Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)

doi:10.3233/FAIA220309

Approximate and Optimal Solutions for the Bipartite Polarization Problem

Teresa ALSINET, Josep ARGELICH [1], Ramón BÉJAR and Santi MARTÍNEZ
INSPIRES Research Center – University of Lleida
Jaume II, 69 – 25001 Lleida, SPAIN

Abstract. In a recent work we introduced a problem about finding the highest polarized bipartition on a weighted and labeled graph that represents a debate developed trough some social network, where nodes represent user's opinions and edges agreement or disagreement between users. Finding this target bipartition is an optimization problem that can be seen as a generalization of the maxcut problem, so we first introduced a basic local search algorithm to find approximate solutions of the problem. In this paper we go one step further, and we present an exact algorithm for finding the optimal solution, based on an integer programming formulation, and compare the performance of a new variant of our local search algorithm with the exact algorithm. Our results show that at least on real instances of the problem, obtained from Reddit debates, the approximate solutions obtained are almost always identical to the optimal solutions.

Keywords. Social Networks, Polarization, Combinatorial Optimization.

1. Introduction

The emergence of polarization in discussions on social networks, and the responsibility of companies in this problem, is a topic that is causing a significant interest among society. For example, Facebook has launched some initiatives to try to mitigate the factors that may be helping the spread of divisive content [5], even if this kind of content may be the one that produces the maximum attention of their users, so being also the one producing maximum economic benefit.

Because each social network company can have its own personal interest regarding when to control this kind of behaviour, one fundamental aspect is to define more transparent ways to monitor such possible non-desirable behaviours so that we can decide to act only in situations where we can deduce that polarization is taking place, and to a certain level of severity, because there is some objective value we can measure for this. So, in a previous work, we defined one such measure

[1] Correspondence to: J. Argelich. INSPIRES Research Centre, University of Lleida. C/Jaume II, 69. Lleida, Spain. Tel.: +34 973702734; E-mail: josep.argelich@udl.cat.

and presented an initial approximate algorithm to compute this measure from a discussion in the Reddit social network [1].

In this paper, we present an exact algorithm for computing this measure and a new variant of the approximate algorithm. The exact algorithm is based on an integer programming formulation, inspired by existing good formulations for the maxcut problem, as our problem can be seen as a generalization of the maxcut problem. Our preliminary experimental results over a set of Reddit debates show that the solutions obtained with the approximate algorithms are almost identical to the optimal solutions found by the exact algorithm, although computed with much less time.

The structure of the rest of the paper is as follows. In Section 2, we present both the representation model of Reddit debates and the measure to quantify the polarization in a debate, studied and developed in [1]. In Section 3, we introduce an exact algorithm for finding the optimal solution, based on an integer programming formulation. Finally, in Section 4, we perform an empirical evaluation to compare the performance of our solving approaches when computing the bipartition of two different sets of Reddit discussions.

2. Problem Definition

Following [1], a *Reddit debate* Γ on a root comment r is a non-empty set of Reddit comments, that were originated as successive answers to the root comment r that contains a link to some news. To represent debates on Reddit, we use a two-sided debate tree model, where nodes are labelled with a binary value that denotes whether the comment is in agreement (1) or in disagreement (-1) with the root comment. [1]

Definition 1 (Two-Sided Debate Tree) *Let Γ be a Reddit debate on a root comment r. A* Two-Sided Debate Tree (SDebT) *for Γ is a tuple $\mathcal{T}_S = \langle C, r, E, W, S\rangle$ defined as follows:*

- *For every comment c_i in Γ, there is a node c_i in C.*
- *Node $r \in C$ is the root node of $\mathcal{T}$.*
- *If a comment $c_1 \in C$ answers another comment $c_2 \in C$, there is a directed edge (c_1, c_2) in E.*
- *W is a labelling function of answers (edges) $W : E \to [-2, 2]$, where the value assigned to an edge $(c_1, c_2) \in E$ denotes the sentiment of the answer c_1 with respect to c_2, from highly negative (-2) to highly positive (2).*
- *S is a labelling function of comments (nodes) $S : C \to \{-1, 1\}$, where the value assigned to a node $c_i \in C$ denotes whether the comment c_i is in agreement (1) or in disagreement (-1) with the root comment r and it is defined as follows:*

  - $S(r) = 1$ *and*

[1] Note that this definition can be applied to other similar social networks.

- *For all node $c_1 \neq r$ in C, $S(c_1) = 1$ if for some node $c_2 \in C$, $(c_1, c_2) \in E$ and either $S(c_2) = 1$ and $W(c_1, c_2) > 0$, or $S(c_2) = -1$ and $W(c_1, c_2) \leq 0$; otherwise, $S(c_1) = -1$.*

Only the nodes and edges obtained by applying this process belong to C and E, respectively.

Given a Reddit debate on a (root) comment, we make its corresponding SDebT using the Python Reddit API Wrapper (PRAW, available at `https://github.com/praw-dev/praw`) to download its set of comments, and then we evaluate the sentiment for each edge $(c_1, c_2) \in E$ using the sentiment analysis software of [4] using the text of the comment c_1.

Our goal in this work is to introduce and investigate a suitable algorithm that allows us to analyse and study the polarization of users in debates. To this end, we group comments of a debate by user, and we consider that the relationship between two users is defined from the agreement and disagreement relationships between the individual comments of these two users.

Next, we present our formalization of a User Debate Graph based on a Two-Sided Debate Tree, where now we aggregate all the comments of a same user into a single node that represents the user's opinion.

Definition 2 (User Debate Graph) *Let Γ be a Reddit debate on a root comment r with users' identifiers $U = \{u_1, \ldots, u_m\}$ and let $\mathcal{T}_S = \langle C, r, E, W, S\rangle$ be a* SDebT *for Γ. A* User Debate Graph (UDebG) *for $\mathcal{T}_S$ is a tuple $\mathcal{G} = \langle \mathcal{C}, \mathcal{E}, \mathcal{S}, \mathcal{W}\rangle$, where:*

- *$\mathcal{C}$ is the set of nodes of $\mathcal{G}$ defined as the set of users' opinions $\{C_1, \ldots, C_m\}$; i.e., $\mathcal{C} = \{C_1, \ldots, C_m\}$ with $C_i = \{c \in \Gamma \mid c \neq r \text{ and } user(c) = u_i\}$, for all users $u_i \in U$.*
- *$\mathcal{E} \subseteq \mathcal{C} \times \mathcal{C}$ is the set of edges of $\mathcal{G}$ defined as the set of interactions between different users in the debate; i.e., there is an edge $(C_i, C_j) \in \mathcal{E}$, with $C_i, C_j \in \mathcal{C}$ and $i \neq j$, if and only if for some $(c_1, c_2) \in E$ we have that $c_1 \in C_i$ and $c_2 \in C_j$.*
- *$\mathcal{S}$ is an opinion weighting scheme for $\mathcal{C}$ that expresses the side of users in the debate based on the side of their comments. We define $\mathcal{S}$ as the mapping $\mathcal{S} : \mathcal{C} \to [-1, 1]$ that assigns to every node $C_i \in \mathcal{C}$ the value*

$$\mathcal{S}(C_i) = \frac{\sum_{c \in C_i} S(c_i)}{|C_i|}$$

  in the real interval $[-1, 1]$ that expresses the side of the user u_i with respect to the root comment, from strictly disagreement (-1) to strictly agreement (1), going through undecided opinions (0).
- *$\mathcal{W}$ is an interaction weighting scheme for $\mathcal{E}$ that expresses both the ratio of positive interactions between the users' opinions and the overall sentiment between users by combining the individual sentiment values assigned to the responses between their comments.*
  We define $\mathcal{W}$ as the mapping $\mathcal{W} : \mathcal{E} \to ([0, 1] \times [-2, 2])$ that assigns to every edge $(C_i, C_j) \in \mathcal{E}$ the pair of values $(p, w) \in ([0, 1] \times [-2, 2])$ defined as follows:

$$p = \frac{|(c_1, c_2) \in E \cap (C_i \times C_j) \ with \ W(c_1, c_2) > 0|}{|(c_1, c_2) \in E \cap (C_i \times C_j)|} \quad and$$

$$w = \sum_{\{(c_1,c_2) \in E \cap (C_i \times C_j)\}} W(c_1, c_2)/|\{(c_1, c_2) \in E \cap (C_i \times C_j)\}|$$

where p expresses the ratio of positive answers from the user u_i to the user u_j in the debate, and w expresses the overall sentiment of the user u_i regarding the comments of the user u_j, from highly negative (-2) to highly positive (2).

Only the nodes and edges obtained by applying this process belong to $\mathcal{C}$ and $\mathcal{E}$, respectively.

Given a User Debate Graph $\mathcal{G} = \langle \mathcal{C}, \mathcal{E}, \mathcal{S}, \mathcal{W} \rangle$, in [1] we introduced a model to measure the level of polarization in the debate between its users. We identified two characteristics that a polarization measure should capture. First, a polarized debate should contain a bipartition of $\mathcal{C}$ into two sets (L, R) such that the set L contains mainly users in disagreement, the set R contains mainly users in agreement, and both sets should be similar in size. The second ingredient is the sentiment between users of L and R. A polarized discussion should contain most of the negative interactions between users of L and users of R, whereas the positive interactions, if any, should be mainly within the users of L and within the users of R.

To capture these two characteristics with a single value, we defined two different measures and their combination in a final one, referred to as *the bipartite polarization level*.

Definition 3 (Bipartite Polarization) *Given a User Debate Graph $\mathcal{G} = \langle \mathcal{C}, \mathcal{E}, \mathcal{S}, \mathcal{W} \rangle$ and a bipartition (L, R) of $\mathcal{C}$, we define:*

- *The level of consistency and balance of (L, R) is a real value in $[0, 0.25]$ defined as follows:*

$$\mathrm{SC}(L, R, \mathcal{G}) = \mathrm{LC}(L, \mathcal{G}) \cdot \mathrm{RC}(R, \mathcal{G})$$

with:

$$\mathrm{LC}(L, \mathcal{G}) = \frac{\sum_{\substack{C_i \in L, \\ \mathcal{S}(C_i) \leq 0}} -\mathcal{S}(C_i)}{|\mathcal{C}|}$$

and

$$\mathrm{RC}(R, \mathcal{G}) = \frac{\sum_{\substack{C_i \in R, \\ \mathcal{S}(C_i) > 0}} \mathcal{S}(C_i)}{|\mathcal{C}|}.$$

- *The sentiment of the interactions between users of different sides is a real value in* $[0, 4]$ *defined as follows:*

$$\text{SWeight}(L, R, \mathcal{G}) = \frac{\sum_{\substack{(i,j)\in\mathcal{E}\cap \\ ((L\times R)\cup(R\times L))}} -c(p(i,j)) \cdot w(i,j)}{|\mathcal{E}|} + 2,$$

with

$$c(p(i,j)) = 2(p(i,j) - 0.5)^2 + 1/2,$$

and where $p(i,j)$ *and* $w(i,j)$ *denote the values of* p *and* w*, respectively, in* $\mathcal{W}(i,j) = (p, w)$*.*

Then, the Bipartite Polarization of $\mathcal{G}$ *on a bipartition* (L, R) *is the value in the real interval* $[0, 1]$ *defined as follows:*

$$BipPol(L, R, \mathcal{G}) = SC(L, R, \mathcal{G}) \cdot SWeight(L, R, \mathcal{G}).$$

Finally, the Bipartite Polarization of $\mathcal{G}$ *is the maximum value of* $BipPol(L, R, \mathcal{G})$ *among all the possible bipartitions* (L, R)*.*

3. Algorithms

We can find the Bipartite Polarization of a User Debate Graph $\mathcal{G} = \langle \mathcal{C}, \mathcal{E}, \mathcal{S}, \mathcal{W} \rangle$ by solving the following integer nonlinear programming formulation (MINLP) of it, where each node C_i from $\mathcal{C}$ is associated with an integer variable x_i, such that $x_i = -1$ represents that C_i is in the L partition and $x_i = +1$ that C_i is in R:

$$max_x \left(\frac{1}{|\mathcal{C}|^2} \sum_{\substack{(x_i,x_j)\ with \\ S(C_i)\leq 0, S(C_j)>0}} -S(C_i)S(C_j)(1 - x_i)(1 + x_j)/4.0 \right) *$$
$$\left(2 + \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} -c(p(i,j)) \cdot w(i,j) * (1 - x_i * x_j)/2.0 \right)$$
$$\text{subject to: } x_i^2 = 1 \quad \forall C_i \in \mathcal{C}$$

Observe that the first sumatory in the objective function represents the term $SC(L, R, \mathcal{G})$ and the second one the term $SWeight(L, R, \mathcal{G})$.

Then, we use the branch-and-bound solver [6] of the SCIP Optimization suite (version 8.0) [3] to optimally solve problem instances with this MINLP formulation.

We can also use approximate algorithms based on local search, like the one we introduced in our previous work [1]. In that algorithm, the search ends as soon as the algorithm reaches a local maximum.

Now in this work, we present a slight variant where the algorithm performs diversification steps (non-improving steps) to try to escape from local maximum and be able to find better solutions later on. The pseudocode of this new variant, that is inspired by a local search for the maxcut problem [2], is shown on Algorithm 1. The search starts with some initial pseudo-random partition [1] (line 1), and then it initiates the search for a local maximum of the objective value, selecting at every step a node v that represents the steepest ascent hill climbing step, if such a move exists (line 6). That is, a node v that when swapped between L and R is the one that improves more the objective value. Each time the algorithm reaches a local maximum (line 9) it updates the best solution found so far (if necessary) and then it selects randomly some vertices to be swapped (line 12), where the probability that a node is selected is controlled with the parameter ω. After the diversification steps, the whole process is repeated, up to *max_restarts* times.

Algorithm 1 Finding a local-optimal solution for the bipartite polarization of $\mathcal{G}= \langle \mathcal{C}, \mathcal{E}, \mathcal{S}, \mathcal{W} \rangle$, using diversification.

Input: $\mathcal{G}= \langle \mathcal{C}, \mathcal{E}, \mathcal{S}, \mathcal{W} \rangle$
Output: a bipartition (L, R) of $\mathcal{G}$ with high bipartite polarization

1: $(L, R) := getInitBPart(\mathcal{G})$ ▷ Get Initial bipartition
2: $bestBipPol := getBipPol(L, R)$ ▷ Save as best bipartite polarization value...
3: $(L', R') := (L, R)$ ▷ and save corresponing bipartition
4: **for all** *max_restarts* **do**
5:     **do**
6:         **if** $\exists v \in SAHC(L, R, \mathcal{G})$ **then** ▷ Steepest Ascent Hill Climbing
7:             $swapNode(L, R, v)$ ▷ Change the set of node v
8:     **while** $(\exists v \in SAHC(L, R, \mathcal{G})$ **and not** $max\_steps)$
9:     **if** *getBipPol*(L, R) > bestBipPol **then** ▷ Save new best solution
10:         $bestBipPol := getBipPol(L, R)$
11:         $(L', R') := (L, R)$
12:     $diversify(L, R, \omega)$ ▷ ω is the probability of diversification for a node
13: **return** (L', R')

4. Experimental Results

We present here an small empirical evaluation of the performance of our three solving approaches (exact algorithm, previous local search algorithm [1] and improved local search algorithm) when computing the bipartite polarization of two different sets of Reddit discussions. We compare both the running time and how close are the approximate solutions obtained by the local search algorithms compared with the optimal solution. The results are shown on Table 1, where for

	Stats	$\|\mathcal{C}\|$	SCIP time	SCIP nodes	SCIP BipPol	LS1 time	LS1 ratio	LS2 time	LS2 ratio
DS1	min	17	0.11	0	0.166071	0.01	0.934859	0.01	0.999998
	median	42	1.19	1	0.478105	0.01	1	0.06	1
	mean	41	6.48	1.75	0.470922	0.01	0.998220	0.08	0.999999
	max	86	104.4	38	0.610654	0.02	1	0.35	1
DS2	min	25	0.2	0	0.443185	0.01	0.999592	0.02	1
	median	50	3.94	1	0.540176	0.01	1	0.11	1
	mean	53	24.59	1.64	0.543161	0.01	0.999988	0.15	1
	max	102	337.51	35	0.674507	0.04	1	0.65	1

Table 1. Experimental results with algorithms for computing bipartite polarization.

the exact algorithm (SCIP) we show statistics (min, median, mean and max) for its solving time over the instances of the first dataset (DS1) and over the second dataset (DS2), as well as the same statistics but for the number of nodes of the branch-and-bound search tree and the bipartite polarization value of the instances (BipPol). We also show the same statistics for the number of vertices (users of the user debate graph) in the two different data sets, so we can observe that the second dataset contains slightly bigger instances than the first one. We also show the results for the two local search approaches, the previous one (LS1) and the new presented in this paper (LS2). The values shown are their execution times and the ratio of the solution obtained by the local search algorithm to the optimal solution found by SCIP.

As the results indicate, the local search algorithms almost always find the optimal solution, but with a much smaller running time. However, it is interesting to note that the solving time, and number of nodes of the search tree, is in general quite small for the SCIP solver, as the median time is in both datasets around 2 seconds, and the median number of nodes is one. The fact that SCIP is able to solve most of the instances with no branching at all, is due to the fact that in those instances it is able to solve them only during the "presolving" phase, where it uses different simplification techniques, although the mean and max values show that there are instances where presolving is not enough. It is not clear how these results will be for bigger instances, but for the instances tested here it is clear that the local search approaches seem to be enough to solve the instances. Regarding the differences between LS1 and LS2, we observe that already LS1 is able to find almost always the optimal solution, but LS2 seems to obtain a better ratio (almost always equal to 1) in all the cases.

5. Conclusions

In this paper we present several algorithms to solve the Bipartite Polarization Problem, that can be seen as a generalization of the maxcut problem. To this end, we first introduce two variants of a basic local search algorithm to find approximate solutions. Next, we also develop a complete algorithm based on the integer nonlinear programming formulation. Both approaches show very good performance, being the incomplete one the fastest and the complete one the more accurate. Also, as we can see in the results, the incomplete approach is pretty accurate as the solutions are always very close to the optimal solution.

Further experimental results will be needed to understand their scaling behavior as the size of the instances increases. As further work, we also plan to explore other solving techniques, like the ones based on Goemans-Williamson's Semidefinite Positive relaxation, and study what features make the instances easier to be solved with the local search approach.

Acknowledgments This work was partially funded by Spanish Project PID2019-111544GB-C22 (MINECO / FEDER), by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreements 723596, 768824, 764025 and 814945, and by 2017 SGR 1537.

References

[1] T. Alsinet, J. Argelich, R. Béjar, and S. Martínez. Measuring polarization in online debates. *Applied Sciences*, 11(24), 2021.

[2] U. Benlic and J. Hao. Breakout local search for the max-cutproblem. *Eng. Appl. Artif. Intell.*, 26(3):1162–1173, 2013.

[3] K. Bestuzheva, M. Besançon, W.-K. Chen, A. Chmiela, T. Donkiewicz, J. van Doornmalen, L. Eifler, O. Gaul, G. Gamrath, A. Gleixner, L. Gottwald, C. Graczyk, K. Halbig, A. Hoen, C. Hojny, R. van der Hulst, T. Koch, M. Lübbecke, S. J. Maher, F. Matter, E. Mühmer, B. Müller, M. E. Pfetsch, D. Rehfeldt, S. Schlein, F. Schlösser, F. Serrano, Y. Shinano, B. Sofranac, M. Turner, S. Vigerske, F. Wegscheider, P. Wellner, D. Weninger, and J. Witzig. The scip optimization suite 8.0. Technical Report 21-41, ZIB, Takustr. 7, 14195 Berlin, 2021.

[4] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[5] G. Rosen. Facebook: Investments to fight polarization. `https://about.fb.com/news/2020/05/investments-to-fight-polarization/`, 2020. date: 2020-05-27.

[6] S. Vigerske and A. Gleixner. Scip: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, 33(3):563–593, 2018.