Review of the paper of
Anusha Laxman, Sayinath Udupa N. V., and N. Prathviraj
"An algorithmic approach to find N-covering sets
and an introduction to ve-poset of a graph"

In the paper under review, the authors consider a variation of the concept of the degree of a graph vertex, when the degree is understood as the sum of the classical degree (the number of incident edges) and the number of graph triangles containing this vertex. Such a concept is motivated by the connection with covering (in an appropriate sense) of all edges of the graph by some set of its vertices and numerical characteristics of such coverings that arise here.

In the presented paper, there are a number of significant inaccuracies in the above formulations of key definitions (both in terms of terminology and in terms of the content of these concepts). They sometimes lead to different interpretations and require clarification or correction. In addition, despite the given Listing of algorithms (which, unfortunately, sometimes also does not have an unambiguous understanding due to the chosen metalanguage), the paper does not provide a justification for even a single key algorithm (although a listing is given for simple algorithms). This also led to inaccuracies, only some of them are noted in the remarks below and it is not possible to give an exhaustive list of them (this would essentially mean to provide a justification or correction for these algorithms).

In view of the above and the remarks below, it is required to significantly rework the paper presented by the authors. After that, the article may be re-reviewed in the SEMR journal.

Remarks.

1) The terms n-covering (n-cover) and n'-covering (n'-cover) do not seem to be entirely successful, and the first in a certain sense duplicates the concept "vertex ve-adjacent to an edge" (see [3]) mentioned in the work and earlier found in the literature. The letter (or numeric) designation n is out of place for these concepts. Perhaps the authors can instead of "n-covering (n-cover)" use the term "ve-covering (ve-cover)" and note that the concept "vertex ve-covers an edge e" coincides with the concept of "vertex is ve-adjacent to an edge e" previously considered in [3]. Likewise for "strong (weak) n-covering set". In this case, instead of $wn_0(G)$ and $sn_0(G)$, it is possible to write $w(G)$ and $s(G)$ (or similar), respectively.

2) Also, similarly, instead of "n'-covering set" it is possible to use the term "ve-separating set", which here more reflects the situation of splitting the vertex set V into S and $V \setminus S$ and does not lead to confusion of the concepts.

It is worth noting here that the statement on p.3, proved in the proof of Theorem 2.1 (that S is a strong (weak) n-covering set of G if and only if $V \setminus S$ is a weak (strong) n'-covering set of G) can be formulated as a lemma (in new terms).

3) An incorrect formulation of the concept "strong (weak) n-covering set of G" has been given. Apparently the authors mean that each edge of graph G is n-covered by a suitable vertex from S. It is also worth noting that the set V of all vertices is a strong (weak) n-covering set of any graph. Here we can also explain the case of null graph $\overline{K_p}$ (p-vertex graph without edges), in this case it is assumed that $wn_0(\overline{K_p}) = sn_0(\overline{K_p}) = 0$.

4) After defining "strong (weak) n'-covering set" it is worth explaining the existence of such a set. In the case of a non-trivial graph, for example, by considering vertices of the maximum and minimum ve-degrees. It is also worth discussing the situation of null p-vertex graph $\overline{K_p}$, here it is assumed that $wn'_0(\overline{K_p}) = sn'_0(\overline{K_p}) = p$.

5) The term "n-covering set" (not "strong (weak) n-covering set"!) used in Example 2.1 should be clarified/given, and we need to note the equality of its cardinality to the neighbourhood number (here, seems to be a confusion of the notions).

6) Despite the Listing of algorithms given in the paper, it is required to justify at least all key algorithms. This is not done in any form. In terms of the SEMR log format, the descriptive nature of the algorithms is preferable. Moreover, such an approach will allow, on the one hand, to avoid obvious or widely known algorithms (they should be excluded, for example, the search of vertex neighbourhood $N(v)$ and closed vertex neighbourhood $N[v]$ which are the part of Algorithm 3.1, Algorithm 3.2.1, Algorithm 3.2.2 which is a kind of the simplest sort, etc.), on the other hand, will also allow to clear justify the Algorithms. This approach will also allow to avoid "hidden" incorrectness in the Algorithms and to find errors. For example, I will note two Algorithms:

6.1. In Algorithm 3.1, it is necessary to add a loop that builds neighbourhoods $N(v)$ for all vertices $v \in V$, not just for the selected vertex. This is used when calculating the ve-degree of v and affects the final Time complexity.

6.2. Algorithm 3.2.3. The new term used here (not previously encountered in the paper) a "minimal neighborhood set of a graph G" may have an ambiguous meaning: minimal in the sense of the number $n_0(G)$ or minimal in the sense of a partial order on the set of all neighborhood sets of G. Algorithm 3.2.3 for an arbitrary graph G constructs a minimal neighborhood set S of graph G and reduces to the following: in the remaining

nonnull graph, select a vertex of the largest ve-degree (the first in the array of vertices pre-sorted in descending order of ve-degrees), add this vertex to the set S and delete edges of the graph generated by its neighbourhood (below, for simplicity, we also delete this vertex itself), then we repeat everything until there are edges in the graph. Consider an example: a 6-vertex simple cycle G with sequentially numbered vertices $v_1, v_2, \ldots, v_6$ during its traversal. We have $deg_{ve} v_i = 2$ for any $i = 1, 2, \ldots, 6$. Consider the following variant of sorting these vertices in descending order of ve-degrees (there are no restrictions on sorting in this paper): $v_1, v_4, v_5, v_3, v_2, v_3$, and this sorting will be preserved (for the remaining graph) even after successive removal vertices placed in the constructed set S. As a result, in S we add vertex v_1 at step 1, vertex v_4 at step 2, vertex v_5 at step 3, vertex v_3 at step 4. After that, there will be no edges left in the graph, as there were before. Thus, according to Algorithm 3.2.3, we obtain that $S = \{v_1, v_4, v_5, v_3\}$ is a minimal neighborhood set of the graph G. However, you can choose $S^* = \{v_1, v_5, v_3\} \subsetneq S$, which is obviously a neighborhood set of the graph G. Moreover, $n_0(G) < |S|$. Therefore, in both senses, the minimality for the set S constructed according to Algorithm 3.2.3 is not satisfied.

We also note that when choosing another sorting of the vertices of the cycle G in descending order of ve-degrees, as a result of this algorithm, you can get a set S^*, which will be really minimal.

7). When algorithms are presented, it is naturally required to estimate their complexity. Perhaps the authors will add these estimates of the complexity.

Reviewer