Large Scale Optimization

Emmanouil Zachariadis, Assistant Professor
Department of Management Science and Technology, Athens University of Economics and Business
E: ezach@aueb.gr
A: Evelpidon 47A and Lefkados 33 street, 9th floor, Room 906, 11362, Athens, Greece
T: +30-210-8203 674

Variable Neighborhood Search - VNS

- The Variable Neighborhood Search – VNS is an algorithmic framework for solving combinatorial optimization problems.
- The VNS family of algorithms includes metaheuristic algorithms and was introduced in 1997 by Hansen and Mladenovic
- The main idea is the "systematic change of the Move Type (Neighborhood Type) during the search in the solution space".

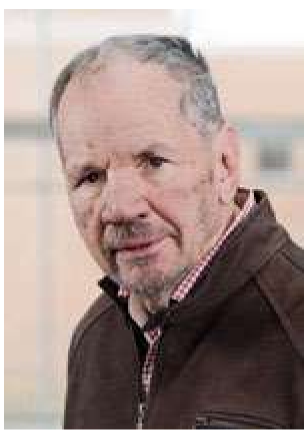
Variable Neighborhood Search - VNS

- VNS takes advantage of the following facts:

1. A local minimum according to a move type is not necessarily the local minimum according to another move type
2. The global minimum is local minimum according to all possible move types.
3. For many problems, the local minima according to different move types are usually “close” to each other in the solution space.

Variable Neighborhood Search - VNS

Nenad Mladenovic



Pierre Hansen

Pergamon

PII: S0305–0548(97)00031–2

Computers Ops Res. Vol. 24, No. 11, pp. 1097–1100, 1997

0305-0548/97 $17.00+0.00

VARIABLE NEIGHBORHOOD SEARCH

N. Mladenović‡ and P. Hansen†§

GERAD and École des Hautes Études Commerciales, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7 Canada

(Received December 1996; in revised form April 1997)

Scope and Purpose—In the last decade much progress was made in the design and application of heuristic algorithms for a large variety of combinatorial and nonconvex continuous optimization problems. General heuristics (or metaheuristics, e.g. simulated annealing, tabu search, genetic search) which avoid being trapped in the first local optimum found have led to much improved results in many practical contexts. However, the sophistication of such heuristics makes it difficult to pinpoint the reasons for their effectiveness. We examine a relatively unexplored reason: change of neighborhood in the search. Using this idea and very little more, i.e., only a local search routine, leads to a new metaheuristic, which is widely applicable. First computational results show this scheme outperforms other heuristics for several combinatorial optimization problems. Its effectiveness is illustrated on the traveling salesman problem without and with backhauls, which have a large number of practical applications such as the drilling of printed circuit boards, automated warehouse routing and operation sequencing on numerically controlled machines.

Abstract—Systematic change of neighborhood within a local search algorithm yields a simple and effective

Variable Neighborhood Search - VNS

- Many variants of the basic VNS scheme have been proposed and applied.
- The variants search the solution space with different ways:
  - Deterministic
  - Stochastic

Variable Neighborhood Descent - VND

- The Variable Neighborhood Descent searches in the solution space in a deterministic way (there is absolutely no stochastic element in the procedure)
- Different move/neighborhood types are defined ($N_1, N_2, \dots, N_{k_{max}}$)
- Set $k \leftarrow 1$, and find the best neighborhood solution of incumbent solution s'
- If $z(s') < z(s)$, set $s \leftarrow s'$ and continue with $k \leftarrow 1$
- Otherwise, set $k \leftarrow k + 1$ and continue by finding the best solution in neighborhood $N_k(s)$
- …
- Continue until none of the defined move types can improve the incumbent solution s

Variable Neighborhood Descent - VND

Initialization:

Define an ordering with k_{max} move types $(N_1, N_2, \dots, N_{k_{max}})$

Set $k \leftarrow 1$

Main Logic:

while $(k \leq k_{max})$:

Find best neighborhood solution $s' \in N_k(s)$

if $(z(s') < z(s))$

$s \leftarrow s'$

$k \leftarrow 1$

else

$k \leftarrow k + 1$

end while

return s

Basic VNS Scheme

- The basic VNS scheme included stochastic decisions in order to diversify the search of the solution space.
- The stochastic decision is a random selection of a neighboring solution (without cost/objective evaluation)
  - Shaking
  - More than one random moves may be applied
  - Also, different operators may be applied
- Next, an Iterative Improvement Algorithm (e.g. local search) is initialized starting from the randomly selected neighboring solution,
- The aforementioned algorithmic elements are included in the framework of iterating on the selected move types.

Variable Neighborhood Search - VNS

- Define an ordering with k_{max} move types $(N_1, N_2, \dots, N_{k_{max}})$

VNS Method (s)

$k \leftarrow 1$

While $(k \leq k_{max})$

Random selection of a neighborhood solution $s' \in N_k(s)$

Iterative Improvement Algorithm with move type N_k for solution s' to produce solution s''

if $(z(s'') < z(s))$

$s \leftarrow s''$

$k \leftarrow 1$

else

$k \leftarrow k + 1$

End while

return s

Variable Neighborhood Search - VNS

- The aforementioned steps can be repeated more than once, until another stopping criterion is violated (e.g., time limit)

VNS Method (s)

while ($t < cpu\_time$)

  $k \leftarrow 1$

  while ($k \leq k_{max}$)

    Random selection of a neighborhood solution $s' \in N_k(s)$

    Iterative Improvement Algorithm with move type N_k *for solution* s' *to produce solution* s''

    if ($z(s'') < z(s)$)

      $s \leftarrow s''$

      $k \leftarrow 1$

    else

      $k \leftarrow k + 1$

  end while

end while

return s

General Variable Neighborhood Search - GVNS

- Finally, the Iterative Improvement Algorithm can be replaced by a more powerful algorithmic element (e.g., an algorithm that uses many neighborhood types, like VND): General Variable Neighborhood Search

  VNS Method (s)

  while $(t \ < cpu\_time)$

  $k \leftarrow 1$

  while $(k \leq k_{max})$

  Random selection of a neighborhood solution $s' \in N_k(s)$

  $s'' = \ VND(s')$

  if $(z(s'') < z(s))$

  $s \leftarrow s''$

  $k \leftarrow 1$

  else

  $k \leftarrow k + 1$

  end while

  end while

  return s

Reduced Variable Neighborhood Search – Reduced VNS

- Both aforementioned algorithmic frameworks included a computationally demanding step: The complete exploration of neighborhood $N_k(s)$ to select the best solution.
- For large and very large-scale instances, the local search requires high computational times and power. In order to tackle this, the Reduced VNS has been proposed.
- The Reduced VNS strategy can be seen as the basic VNS scheme without the Iterative Improvement Algorithm.

Reduced Variable Neighborhood Search – Reduced VNS

- The aforementioned steps can be repeated more than once, until another stopping criterion is violated (e.g., time limit)

VNS Method (s)

while ($t < cpuTime$)

$k \leftarrow 1$

while ($k \leq k_{max}$)

Random selection of a neighborhood solution $s' \in N_k(s)$

~~*Iterative Improvement Algorithm with move type* N_k *for solution* s' *to produce solution* s''~~

if ($z(s') < z(s)$)

$s \leftarrow s'$

$k \leftarrow 1$

else

$k \leftarrow k + 1$

end while

end while

return s

RVNS - Example

- The management department of a color production factory seeks to minimize the total preparation (change-over time) of the production line that produces 5 different colors.
- More specifically, after the production of a specific color, the production line needs to be cleaned in order to be used again. This requires a specific change-over time, which depends on the color produced earlier and, on the color, to be produced.
- The change-over times of the production line for each possible pair of colors are given in the following matrix.

RVNS - Example

	1	2	3	4	5
1	0	56.92	70.45	36.24	68.15
2	56.92	0	18.87	72.37	12.04
3	70.45	18.87	0	90.25	19.31
4	36.24	72.37	90.25	0	80.22
5	68.15	12.04	19.31	80.22	0

- Apply the Reduced Variable Neighborhood Search-RVNS for solving the described problem using the following neighborhood types: Neighborhood N1 includes all solutions produced by the use of 1-0 Exchange move type and neighborhood N2 includes all solutions produced by 1-1 Exchange move type.

RVNS - Example

Random number generator (between 0 and 1)

0.689	**0.400**	**0.555**	**0.950**	**0.045**	**0.601**	**0.281**	**0.469**	**0.133**	**0.351**
0.804	**0.600**	**0.538**	**0.367**	**0.576**	**0.913**	**0.570**	**0.823**	**0.059**	**0.098**
0.452	**0.700**	**0.024**	**0.912**	**0.232**	**0.805**	**0.614**	**0.320**	**0.513**	**0.516**
0.852	**0.966**	**0.835**	**0.717**	**0.812**	**0.816**	**0.672**	**0.699**	**0.196**	**0.652**
0.492	**0.113**	**0.058**	**0.743**	**0.679**	**0.320**	**0.731**	**0.672**	**0.799**	**0.068**
0.652	**0.659**	**0.543**	**0.282**	**0.671**	**0.996**	**0.696**	**0.897**	**0.490**	**0.899**
0.196	**0.051**	**0.517**	**0.866**	**0.902**	**0.926**	**0.297**	**0.191**	**0.398**	**0.915**
0.152	**0.324**	**0.960**	**0.334**	**0.440**	**0.564**	**0.509**	**0.747**	**0.426**	**0.940**
0.830	**0.355**	**0.939**	**0.659**	**0.702**	**0.150**	**0.764**	**0.518**	**0.588**	**0.228**

Let an initial solution $s = \{4,3,1,2,5\}$ with cost $z(s) = 229.66$

RVNS – Solution (1)

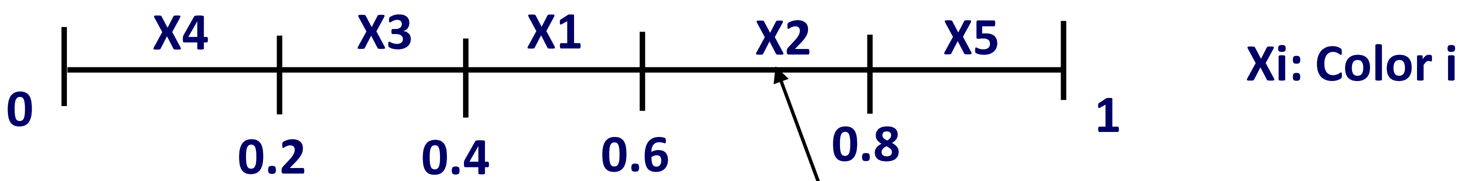
- We have selected *k*=2 neighborhood types:
  - N_1 is the neighborhood produced with the use of 1-0 move operator (relocate)
  - N_2 is the neighborhood produced with the use of 1-1 move operator (swap)

- Let an initial solution $s = \{4,3,1,2,5\}$ with objective $z(s) = 229.66$, which has been returned from a greedy construction algorithm.

RVNS – Solution (2)

Using neighborhood N1

- According to RVNS, we randomly select a solution s' of neighborhood N1(s) (Step 2 of RVNS).
- This requires the stochastic selection of a node that will be used for the 1-0 Exchange move type. The initial solution is $s = \{4,3,1,2,5\}$ with cost $z(s) = 229.66$.

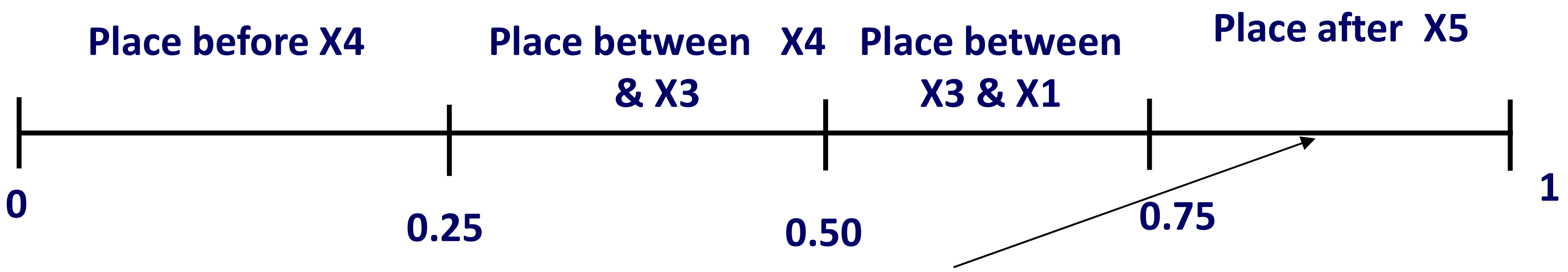


The first random number is *0.689* which is in the range (0.6 -0.8). Therefore, node 2 of the incumbent solution is selected. Note that the order of the colors on the axis, is identical to the order of the nodes in the solution $s = \{4,3,1,2,5\}$.

RVNS – Solution (3)

Using neighborhood N1

- Next, we decide stochastically the position to place node 2. We separate the range$(0, 1)$ to 4 equal ranges:



- The next random number is 0.804 which is in range $(0.75, 1)$. Therefore, the new solution is $s' = \{4,3,1,5,2\}$ with cost $z(s') = 240.89$.

RVNS – Solution (4)

Using neighborhood N1

- Given that the objective is the minimization of the change-over time of the production line for producing the 5 colors, the new solution s' is worse that the incumbent solution s as $z(s') = 240.89 > z(s) = 229.66$.
- Solution s' does not reduce the cost and according to RVNS, <u>we change the neighborhood type</u> in an attempt to escape the current local minimum.

RVNS – Solution (5)

Using neighborhood N2

- Since, the new solution produced by neighborhood type N1 is worse than $s = \{4,3,1,2,5\}$, s will remain the incumbent solution for using neighborhood type N2.

- According to RVNS, we randomly find a solution s' from neighborhood type N2(s) (Step 2).

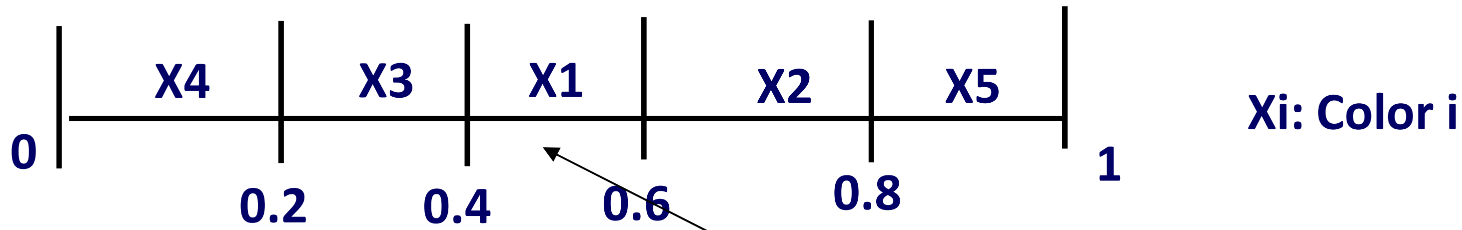
- Given that 1-1 Exchange move type implements the swap of two nodes, we have to randomly select 2 nodes that will swap positions.

RVNS – Solution (6)

Using neighborhood N2

Random selection of the 1st node:

- The initial solution s={4,3,1,2,5} with z(s) = 229.66. Divide range (0,1) in 5 equal ranges.



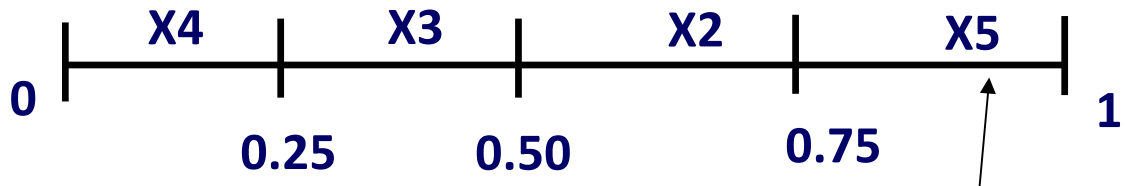
- The next random number is *0.452* in range(0.4 -0.6). Therefore, node 1 of the incumbent solution is selected for the 1-1 Exchange move.

RVNS – Solution (7)

Using neighborhood N2

<u>Random selection of the 2nd node</u>:

- Divide range (0,1) in 4 equal ranges.



Xi: Color i

- The next random number is *0.852* in range (0.75 -1). Therefore, node 5 of the incumbent is selected as the second node for the 1-1 Exchange move type.
- The new solution after the random swap of the two nodes is s:{4,3,5,2,1} with z(s) = 178.52

RVNS – Solution (8)

Using neighborhood N1

- Given that the objective is the minimization of the change-over time of the production line for producing the 5 colors, the new solution s' is better that the incumbent solution s as $z(s') = 178.52 < z(s) = 229.66$. Therefore, we set it as incumbent s=s'
- According to VNS, whenever the solution is improved, we return the first neighborhood type **N1**. Therefore, we attempt to generate a new solution from neighborhood N1 of the new incumbent.

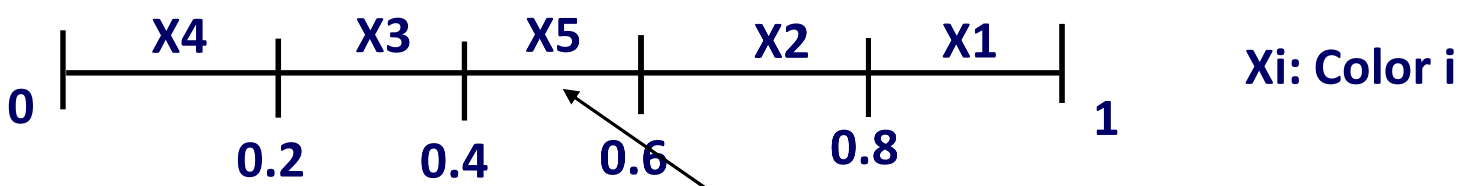
<u>Random selection of the node</u>:

- The incumbent solutions is s: $\{4,3,5,2,1\}$ with $\mathrm{z}(s) = 178.52$. Divide range (0,1) in 5 equal ranges.

RVNS – Solution (9)

Using neighborhood N1

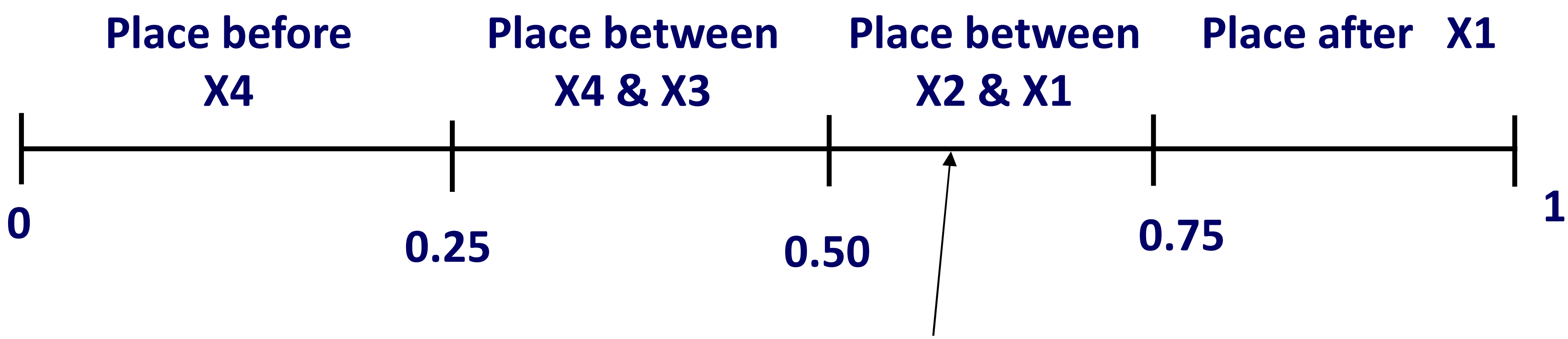
- According to RVNS, we randomly select a solution s’ from neighborhood N1(s) (Step 2).
- This requires the stochastic selection of a node that will be used for the 1-0 Exchange move type. The initial solution is $s =: \{4,3,5,2,1\}$ με κόστος $z(s) = 178.52$.



The next random number is *0.492* in range (0.4 -0.6). Therefore, node 5 of the incumbent is selected.

RVNS – Solution (10)

Using neighborhood N1



The next random number is *0.652* in range (0.5 -0.75). Therefore, the new solution is $s' = \{4,3,2,5,1\}$ with cost $z(s') = 189.31$.

RVNS – Solution (11)

Using neighborhood N1

- Given that the objective is the minimization of the change-over time of the production line for producing the 5 colors, the new solution s' is worse that the incumbent solution s as $\mathrm{z}(s') = 189.31 > z(s) = 178.52$.
- Solution s' does not improve the objective and therefore, we change the neighborhood.

RVNS – Solution (12)

Using neighborhood N2

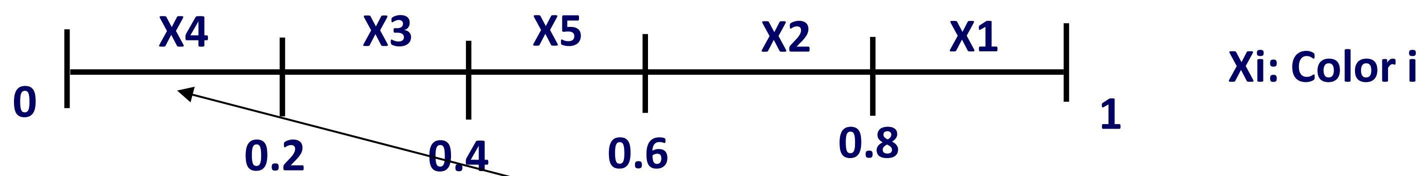
- Since, the new solution produced by neighborhood type N1 is worse than $s = \{4,3,5,2,1\}$, s will remain the incumbent solution for using neighborhood type N2.
- According to RVNS, we randomly select a solution s’ from neighborhood N2(s) (Step 2).
- Given that 1-1 Exchange move type implements the swap of two nodes, we have to randomly select 2 nodes that will swap positions.

RVNS – Solution (13)

Using neighborhood N2

Random selection of the 1st node:

- The incumbent solution $s = \{4,3,5,2,1\}$ with $z(s) = 178.52$. Divide range (0,1) in 5 equal ranges.



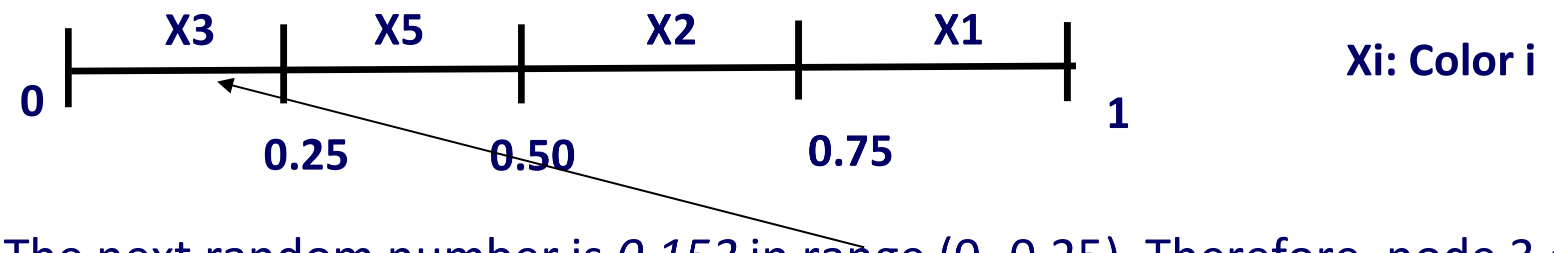
- The next random number is 0.196 in range (0-0.2). Therefore, node 4 of the incumbent solution is selected as 1st node for the 1-1 Exchange move.

RVNS – Solution (14)

Using neighborhood N2

<u>Random selection of the 2nd node</u>:

- Divide range (0,1) in 4 equal ranges.



- The next random number is *0.152* in range (0 -0.25). Therefore, node 3 of the incumbent is selected as the second node for the 1-1 Exchange move type.
- The new solution after the random swap of the two nodes is s:{3,4,5,2,1} με z(s) = 239.43

RVNS – Solution (15)

Using neighborhood N2

- Given that the objective is the minimization of the change-over time of the production line for producing the 5 colors, the new solution s’ is worse that the incumbent solution s as $z(s') = 239.43 > z(s) = 178.52$.
- The solution s' does not reduce the cost and according to RVNS, the algorithms is terminated after iterating over all $k = 2$ defined move types.
- Therefore, the best solution is s: $\{4,3,5,2,1\}$ with cost $z(s) = 178.52$