Les Cahiers du GERAD ISSN: 0711–2440

Variable Neighborhood Search for Metric Dimension and Minimal Doubly Resolving Set Problems

N. Mladenović, J. Kratica,
V. Kovačević-Vujčić,
M. Čangalović

G–2011–27

June 2011

Variable Neighborhood Search for Metric Dimension and Minimal Doubly Resolving Set Problems

Nenad Mladenović

GERAD & School of Mathematics
Brunel University-West London
Uxbridge, UB8 3PH, UK
& Mathematical Institute
Serbian Academy of Sciences and Arts
Kneza Mihaila 36/III, 11 000 Belgrade, Serbia
nenad.mladenovic@brunel.ac.uk

Jozef Kratica

Mathematical Institute
Serbian Academy of Sciences and Arts
Kneza Mihaila 36/III, 11 000 Belgrade, Serbia
jkratica@mi.sanu.ac.rs

Vera Kovačević-Vujčić
Mirjana Čangalović

Faculty of Organizational Sciences
University of Belgrade
Jove Ilića 154, 11 000 Belgrade, Serbia
{verakov, canga}@fon.bg.ac.yu

June 2011

Les Cahiers du GERAD

G–2011–27

Abstract

In this paper we consider two similar NP-hard optimization problems on graphs: the metric dimension problem and the problem of determining minimal doubly resolving sets. Both arise in many diverse areas, including network discovery and verification, robot navigation, chemistry, etc. For each problem we propose a new mathematical programming formulation and test it with CPLEX and Gurobi, the two well-known exact solvers. Moreover, for solving more realistic large size instances, we design Variable Neighborhood Search based heuristic. An extensive experimental comparison on four different types of instances indicates that our VNS approach consistently outperforms genetic algorithm, the only existing heuristic in the literature designed for solving those problems.

Key Words: Metaheuristics, Combinatorial optimization, Variable neighborhood search, Metric dimension, Minimal doubly resolving set.

Résumé

Nous considérons dans cet article deux problèmes similaires et NP-difficiles d'optimisation sur les graphes : le problème de la dimension métrique et le problème de la détermination d'ensembles minimaux doublement résolvants. Tous deux se posent dans de nombreux domaines incluant la découverte de réseaux, la navigation de robots, la chimie, etc. Pour chaque problème, nous proposons une nouvelle formulation de programmation mathématique, testée avec les deux solveurs exacts bien connus, CPLEX et Gurobi. De plus, pour résoudre des grandes instances plus réalistes, nous élaborons une heuristique de Recherche à Voisinage Variable. Une comparaison expérimentale extensive sur quatre types d'instances indique que l'approche RVV est constamment meilleure qu'un algorithme génétique, le seul à avoir été proposé dans la littérature pour résoudre ces problèmes.

Acknowledgments: This research was partially supported by Serbian Ministry of Science under grants 174010 and 1474033.

1 Introduction

The metric dimension problem (MDP) is introduced independently by Slater (1975) and Harary & Melter (1976). Given a simple connected undirected graph $G = (V,E)$, where $V = \{1, 2, ..., n\}$, $|E| = m$, $d(u, v)$ denotes the distance between vertices u and v, i.e. the length of a shortest $u - v$ path. A vertex x of the graph G is said to resolve two vertices u and v of G if $d(u, x) \neq d(v, x)$. A vertex set $B = \{x_1, x_2, ..., x_k\}$ of G is a *resolving set* of G if every two distinct vertices of G are resolved by some vertex of B. Given a vertex t, the k-tuple $r(t, B) = (d(t, x_1), d(t, x_2), ..., d(t, x_k))$ is called the *vector of metric coordinates* of t with respect to B. A *metric basis* of G is a resolving set of the minimum cardinality. The *metric dimension* of G, denoted by $\beta(G)$, is the cardinality of its metric basis.

The MDP has been widely investigated. Since the complete survey of all the results is out of the scope of this paper, we will mention only some relevant recent results. There exist two different integer linear programming (ILP) formulations of the metric dimension problem: Chartrand et al. (2000); Currie & Oellerman (2001). The metric independence of the graph is defined as the fractional dual of the integer linear programming formulation of the metric dimension problem (Fehr et al., 2006). From the theoretical point of view it is important to obtain tight lower and upper bounds for the metric dimension of the Cartesian product of graphs (Peters-Fransen & Oellermann, 2006; Cáceres et al., 2007) and corona product of graphs (Yero et al., 2011). Another interesting theoretical topic is connection between the metric dimension and graph invariants such as diameter, number of vertices, vertex degrees, etc (Hernando et al., 2007, 2010). The relation of the bounds on the metric and partition dimensions of a graph has been established, as well as a construction showing that for all integers α and β with $3 \leq \alpha \leq \beta + 1$ there exists a graph G with partition dimension α and metric dimension β (Chappell et al., 2008). The metric dimension of several interesting classes of graphs have been investigated: Cayley digraphs (Fehr et al., 2006), Grassmann graphs (Bailey & Meagher, 2011), Johnson and Kneser graphs (Bailey & Cameron, 2011), silicate networks (Manuel & Rajasingh, 2011), convex polytopes (Imran et al., 2010) and generalized Petersen graphs (Javaid et al., 2008; Husnine & Kousar, 2010). It has been shown that some infinite graphs have also infinite metric dimension (Cáceres et al., 2009; Rebatel & Thiel, 2011).

The metric dimension arises in many diverse areas including network discovery and verification (Beerliova et al., 2006), geographical routing protocols (Liu & Abu-Ghazaleh, 2006), the robot navigation, connected joints in graphs, chemistry, etc.

The concept of a doubly resolving set of graph G has been recently introduced by Cáceres et al. (2007). Vertices x, y of graph G ($n \geq 2$) are said to doubly resolve vertices u, v of G if $d(u, x) - d(u, y) \neq d(v, x) - d(v, y)$. A vertex set D of G is a *doubly resolving set* of G if every two distinct vertices of G are doubly resolved by some two vertices of D. The minimal doubly resolving set problem (MDRSP) consists of finding a doubly resolving set of G with the minimum cardinality, denoted by $\psi(G)$. Note that if x, y doubly resolve u, v then $d(u, x) - d(v, x) \neq 0$ or $d(u, y) - d(v, y) \neq 0$, and hence x or y resolves u, v. Therefore, a doubly resolving set is also a resolving set and consequently $\beta(G) \leq \psi(G)$.

It has been proved that the metric dimension of the Cartesian product $G \square G$ is tied in a strong sense to doubly resolving sets of G with the minimum cardinality (Cáceres et al., 2007). In the same paper it has been proved that the upper bound for the metric dimension of $G \square H$ can be expressed as the sum of the metric dimension of G and the cardinality of a minimal doubly resolving set of H minus 1. Thus, doubly resolving sets are essential in the study of the metric dimension of Cartesian products.

Both problems are NP-hard in general case. The proofs of NP-hardness are given for the metric dimension problem in (Khuller et al., 1996), for the minimal doubly resolving set problem in (Kratica et al., 2009b). Moreover, in (Hauptmann et al., 2011) it has been proved that the MDP is not approximable within $(1-\epsilon)\ ln\ n$ for any $\epsilon > 0$ and an approximation algorithm which matches the lower bound is given. In a special case, in which the underlying graph is superdense, a greedy constant factor approximation algorithm is presented.

The first metaheuristic approach to the metric dimension problem is proposed in (Kratica et al., 2009a). The genetic algorithm (GA) proposed in that paper uses the binary encoding and the standard genetic operators adapted to the problem. The feasibility is enforced by repairing the individuals. The overall performance of GA is improved by a caching technique. Testing on various ORLIB instances and theoretically challenging classes of graphs shows that GA relatively quickly produces satisfactory results.

A similar genetic approach is used in (Kratica et al., 2009b) for solving the minimal doubly resolving set problem. The GA results for MDRSP on hypercubes are used in a dynamic programming approach to obtain upper bounds for the metric dimension of large hypercubes.

In this paper we propose mathematical programming models for MDP and MDRSP with the different objective functions than in previous papers. Instead of minimizing the cardinality of a resolving set, we rather minimize the number of pairs of vertices from G that are not resolved (doubly resolved) by vertices of a set with a given cardinality s. If the value of the new objective function is zero, we apply the same model with smaller cardinality s. Otherwise, we increase s. In that way the difficulty that arises in solving the plateaux problem, i.e. a large number of solutions with the same objective function values, vanishes with our new objective. Such a reformulation approach is not new. For example, the chromatic number of a graph can be obtained by finding feasible k-colorings with decreasing values of k. Another example is the p-median/p-center problem, which is usually solved by considering a sequence of covering problems with given radii, i.e. radii are changed in each iteration (Garcia et al., 2011). Moreover, if we introduce the distance between two formulations $\mathcal{F}(s_1)$ and $\mathcal{F}(s_2)$ to be k if $|s_1 - s_2| = k$, then this approach may be seen as a Formulation space search (Mladenović et al., 2005, 2007; Kochetov et al., 2008). Really, we search for the best formulation in the formulation space of MDP and MDRSP.

In this paper, we tackle the MDP and MDRSP by a variable neighborhood search (VNS) approach in order to improve the existing upper bounds. Experimental results include three sets of ORLIB test instances: crew scheduling, pseudo boolean and graph coloring. VNS is also tested on theoretically challenging large-scale instances of hypercubes and Hamming graphs. An experimental comparison on these instances indicates that VNS approach consistently outperforms the GA approach, both with respect to solution quality and computation time.

The paper is organized as follows. In Section 2 we present some interesting properties which are used in the sequel. The existing and a new 0-1 linear programming formulations for both problems are given in Section 3. The next two sections contain the main features of the variable neighborhood search for both problems and computational results on various large-scale instances, respectively.

2 Examples and preliminaries

In this section we first illustrate MDP and MDRSP on some simple examples and then we present some of their theoretical properties.

Example 1 *Consider graph G_1 on Figure 1. Set $B_1 = \{v_1, v_2, v_3\}$ is a resolving set of G_1 since the vectors of metric coordinates for all the vertices of G_1 with respect to B_1 are mutually different. More precisely, $r(v_1, B_1)$ = (0,1,1); $r(v_2, B_1)$ = (1,0,2); $r(v_3, B_1)$ = (1,2,0); $r(v_4, B_1)$ = (2,1,1); $r(v_5, B_1)$ = (1,2,1); $r(v_6, B_1)$ = (2,1,2). Using the same vector of metric coordinates and the definition it is easy to check that B_1 is also a doubly resolving set.*

However, B_1 is not a minimal resolving set since $B_2 = \{v_1, v_3\}$ is also a resolving set with smaller cardinality. On the other hand, set $B_3 = \{v_1\}$ is not a resolving set since $d(v_2, v_1) = d(v_3, v_1) = 1$. Using a similar argument it is easy to check that none of singleton vertices forms a resolving set, and hence $\beta(G_1) = 2$.

Note that B_2 is not a doubly resolving set because $d(v_6, v_1)-d(v_5, v_1) = d(v_6, v_3)-d(v_5, v_3) = 1$. Similarly, we can show that none of the subsets of two vertices forms a doubly resolving set. Thus, B_1 is a minimal doubly resolving set and $\psi(G) = 3$.

Example 2 *Consider graph G_2 given on Figure 1. It can be shown that $\{v_1, v_2, v_3\}$ is both a minimal resolving and a minimal doubly resolving set, and therefore $\beta(G_2) = \psi(G_2) = 3$.*

Some properties. The metric dimension has many interesting theoretical properties which are out of the scope of this paper. The interested reader is referred to (Hernando et al., 2005). One of key properties con-

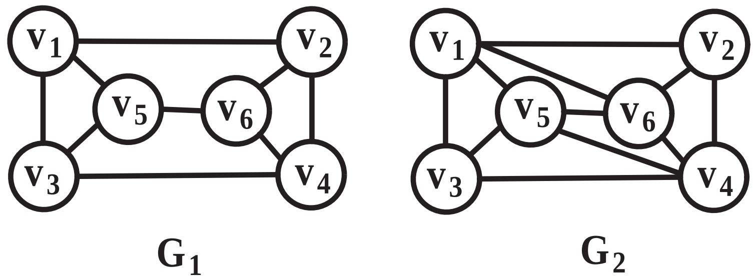

Figure 1: Graphs in Example 1 and Example 2

necting the metric dimension and the cardinality of a minimal doubly resolving set is stated in Proposition 1, which was proved in (Cáceres et al., 2007):

Proposition 1 *For arbitrary graphs* $G{=}(V_G, E_G)$ *and* $H{=}(V_H, E_H)$*, where* $|V_H| \geq 2$*,*

$$\max\{\beta(G), \beta(H)\} \leq \beta(G \square H) \leq \beta(G) + \psi(H) - 1 \tag{1}$$

Here, $G \square H$ is a graph which is the Cartesian product of graphs G and H. The vertex set of $G \square H$ is $V_G \times V_H = \{(a, v)|\ a \in V_G, v \in V_H\}$, while vertex (a, v) is adjacent to vertex (b, w) whenever $a = b$ and $\{v, w\} \in E_H$, or $v = w$ and $\{a, b\} \in E_G$.

For some simple classes of graphs it is possible to determine $\beta(G)$ explicitly: path has $\beta(G) = 1$, cycle has $\beta(G) = 2$, the complete graph with n vertices has $\beta(G) = n - 1$. On the other hand, the metric dimensions of some important classes of graphs such as hypercubes and Hamming graphs are still open problems. In the sequel we give a short description of hypercubes and Hamming graphs.

The *hypercube* Q_r is a graph whose vertices are all r-dimensional binary vectors, where two vertices are adjacent if they differ in exactly one coordinate. It is clear that Q_r has $n = 2^r$ vertices and $m = r \cdot 2^{r-1}$ edges.

Example 3 *Hypercube* Q_4 *has 16 vertices (0,0,0,0), (0,0,0,1), (0,0,1,0), ... , (1,1,1,1). For example, vertex (0,1,1,0) has adjacent vertices (1,1,1,0), (0,0,1,0), (0,1,0,0) and (0,1,1,1).*

The *Hamming graph* $H_{r,k}$ is the Cartesian product:

$$H_{r,k} = \underbrace{K_k \square K_k \square ... \square K_k}_{r} \tag{2}$$

where K_k denotes the complete graph with k vertices. The vertices of Hamming graphs can be considered also as r-dimensional vectors, where every coordinate has a value from the set $\{0, 1, ..., k-1\}$. As for hypercubes, two vertices are adjacent if they differ in exactly one coordinate. According to such an interpretation $Q_r = H_{r,2}$.

Obviously, $H_{r,k}$ has k^r vertices. Also, every vertex has the r-dimensional neighborhood with $k - 1$ neighbors with respect to each coordinate, so the overall number of edges is $k^r \cdot r \cdot (k-1)/2$.

Example 4 *Hamming graph* $H_{4,3}$ *has* $3^4 = 81$ *vertices (0,0,0,0), (0,0,0,1), (0,0,0,2), (0,0,1,0), ... , (2,2,2, 2). For example, vertex (0,1,1,0) has adjacent vertices (0,1,1,1), (0,1,1,2), (0,1,0,0), (0,1,2,0), (0,0,1,0), (0,2,1,0), (1,1,1,0), (2,1,1,0).*

For Hamming graphs it has been proved in (Cáceres et al., 2007) that $\beta(H_{2,k}) = \lfloor \frac{4k-2}{3} \rfloor$. The metric dimension is known exactly for hypercubes $Q_r, r \leq 8$. Upper bounds of $\beta(Q_r)$ for $9 \leq r \leq 14$ are obtained in (Cáceres et al., 2007; Kratica et al., 2009a). It has been proved in (Cáceres et al., 2007) that $\beta(Q_r) \leq r - 5$ for $r \geq 15$. This theoretical upper bound has been improved in (Kratica et al., 2009a) for $r \geq 17$. In (Kratica et al., 2009b) new bounds of $\beta(Q_r)$ for $r \leq 90$ are derived.

3 Mathematical programming formulations

In the literature there exist two integer linear programming (ILP) formulations of the metric dimension problem (Chartrand et al., 2000; Currie & Oellerman, 2001), and one ILP formulation of the minimal doubly resolving set problem (Kratica et al., 2009b). In this section for both problems we present the existing models as well as their new mathematical programming formulations.

3.1 Metric dimension

Let $B \subseteq V = \{1, ..., n\}$ and let $y_j = \begin{cases} 1, & j \in B \\ 0, & j \in V \backslash B \end{cases}$. As suggested in (Chartrand et al., 2000) the metric dimension problem can be formulated as the following 0-1 linear programming problem:

$$\min \quad \sum_{j=1}^{n} y_j \tag{3}$$

subject to

$$\sum_{j=1}^{n} |d_{uj} - d_{vj}| \cdot y_j \geq 1, \qquad 1 \leq u < v \leq n \tag{4}$$

$$y_j \in \{0, 1\}, \qquad (\forall j) \quad 1 \leq j \leq n \tag{5}$$

It is easy to see that each feasible solution of (3)-(5) defines a resolving set B of G, and vice versa. Note that $|d_{uj} - d_{vj}|$ are given constants and therefore the constraints (4) are linear. Instead of $|d_{uj} - d_{vj}|$, the following coefficients may be introduced:

$$A_{(u,v),j} = \begin{cases} 1, & d_{uj} \neq d_{vj} \\ 0, & d_{uj} = d_{vj} \end{cases} \tag{6}$$

Then (4) becomes

$$\sum_{j=1}^{n} A_{(u,v),j} \cdot y_j \geq 1 \qquad 1 \leq u < v \leq n \tag{7}$$

Thus, (3),(7) and (5) define another ILP model (Currie & Oellerman, 2001) for MDP. Note that both formulations have n variables and $n(n-1)/2$ constraints. Although the second formulation seems to have tighter constraints, numerical efficiency of CPLEX with respect to the two formulations is almost identical (Kratica et al., 2009a).

It is clear that the number of resolving sets with the same cardinality might be huge. Therefore, any local search type heuristic has difficulties to continue search after being in such a solution. In order to avoid this problem, we suggest an auxiliary objective function, and decompose MDP into a sequence of subproblems with relaxed resolving requirements and fixed cardinalities of feasible sets. In each subproblem we check if there exists a resolving set B of a given cardinality s. If such a resolving set exists then $\beta(G) \leq s$, otherwise $\beta(G) > s$.

Let B' be a subset of V with $|B'| = s$, and let the objective function $ObjF(B')$ be equal to the number of pairs of vertices of graph G that are not resolved by B'. If $ObjF(B') = 0$ then B' is a resolving set.

Let $A_{(u,v),j}$ be defined by (6), $y_j = \begin{cases} 1, & j \in B' \\ 0, & j \in V \backslash B' \end{cases}$ and let us introduce a new set of variables as

$$z_{uv} = \begin{cases} 1, & \textit{pair } (u, v) \textit{ is not resolved by } B' \\ 0, & \textit{pair } (u, v) \textit{ is resolved by } B' \end{cases} \tag{8}$$

Then for a given cardinality s the subproblem can be modelled as the following ILP which minimizes the $ObjF(B')$ subject to all $B' \subset V, |B'| = s$:

$$\min \quad \sum_{u=1}^{n-1} \sum_{v=u+1}^{n} z_{uv} \tag{9}$$

subject to

$$\sum_{j=1}^{n} y_j = s \tag{10}$$

$$\sum_{j=1}^{n} A_{(u,v),j} \cdot y_j + z_{uv} \geq 1, \quad 1 \leq u < v \leq n \tag{11}$$

$$y_j \in \{0,1\},\ z_{uv} \in \{0,1\}, \quad 1 \leq j \leq n,\ 1 \leq u < v \leq n \tag{12}$$

Proposition 2 *A subset B' is a resolving set of G of cardinality s if and only if the optimal objective function value of (9)-(12) is equal to zero.*

Proof. ($\Rightarrow$) Suppose that B' is a resolving set of cardinality s. Then, by (8), for each $u, v \in V$, $z_{uv} = 0$ and hence $\sum\limits_{u=1}^{n-1} \sum\limits_{v=u+1}^{n} z_{uv} = 0$. Constraint (10) is satisfied by assumption because $|B'| = s$. Since B' is a resolving set, for each $1 \leq u < v \leq n$ there exists $j \in B'$ such that $d_{uj} \neq d_{vj}$, implying $A_{(u,v),j} = 1$. From $j \in B'$ it follows that $y_j = 1$, so $A_{(u,v),j} \cdot y_j = 1$ which implies $\sum\limits_{j=1}^{n} A_{(u,v),j} \cdot y_j \geq 1 = 1 - z_{uv}$.

($\Leftarrow$) If $\sum\limits_{u=1}^{n-1} \sum\limits_{v=u+1}^{n} z_{uv} = 0$ then $z_{uv} = 0$ for all $1 \leq u < v \leq n$. It follows that $\sum\limits_{j=1}^{n} A_{(u,v),j} \cdot y_j \geq 1$ and hence there exists at least one j such that $d_{uj} \neq d_{vj}$, which by definition implies that B' is a resolving set. Constraint (10) guaranties that $|B'| = s$. □

Using the subproblems (9)-(12) we can solve MDP in the following way. We set s to be equal to an upper bound for the metric dimension (in the worst case $\beta(G) \leq n-1$) minus one and iteratively solve subproblems (9)-(12), decreasing s by one as long as the optimal objective function value is zero. If this value is not zero, the metric dimension is equal to $s+1$. Another approach would be to start with s equal to a lower bound for the metric dimension (in the worst case $\beta(G) \geq 1$) and iteratively solve subproblems (9)-(12), increasing s by one as long as the optimal objective function value is greater than zero.

3.2 Minimal doubly resolving set

In the case of MDRSP the ILP formulation in (Kratica et al., 2009b) is defined as follows. Let

$$A_{(u,v),(i,j)} = \begin{cases} 1, & d(u,i) - d(v,i) \neq d(u,j) - d(v,j) \\ 0, & d(u,i) - d(v,i) = d(u,j) - d(v,j) \end{cases} \tag{13}$$

where $1 \leq u < v \leq n, 1 \leq i < j \leq n$. Variable y_i described by (14) determines whether vertex i belongs to a doubly resolving set D. Similarly, x_{ij} determines whether both vertices i, j are in D.

$$y_i = \begin{cases} 1, & i \in D \\ 0, & i \notin D \end{cases} \tag{14}$$

$$x_{ij} = \begin{cases} 1, & i, j \in D \\ 0, & otherwise \end{cases} \tag{15}$$

The ILP model of the MDRSP can now be formulated as:

$$\min \quad \sum_{k=1}^{n} y_k \tag{16}$$

subject to:

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} A_{(u,v),(i,j)} \cdot x_{ij} \geq 1 \qquad 1 \leq u < v \leq n \tag{17}$$

$$x_{ij} \leq \frac{1}{2}y_i + \frac{1}{2}y_j \qquad 1 \leq i < j \leq n \tag{18}$$

$$x_{ij} \geq y_i + y_j - 1 \qquad 1 \leq i < j \leq n \tag{19}$$

$$x_{ij} \in \{0,1\},\ y_k \in \{0,1\} \qquad 1 \leq i < j \leq n,\ 1 \leq k \leq n \tag{20}$$

Note that ILP model (16)-(20) has $\frac{1}{2}n^2 + \frac{1}{2}n$ variables and $\frac{3}{2}n^2 - \frac{3}{2}n$ linear constraints.

MDRSP can be decomposed in a similar way into ILP subproblems. Let D' be a subset of V with $|D'| = s$, and let the objective function $ObjF(D')$ be equal to the number of pairs of vertices of graph G that are not doubly resolved by D'. If $ObjF(D') = 0$ then D' is a doubly resolving set. Let $A_{(u,v),(i,j)}$ be defined by (13), $y_i = \begin{cases} 1, & i \in D' \\ 0, & i \notin D' \end{cases}$, $x_{ij} = \begin{cases} 1, & i,j \in D' \\ 0, & otherwise \end{cases}$ and let us introduce a new set of variables:

$$z_{uv} = \begin{cases} 1, & pair\ (u,v)\ is\ not\ doubly\ resolved\ by\ D' \\ 0, & pair\ (u,v)\ is\ doubly\ resolved\ by\ D' \end{cases} \tag{21}$$

The following ILP minimizes the $ObjF(D')$ subject to all $D' \subset V$ with cardinality s.

$$\min \sum_{u=1}^{n-1}\sum_{v=u+1}^{n} z_{uv} \tag{22}$$

subject to:

$$\sum_{j=1}^{n} y_j = s \tag{23}$$

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} A_{(u,v),(i,j)} \cdot x_{ij} + z_{uv} \geq 1 \qquad 1 \leq u < v \leq n \tag{24}$$

$$x_{ij} \leq \frac{1}{2}y_i + \frac{1}{2}y_j \qquad 1 \leq i < j \leq n \tag{25}$$

$$x_{ij} \geq y_i + y_j - 1 \qquad 1 \leq i < j \leq n \tag{26}$$

$$x_{ij} \in \{0,1\} \qquad 1 \leq i < j \leq n \tag{27}$$

$$y_k \in \{0,1\} \qquad 1 \leq k \leq n \tag{28}$$

$$z_{uv} \in \{0,1\} \qquad 1 \leq u < v \leq n \tag{29}$$

Proposition 3 *A subset D' of cardinality s is a doubly resolving set of G if and only if the optimal objective function value of (22)-(29) is zero.*

The proof goes along the similar lines as the proof of Proposition 2 and will be omitted.

4 Variable neighborhood search for MDP and MDRSP

Variable neighborhood search (VNS) is an effective metaheuristic introduced in (Mladenović & Hansen, 1997). The basic idea of VNS is to use more than one neighborhood structure and to proceed to a systematic change of them within a local search. The algorithm remains in the same solution until another solution better than the incumbent is found and then moves there. Neighborhoods are usually ranked in such a way that intensification of the search around the current solution is followed naturally by diversification. The level of intensification and diversification can be controlled by a few parameters.

There are two crucial factors for a successful VNS implementation:

- a choice of suitable neighborhood structures and a shaking procedure which enables diversification;
- a fast and efficient local search procedure.

The VNS algorithm usually explores different increasingly distant neighborhoods whenever a local optimum is reached by a prescribed local search. Let N^k $(k = k_{min}, \ldots, k_{max})$ be a finite set of neighborhood structures, where $N^k(X)$ is the set of solutions in the k-th neighborhood of the current solution X. The simplest and most common choice is a structure in which the neighborhoods have increasing cardinality: $|N^{k_{min}}(X)| < |N^{k_{min}+1}(X)| < ... < |N^{k_{max}}(X)|$.

Given an incumbent X and an integer $k \in \{k_{\min}, \ldots, k_{\max}\}$ associated to a current neighborhood, shaking procedure generates a feasible solution in $N^k(X)$. Then a local search is applied around generated feasible solution in order to obtain a possibly better solution. If the local search gives a better solution then it becomes the new incumbent and in the standard VNS the next search begins at the first neighborhood $N^{k_{min}}$. Otherwise, the next neighborhood in the sequence is considered in order to try to improve upon the current solution. Should the last neighborhood $N^{k_{max}}$ be reached without a solution better than the incumbent being found, the search begins again at the first neighborhood $N^{k_{min}}$ until a stopping condition, e.g., a maximum number of iterations, is satisfied.

A detailed description of different VNS variants is out of the scope of this paper and can be found in (Hansen et al., 2008, 2010). An extensive computational experience on various optimization problems shows that VNS often produces high quality solutions in a reasonable time. Some of the recent applications are: mixed integer programming (Lazić et al., 2010), minimum labeling Steiner tree (Consoli et al., 2009), bandwidth reduction (Mladenović et al., 2010), uncapacitated single allocation p-hub median problem (Ilić et al., 2010) and uncapacitated multilevel lot-sizing problems (Xiao et al., 2011).

Algorithm 1: VNS pseudo code

```
Function VNS (k_min, k_max, iter_max, p_move)
B ← RSInit()
B' ← DeleteLast(B)
k ← k_min
iter ← 0
repeat
    iter ← iter + 1
    B'' ← Shaking(B', k)
    LocalSearch(B, B'')
    if Compare(B'', B', p_move) then B' ← B''
    else
        if  k < k_max then k ← k + 1
        else k ← k_min
    end
until iter ≤ iter_max
return B
```

Metric dimension. The VNS approach to MDP is based on the idea of decomposition described in Section 3. The initial set B is obtained by a simple procedure *RSInit*() which starts from the empty set and adds randomly chosen vertices from V until B becomes a resolving set. We set s to be equal to $|B| - 1$ and iteratively solve subproblems (9)-(12), decreasing s by one as long as the optimal objective function value is zero.

More precisely, for a given resolving set B we delete the last element using procedure *DeleteLast*(B) and obtain the set B'. Since in the implementation sets are represented as arrays, the last element of the set is the last element of the array.

The following steps are repeated until the stopping criterion is met. For a given k set B'' in $N^k(B')$ is obtained using the function `Shaking()`. Starting from B'' and B the local search procedure `LocalSearch()` tries to improve B'' and updates B whenever a new resolving set with smaller cardinality is generated. Within the function `Compare()` we compare set B' with set B'', update B'' if necessary, and continue VNS procedure. The pseudo-code of VNS implementation for solving MDP is given in Algorithm 1.

Algorithm 2: Pseudo code of the local search

```
Function LocalSearch (B, B'')
repeat
    impr ← false
    objval ← ObjF(B'')
    foreach v_r ∈ B'' do
        foreach v ∈ V \ B'' do z[v] ← 0
        LexSort(V, B'' \ {v_r})
        SetBL ← IdentifyBlocks(V, B'' \ {v_r})
        foreach BL ∈ SetBL do
            foreach p ∈ BL do
                foreach q ∈ BL with q > p do
                    foreach v ∈ V \ B'' do
                        if d(p, v) = d(q, v) then z[v] ← z[v] + 1
        v_min ← arg min {z[v]| v ∈ V \ B''}
        if z[v_min] = 0 then
            B ← B'' ∪ {v_min} \ {v_r}
            B'' ← DeleteLast(B)
            objval ← ObjF(B'')
            impr ← true
        else
            if z[v_min] < objval then
                B'' ← B'' ∪ {v_min} \ {v_r}
                objval ← z[v_min]
                impr ← true
until not impr
```

Neighborhoods and shaking. The neighborhood $N^k(B')$ contains all sets obtained from B' by deleting k of its elements and replacing them by k elements from $V \setminus B'$. It is clear that k must be less or equal to $|B'|$. It is easy to see that such neighborhoods have increasing cardinality, i.e. $|N^k| = \binom{s}{k} \cdot \binom{n-s}{k} < \binom{s}{k+1} \cdot \binom{n-s}{k+1} = |N^{k+1}|$ for every $k < \frac{ns-s^2-1}{n+2}$. Using the function *Shaking()*, for a given k, B'' is chosen randomly from $N^k(B')$.

Local search. In the local search procedure $LocalSearch()$, starting with B'' we interchange one element of set B'' with one element of its complement. We use the best improvement strategy, i.e. in every step we perform an interchange which gives the maximal decrease of the objective function. Whenever the improved set is a resolving set, the current set B is updated, the new set B'' is obtained by deleting the last element and the procedure continues. The procedure stops when there was no improvement.

The objective function value $objval = ObjF(B'')$ is computed as the number of pairs of vertices from V which have the same metric coordinates with respect to B''. In order to speed up computation of the objective function, instead of comparing metric coordinates for each two pairs of vertices from V, we sort vectors of metric coordinates in the lexicographical order. Then we can calculate the objective function value simply by searching the sorted list of vectors of metric coordinates.

In the straight-forward implementation of the best improvement strategy in the local search we have one call of the built-in sorting function *qsort* for each interchange, which gives total of $|B''| \cdot (|V| - |B''|)$ calls. We have obtained a significant speedup in the local search implementation by the following procedure, which requires only $|B''|$ calls of *qsort* in the best improvement strategy.

Let v_r denote the element of B'' which is a candidate for replacing with some vertex from $V \setminus B''$. Let z be an array which will store objective function values $ObjF(B'' \cup \{v\} \setminus \{v_r\})$ for each vertex $v \in V \setminus B''$. Initially we set $z[v] = 0, v \in V \setminus B''$.

We sort vectors of metric coordinates with respect to $B'' \setminus \{v_r\}$ in the lexicographical order (procedure *LexSort*()). Next, the blocks of vertices $SetBL$ with the same metric coordinates are identified by procedure *IdentifyBlocks()*. If a block BL consists of one vertex, the metric coordinates of that vertex are different from all other vertices from V and that vertex has no influence on the objective function, i.e. on the array z. If a block BL consists of two or more vertices then for every pair of vertices p, q from that block and $v \in V \setminus B''$ we increase $z[v]$ by one whenever $d(p, v) = d(q, v)$.

Finally, we determine the vertex v_{min} with minimal value $z[v_{min}]$. If $z[v_{min}] = 0$ then $B := B'' \cup \{v_{min}\} \setminus \{v_r\}$ is a new resolving set of smaller cardinality. The new set B'' is obtained by deleting the last element of B, and the local search continues. Otherwise, if $z[v_{min}] < ObjF(B'')$ we set $B'' := B'' \cup \{v_{min}\} \setminus \{v_r\}$ and continue the local search. If for each $v_r \in B''$ we have $z[v_{min}] \geq ObjF(B'')$, the local search ends with no improvement.

The described local search algorithm can be formally presented as the pseudo code given in Algorithm 2.

Neighborhood change. After the local search procedure we have three possibilities. Within the function $Compare()$ we decide whether to move to the solution B'' or stay in the current solution B':

- In the case when the solution B'' is better than B', i.e. $|B''| < |B'|$ or $ObjF(B'') < ObjF(B')$ we set $B' := B''$ and continue the search with the same neighborhood N^k;
- If $|B''| = |B'|$ and $ObjF(B'') > ObjF(B')$ then we repeat the search with the same B' and the next neighborhood;
- If $|B''| = |B'|$ and $ObjF(B'') = ObjF(B')$ then with probability p_{move} we set $B' := B''$ and continue search with the same neighborhood N^k and with probability $1 - p_{move}$ we repeat search with the same B' and the next neighborhood.

Discussion. The described VNS approach tries to minimize number of pairs of vertices with the same metric coordinates with respect to the set B' ($|B'| = |B| - 1$). In the local search procedure we interchange one element of set B'' with one element of its complement. The interchange procedure assumes that $|B'| - 1 = |B''| - 1 \geq 1$. Therefore, our VNS approach for MDP can be applied only for graphs with metric dimension at least three. This is not a serious drawback since the case when the metric dimension is at most two has been theoretically characterized in (Sudhakara & Kumar, 2009). Moreover, the complexity of solving the MDP by total enumeration in this case is $O(n^2)$.

The following example illustrates the local search procedure on the graph from Example 2.

Example 5 *Let $B = \{v_1, v_2, v_4, v_6\}$ be the initial resolving set and $B' = \{v_1, v_2, v_4\}$. Suppose that after the shaking step with $k = 1$ we have $B'' = \{v_1, v_3, v_4\}$. The corresponding objective function value is $ObjF(B'') = 1$ since all pairs of vectors of metric coordinates are different except one: $r(v_2, B'') = r(v_6, B'') = (1, 2, 1)$. We try to exchange e.g. $v_r = v_1$ with one element $v \in V \setminus B''$ using the best improvement strategy. After sorting the vectors of metric coordinates with respect to $B'' \setminus v_r = \{v_3, v_4\}$ in the lexicographical order and identifying the blocks of vertices with the same metric coordinates, we obtain array z which stores objective function values $ObjF(B'' \cup \{v\} \setminus \{v_r\}) = ObjF(\{v_3, v_4\} \cup \{v\})$ for each vertex $v \in V \setminus B'' = \{v_2, v_5, v_6\}$. The results of this process are displayed in Table 1. Since blocks $\{v_3\}, \{v_4\}, \{v_5\}, \{v_1\}$ consist of only one vertex, they have no influence on array z and entries in the corresponding rows are omitted. As in the array z all entries are zero, v_{min} can be any of the vertices v_2, v_5, v_6. For example, if $v_{min} = v_2$ a new resolving set $B = \{v_2, v_3, v_4\}$ is obtained. After removing the last element we obtain $B'' = \{v_2, v_3\}$. The corresponding*

Table 1: Local search in Example 5 for $B'' = \{v_1, v_3, v_4\}$

	v_3	v_4	v_2	v_5	v_6
v_3	0	1			
v_4	1	0			
v_5	1	1			
v_1	1	2			
v_2	2	1	0	2	1
v_6	2	1	1	1	0
z			0	0	0

Table 2: Local search in Example 5 for $B'' = \{v_2, v_3\}$

	v_3	v_1	v_4	v_5	v_6
v_3	0				
v_1	1	0	2	1	1
v_4	1	2	0	1	1
v_5	1	1	1	0	1
v_2	2	1	1	2	1
v_6	2	1	1	1	0
z		1	1	1	3

objective function value is $ObjF(B'') = 1$ since we have one pair of vectors of metric coordinates that are the same: $r(v_1, B'') = r(v_4, B'') = (1, 1)$. We try to exchange e.g. $v_r = v_2$ with one element $v \in V \setminus B''$ using the best improvement strategy. After sorting the vectors of metric coordinates with respect to $B'' \setminus v_r = \{v_3\}$ in the lexicographical order and identifying the blocks of vertices with the same metric coordinates, we obtain array z which stores objective function values $ObjF(B'' \cup \{v\} \setminus \{v_r\}) = ObjF(\{v_3\} \cup \{v\})$ for each vertex $v \in V \setminus B'' = \{v_1, v_4, v_5, v_6\}$. The results of this process are displayed in Table 2. Since the block $\{v_3\}$ consist of only one vertex, it has no influence on array z and entries in the corresponding row are omitted. As in the array z all entries except last are one, v_{min} can be any of the vertices v_1, v_4, v_5 and $z(v_{min}) = 1 = ObjF(B'')$ and local search procedure stops since there was no improvement. Note that $\beta(G_2) = 3$, and therefore the objective function value could not be further improved with any shaking and/or local search step.

VNS for Minimal doubly resolving set. As the MDRSP is closely related to the MDP the described VNS approach can be easily accommodated to solve the MDRSP. The differences occur in the functions: $RSInit$, $ObjF$ and $LocalSearch$. In each of the functions checking whether the current solution is a resolving set or not is replaced by doubly resolving set checking. In order to increase the efficiency of this identification we use the results of the following proposition.

Proposition 4 *(Kratica et al., 2009b) A subset $D = \{x_1, x_2, ..., x_k\} \subseteq V$ is a doubly resolving set of G if and only if for every $p, q \in V$ there exists $i \in \{1, 2, ..., k\}$ such that*

$$d(p, x_i) - d(p, x_1) \neq d(q, x_i) - d(q, x_1) \tag{30}$$

For each $v \in V$ let $r'(v, D) = (d(v, x_2) - d(v, x_1), ..., d(v, x_k) - d(v, x_1))$. According to Proposition 4 it is sufficient to exchange r with r' and apply a procedure which checks whether D is a resolving set, using vectors r' instead of r. Using this observation, the previous VNS approach for MDP has been effectively adapted to solve MDRSP. It should be noted that this VNS approach can be applied only for graphs with the cardinality of the minimal doubly resolving set at least four. Similarly as for MDP, this is not a serious drawback.

Another interesting related problem is the strong metric dimension problem (SMDP), introduced in (Sebo & Tannier, 2004). A genetic algorithm for SMDP is presented in (Kratica et al., 2008). That problem could also be tackled by VNS. However, a strongly resolving set cannot be identified by sorting vertices as in the case of a resolving set or a doubly resolving set. Therefore, a straightforward adaptation of the described VNS approach for SMDP is not possible.

5 Experimental results

This section presents the results of VNS approach to MDP and MDRSP on various classes of graph instances which have already been tackled by GA approach in (Kratica et al., 2009a,b): pseudo boolean, crew scheduling, graph coloring, hypercubes and Hamming graphs.

All tests were performed on an Intel 2.5 GHz single processor with 1GB memory, under Windows XP operating system. In our experiments we have used the following values for VNS parameters $k_{min} = 2$, $k_{max} = 20$, $p_{move} = 0.2$, $iter_{max} = 100$. The VNS has been run 20 times for each instance and the results are summarized in Tables 3-8. The tables are organized as follows:

- the first three columns contain the test instance name, the corresponding number of vertices and edges respectively;
- the next four columns contain the results of VNS performance for MDP and the comparison with GA from (Kratica et al., 2009a). The fifth and the sixth column contain the cardinality, named *best*, of the best resolving set obtained by GA in 20 runs and average GA running time, named t. The seventh and the eight column contain the experimental VNS results, presented in the same way as for GA. The solutions that are known to be optimal are bolded and underlined;
- the results for MDRSP are presented in the last four columns and organized in the same way as the data for MDP.

For large-scale hypercubes Q_r, $r \geq 13$, the distance matrix can not fit in the memory. In order to overcome this obstacle we have developed a special VNS for MDP and MDRSP on hypercubes. Instead of generating and memorizing whole distance matrix, this VNS computes the distance between two vertices each time it is needed. The code has been optimized using the special structure of hypercubes. The results of one run on hypercubes Q_8 - Q_{17} are presented in Table 9, which is organized in the same way as Tables 3–8.

As can be seen from Tables 3–9, VNS produces much better results on both MDP and MDRSP than GA approach described in (Kratica et al., 2009a,b). More precisely, results of VNS are never worse than GA results and in most of the cases the GA upper bound is significantly improved. For all instances where the metric dimension or the cardinality of a minimal doubly resolving set is known, VNS reaches their values.

Results for crew scheduling ORLIB instances reported in Table 3 show that VNS for MDP has improved GA results in 8 out of 10 cases. The best improvement is achieved for csp400, where the upper bound for the metric dimension is reduced from 26 to 21. VNS for MDRSP is better than GA in 9 out of 10 cases, with the best improvement from 28 to 22. In all cases the running time of VNS is significantly smaller than GA. For example, VNS running time for csp500 is less than 8 seconds, while GA running time is about 157 seconds.

VNS results on graph coloring instances in Table 4 show that the GA upper bounds has been improved by one for MDP and one/two for MDRSP. The running time of VNS is again significantly smaller. Table 5 summarizes results on pseudo boolean instances. In all 40 cases VNS for MDP is better than GA with the improvement ranging from 2 to 10. Similarly, VNS for MDRSP outperforms GA in all cases, with the improvement range from 4 to 15. For both problems the VNS running time at least five times smaller.

Tables 7 and 8 contain results on Hamming graphs. For graphs $H_{2,k}, 3 \leq k \leq 30$, $H_{3,3}$, $H_{3,4}$, $H_{4,3}$ both GA and VNS reach their metric dimensions, previously known from literature. For other Hamming graphs, for which the metric dimension is not known, VNS has improved GA upper bounds in 15 out of 24 cases, with the improvement range 1-3. For MDRSP, VNS has improved GA results in 17 out of 55 cases. The running time of VNS is again considerably smaller than GA.

Since average values of VNS are usually better than the best GA results, we omitted columns with average behaviour of VNS. We give here the details for MDP results only for largest instances from all tables. For csp500 from Table 3, average VNS value is 25.1 while the best GA value is 29. For gcol21-gcol30 instances from Table 4, the worst average VNS value is 11.55, while the best GA value is 12. For the largest frb instances from Table 5, the worst average VNS value is 39.65, while the best GA value is 47. Hypercubes and Hamming graphs, due to symmetry, are easier both for VNS and GA and the results are closer. Nevertheless, for $H_{6,4}$ average VNS value is 8.25, while the best GA value is 9. For MDRSP similar average behaviour can be observed.

Tables 6 and 9 present results for hypercubes. For hypercubes $Q_r, 3 \leq r \leq 10$, both GA and VNS obtain the metric dimensions, previously known from the literature. In other 7 cases, the GA bound is improved by one for Q_{13}, Q_{15},Q_{16} and Q_{17}. For MDRSP, VNS has improved GA results by one in 4 out of 15 cases. Note that in the case of large-scale hypercubes ($r \geq 13$) the VNS running time is larger than GA running time for both MDP and MDRSP. The reason is the fact that for graphs with huge number of vertices ($n \geq 2^{13} = 8192$) the local search procedure is time consuming. In three cases, denoted in Table 9 by asterisk, VNS was stopped after one day running time.

Table 3: VNS results on crew scheduling ORLIB instances

Inst.	n	m	MDP				$MDRSP$			
			GA		VNS		GA		VNS	
			$best$	t	$best$	t	$best$	t	$best$	t
csp50	50	173	8	2.764	8	0.035	11	5.065	11	0.046
csp100	100	715	11	6.297	11	0.172	12	6.642	11	0.192
csp150	150	1355	15	14.016	13	0.404	15	12.932	13	0.463
csp200	200	2543	16	19.974	14	0.761	17	21.131	15	0.889
csp250	250	4152	18	25.421	15	1.240	19	26.922	16	1.414
csp300	300	6108	23	41.578	19	2.051	25	48.569	20	2.407
csp350	350	7882	23	47.615	20	2.814	24	49.850	20	3.253
csp400	400	10760	26	79.810	21	3.914	28	88.577	22	4.518
csp450	450	13510	27	112.939	23	4.996	27	116.412	23	5.819
csp500	500	16695	29	157.422	24	6.502	29	157.759	25	7.620

Table 4: VNS results on graph coloring ORLIB instances

Inst.	n	m	MDP				$MDRSP$			
			GA		VNS		GA		VNS	
			$best$	t	$best$	t	$best$	t	$best$	t
gcol1	100	2487	9	4.411	8	0.153	9	4.603	8	0.153
gcol2	100	2487	9	4.205	8	0.149	9	4.059	8	0.153
gcol3	100	2482	9	4.798	8	0.149	9	4.263	8	0.158
gcol4	100	2503	9	5.228	8	0.150	9	4.812	8	0.150
gcol5	100	2450	9	4.264	8	0.152	9	4.526	8	0.150
gcol6	100	2537	9	4.309	8	0.153	9	4.374	8	0.153
gcol7	100	2505	9	4.804	8	0.153	9	4.550	8	0.157
gcol8	100	2479	9	4.693	8	0.153	9	5.039	8	0.154
gcol9	100	2486	9	4.186	8	0.154	9	4.091	8	0.156
gcol10	100	2506	9	4.370	8	0.149	9	4.443	8	0.153
gcol11	100	2467	9	4.211	8	0.151	9	4.144	8	0.148
gcol12	100	2531	9	3.958	8	0.152	9	3.833	8	0.153
gcol13	100	2467	9	4.479	8	0.153	9	4.279	8	0.159
gcol14	100	2524	9	4.694	8	0.149	9	4.385	8	0.157
gcol15	100	2528	9	4.227	8	0.153	9	4.244	8	0.151
gcol16	100	2493	9	4.245	8	0.156	9	3.986	8	0.148
gcol17	100	2503	9	4.967	8	0.146	9	4.644	8	0.150
gcol18	100	2472	9	4.428	8	0.149	9	4.527	8	0.153
gcol19	100	2527	9	4.728	8	0.152	9	4.117	8	0.150
gcol20	100	2420	9	4.615	8	0.155	9	4.349	8	0.154
gcol21	300	22482	12	16.883	11	1.182	12	15.607	11	1.231
gcol22	300	22569	12	16.325	11	1.171	12	16.039	11	1.218
gcol23	300	22393	12	16.474	11	1.178	13	15.679	11	1.255
gcol24	300	22446	12	17.018	11	1.186	12	16.997	11	1.224
gcol25	300	22360	12	16.324	11	1.192	12	15.646	11	1.243
gcol26	300	22601	12	16.167	11	1.186	12	16.102	11	1.232
gcol27	300	22327	12	17.005	11	1.175	13	15.759	11	1.222
gcol28	300	22472	12	16.519	11	1.184	13	15.619	11	1.214
gcol29	300	22520	12	16.958	11	1.170	12	16.584	11	1.228
gcol30	300	22543	12	16.462	11	1.186	12	15.878	11	1.207

Table 5: VNS results on pseudo boolean instances

Inst.	n	m	*MDP*				*MDRSP*			
			GA		*VNS*		*GA*		*VNS*	
			best	*t*	*best*	*t*	*best*	*t*	*best*	*t*
frb30-15-1	450	17827	24	55.410	20	4.679	25	37.248	21	5.392
frb30-15-2	450	17874	23	70.027	21	4.678	26	41.524	20	5.357
frb30-15-3	450	17809	24	73.131	21	4.739	28	45.207	21	5.474
frb30-15-4	450	17831	25	62.235	20	4.696	27	41.608	20	5.434
frb30-15-5	450	17794	25	58.981	20	4.804	25	40.886	20	5.502
frb35-17-1	595	27856	28	140.927	23	9.011	30	109.885	24	10.620
frb35-17-2	595	27847	28	131.696	24	8.957	31	118.449	23	10.517
frb35-17-3	595	27931	28	121.679	24	8.721	31	91.654	24	9.986
frb35-17-4	595	27842	27	139.799	23	8.505	29	101.825	23	10.040
frb35-17-5	595	28143	28	149.224	23	8.691	29	111.843	23	10.172
frb40-19-1	760	41314	32	219.515	26	15.289	35	214.606	27	18.472
frb40-19-2	760	41263	32	226.957	27	16.100	36	191.55	27	19.088
frb40-19-3	760	41095	32	212.460	27	16.167	34	178.413	27	18.891
frb40-19-4	760	41605	33	198.968	27	16.107	35	168.067	27	19.264
frb40-19-5	760	41619	32	240.240	27	16.117	35	196.174	27	19.188
frb45-21-1	945	59186	36	343.935	30	28.774	37	309.096	30	36.485
frb45-21-2	945	58624	37	320.295	30	30.944	41	288.275	31	37.517
frb45-21-3	945	58245	36	338.475	30	30.045	38	290.948	30	36.574
frb45-21-4	945	58549	38	357.220	30	30.255	38	282.366	30	37.175
frb45-21-5	945	58579	36	357.336	30	31.157	39	296.772	30	38.434
frb50-23-1	1150	80072	41	425.927	33	51.668	43	354.589	33	64.661
frb50-23-2	1150	80851	40	490.511	32	53.123	45	386.198	33	65.820
frb50-23-3	1150	81068	40	535.953	34	55.554	43	372.89	34	67.020
frb50-23-4	1150	80258	40	502.052	34	55.533	43	417.271	34	68.054
frb50-23-5	1150	80035	40	488.454	34	56.176	44	419.106	34	69.353
frb53-24-1	1272	94227	42	635.440	34	68.666	45	521.004	34	84.995
frb53-24-2	1272	94289	41	657.049	35	71.022	47	546.576	35	88.628
frb53-24-3	1272	94127	44	592.430	35	71.011	46	526.495	36	88.643
frb53-24-4	1272	94308	43	597.553	36	71.868	44	548.656	36	90.307
frb53-24-5	1272	94226	43	608.632	36	72.114	45	511.89	36	89.293
frb56-25-1	1400	109676	45	739.162	36	89.995	49	591.043	36	109.515
frb56-25-2	1400	109401	43	690.399	37	90.821	50	525.07	36	109.323
frb56-25-3	1400	109379	45	777.074	37	90.945	51	588.939	36	113.454
frb56-25-4	1400	110038	47	669.820	37	90.449	50	623.492	37	110.670
frb56-25-5	1400	109601	45	745.054	37	91.263	50	679.74	37	112.842
frb59-26-1	1534	126555	47	881.819	38	113.916	51	698.808	38	139.759
frb59-26-2	1534	126163	47	846.511	38	112.644	51	786.456	38	133.025
frb59-26-3	1534	126082	47	888.085	39	115.607	52	716.212	39	136.703
frb59-26-4	1534	127011	47	845.265	38	111.652	51	660.34	38	133.021
frb59-26-5	1534	125982	48	814.813	39	115.925	50	718.742	39	140.197

Table 6: VNS results on hypercubes

Inst.	n	m	*MDP*				*MDRSP*			
			GA		*VNS*		*GA*		*VNS*	
			best	*t*	*best*	*t*	*best*	*t*	*best*	*t*
Q_1	2	1	1	<0.001	-	-	2	0.001	-	-
Q_2	4	4	2	0.042	-	-	3	0.001	-	-
Q_3	8	12	3	0.082	3	0.001	4	0.157	4	0.001
Q_4	16	32	4	0.393	4	0.003	4	0.399	4	0.006
Q_5	32	80	4	0.767	4	0.011	5	0.903	5	0.018
Q_6	64	192	5	1.932	5	0.046	6	2.037	6	0.055
Q_7	128	448	6	4.595	6	0.158	6	4.15	6	0.324
Q_8	256	1024	6	13.762	6	0.892	7	13.546	7	0.954
Q_9	512	2304	7	77.522	7	2.895	7	73.647	7	6.440
Q_{10}	1024	5120	7	217.574	7	19.885	8	219.807	8	19.615
Q_{11}	2048	11264	8	601.708	8	69.528	8	632.325	8	137.291
Q_{12}	4096	24576	8	1668.568	8	464.351	9	1543.678	8	951.600

Table 7: VNS results on Hamming graphs

Inst.	n	m	*MDP*				*MDRSP*			
			GA		*VNS*		*GA*		*VNS*	
			best	t	*best*	t	*best*	t	*best*	t
$H_{2,3}$	9	18	3	0.012	3	0.001	3	0.013	-	-
$H_{2,4}$	16	48	4	0.414	4	0.002	5	0.368	5	0.005
$H_{2,5}$	25	100	6	0.764	6	0.012	6	0.78	6	0.006
$H_{2,6}$	36	180	7	1.168	7	0.018	7	1.179	7	0.020
$H_{2,7}$	49	294	8	1.257	8	0.042	8	1.263	8	0.039
$H_{2,8}$	64	448	10	2.378	10	0.070	10	2.449	10	0.075
$H_{2,9}$	81	648	11	2.886	11	0.122	11	2.915	11	0.134
$H_{2,10}$	100	900	12	3.986	12	0.196	12	4.111	12	0.221
$H_{2,11}$	121	1210	14	5.153	14	0.300	14	5.107	14	0.343
$H_{2,12}$	144	1584	15	7.460	15	0.451	15	7.637	15	0.519
$H_{2,13}$	169	2028	16	10.156	16	0.663	16	10.529	16	0.768
$H_{2,14}$	196	2548	18	13.667	18	0.908	18	13.285	18	1.078
$H_{2,15}$	225	3150	19	18.443	19	1.276	19	21.159	19	1.521
$H_{2,16}$	256	3840	20	25.074	20	1.773	20	25.676	20	2.117
$H_{2,17}$	289	4624	22	33.520	22	2.298	22	31.861	22	2.771
$H_{2,18}$	324	5508	23	43.487	23	3.028	23	45.658	23	3.650
$H_{2,19}$	361	6498	24	68.759	24	3.966	24	65.359	24	4.813
$H_{2,20}$	400	7600	26	102.586	26	4.941	26	101.699	26	6.055
$H_{2,21}$	441	8820	27	132.269	27	6.338	27	135.887	27	7.629
$H_{2,22}$	484	10164	28	167.221	28	7.858	28	177.343	28	9.800
$H_{2,23}$	529	11638	30	230.036	30	9.481	30	213.27	30	11.864
$H_{2,24}$	576	13248	31	262.499	31	11.814	31	261.502	31	15.021
$H_{2,25}$	625	15000	32	336.314	32	14.533	33	344.89	32	18.710
$H_{2,26}$	676	16900	34	412.840	34	17.393	34	372.812	34	22.935
$H_{2,27}$	729	18954	35	500.236	35	23.568	35	526.236	35	28.479
$H_{2,28}$	784	21168	36	497.301	36	28.778	37	555.582	36	35.043
$H_{2,29}$	841	23548	38	596.084	38	31.791	39	643.672	38	42.940
$H_{2,30}$	900	26100	39	638.735	39	42.491	39	773.202	39	53.728

Let us point out that the improvement of the upper bound for the metric dimension of Q_{15} implies the improvement of the theoretical upper bound $\beta(Q_r) \leq r - 5$ for $r \geq 15$ from (Cáceres et al., 2007). Namely, the following proposition holds:

Proposition 5 $\beta(Q_r) \leq r - 6$ *for* $r \geq 15$

Proof. According to Proposition 1

$$\beta(Q_r) = \beta(Q_{r-1} \Box Q_1) \leq \beta(Q_{r-1}) + \psi(Q_1) - 1 = \beta(Q_{r-1}) + 1 \tag{31}$$

Since $\beta(Q_{15}) \leq 9$ then $\beta(Q_r) \leq r - 6$ for all $r \geq 15$. □

Comparison of the new ILP model (9)-(12) with the existing ILP model (3), (7), (5), was performed using CPLEX 12.1 and Gurobi 3.0 on smaller crew scheduling and graph coloring instances. In order to make a fair comparison, we have added to the existing ILP model (3), (7), (5) a new constraint (29), which bounds the cardinality of resolving sets with some given upper bound ub.

$$\sum_{j=1}^{n} y_j < ub \tag{32}$$

If CPLEX and/or Gurobi have found an optimal solution of the ILP model (3), (7), (5), (29), this value is the metric dimension and it is obviously less than ub. On the other hand, if model (3), (7), (5), (29) does not have a feasible solution, then ub is the metric dimension.

For the new ILP model, we set $s = ub - 1$ and iteratively solve subproblems (9)-(12) by both CPLEX and Gurobi, decreasing s by one as long as the optimal objective function value is zero. If this value is not zero, the metric dimension is equal to $s + 1$.

Table 8: VNS results on Hamming graphs

Inst.	n	m	MDP				$MDRSP$			
			GA		VNS		GA		VNS	
			best	t	*best*	t	*best*	t	*best*	t
$H_{3,3}$	27	81	4	0.602	4	0.008	4	0.625	4	0.013
$H_{3,4}$	64	288	6	1.937	6	0.040	6	1.956	6	0.044
$H_{3,5}$	125	750	7	3.905	7	0.175	7	3.942	7	0.193
$H_{3,6}$	216	1620	9	9.588	9	0.500	9	9.601	9	0.539
$H_{3,7}$	343	3087	11	21.548	10	1.482	11	21.067	10	1.652
$H_{3,8}$	512	5376	12	89.063	12	3.739	12	89.396	12	4.143
$H_{3,9}$	729	8748	14	132.708	13	8.453	14	133.122	13	9.715
$H_{3,10}$	1000	13500	16	226.388	15	18.008	16	245.427	15	20.773
$H_{3,11}$	1331	19965	18	413.829	16	36.289	18	394.249	16	41.572
$H_{3,12}$	1728	28512	19	562.828	18	74.154	19	559.288	18	84.736
$H_{3,13}$	2197	39546	21	827.800	19	120.992	21	879.626	19	137.738
$H_{3,14}$	2744	53508	23	1346.247	21	206.352	23	1363.252	21	240.385
$H_{3,15}$	3375	70875	24	1917.173	22	338.940	24	1851.26	22	393.890
$H_{3,16}$	4096	92160	27	2880.383	24	573.980	26	3170.444	24	663.147
$H_{3,17}$	4913	117912	28	3661.674	25	883.357	28	3948.072	25	993.114
$H_{4,3}$	81	324	5	2.744	5	0.062	5	2.797	5	0.110
$H_{4,4}$	256	1536	7	13.908	7	0.600	7	12.172	7	0.627
$H_{4,5}$	625	5000	9	73.462	8	5.622	9	66.995	8	5.748
$H_{4,6}$	1296	12960	11	247.111	10	26.567	11	238.93	10	29.222
$H_{4,7}$	2401	28812	13	661.861	12	107.429	13	663.141	12	116.753
$H_{4,8}$	4096	57344	15	1715.241	14	405.037	15	1785.401	14	431.946
$H_{5,3}$	243	1215	5	10.871	5	1.090	5	10.863	5	1.929
$H_{5,4}$	1024	7680	8	211.686	8	12.051	8	205.811	8	13.609
$H_{5,5}$	3125	31250	10	1005.747	10	164.445	10	877.189	10	175.898
$H_{6,3}$	729	4374	6	74.924	6	7.822	6	78.964	6	14.329
$H_{6,4}$	4096	36864	9	1400.755	8	376.677	9	1491.185	9	335.454
$H_{7,3}$	2187	15309	7	440.175	7	79.295	7	447.028	7	110.884

Table 9: Results of special VNS algorithm on hypercubes

Inst.	n	m	MDP				$MDRSP$			
			GA		VNS		GA		VNS	
			best	t	*best*	t	*best*	t	*best*	t
Q_8	256	1024	6	17.25	6	1.016	7	14.034	7	1.169
Q_9	512	2304	7	51.96	7	2.896	7	33.613	7	7.884
Q_{10}	1024	5120	7	113.95	7	18.332	8	78.261	8	20.012
Q_{11}	2048	11264	8	258.35	8	48.85	8	196.800	8	141.898
Q_{12}	4096	24576	8	637.32	8	308.85	9	403.458	8	896.054
Q_{13}	8192	53248	9	1378.95	8	1970.98	9	980.312	9	2019.484
Q_{14}	16384	114688	9	2524.72	9	4841.12	10	1940.877	9	13511
Q_{15}	32768	245760	10	5414.69	9	31262	10	4752.388	10	26505
Q_{16}	65536	524288	11	15321	10	66831	11	10873	10	86400*
Q_{17}	131072	11114112	11	34162	10	86400*	12	24356	11	86400*

Since the best existing upper bounds are VNS results presented in Tables 3 and 4, they have been used for value of ub. This enables us to additionally evaluate the quality of VNS solutions by checking that CPLEX or Gurobi is unable to improve ub.

The experimental results for MDP are summarized in Table 10, which is organized as follows:

- the first two columns contain the test instance name and the optimal value, if it has been found;
- the next four columns contain the results related to the ILP model (3), (7), (5), (29). The third and fifth column contain the values obtained by CPLEX and Gurobi while the fourth and sixth column represent the corresponding running times in seconds. Bold and underlined results mean that the optimality has been proved;
- the last four columns contain the results for the ILP model (9)-(12), and are organized in a similar way.

Table 10: CPLEX and Gurobi results on small instances

Inst.	*opt*	*existing ILP*				*new ILP*			
		CPLEX		*Gurobi*		*CPLEX*		*Gurobi*	
		res	*t*	*res*	*t*	*res*	*t*	*res*	*t*
MDP									
csp50	8	**8**	0.3	**8**	0.4	**8**	1.17	**8**	1.70
csp100	11	**11**	497	**11**	628	**11**	7154	**11**	6217
csp150	13	**13**	8603	**13**	6047	**13**	93796	**13**	94233
csp200	-	14	296804	14	232443	14	306164	14	89675
gcol1	-	8	148213	8	233077	8	96953	8	128107
gcol2	-	8	53268	8	233077	8	51238	8	67260
MDRSP									
csp50	11	**11**	4933	**11**	1660	**11**	22493	**11**	56591

Since MDRSP is harder than MDP exact methods can deal only with instance csp50. Results of CPLEX and Gurobi are given in the last row of Table 10.

For larger crew scheduling and graph coloring instances CPLEX and Gurobi either run out of memory or the running times significantly exceed one day. It is interesting to note that performance of CPLEX and Gurobi on the new ILP models for MDP and MDRSP is slightly worse than on the existing ones with respect to the running time. The situation with metaheuristics is quite different. Namely, the decomposition approach within VNS is superior to GA which is based on the objective function of models (3), (7), (5) and (16)–(20).

6 Conclusions

In this paper an efficient variable neighborhood search approach to solving the metric dimension problem and the problem of determining minimal doubly resolving sets is presented. VNS apparoach is based on a decomposition of MDP, i.e. MDRSP, into a sequence of subproblems with an auxiliary objective function. Also, for both problems the corresponding new integer linear programming formulations are given.

The new objective function calculates the number of vertices that are not resolved (doubly resolved). The neighborhood structures defined on these models allow effective shaking procedure. Local search is implemented very efficiently which results in excellent overall VNS performance.

An extensive experimental comparison with the only existing heuristic approach based on genetic algorithm indicates superiority of VNS approach with respect to both solution quality and computation time.

This research can be extended in several ways. It would be challenging to investigate application of the presented VNS approach to similar problems on graphs. Also, computational results can be used to generate theoretical hypotheses about the metric dimension and the cardinality of minimal doubly resolving sets for some special classes of graphs.

References

Bailey, R., & Cameron, P. (2011). Base size, metric dimension and other invariants of groups and graphs. *Bulletin of the London Mathematical Society*, *43*, 209–242.

Bailey, R., & Meagher, K. (2011). *On the metric dimension of Grassmann graphs*. Technical Report arXiv:1010.4495.

Beerliova, Z., Eberhard, F., Erlebach, T., Hall, A., Hoffmann, M., Mihalák, M., & Ram, L. (2006). Network discovery and verification. *IEEE Journal on Selected Areas in Communications*, *24*, 2168–2181.

Cáceres, J., Hernando, C., Mora, M., Pelayo, I., & Puertas, M. (2009). On the metric dimension of infinite graphs. *Electronic Notes in Discrete Mathematics*, *35*, 15–20.

Cáceres, J., Hernando, C., Mora, M., Pelayo, I., Puertas, M., Seara, C., & Wood, D. (2007). On the metric dimension of cartesian products of graphs. *SIAM Journal in Discrete Mathematics*, *21*, 423–441.

Chappell, G., Gimbel, J., & Hartman, C. (2008). Bounds on the metric and partition dimensions of a graph. *Ars Combinatoria*, *88*, 349–366.

Chartrand, G., Eroha, L., Johnson, M., & Oellermann, O. (2000). Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics*, *105*, 99–113.

Consoli, S., Darby-Dowman, K., Mladenović, N., & Moreno-Pérez, J. (2009). Variable neighbourhood search for the minimum labelling steiner tree problem. *Annals of the Operations Research*, *172*, 71–96.

Currie, J., & Oellerman, O. (2001). The metric dimension and metric independence of a graph. *Journal of Combinatorial Mathematics and Combinatorial Computing*, *39*, 157–167.

Fehr, M., Gosselin, S., & Oellermann, O. (2006). The metric dimension of cayley digraphs. *Discrete Mathematics*, *306*, 31–41.

Garcia, S., Labbe, M., & Marin, A. (2011). Solving large p-median problems with a radius formulation. *Informs Journal on Computing*, *in press*. Doi:10.1287/ijoc.1100.0418.

Hansen, P., Mladenović, N., & Moreno-Pérez, J. (2008). Variable neighbourhood search: methods and applications (invited survey). *4OR: A Quarterly Journal of Operations Research*, *6*, 319–360.

Hansen, P., Mladenović, N., & Moreno-Pérez, J. (2010). Variable neighbourhood search: algorithms and applications. *Annals of Operations Research*, *175*, 367–407.

Harary, F., & Melter, R. (1976). On the metric dimension of a graph. *Ars Combinatoria*, *2*, 191–195.

Hauptmann, M., Schmied, R., & Viehmann, C. (2011). On approximation complexity of metric dimension problem. *Lecture Notes in Computer Science*, *6460*, 136–139.

Hernando, C., Mora, M., I.M.Pelayo, Seara, C., Cáceres, J., & Puertas, M. (2005). On the metric dimension of some families of graphs. *Electronic Notes in Discrete Mathematics*, *22*, 129–133.

Hernando, C., Mora, M., Pelayo, I., Seara, C., & Wood, D. (2007). Extremal graph theory for metric dimension and diameter. *Electronic Notes in Discrete Mathematics*, *29*, 339–343.

Hernando, C., Mora, M., Pelayo, I., Seara, C., & Wood, D. (2010). Extremal graph theory for metric dimension and diameter. *Electronic Journal of Combinatorics*, *17*, 1–28.

Husnine, S., & Kousar, I. (2010). A subfamily of a generalized petersen graph p(n,3) with constant metric dimension. *Utilitas Mathematica*, *81*, 111–120.

Ilić, A., Urošević, D., Brimberg, J., & Mladenović, N. (2010). A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. *European Journal of Operational Research*, *206*, 289–300.

Imran, M., Bokhary, S. U. H., & Baig, A. (2010). On families of convex polytopes with constant metric dimension. *Computers and Mathematics with Applications*, *60*, 2629–2638.

Javaid, I., Rahim, M., & Ali, K. (2008). Families of regular graphs with constant metric dimension. *Utilitas Mathematica*, *75*, 21–33.

Khuller, S., Raghavachari, B., & Rosenfeld, A. (1996). Landmarks in graphs. *Discrete Applied Mathematics*, *70*, 217–229.

Kochetov, Y., Kononova, P., & Paschenko, M. (2008). Formulation space search approach for the teacher/class timetabling problem. *Yugoslav Journal of Operations Research*, *18*, 1–11.

Kratica, J., Kovačević-Vujčić, V., & Čangalović, M. (2008). Computing strong metric dimension of some special classes of graphs by genetic algorithms. *Yugoslav Journal of Operations Research*, *18*, 143–151.

Kratica, J., Kovačević-Vujčić, V., & Čangalović, M. (2009a). Computing the metric dimension of graphs by genetic algorithms. *Computational Optimization and Applications*, *44*, 343–361.

Kratica, J., Čangalović, M., & Kovačević-Vujčić, V. (2009b). Computing minimal doubly resolving sets of graphs. *Computers & Operations Research*, *36*, 2149–2159.

Lazić, J., Hanafi, S., Mladenović, N., & Urošević, D. (2010). Variable neighbourhood decomposition search for 01 mixed integer programs. *Computers & Operations Research*, *37*, 1055–1067.

Liu, K., & Abu-Ghazaleh, N. (2006). *Virtual coordinate backtracking for void traversal in geographic routing*. Technical Report arXiv:cs.NI/0511059 v2.

Manuel, P., & Rajasingh, I. (2011). Minimum metric dimension of silicate networks. *Ars Combinatoria*, *98*, 501–510.

Mladenović, N., & Hansen, P. (1997). Variable neighbourhood search. *Computers & Operations Research*, *24*, 1097–1100.

Mladenović, N., Plastria, F., & Urosević, D. (2007). Formulation space search for circle packing problems. *Lecture Notes in Computer Science*, *4638*, 212–216.

Mladenović, N., Plastria, F., & Urošević, D. (2005). Reformulation descent applied to circle packing problems. *Computers & Operations Research*, *32*, 2419–2434.

Mladenović, N., Urošević, D., Pérez-Brito, D., & Garca-González, C. (2010). Variable neighbourhood search for bandwidth reduction. *European Journal of Operational Research*, *200*, 14–27.

Peters-Fransen, J., & Oellermann, O. (2006). The metric dimension of cartesian products of graphs. *Utilitas Mathematica*, *69*, 33–41.

Rebatel, F., & Thiel, E. (2011). Metric bases for polyhedral gauges. *Lecture Notes in Computer Science*, *6607*, 116–128.

Sebo, A., & Tannier, E. (2004). On metric generators of graphs. *Mathematics & Operations Research*, *29*, 383–393.

Slater, P. (1975). Leaves of trees. *Congr. Numerantium*, *14*, 549–559.

Sudhakara, G., & Kumar, A. H. (2009). Graphs with metric dimension two - a characterization. *World Academy of Science, Engineering and Technology*, *60*, 621–526.

Xiao, Y., Kaku, I., Zhao, Q., & Zhang, R. (2011). A reduced variable neighborhood search algorithm for uncapacitated multilevel lot-sizing problems. *European Journal of Operational Research*, *in press*. Doi:10.1016/j.ejor.2011.04.015.

Yero, I., Kuziak, D., & Rodriguez-Velazquez, J. (2011). On the metric dimension of corona product graphs. *Computers and Mathematics with Applications*, *61*, 2793–2798.