicting edge pairs in G or equivalently the number of edges in the conflict graph C, i.e. $|E(C)|$. The fifth column stands for the edge density of G, i.e. $d(G)$, which is calculated as the number of edges in G divided by the maximum possible number of edges. We have employed three levels for edge density of graphs 0.2, 0.5 and 0.8 respectively for

Algorithm 1: Branch-and-bound algorithm for solving MWPMC using MWMC relaxations

Input: A graph $G = (V(G), E(G))$ edge weights $w_e \geq 0$, conflict graph $C = (V(C), E(C))$;
Output: A maximum weight conflict free perfect matching $M^* = (V(M^*), E(M^*))$
begin
(*Initialization*): Set $t = 0$, MWPMC$^{(0)} \longleftarrow$ MWPMC, $G^{(0)} = G$, $C^{(0)} = C$, $\mathcal{L} = \{\text{MWPMC}^{(0)}\}$, $I^{(0)} = \emptyset$, $X^{(0)} = \emptyset$, $\underline{z} = -\infty$
(*Termination test*): If $\mathcal{L} = \emptyset$, then output $E(M^*)$ and stop.
(Lower bounding): Select and delete a problem from $\mathcal{L}$, say MWPMC$^{(t)}$,
if $\left|E(G^{(t)})\right| < \frac{|V(G)|}{2} - \left|I^{(t)}\right|$ **then**
  there is no conflict free perfect matching of G with edges in $I^{(t)} \cup E(G^{(t)})$. Set $\overline{z}^{(t)} = -\infty$, to prune MWPMC$^{(t)}$ and go to *Pruning*
else
  Find the maximum cardinality conflict free matching in $G^{(t)}$, which is $S^{(t)}$ and let its size be $\alpha(C^{(t)})$
  if $\alpha(C^{(t)}) < \frac{|V(G)|}{2} - \left|I^{(t)}\right|$ **then**
    there is no conflict free perfect matching of G with edges in $I^{(t)} \cup E(G^{(t)})$.
    Set $\overline{z}^{(t)} = -\infty$ to prune MWPMC$^{(t)}$ and go to *Pruning*
  else
    if $\alpha(^{(t)}) = \frac{|V(G)|}{2} - \left|I^{(t)}\right|$ **then**
      $I^{(t)} \cup S^{(t)}$ are the edges of a conflict free perfect matching;
      if $w(I^{(t)} \cup S^{(t)}) > \underline{z}$ **then**
        Update the lower bound and incumbent by setting $\underline{z} = w(I^{(t)} \cup S^{(t)})$, $E(M^*) \longleftarrow I^{(t)} \cup S^{(t)}$ and go to *Upper bounding*
      end if
    end if
  end if
end if
(Upper bounding): Solve MWMC$^{(t)}$ relaxation on $G^{(t)}$
if MWMC$^{(t)}$ has a solution **then**
  Let $M^{(t)}$ be a maximum weight conflict free matching on $G^{(t)}$ and $z^{(t)}$ be its optimal value.
  Set $\overline{z}^{(t)} = z^{(t)} + w(I^{(t)})$
else
  Set $\overline{z}^{(t)} = -\infty$
end if
(Pruning):
  i. If $\overline{z}^{(t)} \leq \underline{z}$, then go to *Termination test.*
  ii. If there is no $M^{(t)}$-exposed vertex in $G^{(t)}$ (i.e. $M^{(t)}$ is a perfect matching in $G^{(t)}$ and $I^{(t)} \cup E(M^{(t)})$ are the edges of a perfect matching of G) and $\overline{z}^{(t)} < \underline{z}$ then set $\overline{z}^{(t)} = \underline{z}$ and $E(M^*) \longleftarrow I^{(t)} \cup E(M^{(t)})$ go to *Termination test*
  iii. If there is an $M^{(t)}$-exposed vertex in $G^{(t)}$(i.e. $M^{(t)}$ is not a perfect matching in $G^{(t)}$) then go to *Division.*
(Division): Select an $M^{(t)}$-exposed vertex v of $G^{(t)}$ and create $i = 1, 2, \ldots, d^{(t)} = d_{G^{(t)}}(v)$ subproblems.
Let {MWPMC$^{(ti)}$} obtained from {MWPMC$^{(t)}$} by enforcing edge e_i to be in the perfect matching for $e_i \in \delta_{G^{(t)}}(v)$. Add them to the active node list $\mathcal{L}$ with $\overline{z}^{(ti)} = \overline{z}^{(t)}$ for $i = 1, 2, \ldots, d^{(t)}$ and go to *Termination test*
end

low (L), medium (M) and high (H) edge density. The last column is for the density of the conflict graph, namely $d(C)$.

Instance generation process of the MWPMC is not straightforward and hence must be carefully handled. For that purpose, an initial perfect matching is arbitrarily generated and it is kept to guarantee the feasibility of the MWPMC test instance. Therefore, $|V(G)/2|$ edges which correspond to the initial perfect matching and some more edges are randomly generated, summing up to $|E(G)|$ edges. The generation of edges are performed such that each vertex has a degree of at least two in order to avoid trivial solutions. Besides, $|E(C)|$ conflicting edge pairs are randomly selected among possible edge pairs excluding the edge pairs of the initial perfect matching. Finally, the edge weights w_e are randomly generated such that $w_e \in [10, 900]$ is satisfied. The process is repeated for each instance and we have generated a total of 110 test instances in 22 sets reported in Table 1.

4.2 Computational Results

In Table 2 we present the results obtained with the proposed B&B algorithm. Table 3 includes the results output by the solution of STRONG and WEAK formulations with the CPLEX MILP solver. All experiments are performed with a CPU time limit of 600 secs.

In Table 2 and Table 3, the rows correspond to the average values for the instance sets. In the last rows, the overall averages of the corresponding columns are given. The **act** column includes the average number of active nodes remaining in the B&B node list $\mathcal{L}$ when the B&B algorithm stops. The **LB** and **UB** columns incorporate the average lower and upper bound values output by the B&B algorithm, respectively. In the **CPU(s)** column we provide the average CPU time required in seconds. The rightmost column, i.e. column **exp**, reports the average number of explored nodes during the run of the B&B algorithm.

Table 3 introduces the results obtained with the solution of the BILP formulations via CPLEX MILP solver with default options with a CPU time limit of 600 secs. The last row denotes the overall average values of the corresponding columns. The values under **Bound** columns are the average solution values obtained with CPLEX MILP solver. The columns **CPU(s)** are for the average CPU times is seconds required by the CPLEX MILP solver.

Considering the results reported with Table 2 and Table 3, we can observe that the B&B algorithm is more efficient than solving the BILP formulations via CPLEX MILP solver. Observe that, the overall average CPU time requirement of the B&B algorithm is 34.76 secs. compared to the ones by STRONG and WEAK formulations which are 97.12 secs. and 61.20 secs., respectively. Furthermore, we should state that, all instances except the ones in the set `N56-M` are solved to optimality by both the B&B algorithm and CPLEX MILP solver within the CPU time limit of 600 secs.

Last but not least, we should report that, the B&B algorithm could not yield the optimum in only 2 instances in the set `N56-M`. On the other hand, the *STRONG* and *WEAK* formulations solved with CPLEX MILP solver could not output the optimum in 3 and 2 cases in the set `N56-M` within the CPU time limit, respectively.

5 CONCLUDING REMARKS AND DISCUSSION

We have proposed an exact solution procedure and two mathematical programming formulations for the Maximum Weight Matching Problem with Conflicting Edge Pairs (MWPMC). Considering our preliminary computational experiments on randomly generated test instances we can state that the proposed

Table 1: Instance properties

	$\|V(G)\|$	$\|E(G)\|$	$\|E(C)\|$	d(G)	d(C)
N36-H	36	504	80000	0.8	0.74
N36-L	36	126	5000	0.2	0.73
N36-M	36	315	30000	0.5	0.71
N38-H	38	563	90000	0.8	0.67
N38-L	38	141	7500	0.2	0.85
N38-M	38	352	40000	0.5	0.75
N42-H	42	689	110000	0.8	0.56
N42-L	42	173	12000	0.2	0.89
N42-M	42	431	60000	0.5	0.74
N44-L	44	190	14000	0.2	0.86
N44-M	44	473	70000	0.5	0.71
N46-L	46	208	16000	0.2	0.82
N46-M	46	518	80000	0.5	0.68
N48-L	48	226	18000	0.2	0.78
N48-M	48	564	73800	0.5	0.65
N52-L	52	266	22000	0.2	0.70
N52-M	52	663	104000	0.5	0.55
N54-L	54	287	24000	0.2	0.65
N54-M	54	716	108000	0.5	0.49
N56-L	56	309	26000	0.2	0.61
N56-M	56	770	112000	0.5	0.45
N58-L	58	331	28000	0.2	0.58
Average	**45.73**	**400.68**	**51377.27**	**0.40**	**0.69**

Table 2: Performance of the B&B Algorithm

	act	LB	UB	CPU (s)	exp
N36-H	0	537.2	537.2	1.5	1043.4
N36-L	0	485.2	485.2	0.0	34.2
N36-M	0	560.2	560.2	0.3	436.6
N38-H	0	815.6	815.6	6.7	3608.8
N38-L	0	851.2	851.2	0.0	15.2
N38-M	0	455.8	455.8	0.2	335.8
N42-H	0	1032.4	1032.4	129.8	48724.8
N42-L	0	631.4	631.4	0.0	8.8
N42-M	0	491.6	491.6	0.5	442.4
N44-L	0	981.4	981.4	0.0	12
N44-M	0	932.6	932.6	0.7	557.2
N46-L	0	1025.6	1025.6	0.0	23.2
N46-M	0	840.4	840.4	1.7	1101.4
N48-L	0	760.2	760.2	0.0	31.8
N48-M	0	991.0	991.0	2.0	1102.8
N52-L	0	906.2	906.2	0.0	99.6
N52-M	0	1041.8	1041.8	38.7	15486.8
N54-L	0	874.6	874.6	0.1	137.8
N54-M	0	1051.4	1051.4	88.4	28790
N56-L	0	945.6	945.6	0.1	242.2
N56-M	16.4	1092.6	1096.8	491.5	130753
N58-L	0	1095.0	1095	0.2	293.4
Average	**0.75**	**836.32**	**836.51**	**34.67**	**10603.67**

B&B algorithm outperforms the MILP solver. It should be borne in mind that in its current form of the B&B algorithm two NP-hard problems have to be solved at every search node, which can be bound to pay a very high computational price with the increase of instance size and conflict density. However, it can be possible to compute upper bounds to both $\alpha(C^{(t)})$ and the optimum value of $MWMC^{(t)}$ using heuristics and further relaxations with considerably lower costs, which remains as a part of our future investigations. Furthermore, efficient heuristics and as well as meta-heuristics for the solution of MWPMC can be another fertile research avenue.

ACKNOWLEDGMENTS

This work is supported by the Galatasaray University Scientific Research Project (Grant No: 18.402.009) and the Scientific and Technical Research Council of Turkey - TÜBİTAK (Grant No: 217M477)

Table 3: Performance of the BILP Formulations

	STRONG		WEAK	
	Bound	**Cpu (s)**	**Bound**	**Cpu (s)**
N36-H	537.2	7.8	537.2	17.9
N36-L	485.2	0.3	485.2	0.2
N36-M	560.2	1.3	560.2	1.5
N38-H	815.6	77.5	815.6	174.5
N38-L	851.2	0.3	851.2	0.3
N38-M	455.8	2.5	455.8	1.9
N42-H	1032.4	363.0	1032.4	293.3
N42-L	631.4	0.5	631.4	0.3
N42-M	491.6	2.7	491.6	6.1
N44-L	981.4	0.7	981.4	0.5
N44-M	932.6	5.0	932.6	8.4
N46-L	1025.6	1.1	1025.6	2.2
N46-M	840.4	35.5	840.4	9.4
N48-L	760.2	1.4	760.2	0.5
N48-M	978.6	159.3	978.6	15.1
N52-L	906.2	1.6	906.2	1.1
N52-M	1041.8	316.0	1041.8	190.1
N54-L	874.6	1.7	874.6	1.1
N54-M	1051.4	572.8	1051.4	238.0
N56-L	945.6	1.7	945.6	1.4
N56-M	974.2	581.7	1092.6	381.2
N58-L	1095.0	2.1	1095.0	1.3
Average	**830.37**	**97.12**	**835.75**	**61.20**

REFERENCES

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. 1993. *Network Flows: Theory, Algorithms and Applications.* Prentice-Hall, Englewood Cliffs, N.J.

[2] İ. Kuban Altınel, Necati Aras, Zeynep Şuvak, and Z. Caner Taşkın. 2018. Minimum Cost Non-crossing Flow Problem on Layered Networks. *Discrete Appl. Math.* (2018). https://doi.org/10.1016/j.dam.2018.09.016

[3] Mariem Ben Salem, Raouia Taktak, A. Ridha Mahjoub, and Hanene Ben-Abdallah. 2018. Optimization Algorithms for the Disjunctively Constrained Knapsack Problem. *Soft Comput.* 22, 6 (March 2018), 2025–2043. https://doi.

org/10.1007/s00500-016-2465-7

[4] Andrea Bettinelli, Valentina Cacchiani, and Enrico Malaguti. 2017. A Branch-and-Bound Algorithm for the Knapsack Problem with Conflict Graph. *INFORMS J. Comput.* 29, 3 (2017), 457–473. https://doi.org/10.1287/ijoc.2016.0742

[5] Renatha Capua, Yuri Frota, and Luiz Satoru Ochi. 2018. A Study on Exponential-size Neighborhoods for the Bin Packing Problem with Conflicts. *J. Heuristics* 24, 4 (August 2018), 667–695. https://doi.org/10.1007/s10732-018-9372-2

[6] Francesco Carrabs, Raffaele Cerulli, Rosa Pentangelo, and Andrea Raiconi. 2018. Minimum Spanning Tree with Conflicting Edge Pairs: a Branch-and-Cut Approach. *Ann. Oper. Res.* (2018). https://doi.org/10.1007/s10479-018-2895-y

[7] Andreas Darmann, Ulrich Pferschy, Joachim Schauer, and Gerhard J. Woeginger. 2011. Path, Trees and Matchings under Disjunctive Constraints. *Discrete Appl. Math.* 159, 16 (September 2011), 1726–1735. https://doi.org/10.1016/j.dam.2010.12.016

[8] Ran Duan and Seth Pettie. 2014. Linear-Time Approximation for Maximum Weight Matching. *J. Assoc. Comput. Mach.* 61, 1 (January 2014), 1–23. https://doi.org/10.1145/2529989

[9] Jack Edmonds. 1965. Maximum Matching and a Polyhedron with 0,1-Vertices. *J. Res. Natl. Bur. Stand.* 69B, 1 and 2 (1965), 125–130.

[10] Annette M. Ficker, Frits C.R. Spieksma, and Gerhard J. Woeginger. 2018. Transportation Problem with Conflicts. *Ann. Oper. Res.* (2018), 1–21. https://doi.org/10.1007/s10479-018-3004-y

[11] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: a Guide to the Theory of NP-Completeness.* W. H. Freeman, New York.

[12] Michel Gendreau, Gilbert Laporte, and Frederic Semet. 2004. Heuristics and Lower Bounds for the Bin Packing Problem with Conflicts. *Comput. Oper. Res.* 31, 3 (March 2004), 347–358. https://doi.org/10.1016/S0305-0548(02)00195-8

[13] Dries Goossens and Frits C.R. Sieksma. 2009. The Transportation Problem with Exclusionary Side Constraints. *4OR-Q. J. Oper. Res.* 7, 1 (March 2009), 51–60. https://doi.org/10.1007/s10288-007-0067-z

[14] Chien-Chung Huang and Telikepalli Kavitha. 2017. New Algorithms for Maximum Weight Matching and a Decomposition Theorem. *Math. Oper. Res.* 42, 2 (May 2017), 411–426. https://doi.org/10.1287/moor.2016.0806

[15] Temel Öncan and İ. Kuban Altınel. 2018. A Branch-and-Bound Algorithm for the Minimum Cost Bipartite Perfect Matching Problem with Conflict Pair Constraints. *Electron. Notes Discrete Math.* 64 (February 2018), 5–14. https://doi.org/10.1016/j.endm.2018.01.002

[16] Temel Öncan, Ruonan Zhang, and Abraham P. Punnen. 2013. The Minimum Cost Perfect Matching Problem with Conflict Pair Constraints. *Comput. Oper. Res.* 40, 4 (April 2013), 920–930. https://doi.org/10.1016/j.cor.2012.10.022

[17] Patric R.J. Östergård. 1999. A New Algorithm for the Maximum-Weight Clique Problem. *Electron. Notes Discrete Math.* 3 (May 1999), 153–156. https://doi.org/10.1016/S1571-0653(05)80045-9

[18] Patric R.J. Östergård. 2002. A Fast Algorithm for the Maximum Clique Problem. *Discrete Appl. Math.* 120 (August 2002), 197–207. https://doi.org/10.1016/S0166-218X(01)00290-6

[19] Ulrich Pferschy and Joachim Schauer. 2009. The Knapsack Problem with Conflict Graphs. *J. Graph Algorithms Appl.* 13, 2 (2009), 233–249. https://doi.org/10.7155/jgaa.00186

[20] Ulrich Pferschy and Joachim Schauer. 2013. The Maximum Flow Problem with Disjunctive Constraints. *J. Comb. Optim.* 26, 1 (July 2013), 109–119. https://doi.org/10.1007/s10878-011-9438-7

[21] Ruslan Sadykov and François Vanderbeck. 2013. Bin Packing with Conflicts: a Generic Branch-and-Price Algorithm. *INFORMS J. Comput.* 25, 2 (2013), 244–255. https://doi.org/10.1287/ijoc.1120.0499

[22] Phillippe Samer and Sebastian Urrutia. 2015. A Branch and Cut Algorithm for Minimum Spanning Trees under Conflict Constraints. *Optim. Lett.* 9, 1 (January 2015), 41–55. https://doi.org/10.1007/s11590-014-0750-x

[23] Minghe Sun. 2002. The Transportation Problem with Exclusionary Side Constraints and Two Branch-and-Bound Algorithms. *Eur. J. Oper. Res.* 140, 3 (August 2002), 629–647. https://doi.org/10.1016/S0377-2217(01)00239-9

[24] Ruonan Zhang, Santosh N. Kabadi, and Abraham P. Punnen. 2011. The Minimum Spanning Tree Problem with Conflict Constraints and its Variations. *Discrete Optim.* 8, 2 (May 2011), 191–205. https://doi.org/10.1016/j.disopt.2010.08.001