An Iterated Greedy Heuristic for the Minimum-Cardinality Balanced Edge Addition Problem

Ruben Interian, Celso C. Ribeiro

Universidade Federal Fluminense, Institute of Computing, Niterói, RJ 24210-346, Brazil.
rinterian@ic.uff.br, celso@ic.uff.br

Abstract

The Minimum-Cardinality Balanced Edge Addition Problem (MinCBEAP) appears in the context of polarized networks as a strategy to reduce polarization by external interventions using the minimum number of edges. We show that every instance of MinCBEAP can be reduced to an instance of the Minimum-Cardinality-Bounded-Eccentricity Edge Addition Problem (MCBE). A restarted iterated greedy heuristic is developed for solving MinCBEAP via the transformed MCBE. Preliminary computational results are reported.

1 Problem statement

According to the Oxford Dictionaries, polarization is the division into sharply contrasting groups or sets of opinions or beliefs [1]. Academic articles, newspapers, and the media in general constantly report the growth of fake news, misinformation spreading, and polarization. These phenomena are closely interrelated with each other. Fake news spread faster in polarized networks or groups [6]. At the same time, fake and tendentious news can accentuate polarization within already existing echo chambers in the social networks.

Interian and Ribeiro [3] showed that many case studies real-world networks are extremely polarized. A polarized network is divided into two or more strongly connected groups, with few edges between vertices belonging to different groups. Most of the time, only same-group vertices communicate to each other and most of the information that a vertex can receive comes from inside the same group to which it belongs.

In order to reduce polarization, networks can be treated by external interventions consisting of the addition or the removal of vertices and edges. In this work, we address a new optimization problem that treats the issue of polarization reduction by edge additions. Given a graph $G = (V, E)$ and a subset of (polarized) vertices $A \subseteq V$, we seek a minimum-cardinality set of edges $E' \subseteq V \times V \setminus E$ to be added to $G$ such that all vertices that are not in $A$ can be reached from any vertex in $A$ by paths with at most $D$ edges, where $D$ is a problem parameter. The decision version of our optimization problem can be cast as:

**Minimum-Cardinality Balanced Edge Addition Problem (MinCBEAP):**

Instance: Graph $G = (V, E)$, subset $A \subseteq V$, integer $D$, integer $L$.

Goal: Is there a set $E' \subseteq (V \times V \setminus E$ with cardinality $|E'| \leq L$ such that $d_{G'}(v, V \setminus A) \leq D, \forall v \in A$?

In the above formulation, $d_{G'}(v, V \setminus A)$ denotes the number of edges in a shortest path from $v \in A$ to the closest vertex in $V \setminus A$ in $G' = (V, E \cup E')$.

Given a graph $G = (V, E)$, the eccentricity $\epsilon(v)$ of a vertex $v \in V$ is the longest of the shortest paths in $G$ from $v$ to all other vertices in $V \setminus \{v\}$, see [2]. Any instance of MinCBEAP can be reduced to an instance of the Minimum-Cardinality-Bounded-Eccentricity Edge Addition Problem (MCBE):

**Minimum-cardinality-bounded-eccentricity edge addition problem (MCBE):**

Instance: Graph $G = (V, E)$, source vertex $s \in V$, integer $B$, integer $p$.

Question: Is there a supergraph $G' = (V, E \cup E')$ of $G$ with $E' \subseteq (V \times V) \setminus E$ such that $|E'| \leq p$ and $\epsilon_{G'}(s) \leq B$?
Interian and Ribeiro [4] showed that MCBE is NP-complete. They also showed that MinCBEAP is NP-complete for $D$ greater than or equal to 2, using a polynomial transformation from MCBE.

We propose in the following an efficient iterated greedy heuristic for approximately solving MinCBEAP using a transformation to MCBE. This article is organized as follows. In Section 2, we describe the polynomial transformation from an instance of MinCBEAP to one of MCBE and the iterated greedy heuristic. In Section 3, some preliminary computational results are presented.

## 2 Iterated greedy heuristic

We propose the following transformation of an instance of MinCBEAP into an instance of MCBE. Consider an instance of the optimization version of MinCBEAP defined by a graph $G = (V, E)$, a subset $A \subseteq V$, and an integer $D$. To build the corresponding optimization instance of MCBE, consider a graph $\bar{G} = (\bar{V}, \bar{E})$ and an integer $B = D$. Let $\bar{V} = A \cup \{s\}$, where $s$ is the source vertex representing the set $V \setminus A$, in such a way that if there is an edge $(u, v) \in E$ with $u \in A$ and $v \in V \setminus A$, then there will be an edge $(u, s) \in \bar{E}$.

The MCBE instance is solved after the problem transformation. Let $S_H \subseteq (\bar{V} \times \bar{V}) \setminus \bar{E}$ be the optimal solution of MCBE for the instance defined by $\bar{G} = (\bar{V}, \bar{E})$. This solution is transformed to a solution $S_G \subseteq (V \times V) \setminus E$ of the original MinCBEAP instance as follows. Consider any edge $(u, v) \in S_H$. If both $u, v \in A$, then the edge $(u, v)$ also belongs to $S_G$. Otherwise, in case the source $s \notin A$ coincides e.g. with extremity $u$ of edge $(u, v)$, then edge $(w, v)$ is placed in $S_G$, where $w$ is a randomly generated vertex from $V \setminus A$. Following this strategy, we obtain a solution $S_G$ that solves MinCBEAP satisfying $|S_H| = |S_G|$.

The heuristic proposed in this work is based on the following lemma (the proof is omitted here due to space limitations):

**Lemma 2.1.** There is a solution $S^* \subseteq (V \times V) \setminus E$ for MCBE such that all edges in $S^*$ are incident to the source vertex $s$.

Therefore, the set of candidate edges to be inserted in the solution can be restricted to those incident to vertex $s$. The problem is then reduced to determining a subset $X^* \subseteq V \setminus \{s\}$ of vertices such that for every vertex $v \in X^*$ there is an edge $\{v, s\} \in S^*$. In the algorithm, the solution $S^*$ will be represented by the corresponding subset $X^*$ of vertices, where $|S^*| = |X^*|$.

The cost of a solution to MCBE is given by the cardinality $|X|$ of the set $X$. It is well known that local search procedures often do not perform satisfactorily in the context of optimization problems without weights, in which the objective function is related to the cardinality of the solution, due to the lack of gradient information and to the exponential size of neighborhoods based on cardinality improving. In order to overcome this barrier, we propose in the following an iterated greedy heuristic [7] for MCBE, which is solely based on the repeated, successive application of destruction and reconstruction procedures, without appealing to local search.

In summary, iterated greedy starts from a greedy or a semi-greedy candidate solution, and generates a sequence of solutions using two main phases: destruction and reconstruction. During the destruction phase, some edges are removed from the current solution. The reconstruction procedure then applies a greedy constructive heuristic to reconstruct a complete candidate solution. If the cardinality of the newly constructed solution is less than the cardinality of the incumbent solution, then the latter is updated. The heuristic iterates over these steps until some stopping criterion is met [7].

One important building block of the iterated greedy heuristic is the semi-greedy (or greedy randomized) algorithm. There is a number $p(v)$ of non-solved neighbors associated with each vertex $v \in V$, that is, the number of neighbors of $v$ that are still at a distance greater than $B$ from the source vertex $s$. At each step, this algorithm adds one random vertex from a restricted candidate list (RCL) to the initially empty solution set $X$. The RCL is formed by all vertices $w \in V \setminus \{s\}$ that satisfy the condition $(1 - \alpha)p_{\max} \leq p(w) \leq p_{\max}$, where $p_{\max} = \max\{p(v) : v \in V \setminus \{s\}\}$ and $\alpha$ is the greediness
parameter of the semi-greedy algorithm. We recall that $\alpha = 0$ corresponds to a purely greedy criterion, while $\alpha = 1$ corresponds to a completely random selection of the next vertex from $V \setminus \{s\}$.

Algorithm 1 shows the pseudo-code of the proposed iterated greedy heuristic. In line 1, the best solution $X^*$ and its cardinality $f^*$ are initialized. The while loop in lines 2–16 performs a new iteration while a stopping criterion is not met. The semi-greedy algorithm is applied in line 3 to build a solution $X$ using the greediness parameter $\alpha$. This solution is improved by performing a sequence of destruction-reconstruction phases in lines 4–12, until $k_{\text{max}}$ iterations without any improvement are executed. The destruction phase is applied in line 7, by randomly removing a fraction $\beta$ of the vertices in $X$. Next, the resulting solution is reconstructed by a greedy deterministic algorithm in line 8, which corresponds to the semi-greedy algorithm with $\alpha = 0$. If the cardinality of the newly reconstructed solution $X'$ is less than the cardinality of the incumbent $X$, then the latter is updated in line 10. If the solution obtained after the sequence of destruction-reconstruction iterations is better than the best known solution $X^*$, then the best solution $X^*$ and its cardinality $f^*$ are updated in line 14. In line 17, the best solution found $X^*$ is returned.

Algorithm 1 Restarted Iterated Greedy for MCBE

<table>
<thead>
<tr>
<th>Parameters: $\alpha, \beta, k_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: $G = (V, E)$</td>
</tr>
<tr>
<td>Output: $X^*$</td>
</tr>
</tbody>
</table>

1: $X^* \leftarrow V$; $f^* \leftarrow |V|$; 
2: while stopping condition is not met do 
3: $X \leftarrow \text{SemiGreedy}(G, \alpha)$; 
4: $i \leftarrow 0$; 
5: while $i \leq k_{\text{max}}$ do 
6: $i \leftarrow i + 1$; 
7: $X' \leftarrow \text{PartialDestruction}(G, X, \beta)$; 
8: $X' \leftarrow \text{GreedyReconstruction}(G, X')$; 
9: if $|X'| < |X|$ then 
10: $X \leftarrow X'$; $i \leftarrow 0$; 
11: end if; 
12: end while; 
13: if $|X| < f^*$ then 
14: $X^* \leftarrow X$; $f^* \leftarrow |X|$; 
15: end if; 
16: end while; 
17: return $X^*$; 

In the above algorithm, an external loop was added to the iterated greedy heuristic originally described by Ruiz and Stützle [7]. Instead of starting with a completely greedy solution, we start with a semi-greedy solution and the iterated greedy construction can be embedded in a multi-start procedure. This hybrid variant was named Restarted Iterated Greedy (RIG) by Pinto et al. [5].

3 Preliminary numerical results

We generated 13 random graph instances with different number of vertices. The number of edges in each instance was fixed as 8 times the number of vertices. The maximum length of the connecting paths was fixed as $D = 2$. The greediness parameter of the restarted iterated greedy was set at $\alpha = 0.1$, the fraction of the solution to be destructed at $\beta = 50\%$, the number of iterations without improvement at $k_{\text{max}} = 100$, and the time limit of 3600 seconds as the stopping criterion for the heuristic.

Table 1 shows preliminary results for the random instances, including the number of vertices $|X^*| = |S^*|$ in the best solution found and the running time in seconds. We are currently deve-
ping a mixed integer programming (MIP) approach for exactly solving small- and medium-size instance of MCBE. Preliminary results obtained by the MIP approach make it possible to identify the optimality of the solutions found by the restarted iterated greedy heuristic for the instances with up to 1000 vertices.

<table>
<thead>
<tr>
<th>Instance</th>
<th></th>
<th>A</th>
<th></th>
<th>X*</th>
<th>Time (s)</th>
<th>Solved?</th>
</tr>
</thead>
<tbody>
<tr>
<td>inst_100v_8x</td>
<td>100</td>
<td>3</td>
<td>0.32</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_200v_8x</td>
<td>200</td>
<td>7</td>
<td>0.64</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_500v_8x</td>
<td>500</td>
<td>20</td>
<td>0.19</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_1000v_8x</td>
<td>1000</td>
<td>38</td>
<td>246.08</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_2000v_8x</td>
<td>2000</td>
<td>81</td>
<td>3610.24</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_3000v_8x</td>
<td>3000</td>
<td>121</td>
<td>3609.47</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_4000v_8x</td>
<td>4000</td>
<td>160</td>
<td>3609.60</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_5000v_8x</td>
<td>5000</td>
<td>209</td>
<td>3606.40</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_6000v_8x</td>
<td>6000</td>
<td>253</td>
<td>3614.21</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_7000v_8x</td>
<td>7000</td>
<td>292</td>
<td>3621.25</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_8000v_8x</td>
<td>8000</td>
<td>331</td>
<td>3617.79</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_9000v_8x</td>
<td>9000</td>
<td>392</td>
<td>3638.78</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inst_10000v_8x</td>
<td>10000</td>
<td>426</td>
<td>3644.16</td>
<td>(no)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Results for the restarted iterated greedy heuristic.

References


