Topologically sensitive Variable Neighborhood Search - Design and the initial results

Vladimir Filipović

Aleksandar Kartelj

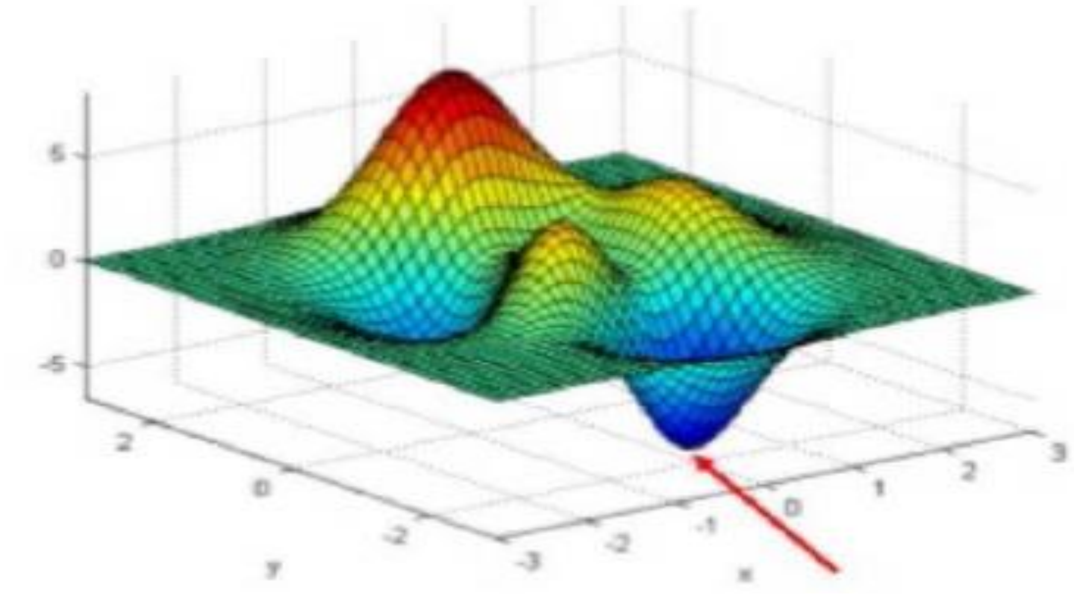
Faculty of Mathematics, University of Belgrade

Contents

INTRODUCTION

- Tremendous growth of interest in applications of optimization
- Optimization is frequently used for designing and modeling complex systems
- Metaheuristic are used for solving complex optimization problems
- Mathematics helps in designing metaheuristic methods and in determining its characteristics
- This presentation is focused on Metric Dimension Problem (MDP) for graphs and Variable Neighborhood Search (VNS) metaheuristics

OPTIMIZATION PROBLEMS

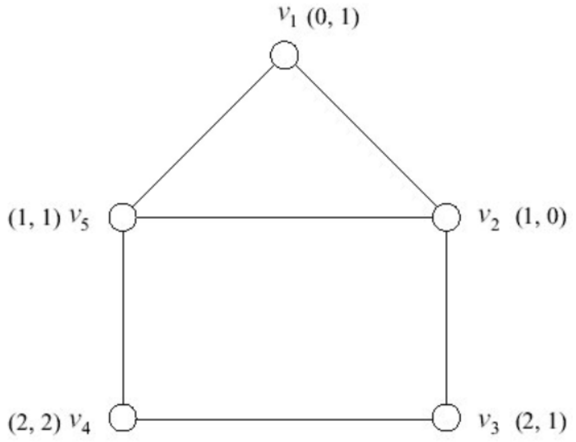


- Following elements are known:
  - search space S
  - solution space X, $X \subseteq S$
  - objective function f, $f: S \longrightarrow R$

- In minimization optimization problems, the goal is to calculate $x^* \in X$, such that $f(x^*) = min\{f(x) | x \in X\}$.

METRIC DIMENSION PROBLEM

- Graph theory is a powerful tool to model the real world applications
- Motivated by the problem of uniquely determining the location of an intruder in a network, the concept of metric dimension of a graph was introduced by Slater (metric generators were called locating sets)
- The same concept was also introduced by Harary and Melter (metric generators were called resolving sets)
- Metric dimension of a graph G is the minimum cardinality of a subset S of vertices such that all other vertices are uniquely determined by their distances to the vertices in S.



Topologically sensitive Variable Neighborhood Search - Design and the initial results

METRIC DIMENSION PROBLEM

- Formal definition:
  - Let G be a connected graph with vertex set V and edge set E
  - The distance between two distinct vertices u and v, denoted by $d(u,v)$, is the length of a shortest (u,v)-path
  - For positive integer k and a vertex $v \in V$, the k-neighborhood of v is the set $N_k(v) = \{u \in V | d(u,v) = k\}$
  - For an ordered subset $W = \{w_1, w_2, \dots, w_k\}$ of vertices and a vertex v, the code of v with respect to W is the ordered k-tuple $c_W(v) = \left(d(w_1,v), d(w_2,v), \dots, d(w_k,v)\right)$
  - The set W is a resolving set for G if every two vertices have distinct codes
  - The metric dimension of G, denoted by $dim(G)$, is the minimum cardinality of a resolving set of G
  - Resolving set containing a minimum number of vertices is called a basis
- MDP is combinatorial optimization problem
- MDP belongs to NP-hard problem class

METAHEURISTIC OPTIMIZATION METHODS

- Metaheuristic methods are generalized computational intelligence methods that can be successfully adopted to various problem domains
- They are trying to obtain the optimal solution, or the solution that is close to optimal one
- Metaheuristic algorithms are characterized with approximation and non-determinism
- Basic metaheuristics concepts are abstractly represented - they should be adapted to problem domain, otherwise they won't obtain enough good solution
- Metaheuristic methods can be population-based (Evolutionary algorithms, Particle Swarm Optimization, Electromagnetism-based Metaheuristics, etc.) or single-solution (Taboo Search, Simulated Annealing, Variable Neighborhood Search etc.)

VARIABLE NEIGHBORHOOD SEARCH

- Variable Neighborhood Search (VNS) method is a robust single-solution metaheuristic, introduced by Mladenović and Hansen
- The main searching principle of a VNS is based on the empirical evidences:
  - multiple local optima are correlated in some sense (usually close to each other)
  - a local optimum found in one neighborhood structure is not necessarily a local optimum for some other neighborhood structure

VNS ALGORITHM

```
input: k_min, k_max, it_max, iterp_max, t_max, prob, ls
output: x
x = initializeSolution();
k = k_min;
it = 1;
while it < it_max && (it-it_lastipmr)< iterp_max && t_run <t_max do
        x'  = shaking(x);
        x'' = localSearch(x', ls);
        move = shouldMove(x, x'', prob);
        if move then
                x = x";
                k = k_min;
        else if k < k_max then
                k = k + 1;
            else
                k = k_min;
        it = it + 1;
return x;
```

VNS ALGORITHM

- The main loop VNS algorithm usually imposes three main procedures: shaking, local search (LS) and neighborhood change.
  - Shaking - in order to escape local suboptimal solutions, a new solution within a parametrized neighborhood of the current best solution is generated.
  - Local search - starting from the new solution obtained in the previous step, other possible solutions within local neighborhood are systematically examined with the aim of finding the local optimum.
  - Neighborhood change - depending on the success of the previous two procedures, the current neighborhood size is adjusted. More precisely, when the current best solution is changed, neighborhood size is reduced to minimal, otherwise it is cyclically increased by 1 (cycle ends at maximal neighborhood size)
- Procedures are iteratively called, until no further improvements of the best solution can be made inside the current neighborhood. When that appears, the algorithm steps into the next neighborhood

VNS ALGORITHM

- Concerning achieving balanced search performances, e.g. balanced ratio between intensification and diversification of the search process:
  - The local search step performs search intensification (exploitation) while the shaking step is related to the diversification of search (exploration).
  - Larger neighborhoods directs toward stronger diversification, while smaller neighborhood forces intensification within search space

Topologically sensitive Variable Neighborhood Search - Design and the initial results

TOPOLOGY AND METAHEURISTICS

- There are two domains where topology an be useful for metaheuristics:
  - In analyzing metaheuristics behavior and performance
  - In designing novel metaheuristics
- Topology-based models and techniques already achieved good results in revealing hidden structures and detecting new regularities, so it can be expected that it will be helpful in this domain (for instance, in fitness landscape analysis, which includes analysis of local optima positions)
- The most important topology (more precisely, algebraic topology) concepts in this domain are simplicial complexes, homology groups and persistent homologies
- Discussion of the previously mentioned concepts is given in the paper: A. Kartelj, V. Filipović, S. Vrećica and R. Živaljević, "Topologically sensitive metaheuristics," arXiv, 2020.

TOPOLOGICALLY SENSITIVE METAHEURISTICS

- The main motivation for integrating topology and design of novel metaheuristics comes from the notion that those novel metaheuristics might use the topological regularities inside the solution space to better maneuver through it
  - For instance, if some topological regularity in fitness landscape is detected, that regularity can be exploited and used for designing metaheuristic that will perform better than the alternatives
- This can become especially useful when the solution space becomes extremely large
  - In such situation, classical metaheuristics might use too much resources in order to search the solution space.
  - Although this sounds like it could lead to premature convergence to local optima, we stress that proposed conceptual design essentially generalizes and encompasses the classical metaheuristic algorithms.
- Proposed metaheuristics, during its execution, will gradually converge to its classical variants

TOPOLOGICALLY SENSITIVE VNS

- Previously mentioned paper gives foundation for creation of various topologically sensitive metaheuristics – we are here focused on Topologically Sensitive VNS (TVNS)
- From topology perspective, classical VNS works only on top of 0-simplices and further uses 1-simplex neighborhoods for solution transitions
  - For example, for two binary-coded solutions 1011110 and 1111110 and Hamming distance function, these two 0-simplices are connected in 1-simplex when their Hamming distance is 1. Therefore, movement (within shaking) from the first one to the second one is possible.
  - In classical VNS algorithm, distance function is parametrized with k (neighborhood size).
  - Shaking procedure in classical VNS moves the current (e.g. best) solution to some other solution that is edge-connected with respect to given distance function

TOPOLOGICALLY SENSITIVE VNS

- In TVNS, neighborhood change procedure doesn't deals only with size of the neighborhood, but also with its topology
- TVNS is essentially conceived as a generalization of VNS that builds on m-simplex data (with special case $m = 0$ being a "classical" VNS)
  - We will also sometimes refer to $(m + 1)$-simplex neighborhood which corresponds to collection of all valid simplices that can be formed by adding 0-simplex to observed m-simplex
  - Therefore, 1-simplex neighborhood correspond to classical VNS, while m-simplex neighborhoods where $m > 0$ refer to its topological generalizations

Topologically sensitive Variable Neighborhood Search - Design and the initial results

TOPOLOGICALLY SENSITIVE VNS

- From topology perspective, TVNS does not work only with 0-simplex and 1-simplex like classical VNS, but it can use simplexes of higher order
- Illustration for 2-simplex neighborhood, when Hamming distance is strictly 2 :
  - With Hamming distance strictly 2, we need at least 2 previous problem solutions in order to generate the new one
  - If previous solutions are, for instance, 1011110 and 1111111, we can generate a new solution 1011011 by randomly picking it from the set of possible alternatives 1011101, 1011011, 1010111, ...
  - New solution is on Hamming distance exactly 2 from previous ones, and together with two previous solutions it forms a 2-simplex with respect to imposed distance function

TOPOLOGICALLY SENSITIVE VNS

- Illustration for 2-simplex neighborhood, when Hamming distance is at most 2 :
  - With Hamming distance at most 2, for example, by combining solutions 1011110 and 1111111 one could get new solution 1011111
  - In this case, old solutions are at Hamming distance 2, but new solution is at distance 1 to both old solutions
- Also, 2-simplex neighborhood could be combined with neighborhoods of higher cardinality
  - For example, when using strict Hamming distance 4, from solutions 1011110 and 1111111 one could get solution 1100111

TVNS ALGORITHM

- Overall structure is same as it was with previously explained classical VNS, with some changes in shaking procedure, in local search procedure and in neighborhood changing
- Adjustment within shaking procedure:
  - With m-simplex degree $m > 0$, it is clear that additional memory is required in comparison to classical VNS, but all previous solutions can not be kept in memory and fixed-sized part of memory will be used for that purpose
  - During shaking, target simplex that will be extended with new solution (0-simplex) by performing random selection among possible candidates
  - New valid solution (one that extends selected m-simplex to (m+1)-simplex) will be generated by randomly select valid solution with respect to selected simplex

TVNS ALGORITHM

- Adjustment within local search procedure:
  - Unlike selecting a single simplex from possible simplex candidates (as it was in shaking), local search checks all simplex candidates for given m and k
  - Local search operates locally which means that neighborhood size is small, $k = 1$, $k = 2$ or at most $k = 3$ (note that when using combination $\mathrm{m} = 0$, $k = 1$, the classical 1-swap local search is performed)
  - Larger neighborhoods $k > 1$ are mostly avoided in classical VNS local search procedures due to their increasing computational costs.
  - However, TVNS can check neighborhoods of size $k > 1$ for $\mathrm{m} > 0$ more efficiently since it performs exhaustive search only with respect to observed m-simplices

TVNS ALGORITHM

- Adjustment of neighborhood changing:
  - The main loop of TVNS should be made in such a way that the sequence of neighborhood structures, that are now parametrized by m and k (not only with k), starts with the most restrictive neighborhood and after that proceeds with the sequence of more relaxed ones
  - Therefore, the neighborhoods will start with smallest neighborhood size $k = k_{min}$ and the most restrictive simplex structure $m = m_{max}$, and further proceed with reduction of m by 1
  - When m reaches 0, it basically means that classical VNS algorithm is to be performed
  - After that, the k is increased by 1 and m is reset to m_{max}
  - The full cycle through neighborhoods is done when k reaches k_{max} and m reaches 0
  - If, at some moment, current solution is improved, both k and m are reset to its initial values

Topologically sensitive Variable Neighborhood Search - Design and the initial results

IMPLEMENTATION

- Implementation of the proposed TVNS for solving MDP is developed on .NET platform, using programming language C#
- Beside TVNS for MDP implantation, in order to make adequate comparison two metaheuristics are implemented: classical VNS and GA
- During computational experiments that are conducted, parameters for the classical VNS are set on the same values like TVNS parameters, and GA parameters are set on the values specified in the relevant paper (J. Kratica, V. Kovačević-Vujčić and M. Čangalović, Computing the metric dimension of graphs by genetic algorithms, Comput Optim Appl)

PROBLEM INSTANCES

- Experiments are executed on 147 different problem instances:
  - csp50 to csp500: 10 instances (50-500 vertices, 157-16695 edges);
  - cgol1to cgol30: 30 instances (100-300 vertices, 2484-22543 edges)
  - frb30-15-1 to frb59-26-5: 40 instances (450-1534 vertices, 17827-125982 edges)
  - Q1 to Q12: 12 instances (2-4096 vertices, 1-24576 edges)
  - H2,3 to H2,30: 28 instances (9-900 vertices, 18-26100 edges)
  - H3,3 to H3,17: 14 instances (27-4903 vertices, 81-117912 edges)
  - H4,3 to H4,8: 6 instances (81-4096 vertices, 324-57344 edges)
  - H5,3 to H5,5: 3 instances (234-3125 vertices, 1215-31250 edges)
  - H6,3: 1 instance (729 vertices, 4374 edges)
  - H6,4: 1 instance (4096 vertices, 36864 edges)
  - H7,3: 1 instance (2187 vertices, 15309 edges)

Topologically sensitive Variable Neighborhood Search - Design and the initial results

RESULTS

- Summary results:

Instance		GA			VNS			TVNS -1, 5		
type and count	mean best known MD	mean best MD	mean time in seconds	# instances with achieved best	mean best MD	mean time in seconds	# instances with achieved best	mean best MD	mean time in seconds	# instances with achieved best
all (147)	17,7619	20,190	292,261	54	18,117	42,83563	126	**17,897**	23,578	**143**
csp (10)	16,7	19,6	50,7836	2	16,8	2,2889	9	**16,7**	2,478	**10**
gcol (30)	9	10	8,532	0	**9**	0,4947	**30**	**9**	1,095	**30**
frb (40)	30,1	36,975	431,263	0	30,55	48,7097	22	**30,175**	32,487	**37**
Q (12)	5,083333	**5,083**	235,177	**12**	5,8	55,777	10	5,454	24,662	11
H (55)	16,52727	16,945	401,252	40	**16,527**	66,67778	**55**	**16,527**	32,981	**55**

- More detailed results are publicly available in the GitHub repository

https://github.com/vladofilipovic/documents-science-public/tree/main/conferences/smscg-2023

CONCLUSIONS AND FUTURE WORK

1. Topological enhancement for the VNS method is proposed, and design of that novel method, named Topology Sensitive VNS, is elaborated
2. Obtained results are very promising, but in order to examine wider applicability more experiments on various problems should be executed
3. Further research will be focused on designing other topologically sensitive metaheuristics, on determining its theoretical characteristics and on testing and evaluating its performances on different NP-hard optimization problems

REFERENCES

1. P. J. Slater, Leaves of trees, *Congressus Numerantium* 14 (1975) 549–559
2. F. Harary and R. A. Melter, On the metric dimension of a graph, Ars Combinatoria 2 (1976) 191–195
3. N. Mladenović and P. Hansen, Variable neighborhood search, Computers & Operations Research 24 (11) (1997) 1097 – 1100
4. A. Zomorodian and G. Carlsson, Computing persistent homology, Discrete Comput. Geom. 33 (2) (2005) 249-274
5. A. Y. Lee, S. D. Barthel, P. Dlotko, S. M. Moosavi, K. Hess and B. Smit, Quantifying similarity of pore-geometry in nanoporous materials, Nature communications 8 (2017) 15396
6. J. Kratica, V. Kovačević-Vujčić and M. Čangalović, Computing the metric dimension of graphs by genetic algorithms, *Comput Optim Appl* 44, 343 (2009)

REFERENCES

7. D. Matić, J. Kratica and V. Filipović, Variable neighborhood search for solving bandwidth coloring problem, Computer Science and Information Systems 14 (2) (2017) 309 – 327
8. M. Grbić, A. Kartelj, S. Janković, D.Matić, V. Filipović, Variable neighborhood search for partitioning sparse biological networks into the maximum edge-weighted k-plexes, IEEE/ACM Transactions on Computational Biology and Bioinformatics 7(5) (2020)1822-1831
9. A. Kartelj, V. Filipović, S. Vrećica and R. Živaljević, Topologically sensitive metaheuristics, arXiv (2020)

Thanks for your attention ☺

vladimir.filipovic@matf.bg.ac.rs

vladofilipovic@hotmail.com