Dear Editor,

We would like to thank you for the editorial job on our article

" **Can greedy-like heuristics be useful for solving the Weighted Orthogonal Art Gallery Problem under regular grid discretization?**"

which we submitted for possible publication in *IJEEC journal.*

We also want to thank the Referees for their valuable comments. We mainly accepted them in the revised version of the paper that is enclosed in the attachment.

We answered to the Referees’ comments in a separated letter as enclosed below.

Sincerely yours,

The Authors

" **Can greedy-like heuristics be useful for solving the Weighted Orthogonal Art Gallery Problem under regular grid discretization?**""

**Answers to Reviewer 1.**

We are grateful to the Referee for his valuable comments and suggestions which have helped us to improve the paper. We now give detailed answers to all the comments.

This paper proposes the use of greedy-like heuristics for the Weighted Orthogonal Art Gallery Problem under regular grid discretization.

While I believe the problem may have interesting applications and the paper is well written, I believe it currently has important issues that need to be addressed and, in my view, it doesn't present enough novelty for publication at this moment. Similar problems and variants of the Art Gallery Problem were considered in recent papers with very successful results in practice. It's not clear to me that these algorithms couldn't be applied with little modification to solve the discretized WOAGP or even its continuous (and more challenging) version where the objective is to cover the whole polygon. The problem itself, after discretization, seems to be almost the same as solving the Weighted Set Cover Problem. That said, it would be more significant to solve the problem without discretization, aiming at the full coverage of the polygon. I also believe that the experiments are not enough and they need to be more detailed and performed on larger data sets.

**Answer:**

We completely understand the Reviewer's remark and we would like to explain the motivation of our approach.

The aim of our paper was not to propose any new state-of-the-art method for solving discrete AGP, but the investigate whether "an easy-to-implement" greedy like heuristics could be applied on the weighted variant of the problem. Having this in mind, we consider that it would be very hard to expect that our relatively simple greedy-like methods could outperform or even be comparable to the well-known and sophisticated state-of-the-art methods. On the other hand, our paper proposes a fast and simple greedy approach which could be used in practical situations whenever regular grid discretization could be applied; for example to help engineers in their tasks related to the problem, such as determining location of cameras in order to minimize the installation costs. We believe that the novelty of our approach lays in the construction of the greedy approach based on balancing the trade-off between the total sum of guards' costs and the total number of not yet covered points from the discretization, also considering different types of weights for benchmarks, based on an approximation of the costs in real situations. To the best of our knowledge, no such a greedy criterion has been considered so-far in the literature.

Still, we consider that Reviewer's comments very valuable and we tried to improve the paper according to his suggestions.

**Reviewer’s comments**

Page 1:

**Remark1**

+ Couto et al. algorithm appears in different papers on the topic, with the last and more robust version being able to solve general polygons with up to 2500 vertices (including orthogonal instances).

[1] Couto, M.C., de Rezende, P.J., de Souza, C.C.: An exact algorithm for minimizing vertex guards on art galleries. Int. Trans. Oper. Res. 18, 425–448 (2011)

How this new algorithm compares with Couto et al. work (same weight for all guards)? Is it possible to easily adapt Couto et al. work to solve this version of the problem? If yes, would be important to compare with it.

**Answer:**

As we mentioned above, we believe that the adaptation of the Couto's algorithm, as well as the algorithms presented in a survey paper written by Rezende et al. will outperform our simple greedy like algorithm (and its hybridizations as well), so we did not include such a comparison.

**Remark2**

+ I believe it's important to add as reference the following survey on Algorithms for the AGP, which presents a comparison of different state-of-the-art algorithms for optimally solving the AGP with vertex guards. It shows a big improvement (speed up) over Couto et al. approach when using C+BS-2013 algorithm, which can solve instances with 5000 vertices in minutes. This is important to mention and to consider if comparing with Couto et al. work.

[2] Pedro J de Rezende, Cid C de Souza, Stephan Friedrichs, Michael Hemmer, Alexander Kröller, Davi C Tozoni: Engineering art galleries. Algorithm Engineering, 379-417 (2016)

+ Missing references for other recent works proposing practical solutions for AGP problems. Some of them include also greedy algorithms.

- [3] Amit, Y., Mitchell, J.S.B., Packer, E.: Locating guards for visibility coverage of polygons. Int. J. Comput. Geom. Appl. 20(5), 601–630 (2010)

- [4] Fekete, S.P., Friedrichs, S., Kröller, A., Schmidt, C.: Facets for art gallery problems. Algorithmica 73(2), 411–440 (2014)

- [5] Bottino, A., Laurentini, A.: A nearly optimal algorithm for covering the interior of an art gallery. Pattern Recogn. 44(5), 1048–1056 (2011).

**Answer**

We appreciate the Referee for this remark. We included the suggested references in our paper. In the new version of the paper, the text is added in Introduction, displayed in red color.

**Page 2:**

**Remark3**

+ Is there any previous work on (continuous or discrete) WOAGP?

**Answer:**

While reviewing the literature, we could not find any relevant work concerning the weighted version of the AGP. In our paper, we only noticed that WOAGP (discrete variant) is related to the Minimum Weighted Set Cover Problem (MWSCP).

**Remark4**

+ "An anytime algorithm to compute..."?

**Answer:**

We adopted the sentence. In the new version it is phrased like:

“An anytime algorithm which computes successively better approximations of the optimum for Minimum Vertex Guard is proposed in [12].”

**Remark5**

+ Why using the regular grid discretization? It seems to not be a good choice even for the experimented data (minArea polygons). Other discretizations are discussed in previous papers that seem to obtain good results, as the one in Couto et al. [1] last work using Shadow AVPs and the one called Chwa Points in Tozoni et al.. I suggest including reference to

[6] Chwa, K., Jo, B., Knauer, C., Moet, E., van Oostrum, R., Shin, C.: Guarding art galleries by guarding witnesses. Int. J. Comput. Geom. Appl. 16(2–3), 205–226 (2006).

**Answer:**

We can agree with the Reviewer that the chosen discretization is not ideal and, in some cases, it fails to provide the full optimal covering of the whole polygon.

In future, we plan to extend our research in that direction, which we also noticed in Conclusion section (in the Future work paragraph). Reference [6] is also cited there.

**Remark6**

+ It is confusing sometimes if WOAGP refers to the discrete or continuous version of the problem. Please, use different names for the two versions of the problem.

**Answer:**

The Referee is right. We adapted the text in the sense that the discrete version is called discrete WOAGP. We corrected the text where it is needed.

Page 3:

**Remark7**

+ "Solution component of the problem is a guard"?

**Answer**

This sentence is removed from the text since we believe that it is clear from the context of our explanation what has to be included in order to extend current partial solution (not-yet-included guard).

**Remark8**

+ Meaning of "S" symbol in equations is confusing. In first line of the page (C contained in {S\_1,...,S\_i}) seems to refer to points that can be guards. Later on, it seems to refer to sets of points in D(P) that are visible from vertices.

**Answer:**

We changed the text in this paragraph, in order to be more consistent with the notation.

**Remark9**

Page 4:  
+ In Equation (10), why does it compute g' without excluding the area already covered by s^{ps}?

**Answer:** The referee is right**,** we could do this as well. But, please note that this is a secondary criterion, used for breaking ties. The main point in obtaining reasonable results is given by employing greedy criterion *g*. We believe that this small adaptation would not impact the results significantly.

+ When using hybrid GREEDY+CPLEX, why not solving greedy first, since its fast, and then providing final viable solution to CPLEX as an initial good guess?

**Answer:**

In our experimental evaluation, we have shown that Greedy heuristic delivers reasonable solution, but it was not that good as the ones obtained from Greedy+CPLEX on most of the instances we have tested. So, providing such a solution to the CPLEX (as an initial guess, or as additional constraint to the model) would most likely be just a weak bound to the optimal solution and we would not benefit a lot from it. We believe that the internal stuff that is executed as the core of the CPLEX (such as relaxations, decomposition techniques, primal heuristics, etc.) do much more from adding just a week constraint into the model. Therefore, it would probably easily outperform this suggestion.

+ I believe it would be important to experiment using a wider set of instances. <https://www.ic.unicamp.br/~cid/Problem-instances/Art-Gallery/index.html> has hundreds of random orthogonal instances with up to 5000 vertices (30 instances per size)

**Answer:** We agree with the reviewer. However, as we said, our purpose was not to beat any state-of-the-art methods but to show that some simpler approaches can be useful in practice up to some point. Running more experiments than we did in this stage would not change much from the conclusions we tend to show here, only would insist for significant extension of our experimental section.

+ It's not clear to me why cameras watching over a large range of discrete points should be more expensive.

**Answer**: One reason for that could be that those cameras are ones with better resolution – therefore more expensive.

+ Why not using information about the visibility polygons of these points to define weights?

**Answer**: Yes, that could be used. Our point-based related weights are oriented similarly to this idea.

Page 5:  
+ Is there a reason why times using W0 are so much larger than compared to W1, including for pure greedy algorithms (specially for MinArea polygons)?

**Answer**: TODO  
  
Page 6:  
+ Overall, seems to me that using ILP directly is considerably better for FAT polygons. The advantage in MinArea polygons is mostly on performance. What happens if timeout of cplex is set to be equal to the time used to produce the solution with the greedy (or hybrid) approach? Is the solution better? From the experiments, it's not possible to conclude that these greedy methods would be a better choice compared to simply using ILP.

**Answer: TODO**

+ Tables show average of results between polygons of considerably different size, making it difficult to understand how the algorithm behaves for different number of vertices. Outliers on larger instances may generate distortions on results. Also, for a small polygon with 10 vertices, having a difference of 1 in number of guards is much more significant than having difference 1 in a polygon with 200 vertices. Is there some kind of normalization in the statistics presented? It would be better to group polygons according their size.

**Answer: TODO**

+ Some places seems to have a typo where "lower" is used instead of "higher" to describe average running time.

**Answer:** TODO

Page 7:  
+ I believe missing polygon coverage around 15% is very significant in practical scenarios

**Answer: TODO**

+ Figure 4 seems to show that ILP (cplex) is a reasonable option for most cases in comparison with the other techniques presented.

**Answer: TODO**

+ Caption of Figure 4 seems to be wrong.

**Answer: TODO**