Response to the Reviewers' Comments for VNS with GA-based parameter tuning for solving the k-domination problem

Milan Predojević, Aleksandar Kartelj, and Marko Djukanović

April 17, 2023

We thank the reviewers for their comments. All comments have been carefully considered. Below we address all comments and describe the respective changes we have made to the manuscript. The reviewers' comments are quoted for clarity. All changes we made to the paper are highlighted in red.

Comments from *Reviewer #1*

> In this paper an interesting implementation of VNS algorithm is proposed. The only issue I found is missing explanation why the parameters are tuned using Genetic Algorithm and not some other package, for example IRACE? Other than that, the reported results are really outstanding.

Answers. TODO

Comments from *Reviewer #2*

> The main contributions of the paper are summarized in the introduction (Sec. 1). Some of them, however, are not properly supported by the experiments presented later.
>
> 1. Authors claim that the VNS significantly improves the state-of-the-art, but the comparison to the literature is not completely fair. Different from the literature approaches, the VNS uses specific parameter configurations for each instance. Besides that, the termination criteria of the literature approaches are not discussed and contrasted to those used for the VNS, and using different machines or different running time limits (for example) may impact in the quality of the results.
>
> 2. Regarding the third conclusion, which states that "VNS is able to quickly produce solutions of reasonable quality which is not the case for approaches proposed in the literature", the paper only shows the quality of the final solutions

> and the running time required by the VNS to find them, but no information about the solutions found during the execution is given. It would be interesting to visualize the performance profile of some runs, identifying the best solutions found over time, comparing with the same visualizations for the literature approaches (and then argue that the VNS is the only approach that finds good solutions quickly). Moreover, Figures 3 and 4 show running times larger than 30 minutes (i.e., larger than the termination criterion) for most instances and values of k, which cannot be considered "quickly" without a more informed discussion.

Answers.

1. In the new version of the paper, a single parameter configuration was used for all instances. The configuration was obtained by the grid search method on a random sample of 20 instances (medium of instances and k from [1,2,4]).

   In the paper, we added a didcusia that justifies the comparisons: "This choice of termination criteria is inspired by the times presented in the literature [Corcoran and Gagarin(2021)]. The best results were given by BS4, where 4 stands for beam widths. For that algorithm, the authors for k=2 showed mean running times in seconds of 1736, 8834, 1257, 7156 and 3327 for the cities of Bath, Belfast, Brighton, Bristol and Cardiff respectively.

   As stated in [Corcoran and Gagarin(2021)], all algorithms were implemented in the Python programming language and executed on a desktop computer with an Intel Core i7-8700 CPU. The criteria for termination of SG, PG and BS is finding the first feasible solution."

   Unfortunately, the source code from the literature [Corcoran and Gagarin(2021)] is not available and therefore we are not able to perform the comparisons as you described.

2. We have been offered to accept this work as a poster. We accepted that, and because of that we are limited with the number of pages (4). For this reason, we are unable to provide a detailed analysis of what is requested. Implicitly, from the definition of the VNS algorithm and the previous citation/answer, it can be concluded that VNS generates many "good" solutions in a reasonable amount of time, which is not the case with BS4.

> According to Section 3, the proposed VNS algorithm explores neighborhoods with different sizes (number of vertices added and removed from the solution). However, such variable neighborhood strategy is used only in the shaking procedure, where a neighboring solution of a given local optima is randomly selected from one of these neighborhoods. The local search does not explore different neighborhoods. Instead, it performs a simple iterative (best) improvement procedure. Based on this, can you call this algorithm a VNS?

Answers. The proposed VNS is defined in the literature [Mladenović and Hansen(1997)] and we only refer to it. I would disagree with "Local search doesn't explore different neighborhoods". Local search tries to find a local minimum (best neighbor) in a "fast" way.

> Regarding the running time limit (30 minutes), it would be interesting to scale the time limit according to the size of instance. Figures 3 and 4 show that the required computational effort increases as instance size grows. Besides that, the paper should mention the termination criteria of the approaches from the literature and how it compares to the one used in the experimental evaluation, to ensure a fair comparison of results.

Answers. Due to page limitations we are unable to display this scaling.

Thank you for your comments regarding the comparison and . In the work, we have added the following all the posuses that explain what was done:

"This choice of termination criteria is inspired by the times presented in the literature [Corcoran and Gagarin(2021)]. The best results were given by BS4, where 4 stands for beam widths. For that algorithm, the authors for k=2 showed mean running times in seconds of 1736, 8834, 1257, 7156 and 3327 for the cities of Bath, Belfast, Brighton, Bristol and Cardiff respectively.

As stated in [Corcoran and Gagarin(2021)], all algorithms were implemented in the Python programming language and executed on a desktop computer with an Intel Core i7-8700 CPU. The criteria for termination of SG, PG and BS is finding the first feasible solution."

> The parameter tuning step, detailed in Section 4.1, does not follow a well-established methodology for this task. First, the instances should be split into training and testing sets. The tuning is performed using the training set, and the parameter configurations are evaluated on the testing set. This avoids the so-called overtuning, producing parameter configurations whose performance is the same when solving new/unseen instances. As a consequence, we can compare the algorithm (with the produced configuration) with other approaches from the literature, since the latter use a single configuration/algorithm to solve all instances. In contrast, if specific configurations are used for each instance (as presented in the paper), the tuning time should be accounted in the comparison.

Answers. Yes, we did this very badly. In the new version of the paper, a single parameter configuration was used for all instances. The configuration was obtained by the grid search method on a random sample of 20 instances.

> Finally, Tables 3 - 5 present the results for the VNS and the best approach from the literature. It would be better to show the complete results, i.e. for all tested algorithms.

Answers. Due to page limitations, we are unable to provide a more detailed view. We are of the opinion that this is enough, because we have singled out the best solution with which we are comparing it.

> Minor issues:
>
> 1. More detail about the applications of the k-domination problem (lines 100 - 103) can be given.
> 2. The discussion about the results (lines 499 - 512) are repetitive, since the observations are the same for the different values of k. It could be summarized.
> 3. Figures 3 and 4 can use filling patterns for the different values of k to ease the visualization in grayscale.
> 4. The running times reported in Figures 3 and 4 are averages over how many replications?
> 5. Instead of using the experimental results reported in the literature (for SG, PG and BS), why not running them again, given that the source code is available?
> 6. If space is an issue to accomodate any suggestion, Tables 3, 4 and 5 can be aggregated in a single table, Figures 3 and 4 can be aggregated in a single figure, and Figures 1 and 2 can be replaced by a single table, aggregated with Table 1.

Answers.

1. Unfortunately, we do not have enough space for a more detailed description.
2. TODO: dobar prijedlog.
3. TODO: radicu kako je covjek rekao
4. For a maximum of 20,000 iterations/repetitions. If the displayed time is less than 1800s, it means that exactly 20,000 iterations have occurred, which terminated the VNS. We don't think this is worth discussing and that's why it's not explained in detail.
5. The source code is not available. We only received instances from the authors.
6. This is a good appetizer, thank you. We did as you said.

Comments from *Reviewer #3*

> There is a fundamental issue with the tuning performed, which is done per instance without any intention of searching for parameter configurations that can generalize to unseen instances. Moreover, it seems the same effort was not applied to tune the other competing algorithms, which makes the comparison quite unfair. In particular, beam-search has a number of very fundamental parameters that would benefit from being tuned according to the instance size.

Answers. TODO

> The approach makes even less sense in the case of parameters such as dmin and dmax that have a strong effect in constraining the set of solutions that are searched. Moreover, a wrong value of those parameters can effectively make impossible finding an infeasible solution. Thus tuning them only makes sense if one knows the actual optimal solution, which is not generally true. For unseen instances of a given size, one could calculate good values based on the properties of the graph.

Answers. TODO

> Another issue is that results on small and medium instances are taken from a different paper but it is unclear if those papers used a similar termination criterion on a similar powerful computer. Even assuming that for the small and medium instances the timelimit of 30 minutes is not reached and the VNS stops after the maximum number of 2000 iterations, it is unclear how those iterations are comparable with the termination criterion of SG, PG and BS.

Answers. TODO

> The paper lists as a contribution that the parameters are tuned. But properly tuning parameters is just a step of proper experimental comparison and not a scientific contribution. It would similar to saying that a contribution of the paper is to do statistical tests to assess significance.

Answers. TODO

> The tables should report variance (or std. deviation) to show the variance of across multiple runs. Without variances, it cannot be said if the performance difference is sufficiently clear or a statistical test is needed to assess significance.

Answers. TODO

> The tables could also report running times, both for VNS and the algorithm of the literature (there is plenty of space in the paper).

Answers. TODO

> The paper claims that VNS is stopped after 30 minutes, however, figures 3 and 4 show runtimes longer than 1800 seconds

Answers. TODO

> The stacked bars in Figs 3 and 4 are difficult to understand: For example, does a run for Manchester with k=4 take 5000 seconds or 5000 - 2500? Having the bars side by side will make more sense and avoid such questions.

Answers. TODO

> "The time limit of 30 minutes is not checked during initialization [...] This means that VNS can take more than 30 minutes to finish" -¿ However, it also means that VNS does not execute and only the initial local search has an effect. So one may wonder how much of the work reported is done by the initial local search rather than the actual VNS algorithm.

Answers. TODO

> "each processor outside the must have" -¿ the dominating set?

Answers. TODO

> "All (20) small to medium sized problem instances were tuned separately for each $k \in \{1, 2, 4\}$, resulting in a total of 60 control parameter configurations. [...] Parameter tuning for large instances was too inefficient. Therefore, we used reasonable manually configured parameters for these instances" -¿ This approach is fundamentally wrong since it is overfitting the parameters to each problem instance. Which parameter settings should be used for a new unseen instance of the problem? The fact that it was not possible to extrapolate the parameters found in small and medium sized instances to larger instances clearly indicates that nothing was learned from this tuning effort due to being too specific and not general enough.
>
> This is why actual parameter tuning methods encourage a clear separation between training and testing instances. See:
>
> https://doi.org/10.1613/jair.2861
>
> https://doi.org/10.1016/j.orp.2016.09.002
>
> Moreover, the parameters could be tuned per groups of similar instance size and an extrapolation done for larger instances: Franco Mascia, Mauro Birattari, and Thomas Stützle. Tuning Algorithms for Tackling Large Instances: An Experimental Protocol. In P. M. Pardalos and G. Nicosia, editors, Learning and Intelligent Optimization, 7th International Conference, LION 7, volume 7997 of Lecture Notes in Computer Science, pp. 410–422. Springer, Heidelberg, 2013.

Answers. TODO

> The datasets and codes could have been made available for review via the supplementary material or an anonymous Zenodo upload.

Answers. TODO

Comments from *Reviewer #4*

> The paper proposes an efficient search metaheuristic for solving the k-domination problem.
>
> It describes the motivation and method clearly. The experiment description, including parameter tuning, is detailed and could thus be likely reproduced from the paper.
>
> The algorithmic components are motivated well (both in terms of efficiency and solution quality) and demonstrably improve over the chosen baselines empirically. While I feel that I cannot judge whether the baselines and benchmark are appropriate for the evaluation, the results are convincing to me. I therefore recommend accepting the paper.

Answers. TODO

References

[Corcoran and Gagarin(2021)] Padraig Corcoran and Andrei Gagarin. 2021. Heuristics for k-domination models of facility location problems in street networks. *Computers & Operations Research* 133 (2021), 105368.

[Mladenović and Hansen(1997)] Nenad Mladenović and Pierre Hansen. 1997. Variable neighborhood search. *Computers & operations research* 24, 11 (1997), 1097–1100.