Les Cahiers du GERAD

ISSN: 0711–2440

Reduced variable neighborhood programming for the preventive maintenance planning of railway infrastructure

S. Elleuch, B. Jarboui,
N. Mladenović

G–2016–92

November 2016

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Reduced variable neighborhood programming for the preventive maintenance planning of railway infrastructure

Souhir Elleuch [a]

Bassem Jarboui, [b]

Nenad Mladenović [c,d]

[a] *MODILS, University of Sfax, Tunisie*

[b] *Emirates College of Technology, Abu Dhabi, United Arab Emirates*

[c] *Mathematical Institute, SANU, Belgrade, Serbia*

[d] *GERAD, Montréal, Canada*

elleuchsouhir1@gmail.com
bassem_jarboui@yahoo.fr
nenadmladenovic12@gmail.com

November 2016

Les Cahiers du GERAD
G–2016–92

Abstract: In this paper we present our solution for the Challenge problem publicly announced by Railway Application Section (RAS), which operates within INFORMS. Variables in the problem represent decisions should the each particular railway truck segment be preventively maintained or replaced. The huge training data set is provided containing several attributes of each track segment, including the right decision regarding maintenance. Correct decisions are removed from the testing set. Research teams were competed who would have less wrong decisions on testing data. In order to solve this problem, we divide it into two phases: prediction and classification. Both phases are solved by using rules of recently proposed automatic programming method called Variable neighborhood programming. It appeared that our team was among awarded.

Keywords: Preventive maintenance, railway infrastructure maintenance, Variable Neighborhood Search, genetic programming, Variable Neighborhood Programming

1 Introduction

The railway transportation becomes more important due to the environmental (CO_2 emission) and low energy consumption requirements. To ensure an efficient railway system planning, one must include maintenance of infrastructure. Since the number of travel lines is increasing, the maintenance planning becomes more and more important and complex activity.

Problem description Railroad companies make an inspection run for each period and record the characteristic of found defects in rail tracks. These defects have to meet Federal Railroad Administration (FRA) standards. They can be classified into two classes. The first one includes defects which do not meet FRA standards and must be corrected immediately. These defects are also called red defects. The second class gathers yellow defects, which meet FRA standards, but violate railroad's own standards. The challenge is to predict when yellow defects will turn into red defects. This allows railroad companies to more efficiently maintain the railway tracks. The real-world challenge problem of that type is publicly announced by Railway Application Section (RAS), which operates within INFORMS (https://www.informs.org/Community/RAS/Problem-Solving-Competition). One needs to decide should some segment of the railway truck be preventively maintained or replaced? The huge training data set provided by RAS contains several attributes of each track segment, including the column with right decision regarding maintenance. Correct decisions are removed from the testing set. Research teams were competed who would have more correct decisions on testing data.

Solution technique This problem can be solved by classification and also by forecasting techniques. There are many methods applied on prediction as fuzzy systems [1, 2] , evolutionary algorithm [3], Neural Networks [4], Genetic Programming (GP) [5]. The solution approach we develop in this paper is based on Automatic programming (AP), an effective technique in artificial intelligence (AI) field. The output (solution) of AP problem is computer program. It is generated in an automatic way. One among such AP techniques is well-known Genetic programming , introduced by Koza in 1992 [6]. GP is based on operators usually used in genetic algorithm (GA). All movements in GP follow natural evolution. GP starts from an initial generation (population) of programs and then applies genetic operators in each generation (iteration) of the method, in order to reach the near optimal solution, or program. GP is used in solving successfully many problems [7]. As mentioned earlier, programs are frequently presented as trees within GP algorithm. Beside GP, let us mentioned an another evolutionary based algorithm called Immune Programming, which presents another solution to automatic programming problems. This algorithm is based on the vertebrate immune system [8].

It has been noted that all evolutionary algorithms based on population explore the search space horizontally, risking to pass by the optimal solution without managing to reach it. The consequence is the slow convergence to a near-optimum solution [9]. To overcome this, many GP based variants include local search algorithms as well [10, 11]. Local search (LS), or steepest descent procedure applies a sequence of local changes of an initial solution, which improves each time its quality, until there are no more changes of that type that produce a better solution, i.e., until attaining a local optimum. This fact advocates that some other metaheuristics of local search type, such as Variable neighborhood search (VNS), can be adapted in solving AP problem [12].

Variable Neighborhood Search (VNS) metaheuristic [13] is based on local search but does not follow a single trajectory. The neighborhood structures are defined to explore the vicinity of the current incumbent (best known so far) solution in the search space. In fact, VNS explores search space in depth, and conduct the process to reach a good solution in a reasonable time [14]. It has many degrees of freedom, and being very general, it can be applied to solve different problems [15]. In addition, many related works in solving global optimization problems have proved that heuristics based on VNS, compared with population based algorithms [16], provide solutions with better quality (see e.g. [17, 14, 18]).

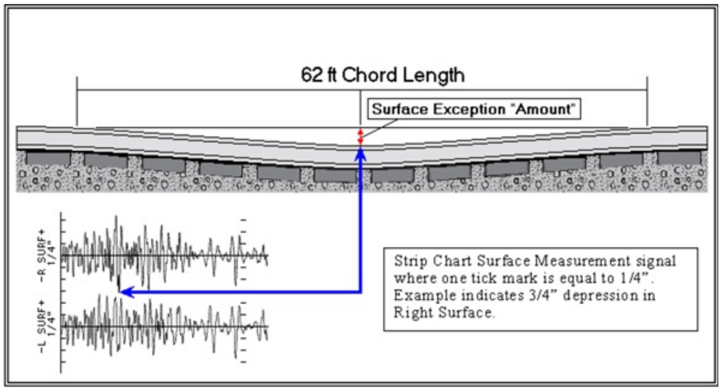
Contribution This paper proposes a decision support system approach for solving preventive maintenance problem of railway infrastructure. For that purposes, the problem is first decomposed into two stages: prediction and classification. Both phases consist of learning phase and testing phase. Moreover, the solution techniques for both stages are based on recent Variable neighborhood programming (VNP) metaheuristic [12]. It follows steps of VNS, taking into account that the solution of the problem is a tree that contains different types of nodes. VNP based heuristics are successfully tested on the three known problems usually used in GP as well: regression, time series forecasting and classification problems [12]. For solving preventive maintenance problem as defined by RAS, we here develop reduced version of VNP (RVNP), i.e., there is no local search routine and different neighborhoods are used just in shaking step of VNP. Therefore, our contribution consists in decomposing the original maintenance railway problem into two more elementary problems and also in solving both subproblems as Automatic programming problems, using the recent VNP approach. The computational results approve the quality of our approach.

Outline The rest of this work is organized in the following way. First, we give a detailed explanation of the maintenance planning problem as announced and presented by Railway Association Society (RAS). Second, we present the new Reduced VNP algorithm in general and then we explain its application in the present problem. Finally, we end up with results and conclusions.

2 Detailed description of the problem

Trains are used to transport goods and passengers. Railway transportation is highly regulated by the state to protect travelers. The maintenance of the railway infrastructure is important part of railway transportation, for keeping freight and passenger trains moving safely, but it is very expensive and very difficult to plan [19]. A survey in this area could be find in [20]. RAS was founded in 1994, having in mind that the productivity improvements can be achieved by merging professionals and scientist. Each year they organize a competition for solving the most challenging problems in the railway. RAS is a subdivision of Institute for Operations Research and Management Sciences (INFORMS). The challenge in this competition was to develop an accurate system able to provide a preventive maintenance planning. Railway companies can efficiently maintain the rail and remain in Federal Railroad Administration (FRA) compliance.

Railway tracks are divided in segments. Railway companies make an inspection run for each time period and record the characteristic of found defects at the corresponding track segment. If a defect does not meet FRA standards, then it is classified as a red one and must be repaired immediately. Otherwise, the defect belongs to yellow class and its fixation is not urgent. The RAS provided Challenge participants the history of the data describing the status of a several track segments. The problem is to predict the defect color of a selected track segment in a predefined milepost value after a given period. Railroad companies distinguish three types of defect: surface, cross-level, and alignment. These defects are detailed on the site of 2015 RAS competition. For the completeness of the paper, we briefly repeat those technical conditions.The three first figures presenting defects are taken from RAS competition overview (https://www.informs.org/Community/RAS/Problem-Solving-Competition). The surface defect is detected by measuring the uniformity of the rail. These measures are recorded over a 62-foot chord for the right and the left rails. Figure 1 shows an example of surface defect measurement.



Strip Chart Surface Measurement signal where one tick mark is equal to 1/4". Example indicates 3/4" depression in Right Surface.

Figure 1: Surface defect

Cross level defect D is measured by calculating the difference in height between the top of one rail to the top of the opposite rail. This difference is indicated as D in the Figure 2. Zero crosslevel is reached when $D = 0$.

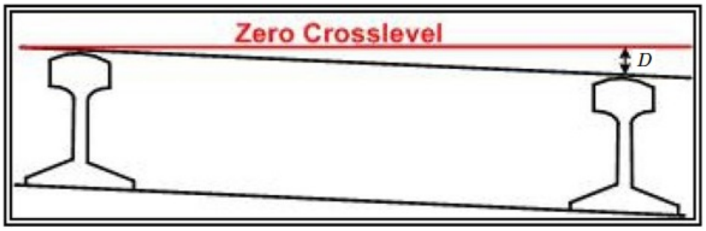


Figure 2: Cross-level defect

The third defect Alignment is also called Dip. It is calculated over a 31 foot. As illustrated at Figure 3, Dip represents a fall in the track and the corresponding rise. The measured value is positive if the defect is a rise and it's negative if the defect is a fall. We note that this defect is calculated in the center within a short distance.

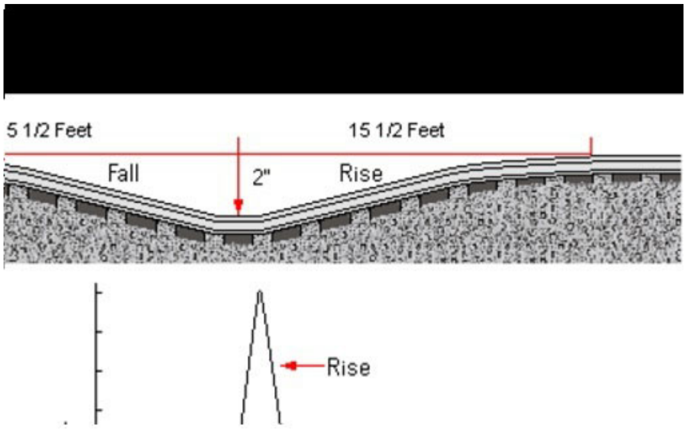


Figure 3: Dip defect

RAS provides a training dataset and a testing dataset. In the training dataset, there are about twenty attributes (columns in the input matrix). Data set has around 36,000 defect samples (rows in input matrix). We present attributes in three tables. In Table 1, the most important attributes with short descriptions derived from 2015 RAS training dataset are given (for more details see the training dataset available on the 2015 RAS competition website). As mentioned, the measure of a defect depends on its type, which makes the problem more complicated and needs many analysis.

3 VNP for solving automatic programming problem

Automatic programming (AP) is a challenging problem in artificial intelligence field. It consists in searching automatically the best program for solving some problems, where program is presented as a tree with different types of nodes. As we have mentioned in the introduction, GP is well-known metaheuristic designed for solving AP. It is based on GA operators. In GP, the solution (program) is presented as a tree for the first time [6]. The solution of automatic programming problem can be presented as lines of some code, an arithmetic function, etc. Let

$$F = \{f_1, f_2, ..., f_n\}$$

denotes the functional set and $E = X \cup C$ denotes the terminal set where $X = \{x_1, x_2, ..., x_{k'}\}$ is the set of the problem variables, and $C = \{c_1, c_2, ..., c_{k''}\}$ is the constant set; $k = k' + k''$ is the cardinality of E. We will now present basic facts of VNP metaheuristic using simple example.

Table 1: Training dataset columns

Column	Type	Characteristics
LINE_SEG-NBR	INTEGER	Every track on the railroad has a unique identifying line segment number. It Could be single or double tracks. Using line segment(LINE_SEG-NBR) and milepost(MILPOST_START and MILEPOST_END) you can identify any location on the system.
MILEPOST	DECIMAL	A Milepost is a point on the track.
TRACK_SDTK_NBR	CHARACTER	Distinguishes individual track segments. Mainline and branch numbers: 0=SINGLE TRACK, 1-9=MULTIPLE MAIN LINES (For example, 1=NORTH MAIN, 2=SOUTH MAIN).
TEST_DT	DATE	The date on which testing was performed.
DEF_NBR	INTEGER	Every defect is detected by a unique id Defect number.
GEO_CAR_NME	CHARACTER	Geometry cars names.
DEF_PRTY	CHARACTER	The severity of the defect: Yellow or red
DEF_LGTH	INTEGER	The length of defect in feet DEF_LGTH is reported by the measurement car.
DEF_AMPLTD	DECIMAL	The Defect amplitude – maximum size of defect in inches or degrees within defect length.
TSC_CD	CHARACTER	The Track codes including tangent, spiral and curve.
CLASS	CHARACTER	All tracks get a number between one and five. Each class represents operating speed limits for passenger and freight traffic. Class one has the lowest speed limit and class five has the highest speed limit.
TEST_FSPD	CHARACTER	Operates speed for freight trains.
TEST_PSPD	CHARACTER	Operates speed for passenger trains. If the value = 0, then it means that it does not have passenger traffic.
DFCT_TYPE	CHARACTER	The defect type the geometric defect type such as $XLEVEL$, $SURFACE$, and DIP.
TOT_DFLT_MGT	DECIMAL	The sum of total gross tons traveling across the section.

3.1 Solution presentation

Let us consider the following continuous maximization problem

$$\max\left\{\frac{3}{x_1+(-2)}; 1.8 - x_2\right\} \tag{1}$$

At Figure 4 the problem is presented as a tree, with different node types.

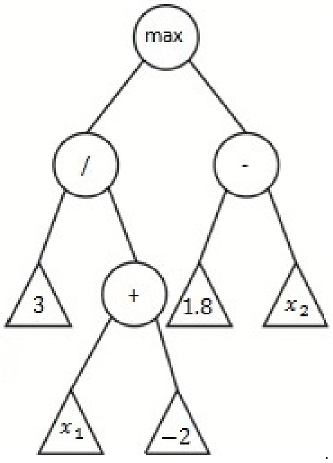


Figure 4: Solution presentation in GP

There are 9 nodes in the tree. Functional set is presented by $n = 4$ circular nodes, $\{\max, /, -, +\}$; there are $k' = 2$ variables and $k'' = 3$ coefficients, therefore $k = 5$. They are presented as triangles at Figure 4. The transformation of the tree to expression (1) is obtained by reading tree from left to right (see e,g, [7]). VNP uses the presentation of solution as a tree as well. However, a new extended solution presentation introduces the parameter set $P = \{\alpha_1, \alpha_2, ..., \alpha_k\}$. Each parameter is assign to each terminal (triangle) node of the tree. It allows the adjustment of the terminal set values by multiplying variables with corresponding coefficients. Obviously, the parameter set and the terminal set have the same cardinality k. Figure 5 represents maximization problem (1) in a new way, with parameter values taken at random from $[0, 1]$.

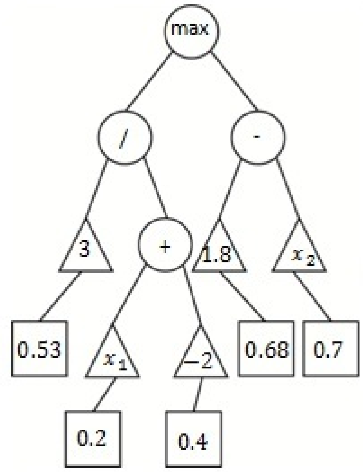


Figure 5: Solution representation in VNP

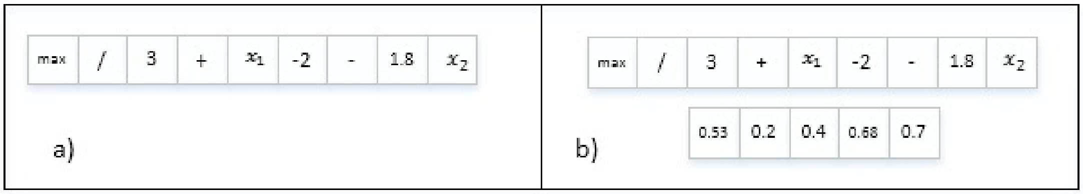
For the sake of clarity, in this section we give this new solution presentation at the same example. Figure 5 illustrates the following expression:

$$\max\left\{\left(\frac{0.53\times 3}{0.2\times x_1+0.4\times(-2)}\right); 0.68\times 1.8-0.7\times x_2\right\}, \tag{2}$$

where $F = \{max, /, +, -\}$, $E = \{x_1, x_2\} \cup \{3, -2, 1.8\}$ and $P = \{0.53, 0.2, 0.4, 0.68, 0.7\}$ The expression (1) can be also written in a *prefix* notation [7]:

$$\max(/(3 + (x_1 - 2))(-(1.8\ x_2)) \tag{3}$$

The tree structure of the expression (1) can be presented in the computer memory as an array whose number of elements equals to the number of nodes (see Figure 6 a). The tree of the expression (2) needs two arrays to be saved in the computer as given in (Figure 6 b). The second array is reserved to save coefficients. So, its size is equal to the number of terminal nodes.



max	/	3	+	x_1	-2	-	1.8	x_2

a)

max	/	3	+	x_1	-2	-	1.8	x_2

0.53	0.2	0.4	0.68	0.7

b)

Figure 6: Computer solution recording a) GP b) VNP

3.2 Neighborhood structures

In [12] a list of 9 possible neighborhood structures is proposed for solving AP problem by using VNP. In this paper we use just 3 of them to solve the maintenance planning of railway infrastructure problem. Again, we will illustrate them on our example. Of course, in solving maximization problem from the example, we need to change values of the set X only. In order to present more capabilities of VNP approach, let us also assume that we are solving symbolic regression problem, i.e., that we are searching for the formula that approximate some data in the best possible way. In that case we are allowed to change functional and parameter sets as well.

1. **Changing a node value operator** (N_1)**.** This neighborhood structure conserves the skeleton of the tree and changes only the values of a functional or a terminal node. Each node can obtain many values from its corresponding set. Let S be the current solution, its neighbor S' differs from S by just a single node. Figure 7 shows a move within this neighborhood structure.

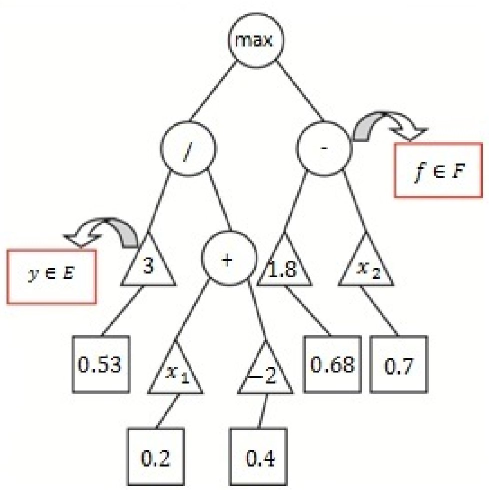


Figure 7: Neighborhood structure N_1: Changing a node value

2. **Swap operator** (N_2)**.** In this operator we choose first a node from the current tree at random and generate a new sub-tree as presented in Figure 8 a1) and Figure 8 a2). Then, we attach it in the place of the sub-tree, corresponding to the selected node. In this move, we must respect the constraint related to the maximum tree size. More details are given in Figure 8.

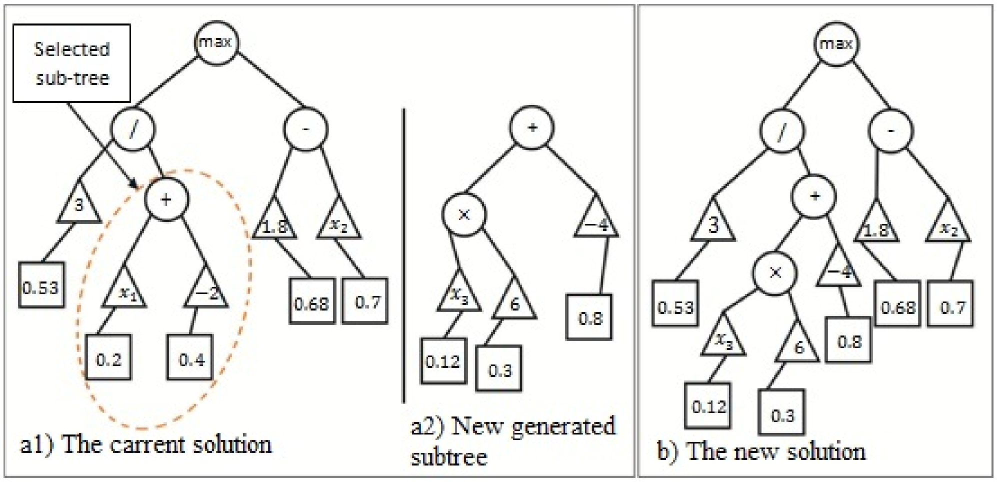


Figure 8: Neighborhood structure N_2: swap

3. **Changing a parameter value operator** (N_3)**.** In the previous neighborhood structures we just considered the tree form and its node values. Here we focus on the parameter optimization. So, we keep the position and value of nodes to search the neighbors in the parametric space. Figure 9 illustrates details. The change from one value to another is random.

The steps of the Reduced VNP are given in the pseudo code of Algorithm 1.

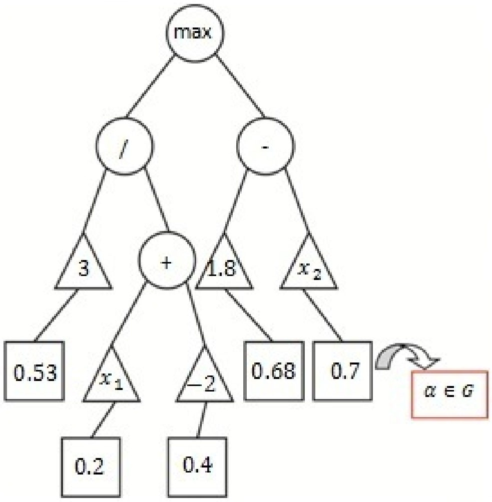


Figure 9: Neighborhood structure N_3: changing a parameter value

```
Algorithm 1: Reduced VNP($k_{max}$)
  Initialization
  - Define the set of neighborhood structures $N_k, k = 1...k_{max}$;
  - Select appropriate set of functions $F$ and terminal set $E$;
  - Generate an initial tree $T$ at random as presented in Figure 1;
  - Choose the stopping condition;
  begin
      while termination condition is not met do
          $k \longleftarrow 1$
          while $k \leq k_{max}$ do
              $T' \longleftarrow$ Shake$(T, k)$// Find a neighbor T’ from $N_k(T)$
              $k \longleftarrow k + 1$
              if fitness$(T')$ is better than fitness$(T)$ then
                  $T \longleftarrow T'$ ; $k \longleftarrow 1$; // Move
      return $T$
```

4 Reduced VNP for solving the preventive maintenance planning of railway infrastructure

The preventive maintenance planning of railway infrastructure (PMPoRI) consists on identifying defects, which have to be corrected immediately. The developed solution should predict the color of the defect after a given period (more details are explained in Section 2). A huge dataset with about twenty attributes are provided. Each line in the file corresponds to one defect and each defect is recorded in a known milepost value. The hardness of the problem is presented by the fact that the color of the defect we need to assign is given for the future time period given by organizers. This period differs from one defect to another. To minimize the difficulty, we decide to divide the original problem into two different sub-problems: prediction and classification. The first one is the prediction of attribute values for a given period. Then the second task is a classification of defects, based on data obtained by prediction. Reduced VNP algorithm can be applied in both phases. In fact the VNP solution can be presented in different complexities. Therefore, during the execution, a solution can change its size and its shape to satisfy to the problem requirements. Next sections illustrate the application of the proposed interwoven algorithm Reduced VNP in the two cited problems. In the two stages, we need a training step to extract statistical features, and a testing step to evaluate resulting model.

4.1 Learning for Stage 1 – Prediction

The prediction in this work consists in finding a function that approximates the sample data. Before starting this phase, we have to analyze and organize a data sample. When studying the available data we distinguish three types of defects. Each defect differs from the two others in the amplitude measure and in the conditions, which lead to red level. So we decided to study each defect alone. Most columns in the dataset are not time-dependent. So we have to select the ones which are responsible for the determination of the defect severity after a determined number of days. When we understand the role of each column, we can note that the only column, which must be updated after a specified time, is the defect amplitude attributes "DEF_AMPLTD". This indicates that our output is the updated attribute "DEF_AMPLTD". Now what remains is just to determine the other prediction inputs. This step is accomplished by many tests. In each test, we select a set of attributes and according to training error we decide to remove or to add attributes to the input list. In fact we can note that the sum of total gross tons traveling across the section column "TOT_DFLT_MGT" and the operating speed for freight trains and passenger train columns "TEST_FSPD" and "TEST_PSPD" seem to be good attributes to perform the prediction step and find a good updated value of the defect amplitude. After defining the inputs and the output of our algorithm we have to collect and filter data samples to learn the VNP algorithm. The problem here is that the inspection run is not always done in the same milepost point. That is to say the information of the value of defect amplitude in a given milepost point after a certain period is not available. To overcome this, we have to consider that two milepost values (having the same line segment number and the same track segments number and separated by a value less than 0.001) are in fact the same point (Table 2). The attributes corresponding to the milepost recorded in an earlier date form the inputs however; the defect amplitude value corresponding to the milepost recorded in the recent date forms the output. Table 3 shows how the information in the Table 2 is used for prediction.The "DEF_AMPLTD" value recorded at 18 August 2013 is considered as input and "DEF_AMPLTD" value recorded at 17 September 2013 is the desired output since its the most recent.

Table 2: Example of the collected samples

Variable	Input1	Input2
LINE_ SEG_ NBR	2	2
MILEPOST	21.61269	21.62043
TRACK_ SDTK_ NBR	0	0
TEST_ DT	18 Aug 2013	17 Sep 2013
DEF_ AMPLTD	1.34	1.51
CLASS	5	5
TEST_ FSPD	70	70
TEST_ PSPD	79	79
DEF_ PRTY	DIP	DIP
TOT_ DFLT_ MGT	4.60	5.11

Table 3: Inputs and output sample in the learning data

Inputs				Output
DEF_AMPLTD	TEST_FSPD	TEST_PSPD	TOT_DFLT_MGT	DEF_AMPLTD
1.34	70	79	4.6+5.11	1.51

4.2 Learning for Stage 2 – Classification

Classification is the processing of searching a set of models, which distinguish and characterize data classes or concepts. In our work, a model is a program presented as a tree (Figure 5). Indeed a program is a mathematical function, which fits the data points to be able to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data. We can fix two different classes. The first gathers the milepost points having red defects and the second gathers those having yellow defects.

In the present work, the classification is more complicated than the prediction. In fact, we aim, in this step, to extract the features of each class. At the beginning of our research, initial algorithm did not give satisfactory results. After many tests, we note that for each defect the extracted features depend on the class of tracks attribute “CLASS”. It represents operating speed limits for passenger and freight traffic. Therefore, we decide to divide the dataset again, according to the defect type and to the class of each defect. An example of a program presentation is given at Figure 10. The value of a program corresponds to the output. We fix the value “0” for yellow class and “1” for the red class. If the result is less than “0.5”, it means that the corresponding inputs belong to the yellow class. Otherwise it belongs to the red one. The general model can be written as follow:

$$F(r,q,t,z,w) = \begin{cases} f_1(r,q,t) & \text{if} \quad z=2 \quad \textit{and} \quad w = \text{“}DIP\text{”} \\ f_2(r,q,t) & \text{if} \quad z=3 \quad \textit{and} \quad w = \text{“}DIP\text{”} \\ f_3(r,q,t) & \text{if} \quad z=4 \quad \textit{and} \quad w = \text{“}DIP\text{”} \\ f_4(r,q,t) & \text{if} \quad z=5 \quad \textit{and} \quad w = \text{“}DIP\text{”} \\ f_5(r,q,t) & \text{if} \quad z=2 \quad \textit{and} \quad w = \text{“}SURAFCE\text{”} \\ f_6(r,q,t) & \text{if} \quad z=3 \quad \textit{and} \quad w = \text{“}SURAFCE\text{”} \\ f_7(r,q,t) & \text{if} \quad z=4 \quad \textit{and} \quad w = \text{“}SURAFCE\text{”} \\ f_8(r,q,t) & \text{if} \quad z=5 \quad \textit{and} \quad w = \text{“}SURAFCE\text{”} \\ f_9(r,q,t) & \text{if} \quad z=2 \quad \textit{and} \quad w = \text{“}XLEVEL\text{”} \\ f_{10}(r,q,t) & \text{if} \quad z=3 \quad \textit{and} \quad w = \text{“}XLEVEL\text{”} \\ f_{11}(r,q,t) & \text{if} \quad z=4 \quad \textit{and} \quad w = \text{“}XLEVEL\text{”} \\ f_{12}(r,q,t) & \text{if} \quad z=5 \quad \textit{and} \quad w = \text{“}XLEVEL\text{”} \end{cases} \tag{4}$$

Where r is the DEF_AMPLTD, q is the TEST_FSPD, t is the TEST_PSPD, z is the CLASS of speed and w is the type of defect. This function draws a curve separating the two classes yellow and red.

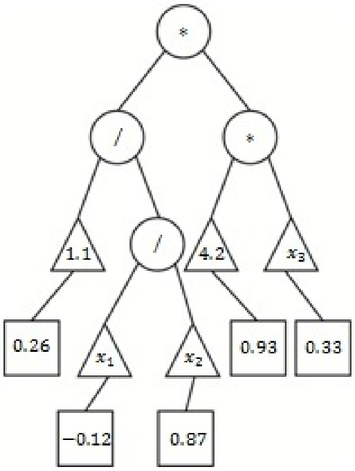


Figure 10: Program presentation in classification problem; x_1 is the DEF_AMPLTD input, x_2 is the TEST_FSPD input and x_3 is the TEST_PSPD input

5 Computation results

5.1 Prediction

To apply VNP algorithm we have to follow four steps:

1. Terminal and function sets identification
2. Neighborhood structure choice with due respect to selection criteria
3. Parameter adjustments
4. Definition of an adaptability function

Table 4: Reduced VNP parameters adjustment for prediction problem

The functional set	$F = \{+, -, *, /\}$
The terminal set	DEF_AMPLTD, TEST_FSPD, TEST_PSPD, TOT_DFLT_MGT $\cup\, c, c \in [-10, 10]$
The parameter set	$P = \alpha \in [0, 1]$
Neighborhood structures	N_1, N_2, N_3
Minimum tree length	3 nodes
Maximum tree length	1500 nodes
Maximum iteration number	1000 iterations
Fitness function	MAE Error

Functional set includes just arithmetic operators. Terminal set includes the input attributes and random real numbers between -10 and 10.The parameter set values serve to give a coefficient α to each terminal node thereby not giving the same importance to each of them. Therefore the value of α is always between 0 and 1. We use all neighborhood structures presented in the previous section. The next table summarizes the Reduced VNP parameters adjustment. The learning step for each defect is run separately. During this process, the algorithm searches to minimize the error between the desired output and the output of our algorithm. The error used is the Mean Absolute Error (MAE) and it is calculated as follows:

$$MAE = \frac{1}{n} \sum_{j=1}^{n} \left| y_t^j - y_{out}^j \right|$$

Where n represents the total number of samples, y_{out}^j and y_t^j are the VNPs given model, and the desired output of the sample number j. Our algorithm is implemented in java language and executed by Intel core i3 processor (2.3 GHz) using Windows 7 Operating System. The average time of execution is five minutes for the learning process of each defect and the MAE in this step is 0.07. The prediction provides a mathematical function, which approximates the maximum points of data, and this function is used to update the amplitude defect value after the period mentioned in the testing dataset. The outputs of this phase are employed in the classification phase.

5.2 Classification

Before running the algorithm we have to proceed as we did in the prediction step and fix the functional and terminal sets and the other parameters. Indeed we choose the same selection as shown in the Table 3, the only difference is in the terminal set, it includes:{DEF_AMPLTD, TEST_FSPD, TEST_PSPD} $\cup$ c, c is selected uniformly at random from the range of [-10,10]. To measure the efficiency of our method in this classification problem, we employed another performance measure named Accuracy (Acc). Accuracy presents the percentage of correctly classified instances. It is one of the most popular metric in this area:

$$Acc = \frac{TN + TP}{TP + FP + FN + TN} \tag{5}$$

Where TP, TN, FP and FN are the True Positive, True Negative, False Positive, and False Negative, respectively. After several experiments, we reach an accuracy value equal to 99.7% in the training process and we find a good representative model for each class in each defect. These models are just applied in the testing phase. RAS informed as that the application of our found model in the testing datatset gives a error equal to 23%. They appreciated the quality of our work and they gave as a honor mention for the originality of our idea. In this competition we had the fourth place.

6 Conclusions and future works

This paper proposes a decision support system, which is able to plan railway track maintenance. It consists in determining the color of a railway defect at a given date. This approach is based on the new automatic programming algorithm Reduced VNP. In fact, to solve the present problem, we divided it into a prediction problem and a classification problem. The prediction serves to update the important attributes after a number of days. The classification provides the color of a given defect based on the information provided by the prediction step. Reduced VNP algorithm is applied in the two problems and gives good results in the training and testing stages.

References

[1] G. Feng, L. Lin, W. Chen, Novel continuous function prediction model using an improved Takagi-Sugeno fuzzy rule and its application based on chaotic time series, Engineering Applications of Artificial Intelligence 55 (2016) 155–164.

[2] L. Cheng-Jian, C. Cheng-Hung, Identification and prediction using recurrent compensatory neuro-fuzzy systems, Fuzzy Sets and Systems 150 (2005) 155–164.

[3] L. Kewen, L. Lu, Z. Jiannan, M. K. Taghi, L. Timing, The improved grey model based on particle swarm optimization algorithm for time series prediction, Engineering Applications of Artificial Intelligence 55 (2016) 285–291.

[4] W. Jiaqiu, T. Ioannis, Z. Chen, A space-time delay neural network model for travel time prediction, Engineering Applications of Artificial Intelligence 52 (2016) 145–160.

[5] G. Akhil, G. Ankit, T. K., S. S., An integrated SRM-multi-gene genetic programming approach for prediction of factor of safety of 3-D soil nailed slopes, Engineering Applications of Artificial Intelligence 30 (2014) 30–40.

[6] J. R. Koza, Genetic programming: on the programming of computers by means of natural selection, MIT Press Cambridge, MA, USA, 1992.

[7] J. R. Koza, Genetic programming II: automatic discovery of reusable programs, MIT Press Cambridge, MA, USA, 1994.

[8] P. Musilek, A. Lau, M. Reformat, L. Wyardscott, Immune programming, Information Sciences 176 (8) (2006) 972–1002.

[9] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Transactions on Evolutionary Computation 3 (2) (1999) 82–102.

[10] E. Z-Flores, L. Trujillo, O. Schütze, P. Legrand, A Local Search Approach to Genetic Programming for Binary Classification, in: Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO '15, ACM Press, New York, New York, USA, 2015, 1151–1158.

[11] R. Ffrancon, M. Schoenauer, Memetic Semantic Genetic Programming, in: Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO '15, ACM Press, New York, New York, USA, 2015, 1023–1030.

[12] S. Elleuch, B. Jarboui, N. Mladenovic, Variable neighborhood programming - A new automatic programming method in artificial intelligence, Tech. rep. G-2016-21, GERAD (2016).

[13] N. Mladenović, P. Hansen, Variable neighborhood search, Computers & Operations Research 24 (11) (1997) 1097–1100.

[14] M. García-Torres, F. Gómez-Vela, B. Melián-Batista, J. M. Moreno-Vega, High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach, Information Sciences 326 (2016) 102–118.

[15] N. Mladenović, R. Todosijević, D. Urošević, Less is more: Basic variable neighborhood search for Minimum differential dispersion problem, Information Sciences 326 (2015) 160–171.

[16] C. Blum, A. Roli, Metaheuristics in combinatorial optimization, ACM Computing Surveys 35 (3) (2003) 268–308.

[17] J. Brimberg, N. Mladenović, D. Urošević, Solving the maximally diverse grouping problem by skewed general variable neighborhood search, Information Sciences 295 (2015) 650–675.

[18] S. Hanafi, J. Lazic, N. Mladenovic, C. Wilbaut, I. Crevits, New VNS based 0-1 MIP Heuristics, Yugoslav Journal of Operations Research 25 (2) (2014) 343–360.

[19] M. Macchi, M. Garetti, D. Centrone, L. Fumagalli, G. Piero Pavirani, Maintenance management of railway infrastructures based on reliability analysis, Reliability Engineering & System Safety 104 (2012) 71–83.

[20] T. Lidén, Railway Infrastructure Maintenance - A Survey of Planning Problems and Conducted Research, Transportation Research Procedia 10 (2015) 574–583.