

An Efficient Riemannian Gradient based Algorithm for Max-Cut Problems

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

28-01-2021 / 03-02-2021

CITATION

Mohades, Mohamad Mahdi; Kahaei, Mohammad Hossein (2021): An Efficient Riemannian Gradient based Algorithm for Max-Cut Problems. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.13660991.v1>

DOI

[10.36227/techrxiv.13660991.v1](https://doi.org/10.36227/techrxiv.13660991.v1)

An Efficient Riemannian Gradient based Algorithm for Max-Cut Problems

Mohamad Mahdi Mohades and Mohammad Hossein Kahaei

Abstract—The max-cut problem addresses the problem of finding a cut for a graph that splits the graph into two subsets of vertices so that the number of edges between these two subsets is as large as possible. However, this problem is NP-Hard, which may be solved by suboptimal algorithms. In this paper, we propose a fast and accurate Riemannian optimization algorithm for solving the max-cut problem. To do so, we develop a gradient descent algorithm and prove its convergence. Our simulation results show that the proposed method is extremely efficient on some already-investigated graphs. Specifically, our method is on average 50 times faster than the best well-known techniques with slightly losing the performance, which is on average 0.9729 of the max-cut value of the others.

Index Terms—Max-cut problem, manifold optimization, polynomial time, NP-hard.

I. INTRODUCTION

THE max-cut problem has diverse applications in very-large-scaled-integrated (VLSI) circuit design, statistical physics [1], social networks [2], semisupervised learning [3], and biology [4]. This problem can be defined by considering an undirected graph $G = (V, E, \mathbf{A})$, where V denotes the set of vertices, E is the set of edges connecting the vertices in V , and \mathbf{A} denotes the adjacency matrix with nonnegative elements. Without loss of generality and for simplicity of presentation, we address the simple graphs which have finitely many vertices and with no loops or multiple edges. A cut for the graph G splits the vertices into two subsets S and \bar{S} , where $S \cap \bar{S} = \emptyset$ and $S \cup \bar{S} = V$. Also, the corresponding cut value is calculated as

$$w(S, \bar{S}) = \sum_{\{i,j\} \in E(S, \bar{S})} a_{ij}, \quad (1)$$

where $E(S, \bar{S})$ is the collection of the edges with one endpoint in S and the other in \bar{S} , and a_{ij} is the (i, j) th element of the adjacency matrix \mathbf{A} . Then, the max-cut problem addresses finding a cut with the maximum cut value.

A naive exact solution for the max-cut problem is to calculate the cut value of all 2^n possible cuts and comparing their corresponding cut values for finding the max-cut, where n is the number of the graph's vertices. This approach demands $\mathcal{O}(m2^n)$ computations, where m is the number of the corresponding edges, which is in general unaffordable. This exact algorithm can be contemplated as a complete traversal of a binary decision tree in which each branch is obtained by either allocating a vertex to a given set or not. Then, to reduce the computational complexity of this algorithm, the upper and lower bounds of the max-cut are used to limit the traversal of

the decision tree. Such bounds can be obtained by utilizing heuristic functions [5].

To cope with the NP-hardness of the max-cut problem, polynomial-time ρ -approximation algorithms may be utilized. A ρ -approximation algorithm finds a solution that is at least ρ times the optimal value, where the constant ρ is called the performance guarantee of the algorithm. Also, ρ -approximation for randomized algorithms means that the solution is on average ρ times the optimal value. In [6], a 0.5-approximation algorithm has been presented for the max-cut problem. This algorithm can be simply explained as the random assignment of the set of vertices of the graph to two subsets based on the uniform bipolar distribution. Following this method, some other approximation algorithms have been presented with slight improvement; such as [7] with a performance guarantee of $0.5 + 0.5/\Delta$, where Δ is the maximum degree. In [8], the authors develop a polynomial-time algorithm based on the SDP and randomized rounding approach which achieves 0.878-approximation of the max-cut problem. Then, the max-cut problem is formulated as

$$\begin{aligned} \max_{\mathbf{y}} \quad & \frac{1}{2} \sum_{i < j} a_{ij} (1 - y_i y_j) \\ \text{s.t.} \quad & y_i \in \{-1, 1\}, \quad i = 1, \dots, n, \end{aligned} \quad (2)$$

where $\mathbf{y} = (y_1, \dots, y_n)$ is a cut for G showing V_i belongs to S for $y_i = 1$ and to \bar{S} for $y_i = -1$. However, since (2) is NP-complete, the following semidefinite relaxation is used,

$$\begin{aligned} \max_{\mathbf{x}_1, \dots, \mathbf{x}_n} \quad & \frac{1}{2} \sum_{i < j} a_{ij} (1 - \langle \mathbf{x}_i, \mathbf{x}_j \rangle) \\ \text{s.t.} \quad & \|\mathbf{x}_i\|_2 = 1, \quad i = 1, \dots, n. \end{aligned} \quad (3)$$

Then, a randomized rounding (RR) approach is used to map the optimal solution $(\mathbf{x}_1^*, \dots, \mathbf{x}_n^*)$ resulted from (3) to a cut. To perform the RR, first a random vector \mathbf{r} , is chosen with uniform distribution on the unit sphere S^{n-1} . Then, the label y_i is estimated as $\text{sign}(\langle \mathbf{x}_i^*, \mathbf{r} \rangle)$. In [9], it is proved that the method of [8] results in the best polynomial time ρ -approximation algorithm for a maximum cut, provided that the unique games conjecture [10] holds. Also, Håstad proved that approximating the max-cut problem with a performance guarantee better than $16/17 \simeq 0.941176$ is NP-hard [11].

Apart from the exact and the ρ -approximation algorithms for solving the max-cut problem, a set of inexact methods have been also addressed with no performance guarantee. In [12], the authors use the solution of [8] to initialize a local search to improve the max-cut solution, and also investigate some heuristic methods like Metropolis and simulated annealing (SA). The results show that by growing a graph, the SA performs effectively in terms of quality. In [13], the quality

The authors are with the School of Electrical Engineering, Iran University of Science & Technology, Tehran 16846-13114, Iran (e-mail: mohamad_mohades@elec.iust.ac.ir; kahaei@iust.ac.ir).

of the SDP method of [8], random strategy [6], genetic algorithms, combinatorial algorithm [7], and a divide and conquer (D&C) strategy are investigated. The results show that genetic algorithms give a quite similar solution to that of the random strategy [6], however, with much more running time. Moreover, it is shown that although the combinatorial method performs worse than the SDP, it is much faster. It is also discussed that the D&C strategy can present a good compromise between the quality and running time.

In [14], a heuristic rank-2 relaxation is developed and the CirCut algorithm is introduced for solving the max-cut problem. In [15], a variable neighborhood search (VNS) method; which is a local search-based metaheuristic method [16], is combined with the path-relinking (PR) method [17]. The resulting method, called VNSPR, shows well performance in either runtime or accuracy. In [18], an efficient implementation is presented for the scatter search (SS) method of [19] with satisfying approximation results. Also, a dynamic convexized (DC) method in [20] performs a local search on a dynamically updated auxiliary function, which shows good performance in solving the max-cut problem.

In this paper, we present a Riemannian Gradient-based algorithm for solving a Max-Cut problem, which we refer to as the RGMC. It is shown how we can formulate a max-cut problem as an unconstrained optimization problem over a Riemannian manifold. Then, we provide an argument for convergence and prove the convergence of the proposed gradient-based algorithm. Using simulations, we show that our method is much faster than the other ones by a slight decrease in the performance. Specifically, on average it is 50 times faster than the best well-known methods and finds a cut with a value of 0.9729 of the max-cut value of the others.

The remaining of the paper is as follows. Section II includes our main results. Simulations are given in Section III. Section IV concludes the paper and the definitions and proofs are presented in Section V.

II. MAIN RESULTS

The max-cut problem in 2 can be rewritten as

$$\begin{aligned} \max_{\mathbf{y}} \quad & \frac{1}{4} \mathbf{y}^T \mathbf{L} \mathbf{y} \\ \text{s.t.} \quad & y_i^2 = 1, \quad i = 1, \dots, n, \end{aligned} \quad (4)$$

where \mathbf{L} denotes the Laplacian of the graph defined as,

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \quad (5)$$

with \mathbf{D} being the degree matrix of the graph. It can be shown that (4) is equivalent to the problem:

$$\max_{\mathbf{x} \in \mathbb{R}^n} g(\mathbf{x}) = \frac{1}{4} \text{sgn}(\mathbf{x})^T \mathbf{L} \text{sgn}(\mathbf{x}), \quad (6)$$

where $\text{sgn}(\cdot)$ acts elementwise and

$$\text{sgn}(q) = \begin{cases} 1 & q \geq 0 \\ -1 & q < 0 \end{cases}.$$

Then, for the optimal solutions of (4) and (6), respectively shown by \mathbf{y}^* and \mathbf{x}^* , we get $\mathbf{y}^* = \text{sgn}(\mathbf{x}^*)$. In this way, we have formulated the max-cut problem as a nonsmooth

optimization problem over the Euclidean space.

To make (6) differentiable, we approximate $\text{sgn}(q)$ by the differentiable function $\frac{2}{\pi} \tan^{-1}(\alpha q)$, which becomes more accurate as α increases. Then, (6) can be presented as

$$g(\mathbf{x}) \simeq \frac{1}{\pi^2} \tan^{-1}(\alpha \mathbf{x})^T \mathbf{L} \tan^{-1}(\alpha \mathbf{x}). \quad (7)$$

Moreover, to incorporate the approximation accuracy in the optimization problem, we use the following penalty term:

$$|(\text{sgn}(q))^2 - (\frac{2}{\pi} \tan^{-1}(\alpha q))^2| = |1 - (\frac{2}{\pi} \tan^{-1}(\alpha q))^2|.$$

In this way, we propose the following maximization problem for the max-cut problem,

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}^n, \alpha \in \mathbb{R}} \quad & h(\mathbf{x}, \alpha) = \frac{1}{\pi^2} \tan^{-1}(\alpha \mathbf{x})^T \mathbf{L} \tan^{-1}(\alpha \mathbf{x}) - \\ & \sum_i \gamma_i |1 - (\frac{2}{\pi} \tan^{-1}(\alpha x_i))^2|, \end{aligned} \quad (8)$$

where γ_i 's are positive regularization parameters and x_i is the i th element of \mathbf{x} . Then, the max-cut solution is estimated as $\mathbf{y}^* = \text{sgn}(\mathbf{x}^*)$, where (\mathbf{x}^*, α^*) is the local optimum of (8). Since this solution is independent of α , we can rewrite (8) as,

$$\begin{aligned} \max_{\mathbf{z} \in \mathbb{R}^n} \quad & h(\mathbf{z}) = \frac{1}{\pi^2} \tan^{-1}(\mathbf{z})^T \mathbf{L} \tan^{-1}(\mathbf{z}) - \\ & \sum_i \gamma_i |1 - (\frac{2}{\pi} \tan^{-1}(z_i))^2|, \end{aligned} \quad (9)$$

where \mathbf{z}^* is a local optimum solution for (9), and $\mathbf{y}^* = \text{sgn}(\mathbf{z}^*)$.

Next, by expanding (9), we can rewrite it as

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{R}^n} \quad & -f(\mathbf{z}) = \\ & -\frac{1}{\pi^2} \tan^{-1}(\mathbf{z})^T (\mathbf{L} + 4 \text{diag}(\boldsymbol{\gamma})) \tan^{-1}(\mathbf{z}) + \sum_i \gamma_i, \end{aligned} \quad (10)$$

where $\boldsymbol{\gamma}$ is the vector of γ_i 's and $\text{diag}(\cdot)$ transfers the input vector to a diagonal matrix. To optimize the regularization parameters, we also incorporate $\boldsymbol{\gamma}$ in the problem as

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{R}^n, \boldsymbol{\gamma} \in \mathbb{R}_+^n} \quad & F(\mathbf{z}, \boldsymbol{\gamma}) = \\ & -\frac{1}{\pi^2} \tan^{-1}(\mathbf{z})^T (\mathbf{L} + 4 \text{diag}(\boldsymbol{\gamma})) \tan^{-1}(\mathbf{z}) \\ & + \sum_i \gamma_i, \end{aligned} \quad (11)$$

where \mathbb{R}_+ is the set of positive real numbers. To solve (11), we propose a gradient descent method over the product manifold $\mathcal{M} = \mathbb{R}^n \times \mathbb{R}_+^n$, as presented in Alg. 1. Then, the max-cut solution is,

$$\mathbf{y}^* = \text{sgn}(\mathbf{z}^*), \quad (12)$$

where $\mathbf{X}^* = (\mathbf{z}^*, \boldsymbol{\gamma}^*) \in \mathcal{M}$ denotes the optimum value obtained from Alg. 1 [21].

Algorithm 1: Gradient descent method on a Riemannian manifold.

Requirements;

Differentiable cost function F , manifold \mathcal{M} , inner product $\langle \cdot, \cdot \rangle$, retraction function R .

Initialization;

Initial point $\mathbf{X}_0 \in \mathcal{M}$, Scalars $\bar{\alpha} > 0$, $\beta, \sigma \in (0, 1)$, tolerance $\tau > 0$.

Result: $\mathbf{X}^* \in \mathcal{M}$.

for $i = 0, 1, 2, \dots$ **do**

Step 1: Set ξ as the negative direction of the gradient,

$\xi_j := -\text{grad}F(\mathbf{X}_j)$

Step 2: Convergence evaluation,

if $\|\xi_j\| < \tau$ **then**

└ break

Step 3: Find the smallest m satisfying

$F(\mathbf{X}_j) - F(R_{\mathbf{X}_j}(\bar{\alpha}\beta^m \xi_j)) \geq \sigma \bar{\alpha} \beta^m \langle \xi_j, \xi_j \rangle_{\mathbf{X}_j}$

Step 4: Find the modified point as

$\mathbf{X}_{j+1} := R_{\mathbf{X}_j}(\bar{\alpha}\beta^m \xi_j)$

The convergence of Alg. 1 for the problem defined by (11) is investigated in the following.

Lemma 1. *A sequence generated by Alg. 1 for (11) does not necessarily converge to a point in $\mathcal{M} = \mathbb{R}^n \times \mathbb{R}_+^n$.*

Proof. See Appendix V-B. ■

To guarantee convergence of Alg. 1, we modify problem (11) so that any sequence generated by Alg. 1 lies in a compact set. After that, we present a theorem for convergence of Alg. 1. To realize the aforementioned compactness, we should create a set that is bounded and closed. For this purpose, we modify problem (11) in two steps. In the first step, we impose the following constraints to (11), which make the feasible set bounded,

$$\begin{cases} \|\mathbf{z}\|_2 = 1 \\ \sum_i \gamma_i = 1 \end{cases},$$

where $\|\cdot\|_2$ is the l_2 -norm. Then, (11) turns into

$$\min_{\mathbf{z} \in S^{n-1}, \gamma \in S_{n-1}} F(\mathbf{z}, \gamma) = -\frac{1}{\pi^2} \tan^{-1}(\mathbf{z})^T (\mathbf{L} + 4\text{diag}(\gamma)) \tan^{-1}(\mathbf{z}), \quad (13)$$

where S^{n-1} is an $(n-1)$ -dimensional sphere and S_{n-1} is an $(n-1)$ -dimensional statistical manifold (See Definition 5).

In the second step, we use the logarithmic-barrier function to confine γ within a compact set. Thus, we modify (13) as

$$\min_{\mathbf{z} \in S^{n-1}, \gamma \in S_{n-1}} H(\mathbf{z}, \gamma) = -\frac{1}{\pi^2} \tan^{-1}(\mathbf{z})^T (\mathbf{L} + 4\text{diag}(\gamma)) \tan^{-1}(\mathbf{z}) - v \sum_i \log(\gamma_i - \varepsilon), \quad (14)$$

where v is an infinitesimal positive value and ε is a very small value in the interval $(0, 1)$. By this construction, any sequence generated by Alg. 1 for (14) will belong to the compact set $S^{n-1} \times [\varepsilon, 1 - \varepsilon]^{n-1}$. However, since $\gamma_i = \varepsilon$ is not in the feasible set of (14), we present the following

theorem to derive a guarantee for the convergence of (13).

Theorem 1. *Let v be an infinitesimal positive value and ε be a very small value in the interval $(0, 1)$ for (14). Also, assume that Alg. 1 generates the infinite sequence $\{\mathbf{X}_j\}_{j=0}^{j \rightarrow \infty}$ for (14). Then, the sequence $\{\mathbf{X}_j\}_{j=0}^{j \rightarrow \infty}$ will have a convergent point, namely $\mathbf{X}_* \in \mathcal{N} = S^{n-1} \times S_{n-1}$, for problem (13).*

Proof. See Appendix V-C. ■

III. SIMULATION RESULTS

To evaluate the performance of the proposed algorithm, we consider four other approaches for comparison purposes, including the SS, CirCut, VNSPR, and DC. Our criteria for evaluation of the methods are the value of max-cut solution as well as the runtime. All the experiments are performed in MATLAB 2015 on a computer with Intel(R) Core(TM) i5-4210U CPU (1.7GHz) and 6 GB RAM.

In the first experiment, we compare the methods over the graph tests G1, G2, G3, G4, G5, G12, G13, G14, G15, G16, G17, G22, G23, G24, G25, G26, G35, G36, G37, G38, G44, G45, G46, G47, G48, G49, G50, G51, G52, G53, G54, which have binary weights generated by Helmborg et al. [22]. The results presented in Table I show that our algorithm is more fast and accurate by finding a cut value with the average ratio of 0.9729 to the best cut and an average runtime ratio of 0.02 to the best of the other algorithms. To initialize the algorithm for solving (14), we apply the singular value decomposition (SVD) to the Laplacian matrix of the graph and choose a singular vector for initializing \mathbf{z} over S^{n-1} . We also generate a random point over S_{n-1} to initialize γ .

IV. CONCLUSION

We proposed an efficient Riemannian optimization algorithm for the max cut problem. To solve this problem, a Riemannian gradient descent was developed and its convergence was proved. Simulation results showed high accuracy and speed of the proposed algorithm in terms of the cut value and the runtime. Specifically, by inspecting the runtimes, it was concluded that our method is almost 50 times faster than some well-known algorithms at the cost of slightly losing the max-cut value, which is 0.9729 of the others.

V. APPENDICES

A. Some Definitions

In the following some definitions required for Alg. 1 are presented. More information may be found in [21].

Definition 1. *Consider the manifold \mathcal{M} . The set of all tangent vectors at a point $\mathbf{x} \in \mathcal{M}$ denoted by $T_{\mathbf{x}}\mathcal{M}$ is called the tangent space at this point. Also, the disjoint union of tangent spaces is called the tangent bundle denoted by $T\mathcal{M}$.*

Definition 2. *Let \mathcal{M} be a manifold. A smooth function, which assigns a tangent vector to each point of \mathcal{M} is called a tangent vector field ξ on \mathcal{M} . An example of the tangent vector field*

TABLE I
COMPARISON OF DIFFERENT MAX-CUT ALGORITHMS.

Instance			SS		circuit		VNSPR		DC		RGMC		Ratio to Best	
Name	V	E	value	time	value	time	value	time	value	time	value	time	value	time
G1	800	19176	11624	139	11624	352	11621	22732	11624	184.422	11310	1.68	0.973	0.012
G2	800	19176	11620	167.2	11617	283	11615	22719	11620	183.938	11304	2.22	0.973	0.013
G3	800	19176	11622	180.1	11622	330	11622	23890	11622	183.094	11313	1.17	0.973	0.006
G4	800	19176	11646	194.4	11641	523.6	11600	24050	11646	187.297	11293	3.53	0.97	0.019
G5	800	19176	11631	205.2	11627	1128	11598	23134	11631	184.391	11300	1.54	0.972	0.008
G14	800	4649	3060	186.5	3058	128	3055	16734	3057	115.75	2969	2.81	0.97	0.024
G15	800	4661	3049	142.8	3049	155	3043	17184	3044	115.407	2959	7.76	0.97	0.067
G16	800	4672	3045	161.9	3045	142	3043	16562	3052	113.563	2958	7.3	0.969	0.064
G17	800	4667	3043	312.8	3037	365.5	3030	18555	3043	114.25	2951	3.53	0.97	0.031
G22	2000	19990	13346	1336	13346	493	13295	197654	13339	957.14	12941	6.87	0.97	0.014
G23	2000	19990	13317	1022	13317	457	13290	193707	13323	987.328	12958	6.8	0.973	0.015
G24	2000	19990	13303	1191	13314	521	13276	195749	13314	968.891	12931	6.84	0.971	0.013
G25	2000	19990	13320	1299	13326	1600	12298	212563	13324	972.625	12973	9.16	0.974	0.009
G26	2000	19990	13294	1415	13314	1569	12290	228969	13313	965.453	12950	7.33	0.973	0.008
G35	2000	11778	7668	1258	7670	440	7635	167221	7647	860.312	7393	7	0.964	0.016
G36	2000	11766	7660	1392	7660	400	7632	167203	7625	858.406	7353	8.44	0.96	0.021
G37	2000	11785	7664	1387	7666	382	7643	170786	7640	873.203	7368	13.07	0.961	0.034
G38	2000	11779	7681	1012	7646	1189	7602	178570	7641	886.125	7372	15.25	0.96	0.017
G44	1000	9990	6648	355.9	6643	192	6642	34519	6650	218.172	6481	3.31	0.975	0.017
G45	1000	9990	6642	354.3	6652	210	6646	34179	6647	217.391	6454	2.36	0.97	0.011
G46	1000	9990	6634	498.1	6645	638.7	6630	38854	6647	219.828	6452	3.38	0.971	0.015
G47	1000	9990	6649	359.1	6656	632.9	6640	36587	6657	220.937	6454	3.48	0.97	0.016
G48	3000	6000	6000	20.1	6000	119	6000	64713	6000	588.203	6000	0	1	0
G49	3000	6000	6000	35.1	6000	134	6000	64749	6000	616.359	6000	0	1	0
G50	3000	6000	5880	26.8	5880	231	5880	147132	5880	773.515	5880	0	1	0
G51	1000	5909	3846	513	3837	497.3	3808	89966	3842	192.578	3735	6.12	0.971	0.032
G52	1000	5916	3849	550.8	3833	506.8	3816	95985	3840	185.765	3732	6.6	0.97	0.036
G53	1000	5914	3846	423.8	3842	502.5	3802	92459	3844	190.75	3729	11.51	0.97	0.06
G54	1000	5916	3846	429	3842	523.6	3820	98458	3831	190.453	3731	2.6	0.97	0.014

is the gradient of a real-valued smooth function f over \mathcal{M} , which is denoted by $\text{grad}f$.

Definition 3. A Riemannian manifold is a manifold equipped with a smoothly varying inner product. The inner product over a manifold is denoted by $\langle \cdot, \cdot \rangle$ and its restriction to the tangent space $T_x\mathcal{M}$ is denoted by $\langle \cdot, \cdot \rangle_x$.

Definition 4. Let \mathcal{M} be a Riemannian manifold with tangent bundle $T\mathcal{M}$. The exponential map denoted by $\text{Exp}(\cdot)$ maps the elements of $T\mathcal{M}$ to \mathcal{M} , so that for $v \in T\mathcal{M}$, $\text{Exp}(v)$ equals to h at time 1; where h is a unique geodesic starting from the base point of v with velocity v at time 0. Moreover, the retraction function brings a point on tangent bundle back to the manifold which satisfies some properties mentioned in Definition 4.1.1 of [21]. The first-order approximation of the exponential map is a way of constructing the retraction

function.

Definition 5. Let S_n be the set of all positive measures:

$$S_n = \{ \mathbf{p} = (p_0, \dots, p_{n-1}) \mid 0 < p_i, i = 0, \dots, n-1 \}. \quad (15)$$

Then, the subspace S_{n-1} of S_n with the property $\sum_{i=0}^{n-1} p_i = 1$ is called the statistical manifold.

B. Proof of Lemma 1

Proof. Let $\{\mathbf{X}_j\}_{j=0}^{j \rightarrow \infty}$ be a sequence generated by Alg. 1 for (11), where $\mathbf{X}_j = (\mathbf{z}_j, \gamma_j) \in \mathcal{M}$. Also, let \mathbf{X}^* be the limit point of the sequence $\{\mathbf{X}_j\}_{j=0}^{j \rightarrow \infty}$ and let the subsequence $\{\mathbf{X}_j\}_{j=N}^{j \rightarrow \infty}$ belong to $B_\epsilon(\mathbf{X}^*)$, which is defined as the l_2 -norm ball of radius ϵ around \mathbf{X}^* . Now, we prove by contradiction that \mathbf{X}^* does not necessarily belong to \mathcal{M} . Assume that

$\mathbf{X}^* \in \mathcal{M}$ is a local optimal point for (11). In other words, there exists an arbitrarily small ρ -neighbourhood $B_\rho(\mathbf{X}^*)$, for which we can write,

$$F(\mathbf{z}_*, \gamma_*) \leq F(\mathbf{z}, \gamma), \forall (\mathbf{z}, \gamma) \in B_\rho(\mathbf{X}^*) \cap \mathcal{M}.$$

It is easy to verify that for the sequence $\{(\mathbf{z}_*, (1 - 2^{-j})\gamma_*)\}_{j=1}^{\infty}$, there exists an integer N , where $(\mathbf{z}_*, (1 - 2^{-j})\gamma_*) \in B_\rho(\mathbf{X}^*) \cap \mathcal{M}, \forall j > N$ and $F(\mathbf{z}_*, (1 - 2^{-j})\gamma_*) < F(\mathbf{z}_*, \gamma_*)$. This contradiction proves that Alg. 1 does not have a convergent point for (11) belonging to \mathcal{M} .

C. Proof of Theorem 1

Let $\{\mathbf{X}_j\}_{j=0}^{\infty}$ be any arbitrary sequence generated by Alg. 1 for (14), which is due to Step 3 of Alg. 1, a decreasing sequence. By construction, we have formulated (14), so that the elements of $\{\mathbf{X}_j\}_{j=0}^{\infty}$ will belong to a subset of the compact set $S^{n-1} \times [\varepsilon, 1 - \varepsilon]^{n-1}$. We prove that $\{\mathbf{X}_j\}_{j=0}^{\infty}$ is also a valid sequence for (13), since it is a decreasing sequence for the objective function of (13), i.e., $F(\mathbf{X}_{j+1}) \leq F(\mathbf{X}_j), \forall k \geq 0$. Because $\{\mathbf{X}_j\}_{j=0}^{\infty}$ is decreasing for (14), for this problem we have

$$\begin{aligned} H(\mathbf{z}_{j+1}, \gamma_{j+1}) &\leq H(\mathbf{z}_j, \gamma_j) \Rightarrow \\ F(\mathbf{z}_{j+1}, \gamma_{j+1}) - v \sum_i \log(\gamma_{i_{j+1}} - \varepsilon) &\leq \\ F(\mathbf{z}_j, \gamma_j) - v \sum_i \log(\gamma_{i_j} - \varepsilon) &\Rightarrow \\ F(\mathbf{z}_{j+1}, \gamma_{j+1}) - F(\mathbf{z}_j, \gamma_j) &\leq \\ v \left(\sum_i \log(\gamma_{i_{j+1}} - \varepsilon) - \sum_i \log(\gamma_{i_j} - \varepsilon) \right) &\simeq 0. \end{aligned} \quad (16)$$

From (16), we see that for an infinitesimal value of v , a decreasing sequence for (14) is also a decreasing sequence for (13). This means that unlike (14), for (13) we can consider the sequence $\{\mathbf{X}_j\}_{j=0}^{\infty}$ belonging to the compact set $S^{n-1} \times [\varepsilon, 1 - \varepsilon]^{n-1}$. This is because unlike (14), $\gamma_i = \varepsilon$ belongs to the feasible set of (13).

It is known that for a compact set any sequence owns a convergent subsequence [23]. Consequently, $\{\mathbf{X}_j\}_{j=0}^{\infty}$ will have a convergent point, namely \mathbf{X}_* . By using Theorem 4.3.1 of [21]; which states every convergent point of a sequence generated by Alg. 1 is a critical point, we conclude the convergence of Alg. 1. ■

REFERENCES

- [1] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt, "An application of combinatorial optimization to statistical physics and circuit layout design," *Operations Research*, vol. 36, no. 3, pp. 493–513, 1988.
- [2] V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, and C. Mathieu, "Hierarchical clustering: Objective functions and algorithms," *Journal of the ACM (JACM)*, vol. 66, no. 4, pp. 1–42, 2019.
- [3] Y. Zhou, Y. Wang, P. Tang, S. Bai, W. Shen, E. Fishman, and A. Yuille, "Semi-supervised 3d abdominal multi-organ segmentation via deep multi-planar co-training," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 121–140, 2019.
- [4] M. M. Mohades, S. Majidian, and M. H. Kahaei, "Haplotype assembly using manifold optimization and error correction mechanism," *IEEE Signal Processing Letters*, vol. 26, pp. 868–872, June 2019.
- [5] H. Shen and H. Zhang, "Improving exact algorithms for max-2-sat," *Annals of Mathematics and Artificial Intelligence*, vol. 44, no. 4, pp. 419–436, 2005.

- [6] S. Sahni and T. Gonzalez, "P-complete approximation problems," *Journal of the ACM (JACM)*, vol. 23, no. 3, pp. 555–565, 1976.
- [7] T. Hofmeister and H. Lefmann, "A combinatorial design approach to maxcut," in *Annual Symposium on Theoretical Aspects of Computer Science*, pp. 439–452, Springer, 1996.
- [8] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.
- [9] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, "Optimal inapproximability results for max-cut and other 2-variable csps?," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 319–357, 2007.
- [10] S. Khot, "On the power of unique 2-prover 1-round games," in *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pp. 767–775, ACM, 2002.
- [11] J. Håstad, "Some optimal inapproximability results," *Journal of the ACM (JACM)*, vol. 48, no. 4, pp. 798–859, 2001.
- [12] W. Gosti, G. Nguyen, M. Wan, and M. Zhou, "Approximation algorithms for the max-cut problem," URL: <http://citeseer.ist.psu.edu/489604.html>, 1995.
- [13] O. Dolezal, T. Hofmeister, and H. Lefmann, "A comparison of approximation algorithms for the maxcut-problem," in *Manuscript, Universität Dortmund, Lehrstuhl Informatik*, Citeseer, 1999.
- [14] S. Burer, R. D. Monteiro, and Y. Zhang, "Rank-two relaxation heuristics for max-cut and other binary quadratic programs," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 503–521, 2002.
- [15] P. Festa, P. M. Pardalos, M. G. Resende, and C. C. Ribeiro, "Randomized heuristics for the max-cut problem," *Optimization methods and software*, vol. 17, no. 6, pp. 1033–1058, 2002.
- [16] P. Hansen and N. Mladenović, "Developments of variable neighborhood search," in *Essays and surveys in metaheuristics*, pp. 415–439, Springer, 2002.
- [17] F. Glover, "Tabu search and adaptive memory programming advances, applications and challenges," in *Interfaces in computer science and operations research*, pp. 1–75, Springer, 1997.
- [18] R. Martí, A. Duarte, and M. Laguna, "Advanced scatter search for the max-cut problem," *INFORMS Journal on Computing*, vol. 21, no. 1, pp. 26–38, 2009.
- [19] M. Laguna and R. Martí, *Scatter search: methodology and implementations in C*, vol. 24. Springer Science & Business Media, 2012.
- [20] W. Zhu, G. Lin, and M. M. Ali, "Max-k-cut by the discrete dynamic convexized method," *INFORMS Journal on Computing*, vol. 25, no. 1, pp. 27–40, 2013.
- [21] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
- [22] C. Helmberg and F. Rendl, "A spectral bundle method for semidefinite programming," *SIAM Journal on Optimization*, vol. 10, no. 3, pp. 673–696, 2000.
- [23] V. I. Bogachev and O. G. Smolyanov, *Real and functional analysis*, vol. 4. Springer, 2020.