Journal of Computational Mathematics
Vol.xx, No.x, 2023, 1–29.

http://www.global-sci.org/jcm
doi:10.4208/jcm.2303-m2021-0309

# A SIMPLE ITERATIVE ALGORITHM FOR MAXCUT*

Sihong Shao[1)]
*CAPT, LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China*
*Email: sihong@math.pku.edu.cn*
Dong Zhang
*LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China*
*Email: dongzhang@math.pku.edu.cn*
Weixi Zhang
*LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China*
*Email: zjwzrazwx@gmail.com*

**Abstract**

We propose a simple iterative (SI) algorithm for the maxcut problem through fully using an equivalent continuous formulation. It does not need rounding at all and has advantages that all subproblems have explicit analytic solutions, the cut values are monotonically updated and the iteration points converge to a local optima in finite steps via an appropriate subgradient selection. Numerical experiments on G-set demonstrate the performance. In particular, the ratios between the best cut values achieved by SI and those by some advanced combinatorial algorithms in [Ann. Oper. Res., 248 (2017), 365–403] are at least 0.986 and can be further improved to at least 0.997 by a preliminary attempt to break out of local optima.

*Mathematics subject classification:* 90C27, 05C85, 65K10, 90C26, 90C32.
*Key words:* Maxcut, Iterative algorithm, Exact solution, Subgradient selection, Fractional programming.

## 1. Introduction

Given an undirected simple graph $G = (V, E)$ of order $n$ with the vertex set $V$ and the edge set $E$, a set pair $(S, S')$ is called a cut of $G$ if $S \cap S' = \varnothing$ and $S \cup S' = V$. The maxcut problem, one of Karp's 21 NP-complete problems [10], aims at finding a specific cut $(S, S')$ of $G$ to maximize the cut value

$$\operatorname{cut}(S) = \sum_{\{i,j\}\in E(S,S')} w_{ij}, \tag{1.1}$$

where $E(S, S')$ collects all edges cross between $S$ and $S'$ in $E$, and $w_{ij}$ denotes the nonnegative weight on the edge $\{i, j\} \in E$.

Due to its widespread applications in various areas [1,2,4], several maxcut algorithms have been proposed to search for approximate solutions and usually fall into two distinct categories: discrete algorithms and continuous ones. The former mainly refer to the combinatorial algorithms for maxcut, which directly deal with the discrete objective function (1.1) and usually adopt both complicated techniques to break out of local optima and advanced heuristics improving the solution quality, such as the scatter search [12], the tabu search [15] and hybrid

* Received November 2, 2021 / Revised version received March 1, 2022 / Accepted March 27, 2023 / Published online October 30, 2023 /
1) Corresponding author

strategies within the framework of evolutionary algorithms [11, 13]. In contrast, the objective functions for the latter, often obtained from the relaxation of the discrete objective function (1.1), are continuous, and thus standard continuous optimization algorithms can be applied into the relaxed problems in a straightforward manner, for instance, the Goemans-Williamson (GW) algorithm [9], the CirCut algorithm resulted from the rank-two relaxation heuristics [3], the spectral cut (SC) algorithm [8, 16] and its recursive implementation (RSC) [5, 14, 18]. For all these continuous algorithms, rounding is essential and indispensable in obtaining a cut from a solution of the corresponding relaxed problem. The GW algorithm rounds the solution of a relaxing semidefinite programming via randomly selecting the hyperplanes until it achieves an expected cut value. A deterministic strategy, named Procedure-CUT, is adopted by CirCut to round the angle-vector solution to get a best possible associated cut. The SC algorithm obtains a cut by rounding the maximal eigenvector of graph Laplacian by a threshold, while the RSC algorithm recursively distributes part of unabsorbed points into two sets corresponding to a cut where the selection and assignment are determined by rounding the approximate solution of the dual Cheeger cut problem.

In this work, we propose a novel continuous algorithm for the maxcut problem, i.e. a simple iterative (SI) algorithm. Compared with the above-mentioned continuous maxcut algorithms, the proposed SI algorithm has the following distinct advantages. First, our inner subproblem can be solved analytically (see Theorem 3.1), whereas no matter the GW algorithm or the RSC algorithm needs call other optimization solvers for the inner subproblems. This constitutes the main reason why we use the adjunct word simple for the proposed algorithm. Second, our continuous optimization problem is directly equivalent to the maxcut problem (see Theorem 2.1), and the corresponding cut is updated in an iterative manner and converges to the local maximum (see Theorem 3.4). That is, the SI algorithm does not need any rounding at all. Finally, as an iterative algorithm, SI may select the cut by SC to be the initial point (see Section 4). In other words, it can also be used to improve the quality of the solution obtained from any other algorithms.

The rest is organized as follows. Section 2 establishes an equivalent continuous formulation of the maxcut problem (1.1) and a Dinkelbach-type iterative algorithm with global convergence. However, the solvability of the related inner subproblem can not be assured due to both the NP-hardness and the lack of convexity. To this end, in Section 3, we propose our simple iterative algorithm with an analytical solution to the inner problem. Both cost analysis and quality check are performed through numerical experiments on G-set in Section 4. Besides, in order to further improve the quality, a preliminary attempt to break out of local optima is also implemented there. We are concluded with a few remarks in Section 5.

## 2. Equivalent Continuous Problems

Given an undirected graph $G = (V, E)$ with nonnegative weights, let

$$I(\boldsymbol{x}) = \sum_{\{i,j\}\in E} w_{ij}|x_i - x_j|, \tag{2.1}$$

$$\|\boldsymbol{x}\|_\infty = \max\{|x_1|, \cdots, |x_n|\}, \tag{2.2}$$

$$F(\boldsymbol{x}) = \frac{I(\boldsymbol{x})}{\|\boldsymbol{x}\|_\infty}. \tag{2.3}$$

It can be readily verified that the nonnegative function $F(\boldsymbol{x})$ can achieve its maximum on $\mathbb{R}^n \setminus \{\mathbf{0}\}$ provided by its homogeneity of degree zero.

Let

$$S^{\pm}(\boldsymbol{x}) = \{i \in V : x_i = \pm\|\boldsymbol{x}\|_\infty\}, \tag{2.4}$$

$$S^{<}(\boldsymbol{x}) = \{i \in V : |x_i| < \|\boldsymbol{x}\|_\infty\}. \tag{2.5}$$

Then for any $\boldsymbol{x}^* \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, we have

$$M(\boldsymbol{x}^*) := \left\{\boldsymbol{x} \mid \|\boldsymbol{x}\|_\infty = \|\boldsymbol{x}^*\|_\infty,\ S^{\pm}(\boldsymbol{x}^*) \subset S^{\pm}(\boldsymbol{x})\right\} \tag{2.6}$$

is a convex polytope. In fact, the convexity of $I(\boldsymbol{x})$ directly implies that, if $\boldsymbol{x}^* \in M(\boldsymbol{x}^*)$ is a maximizer of $F(\boldsymbol{x})$ on $\mathbb{R}^n \setminus \{\mathbf{0}\}$, so does any $\boldsymbol{x} \in M(\boldsymbol{x}^*)$.

For any nonempty subset $S \subset V$, we define an indicative vector $\mathbf{1}_S$

$$(\mathbf{1}_S)_i = \begin{cases} 1, & i \in S, \\ 0, & i \notin S, \end{cases}$$

and then

$$\boldsymbol{x} = \mathbf{1}_S - \mathbf{1}_{V\setminus S},$$

which satisfies

$$\frac{1}{2}F(\boldsymbol{x}) = \frac{1}{2}I(\boldsymbol{x}) = \sum_{\{i,j\}\in E} w_{ij}\frac{|x_i - x_j|}{2} = \mathrm{cut}(S). \tag{2.7}$$

**Theorem 2.1.** *The maxcut problem* (1.1) *can be rewritten into*

$$\max_{S\subset V} \mathrm{cut}(S) = \frac{1}{2}\max_{\boldsymbol{x}\in\mathbb{R}^n\setminus\{\mathbf{0}\}} F(\boldsymbol{x}), \tag{2.8}$$

*and any vector $\boldsymbol{x}^*$ reaching the maximum of $F(\boldsymbol{x})$ produces a maxcut $(S, S')$ where the subset $S$ satisfies $S^+(\boldsymbol{x}^*) \subset S \subset (S^-(\boldsymbol{x}^*))^c$.*

*Proof.* Combining $\mathrm{cut}(\varnothing) = 0$ and Eq. (2.7) for the nonempty set situation together leads directly to

$$\frac{1}{2}\max_{\boldsymbol{x}\in\mathbb{R}^n\setminus\{\mathbf{0}\}} F(\boldsymbol{x}) \geq \max_{S\subset V}\mathrm{cut}(S). \tag{2.9}$$

On the other hand, suppose $\boldsymbol{x}^*$ is a maximizer of $F(\boldsymbol{x})$ on $\mathbb{R}^n \setminus \{\mathbf{0}\}$, i.e.

$$\frac{1}{2}F(\boldsymbol{x}^*) = \frac{1}{2}\max_{\boldsymbol{x}\in\mathbb{R}^n\setminus\{\mathbf{0}\}} F(\boldsymbol{x}). \tag{2.10}$$

For any $S^*$ satisfying $S^+(\boldsymbol{x}^*) \subset S^* \subset (S^-(\boldsymbol{x}^*))^c$, there exists $\hat{\boldsymbol{x}} \in M(\boldsymbol{x}^*)$ defined in Eq. (2.6) such that

$$\frac{\hat{\boldsymbol{x}}}{\|\hat{\boldsymbol{x}}\|_\infty} = \mathbf{1}_{S^*} - \mathbf{1}_{V\setminus S^*} \tag{2.11}$$

also maximizes $F(\boldsymbol{x})$ on $\mathbb{R}^n \setminus \{\mathbf{0}\}$ thanks to the zeroth-order homogeneousness of $F(\boldsymbol{x})$. That is, we have

$$\frac{1}{2}\max_{\boldsymbol{x}\in\mathbb{R}^n\setminus\{\mathbf{0}\}} F(\boldsymbol{x}) = \frac{1}{2}F\left(\frac{\hat{\boldsymbol{x}}}{\|\hat{\boldsymbol{x}}\|_\infty}\right) = \mathrm{cut}(S^*) \leq \max_{S\subset V}\mathrm{cut}(S), \tag{2.12}$$

where Eq. (2.7) has been applied. The proof is finished as a result of Eqs. (2.9) and (2.12). □

Theorem 2.1 establishes an equivalent continuous optimization for the maxcut problem which will serve as the cornerstone of the subsequent design of an iterative algorithm. Now we only need to consider

$$r_{\max} = \max_{\boldsymbol{x}\in\mathbb{R}^n\setminus\{\mathbf{0}\}} F(\boldsymbol{x}). \tag{2.13}$$

As shown in Theorem 2.2, it can be solved via the following so-called Dinkelbach iterative scheme [7]:

$$\begin{cases} x^{k+1} = \underset{\|\boldsymbol{x}\|_p=1}{\arg\min} \left\{ r^k \|\boldsymbol{x}\|_\infty - I(\boldsymbol{x}) \right\}, \quad p \in [1,\infty], & (2.14\text{a}) \\ r^{k+1} = F(\boldsymbol{x}^{k+1}), & (2.14\text{b}) \end{cases}$$

where $\|\cdot\|_p$ denotes the standard $p$-norm in $\mathbb{R}^n$, i.e.

$$\|\boldsymbol{x}\|_p = \left(|x_1|^p + |x_2|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}}. \tag{2.15}$$

**Theorem 2.2 (Global Convergence).** *The sequence $\{r^k\}$ generated by the iterative scheme* (2.14) *from any initial point $\boldsymbol{x}^0 \in \mathbb{R}^n\setminus\{\mathbf{0}\}$ increases monotonically to the global maximum $r_{\max}$.*

*Proof.* The definition of $\boldsymbol{x}^{k+1}$ (see Eq. (2.14a)) implies

$$r^k\|\boldsymbol{x}\|_\infty - I(\boldsymbol{x}) \geq r^k\|\boldsymbol{x}^{k+1}\|_\infty - I(\boldsymbol{x}^{k+1}), \quad \forall \boldsymbol{x} \quad \text{s.t.} \quad \|\boldsymbol{x}\|_p = 1,$$

and substituting $\boldsymbol{x} = \boldsymbol{x}^k$ into the above inequality yields

$$0 = r^k\|\boldsymbol{x}^k\|_\infty - I(\boldsymbol{x}^k) \geq r^k\|\boldsymbol{x}^{k+1}\|_\infty - I(\boldsymbol{x}^{k+1}),$$

which means

$$r^k \leq r^{k+1} \leq r_{\max}, \quad \forall k \in \mathbb{N}^+.$$

Therefore,

$$\exists\, r^* \in [0, r_{\max}] \quad \text{s.t.} \quad \lim_{k\to+\infty} r^k = r^*,$$

and it suffices to show $r_{\max} \leq r^*$. To this end, we denote

$$f(r) = \min_{\|\boldsymbol{x}\|_p=1} \big(r\|\boldsymbol{x}\|_\infty - I(\boldsymbol{x})\big),$$

which must be continuous on $\mathbb{R}$ by the compactness of the unit closed sphere

$$S_p = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_p = 1\}.$$

Note that

$$\begin{aligned} f(r^k) &= r^k\|\boldsymbol{x}^{k+1}\|_\infty - I(\boldsymbol{x}^{k+1}) \\ &= r^k\|\boldsymbol{x}^{k+1}\|_\infty - r^{k+1}\|\boldsymbol{x}^{k+1}\|_\infty \\ &= \|\boldsymbol{x}^{k+1}\|_\infty(r^k - r^{k+1}) \;\to\; 0 \quad \text{as} \quad k \;\to\; +\infty, \end{aligned}$$

then we have

$$f(r^*) = \lim_{k\to+\infty} f(r^k) = 0,$$

which implies

$$r^*\|\boldsymbol{x}\|_\infty - I(\boldsymbol{x}) \geq 0, \quad \forall \boldsymbol{x} \quad \text{with} \quad \|\boldsymbol{x}\|_p = 1.$$

Hence, $\forall \boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$,

$$F(\boldsymbol{x}) = \frac{I(\boldsymbol{x})}{\|\boldsymbol{x}\|_\infty} = \frac{I(\boldsymbol{x})/\|\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_\infty/\|\boldsymbol{x}\|_p} = \frac{I(\hat{\boldsymbol{x}})}{\|\hat{\boldsymbol{x}}\|_\infty} \le r^*,$$

where $\hat{\boldsymbol{x}} = \boldsymbol{x}/\|\boldsymbol{x}\|_p$. □

Although the inner subproblem (2.14a) is not only non-convex but also non-solvable in polynomial time due to the NP-hardness of the maxcut problem, the Dinkelbach scheme (2.14) provides us a good starting point to a feasible iterative algorithm for the maxcut problem.

**Remark 2.1.** Obviously, the equivalent continuous optimization (2.13) has a fractional form, i.e. a ratio between two convex functions, but such kind of fractions have been hardly touched in the field of fractional programming [17], where concave optimization problems, like optimizing the ratio of a concave function to a convex one, are usually considered.

## 3. A Simple Iterative Algorithm

The non-convex subproblem (2.14a) brings us a significant insight to deal with a relaxed subproblem alternatively, though it can not be solved in polynomial time.

Denote the subgradient of $I(\boldsymbol{x})$ by [6]

$$\partial I(\boldsymbol{x}) = \left\{ \boldsymbol{s} = (s_1, \cdots, s_n) \,|\, s_i = \sum_{j:\{i,j\}\in E} w_{ij} z_{ij},\ z_{ij} \in \mathrm{Sgn}(x_i - x_j) \text{ and } z_{ij} = -z_{ji} \right\}, \tag{3.1}$$

where we have extended the sign function (note that sign(0) = 1 here)

$$\mathrm{sign}(t) = \begin{cases} 1, & \text{if } t \ge 0, \\ -1, & \text{if } t < 0, \end{cases} \tag{3.2}$$

into a set-valued function

$$\mathrm{Sgn}(t) = \begin{cases} \{1\}, & \text{if } t > 0, \\ \{-1\}, & \text{if } t < 0, \\ [-1,1], & \text{if } t = 0. \end{cases} \tag{3.3}$$

For $\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$, we are able to define corresponding vectorized versions in an element-wise manner

$$\mathrm{sign}(\boldsymbol{x}) = (s_1, s_2, \cdots, s_n), \quad s_i = \mathrm{sign}(x_i), \quad i = 1, 2, \ldots, n, \tag{3.4}$$
$$\mathrm{Sgn}(\boldsymbol{x}) = \{(s_1, s_2, \cdots, s_n) \mid s_i \in \mathrm{Sgn}(x_i),\ i = 1, 2, \ldots, n\}. \tag{3.5}$$

Since the function $I(\cdot)$ is convex, it holds

$$I(\boldsymbol{x}) \ge I(\boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{s}), \quad \forall \boldsymbol{s} \in \partial I(\boldsymbol{y}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \tag{3.6}$$

where $(\cdot,\cdot)$ denotes the standard inner product in $\mathbb{R}^n$. Further considering the fact that $I(\cdot)$ is homogeneous of degree one, we have

$$I(\boldsymbol{y}) = (\boldsymbol{y}, \boldsymbol{s}), \quad \forall \boldsymbol{s} \in \partial I(\boldsymbol{y}), \quad \forall \boldsymbol{y} \in \mathbb{R}^n, \tag{3.7}$$

and

$$I(\boldsymbol{x}) \geq I(\boldsymbol{y}) + (\boldsymbol{x}-\boldsymbol{y}, \boldsymbol{s}) = (\boldsymbol{x}, \boldsymbol{s}), \quad \forall \boldsymbol{s} \in \partial I(\boldsymbol{y}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n. \tag{3.8}$$

Plugging the relaxation (3.8) into the subproblem (2.14a) modifies the two-step Dinkelbach iterative scheme into the following three-step one:

$$\begin{cases} \boldsymbol{x}^{k+1} = \underset{\|\boldsymbol{x}\|_p=1}{\arg\min}\{r^k\|\boldsymbol{x}\|_\infty - (\boldsymbol{x}, \boldsymbol{s}^k)\}, \quad p \in [1,\infty], & (3.9\text{a}) \\ r^{k+1} = F(\boldsymbol{x}^{k+1}), & (3.9\text{b}) \\ \boldsymbol{s}^{k+1} \in \partial I(\boldsymbol{x}^{k+1}) & (3.9\text{c}) \end{cases}$$

with an initial data $\boldsymbol{x}^0 \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$, $r^0 = F(\boldsymbol{x}^0)$ and $\boldsymbol{s}^0 \in \partial I(\boldsymbol{x}^0)$.

It can be readily verified that the subproblem (3.9a) is convex via the relaxation (3.8) from Eq. (2.14a). More importantly, we can write down a solution to the inner subproblem (3.9a) in an analytical manner (see Theorem 3.1). That is, no any other optimization solver is needed in (3.9) and so that a simple iterative algorithm we name it. Actually, we are able to prove that such iterative scheme (3.9) still keeps the monotonicity (see Theorem 3.2) and has local convergence (see Theorem 3.4).

## 3.1. Exact solution to the inner subproblem

Let

$$L(r, \boldsymbol{v}) := \min_{\|\boldsymbol{x}\|_p=1}\{r\|\boldsymbol{x}\|_\infty - (\boldsymbol{x}, \boldsymbol{v})\}, \quad r \in \mathbb{R}, \quad \boldsymbol{v} \in \mathbb{R}^n, \tag{3.10}$$

denote the minimal value in Eq. (3.9a). In order to obtain the exact solution, we need to show first a property of the simple iteration (3.9) (see Lemma 3.1), and use it to narrow the scope of discussion.

**Lemma 3.1.** *Suppose $\boldsymbol{x}^k$, $r^k$ and $\boldsymbol{s}^k$ are generated by the simple iteration* (3.9)*. Then, for $k \geq 1$, we always have $r^k \leq \|\boldsymbol{s}^k\|_1$. In particular,*

*(1) $r^k = \|\boldsymbol{s}^k\|_1$ if and only if $\boldsymbol{x}^k/\|\boldsymbol{x}^k\|_\infty \in \mathrm{Sgn}(\boldsymbol{s}^k)$.*

*(2) $r^k < \|\boldsymbol{s}^k\|_1$ if and only if $L(r^k, \boldsymbol{s}^k) < 0$.*

*Proof.* From Hölder's inequality,

$$r^k\|\boldsymbol{x}\|_\infty - (\boldsymbol{x}, \boldsymbol{s}^k) \geq r^k\|\boldsymbol{x}\|_\infty - \|\boldsymbol{s}^k\|_1\|\boldsymbol{x}\|_\infty = \big(r^k - \|\boldsymbol{s}^k\|_1\big)\|\boldsymbol{x}\|_\infty, \tag{3.11}$$

where the equality holds if and only if $\boldsymbol{x} \in \|\boldsymbol{x}\|_\infty \mathrm{Sgn}(\boldsymbol{s}^k)$. Plugging $\boldsymbol{x} = \boldsymbol{x}^k$ into Eq. (3.11) and using Eq. (3.7) lead to

$$\big(r^k - \|\boldsymbol{s}^k\|_1\big)\|\boldsymbol{x}^k\|_\infty \leq r^k\|\boldsymbol{x}^k\|_\infty - (\boldsymbol{x}^k, \boldsymbol{s}^k) = r^k\|\boldsymbol{x}^k\|_\infty - I(\boldsymbol{x}^k) = 0,$$

and thus $r^k \leq \|\boldsymbol{s}^k\|_1$.

Meanwhile, the statement (1) corresponds to the situation in which the equality holds.

Next we will show the statement (2) is true. On one hand, if $L(r^k, \boldsymbol{s}^k) < 0$, then there exists a vector $\boldsymbol{x}^{k+1}$ satisfying

$$L(r^k, \boldsymbol{s}^k) = r^k\|\boldsymbol{x}^{k+1}\|_\infty - (\boldsymbol{x}^{k+1}, \boldsymbol{s}^k) < 0. \tag{3.12}$$

Substituting $\boldsymbol{x} = \boldsymbol{x}^{k+1}$ into Eq. (3.11) and using Eq. (3.12) yield

$$\big(r^k - \|\boldsymbol{s}^k\|_1\big)\|\boldsymbol{x}^{k+1}\|_\infty \le r^k\|\boldsymbol{x}^{k+1}\|_\infty - (\boldsymbol{x}^{k+1}, \boldsymbol{s}^k) = L(r^k, \boldsymbol{s}^k) < 0,$$

and thus $r^k < \|\boldsymbol{s}^k\|_1$.

On the other hand, assume $r^k < \|\boldsymbol{s}^k\|_1$ holds. Choose $\boldsymbol{y} \in \mathrm{Sgn}(\boldsymbol{s}^k)$ and let $\boldsymbol{x}^* = \boldsymbol{y}/\|\boldsymbol{y}\|_p$. It is obvious that $\|\boldsymbol{y}\|_\infty = 1, \|\boldsymbol{x}^*\|_p = 1$ and $(\boldsymbol{y}, \boldsymbol{s}^k) = \|\boldsymbol{s}^k\|_1$. In consequence, we have

$$\begin{aligned} L(r^k, \boldsymbol{s}^k) &\le r^k\|\boldsymbol{x}^*\|_\infty - (\boldsymbol{x}^*, \boldsymbol{s}^k) \\ &= r^k\left\|\frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_p}\right\|_\infty - \left(\frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_p}, \boldsymbol{s}^k\right) \\ &= \frac{1}{\|\boldsymbol{y}\|_p}\big(r^k - \|\boldsymbol{s}^k\|_1\big) < 0. \end{aligned}$$

The proof is complete. $\square$

Given a real number $r > 0$ and a vector $\boldsymbol{v} = (v_1, \cdots, v_n) \in \mathbb{R}^n$, in view of Lemma 3.1 and the subproblem (3.9a), we only need to consider the minimization problem

$$\boldsymbol{x}^* = \underset{\|\boldsymbol{x}\|_p=1}{\arg\min}\{r\|\boldsymbol{x}\|_\infty - (\boldsymbol{x}, \boldsymbol{v})\} \tag{3.13}$$

under the condition

$$0 < r \le \|\boldsymbol{v}\|_1. \tag{3.14}$$

Without loss of generality, it suffices to discuss an ordered situation

$$|v_1| \ge |v_2| \ge \cdots \ge |v_n| \ge |v_{n+1}| = 0, \tag{3.15}$$

where we have extended the index set into $\{1, \ldots, n+1\}$ and introduced an auxiliary element $v_{n+1} = 0$ for convenience. Consider the accumulation of increment

$$A(m) = \sum_{j=1}^{m}(|v_j| - |v_{m+1}|), \quad m \in \{1, 2, \ldots, n\}, \tag{3.16}$$

and then from Eq. (3.15) we have

$$0 = A(0) \le A(1) \le A(2) \le \cdots \le A(n) = \|\boldsymbol{v}\|_1. \tag{3.17}$$

Meanwhile, we need a key step to increase the regularity for $p \in (1, \infty)$ through rewriting the minimization problem (3.13) into

$$\begin{aligned} \min_{\|\boldsymbol{x}\|_p=1}\{r\|\boldsymbol{x}\|_\infty - (\boldsymbol{x}, \boldsymbol{v})\} &= \min_{\|\boldsymbol{u}\|_p=1}\{r\|\boldsymbol{u}\|_\infty - (\boldsymbol{u}, |\boldsymbol{v}|)\} &(3.18)\\ &= \min_{\boldsymbol{w}\neq\boldsymbol{0}} \frac{r\|\boldsymbol{w}\|_\infty - (\boldsymbol{w}, |\boldsymbol{v}|)}{\|\boldsymbol{w}\|_p} &(3.19)\\ &= \min_{\|\boldsymbol{z}\|_\infty=1} \frac{r - (\boldsymbol{z}, |\boldsymbol{v}|)}{\|\boldsymbol{z}\|_p} &(3.20)\\ &= \min_{\boldsymbol{z}\in B_\infty} G(\boldsymbol{z}), &(3.21) \end{aligned}$$

where

$$G(\boldsymbol{z}) = \frac{r - (\boldsymbol{z}, |\boldsymbol{v}|)}{\|\boldsymbol{z}\|_p}, \tag{3.22}$$

$$B_\infty = \{\boldsymbol{z} \in \mathbb{R}^n : \|\boldsymbol{z}\|_\infty \leq 1\}, \tag{3.23}$$

and the absolute value is taken in an element-wise manner, e.g., $|\boldsymbol{v}| = (|v_1|, |v_2|, \cdots, |v_n|)$. Here we have used extensively the fact that $\|\cdot\|_p$, $\|\cdot\|_\infty$ and $(\cdot, |\boldsymbol{v}|)$ are all homogeneous of degree one, and in Eq. (3.21) the fact that $G(\boldsymbol{z})$ achieves its minimum on the boundary of the feasible region $B_\infty$ due to

$$G(\lambda \boldsymbol{z}) - G(\boldsymbol{z}) = \left(\frac{1}{\lambda} - 1\right) \frac{r}{\|\boldsymbol{z}\|_p} < 0 \quad \text{for} \quad \lambda > 1. \tag{3.24}$$

If one denotes the corresponding minimizers in Eqs. (3.18)-(3.21) by $\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{w}^*$, and $\boldsymbol{z}^*$, respectively, then we have

$$\boldsymbol{u}^* = \operatorname{sign}(\boldsymbol{v}) \cdot \boldsymbol{x}^*, \tag{3.25}$$

$$\boldsymbol{w}^* = \|\boldsymbol{w}^*\|_p \boldsymbol{u}^*, \tag{3.26}$$

$$\boldsymbol{z}^* = \frac{\boldsymbol{w}^*}{\|\boldsymbol{w}^*\|_\infty}, \tag{3.27}$$

where $\boldsymbol{x} \cdot \boldsymbol{y}$ in Eq. (3.25) denotes element-by-element multiplication of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. It is obvious that we only need to search for the minimizer $\boldsymbol{z}^*$ to the minimization problem (3.21), with which we are able to reach our target

$$\boldsymbol{x}^* = \frac{\operatorname{sign}(\boldsymbol{v}) \cdot \boldsymbol{z}^*}{\|\boldsymbol{z}^*\|_p} \tag{3.28}$$

by virtue of the one-to-one mappings (3.25)-(3.27).

According to the condition (3.14), the rest of the discussion falls into the following three scenarios. Before that, we need the following lemma to characterize the minimizer of $G(\boldsymbol{z})$ on $B_\infty$, denoted by $\boldsymbol{z}^* = (z_1^*, z_2^*, \cdots, z_n^*)$.

**Lemma 3.2.** *Let $\boldsymbol{z}^* = (z_1^*, z_2^*, \cdots, z_n^*)$ be the minimizer of $G(\boldsymbol{z})$ on $B_\infty$. If $r < \|\boldsymbol{v}\|_1$ and $1 \leq p < \infty$, then $\boldsymbol{z}^*$ has not only nonnegative elements, but also the same order as $|\boldsymbol{v}|$.*

*Proof.* Since $\boldsymbol{z}^* = (z_1^*, z_2^*, \cdots, z_n^*)$ is the minimizer of $G(\boldsymbol{z})$ on $B_\infty$, it holds

$$G(\boldsymbol{z}^*) \leq G(\boldsymbol{z}), \quad \forall \boldsymbol{z} \in B_\infty.$$

In the following, the proof is by contradiction and split into three steps.

First, the condition $r < \|\boldsymbol{v}\|_1$ directly implies that $\boldsymbol{z}^*$ satisfies

$$r - (\boldsymbol{z}^*, |\boldsymbol{v}|) = G(\boldsymbol{z}^*)\|\boldsymbol{z}^*\|_p \leq G(\mathbf{1})\|\boldsymbol{z}^*\|_p = \frac{\|\boldsymbol{z}^*\|_p}{\|\mathbf{1}\|_p}(r - \|\boldsymbol{v}\|_1) < 0, \tag{3.29}$$

where $\mathbf{1} \in B_\infty$.

Second, if there exists some $i \in \{1, \ldots, n\}$ such that $z_i^* < 0$, and let $\hat{\boldsymbol{z}} = \{\hat{z}_1, \cdots, \hat{z}_n\}$ be a vector satisfying: $\hat{z}_i = 0$ and $\hat{z}_j = z_j^*$ for $j \neq i$, then we have $\hat{\boldsymbol{z}} \in B_\infty$ and

$$G(\hat{\boldsymbol{z}}) = \frac{r - (\hat{\boldsymbol{z}}, |\boldsymbol{v}|)}{\|\hat{\boldsymbol{z}}\|_p} \leq \frac{r - (\boldsymbol{z}^*, |\boldsymbol{v}|)}{\|\hat{\boldsymbol{z}}\|_p} < \frac{r - (\boldsymbol{z}^*, |\boldsymbol{v}|)}{\|\boldsymbol{z}^*\|_p} = G(\boldsymbol{z}^*) \tag{3.30}$$

from (3.29). This contradicts to the fact that $\boldsymbol{z}^*$ is the minimizer of $G(\boldsymbol{z})$ on $B_\infty$.

Third, if there exists $i, j \in \{1, \ldots, n\}$ such that $(z_i^* - z_j^*)(|v_i| - |v_j|) < 0$, let $\hat{\boldsymbol{z}}$ be a vector obtained by exchanging the $i$-th and the $j$-th elements of $\boldsymbol{z}$, then $\hat{\boldsymbol{z}} \in B_\infty$, $\|\hat{\boldsymbol{z}}\|_p = \|\boldsymbol{z}^*\|_p$, and

$$G(\hat{\boldsymbol{z}}) - G(\boldsymbol{z}^*) = \frac{(z_i^* - z_j^*)(|v_i| - |v_j|)}{\|\boldsymbol{z}\|_p} < 0. \tag{3.31}$$

This also contradicts with the fact that $\boldsymbol{z}^*$ achieves the minimum of $G(\boldsymbol{z})$ on $B_\infty$. □

More importantly, under the conditions of Lemma 3.2, the objective function $G(\boldsymbol{z})$ is now differentiable with the $i$-th partial derivative being

$$\frac{\partial G(\boldsymbol{z}^*)}{\partial z_i} = -\frac{|v_i|\|\boldsymbol{z}^*\|_p + \big(r - (\boldsymbol{z}^*, |\boldsymbol{v}|)\big)\|\boldsymbol{z}^*\|_p^{1-p}(z_i^*)^{p-1}}{\|\boldsymbol{z}^*\|_p^2}. \tag{3.32}$$

- **Scenario 1:**

$$r < \|\boldsymbol{v}\|_1 \quad \text{and} \quad 1 < p < \infty. \tag{3.33}$$

In view of Lemma 3.2 and the decreasing order of $|\boldsymbol{v}|$ described in Eq. (3.15), we may assume that there exists an integer $m_0 \in \{1, 2, \ldots, n\}$ such that $\boldsymbol{z}^*$ satisfies

$$1 = z_1^* = z_2^* = \cdots = z_{m_0}^* > z_{m_0+1}^* \geq \cdots \geq z_n^* \geq 0. \tag{3.34}$$

Let

$$T = \big\{\boldsymbol{t} \in \mathbb{R}^n : \boldsymbol{z}^* + \epsilon \boldsymbol{t} \in B_\infty \text{ for sufficiently small } \epsilon > 0\big\} \tag{3.35}$$

denote the tangent cone of $B_\infty$ at $\boldsymbol{z}^*$.

From Eq. (3.34), the tangent cone can be readily rewritten into

$$T = \big\{\boldsymbol{t} = (t_1, t_2, \cdots, t_n) \in \mathbb{R}^n : t_i \leq 0, \ i = 1, 2, \ldots, m_0\big\}. \tag{3.36}$$

Since the minimizer $\boldsymbol{z}^*$ achieves a local minimum of $G(\boldsymbol{z})$ on $B_\infty$, according to Eq. (3.36), we have

$$\frac{\partial G(\boldsymbol{z}^*)}{\partial \boldsymbol{t}} \geq 0, \qquad \forall \boldsymbol{t} \in T \tag{3.37}$$

$$\Leftrightarrow \begin{cases} \dfrac{\partial G(\boldsymbol{z}^*)}{\partial z_i} \leq 0, & i = 1, 2, \ldots, m_0, \\ \dfrac{\partial G(\boldsymbol{z}^*)}{\partial z_i} = 0, & i = m_0 + 1, \ldots, n. \end{cases} \tag{3.38}$$

For the sake of subsequent discussion, it is convenient to introduce the following three auxiliary quantities:

$$\alpha = \sum_{i=1}^{m_0} |v_i| - r, \tag{3.39}$$

$$\beta = \sum_{i=m_0+1}^{n} z_i^* |v_i|, \tag{3.40}$$

$$\gamma = \sum_{i=m_0+1}^{n} (z_i^*)^p, \tag{3.41}$$

and then

$$\alpha + \beta = (\boldsymbol{z}^*, |\boldsymbol{v}|) - r > 0, \tag{3.42}$$

$$m_0 + \gamma = \|\boldsymbol{z}^*\|_p^p > 0. \tag{3.43}$$

Therefore, the partial derivative (3.32) becomes

$$\frac{\partial G(\boldsymbol{z}^*)}{\partial z_i} = \frac{1}{\|\boldsymbol{z}^*\|_p} \left( \frac{\alpha + \beta}{m_0 + \gamma} (z_i^*)^{p-1} - |v_i| \right), \tag{3.44}$$

and then Eq. (3.38) directly implies

$$\begin{cases} (z_i^*)^{p-1} \le \dfrac{m_0 + \gamma}{\alpha + \beta} |v_i|, & i = 1, 2, \ldots, m_0, \\ (z_i^*)^{p-1} = \dfrac{m_0 + \gamma}{\alpha + \beta} |v_i|, & i = m_0 + 1, \ldots, n. \end{cases} \tag{3.45}$$

Substituting Eq. (3.45) into Eq. (3.41) and using Eq. (3.40) yields

$$\gamma = \sum_{i=m_0+1}^{n} (z_i^*)^{p-1} z_i^* = \sum_{i=m_0+1}^{n} \frac{m_0 + \gamma}{\alpha + \beta} |v_i| z_i^* = \frac{m_0 + \gamma}{\alpha + \beta} \beta, \tag{3.46}$$

and then, from $m_0 > 0$, we have

$$\frac{m_0 + \gamma}{\alpha + \beta} = \frac{m_0}{\alpha}, \quad \alpha > 0. \tag{3.47}$$

That is, the condition for local minimizers, Eq. (3.38) or Eq. (3.45), can be further simplified into

$$\begin{cases} (z_i^*)^{p-1} \le a_i, & i = 1, 2, \ldots, m_0, \\ (z_i^*)^{p-1} = a_i, & i = m_0 + 1, \ldots, n, \end{cases} \tag{3.48}$$

where

$$a_i = \frac{m_0}{\alpha} |v_i| = \frac{m_0 |v_i|}{\sum_{j=1}^{m_0} |v_j| - r}, \quad i = 1, 2, \ldots, n. \tag{3.49}$$

Combining Eqs. (3.34) and (3.48) determines the minimizer $\boldsymbol{z}^* = (z_1^*, z_2^*, \cdots, z_n^*)$

$$z_i^* = \min \left\{ 1, a_i^{\frac{1}{p-1}} \right\}, \quad i = 1, 2, \ldots, n, \tag{3.50}$$

and then we are able to reach our target $\boldsymbol{x}^*$ by Eq. (3.28).

Therefore the exact solution to the inner subproblem for Scenario 1 has been obtained. The only remaining thing is how to determine $m_0$ efficiently used in Eq. (3.49). This can be achieved with the help of the accumulation function $A(m)$ defined in Eq. (3.16). Namely, $m_0$ is the smallest integer $m$ satisfying $A(m) > r$

$$m_0 = \min \left\{ m \in \{1, 2, \ldots, n\} : A(m) > r \right\}, \tag{3.51}$$

which can be readily verified as follows:

$$\begin{aligned} & a_{m_0} \ge \left(z_{m_0}^*\right)^{p-1} = 1 > \left(z_{m_0+1}^*\right)^{p-1} = a_{m_0+1} \\ \Leftrightarrow\ & \frac{m_0 |v_{m_0}|}{\sum_{j=1}^{m_0} |v_j| - r} \ge 1 > \frac{m_0 |v_{m_0+1}|}{\sum_{j=1}^{m_0} |v_j| - r} \\ \Leftrightarrow\ & \sum_{j=1}^{m_0} (|v_j| - |v_{m_0+1}|) > r \ge \sum_{j=1}^{m_0-1} (|v_j| - |v_{m_0}|) \\ \Leftrightarrow\ & A(m_0) > r \ge A(m_0 - 1). \end{aligned}$$

- **Scenario 2:**

$$r < \|\boldsymbol{v}\|_1 \quad \text{and} \quad p = 1. \tag{3.52}$$

In this scenario, we still have the minimizer $\boldsymbol{z}^* = (z_1^*, z_2^*, \cdots, z_n^*)$ has nonnegative elements according to Lemma 3.2, namely, $\forall i \in \{1, 2, \ldots, n\}, 0 \le z_i^* \le 1$.

Let $m_1$ be the largest integer satisfying $A(m-1) < r$, i.e.

$$m_1 = \max\big\{m \in \{1, 2, \ldots, n\} : A(m-1) < r\big\}, \tag{3.53}$$

where $A(m)$ is the accumulation function defined by Eq. (3.16). Comparing Eq. (3.53) with Eq. (3.51), we have $m_1 \le m_0$ where $m_0$ is defined by Eq. (3.51), and specifically

$$A(m_1 - 1) < r = A(m_1) = A(m_1 + 1) = \cdots = A(m_0 - 1) < A(m_0), \tag{3.54}$$

thereby indicating

$$|v_{m_1}| > |v_{m_1+1}| = \cdots = |v_{m_0}| > |v_{m_0+1}|. \tag{3.55}$$

For $i \in \{1, 2, \ldots, n\}$, consider a continuous function on $[0, 1]$

$$G_i(t) = G(z_1^*, \cdots, z_{i-1}^*, t, z_{i+1}^*, \cdots z_n^*) = -|v_i| + \frac{r - \sum_{j \ne i} z_j^*(|v_j| - |v_i|)}{\sum_{j \ne i} |z_j^*| + t}. \tag{3.56}$$

Since $\boldsymbol{z}^*$ is a minimizer of $G(\boldsymbol{z})$ on $B_\infty$, it can be readily verified that $\min_{t \in [0,1]} G_i(t) = G(\boldsymbol{z}^*)$, i.e.

$$z_i^* \in \arg\min_{t \in [0,1]} G_i(t), \quad \forall i \in \{1, 2, \ldots, n\}, \tag{3.57}$$

from which we are able to determine $z_i^*$. The related discussion needs to split the index set into the following three cases:

(1) For $1 \le i \le m_1$, we claim that $G_i(t)$ is a strictly monotonically decreasing function due to $r - \sum_{j \ne i} z_j^*(|v_j| - |v_i|) > 0$ and thus $z_i^* = 1$. The verification shown below is in a straightforward manner

$$\begin{aligned} r - \sum_{j \ne i} z_j^*(|v_j| - |v_i|) &\ge r - \sum_{j=1}^{i-1} z_j^*(|v_j| - |v_i|) \\ &\ge r - \sum_{j=1}^{i-1} (|v_j| - |v_i|) \\ &= r - A(i-1) > 0. \end{aligned}$$

(2) For $m_0 < i \le n$, we claim that $z_i^* = 0$. Suppose the contrary that $z_{i_0}^* > 0$ is the positive element with the largest index and $i_0 > m_0$. Then the order given in Eqs. (3.15) and (3.55) directly implies

$$\begin{aligned} r - \sum_{j \ne i_0} z_j^*(|v_j| - |v_{i_0}|) &= r - \sum_{j=1}^{m_1} (|v_j| - |v_{i_0}|) - \sum_{j=m_1+1}^{i_0-1} z_j^*(|v_j| - |v_{i_0}|) \\ &\le r - \sum_{j=1}^{m_1} (|v_j| - |v_{i_0}|) \end{aligned}$$

$$\begin{cases} < r - \sum_{j=1}^{m_1}(|v_j| - |v_{m_1+1}|) = r - A(m_1) = 0, & \text{if} \quad m_1 < m_0, \\ \leq r - \sum_{j=1}^{m_1}(|v_j| - |v_{m_1+1}|) = r - A(m_0) < 0, & \text{if} \quad m_1 = m_0, \end{cases}$$

and thus we have $G_{i_0}(t)$ is a strictly monotonically increasing function. That is, the minimizer of $G_{i_0}(t)$ on $[0,1]$ is 0, i.e. $z_{i_0}^* = 0$, which is obviously a contradiction.

(3) For $m_1 < i \leq m_0$, using the results for above two cases and the order (3.55) yields

$$\begin{aligned} r - \sum_{j \neq i} z_j^*(|v_j| - |v_i|) &= r - \sum_{j=1}^{m_1}(|v_j| - |v_i|) - \sum_{j=m_1+1}^{m_0} z_j^*(|v_j| - |v_i|) \\ &= r - \sum_{j=1}^{m_1}(|v_j| - |v_{m_1+1}|) = r - A(m_1) = 0, \end{aligned}$$

and thus we have $G_i(t)$ is a constant function on $[0,1]$, i.e. $G_i(t) \equiv -|v_i|$. That is, the minimizer $z_i^*$ can take any value in $[0,1]$.

In a word, the minimizers $\boldsymbol{z}^* = (z_1^*, \cdots, z_n^*)$ for Scenario 2 constitute the following set:

$$\big\{(z_1^*, \cdots, z_n^*) \in [0,1]^n \mid z_i^* = 1,\, i = 1, \ldots, m_1, \text{ and } z_i^* = 0,\, i = m_0+1, \ldots, n\big\}, \tag{3.58}$$

and thus $\boldsymbol{x}^*$ can be also obtained through Eq. (3.28).

- **Scenario 3:**

$$r = \|\boldsymbol{v}\|_1 \quad \text{or} \quad p = \infty. \tag{3.59}$$

For $r = \|\boldsymbol{v}\|_1$, according to Lemma 3.1, we have that $\boldsymbol{x}^*$ is a minimizer of problem (3.13) if and only if

$$\frac{\boldsymbol{x}^*}{\|\boldsymbol{x}^*\|_\infty} \in \operatorname{Sgn}(\boldsymbol{v}), \quad \|\boldsymbol{x}^*\|_p = 1. \tag{3.60}$$

For $p = \infty$, the minimization problem Eq. (3.13) becomes a linear optimization problem

$$\boldsymbol{x}^* = \mathop{\arg\min}_{\|\boldsymbol{x}\|_\infty = 1} \{r - (\boldsymbol{x}, \boldsymbol{v})\}, \tag{3.61}$$

the solution of which can be still represented by Eq. (3.60). Hence, the solution Eq. (3.60) solves the minimization problem (3.13) for Scenario 3.

Summarizing the above analysis for three scenarios, we have figured out the exact solution to the inner subproblem (3.9a), as stated in the following theorem.

**Theorem 3.1 (Exact Solution).** *The solution of the minimization problem* (3.13) *under the condition* (3.14) *can be expressed analytically in Eqs.* (3.28) *and* (3.50) *for* $r < \|\boldsymbol{v}\|_1$*, and* $1 < p < \infty$*, Eqs.* (3.28) *and* (3.58) *for* $r < \|\boldsymbol{v}\|_1$*, and* $p = 1$*, and Eq.* (3.60) *otherwise.*

Hereafter we use a set denoted by $X_p^{k+1}$ to collect all those analytical solutions of the inner subproblem (3.9a), i.e. $\boldsymbol{x}^{k+1} \in X_p^{k+1}$. Obviously, $X_p^{k+1}$ is closed.

Using the above analytical solution, we are able to get a lower bound for $F(\boldsymbol{x})$ below, which will be useful in the subsequent convergence analysis.

**Corollary 3.1.** *Under the condition* (3.14) *the minimizer* $\boldsymbol{x}^*$ *to the problem* (3.13) *satisfies*

$$\frac{(\boldsymbol{x}^*, \boldsymbol{v})}{\|\boldsymbol{x}^*\|_\infty} \geq \sum_{i=1}^{m} |v_i|, \tag{3.62}$$

*where* $m = m_0$ *for both Scenario* 1 *and Scenario* 2*, and* $m = n$ *for Scenario* 3*.*

*Proof.* For both Scenario 1 and Scenario 2, there exists a minimizer $\boldsymbol{z}^* = (z_1^*, z_2^*, \cdots, z_n^*) \in B_\infty \cap \mathbb{R}_+^n$ to the equivalent problem (3.21), which satisfies

$$z_1^* = \cdots = z_{m_0}^* = 1,$$

according to Eqs. (3.50) and (3.58), respectively. Then, from Eq. (3.28), we have

$$\frac{(\boldsymbol{x}^*, \boldsymbol{v})}{\|\boldsymbol{x}^*\|_\infty} = \big(\mathrm{sign}(\boldsymbol{v}) \cdot \boldsymbol{z}^*, \boldsymbol{v}\big) = (\boldsymbol{z}^*, |\boldsymbol{v}|) \geq \sum_{i=1}^{m_0} |v_i|. \tag{3.63}$$

As for Scenario 3, using Eq. (3.60), it can be easily verified that

$$\frac{(\boldsymbol{x}^*, \boldsymbol{v})}{\|\boldsymbol{x}^*\|_\infty} = \left(\frac{\boldsymbol{x}^*}{\|\boldsymbol{x}^*\|_\infty}, \boldsymbol{v}\right) = \big(\mathrm{Sgn}(\boldsymbol{v}), \boldsymbol{v}\big) = \|\boldsymbol{v}\|_1 = \sum_{i=1}^{n} |v_i|. \tag{3.64}$$

Thus the proof is finished. □

### 3.2. Subgradient selection and convergence analysis

Besides the inner subproblem (3.9a), another key issue to implement the simple iteration (3.9) is how to choose the subgradient (see Eq. (3.9c)). For a general selection, according to Eqs. (3.7)-(3.9), we are able to obtain $r^k \leq r^{k+1}$, because

$$\begin{aligned} 0 &= r^k \|\boldsymbol{x}^k\|_\infty - I(\boldsymbol{x}^k) = r^k \|\boldsymbol{x}^k\|_\infty - (\boldsymbol{x}^k, \boldsymbol{s}^k) \\ &\geq r^k \|\boldsymbol{x}^{k+1}\|_\infty - (\boldsymbol{x}^{k+1}, \boldsymbol{s}^k) \geq r^k \|\boldsymbol{x}^{k+1}\|_\infty - I(\boldsymbol{x}^{k+1}) \\ &= \|\boldsymbol{x}^{k+1}\|_\infty (r^k - r^{k+1}). \end{aligned}$$

**Theorem 3.2 (Monotonicity).** *The sequence* $\{r^k\}$ *generated by the iterative scheme* (3.9) *from any initial point* $\boldsymbol{x}^0 \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ *increases monotonically.*

The monotone increasing in Theorem 3.2 is also a direct consequence of Lemma 3.1 and Corollary 3.1 because there exists an index set $\iota \subset \{1, 2, \ldots, n\}$, the size of which should be $m$ in Eq. (3.62) (the upper bound of summation index) such that

$$r^{k+1} = \frac{I(\boldsymbol{x}^{k+1})}{\|\boldsymbol{x}^{k+1}\|_\infty} \geq \frac{(\boldsymbol{x}^{k+1}, \boldsymbol{s}^k)}{\|\boldsymbol{x}^{k+1}\|_\infty} \geq \sum_{i \in \iota} |s_i^k| \geq r^k, \tag{3.65}$$

where the subgradient $\boldsymbol{s}^k$ is not required to be ordered like Eq. (3.15). In particular, such monotone increasing could be strict, i.e. $r^{k+1} > r^k$ via improving the last "$\geq$" in Eq. (3.65) into "$>$", if $\|\boldsymbol{s}^k\|_1 > r^k$ holds in each step. To this end, we only need to determine a subgradient $\boldsymbol{s}^\sigma \in \partial I(\boldsymbol{x}^k)$ such that $\|\boldsymbol{s}^\sigma\|_1 > r^k$ if there exists $\boldsymbol{s} \in \partial I(\boldsymbol{x}^k)$ such that $\|\boldsymbol{s}\|_1 > r^k$.

Suppose $\boldsymbol{x}^k = (x_1^k, \cdots, x_n^k)$, $\boldsymbol{s} = (s_1, \cdots, s_n)$, and then we have

$$\lambda_i(t) = |t| - \frac{x_i^k}{\|\boldsymbol{x}^k\|_\infty} t \geq 0, \quad \forall i \in \{1, \ldots, n\}, \tag{3.66}$$

$$\|\boldsymbol{s}\|_1 - r^k = \|\boldsymbol{s}\|_1 - \frac{(\boldsymbol{x}^k, \boldsymbol{s})}{\|\boldsymbol{x}^k\|_\infty} = \sum_{i=1}^n \lambda_i(s_i), \tag{3.67}$$

which yields

$$\|\boldsymbol{s}\|_1 - r^k > 0 \;\Leftrightarrow\; \exists\, i \in \{1, \ldots, n\}, \quad \text{s.t.} \quad \lambda_i(s_i) > 0. \tag{3.68}$$

Given $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$, let

$$\boldsymbol{q} = (q_1, \cdots, q_n) \in (\mathbb{R}^+)^n, \quad q_i = \sum_{j:\{i,j\}\in E} w_{ij} \mathbf{1}_{x_i = x_j}, \tag{3.69}$$

$$\boldsymbol{p} = (p_1, \cdots, p_n), \quad p_i = \sum_{j:\{i,j\}\in E} w_{ij} \operatorname{sign}(x_i - x_j) - q_i. \tag{3.70}$$

It can be readily verified that $\boldsymbol{p} \in \partial I(\boldsymbol{x})$, and

$$s_i \in \big(\partial I(\boldsymbol{x})\big)_i = [p_i - q_i, p_i + q_i], \quad \forall \boldsymbol{s} = (s_1, \cdots, s_n) \in \partial I(\boldsymbol{x}). \tag{3.71}$$

Here $(\partial I(\boldsymbol{x}))_i$ denotes the projected interval of the convex domain $\partial I(\boldsymbol{x})$ onto the $i$-th coordinate.

Denote $\bar{\boldsymbol{p}} = (\bar{p}_1, \cdots, \bar{p}_n)$ by

$$\bar{p}_i = \begin{cases} p_i \mp q_i, & \text{if} \quad i \in S^{\pm}(\boldsymbol{x}), \\ p_i + \operatorname{sign}(p_i) q_i, & \text{if} \quad i \in S^{<}(\boldsymbol{x}), \end{cases} \tag{3.72}$$

which is located on the boundary of $(\partial I(\boldsymbol{x}))_i = [p_i - q_i, p_i + q_i]$.

Accordingly, combining Eqs. (3.68), (3.71) and the convexity of $\lambda_i(t)$ given in Eq. (3.66) leads to

$$\begin{aligned} & \exists\, \boldsymbol{s} \in \partial I(\boldsymbol{x}^k), && \text{s.t.} \quad \|\boldsymbol{s}\|_1 > r^k \\ \Leftrightarrow\; & \exists\, i \in \{1, \ldots, n\}, && \text{s.t.} \quad \max_{s_i \in (\partial I(\boldsymbol{x}))_i} \lambda_i(s_i) > 0 \\ \Leftrightarrow\; & \exists\, i \in \{1, \ldots, n\}, && \text{s.t.} \quad \lambda_i(\bar{p}_i) > 0. \end{aligned}$$

That is, if there exists $i \in \{1, \ldots, n\}$ such that $\lambda_i(\bar{p}_i) > 0$, then there should exist a subgradient $\boldsymbol{s} \in \partial I(\boldsymbol{x}^k)$ satisfying $s_i = \bar{p}_i$ and $\|\boldsymbol{s}\|_1 > r^k$. Hence, if $\bar{\boldsymbol{p}} \in \partial I(\boldsymbol{x})$ (but hardly holds in general), then we may directly select $\bar{\boldsymbol{p}}$; otherwise we are able to use $\bar{\boldsymbol{p}}$ as an indicator for the subgradient selection.

Define a partial order relation "$\leq$" on $\mathbb{R}^2$ by $(x_1, y_1) \leq (x_2, y_2)$ if and only if either $x_1 < x_2$ or $x_1 = x_2, y_1 \leq y_2$ holds. Let $\Sigma(\boldsymbol{x})$ be a collection of permutations of $\{1, 2, \ldots, n\}$ such that for any $\sigma \in \Sigma(\boldsymbol{x})$, it holds

$$\big(x_{\sigma(1)}, \bar{p}_{\sigma(1)}\big) \leq \big(x_{\sigma(2)}, \bar{p}_{\sigma(2)}\big) \leq \cdots \leq \big(x_{\sigma(n)}, \bar{p}_{\sigma(n)}\big). \tag{3.73}$$

For any $\sigma \in \Sigma(\boldsymbol{x})$, we select

$$\boldsymbol{s}^\sigma = \big(s_1^\sigma, s_2^\sigma, \cdots, s_n^\sigma\big) \in \partial I(\boldsymbol{x}), \tag{3.74}$$

$$s_i^\sigma = \sum_{j:\{i,j\}\in E} w_{ij} z_{ij}, \quad i = 1, 2, \ldots, n, \tag{3.75}$$

$$z_{ij} = \operatorname{sign}\big(\sigma^{-1}(i) - \sigma^{-1}(j)\big) \in \operatorname{Sgn}(x_i - x_j), \quad i, j = 1, 2, \ldots, n. \tag{3.76}$$

The following theorem demonstrates that such subgradient selection is sufficient to guarantee the strict monotonicity $r^{k+1} > r^k$ (if any) with the help of the statement (1) of Lemma 3.1.

**Theorem 3.3.** *For any permutation $\sigma \in \Sigma(\boldsymbol{x}^k)$, we have $\|\boldsymbol{s}^\sigma\| > r^k$ if and only if there exists $\boldsymbol{s} \in \partial I(\boldsymbol{x}^k)$ such that $\|\boldsymbol{s}\| > r^k$.*

*Proof.* The necessity is obvious, we only need to prove the sufficiency. Suppose $\boldsymbol{s} = (s_1, \cdots, s_n) \in \partial I(\boldsymbol{x}^k)$ satisfies $\|\boldsymbol{s}\| > r^k$. Then it can be derived that $\boldsymbol{x}^k/\|\boldsymbol{x}^k\|_\infty \notin \mathrm{Sgn}(\boldsymbol{s})$ using the statement (1) of Lemma 3.1, i.e. there exists an $i_0 \in \{1, 2, \dots, n\}$ such that

$$\frac{x^k_{i_0}}{\|\boldsymbol{x}^k\|_\infty} \notin \mathrm{Sgn}(s_{i_0}). \tag{3.77}$$

Denote $J = \{j \mid x^k_j = x^k_{i_0}\}$. Let $j_1, j_2 \in J$ be the indexes minimizing and maximizing function $\sigma^{-1}(\cdot)$ over $J$, respectively. Then we have

$$\begin{aligned} s^\sigma_{j_1} &= \sum_{t:\{t,j_1\}\in E} w_{j_1 t}\mathrm{sign}\big(\sigma^{-1}(j_1) - \sigma^{-1}(t)\big) = p_{j_1} - q_{j_1}, \\ s^\sigma_{j_2} &= \sum_{t:\{t,j_2\}\in E} w_{j_2 t}\mathrm{sign}\big(\sigma^{-1}(j_2) - \sigma^{-1}(t)\big) = p_{j_2} + q_{j_2}. \end{aligned}$$

We claim that either $j_1$ or $j_2$ is an integer $j$ such that $x^k_j \notin \|\boldsymbol{x}^k\|_\infty \mathrm{Sgn}(s^\sigma_j)$. This directly yields $\|\boldsymbol{s}^\sigma\| > r^k$ according to the statement (1) of Lemma 3.1 and thus completes the proof. Suppose the contrary

$$x^k_{i_0} = x^k_j \in \|\boldsymbol{x}^k\|_\infty \mathrm{Sgn}(s^\sigma_j), \quad j = j_1, j_2, \tag{3.78}$$

and split the discussion into the following two cases:

(1) $i_0 \in S^<(\boldsymbol{x}^k)$. Eq. (3.78) implies that

$$s^\sigma_{j_1} = s^\sigma_{j_2} = 0 \;\Leftrightarrow\; p_{j_1} = q_{j_1} \geq 0, \quad p_{j_2} = -q_{j_2} \leq 0,$$

and thus, from Eq. (3.72), we have

$$\begin{aligned} \bar{p}_{j_1} &= p_{j_1} + \mathrm{sign}(p_{j_1})q_{j_1} = p_{j_1} + \mathrm{sign}(p_{j_1})p_{j_1} \geq 0, \\ \bar{p}_{j_2} &= p_{j_2} + \mathrm{sign}(p_{j_2})q_{j_2} = p_{j_2} - \mathrm{sign}(p_{j_2})p_{j_2} \leq 0. \end{aligned}$$

Since

$$(x_{j_1}, \bar{p}_{j_1}) \leq (x_{i_0}, \bar{p}_{i_0}) \leq (x_{j_2}, \bar{p}_{j_2}), \quad x_{j_1} = x_{i_0} = x_{j_2},$$

it yields

$$0 \geq \bar{p}_{j_2} \geq \bar{p}_{i_0} \geq \bar{p}_{j_1} \geq 0 \;\Rightarrow\; \bar{p}_{i_0} = 0 \;\Rightarrow\; p_{i_0} = q_{i_0} = 0 \;\Rightarrow\; s_{i_0} = 0,$$

which contradicts Eq. (3.77).

(2) $i_0 \in S^\pm(\boldsymbol{x}^k)$. Eq. (3.78) implies that

$$\begin{cases} s_{i_0} \geq p_{i_0} - q_{i_0} = \bar{p}_{i_0} \geq \bar{p}_{j_1} = s^\sigma_{j_1} \geq 0, & \text{if} \quad i_0 \in S^+(\boldsymbol{x}^k), \\ s_{i_0} \leq p_{i_0} + q_{i_0} = \bar{p}_{i_0} \leq \bar{p}_{j_2} = s^\sigma_{j_2} \leq 0, & \text{if} \quad i_0 \in S^-(\boldsymbol{x}^k), \end{cases} \tag{3.79}$$

both of which contradict Eq. (3.77). □

In a word, we choose $\boldsymbol{s}^k = \boldsymbol{s}^\sigma \in \partial I(\boldsymbol{x}^k)$ with $\sigma \in \Sigma(\boldsymbol{x}^k)$ in the simple iterative scheme (3.9). Besides the above-mentioned strict increasing, we are able to show below that such subgradient selection assures finite-step local convergence. Before that, we would like to further specify the choice of $\boldsymbol{x}^{k+1}$ from the closed solution set $X_p^{k+1}$ at the first step. There is a natural isomorphism $h$ by a central projection between $S_p$ and $S_\infty$ which are the unit spheres in norms $p$ and $\infty$, respectively. Denote $\partial X_p^{k+1} \in X_p^{k+1}$ be the collection of points corresponding to the vertices of $h(X_p^{k+1})$ which is a convex polytope, in view of the fact that the convex function $I(\boldsymbol{x})$ achieves its maximum values on vertices among $h(X_p^{k+1})$. Hence, the three-step iterative scheme (3.9) can be crystallized into

$$\begin{cases} \boldsymbol{x}^{k+1} \in \partial X_p^{k+1}, & (3.80a) \\ r^{k+1} = F(\boldsymbol{x}^{k+1}), & (3.80b) \\ \boldsymbol{s}^{k+1} = \boldsymbol{s}^\sigma, \quad \sigma \in \Sigma(\boldsymbol{x}^{k+1}). & (3.80c) \end{cases}$$

Let

$$C = \Big\{\boldsymbol{x} \in \mathbb{R}^n \mid F(\boldsymbol{y}) \le F(\boldsymbol{x}),\ \forall \boldsymbol{y} \in \big\{T_i\boldsymbol{x} : i \in \{1,\dots,n\}\big\}\Big\}, \tag{3.81}$$

where $T_i\boldsymbol{x}$ is defined as

$$(T_i\boldsymbol{x})_j = \begin{cases} x_j, & j \ne i, \\ -x_j, & j = i. \end{cases} \tag{3.82}$$

**Theorem 3.4 (Finite-Step Local Convergence).** *Assume the sequences $\{\boldsymbol{x}^k\}$ and $\{r^k\}$ are generated by the simple iterative scheme* (3.80) *from any initial point $\boldsymbol{x}^0 \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$. There must exist $N \in \mathbb{Z}^+$ and $r^* \in \mathbb{R}$ such that, for any $k > N$, $r^k = r^*$, and $\boldsymbol{x}^{k+1} \in C$ are local maximizers.*

*Proof.* According to Eq. (3.65), for any integer $k > 0$, there exists $\iota \subset \{1,\dots,n\}$ such that $r^k \le \sum_{i\in\iota} |s_i^k| \le r^{k+1}$, which means the sequence $\{r^k\}$ can only take finite values because the set

$$\left\{ \sum_{i\in\iota} |s_i| \;\middle|\; \boldsymbol{s}^\sigma = (s_1,\cdots,s_n),\ \forall\sigma \in \Sigma(\boldsymbol{x}),\ \forall \boldsymbol{x}, \iota \right\}$$

is finite. Thus there exist $N \in \mathbb{Z}^+$ and $r^* \in \mathbb{R}$ such that $r^k = r^*$ for any $k > N$, thereby implying that

$$\begin{aligned} r^k &= \|\boldsymbol{s}^k\|_1, \\ X_p^{k+1} &= \left\{ \boldsymbol{x} \;\middle|\; \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_\infty} \in \mathrm{Sgn}(\boldsymbol{s}^k),\ \|\boldsymbol{x}\|_p = 1 \right\} \end{aligned}$$

by virtue of Lemma 3.1. That is, $\forall \boldsymbol{x}^{k+1} \in \partial X_p^{k+1}$, we have

$$\frac{\boldsymbol{x}^{k+1}}{\|\boldsymbol{x}^{k+1}\|_\infty} \in \mathrm{Sgn}(\boldsymbol{s}), \quad \forall s \in \partial I(\boldsymbol{x}^{k+1}), \tag{3.83}$$

$$S^<(\boldsymbol{x}^{k+1}) = \varnothing. \tag{3.84}$$

Now we will show $\boldsymbol{x}^{k+1} \in C$ and neglect the superscript $k+1$ hereafter for simplicity. Suppose the contrary that there exists $i \in S^\pm(\boldsymbol{x})$ satisfying $F(T_i\boldsymbol{x}) > F(\boldsymbol{x})$, and then we have

$$\|T_i\boldsymbol{x}\|_\infty = \|\boldsymbol{x}\|_\infty, \quad I(T_i\boldsymbol{x}) - I(\boldsymbol{x}) = \pm \sum_{j:\{j,i\}\in E} w_{ij} 2x_j > 0. \tag{3.85}$$

Let $\boldsymbol{s} = (s_1, \cdots, s_n) \in \partial I(\boldsymbol{x})$ be generated by

$$z_{ij} = -\frac{x_j}{\|\boldsymbol{x}\|_\infty} \in \operatorname{Sgn}(x_i - x_j),$$

and thus

$$x_i s_i = x_i \sum_{j:\{j,i\}\in E} w_{ij} z_{ij} = -\sum_{j:\{j,i\}\in E} \frac{w_{ij} x_i x_j}{\|\boldsymbol{x}\|_\infty} = -\frac{1}{2}\big(I(T_i\boldsymbol{x}) - I(\boldsymbol{x})\big) < 0,$$

which contradicts Eq. (3.83).

Finally, we want to show $\boldsymbol{x}$ is a local maximizer of $F(\cdot)$ on $\mathbb{R}^n \setminus \{\boldsymbol{0}\}$. Let $U$ be a neighborhood of $\boldsymbol{x}$ such that

$$\left\| \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_\infty} - \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_\infty} \right\|_\infty < \frac{1}{2}, \qquad \forall \boldsymbol{y} \in U,$$
$$\boldsymbol{y}' = \frac{\|\boldsymbol{x}\|_\infty}{\|\boldsymbol{y}\|_\infty}\boldsymbol{y}, \quad g(t) = I\big(t(\boldsymbol{y}' - \boldsymbol{x}) + \boldsymbol{x}\big), \quad \forall \boldsymbol{y} \in U.$$

We claim

$$g(t) - g(0) \le 0, \quad \forall t \in [0,1], \tag{3.86}$$

with which we are able to verify $\boldsymbol{x}$ is a local maximizer as follows:

$$F(\boldsymbol{y}) - F(\boldsymbol{x}) = F(\boldsymbol{y}') - F(\boldsymbol{x}) = \frac{1}{\|\boldsymbol{x}\|_\infty}\big(I(\boldsymbol{y}') - I(\boldsymbol{x})\big) = \frac{1}{\|\boldsymbol{x}\|_\infty}\big(g(1) - g(0)\big) \le 0, \quad \forall \boldsymbol{y} \in U.$$

The only remaining thing is to prove Eq. (3.86). First, it can be easily shown that $g(t)$ is linear on $[0,1]$, and there exists $\boldsymbol{s} = (s_1, \cdots, s_n) \in \partial I(\boldsymbol{x})$ such that its slope can be determined by

$$g(t) - g(0) = t(\boldsymbol{s}, \boldsymbol{y}' - \boldsymbol{x}).$$

Then, the verification of Eq. (3.86) can be completed by

$$x_j\big(y_j' - x_j\big) \le 0 \ \Rightarrow \ s_j\big(y_j' - x_j\big) \le 0, \quad j = 1, 2, \ldots, n,$$

where we have used $\|\boldsymbol{y}'\|_\infty = \|\boldsymbol{x}\|_\infty$ as well as Eqs. (3.83) and (3.84). $\square$

**Remark 3.1.** Notice that $\{T_i\}_{i=1}^n$ can generate a commutative group $\mathcal{T}$ on $\mathbb{R}^n$. If we further restrict its domain to be $\{\boldsymbol{x} \mid S^<(\boldsymbol{x}) = \varnothing\}$, then $\boldsymbol{x}$ is a global maximizer of $F(\cdot)$ on $\mathbb{R}^n \setminus \{\boldsymbol{0}\}$ if and only if $F(\boldsymbol{y}) \le F(\boldsymbol{x})$ holds for any $\boldsymbol{y} = T\boldsymbol{x}$, $\forall T \in \mathcal{T}$. In such sense, the set $C$ given in (3.81) is the first order approximation to global maximizers.

## 4. Numerical Experiments

In this section, we conduct performance evaluation of the proposed SI algorithm (3.80) on the graphs with positive weight in G-set (`https://web.stanford.edu/ yyye/yyye/Gset/`), and always set the initial data $\boldsymbol{x}^0$ to be the maximal eigenvector of the graph Laplacian [8,16]. The three bipartite graphs $G48$, $G49$, $G50$ will not be considered because their optima cuts can be achieved at the initial step. The recently updated cut values achieved by a multiple search operator heuristic are chosen to be the reference [13]. SI is capable of producing approximate cuts with high quality: the ratios between the resulting cut values and the reference ones are at least 0.986 (see Table 4.2), and can be improved to 0.997 (see Table 4.3) after introducing a straightforward perturbation to break out of local optima.

### 4.1. Implementation and cost analysis

We begin with the algorithm implementation plus a preliminary cost analysis. Algorithm 4.1 gives the pseudo-code of the SI algorithm (3.80), where $\boldsymbol{x}^{k+1}, r^{k+1}$ and $\boldsymbol{s}^{k+1}$ are generated in Line 4, Line 5 and Lines 6-9, respectively. In Line 4, the function EXACT_SOLUTION randomly selects the iteration point $\boldsymbol{x}^{k+1} \in \partial X_p^{k+1}$ where the exact solution set $X_p^{k+1}$ is given in Theorem 3.1. After $\bar{\boldsymbol{p}}$ is obtained in Line 8 using Eq. (3.72), we are able to arrive at the partial order (3.73) through the Bubble Sort procedure during which the requested subgradient $\boldsymbol{s}^{k+1}$ can be automatically updated in an iterative manner (see Lines 17 and 18). This constitutes the subroutine named by SUBGRADIENT which starts from Line 12. It should be pointed out that there is randomness in determining both $\boldsymbol{x}^{k+1}$ and $\boldsymbol{s}^{k+1}$, so that the performance evaluation below is conducted in the sense of average by re-running SI (3.80) 100 times from the same initial data. A preliminary estimate of the cost for three iteration steps $T = 50, 500, 2000$ is presented in Table 4.1 where we have set $p = 2$ for instance.

The main task for reaching the exact solution set $X_p^{k+1}$ by Theorem 3.1 is to determine $m_0$ with which $m_1$ can be automatically obtained by Eq. (3.55). In our implementation, EXACT_SOLUTION uses the Bubble Sort to arrange $\boldsymbol{s}^k$ in an ascending order such that the accumulation of increment $A(n), A(n-1), \cdots, A(1)$ is calculated, successively, until $m_0(k)$

**Algorithm 4.1:** Pseudo-Code of the Simple Iterative (SI) Algorithm.

```
1  Initialize $\boldsymbol{x}^0$, $r^0$, $\boldsymbol{q}^0$, $\boldsymbol{p}^0$, $\sigma^0$, $\boldsymbol{s}^0$, $T$.
2  $k \leftarrow 0$.
3  while $k < T$ do
4      $\boldsymbol{x}^{k+1} \leftarrow$ EXACT_SOLUTION$(\boldsymbol{s}^k, r^k)$;
5      $r^{k+1} \leftarrow r^k + \delta_F(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k)$;
6      $\boldsymbol{q}^{k+1} \leftarrow \boldsymbol{q}^k + \delta_{\boldsymbol{q}}(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k)$;
7      $\boldsymbol{p}^{k+1} \leftarrow \boldsymbol{p}^k + \delta_{\boldsymbol{p}}(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k)$;
8      $\bar{\boldsymbol{p}}^{k+1} \leftarrow$ P_BAR$(\boldsymbol{x}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{p}^{k+1})$;
9      $(\boldsymbol{s}^{k+1}, \sigma^{k+1}) \leftarrow$ SUBGRADIENT$(\boldsymbol{x}^{k+1}, \bar{\boldsymbol{p}}^{k+1}, \boldsymbol{s}^k, \sigma^k)$;
10     $k \leftarrow k + 1$.
11 end
12 function$(\boldsymbol{s}, \sigma)$ = SUBGRADIENT$(\boldsymbol{x}, \bar{\boldsymbol{p}}, \boldsymbol{s}, \sigma)$
13 for $i = 1$ to $n - 1$ do
14     for $j = i + 1$ to $2$ do
15         if $(\boldsymbol{x}(\sigma(j)), \bar{\boldsymbol{p}}(\sigma(j))) \le (\boldsymbol{x}(\sigma(j-1)), \bar{\boldsymbol{p}}(\sigma(j-1)))$ then
16             $\sigma \leftarrow$ SWAP$(\sigma, j-1, j)$;
17             $\boldsymbol{s}(\sigma(j-1)) \leftarrow \boldsymbol{s}(\sigma(j-1)) - 2w_{\sigma(j)\sigma(j-1)}$ ;
18             $\boldsymbol{s}(\sigma(j)) \leftarrow \boldsymbol{s}(\sigma(j)) + 2w_{\sigma(j)\sigma(j-1)}$.
19         end
20         else
21             break.
22         end
23     end
24 end
25 end function
```

Table 4.1: Cost analysis: Mean values of $(n - m_0^k)/n, \delta_\sigma(k)/n$, and $|V(k)|/n$ obtained from 100 runs of the simple iterative (SI) algorithm (3.80) with $p = 2$. Here $(n - m_0^k), \delta_\sigma(k)$, and $|V(k)|$ are given in Eq. (3.51), Eq. (4.1), and Eq. (4.5), respectively, $n$ denotes the size of graph and $T$ gives the total step of iterations. The time complexity of SI is $\mathcal{O}(\text{mean}(c(k))nT)$ with $c(k) = (n - m_0(k)) + \delta_\sigma(k) + |V(k)|$ depending on the underlying graph. The values of $\text{mean}(c(k))/n$ are at most $0.095, 0.077, 0.077$ for $T = 50, 500, 2000$, respectively, thereby indicating $\text{mean}(c(k))$ is much smaller than $n$.

| Graph | $n$ | $\text{mean}(n - m_0^k)/n$ | | | $\text{mean}(\delta_\sigma(k))/n$ | | | $\text{mean}(\|V(k)\|)/n$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $T = 50$ | $T = 500$ | $T = 2000$ | $T = 50$ | $T = 500$ | $T = 2000$ | $T = 50$ | $T = 500$ | $T = 2000$ |
| G1 | 800 | 0.0073 | 0.0046 | 0.0045 | 0.0091 | 0.0082 | 0.0083 | 0.0010 | 0.0000 | 0.0000 |
| G2 | 800 | 0.0069 | 0.0043 | 0.0042 | 0.0090 | 0.0084 | 0.0082 | 0.0008 | 0.0000 | 0.0000 |
| G3 | 800 | 0.0072 | 0.0048 | 0.0046 | 0.0093 | 0.0081 | 0.0081 | 0.0009 | 0.0000 | 0.0000 |
| G4 | 800 | 0.0061 | 0.0042 | 0.0040 | 0.0090 | 0.0082 | 0.0081 | 0.0061 | 0.0042 | 0.0040 |
| G5 | 800 | 0.0059 | 0.0045 | 0.0043 | 0.0089 | 0.0081 | 0.0079 | 0.0007 | 0.0000 | 0.0000 |
| G14 | 800 | 0.0510 | 0.0423 | 0.0412 | 0.0246 | 0.0243 | 0.0251 | 0.0034 | 0.0000 | 0.0000 |
| G15 | 800 | 0.0550 | 0.0444 | 0.0440 | 0.0248 | 0.0253 | 0.0247 | 0.0068 | 0.0000 | 0.0000 |
| G16 | 800 | 0.0532 | 0.0445 | 0.0428 | 0.0229 | 0.0237 | 0.0237 | 0.0532 | 0.0445 | 0.0428 |
| G17 | 800 | 0.0567 | 0.0471 | 0.0461 | 0.0244 | 0.0244 | 0.0244 | 0.0020 | 0.0000 | 0.0000 |
| G22 | 2000 | 0.0146 | 0.0106 | 0.0102 | 0.0120 | 0.0125 | 0.0120 | 0.0018 | 0.0001 | 0.0001 |
| G23 | 2000 | 0.0119 | 0.0100 | 0.0099 | 0.0123 | 0.0117 | 0.0118 | 0.0021 | 0.0000 | 0.0000 |
| G24 | 2000 | 0.0135 | 0.0100 | 0.0097 | 0.0125 | 0.0117 | 0.0120 | 0.0135 | 0.0100 | 0.0097 |
| G25 | 2000 | 0.0161 | 0.0115 | 0.0111 | 0.0132 | 0.0124 | 0.0122 | 0.0161 | 0.0115 | 0.0111 |
| G26 | 2000 | 0.0140 | 0.0101 | 0.0097 | 0.0126 | 0.0127 | 0.0127 | 0.0038 | 0.0000 | 0.0000 |
| G35 | 2000 | 0.0475 | 0.0380 | 0.0370 | 0.0274 | 0.0259 | 0.0255 | 0.0475 | 0.0380 | 0.0370 |
| G36 | 2000 | 0.0566 | 0.0484 | 0.0466 | 0.0252 | 0.0286 | 0.0278 | 0.0054 | 0.0005 | 0.0000 |
| G37 | 2000 | 0.0516 | 0.0427 | 0.0414 | 0.0260 | 0.0272 | 0.0271 | 0.0076 | 0.0000 | 0.0000 |
| G38 | 2000 | 0.0595 | 0.0478 | 0.0466 | 0.0260 | 0.0292 | 0.0302 | 0.0091 | 0.0000 | 0.0000 |
| G43 | 1000 | 0.0140 | 0.0102 | 0.0100 | 0.0131 | 0.0125 | 0.0122 | 0.0024 | 0.0000 | 0.0000 |
| G44 | 1000 | 0.0169 | 0.0134 | 0.0133 | 0.0126 | 0.0135 | 0.0129 | 0.0013 | 0.0000 | 0.0000 |
| G45 | 1000 | 0.0152 | 0.0137 | 0.0135 | 0.0126 | 0.0130 | 0.0132 | 0.0152 | 0.0137 | 0.0135 |
| G46 | 1000 | 0.0112 | 0.0093 | 0.0091 | 0.0121 | 0.0121 | 0.0121 | 0.0011 | 0.0000 | 0.0000 |
| G47 | 1000 | 0.0125 | 0.0103 | 0.0100 | 0.0125 | 0.0125 | 0.0122 | 0.0125 | 0.0103 | 0.0100 |
| G51 | 1000 | 0.0527 | 0.0440 | 0.0427 | 0.0228 | 0.0241 | 0.0239 | 0.0016 | 0.0004 | 0.0000 |
| G52 | 1000 | 0.0485 | 0.0406 | 0.0391 | 0.0203 | 0.0239 | 0.0235 | 0.0485 | 0.0406 | 0.0391 |
| G53 | 1000 | 0.0544 | 0.0462 | 0.0453 | 0.0212 | 0.0242 | 0.0245 | 0.0031 | 0.0000 | 0.0000 |
| G54 | 1000 | 0.0616 | 0.0544 | 0.0520 | 0.0215 | 0.0241 | 0.0235 | 0.0017 | 0.0000 | 0.0000 |

(i.e. $m_0$ at the $k$-step iteration) can be determined via Eq. (3.51). The cost of this procedure is $\mathcal{O}((n - m_0(k))n)$. Table 4.1 shows that the mean values of $(n - m_0(k))/n$ are far less that 1, which are at most $0.060, 0.048, 0.047$ for $T = 50, 500, 2000$, respectively. In particular, EXACT_SOLUTION reduces to Scenario 3 and thus only costs $\mathcal{O}(n)$ for $p = \infty$.

The subroutine SUBGRADIENT is also a Bubble Sort procedure and costs $\mathcal{O}(n^2)$ in worst cases. However, it should be pointed out that the efficiency of Bubble Sort depends on the initial order and actually costs $\mathcal{O}((\delta_\sigma(k)+1)n)$, where

$$\delta_\sigma(k) = \frac{1}{2n}\sum_{i=1}^{n}\left|\sigma^{k+1}(i)-\sigma^k(i)\right| \tag{4.1}$$

denotes an average displacement of the permutation $\sigma$ used in Eq. (3.73) for the $k$-th iteration. Table 4.1 shows that the ratios between mean$(\delta_\sigma(k))$ and $n$ are at most 0.03.

It remains to estimate the cost for updating $r, \boldsymbol{q}, \boldsymbol{p}$ in Lines 5-7, where we prefer to only calculate the increment

$$\delta_\Gamma(\boldsymbol{x}^{k+1},\boldsymbol{x}^k) := \Gamma(\boldsymbol{x}^{k+1}) - \Gamma(\boldsymbol{x}^k), \quad \Gamma \in \{F, \boldsymbol{q}, \boldsymbol{p}\}. \tag{4.2}$$

Let

$$\boldsymbol{z}^k = \frac{\boldsymbol{x}^k}{\|\boldsymbol{x}^k\|_\infty}, \quad \boldsymbol{z}^{k,i} = \left(z_1^{k,i},\cdots,z_n^{k,i}\right) = \left(z_1^{k+1},\cdots,z_i^{k+1},z_{i+1}^k,\cdots,z_n^k\right). \tag{4.3}$$

Then we have

$$\begin{aligned}\delta_\Gamma(\boldsymbol{x}^{k+1},\boldsymbol{x}^k) = \delta_\Gamma(\boldsymbol{z}^{k+1},\boldsymbol{z}^k) &= \sum_{i=1}^{n}\delta_\Gamma(\boldsymbol{z}^{k,i},\boldsymbol{z}^{k,i-1})\\ &= \sum_{i\in V(k)}\delta_\Gamma(\boldsymbol{z}^{k,i},\boldsymbol{z}^{k,i-1}), \quad \Gamma \in \{F,\boldsymbol{q},\boldsymbol{p}\},\end{aligned} \tag{4.4}$$

where

$$V(k) = \left\{i \mid z_i^{k+1} \neq z_i^k\right\}. \tag{4.5}$$

Consequently, the complexity of calculating $\delta_\Gamma(\boldsymbol{x}^{k+1},\boldsymbol{x}^k)$ is $\mathcal{O}(|V(k)|n)$ since we need $\mathcal{O}(n)$ to produce each component $\delta_\Gamma(\boldsymbol{z}^{k,i},\boldsymbol{z}^{k,i-1})$. Table 4.1 reveals that mean$(|V(k)|)$ is much smaller than $n$ and sometimes vanishes as $T$ increases.

In total, the SI algorithm costs $\mathcal{O}(c(k)n)$ in the $k$-th step and

$$c(k) := \big(n - m_0(k)\big) + \delta_\sigma(k) + |V(k)|$$

depends on the underlying graph like its weights and order. Actual numerical experiments in Table 4.1 show that the mean value of $c(k)$ is much smaller than $n$.

### 4.2. Quality check

We are now ready for quality check of numerical solutions achieved by the simple algorithm (3.80). The numerical results for the RSC algorithm based on graph Laplacian ($\Delta_2$-RSC) [14] and graph 1-Laplacian ($\Delta_1$-RSC) [5], as well as the GW algorithm [14] are adopted for comparison.

The quality check is performed based on numerical solutions until $T = 2000$. Table 4.2 shows the minimum, mean and maximum cut values during 100 runs for $p = 1, 2, \infty$. It can be easily seen there that the results for different $p$ $(= 1, 2, \infty)$ are comparable and are all very close to the reference values. Actually, the ratios between the best cut values by SI (chosen from the maximum cut values over $p = 1, 2, \infty$) and the reference ones are at least 0.986 (see the results for G36), while the numerical lower bound for such ratios is about $0.946, 0.933$ and $0.949$ for the GW, $\Delta_2$-RSC and $\Delta_1$-RSC algorithms, respectively. In particular, for the case of $p = \infty$, the ratios between the minimum, mean, maximum cut values and the reference ones are at

Table 4.2: Quality check: Cut values for 27 problem instances in G-set obtained from 100 runs of the simple iterative (SI) algorithm (3.80) with $p = 1, 2, \infty$. Each run starts from the maximal eigenvector of the graph Laplacian and undergoes $T = 2000$ iterations. The minimum, mean and maximum cut values are recorded and compared to the reference ones obtained by combinatorial algorithms [13], while the initial cut values are listed in the second column. The ratios between the best cut values (in italics) by SI and the reference values are at least 0.986 (see G36).

| Graph | Initial | $p = 1$ | | | $p = 2$ | | | $p = \infty$ | | | Reference |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | min | mean | max | min | mean | max | min | mean | max | |
| G1 | 11221 | 11477 | 11524.52 | *11554* | 11505 | 11527.27 | 11548 | 11499 | 11523.3 | 11553 | 11624 |
| G2 | 11283 | 11483 | 11525.91 | *11583* | 11484 | 11505.6 | 11535 | 11473 | 11517.42 | 11557 | 11620 |
| G3 | 11298 | 11497 | 11542.53 | 11599 | 11564 | 11586.14 | 11602 | 11557 | 11586.99 | *11605* | 11622 |
| G4 | 11278 | 11520 | 11561.1 | 11597 | 11545 | 11564.49 | *11606* | 11539 | 11568.26 | 11598 | 11646 |
| G5 | 11370 | 11530 | 11568.89 | 11602 | 11561 | 11579.89 | *11607* | 11558 | 11577.22 | 11602 | 11631 |
| G14 | 2889 | 2998 | 3016.89 | *3036* | 3015 | 3022.74 | 3030 | 3012 | 3026.18 | 3033 | 3064 |
| G15 | 2771 | 2968 | 2986.36 | 3005 | 2989 | 2995.56 | 3006 | 2985 | 2996.68 | *3008* | 3050 |
| G16 | 2841 | 2973 | 2995.32 | 3009 | 3005 | 3010.14 | 3014 | 3005 | 3011.76 | *3017* | 3052 |
| G17 | 2866 | 2963 | 2981.65 | 3004 | 2996 | 3002.27 | *3012* | 2995 | 3002.7 | 3010 | 3047 |
| G22 | 12876 | 13198 | 13243.55 | *13302* | 13237 | 13267.7 | 13286 | 13248 | 13266.04 | 13290 | 13359 |
| G23 | 12817 | 13173 | 13220.99 | 13267 | 13238 | 13252.85 | *13271* | 13232 | 13246.72 | 13264 | 13344 |
| G24 | 12826 | 13194 | 13221.16 | 13244 | 13235 | 13260.34 | *13295* | 13233 | 13262.4 | 13287 | 13337 |
| G25 | 12781 | 13155 | 13189.61 | *13245* | 13185 | 13221 | 13240 | 13187 | 13213.86 | 13238 | 13340 |
| G26 | 12752 | 13140 | 13176.81 | *13222* | 13185 | 13201.6 | 13221 | 13183 | 13197.21 | 13210 | 13328 |
| G35 | 7194 | 7512 | 7540.39 | 7572 | 7558 | 7576.68 | *7595* | 7560 | 7575.4 | 7586 | 7687 |
| G36 | 7124 | 7502 | 7529.77 | 7557 | 7535 | 7548.22 | 7566 | 7531 | 7553.36 | *7576* | 7680 |
| G37 | 7162 | 7505 | 7541.03 | 7563 | 7565 | 7581.08 | *7603* | 7575 | 7590.59 | 7602 | 7691 |
| G38 | 7122 | 7513 | 7542.91 | 7568 | 7561 | 7573.63 | 7589 | 7544 | 7565.93 | *7593* | 7688 |
| G43 | 6395 | 6570 | 6609.44 | 6635 | 6599 | 6625.47 | *6645* | 6604 | 6625.8 | 6644 | 6660 |
| G44 | 6439 | 6575 | 6598.42 | 6618 | 6593 | 6608.63 | 6617 | 6590 | 6609.27 | *6621* | 6650 |
| G45 | 6364 | 6574 | 6599.49 | *6624* | 6587 | 6595.39 | 6612 | 6586 | 6593.07 | 6599 | 6654 |
| G46 | 6389 | 6557 | 6586.68 | *6612* | 6562 | 6583.31 | 6597 | 6569 | 6583.93 | 6602 | 6649 |
| G47 | 6353 | 6552 | 6583.5 | *6614* | 6584 | 6598.8 | 6609 | 6584 | 6597.24 | 6610 | 6657 |
| G51 | 3645 | 3769 | 3787.88 | 3810 | 3788 | 3797.76 | *3811* | 3785 | 3794.12 | 3806 | 3848 |
| G52 | 3645 | 3766 | 3787.31 | 3807 | 3808 | 3811.55 | 3816 | 3806 | 3811.85 | *3820* | 3851 |
| G53 | 3630 | 3778 | 3793.98 | 3812 | 3795 | 3804.37 | *3818* | 3797 | 3808.45 | 3817 | 3850 |
| G54 | 3655 | 3767 | 3789.34 | 3805 | 3800 | 3809.05 | 3818 | 3808 | 3813.69 | *3821* | 3852 |

least $0.979, 0.982, 0.986$, respectively, all of which are larger than the average ratios over these 27 graphs for the $\Delta_1$-RSC ($\simeq 0.971$), GW ($\simeq 0.960$), $\Delta_2$-RSC ($\simeq 0.958$), and SC ($\simeq 0.951$) algorithms. The SC cuts are obtained by rounding the initial data with a threshold of 0 and the cut values are shown in the second column of Table 4.2. In order to further show the overall performance in achieving high ratio by the SI algorithm, we plot the histogram of the ratios for all $3 \times 2700$ runs in Fig. 4.1. We are able to clearly observe there that:

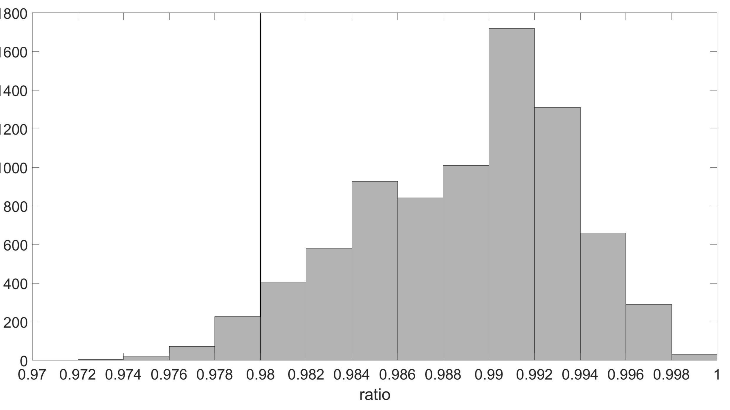


Fig. 4.1. Quality check: Histogram of the ratios for all $2700 \times 3$ runs depicted in Table 4.2 (more explanations are referred to Table 4.2). The percent of runs obtained a ratio larger than 0.980, which lie on the right of the black vertical line, exceeds 95%.

(1) More than 95% of runs achieve ratios exceeding 0.986 (see the black vertical line of Fig. 4.1).

(2) The percent of runs obtained a ratio larger than 0.986 exceeds 72%.

### 4.3. Breaking out of local optima

Within the SI algorithm, we are allowed to plug into local breakout techniques to further improve the solution quality. A preliminary attempt is to generate a new point $\tilde{\boldsymbol{x}}^{k+1} = (\tilde{x}_1, \cdots, \tilde{x}_n)$ in a stochastic manner

$$\tilde{x}_i = \begin{cases} -x_i^k & \text{with the probability of} \quad e^{-\beta|\bar{p}_i^k|}, \\ x_i^k & \text{with the probability of} \quad 1 - e^{-\beta|\bar{p}_i^k|}, \end{cases} \tag{4.6}$$

when SI is stuck at $\boldsymbol{x}^k = (x_1^k, \cdots, x_n^k)$, namely, it cannot make the function $F(\cdot)$ increase. Here we choose $\beta$ to be a controllable parameter, and $\bar{\boldsymbol{p}}^k = (\bar{p}_1^k, \cdots, \bar{p}_n^k)$ to generate $\tilde{\boldsymbol{x}}^{k+1}$ in order to decrease $F(\cdot)$ (if any) as little as possible in view of the following fact:

$$F(T_i \boldsymbol{x}^k) - F(\boldsymbol{x}^k) = -\left|\bar{p}_i^k\right|, \quad i = 1, \ldots, n, \tag{4.7}$$

where $T_i$ is defined in Eq. (3.82). In such sense, we regard the manner (4.6) as a special kind of perturbation, and the resulting algorithm is named by the simple iteration with perturbation (SI-P).

Algorithm 4.2 presents the pseudo-code of SI-P. In Lines 17-39, SI_PERTURB undergoes the simple iteration equipped with jumping out of local optima until a given final time $T$ during which the perturbation (4.6) triggers with a prescribed $\beta$ if the function values remain unchanged for $t$ steps (see Lines 29-32). It should be noted that SI_PERTURB returns the maximal cut value $r_{opt}$ and the corresponding point $\boldsymbol{x}_{opt}$ during the period of $T$. We are now left only to choose $\beta$ which should not only depend on the iteration time but also have a specific range because $\tilde{\boldsymbol{x}}^{k+1} = -\boldsymbol{x}^k$ when $\beta = 0$ and $\tilde{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^k$ when $\beta \to \infty$. Here Algorithm 4.2 adopts

**Algorithm 4.2:** Pseudo-Code of the Simple Iteration with Perturbation (SI-P).

```
Require: initial $\boldsymbol{x}^0$, $L$, $T$.
Ensure : $count$, $\boldsymbol{x}^{count}$, $r^{count}$.
$r^0 \leftarrow 0$, $count \leftarrow 0$.
while $True$ do
    for $ipert = 1$ to $L$ do
        $\beta$: chosen from $(0,1)$ randomly;
        $(\tilde{\boldsymbol{x}}^{ipert}, \tilde{r}^{ipert}) \leftarrow$ SI_PERTURB$(\boldsymbol{x}^{count}, T, t, \beta)$.
    end
    $ipert \leftarrow \text{argmax}_{ipert \in \{1,2,\cdots,L\}} \tilde{r}^{ipert}$;
    $count \leftarrow count + 1$;
    if $\tilde{r}^{ipert} > r^{count-1}$ then
        $\boldsymbol{x}^{count} \leftarrow \tilde{\boldsymbol{x}}^{ipert}$;
        $r^{count} \leftarrow \tilde{r}^{ipert}$.
    end
    else
        break.
    end
end
function $(\boldsymbol{x}^{opt}, r^{opt}) =$ SI_PERTURB$(\boldsymbol{x}^0, T, t, \beta)$
initial $r^0$, $\boldsymbol{q}^0$, $\boldsymbol{p}^0$, $\sigma^0$, $\boldsymbol{s}^0$.
$k \leftarrow 0$.
$r^{opt} \leftarrow 0$.
$\bar{\boldsymbol{p}}^k \leftarrow$ P_BAR$(\boldsymbol{x}^k, \boldsymbol{q}^k, \boldsymbol{p}^k)$.
while $k < T$ do
    $\boldsymbol{x}^{k+1} \leftarrow$ EXACT_SOLUTION$(\boldsymbol{s}^k, r^k)$;
    $r^{k+1} \leftarrow r^k + \delta_F(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k)$;
    if $r^{k+1} > r^{opt}$ then
        $r^{opt} \leftarrow r^{k+1}$;
        $\boldsymbol{x}^{opt} \leftarrow \boldsymbol{x}^{k+1}$.
    end
    if $r^{k+1} = r^k = \cdots = r^{k-t}$ then
        $\boldsymbol{x}^{k+1} \leftarrow$ PERTURB$(\bar{\boldsymbol{p}}^k, \beta)$;
        $r^{k+1} \leftarrow r^k + \delta_F(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k)$.
    end
    $\boldsymbol{q}^{k+1} \leftarrow \boldsymbol{q}^k + \delta_{\boldsymbol{q}}(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k)$;
    $\boldsymbol{p}^{k+1} \leftarrow \boldsymbol{p}^k + \delta_{\boldsymbol{p}}(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k)$;
    $\bar{\boldsymbol{p}}^{k+1} \leftarrow$ P_BAR$(\boldsymbol{x}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{p}^{k+1})$;
    $(\boldsymbol{s}^{k+1}, \sigma^{k+1}) \leftarrow$ SUBGRADIENT$(\boldsymbol{x}^{k+1}, \bar{\boldsymbol{p}}^{k+1}, \boldsymbol{s}^k, \sigma^k)$;
    $k \leftarrow k + 1$.
end
end function
```

a naive way via an outer loop in Lines 2-16, and takes $L$ runs of SI_PERTURB with different $\beta$ randomly chosen from $(0, 1)$ within each turn of loop. This outer loop continues until the cut value stops increasing (see Lines 9-15) and the variable *count* records the total number of turns (see Line 8).

Table 4.3 shows the numerical results by SI-P with $t = 3, T = 2000, L = 20$, and $p = \infty$. Now the cut values are all increased for those 27 problem instances in G-set and the ratios between the best cut values and the reference ones become at least 0.997. Moreover, the complexity of the function SI_PERTURB is almost the same as SI. Therefore, SI-P calls SI_PERTURB $count \times L$ times and performs $count \times L \times T$ iterations in total. This is the price we should pay for the improved cut values, which is at most $17 \times 20 \times 2000 = 680000$ iteration steps in the numerical experiments on G-set (see the last column of Table 4.3).

Table 4.3: Improved cut values achieved by the simple iteration with perturbation (SI-P) depicted in Algorithm 4.2 with $t = 3$, $L = 20$, $T = 2000$, and $p = \infty$. The ratios between the best cut values by SI-P and the reference ones by combinatorial algorithms [13] are at least 0.997 (see G37). The number of turns of outer loop, recorded by *count* in Line 8 of Algorithm 4.2, is at most 17, thereby meaning that the SI-P algorithm runs at most $count \times L \times T = 17 \times 20 \times 2000 = 680000$ iterations.

| Graph | Reference | Result | Ratio | Ratio without perturbation | *count* |
|---|---|---|---|---|---|
| G1 | 11624 | 11624 | 1.0000 | 0.9939 | 9 |
| G2 | 11620 | 11620 | 1.0000 | 0.9946 | 4 |
| G3 | 11622 | 11622 | 1.0000 | 0.9985 | 3 |
| G4 | 11646 | 11646 | 1.0000 | 0.9959 | 17 |
| G5 | 11631 | 11630 | 0.9999 | 0.9975 | 9 |
| G14 | 3064 | 3063 | 0.9997 | 0.9899 | 5 |
| G15 | 3050 | 3050 | 1.0000 | 0.9862 | 12 |
| G16 | 3052 | 3052 | 1.0000 | 0.9885 | 6 |
| G17 | 3047 | 3046 | 0.9997 | 0.9879 | 14 |
| G22 | 13359 | 13358 | 0.9999 | 0.9948 | 8 |
| G23 | 13344 | 13339 | 0.9996 | 0.9940 | 7 |
| G24 | 13337 | 13335 | 0.9999 | 0.9963 | 7 |
| G25 | 13340 | 13337 | 0.9998 | 0.9924 | 4 |
| G26 | 13328 | 13318 | 0.9992 | 0.9911 | 3 |
| G35 | 7687 | 7663 | 0.9969 | 0.9869 | 11 |
| G36 | 7680 | 7656 | 0.9969 | 0.9865 | 13 |
| G37 | 7691 | 7665 | 0.9966 | 0.9884 | 7 |
| G38 | 7688 | 7673 | 0.9980 | 0.9876 | 12 |
| G43 | 6660 | 6660 | 1.0000 | 0.9976 | 2 |
| G44 | 6650 | 6650 | 1.0000 | 0.9956 | 4 |
| G45 | 6654 | 6654 | 1.0000 | 0.9917 | 2 |
| G46 | 6649 | 6646 | 0.9995 | 0.9929 | 5 |
| G47 | 6657 | 6657 | 1.0000 | 0.9929 | 3 |
| G51 | 3848 | 3841 | 0.9982 | 0.9891 | 4 |
| G52 | 3851 | 3849 | 0.9995 | 0.9920 | 8 |
| G53 | 3850 | 3846 | 0.9990 | 0.9914 | 10 |
| G54 | 3852 | 3845 | 0.9982 | 0.9920 | 9 |

## 4.4. Comparison with CirCut

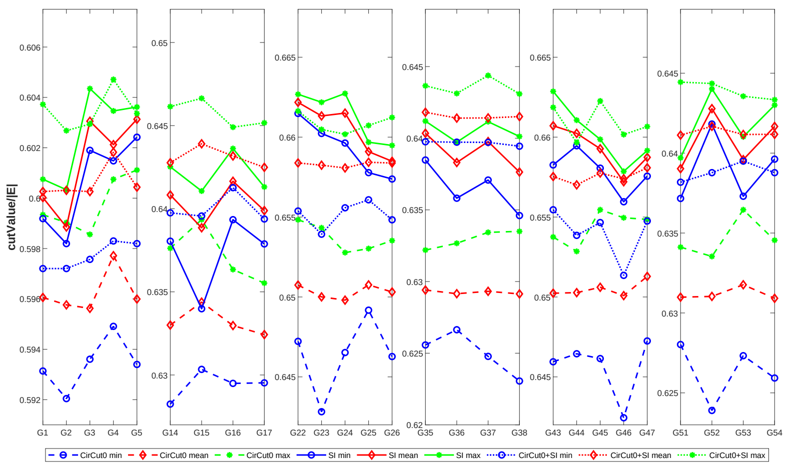
Finally, we compare SI in Algorithm 4.1 and SI_PERTURB in Algorithm 4.2 with the primal CirCut algorithm given in [3, Algorithm 1] which does not invoke any local search techniques. For convenience, we adopt the same notations as those used in [3] unless otherwise specified. It should be noted that CirCut do need a simple gradient algorithm with a backtracking Armijo line-search to minimize the nonconvex objective function $f(\theta)$ from $\theta^0$ instead of calling an external solver. The stopping condition for the line-search is, either the relative change in $f(\theta)$ is less than $\epsilon_f$ or the relative change in its gradient is less than $\epsilon_g$. We use $T_f$ (resp. $T_g$) to count the total number of times $f(\theta)$ (resp. $\nabla f(\theta)$) has been evaluated during the line-search. CirCut stops until $N$ consecutive random perturbations cannot improve the approximate cut, and let $T_P$ count the total number of perturbations. For a fair comparison, we extract the first two steps: line-search and Procedure-CUT, to form a pure rank-two relaxation for maxcut, abbreviated as CirCut0, which excludes the perturbation step, and set it against SI. When the random perturbation is invoked, CirCut is compared with SI_PERTURB. For a given graph, one usually runs the algorithm $M$ times with multiple random starting points: $\theta^0 \sim \mathcal{U}(0, 2\pi)$, and $\bar{T}_\gamma$ gives the average value of $T_\gamma$ over these $M$ times for $\gamma \in \{f, g, P\}$. The iteration of SI is not stopped until the cut values remain unchanged for $t$ consecutive steps and we run SI_PERTURB $\bar{T}_P$ times staring from a single initial data where $T_S$ counts the total number of iterations. Both SI and SI_PERTURB are re-run $M$ times from the same initial data given by the maximal eigenvector of the graph Laplacian where $\bar{T}_S$ denotes the average value of $T_S$.

We set $N = 10, M = 20, t = 3$ and use the implementation of CirCut available at Github[1)] where the parameters for the line-search are: the maximum number of rounds $n_{\max} = 200$, and the tolerances $\epsilon_f = \epsilon_g =$ 1e-4. Fig. 4.2 plots the minimum, mean and maximum cut values (normalized by the number of edges $|E|$) from the simulations with the $M$ starting points. We are able to observe there that the quality of SI solutions is much better than CirCut0, and the approximate cuts produced by SI_PERTURB are of comparable quality to CirCut with the same number of perturbations.

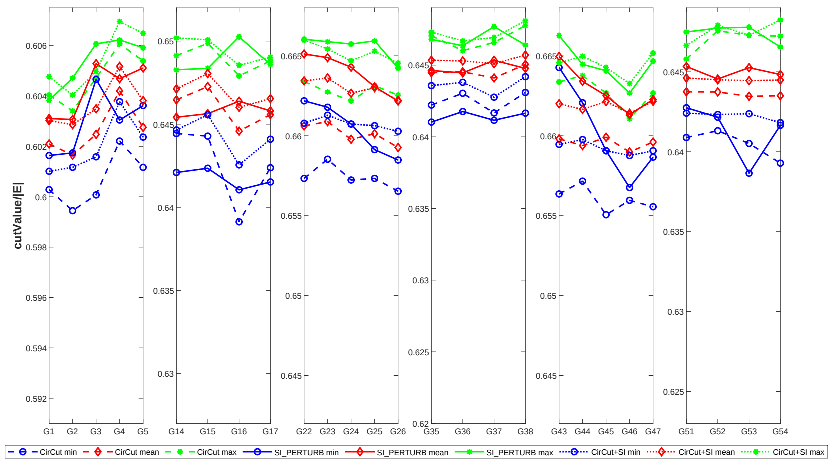
According to Theorems 3.2 and 3.4, the cut values obtained by SI are monotonically updated, and the iterative solution converges to a local optimum from any given initial data. So it will be interesting to see whether SI can improve the output cuts of CirCut0 and the results are displayed in Fig. 4.2(a) with dot lines and legend CirCut0+SI where the maximum number of SI iterations is set to be 100. It is shown there that SI does improve the quality of solutions obtained by CirCut0, which clearly indicates that CirCut0 is not guaranteed to get local optimum. Almost the same story happens with CirCut+SI (see Fig. 4.2(b)). On the other hand, we should point out that CirCut0 cannot improve the quality of the solution produced by SI. In fact, the solution obtained by SI must be a cut, while, by [3, Theorem 3.4], any cut corresponds to a critical point of $f(\theta)$, which means that the stopping condition $\nabla f(\theta) \approx \mathbf{0}$ in the line-search is immediately satisfied, and there is obviously no way to improve it further.

We run all above-mentioned algorithms in Matlab (r2019b) on a High-Performance Computing Platform: 2*Intel Xeon E5-2650-v4 (2.2 GHz, 30 MB Cache, 9.6 GT/s QPI Speed, 12 Cores, 24 Threads) with 128 GB Memory. In Table 4.4, we list the values of $\bar{T}_\gamma$ with $\gamma \in \{f, g, S, P\}$ as well as the average wall-clock time in seconds needed by only one thread for each run. In each step of the line-search, it is required to calculate $f(\theta)$ and its gradient the complexity of which

[1)] see `https://github.com/MQLib/MQLib/tree/master/src/heuristics/maxcut`



(a) SI *vs* CirCut0



(b) SI_PERTURB *vs* CirCut

Fig. 4.2. Comparison with CirCut: The minimum, mean, and maximum cut values (normalized by the number of edges $|E|$) produced by CirCut0, SI, and CirCut0+SI from multiple starting points are displayed in (a), while those obtained by CirCut, SI_PERTURB, and CirCut+SI are presented in (b). CirCut0 refers to the pure rank-two relaxation which consists of the line-search and Procedure-CUT, only the first two steps of [3, Algorithm 1]. CirCut0+SI means the output of CirCut0 serves as the input to SI for possible solution quality improvements and so does CirCut+SI.

is $\mathcal{O}(n^2)$ and does not change significantly as the search proceeds. Then, it can be deduced that the run time of both CirCut0 and CirCut should be roughly proportional to $\bar{T}_f + \bar{T}_g$, which can be readily verified in Table 4.4. By comparison, thanks to its monotonicity in Theorem 3.2 and local adjustability in Theorem 3.4, the complexity of one iteration step decreases as SI goes on, and its average over all steps is $\mathcal{O}(\text{mean}(c(k))n)$ and usually much less than $\mathcal{O}(n^2)$ as already shown in Table 4.1. There are two important implications. One is that the time ratio between SI and CirCut0 (or CirCut) should be roughly proportional to $\bar{T}_S/(\bar{T}_f + \bar{T}_g)$ where $\bar{T}_S$ for SI is not so large (less than 50). This can be easily checked in Table 4.4. The other is the run time of

Table 4.4: Comparison with CirCut: The wall-clock time in seconds, the numbers of calculating $f(\theta)$ and its gradient $\bar{T}_f$ and $\bar{T}_g$ in CirCut0 and CirCut, the number of iterations $\bar{T}_S$ in SI and SI_PERTURB, and the same number of perturbations $\bar{T}_P$ in CirCut and SI_PERTURB. It is clearly shown that SI and SI_PERTURB require much fewer iterations and thus run much faster than CirCut0 and CirCut, respectively. CirCut0 refers to the pure rank-two relaxation which only contains the first two steps of [3, Algorithm 1].

| Graph | CirCut0 | | | SI | | CirCut | | | SI_PERTURB | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time | $\bar{T}_g$ | $\bar{T}_f$ | Time | $\bar{T}_s$ | Time | $\bar{T}_g$ | $\bar{T}_f$ | Time | $\bar{T}_s$ | $\bar{T}_p$ |
| G1 | 1.73 | 74.45 | 126 | 0.34 | 32.65 | 39.37 | 1589.05 | 2817.9 | 0.61 | 643.55 | 30 |
| G2 | 1.76 | 74.75 | 125.55 | 0.37 | 33.15 | 38.33 | 1544 | 2763.15 | 0.55 | 652.9 | 30 |
| G3 | 1.75 | 73.85 | 124.35 | 0.31 | 32.1 | 41.82 | 1745.25 | 3103.15 | 0.50 | 637.25 | 33 |
| G4 | 2.09 | 92.95 | 156.85 | 0.32 | 34.65 | 37.95 | 1585.1 | 2788.6 | 0.54 | 617.7 | 30 |
| G5 | 1.88 | 77 | 130 | 0.33 | 29 | 39.18 | 1657.25 | 2909.15 | 0.56 | 677.9 | 31 |
| G14 | 1.09 | 75 | 118.3 | 0.10 | 24.15 | 27.09 | 2070.7 | 3183.85 | 0.60 | 791.1 | 34 |
| G15 | 1.09 | 75.5 | 118 | 0.11 | 28.25 | 26.14 | 2038.9 | 3137.1 | 0.58 | 780.15 | 32 |
| G16 | 1.17 | 81.4 | 127.5 | 0.11 | 25 | 22.18 | 1695.25 | 2628.1 | 0.46 | 660.8 | 28 |
| G17 | 1.00 | 76.25 | 119.35 | 0.10 | 23.3 | 22.75 | 1752 | 2694.95 | 0.45 | 658.35 | 29 |
| G22 | 4.26 | 65.6 | 101.9 | 0.90 | 31.05 | 90.58 | 1344.4 | 2076.35 | 3.81 | 801.15 | 31 |
| G23 | 3.51 | 54.4 | 84.7 | 0.91 | 31.25 | 104.28 | 1538.15 | 2395.05 | 3.74 | 795.7 | 34 |
| G24 | 3.94 | 62.35 | 96.3 | 0.91 | 32.9 | 94.22 | 1404.1 | 2175.95 | 3.69 | 805.25 | 31 |
| G25 | 3.95 | 63 | 97.7 | 0.90 | 32.45 | 87.42 | 1297.55 | 2007.25 | 3.50 | 732.9 | 29 |
| G26 | 4.02 | 63.8 | 99.9 | 0.86 | 32.45 | 83.02 | 1244.25 | 1916.05 | 3.64 | 750.6 | 28 |
| G35 | 4.50 | 86.15 | 136.7 | 0.58 | 27.6 | 155.32 | 2968.25 | 4606.1 | 5.57 | 1020 | 40 |
| G36 | 4.28 | 83.6 | 131.2 | 0.64 | 30.45 | 176.06 | 3405.65 | 5276.25 | 6.67 | 1072.1 | 42 |
| G37 | 4.67 | 88.15 | 141.5 | 0.64 | 31.1 | 165.54 | 3206.85 | 4959.4 | 6.44 | 1042.5 | 41 |
| G38 | 4.87 | 92.75 | 145.7 | 0.64 | 29.9 | 164.64 | 3217.9 | 4957.6 | 7.35 | 1170.75 | 43 |
| G43 | 1.39 | 62.55 | 97.45 | 0.24 | 33.35 | 29.69 | 1275.85 | 1998.3 | 0.62 | 598.75 | 29 |
| G44 | 1.41 | 63.25 | 99.1 | 0.24 | 29.9 | 34.89 | 1517.6 | 2402.25 | 0.72 | 746.2 | 34 |
| G45 | 1.47 | 65.55 | 102.6 | 0.24 | 28.2 | 33.33 | 1458.75 | 2319.2 | 0.69 | 739.2 | 32 |
| G46 | 1.37 | 60.95 | 95.15 | 0.24 | 27.35 | 29.63 | 1262.55 | 2002.3 | 0.55 | 606.75 | 28 |
| G47 | 1.37 | 64.25 | 99.95 | 0.23 | 27.55 | 25.59 | 1127.9 | 1777.85 | 0.58 | 624.6 | 25 |
| G51 | 1.56 | 82.5 | 129.95 | 0.15 | 25.1 | 41.71 | 2281.45 | 3516.85 | 1.02 | 793.35 | 35 |
| G52 | 1.50 | 79.85 | 125.55 | 0.15 | 24.3 | 45.90 | 2506.95 | 3875 | 1.22 | 922.45 | 38 |
| G53 | 1.66 | 88.85 | 139.55 | 0.17 | 25.4 | 36.32 | 1959.1 | 3017.85 | 0.93 | 753.15 | 32 |
| G54 | 1.49 | 79.65 | 124.25 | 0.15 | 23.25 | 35.04 | 1873.75 | 2885.05 | 0.88 | 690.9 | 30 |

SI_PERTURB should grow much more slowly as the iteration goes on where $\bar{T}_S$ is always larger than 500, which can be seen by comparing the time between SI and SI_PERTURB in Table 4.4. Hence, we are able to tell that SI and SI_PERTURB require much fewer iterations and thus run much faster than CirCut0 and CirCut, respectively. That is, although both SI and CirCut do not need call any external solver, quickly locating and checking stationary points of $f(\theta)$ as many as possible in CirCut is not that simple.

## 5. Conclusion and Outlook

An equivalent continuous fractional optimization problem and a simple iterative (SI) algorithm going from one cut to another in a monotonic and rounding-free way for the maxcut problem were proposed. "Simple" means SI utilizes the exact solutions of the inner subproblems. Numerical experiments on G-set demonstrated that the continuous SI algorithm can produce more qualified solutions than all other existing continuous algorithms. The underlying guiding thought is to build a firm bridge between discrete data world and continuous math field and then use it to design more efficient algorithms. Introducing more advanced combinatorial heuristics into SI and further improving the quality of solutions are on the way. Our attempts on the maxcut problem may provide a valuable reference for other combinatorial problems and fractional programming problems.

**Acknowledgements.** The authors would like to thank Professor Kung-Ching Chang for his long-term guidance, encouragement and support in mathematics, and useful comments on an earlier version of this paper. The authors also thank Chuan Yang for her kind help in the comparison with CirCut as well as the anonymous referees for their valuable suggestions.

This work was supported by the National Key R & D Program of China (Grant Nos. 2020AAA0105200, 2022YFA1005102), by the National Natural Science Foundation of China (Grant Nos. 12325112, 12288101, 11822102) and by the China Postdoctoral Science Foundation (Grant No. BX201700009). S. Shao is partially supported by the Beijing Academy of Artificial Intelligence.

## References

[1] Y. Aizenbud and Y. Shkolnisky, A max-cut approach to heterogeneity in cryo-electron microscopy, *J. Math. Anal. Appl.*, **479**:1 (2019), 1004–1029.

[2] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt, An application of combinatorial optimization to statistical physics and circuit layout design, *Oper. Res.*, **36**:3 (1988), 493–513.

[3] S. Burer, R.D.C. Monteiro, and Y. Zhang, Rank-two relaxation heuristics for MAX-CUT and other binary quadratic programs, *SIAM J. Optim.*, **12** (2001), 503–521.

[4] K.C. Chang and D.H.C. Du, Efficient algorithms for layer assignment problem, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **6**:1 (1987), 67–78.

[5] K.C. Chang, S. Shao, and D. Zhang, Spectrum of the signless 1-Laplacian and the dual Cheeger constant on graphs, *arXiv:1607.00489*, 2016.

[6] K.C. Chang, S. Shao, and D. Zhang, Nodal domains of eigenvectors for 1-Laplacian on graphs, *Adv. Math.*, **308** (2017), 529–574.

[8] C. Delorme and S. Poljak, Laplacian eigenvalues and the maximum cut problem, *Math. Program.*, **62** (1993), 557–574.

[7] W. Dinkelbach, On nonlinear fractional programming, *Manage. Sci.*, **13**:7 (1967), 492–498.

[9] M.X. Goemans and D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM*, **42** (1995), 1115–1145.

[10] R.M. Karp, Reducibility Among Combinatorial Problems, in: *Complexity of Computer Computations*, Springer, (1972), 85–103.

[11] G. Lin and J. Guan, An integrated method based on PSO and EDA for the max-cut problem, *Comput. Intell. Neurosci.*, (2016), 1–13.

[13] F. Ma and J. Hao, A multiple search operator heuristic for the max-k-cut problem, *Ann. Oper. Res.*, **248** (2017), 365–403.

[12] R. Martí, A. Duarte, and M. Laguna, Advanced scatter search for the max-cut problem, *INFORMS J. Comput.*, **21** (2009), 26–38.

[14] G. Ottaviano, Spectral Approximation Algorithms for Graph Cut Problems, Master's Thesis, Università di Pisa, 2008.

[15] G. Palubeckis and V. Krivickiene, Application of multistart tabu search to the max-cut problem, *Inf. Technol. Control.*, **31** (2004), 29–35.

[16] S. Poljak and F. Rendl, Solving the max-cut problem using eigenvalues, *Discrete Appl. Math.*, **62** (1995), 249–278.

[17] S. Schaible and T. Ibaraki, Fractional programming, *European J. Oper. Res.*, **12**:4 (1983), 325–338.

[18] L. Trevisan, Max cut and the smallest eigenvalue, *SIAM J. Comput.*, **41** (2012), 1769–1786.