European Journal of Operational Research 220 (2012) 328–337

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Variable neighborhood search for metric dimension and minimal doubly resolving set problems

Nenad Mladenović [a,b], Jozef Kratica [b,*], Vera Kovačević-Vujčić [c], Mirjana Čangalović [c]

[a] School of Mathematics, Brunel University–West London, Uxbridge UB8 3PH, UK
[b] Mathematical Institute, Serbian Academy of Sciences and Arts, Kneza Mihaila 36/III, 11 000 Belgrade, Serbia
[c] Faculty of Organizational Sciences, University of Belgrade, Jove Ilića 154, 11 000 Belgrade, Serbia

ARTICLE INFO

Article history:
Received 20 June 2011
Accepted 13 February 2012
Available online 21 February 2012

Keywords:
Metaheuristics
Combinatorial optimization
Variable neighborhood search
Metric dimension
Minimal doubly resolving set

ABSTRACT

In this paper, two similar NP-hard optimization problems on graphs are considered: the metric dimension problem and the problem of determining a doubly resolving set with the minimum cardinality. Both are present in many diverse areas, including network discovery and verification, robot navigation, and chemistry. For each problem, a new mathematical programming formulation is proposed. For solving more realistic large-size instances, a variable neighborhood search based heuristic is designed. An extensive experimental comparison on five different types of instances indicates that the VNS approach consistently outperforms a genetic algorithm, the only existing heuristic in the literature designed for solving those problems.

1. Introduction

1.1. Metric dimension problem

The metric dimension problem (MDP) is introduced independently by Slater (1975) and Harary and Melter (1976). Roughly speaking, the metric dimension of an undirected and connected graph, having all edge weights equal to 1, is the minimum cardinality vertex subset B with the property that all other vertices are uniquely determined by their shortest distances to the vertices in B. In order to give formal definitions, the following notation is needed. Given a simple connected undirected graph $G = (V,E)$, with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set $E(|E| = m)$, $d(u,v)$, denotes the distance between vertices u and v, i.e., $d(u,v)$ gives the number of edges on the shortest $u - v$ path.

Example 1. Consider a graph G_1 given in Fig. 1. It has $n = 6$ vertices and $m = 9$ edges. Vertex v_1 is connected with vertices v_2, v_3 and v_5. Therefore, $d(v_1,v_2) = d(v_1,v_3) = d(v_1,v_5) = 1$. Its distance to the remaining two vertices is equal to 2 ($d(v_1,v_4) = d(v_1,v_6) = 2$), since the shortest path between v_1 and v_4 or v_6 contains two edges. In the same way the distances between any two vertices of the given graph G_1 can be found.

A vertex x of the graph G is said to *resolve two vertices* u and v of G if $d(u,x) \neq d(v,x)$. If looked carefully at graph G_1 from Fig. 1, it can be noticed that vertex v_1 resolves vertices v_2 and v_6, since $d(v_1,v_2) \neq d(v_1,v_4)$. Clearly, v_1 does not resolve pair of vertices v_2 and v_3 or pair v_2 and v_5.

A vertex set $B = \{x_1,x_2,\ldots,x_k\}$ of G is a *resolving set* of G if every two distinct vertices of G are resolved by some vertex of B. Set $B_1 = \{v_1\}$ of the graph G_1 from Fig. 1 is not a resolving set since $d(v_2,v_1) = d(v_3,v_1) = 1$. Using the similar argument, it is easy to show that none of the singleton vertices forms a resolving set.

Given a vertex t, the k-tuple $r(t,B) = (d(t,x_1), d(t,x_2), \ldots, d(t,x_k))$ is called the *vector of metric coordinates* of t with respect to B. A *metric basis* of G is a resolving set with the minimum cardinality. The *metric dimension* of G, denoted by $\beta(G)$, is the cardinality of its metric basis. Set $B_3 = \{v_1,v_2,v_3\}$ from Fig. 1 is a resolving set of G_1 since the vectors of metric coordinates for all the vertices of G_1 with respect to B_3 are mutually different. More precisely, $r(v_1,B_3) = (0,1,1)$; $r(v_2,B_3) = (1,0,2)$; $r(v_3,B_3) = (1,2,0)$; $r(v_4,B_3) = (2,1,1)$; $r(v_5,B_3) = (1,2,1)$; $r(v_6,B_3) = (2,1,2)$. However, B_3 is not a minimal resolving set since $B_2 = \{v_1,v_3\}$ is also a resolving set with smaller cardinality: $r(v_1,B_2) = (0,1)$; $r(v_2,B_2) = (1,2)$; $r(v_3,B_2) = (1,0)$; $r(v_4,B_2) = (2,1)$; $r(v_5,B_2) = (1,1)$; $r(v_6,B_2) = (2,2)$. Thus $\beta(G_1) = 2$. Note that B_2 is not a unique solution, the set $\{v_2,v_4\}$ is also a resolving set with cardinality equal to 2.

The MDP has been widely investigated. Since the complete survey of all the results is out of the scope of this paper, only some relevant recent results will be mentioned. There exist two different integer linear programming (ILP) formulations of the metric

* Corresponding author.
E-mail address: jkratica@mi.sanu.ac.rs (J. Kratica).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2012.02.019

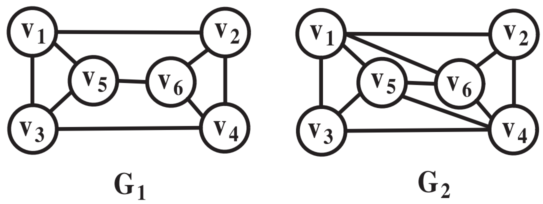

Fig. 1. Graphs in Examples 1 and 2.

dimension problem: Chartrand et al. (2000), Currie and 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 and Oellermann, 2006; Cáceres et al., 2007) and the corona product of graphs (Yero et al., 2011). Another interesting theoretical topic is the connection between the metric dimension and graph invariants such as diameter, number of vertices, and vertex degrees (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 $3 \leqslant \alpha \leqslant \beta + 1$ there exists a graph G with partition dimension α and metric dimension β (Chappell et al., 2008). Metric dimensions of the several interesting classes of graphs have been investigated: Cayley digraphs (Fehr et al., 2006), Grassmann graphs (Bailey and Meagher, 2011), Johnson and Kneser graphs (Bailey and Cameron, 2011), silicate networks (Manuel and Rajasingh, 2011), convex polytopes (Imran et al., 2010) and generalized Petersen graphs (Javaid et al., 2008; Husnine and Kousar, 2010). It has been shown that some infinite graphs have also infinite metric dimension (Cáceres et al., 2009; Rebatel and Thiel, 2011).

The metric dimension arises in many diverse areas, including telecommunications, the robot navigation, connected joints in graphs, and chemistry. In the area of telecommunications, especially interesting is the MDP application to network discovery and verification (Beerliova et al., 2006). Due to its fast, dynamic, and distributed growth process, it is hard to obtain an accurate map of the global network. A common way to obtain such maps is to make certain local measurements at a small subset of the nodes, and then to combine them in order to discover the actual graph. Each of these measurements is potentially quite costly. It is thus a natural objective to minimize the number of measurements, which still discover the whole graph, i.e. to determine the metric dimension of the graph. In (Beerliova et al., 2006) simple greedy strategies were used in a simulation with various types of randomly generated graphs. The results of the simulation were presented as two dimensional diagrams displaying the average number of measurements (cardinality of a resolving set) as a function of the degree of a particular graph class.

An application of the metric dimension problem in chemistry is described in (Chartrand et al., 2000). The structure of a chemical compound can be represented as a labeled graph where the vertex and edge labels specify the atom and bond types, respectively. Under the traditional view, it can be determined whether any two compounds in the collection share the same functional property at a particular position. These positions simply reflect uniquely defined atoms (vertices) of the substructure (common subgraph). It is important to find smallest number of these positions which is functionally equivalent to the metric dimension of the given graph. This observation can be used in drug discovery whenever it is to be determined whether the features of a compound are responsible for its pharmacological activity. For more details see (Chartrand et al., 2000).

Another interesting application of the metric dimension problem arises in robot navigation (Khuller et al., 1996). Suppose that a robot is navigating in a space modeled by a graph and wants to know its current position. It can send a signal to find out how far it is from each among a set of fixed landmarks. The problem of computing the minimum number of landmarks and their positions such that the robot can always uniquely determine its location is equivalent to the metric dimension problem.

1.2. Double resolving set

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 ($n \geqslant 2$) 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, $\beta(G) \leqslant \psi(G)$.

Observe that vertex set B_2 is not a doubly resolving set in graph G_1 from Fig. 1 because $d(v_6, v_1) - d(v_5, v_1) = d(v_6, v_3) - d(v_5, v_3) = 1$. Similarly, it can be shown that none of the subsets of two vertices forms a doubly resolving set. Thus, B_3 is a minimal doubly resolving set and $\psi(G) = 3$.

Example 2. Consider graph G_2 given in Fig. 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, $\beta(G_2) = \psi(G_2) = 3$.

1.3. Complexity

It has been proven that the metric dimension of the Cartesian product $G\Box G$ 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 proven that the upper bound for the metric dimension of $G\Box H$ 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 proven 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 super dense, a greedy constant factor approximation algorithm is presented.

1.4. Previous heuristics

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, i.e. if the current individual does not represent a resolving set, randomly generated vertices are added to the corresponding set until it becomes a resolving set. The repairing technique is the main flaw of the genetic approach. Namely, it has two contradictory requirements. One is to produce feasible solutions as fast as possible, and the other is to keep the diversity of the genetic material. The overall performance of the GA is improved by a caching technique, which speeds up the running time by avoiding computation of the same objective value each time when genetic operators produce individuals with the same genetic code. Testing on various instances from OR-Library (founded by Beasley (1990)) and theoretically challenging classes of graphs shows that the 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 the MDRSP on hypercubes are used in a dynamic programming approach to obtain upper bounds for the metric dimension of large hypercubes.

1.5. New results

In this paper, mathematical programming models for the MDP and the MDRSP are proposed, with different objective functions than in previous papers. Instead of minimizing the cardinality of a resolving set, the number of pairs of vertices from G, that are not resolved (doubly resolved) by vertices of a set with a given cardinality s, is minimized. If the value of the new objective function is zero, the same model with smaller cardinality s is applied. Otherwise, s is increased. In that way, the difficulty that arises when solving the plateaux problem, i.e. problems with a large number of solutions with the same objective function values (see Davidovic et al. (2005)), vanishes with the 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., in press). Moreover, if the distance k between two formulations $\mathcal{F}(s_1)$ and $\mathcal{F}(s_2)$ is defined as $k = |s_1 - s_2|$, then this approach may be seen as a formulation space search (Mladenović et al., 2005, 2007; Kochetov et al., 2008). Actually, the best formulation of the MDP and the MDRSP is searched for in the formulation space.

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

The paper is organized as follows. In Section 2, some interesting properties are presented, which are used in the following sections. The existing and 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. Section 6 contains conclusions and directions to future work.

2. Examples and preliminaries

In this section, some theoretical properties of both the MDP and the MDRSP are presented and illustrated on some simple examples.

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

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

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

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 following paragraphs, a short description of hypercubes and Hamming graphs is given.

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. Clearly 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:

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

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 $\{0, 1, \ldots, k - 1\}$. 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. Furthermore, 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 proven in (Cáceres et al., 2007) that $\beta(H_{2,k}) = \lfloor \frac{4k-2}{3} \rfloor$. The metric dimension is known for hypercubes Q_r, $r \leqslant 8$. Upper bounds of $\beta(Q_r)$ for $9 \leqslant r \leqslant 14$ are obtained in (Cáceres et al., 2007; Kratica et al., 2009a). It has been proven in (Cáceres et al., 2007) that $\beta(Q_r) \leqslant r - 5$ for $r \geqslant 15$. This theoretical upper bound has been improved in (Kratica et al., 2009a) for $r \geqslant 17$. In (Kratica et al., 2009b) new bounds of $\beta(Q_r)$ for $r \leqslant 90$ 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 and Oellerman, 2001), and one ILP formulation of the minimal doubly resolving set problem (Kratica et al., 2009b). In this section, the existing models are presented, as well as their new mathematical programming formulations, for both problems. The new formulations used in the VNS approach for calculating the objective functions of the MDP and MDRSP are described in Section 4 in detail.

3.1. Metric dimension

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

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

subject to

$$\sum_{j=1}^{n} |d(u,j) - d(v,j)| \cdot y_j \geqslant 1, \quad 1 \leqslant u < v \leqslant n \tag{4}$$

$$y_j \in \{0,1\}, \quad 1 \leqslant j \leqslant n \tag{5}$$

The objective function (3) represents the cardinality of a feasible solution B. Constraints (4) make sure that for each two vertices u and v there exists at least one vertex from B which resolves u and v, i.e. B is a resolving set. Constraints (5) reflect the binary nature of decision variables y_j.

Note that distances $d(u,j)$ and $d(v,j)$ should be calculated in advance. So, $|d(u,j) - d(v,j)|$ are given constants and therefore constraints (4) are linear. Instead of $|d(u,j) - d(v,j)|$, the following coefficients may be introduced:

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

Then inequality (4) becomes

$$\sum_{j=1}^{n} A_{(u,v),j} \cdot y_j \geqslant 1 \quad 1 \leqslant u < v \leqslant n \tag{7}$$

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

Clearly the number of resolving sets with the same cardinality might be huge. Therefore, any local search heuristic has difficulties to continue searching after being in such a solution, because neighboring solutions usually have the same cardinality, so that there are small chances for improvement. In order to avoid this problem, in this paper an auxiliary objective function is proposed, as well as the decomposition of the MDP into a sequence of subproblems with relaxed resolving requirements and fixed cardinalities of feasible sets. In each subproblem, it is checked if a resolving set B with a given cardinality s exists. If such a resolving set exists, then $\beta(G) \leqslant s$, 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 any vertex from B'. If $ObjF(B') = 0$ then B' is a resolving set.

Let $A_{(u,v),j}$ be defined by formula (6), $y_j = \begin{cases} 1, & j \in B' \\ 0, & j \in V \setminus B' \end{cases}$ 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 modeled 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

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

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

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

Constraints (10) provide that the cardinality of a feasible solution B' is equal to s. Constraints (11) make sure that if two vertices u and v are not resolved by any vertex from B' then $z_{u,v}$ should be equal to 1. The minimality of objective function (9) ensure that $z_{u,v} = 0$ for all pairs of vertices u and v resolved by B'.

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

Proof. ($\Rightarrow$) Suppose that B' is a resolving set with cardinality s. Then, by formula (8), for each u, $v \in V$, $z_{uv} = 0$ and hence $\sum_{u=1}^{n-1} \sum_{v=u+1}^{n} z_{uv} = 0$. Constraint (10) is satisfied by the assumption because $|B'| = s$. Since B' is a resolving set, for each $1 \leqslant u < v \leqslant n$ there exists $j \in B'$ such that $d(u,j) \neq d(v,j)$, 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 $\sum_{j=1}^{n} A_{(u,v),j} \cdot y_j \geqslant 1 = 1 - z_{uv}$.

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

Using subproblems (9)–(12) the MDP can be solved in the following way. Let s be equal to an upper bound for the metric dimension (in the worst case $\beta(G) \leqslant n - 1$) minus one, and let us 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 $\beta(G) \geqslant 1$) 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 the 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 $1 \leqslant u < v \leqslant n$, $1 \leqslant i < j \leqslant n$. Variable y_i described by the formula (14) determines whether vertex i belongs to a doubly resolving set D or not. Similarly, x_{ij} determines whether both vertices i, j are in D.

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

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

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

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

subject to:

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

$$x_{ij} \leqslant \frac{1}{2}y_i + \frac{1}{2}y_j, \quad 1 \leqslant i < j \leqslant n \tag{18}$$

$$x_{ij} \geqslant y_i + y_j - 1, \quad 1 \leqslant i < j \leqslant n \tag{19}$$

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

The objective function (16) represents the cardinality of a feasible solution D, while constraints (17) imply that for each two vertices u and v there exists at least one pair of vertices from D which doubly resolve u, v. Constraints (18) and (19) ensure that $x_{ij} = 1$ if and only if $y_i = y_j = 1$. 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.

The 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 formula (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:

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

subject to:

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

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

$$x_{ij} \leqslant \frac{1}{2}y_i + \frac{1}{2}y_j, \quad 1 \leqslant i < j \leqslant n \tag{25}$$

$$x_{ij} \geqslant y_i + y_j - 1, \quad 1 \leqslant i < j \leqslant n \tag{26}$$

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

$$y_k \in \{0,1\}, \quad 1 \leqslant k \leqslant n \tag{28}$$

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

Proposition 3. *A subset D' with cardinality s is a doubly resolving set of G if and only if the optimal objective function value of the problem (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 the MDP and the MDRSP

Variable neighborhood search (VNS) is an effective metaheuristic introduced in (Mladenović and Hansen, 1997). The basic idea of VNS is to use more than one neighborhood structure and to proceed with their systematic change 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 the structure in which the neighborhoods have increasing cardinality: $|N^{k_{min}}(X)| < |N^{k_{min}+1}(X)| < \ldots < |N^{k_{max}}(X)|$.

Given an incumbent X and an integer $k \in \{k_{\min}, \ldots, k_{\max}\}$ associated to a current neighborhood, the shaking procedure generates a feasible solution in $N^k(X)$. Then a local search is applied around the 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. In the standard VNS the next search begins with the first neighborhood $N^{k_{min}}$. Otherwise, the next neighborhood in the sequence is considered to improve the current solution. Should the last neighborhood $N^{k_{max}}$ be reached without a solution better than the found incumbent, 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. In the next paragraph, some recent successful VNS applications are listed:

- Mladenović et al. (2010) considered the problem of reducing the bandwidth of a matrix. VNS based heuristic for reducing the bandwidth gives better results compared with all previous methods.
- Ilić et al. (2010) proposed a new general VNS approach for the uncapacitated single allocation p-hub median problem in networks. It outperforms the best-known heuristics in terms of solution quality and computational effort.
- Xiao et al. (2012) presented a reduced VNS algorithm and several implementation techniques for solving uncapacitated multilevel lot-sizing problems. The algorithm is competitive against other methods, enjoying good effectiveness as well as high computational efficiency.
- Muller et al. (2012) presented a hybrid of a general heuristic framework and a general purpose mixed-integer programming solver. The general reoptimization approach used for repairing solutions is specifically suited for combinatorial problems where it may be hard to otherwise design suitable repair neighborhoods. The hybrid heuristic framework is applied to the multi-item capacitated lot sizing problem with setup times with excellent results.
- Mladenović et al. (in press) proposed a VNS approach for solving the one-commodity pickup-and-delivery travelling salesman problem. They adapt a collection of neighborhood structures, kopt, double-bridge and insertion operators mainly used for solving the classical travelling salesman problem. A binary indexed tree data structure is used, which enables efficient feasibility checking and updating of solutions in these neighborhoods. Extensive computational analysis shows that the proposed method outperforms the best-known algorithms in terms of both the solution quality and computational efforts.

- Mansini et al. (in press) proposed a VNS procedure based on the idea of exploring, most of the time, granular instead of complete neighborhoods in order to improve the algorithms efficiency without loosing effectiveness. The method provides a general way to deal with granularity for those routing problems based on profits and complicated by time constraints. Performance of the proposed algorithm is compared to optimal solution values, when available, or to best known solution values obtained by state-of-the-art algorithms.

Algorithm 1. VNS pseudo code.

Function `VNS` (k_{min}, k_{max}, $iter_{max}$, p_{move})
$B \leftarrow$ `RSInit ()`
$B' \leftarrow$ `DeleteLast`(B)
$k \leftarrow k_{min}$
$iter \leftarrow 0$
repeat
  $iter \leftarrow iter + 1$
  $B'' \leftarrow$ `Shaking`(B', k)
  `LocalSearch`(B,B'')
  if `Compare`(B'', B',p_{move}) **then** $B' \leftarrow B''$
  else
    if $k < k_{max}$ **then** $k \leftarrow k + 1$
    else $k \leftarrow k_{min}$
  end if
until $iter \leqslant iter_{max}$
return B

4.1. Metric dimension

The VNS approach for the 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. Value of s is set to be equal to $|B| - 1$ and subproblems (9)–(12) are iteratively solved, decreasing s by one as long as the optimal objective function value is zero.

More precisely, for a given resolving set B the last element is deleted, using the procedure *DeleteLast*(B), to 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()` sets B' and B'' are compared, the set B'' is updated if necessary, and the VNS procedure is continued until the stopping criterion is met. The pseudo-code of the VNS implementation for solving the MDP is given in Algorithm 1.

4.2. 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 $N^k(B')$.

4.3. Local search

In the local search procedure *LocalSearch*( ), starting with B'', one element from set B'' is interchanged with one element of its complement. The best improvement strategy is used, i.e. in every step an interchange is performed, 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 is no improvement.

Algorithm 2. Pseudo code of the local search.

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

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 the computation of the objective function, instead of comparing metric coordinates for each two pairs of vertices from V, vectors of metric coordinates are sorted in the lexicographical order. Then, the objective function value is calculated 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, there is one call of the built-in sorting function *qsort* for each interchange, which gives a total of $|B''| \cdot (|V| - |B''|)$ calls. A significant speedup in the local search implementation is obtained 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 to be replaced 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, set $z[v] = 0$, $v \in V \setminus B''$.

Let us 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 $v \in V \setminus B'' z[v]$ is increased by one, whenever $d(p,v) = d(q,v)$.

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

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

4.4. Neighborhood change

After the local search procedure, there are three possibilities. Within the function *Compare*() the decision is made 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 $ObjF(B'') < ObjF(B')$ set $B' := B''$ and continue the search with the same neighborhood N^k;
- If $|B''| = |B'|$ and $ObjF(B'') > ObjF(B')$, then repeat the search with the same B' and the next neighborhood;
- If $|B''| = |B'|$ and $ObjF(B'') = ObjF(B')$, then with probability p_{move} set $B' := B''$ and continue search with the same neighborhood N^k, and with probability $1 - p_{move}$ repeat the search with the same B' and the next neighborhood.

4.5. Discussion

The described VNS approach tries to minimize the number of pairs of vertices with the same metric coordinates with respect to the set B' ($|B'| = |B| - 1$). In the local search procedure one element of set B'' is interchanged with one element of its complement. The interchange procedure assumes that $|B'| - 1 = |B''| - 1 \geqslant 1$. Therefore, the VNS approach for the MDP can be applied only for the graphs with the 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 and 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 of 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 $k = 1$, $B'' = \{v_1, v_3, v_4\}$. The corresponding objective function value is $ObjF(B'') = 1$ since all pairs of vectors of metric coordinates are different except one:$r(v_2, B'') = r(v_6, B'') = (1,2,1)$. Let us try to exchange e.g.$v_r = v_1$ with one element $v \in V \backslash B''$ using the best improvement strategy. After sorting the vectors of metric coordinates with respect to $B'' \backslash v_r = \{v_3, v_4\}$ in the lexicographical order and identifying the blocks of vertices with the same metric coordinates, array z is obtained, which stores objective function values $ObjF(B'' \cup \{v\} \backslash \{v_r\}) = ObjF(\{v_3, v_4\} \cup \{v\})$ for each vertex $v \in V \backslash B'' = \{v_2, v_5, v_6\}$. 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 the 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, set $B'' = \{v_2, v_3\}$ is obtained. The corresponding objective function value is $ObjF(B'') = 1$ since there is one pair of vectors with the same metric coordinates:$r(v_1, B'') = r(v_4, B'') = (1,1)$. Let us exchange, e.g., $v_r = v_2$ with one element $v \in V \backslash B''$ using the best improvement strategy. After sorting the vectors of metric coordinates with respect to $B'' \backslash v_r = \{v_3\}$ in the lexicographical order, and identifying the blocks of vertices with the same metric coordinates, an array z is obtained, which stores objective function values $ObjF(B'' \cup \{v\} \backslash \{v_r\}) = ObjF(\{v_3\} \cup \{v\})$ for each vertex $v \in V \backslash B'' = \{v_1, v_4, v_5, v_6\}$. The results of this process are displayed in Table 2. Since the block $\{v_3\}$ consists of only one vertex, it has no influence on array z and the entries in the corresponding row are omitted. As in the array z all entries, except the 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 the local search procedure stops since there is 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.

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

4.6. VNS for minimal doubly resolving set problem

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, the checking whether the current solution is a resolving set or not is replaced by a doubly resolving set checking. In order to increase the efficiency of this identification, the results of the following proposition are used.

Proposition 4. Kratica et al., 2009b*A subset $D = \{x_1, x_2, \ldots, x_k\} \subseteq V$ is a doubly resolving set of G if and only if for every $p,q \in V$ there exists $i \in \{1,2,\ldots,k\}$ 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 $r'(v,D) = (d(v,x_2) - d(v,x_1), \ldots, d(v,x_k) - d(v,x_1))$. 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 the MDP has been effectively adapted to solve the MDRSP. It should be noted that this VNS approach can be applied to graphs with the cardinality of the minimal doubly resolving set at least four. Similarly, as for the MDP, this is not a serious drawback.

Another interesting related problem is the strong metric dimension problem (SMDP), introduced in (Sebo and 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 the vertices as in the case of a resolving set or a doubly resolving set. Therefore, a straightforward adaptation of the described VNS approach for the SMDP is impossible.

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

5. Experimental results

All tests were performed on a single processor from an Intel Quad 2.5 GHz computer with 1 GB memory, under Windows XP operating system. The VNS implementations were coded in C programming language. In our experiments, the following values for the VNS parameters were used: $k_{min} = 2$, $k_{max} = 20$, $p_{move} = 0.2$, $iter_{max} = 100$. This choice is motivated by the following reasons. Since the local search explores all solutions which can be obtained from the current solution by interchanging one of its element, then the function *Shaking()* is rational only for $k \geqslant 2$. Although a theoretical upper bound for the cardinality of B' is $n-1$, in practice it is usually much smaller. Namely, the initial set B obtained by *RSInit()* in most cases has relatively small cardinality. Since for the current neighborhood $N^k(B')$ it holds that $k < |B'| \leqslant |B|$, large values for k_{max} would not have any practical effect. Therefore, $k_{max} = 20$ is a good compromise. As the MDP and the MDRSP objective functions have integer values, there are many different solutions with the same objective function value. Consequently, values of the parameter p_{move} close to 1 could cause the cycling, while the values close to 0 could reduce the diversification in the VNS search procedure. The choice $p_{move} = 0.2$ is motivated by this observation. Relatively small value for the maximal number of the VNS iterations (stopping criterion $iter_{max} = 100$) is motivated by the fact that the local search in the VNS procedure can be time consuming for large-scale instances.

5.1. Test instances

This section presents the results of the VNS approach to the MDP and the MDRSP, and tested on various classes of graph instances: pseudo boolean, crew scheduling, graph coloring, hypercubes and Hamming graphs. These instances were already been tested by the GA approach in (Kratica et al., 2009a,b). Characteristics of all these classes are presented in Table 3. For each instance class its full name, its abbreviation, the number of instances it contains, then the minimal, the maximal and the average number of vertices (in columns 4-6) and edges (in last three columns) are given, respectively.

The VNS was run 20 times for each instance and the results for the MDP were summarized in Table 4, together with the corresponding GA results. Both VNS and GA use the well-known Floyd-Warshall shortest path algorithm for computing the distances between all pairs of vertices. For the sake of conciseness, only the average values for each class are presented. More precisely, the best/average results in 20 runs for each particular instance are computed and then their average values for the whole class are presented. Table 4 is organized as follows:

- The first column contains the abbreviation of the instance class.
- The next three columns contain the average GA results for each instance class: the cardinality of the best obtained resolving set (denoted by *best*), the average resolving set cardinality and the average GA running time (denoted by *avg* and *t*, respectively).
- The next three columns contain the average VNS results, presented in the same way as for the GA.
- The last column presents the number of instances where the best GA solution is improved by the VNS.

The GA and the VNS results for the MDRSP are given in the same way in Table 5.

For detailed results of the VNS and the GA for each particular instance see the supplementary material attached to this paper (or see http://www.mi.sanu.ac.rs/~jkratica/vns_results/mdp_mdrsp/supplementary_data.pdf).

From Tables 4 and 5, it is clear that the results obtained by the VNS based heuristic outperform those obtained by the GA approach (Kratica et al., 2009a,b). More precisely, the following observations hold:

(i) The best results obtained by the VNS are never worse than those reported by the GA. Moreover, the average VNS results for all instance classes, except the hypercubes, are better than the best GA results.

(ii) It can be seen from the supplementary material that the new upper bounds for most of the test instances. are obtained by VNS. For example, in the case of *csp* instances the VNS for the MDP has improved the GA results in 8 out of 10 cases, while the VNS for the MDRSP is better than the GA in 9 out of 10 cases. For all instances of *gcol* and *frb* classes, the VNS gives strictly better results than the GA, both for the MDP and the MDRSP. For Hamming graphs $H_{2,k}$, the VNS has always reached the exact value of the metric dimension ($\beta(H_{2,k}) = \lfloor \frac{4k-2}{3} \rfloor$). For those Hamming graphs, for which the metric dimension is not known, the VNS for the MDP has improved the GA upper bounds in 15 out of 54 cases, while for the MDRSP the VNS has improved the GA results in 17 out of 54 cases.

(iii) The VNS running times are significantly smaller than the GA running times for all instance classes. For *csp* and *gcol* instances the VNS running times are more than 15 times smaller than the GA running times, both for the MDP and the MDRSP. For larger *frb* instances the VNS is about 9 times faster than the GA for the MDP, and about 6 times faster for the MDRSP. In the case of hypercubes and Hamming graphs the VNS is still at least 2 times faster than the GA, both for the MDP and the MDRSP.

Good quality of the VNS results can be justified as follows. The VNS neighborhood structures appropriately chosen for the new objective function contributes to efficient diversification. Another significant characteristic of the VNS is its local search procedure, which effectively contributes to intensification of the search process. The local search is very fast for graphs up to 1000 vertices. For larger graphs up to 5000 vertices, it is somewhat slower since it uses sorting, but it still contributes to the overall efficiency. It should be noted that for these graphs the shortest path algorithm, is very time consuming. Fortunatelly, it is executed only once in the initialization phase of both GA and VNS.

5.2. Large hypercubes

For large-scale hypercubes Q_r, $r \geqslant 13$, the distance matrix cannot fit in the memory. In order to overcome this obstacle, a special VNS for the MDP and the MDRSP on hypercubes is developed. Instead of generating and memorizing the whole distance matrix, this VNS computes the distance between two vertices each time

Table 3
Characteristics of instance classes.

Inst. class	*Abbr.*	*Num.*	n_{min}	n_{max}	n_{avg}	m_{min}	m_{max}	m_{avg}
Crew scheduling	csp	10	50	500	275	173	16,695	6389
Graph coloring	gcol	30	100	300	166	2420	22,601	9152
Pseudo boolean	frb	40	450	1534	1013	17,794	127,011	69,554
Hypercubes	hyp	10	8	4096	818	12	24,576	4505
Hamming graphs	Ham	54	16	4913	981	48	117,912	16,024

Table 4
Comparison of GA and VNS for the MDP.

Class	GA			VNS			Impr
	best	avg	t(sec)	best	avg	t(sec)	
csp	19.600	22.365	50.78	16.800	17.590	2.29	8/10
gcol	10.000	10.527	8.53	9.000	9.580	0.49	30/30
frb	36.975	40.758	431.26	30.550	31.215	48.71	40/40
hyp	5.800	5.800	258.69	5.800	5.800	55.77	0/10
Ham	17.204	17.696	408.68	16.778	16.820	67.91	15/54

Table 5
Comparison of GA and VNS for the MDRSP.

Class	GA			VNS			Impr
	best	avg	t(sec)	best	avg	t(sec)	
csp	20.700	22.890	53.39	17.600	18.200	2.66	9/10
gcol	10.100	10.537	8.24	9.000	9.708	0.51	30/30
frb	39.975	42.691	358.75	30.600	31.281	59.37	40/40
hyp	6.400	6.420	249.06	6.300	6.300	111.63	1/10
Ham	17.259	17.711	424.02	16.815	16.853	75.90	17/54

it is needed. The code has been optimized using the special structure of hypercubes. The VNS for large hypercubes is run only once, since it is very time consuming due to large dimensions. The results are compared with the special GA for large hypercubes (Kratica et al., 2009a,b). The comparison is presented in Table 6, which is 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 the VNS performance for the MDP and the GA results from (Kratica et al., 2009a). The fifth and the sixth column contain the cardinality (denoted by *best* and t), of the resolving set obtained by the GA, and the corresponding running time. The seventh and the eight column contain the experimental VNS results, presented in the same way as for the GA;
- results for the MDRSP are presented in the last four columns and organized in the same way as the data for the MDP.

For hypercubes Q_r, $3 \leqslant r \leqslant 10$, both GA and VNS reach 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 the MDRSP, the VNS has improved the GA results by one in 4 out of 15 cases. Note that in the case of large-scale hypercubes ($r \geqslant 13$) the VNS running time is longer than the GA running time, both for MDP and MDRSP. The reason is the fact that for graphs with a huge number of vertices ($n \geqslant 2^{13} = 8192$) the local search procedure is time consuming. Since the GA does not use a local search, the running time for large hypercubes is smaller. However, in all cases, the VNS produces better results than the GA, either for the MDP or for the MDRSP, or for both. In three cases, denoted in Table 6 by the asterisk, the VNS was stopped after one day running time.

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 $\beta(Q_r) \leqslant r - 5$ for $r \geqslant 15$ from (Cáceres et al., 2007). Namely, the following proposition holds:

Proposition 5. *$\beta(Q_r) \leqslant r - 6$ for $r \geqslant 15$*

Proof. According to Proposition 1

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

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

6. Conclusions

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

The new objective function calculates the number of pairs of vertices not resolved (doubly resolved) by the vertices of a set with the given cardinality. As it can be seen from the experimental results, the difficulties appearing when solving the MDP and MDRSP, which have a large number of resolving (doubly resolving) sets with the same cardinality, are successfully reduced by using the new objective function. The corresponding neighborhood structures allow an effective shaking procedure which successfully

Table 6
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*
avg	26188.8	1209638	8.6	5988.0	8.2	19168	9.1	4362.9	8.7	21590.2

diversifies the search process. The local search implementation was conducted very efficiently and it resulted in an excellent overall VNS performance for graphs with up to 5000 vertices.

An extensive experimental comparison with the only existing heuristic approach based on the genetic algorithm indicates the superiority of the VNS approach with respect to the solution quality. The VNS running times are smaller than the GA running times, except for large hypercubes. The VNS has reached exact values of all theoretically known metric dimensions for Hamming graphs. Moreover, for each of the theoretically challenging large hypercubes, VNS has made improvements of the previously known upper bounds, either for the MDP, or for the MDRSP, or for the both.

This research can be extended in several ways. It would be challenging to investigate applications of the presented VNS approach to similar problems on graphs. In addition, 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.

Acknowledgement

This research was partially supported by Serbian Ministry of Education and Science under Grants 174010 and 174033.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2012.02.019.

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.

Beasley, J., 1990. Or-library: distributing test problems by electronic mail. Journal of the Operational Research Society 41, 1069–1072.

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., Seara, C., Wood, D., 2007. On the metric dimension of cartesian products of graphs. SIAM Journal in Discrete Mathematics 21, 423–441.

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.

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.

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

Davidovic, T., Hansen, P., Mladenovic, N., 2005. Permutation based genetic, tabu and variable neighborhood search heuristics for multiprocessor scheduling with communication delays. Asia–Pacific Journal of Operational Research 22, 297–326.

Fehr, M., Gosselin, S., Oellermann, O., 2006. The metric dimension of Cayley digraphs. Discrete Mathematics 306, 31–41.

Garcia, S., Labbe, M., & Marin, A., in press. Solving large p-median problems with a radius formulation. Informs Journal on Computing. 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., Pelayo, I.M., 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.

Mansini, R., Labadie, N., Melechovský, J., & Calvo, R., in press. The team orienteering problem with time windows: an LP-based granular variable neighborhood search. European Journal of Operational Research. doi:10.1016/j.ejor.2012.01.030.

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., Urošević, D., 2005. Reformulation descent applied to circle packing problems. Computers & Operations Research 32, 2419–2434.

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

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

Mladenović, N., Urošević, D., Hanafi, S., & Ilić, A., in press. A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem. European Journal of Operational Research. doi:10.1016/j.ejor.2012.01.036.

Muller, L., Spoorendonk, S., Pisinger, D., 2012. A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. European Journal of Operational Research 218, 614–623.

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. Congressus 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, 621–626.

Xiao, Y., Kaku, I., Zhao, Q., Zhang, R., 2012. A reduced variable neighborhood search algorithm for uncapacitated multilevel lot-sizing problems. European Journal of Operational Research 214, 223–231.

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