A THRESHOLD-SEARCH PROJECTED GRADIENT METHOD FOR GRAPH-CUT OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many graph-cut clustering models can be formulated as the minimization of differentiable functions over one-hot assignment matrices. Projected gradient (PG) methods provide a natural first-order approach for such problems. Their usual stepsize strategies, however, may lead to stagnation due to the discreteness of the feasible set: once the stepsize is sufficiently small, the projection returns the current assignment matrix unchanged. To address this issue, we characterize the threshold structure of the projection mapping with respect to the projection parameter. We show that it is piecewise constant and can change only at a finite set of explicitly computable row-wise thresholds. Building on this structure, we propose the Threshold-Search Projected Gradient (TSPG) method, which searches directly over this finite set of thresholds rather than continuously adjusting the stepsize. TSPG accepts successive trial moves that satisfy sufficient decrease and terminates after finitely many outer iterations. For graph-cut objectives, we further derive incremental updates that reduce the cost of objective and gradient evaluations. Experimental results demonstrate the effectiveness and computational efficiency of TSPG compared with existing graph-cut solvers.

1 INTRODUCTION

Clustering is a fundamental task in unsupervised learning with a wide range of applications. Over the past decades, numerous clustering methods have been developed, including K-means (MacQueen, 1967), hierarchical clustering (Johnson, 1967), and spectral clustering (Yu & Shi, 2003). Graph-cut clustering is particularly useful for capturing cluster structures through pairwise similarities, without requiring clusters to be characterized by centroids (Shi & Malik, 2000; Ng et al., 2002; von Luxburg, 2007). It represents data points as vertices of a weighted graph and seeks partitions with strong within-cluster connections and weak inter-cluster connections. For representative applications of graph-cut methods, see, e.g., (Chen et al., 2017; Wang et al., 2023). Two classical criteria are Min-Max cut (MMcut) (Ding et al., 2001; Nie et al., 2010) and normalized cut (Ncut) (Shi & Malik, 2000). They normalize inter-cluster similarity by within-cluster similarity and cluster volume, respectively, to discourage highly unbalanced partitions. Both criteria can be formulated over one-hot clustering-assignment matrices that assign each vertex to exactly one cluster.

Directly optimizing these graph-cut objectives over one-hot assignment matrices is challenging, since the objectives are generally nonconvex and the feasible set of one-hot assignments is discrete. Many spectral clustering methods address this issue by relaxing the discrete constraints and computing continuous embeddings, followed by a discretization step to obtain cluster assignments (Ng et al., 2002; Yu & Shi, 2003; Huang et al., 2013). However, these continuous embeddings are not constrained to satisfy the original one-hot constraints, and the discretization step does not directly optimize the original graph-cut objectives (Chen et al., 2018; Nie et al., 2024a).

To optimize graph-cut objectives while retaining one-hot assignments, recent studies have developed direct optimization approaches (Nie et al., 2024a; Chen et al., 2024; Li et al., 2026). These approaches, however, typically rely on specialized optimization schemes rather than a general projected first-order framework based on objective gradients. Projected gradient (PG) methods provide a simple and natural alternative: they exploit the objective gradients and generate feasible iterates by projecting onto the one-hot assignment set. This motivates us to directly apply PG to graph-cut optimization. The remaining challenge is to select an appropriate stepsize.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In continuous constrained optimization, using a sufficiently small stepsize or progressively reducing the trial stepsize through a line search is a standard strategy for obtaining a projected point that satisfies an appropriate line-search condition (Bertsekas, 1976; Calamai & Moré, 1987; Birgin et al., 2000). Over the discrete one-hot assignment set, however, a sufficiently small stepsize makes the projection return the current assignment exactly. Consequently, a conventional PG iteration may stagnate even when its line-search condition is satisfied.

Our key observation is that the projected-gradient update over the one-hot assignment set admits a finite threshold structure. Hence, we develop a threshold-search strategy that explores the finite set of projected points generated by these thresholds instead of continuously adjusting the stepsize.

Our main contributions are summarized as follows.

- **Threshold structure of projections onto one-hot assignment sets.** We characterize the Euclidean projection onto the one-hot assignment set and reveal its finite threshold structure with respect to the projection parameter. Specifically, the projected point changes only at a finite set of row-wise thresholds.
- **Threshold-search projected gradient method.** Building on the threshold structure, we develop the Threshold-Search Projected Gradient (TSPG) method, which searches directly over these projection thresholds instead of continuously adjusting the stepsize parameter. The resulting method generates feasible graph partitions, satisfies a sufficient decrease condition, and admits finite-termination guarantees.
- **Incremental implementation for graph-cut objectives.** We develop an incremental update scheme for TSPG by exploiting quadratic terms in graph-cut objectives.

Notation. For a positive integer m, let $\mathbb{Z}_m^+ := \{1, \dots, m\}$ and let $\mathbf{1}_m \in \mathbb{R}^m$ denote the all-ones vector. For a matrix $X \in \mathbb{R}^{N\times K}$, X_{ij} denotes the entry in its i-th row and j-th column, while $X_k \in \mathbb{R}^N$ denotes its k-th column. When the dimension is clear from the context, e_i denotes the i-th standard basis vector of the corresponding Euclidean space. For $\tau \in \mathbb{Z}_N^+$ and $\iota \in \mathbb{Z}_K^+$, $X_{\tau\to\iota}$ denotes the matrix obtained from X by replacing its τ-th row with $e_\iota^\top \in \mathbb{R}^{1\times K}$, while leaving all other rows unchanged. For matrices of the same size, $\langle X, Y\rangle := \mathrm{tr}(X^\top Y)$ and $\|X\|_F := \sqrt{\langle X, X\rangle}$ denote the Frobenius inner product and norm, respectively, and $\|X\|_\infty := \max_i \sum_j |X_{ij}|$ denotes the matrix infinity norm. For a nonempty closed set $\mathcal{S}$, $\mathrm{Proj}_{\mathcal{S}}(Y) := \arg\min_{Z\in\mathcal{S}} \frac{1}{2}\|Z - Y\|_F^2$ denotes the Euclidean projection of Y onto $\mathcal{S}$, which may be set-valued when $\mathcal{S}$ is nonconvex.

2 Clustering with one-hot assignment matrices

Many clustering problems can be formulated directly in terms of assignments of data samples to clusters. We first introduce the one-hot assignment-matrix representation through several classical graph-cut criteria, and then discuss projected gradient optimization over such matrices.

2.1 Graph-cut models

Suppose that a dataset consists of N data points, indexed by $\mathbb{Z}_N^+$. We consider clustering into $K \geq 2$ clusters and represent the data points as the vertices of a weighted undirected graph with a symmetric similarity matrix $W = [w_{ij}] \in \mathbb{R}^{N\times N}$, where $w_{ij} \geq 0$. Let $d_i := \sum_{j=1}^N w_{ij}$ be the degree of vertex i and $d := (d_1, \dots, d_N)^\top$. For $\mathcal{A}, \mathcal{B} \subseteq \mathbb{Z}_N^+$, their total similarity is defined as $s(\mathcal{A}, \mathcal{B}) := \sum_{i\in\mathcal{A}} \sum_{j\in\mathcal{B}} w_{ij}$, while the volume of $\mathcal{A}$ is defined as $\mathrm{vol}(\mathcal{A}) := \sum_{i\in\mathcal{A}} d_i$. A clustering assignment is represented by a *one-hot assignment matrix*. We define the set of such matrices as

$$\mathcal{I}^{N\times K} := \{X \in \{0,1\}^{N\times K} : X\mathbf{1}_K = \mathbf{1}_N\}. \tag{1}$$

Thus, every row of $X \in \mathcal{I}^{N\times K}$ contains exactly one entry equal to one, indicating the cluster assignment of the corresponding sample. For $X \in \mathcal{I}^{N\times K}$, let $\mathcal{A}_k := \{i \in \mathbb{Z}_N^+ : X_{ik} = 1\}$. Then we have $s(\mathcal{A}_k, \mathcal{A}_k) = X_k^\top W X_k$ and $\mathrm{vol}(\mathcal{A}_k) = d^\top X_k$. Moreover, letting $\mathcal{A}_k^\complement := \mathbb{Z}_N^+ \setminus \mathcal{A}_k$ denote the complement of $\mathcal{A}_k$, we have $s(\mathcal{A}_k, \mathcal{A}_k^\complement) = X_k^\top(-W)X_k + d^\top X_k$.

Graph clustering seeks partitions with weak connections across different clusters. Directly minimizing the total inter-cluster similarity $\sum_{k=1}^{K} s(\mathcal{A}_k, \mathcal{A}_k^{\complement})$, however, tends to produce highly unbalanced partitions. MMcut and Ncut address this issue by normalizing $s(\mathcal{A}_k, \mathcal{A}_k^{\complement})$ by the within-cluster similarity $s(\mathcal{A}_k, \mathcal{A}_k)$ and the cluster volume $\mathrm{vol}(\mathcal{A}_k)$, respectively. Using the identities above and removing additive constants that do not affect the minimizers, the corresponding objective functions can be expressed as

$$\Phi_{\mathrm{MMcut}}(X) := \sum_{k=1}^{K} \frac{d^\top X_k}{X_k^\top W X_k}, \qquad \Phi_{\mathrm{Ncut}}(X) := \sum_{k=1}^{K} \frac{X_k^\top (-W) X_k}{d^\top X_k}. \tag{2}$$

Thus, both graph-cut problems can be written in the unified form

$$\operatorname*{arg\,min}_{X \in \mathcal{I}^{N\times K}} \Phi(X), \tag{3}$$

where Φ is either of the objective functions in (2).

2.2 Projected gradient over one-hot assignment matrices

For a continuously differentiable objective Φ, given a feasible point $\bar{X} \in \mathcal{I}^{N\times K}$, the projected gradient (PG) update for (3) generates a new trial $\hat{X}(\alpha)$ from

$$\mathrm{Proj}_{\mathcal{I}^{N\times K}}\big(\bar{X} - \alpha V\big), \tag{4}$$

where $V = \nabla\Phi(\bar{X})$, and the stepsize $\alpha > 0$. In continuous constrained optimization, using a sufficiently small stepsize or progressively reducing the trial stepsize through a line-search procedure is a standard strategy for seeking a trial point $\hat{X}(\alpha)$ satisfying a sufficient-decrease condition (Bertsekas, 1999; Nocedal & Wright, 2006). One such condition is (Beck, 2017; Bolte et al., 2014):

$$\Phi(\hat{X}(\alpha)) + \frac{c}{2}\|\hat{X}(\alpha) - \bar{X}\|_F^2 \le \Phi(\bar{X}), \qquad c > 0. \tag{5}$$

Over the discrete set $\mathcal{I}^{N\times K}$ of one-hot assignment matrices, however, an excessively small stepsize may make the projected update uninformative: the projection can return the current assignment matrix itself. The following proposition makes this phenomenon precise.

Proposition 1. *Let $\bar{X} \in \mathcal{I}^{N\times K}$ and $V \in \mathbb{R}^{N\times K}$. Then there exists $\bar{\alpha} > 0$ such that*

$$\mathrm{Proj}_{\mathcal{I}^{N\times K}}(\bar{X} - \alpha V) = \{\bar{X}\}, \qquad \forall\, \alpha \in (0, \bar{\alpha}). \tag{6}$$

Here, any $\bar{\alpha} > 0$ is valid when $V = 0$, while $\bar{\alpha} = 1/(\sqrt{2}\|V\|_F)$ suffices when $V \neq 0$.

The proof is given in Appendix A. Taking $V = \nabla\Phi(\bar{X})$ shows that sufficiently small stepsizes always make the PG update remain at $\bar{X}$. Thus, progressively shrinking the stepsize in search of sufficient decrease may eventually cause the PG iteration to stagnate, even though the acceptance condition (5) is satisfied. This observation motivates examining how the projected point changes as the projection parameter varies.

3 Threshold-search projected optimization

Building on the observation at the end of Section 2.2, we first study the threshold structure of the Euclidean projection onto $\mathcal{I}^{N\times K}$. This characterization is then used to construct and analyze our Threshold-Search Projected Gradient (TSPG) method for problem (3).

3.1 Threshold structure of parameterized projections

We first characterize how the projected-gradient update changes with respect to the stepsize. For this purpose, we introduce some notation for the Euclidean projection onto $\mathcal{I}^{N\times K}$. For $X \in \mathbb{R}^{N\times K}$, define the row-wise maximizer mapping by

$$\mathcal{M}(X) := \mathcal{M}_1(X) \times \cdots \times \mathcal{M}_N(X), \qquad \mathcal{M}_i(X) := \operatorname*{arg\,max}_{j \in \mathbb{Z}_K^+} X_{ij}, \qquad i \in \mathbb{Z}_N^+.$$

For $\ell = (\ell_1, \ldots, \ell_N)^\top \in (\mathbb{Z}_K^+)^N$, let $\mathcal{E}(\ell) \in \mathcal{I}^{N\times K}$ denote its one-hot encoding, i.e., $(\mathcal{E}(\ell))_{ij} = 1$ if $j = \ell_i$ and 0 otherwise. Since the projection onto $\mathcal{I}^{N\times K}$ is separable across rows, we have

$$\mathrm{Proj}_{\mathcal{I}^{N\times K}}(X) = \{\mathcal{E}(\ell) : \ell \in \mathcal{M}(X)\}. \tag{7}$$

In particular, positive scaling does not change the row-wise maximizers, and therefore

$$\mathrm{Proj}_{\mathcal{I}^{N\times K}}(\alpha X) = \mathrm{Proj}_{\mathcal{I}^{N\times K}}(X), \qquad \forall \alpha > 0. \tag{8}$$

We next study the parameterized projection mapping

$$\rho \longmapsto \mathrm{Proj}_{\mathcal{I}^{N\times K}}(\rho\bar{X} - V), \tag{9}$$

where $\bar{X} \in \mathcal{I}^{N\times K}$ and $V \in \mathbb{R}^{N\times K}$ are fixed. Since each row of $\bar{X}$ contains one entry equal to one, the associated label vector is uniquely determined. Let $\bar{\ell} = (\bar{\ell}_1, \ldots, \bar{\ell}_N)^\top$ denote this label vector, i.e., $\mathcal{E}(\bar{\ell}) = \bar{X}$. Then, for each $i \in \mathbb{Z}_N^+$, the entries of the i-th row of $\rho\bar{X} - V$ are given by

$$(\rho\bar{X} - V)_{ij} = \begin{cases} \rho - V_{i\bar{\ell}_i}, & j = \bar{\ell}_i, \\ -V_{ij}, & j \neq \bar{\ell}_i, \end{cases} \qquad j \in \mathbb{Z}_K^+. \tag{10}$$

Hence, the maximizing index of the i-th row changes only at

$$\bar{\rho}_i := V_{i\bar{\ell}_i} - \min_{j\neq\bar{\ell}_i} V_{ij}, \qquad i \in \mathbb{Z}_N^+. \tag{11}$$

More precisely, we have

$$\mathcal{M}_i(\rho\bar{X} - V) = \begin{cases} \{\bar{\ell}_i\}, & \rho > \bar{\rho}_i, \\ \{\bar{\ell}_i\} \cup \arg\min\limits_{j\neq\bar{\ell}_i} V_{ij}, & \rho = \bar{\rho}_i, \\ \arg\min\limits_{j\neq\bar{\ell}_i} V_{ij}, & \rho < \bar{\rho}_i. \end{cases} \tag{12}$$

Combining (7) with (12) gives the following proposition.

Proposition 2. *Let $\bar{X} \in \mathcal{I}^{N\times K}$, $V \in \mathbb{R}^{N\times K}$, and $\rho \in \mathbb{R}$. Let $\bar{\ell}$ be the unique element of $\mathcal{M}(\bar{X})$, and let $\{\bar{\rho}_i\}_{i=1}^N$ be defined by* (11)*. For each $i \in \mathbb{Z}_N^+$, choose*

$$\widetilde{\ell}_i \in \arg\min_{j\neq\bar{\ell}_i} V_{ij}, \tag{13}$$

and define $\widetilde{\ell} := (\widetilde{\ell}_1, \ldots, \widetilde{\ell}_N)^\top$. If $\widehat{\ell} = (\widehat{\ell}_1, \ldots, \widehat{\ell}_N)^\top$ satisfies

$$\widehat{\ell}_i \in \begin{cases} \{\bar{\ell}_i\}, & \rho > \bar{\rho}_i, \\ \{\bar{\ell}_i, \widetilde{\ell}_i\}, & \rho = \bar{\rho}_i, \\ \{\widetilde{\ell}_i\}, & \rho < \bar{\rho}_i, \end{cases} \qquad i \in \mathbb{Z}_N^+, \tag{14}$$

then $\mathcal{E}(\widehat{\ell}) \in \mathrm{Proj}_{\mathcal{I}^{N\times K}}(\rho\bar{X} - V)$. When the minimizer in (13) *is not unique, any minimizer yields an element of the projection set.*

Proposition 2 reveals that the projection mapping (9) has a simple row-wise threshold structure. For fixed $\bar{X}$ and V, let $\bar{\ell}$ denote the current label vector associated with $\bar{X}$, and let $\widetilde{\ell}$ be the label vector defined in (13), where $\widetilde{\ell}_i$ is the candidate label that minimizes V_{ij} among all $j \neq \bar{\ell}_i$. Then, as ρ decreases, the label of row i remains $\bar{\ell}_i$ for $\rho > \bar{\rho}_i$ and switches to $\widetilde{\ell}_i$ for $\rho < \bar{\rho}_i$, with both choices possible at $\rho = \bar{\rho}_i$. Consequently, once the candidate labels in (13) are fixed, the projected point is piecewise constant in ρ and can change only at the finite set of thresholds $\{\bar{\rho}_i\}_{i=1}^N$ defined in (11).

The above characterization directly describes how the projected gradient update (4) changes as the stepsize α varies. By the positive-scaling invariance in (8), the update in (4) is equivalent to evaluating the parameterized projection mapping (9) with $\rho = \alpha^{-1}$. Therefore, the threshold structure shows that continuously adjusting the PG stepsize in (4) cannot generate new projected assignment matrices between two consecutive thresholds. Instead, it is sufficient to examine the finite collection of distinct projected assignment matrices generated by these thresholds. This observation motivates the threshold-search strategy developed in the next subsection.

3.2 THRESHOLD-SEARCH PROJECTED GRADIENT METHOD

We now exploit the threshold structure established in Proposition 2 to develop our Threshold-Search Projected Gradient (TSPG) method for problem (3), where Φ is differentiable. Starting from a feasible point $X^{(0)} \in \mathcal{I}^{N\times K}$ and its associated label vector $\ell^{(0)} \in \mathcal{M}(X^{(0)})$, the method generates a sequence of feasible iterates by computing the thresholds at each iteration, examining the corresponding trial points in descending order, and accepting one that yields a sufficient decrease in the objective value. Throughout the iterations, we maintain the equivalent representations

$$\ell^{(t)} \in \mathcal{M}(X^{(t)}), \qquad X^{(t)} = \mathcal{E}(\ell^{(t)}).$$

For the t-th iteration, we first compute $V := \nabla\Phi(\bar{X})$, and write $\bar{\ell} := \ell^{(t)}$ and $\bar{X} := X^{(t)}$. Choose $\widetilde{\ell}_i$ according to (13), and compute the corresponding thresholds $\{\bar{\rho}_i\}_{i=1}^N$ from (11). Let $\pi_1, \ldots, \pi_N$ be any permutation of $\mathbb{Z}_N^+$ such that

$$\bar{\rho}_{\pi_1} \geq \bar{\rho}_{\pi_2} \geq \cdots \geq \bar{\rho}_{\pi_N}. \tag{15}$$

This ordering corresponds to decreasing the projection parameter ρ across the breakpoints of the projection mapping (9). Using this ordering, we perform an inner threshold-search procedure. Starting from $\widehat{\ell}^{[0]} := \bar{\ell}$ and $\widehat{X}^{[0]} := \bar{X}$, we construct a sequence of trial labels $\{\widehat{\ell}^{[i]}\}_{i=0}^N$ by successively switching their components following the threshold structure established in Proposition 2. The resulting sequences of trial labels and their corresponding iterates are defined by

$$\widehat{\ell}_j^{[i]} := \begin{cases} \widetilde{\ell}_j, & j = \pi_i, \\ \widehat{\ell}_j^{[i-1]}, & j \neq \pi_i, \end{cases} \qquad \widehat{X}^{[i]} := \mathcal{E}(\widehat{\ell}^{[i]}). \tag{16}$$

Following the sufficient-decrease condition (5), we accept a successive trial move only if

$$\Phi(\widehat{X}^{[i]}) + \frac{c}{2}\|\widehat{X}^{[i]} - \widehat{X}^{[i-1]}\|_F^2 \leq \Phi(\widehat{X}^{[i-1]}), \tag{17}$$

where $c > 0$ is fixed. To accommodate objectives that are not defined on all of $\mathcal{I}^{N\times K}$, let $\mathrm{dom}(\Phi)$ denote the domain of Φ. For TSPG, we therefore require $X^{(0)} \in \mathcal{I}^{N\times K} \cap \mathrm{dom}(\Phi)$ and regard any trial point outside $\mathrm{dom}(\Phi)$ as failing (17), without evaluating Φ there. Consequently, every accepted iterate remains in $\mathrm{dom}(\Phi)$. The inner search terminates as soon as a trial point fails (17), and the last accepted trial point is taken as the next outer iterate. Algorithm 1 summarizes TSPG.

PG interpretation. We next relate the threshold search in TSPG to PG updates. Fix a sufficiently large $\lambda > 0$ and consider the shifted problem

$$\operatorname*{arg\,min}_{X\in\mathcal{I}^{N\times K}} \Phi(X) + \frac{\lambda}{2}\|X\|_F^2. \tag{18}$$

Since $\|X\|_F^2 = N$ for every $X \in \mathcal{I}^{N\times K}$, (18) is equivalent to (3). At $\bar{X}$, with $V = \nabla\Phi(\bar{X})$, a PG update for (18) is obtained from

$$\mathrm{Proj}_{\mathcal{I}^{N\times K}}\big((1-\beta\lambda)\bar{X} - \beta V\big), \qquad \beta > 0. \tag{19}$$

For any $\alpha > 0$, setting $\beta = \alpha/(1+\alpha\lambda)$ and using (8) reduces (19) exactly to the classical PG update (4). Hence, varying $\alpha \in (0, +\infty)$ in (4) is equivalent to varying $\beta \in (0, 1/\lambda)$ in (19).

Now consider the t-th outer iteration. By Proposition 2 and (16), we know that $\widehat{X}^{[i]}$ belongs to $\mathrm{Proj}_{\mathcal{I}^{N\times K}}(\bar{\rho}_{\pi_i}\bar{X} - V)$. By (11), we have $|\bar{\rho}_j| \leq \|V\|_\infty$ for all j; hence, taking $\lambda > \|V\|_\infty$ ensures $\bar{\rho}_j + \lambda > 0$. Thus, with $\bar{\beta}_i := (\bar{\rho}_{\pi_i} + \lambda)^{-1} > 0$, the positive-scaling invariance in (8) gives

$$\widehat{X}^{[i]} \in \mathrm{Proj}_{\mathcal{I}^{N\times K}}(\bar{\rho}_{\pi_i}\bar{X} - V) = \mathrm{Proj}_{\mathcal{I}^{N\times K}}\big((1-\bar{\beta}_i\lambda)\bar{X} - \bar{\beta}_i V\big). \tag{20}$$

Hence, every trial point generated by (16) is a PG update of (18) at a breakpoint stepsize $\bar{\beta}_i$. Moreover, the descending threshold order (15) gives $0 < \bar{\beta}_1 \leq \cdots \leq \bar{\beta}_N$. Thus, the inner threshold search successively examines the PG breakpoint stepsizes of the shifted problem from smaller to larger values. In contrast to the classical update (4), whose stepsize varies over $\alpha \in (0, +\infty)$ and thus corresponds to $\beta \in (0, 1/\lambda)$ in (19), the shifted formulation allows the broader range $\beta > 0$. Our method does not search this range continuously; instead, it tests only the finitely many breakpoint stepsizes at which the projected point can change, progressively considering larger ones as long as the successive trial points satisfy (17).

Algorithm 1 Threshold-Search Projected Gradient Method (TSPG)

Require: $X^{(0)} \in \mathcal{I}^{N\times K}$, and $c > 0$.
Choose $\ell^{(0)} \in \mathcal{M}(X^{(0)})$.
for $t = 0, 1, 2, \ldots$ **do**
  Set $\bar{\ell} \leftarrow \ell^{(t)}$, $\bar{X} \leftarrow X^{(t)}$, and $V \leftarrow \nabla\Phi(\bar{X})$.
  Compute $\widetilde{\ell}$, $\{\bar{\rho}_i\}_{i=1}^N$, and $\{\pi_i\}_{i=1}^N$ according to (13), (11), and (15).
  Set $(\widehat{\ell}^{[0]}, \widehat{X}^{[0]}, m_t) \leftarrow (\bar{\ell}, \bar{X}, 0)$.
  for $i = 1, \ldots, N$ **do**
    Construct $\widehat{\ell}^{[i]}$ and $\widehat{X}^{[i]}$ according to (16).
    if (17) holds, **set** $m_t \leftarrow i$; **otherwise, break**.
  end for
  If $m_t > 0$, **set** $(\ell^{(t+1)}, X^{(t+1)}) \leftarrow (\widehat{\ell}^{[m_t]}, \widehat{X}^{[m_t]})$; **otherwise, stop**.
end for

3.3 Theoretical properties and computational complexity

For a one-hot assignment matrix $X \in \mathcal{I}^{N\times K}$, its *one-flip neighborhood* is defined by $\mathcal{N}_1(X) := \{X_{\tau\to\iota} : \tau \in \mathbb{Z}_N^+,\ \iota \in \mathbb{Z}_K^+\}$, and $\mathring{\mathcal{N}}_1(X) := \mathcal{N}_1(X) \setminus \{X\}$ denotes the nontrivial part. We refer to the transition from X to any $Y \in \mathring{\mathcal{N}}_1(X)$ as a *one-flip move*, which reassigns exactly one vertex to a different cluster. The following theorem summarizes the descent and finite-termination properties of TSPG; its proof is deferred to Appendix B.

Theorem 1. *Let Φ be differentiable on its domain $\mathrm{dom}(\Phi)$, $X^{(0)} \in \mathcal{I}^{N\times K} \cap \mathrm{dom}(\Phi)$, and $c > 0$. Then TSPG has the following properties.*

(i) *At the t-th outer iteration, the trial sequence $\{\widehat{X}^{[i]}\}_{i=0}^N$ satisfies*
$$\widehat{X}^{[i]} = \big(\widehat{X}^{[i-1]}\big)_{\pi_i\to\widetilde{\ell}_{\pi_i}} \in \mathring{\mathcal{N}}_1\big(\widehat{X}^{[i-1]}\big) \subseteq \mathcal{I}^{N\times K}, \qquad i \in \mathbb{Z}_N^+.$$

(ii) *If $m_t \geq 1$ at the t-th outer iteration, then* $\Phi(X^{(t+1)}) + m_t c \leq \Phi(X^{(t)})$.

(iii) *Let $\Phi_{\min} := \min\{\Phi(X) : X \in \mathcal{I}^{N\times K} \cap \mathrm{dom}(\Phi)\}$. If T denotes the number of nontrivial outer updates before termination, then*
$$T \leq \sum_{t=0}^{T-1} m_t \leq \left\lfloor \left(\Phi(X^{(0)}) - \Phi_{\min}\right)/c \right\rfloor.$$
Consequently, TSPG terminates after finitely many outer iterations.

One-flip local optimality. Neighborhood-based local optimality is widely used in combinatorial optimization, where local optimality is defined with respect to a prescribed set of neighboring solutions (Schäffer & Yannakakis, 1991; Poljak, 1995). For the assignment matrices considered here, we focus on the one-flip neighborhood $\mathcal{N}_1(\bar{X})$. For $c \geq 0$, we say that $\bar{X}$ is *one-flip c-optimal* for Φ if $\Phi(\bar{X}) \leq \Phi(Y) + c$ for all $Y \in \mathcal{N}_1(\bar{X}) \cap \mathrm{dom}(\Phi)$. The case $c = 0$ corresponds to the usual notion of a *one-flip optimum*. To quantify the deviation from first-order behavior over this discrete neighborhood, we define the *one-flip linearization remainder*
$$R_1(\bar{X}) := \max_{Y\in\mathcal{N}_1(\bar{X})\cap\mathrm{dom}(\Phi)} \left|\Phi(Y) - \Phi(\bar{X}) - \left\langle \nabla\Phi(\bar{X}), Y - \bar{X}\right\rangle\right|. \tag{21}$$
This quantity measures the worst-case error of the first-order linearization over the one-flip neighborhood and plays a role analogous to the Taylor remainder term in smooth optimization (Beck, 2017, Sec. 5.1.1). For $c \geq 0$, consider the auxiliary quadratic model
$$Q_c(X \mid \bar{X}) := \Phi(\bar{X}) + \left\langle \nabla\Phi(\bar{X}), X - \bar{X}\right\rangle + \frac{R_1(\bar{X}) + c}{2}\|X - \bar{X}\|_F^2. \tag{22}$$
By (21), $Q_c(Y \mid \bar{X}) \geq \Phi(Y) + c$ for every $Y \in \mathring{\mathcal{N}}_1(\bar{X}) \cap \mathrm{dom}(\Phi)$, whereas $Q_c(\bar{X} \mid \bar{X}) = \Phi(\bar{X})$. Consequently, if $\bar{X}$ is one-flip c-optimal for Φ, then $\bar{X}$ is a one-flip optimum of $Q_c(\cdot \mid \bar{X})$ over $\mathcal{N}_1(\bar{X}) \cap \mathrm{dom}(\Phi)$. The next theorem shows that a terminal point of TSPG satisfies this auxiliary one-flip optimality directly; its proof is deferred to Appendix C.

Theorem 2. *Let $\Phi : \mathrm{dom}(\Phi) \to \mathbb{R}$ be differentiable, $X^{(0)} \in \mathcal{I}^{N\times K} \cap \mathrm{dom}(\Phi)$, and $c > 0$. If TSPG terminates at $\bar{X}$ with $\widehat{X}^{[1]} \in \mathrm{dom}(\Phi)$, then $\bar{X}$ is a one-flip optimum of $Q_c(\cdot \mid \bar{X})$.*

Computational complexity. At the t-th outer iteration of TSPG, evaluating the gradient $V = \nabla\Phi(\bar{X})$ incurs a cost of $C_{\nabla\Phi}$, while computing and sorting the N row-wise thresholds costs $O(NK + N\log N)$. Since the inner search checks at most $\min\{m_t + 1, N\}$ trial points, and C_{trial} denotes the cost of checking one trial point, the total cost of this outer iteration is

$$O(C_{\nabla\Phi} + NK + N\log N + \min\{m_t + 1, N\}C_{\mathrm{trial}})\,.$$

In the next subsection, we exploit the quadratic terms in the graph-cut objectives to reduce the costs associated with $C_{\nabla\Phi}$ and C_{trial} through incremental updates.

3.4 Incremental updates for graph-cut objectives

For the graph-cut objectives in (2), the objective function is separable with respect to the columns of X and can be written as $\Phi(X) = \sum_{k=1}^{K}\varphi(X_k)$, where φ is composed of quadratic functions. Consider, therefore, a quadratic function $h(x) := x^\top Qx + q^\top x + r$, where $Q \in \mathbb{R}^{N\times N}$ is symmetric. Let $X \in \mathcal{I}^{N\times K}$ and suppose that $X_{\tau\theta} = 1$. For $\iota \neq \theta$, set $\widehat{Z} := X_{\tau\to\iota}$. Then the collections $\{h(\widehat{Z}_k)\}_{k=1}^K$ and $\{\nabla h(\widehat{Z}_k)\}_{k=1}^K$ coincide with $\{h(X_k)\}_{k=1}^K$ and $\{\nabla h(X_k)\}_{k=1}^K$, respectively, except for their θ-th and ι-th entries, which satisfy

$$\begin{aligned} h(\widehat{Z}_\theta) &= h(X_\theta) - (\nabla h(X_\theta))_\tau + Q_{\tau\tau}, \qquad & \nabla h(\widehat{Z}_\theta) &= \nabla h(X_\theta) - 2Q_\tau, \\ h(\widehat{Z}_\iota) &= h(X_\iota) + (\nabla h(X_\iota))_\tau + Q_{\tau\tau}, \qquad & \nabla h(\widehat{Z}_\iota) &= \nabla h(X_\iota) + 2Q_\tau. \end{aligned} \tag{23}$$

Hence, once the function values and gradients of the quadratic functions involved in φ are maintained for all columns, a one-flip move updates the corresponding function values in $O(1)$ time and their gradients in $O(N)$ time.

By Theorem 1(i), successive inner-search trial points differ by a single one-flip move. Hence, $\Phi(\widehat{X}^{[i]})$ can be evaluated incrementally from the quadratic quantities for $\widehat{X}^{[i-1]}$. The corresponding quadratic gradients are updated only if $\widehat{X}^{[i]}$ is accepted by (17). Detailed incremental formulas for Φ are given in Appendix D. Moreover, $X^{(t)}$ and $\widehat{X}^{[i]}$ need not be formed explicitly; the label vectors and required quadratic quantities suffice.

We next refine the complexity bound derived at the end of Subsection 3.3. For the graph-cut objectives considered in (2), the gradient $V = \nabla\Phi(\bar{X})$ can be evaluated from the maintained quantities in $O(NK)$ operations, while computing and sorting the row-wise thresholds still costs $O(NK + N\log N)$. During the inner search, evaluating each of the at most $\min\{m_t + 1, N\}$ trial points costs $O(1)$, and each of the m_t accepted one-flip moves requires an additional $O(N)$ update of the maintained gradients. Therefore, the complexity of the t-th outer iteration is

$$O(NK + N\log N + m_t N)\,.$$

4 Experiments

We evaluate the proposed TSPG method on the Ncut and MMcut models over benchmark datasets. The sufficient-decrease parameter of TSPG is fixed at $c = 10^{-9}$ throughout all experiments. All experiments are conducted in MATLAB R2025b on a desktop equipped with an Intel Core i9-13900K CPU at 3.00 GHz and 32 GB of RAM.

Compared methods. For both MMcut and Ncut, we include the coordinate descent (CD) methods proposed in (Nie et al., 2024b; Li et al., 2026). Coordinate descent (Beck & Tetruashvili, 2013; Wright, 2015) updates one coordinate block while fixing the remaining variables. For the graph-cut problems considered here, each CD step updates one row of the assignment matrix by selecting the cluster that gives the best objective value. The implementations in (Nie et al., 2024b; Li et al., 2026) further exploit incremental updates to reduce the computational cost.

For MMcut, we additionally compare with the unrefined Mcut method (Ding et al., 2001). For a binary split, the unrefined Mcut computes the eigenvector associated with the second-largest eigenvalue of $D^{-1/2}WD^{-1/2}$, where D is the diagonal degree matrix with $D_{ii} = d_i$ and d is the degree vector introduced in Section 2.1. The samples are then ordered according to the entries of the computed eigenvector, and a cut point is selected along this ordering. We use MATLAB's `eigs` function to compute this eigenvector. For multi-class clustering, we recursively apply the binary splitting procedure to the largest cluster until the desired number of clusters is obtained.

For Ncut, we include the fractional programming-based clustering (FPC) method (Chen et al., 2024). Building on the quadratic transform (Shen & Yu, 2018), FPC transforms the fractional Ncut problem into some tractable subproblems. We use the implementation publicly released by the authors.[1]

Experimental setup. We consider eight datasets from the UCI Machine Learning Repository.[2] With reference to the experimental setup in Chen et al. (2024), we normalize each feature dimension, and construct the similarity matrix by $w_{ij} = \exp(-\|v_i - v_j\|_2)$, where v_i and v_j denote the feature vectors of samples i and j, respectively. For each dataset, we perform 20 runs with randomly generated initial assignment matrices $X^{(0)}$ with all clusters nonempty.

Evaluation metrics. We evaluate the compared methods from two perspectives. For computational efficiency, we report the average running time in seconds (Avg-Time), the assignment evaluation measure (Avg-AssignE), and the average objective value (Avg-Obj) over the 20 runs. Since different methods employ different update mechanisms, the notion of an iteration is not directly comparable across methods. For CD and FPC, Avg-AssignE is measured by the number of outer iterations, since each CD iteration performs a complete sweep over all samples and each FPC iteration evaluates a newly generated assignment matrix. For TSPG, each evaluated trial point in the inner search corresponds to a one-flip move, and Avg-AssignE is computed as $\sum_t \min\{m_t + 1, N\}/N$, where m_t denotes the number of accepted one-flip moves in the t-th outer iteration and N is the number of samples. The Avg-AssignE value is omitted for unrefined Mcut, whose cost is dominated by eigenvector computation rather than assignment evaluations. For clustering quality, we report clustering accuracy (ACC), normalized mutual information (NMI), and adjusted Rand index (ARI). These metrics are obtained from the run with the minimum objective value.

Experimental results. Table 1 summarizes the experimental results. The first two columns report the datasets and evaluated methods, where the prefixes "MM-" and "N-" indicate that the corresponding methods are applied to the MMcut and Ncut models, respectively. The remaining columns report the evaluation metrics described above.

For MMcut, the proposed TSPG method achieves running times comparable to the unrefined Mcut method and shorter than the iterative CD solver. Meanwhile, TSPG obtains objective values and clustering performance comparable to CD, while achieving lower objective values and competitive clustering performance compared with Mcut.

For Ncut, compared with the iterative CD and FPC solvers, TSPG achieves shorter running times while obtaining comparable objective values and clustering performance.

The Avg-AssignE results further show that TSPG requires fewer assignment evaluations than the other iterative solvers on the tested datasets for both graph-cut models. This can be attributed to the combination of gradient information and the projection threshold structure: instead of sequentially examining all samples as in CD or recomputing a complete assignment matrix at each iteration as in FPC, TSPG generates candidate trial assignments by exploring the threshold structure of the projection mapping. Furthermore, since consecutive trial assignments differ by only one sample reassignment, the incremental update strategy described in Section 3.4 further reduces the cost of evaluating each trial point, which contributes to the observed running-time performance of TSPG.

Experimental results on eight datasets show lower average running times than the tested graph-cut solvers, while maintaining competitive objective values and clustering performance.

[1] `https://github.com/zhanchendao/FPC`

[2] `https://archive.ics.uci.edu/datasets`

Table 1: Experimental results on eight datasets.

Dataset	Algorithm	Avg-Time	Avg-AssignE	Avg-Obj	ACC	NMI	ARI
CTG_NSP	MM-TSPG	0.054	**0.88**	**8.887**	**0.498**	0.130	0.074
$N = 2126$	MM-CD	0.428	112.65	**8.887**	**0.498**	**0.131**	**0.075**
$K = 3$	MM-Mcut	**0.045**	–	9.868	0.413	0.083	-0.012
	N-TSPG	**0.042**	**0.92**	**-1.014**	**0.618**	**0.149**	**0.200**
	N-CD	0.067	9.40	**-1.014**	**0.618**	**0.149**	**0.200**
	N-FPC	0.119	15.50	**-1.014**	**0.618**	**0.149**	**0.200**
Optdigits	MM-TSPG	0.651	**1.50**	97.941	**0.651**	**0.576**	**0.485**
$N = 5620$	MM-CD	1.041	57.60	**97.909**	0.647	0.571	0.477
$K = 10$	MM-Mcut	**0.576**	–	109.610	0.416	0.371	0.256
	N-TSPG	**0.451**	**1.48**	**-1.023**	0.544	0.523	0.409
	N-CD	0.604	21.75	**-1.023**	**0.578**	**0.544**	0.426
	N-FPC	1.988	24.95	**-1.023**	0.562	0.539	**0.430**
PolishBankruptcy1	MM-TSPG	**0.324**	**0.57**	**3.994**	**0.516**	**0.003**	**0.001**
$N = 7027$	MM-CD	1.203	88.05	**3.994**	0.515	**0.003**	**0.001**
$K = 2$	MM-Mcut	0.414	–	**3.994**	0.511	0.001	0.000
	N-TSPG	**0.301**	**0.65**	**-1.002**	**0.856**	**0.001**	**-0.014**
	N-CD	0.526	20.45	**-1.002**	**0.856**	**0.001**	**-0.014**
	N-FPC	1.081	15.75	**-1.002**	0.841	**0.001**	-0.017
BEED	MM-TSPG	**0.798**	**1.15**	**15.883**	**0.517**	**0.252**	**0.218**
$N = 8000$	MM-CD	6.546	415.80	**15.883**	0.504	0.248	0.212
$K = 4$	MM-Mcut	0.851	–	15.899	0.495	0.233	0.204
	N-TSPG	**0.632**	**1.21**	**-1.011**	0.351	**0.283**	0.163
	N-CD	0.944	25.35	**-1.011**	0.354	**0.283**	0.162
	N-FPC	2.438	23.85	**-1.011**	**0.356**	**0.283**	**0.164**
Phishing	MM-TSPG	**0.775**	**0.56**	**3.989**	**0.711**	**0.134**	**0.178**
$N = 11055$	MM-CD	2.877	122.20	**3.989**	0.709	0.132	0.175
$K = 2$	MM-Mcut	1.308	–	3.990	0.536	0.004	0.005
	N-TSPG	**0.752**	**0.65**	**-1.003**	**0.508**	**0.004**	**-0.002**
	N-CD	0.874	9.45	**-1.003**	0.507	**0.004**	**-0.002**
	N-FPC	1.785	10.35	**-1.003**	**0.508**	**0.004**	**-0.002**
Epileptic	MM-TSPG	**1.533**	**1.11**	**24.623**	0.407	0.234	0.207
$N = 11500$	MM-CD	3.300	104.35	**24.623**	0.408	0.234	**0.208**
$K = 5$	MM-Mcut	1.812	–	27.482	**0.418**	**0.266**	0.180
	N-TSPG	**1.443**	**1.24**	**-1.017**	**0.324**	**0.232**	0.128
	N-CD	2.079	22.70	**-1.017**	**0.324**	**0.232**	**0.129**
	N-FPC	5.163	22.35	**-1.017**	0.323	**0.232**	**0.129**
GasSensorDrift	MM-TSPG	**2.245**	**1.25**	35.033	**0.494**	0.314	0.250
$N = 13910$	MM-CD	9.881	229.90	**35.030**	0.493	**0.322**	**0.251**
$K = 6$	MM-Mcut	2.671	–	38.762	0.359	0.204	0.122
	N-TSPG	**2.304**	**1.41**	**-1.030**	**0.403**	0.218	0.108
	N-CD	2.834	19.80	**-1.030**	0.386	**0.228**	**0.125**
	N-FPC	8.187	22.30	-1.028	0.387	**0.228**	**0.125**
CreditDefault	MM-TSPG	**5.968**	**0.55**	**3.989**	0.558	**0.015**	0.013
$N = 30000$	MM-CD	19.890	306.50	**3.989**	0.558	**0.015**	0.013
$K = 2$	MM-Mcut	30.821	–	**3.989**	**0.559**	**0.015**	**0.014**
	N-TSPG	**5.330**	**0.63**	**-1.003**	**0.552**	**0.007**	**-0.020**
	N-CD	5.502	10.45	**-1.003**	**0.552**	**0.007**	-0.021
	N-FPC	22.021	10.65	**-1.003**	**0.552**	**0.007**	**-0.020**

5 CONCLUSION

We studied the minimization of differentiable functions over one-hot assignment matrices, with graph-cut clustering as an important application. By characterizing the piecewise-constant threshold structure of the Euclidean projection onto this discrete set, we developed the Threshold-Search Projected Gradient (TSPG) method, which searches over the finite set of projection thresholds rather than continuously adjusting the stepsize. We established the sufficient-decrease property and finite-termination guarantee of TSPG, and derived incremental updates that reduce the cost of objective and gradient evaluations for graph-cut objectives with quadratic structures. Experiments demonstrate that TSPG achieves competitive objective values and clustering quality while reducing computational costs compared with existing graph-cut solvers.

Future work includes developing variants of TSPG with stronger termination optimality guarantees and extending the framework to nonsmooth optimization settings.

AI USE STATEMENT

During the preparation of this manuscript, we used generative AI tools only for language editing, improving the clarity and presentation of the paper, and generating simple scripts for converting the file formats of datasets from the UCI Machine Learning Repository before experimental use.

REPRODUCIBILITY STATEMENT

The proposed method and its main algorithmic details are described in the main text. Complete proofs of the theoretical results and detailed incremental-update formulas used in the experiments are provided in the appendix. A demonstration implementation of the proposed method will be released upon publication.

REFERENCES

Amir Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2017.

Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

Dimitri P. Bertsekas. On the Goldstein–Levitin–Polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21(2):174–184, 1976.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 2nd edition, 1999.

Ernesto G. Birgin, José Mario Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

Paul H. Calamai and Jorge J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39(1):93–116, 1987.

Xiaojun Chen, Joshua Zhexue Huang, Feiping Nie, Renjie Chen, and Qingyao Wu. A self-balanced min-cut algorithm for image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2080–2088, 2017.

Xiaojun Chen, Weijun Hong, Feiping Nie, Dan He, Min Yang, and Joshua Zhexue Huang. Spectral clustering of large-scale data by directly solving normalized cut. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1206–1215, 2018.

Yannan Chen, Beichen Huang, Licheng Zhao, and Kaiming Shen. Multidimensional fractional programming for normalized cuts. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 107–114, 2001.

Jin Huang, Feiping Nie, and Heng Huang. Spectral rotation versus K-means in spectral clustering. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pp. 431–437, 2013.

Stephen C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

Zhengxin Li, Feiping Nie, Jingjing Xue, and Xuelong Li. An optimization solver of min-max cut for spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 38(5):2578–2586, 2026.

James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 281–297, 1967.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pp. 849–856, 2002.

Feiping Nie, Chris Ding, Dijun Luo, and Heng Huang. Improved minmax cut graph clustering with nonnegative relaxation. In *Machine Learning and Knowledge Discovery in Databases*, pp. 451–466, 2010.

Feiping Nie, Chaodie Liu, Rong Wang, and Xuelong Li. A novel and effective method to directly solve spectral clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (12):10863–10875, 2024a.

Feiping Nie, Jitao Lu, Danyang Wu, Rong Wang, and Xuelong Li. A novel normalized-cut solver with nearest neighbor hierarchical initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):659–666, 2024b.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2nd edition, 2006.

Svatopluk Poljak. Integer linear programs and local search for Max-Cut. *SIAM Journal on Computing*, 24(4):822–839, 1995.

Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.

Kaiming Shen and Wei Yu. Fractional programming for communication systems—part I: Power control and beamforming. *IEEE Transactions on Signal Processing*, 66(10):2616–2630, 2018.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L. Crowley, and Dominique Vaufreydaz. TokenCut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15790–15801, 2023.

Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 313–319, 2003.

A Proof of Proposition 1

Proof. If $V = 0$, then $\mathrm{Proj}_{\mathcal{I}^{N\times K}}(\bar{X}) = \{\bar{X}\}$, so the conclusion holds for any $\alpha > 0$. Now suppose that $V \neq 0$ and $0 < \alpha < 1/(\sqrt{2}\|V\|_F)$. Let $X^+ \in \mathrm{Proj}_{\mathcal{I}^{N\times K}}(\bar{X} - \alpha V)$. Since $\bar{X} \in \mathcal{I}^{N\times K}$ is feasible for the projection, the optimality of X^+ gives

$$\|X^+ - \bar{X} + \alpha V\|_F^2 \leq \|\alpha V\|_F^2.$$

Hence, we have

$$\|X^+ - \bar{X}\|_F^2 \leq -2\alpha\langle V, X^+ - \bar{X}\rangle \leq 2\alpha\|V\|_F\|X^+ - \bar{X}\|_F.$$

If $X^+ \neq \bar{X}$, dividing by $\|X^+ - \bar{X}\|_F$ yields

$$\|X^+ - \bar{X}\|_F \leq 2\alpha\|V\|_F < \sqrt{2}. \tag{24}$$

On the other hand, any two distinct matrices $X, Z \in \mathcal{I}^{N\times K}$ differ in at least one one-hot row. Each differing row contributes 2 to their squared Frobenius distance, and therefore

$$\|X - Z\|_F^2 \geq 2, \qquad X, Z \in \mathcal{I}^{N\times K}, \quad X \neq Z.$$

Thus, $X^+ \neq \bar{X}$ would imply $\|X^+ - \bar{X}\|_F \geq \sqrt{2}$, contradicting (24). Hence $X^+ = \bar{X}$. Since X^+ was arbitrarily chosen from $\mathrm{Proj}_{\mathcal{I}^{N\times K}}(\bar{X} - \alpha V)$, the desired conclusion follows. □

B Proof of Theorem 1

Proof. We first prove (i). Fix an outer iteration t and consider any $i \in \mathbb{Z}_N^+$. Since $\pi_1, \dots, \pi_N$ is a permutation of $\mathbb{Z}_N^+$, the component indexed by π_i has not been modified in the first $i-1$ trial moves. Hence, by (16),

$$\widehat{\ell}_{\pi_i}^{[i-1]} = \widehat{\ell}_{\pi_i}^{[0]} = \bar{\ell}_{\pi_i} \neq \widetilde{\ell}_{\pi_i} = \widehat{\ell}_{\pi_i}^{[i]},$$

where the middle inequality follows from (13). For every $j \neq \pi_i$, (16) gives $\widehat{\ell}_j^{[i]} = \widehat{\ell}_j^{[i-1]}$. Therefore,

$$\widehat{X}^{[i]} = \big(\widehat{X}^{[i-1]}\big)_{\pi_i \to \widetilde{\ell}_{\pi_i}} \in \mathring{\mathcal{N}}_1\big(\widehat{X}^{[i-1]}\big) \subseteq \mathcal{I}^{N\times K}, \qquad i \in \mathbb{Z}_N^+, \tag{25}$$

where the last inclusion follows from $\widehat{X}^{[i-1]} \in \mathcal{I}^{N\times K}$. This proves (i). In particular, by (25), $\widehat{X}^{[i]}$ and $\widehat{X}^{[i-1]}$ differ in exactly one one-hot row, hence

$$\|\widehat{X}^{[i]} - \widehat{X}^{[i-1]}\|_F^2 = 2, \qquad i \in \mathbb{Z}_N^+. \tag{26}$$

We next prove (ii). Suppose that $m_t \geq 1$ at the t-th outer iteration. By the construction of TSPG, condition (17) holds for $i = 1, \dots, m_t$. Summing these inequalities and using (26) gives

$$\Phi(\widehat{X}^{[m_t]}) + m_t c \leq \Phi(\widehat{X}^{[0]}).$$

Together with $\widehat{X}^{[0]} = X^{(t)}$ and $\widehat{X}^{[m_t]} = X^{(t+1)}$, this proves (ii).

Finally, we prove (iii). Since $\mathcal{I}^{N\times K} \cap \mathrm{dom}(\Phi)$ is nonempty and finite and Φ is real-valued on this set, $\Phi_{\min}$ is finite. If the algorithm performs T nontrivial outer updates, summing the inequality in Theorem 1(ii) over $t = 0, \dots, T-1$ gives

$$\Phi(X^{(T)}) + c\sum_{t=0}^{T-1} m_t \leq \Phi(X^{(0)}). \tag{27}$$

Since $\Phi(X^{(T)}) \geq \Phi_{\min}$ and $m_t \geq 1$ for every nontrivial outer update, we prove (iii) with (27). □

C Proof of Theorem 2

Proof. Suppose that TSPG terminates at $\bar{X}$ at the T-th outer iteration, and let $V := \nabla\Phi(\bar{X})$. By the initialization and acceptance rules, $\bar{X} \in \mathcal{I}^{N\times K} \cap \mathrm{dom}(\Phi)$. Since the algorithm terminates at this iteration, $m_T = 0$, so the first trial point fails (17). As $\widehat{X}^{[1]} \in \mathrm{dom}(\Phi)$ by assumption, we obtain

$$\Phi(\widehat{X}^{[1]}) + \frac{c}{2}\|\widehat{X}^{[1]} - \bar{X}\|_F^2 > \Phi(\bar{X}). \tag{28}$$

By Theorem 1(i), $\widehat{X}^{[1]} \in \mathring{\mathcal{N}}_1(\bar{X})$, and hence $\|\widehat{X}^{[1]} - \bar{X}\|_F^2 = 2$. Thus, (28) gives

$$\Phi(\widehat{X}^{[1]}) + c > \Phi(\bar{X}). \tag{29}$$

By (20) with $i = 1$, we obtain $\widehat{X}^{[1]} \in \mathrm{Proj}_{\mathcal{I}^{N\times K}}(\bar{\rho}_{\pi_1}\bar{X} - V)$. Hence, for any $Y \in \mathring{\mathcal{N}}_1(\bar{X})$, we have

$$\big\|\widehat{X}^{[1]} - \bar{\rho}_{\pi_1}\bar{X} + V\big\|_F^2 \leq \big\|Y - \bar{\rho}_{\pi_1}\bar{X} + V\big\|_F^2. \tag{30}$$

Writing $\widehat{X}^{[1]} - \bar{\rho}_{\pi_1}\bar{X} + V = \widehat{X}^{[1]} - \bar{X} + (1-\bar{\rho}_{\pi_1})\bar{X} + V$ and similarly for Y, (30) is equivalent to

$$\begin{aligned}&\frac{1}{2}\|\widehat{X}^{[1]} - \bar{X}\|_F^2 + \Big\langle \widehat{X}^{[1]} - \bar{X},\, (1-\bar{\rho}_{\pi_1})\bar{X} + V\Big\rangle \\ &\qquad \leq \frac{1}{2}\|Y - \bar{X}\|_F^2 + \big\langle Y - \bar{X},\, (1-\bar{\rho}_{\pi_1})\bar{X} + V\big\rangle.\end{aligned} \tag{31}$$

For any $Y \in \mathring{\mathcal{N}}_1(\bar{X})$, we have $\|Y - \bar{X}\|_F^2 = 2$ and $\langle Y - \bar{X}, \bar{X}\rangle = -1$. Since $\widehat{X}^{[1]} \in \mathring{\mathcal{N}}_1(\bar{X})$ by Theorem 1(i), these identities also hold with Y replaced by $\widehat{X}^{[1]}$. Hence (31) yields

$$\Big\langle V, \widehat{X}^{[1]} - \bar{X}\Big\rangle \leq \langle V, Y - \bar{X}\rangle, \qquad \forall Y \in \mathring{\mathcal{N}}_1(\bar{X}). \tag{32}$$

For any $Y \in \mathring{\mathcal{N}}_1(\bar{X})$, since $\|Y - \bar{X}\|_F^2 = 2$, the definition (22) gives

$$\begin{aligned} Q_c(Y \mid \bar{X}) &= \Phi(\bar{X}) + \langle V, Y - \bar{X}\rangle + R_1(\bar{X}) + c \\ &\geq \Phi(\bar{X}) + \langle V, Y - \bar{X}\rangle + \Phi(\widehat{X}^{[1]}) - \Phi(\bar{X}) - \left\langle V, \widehat{X}^{[1]} - \bar{X}\right\rangle + c \\ &\geq \Phi(\widehat{X}^{[1]}) + c > \Phi(\bar{X}) = Q_c(\bar{X} \mid \bar{X}), \end{aligned}$$

where the first inequality follows from (21) and $\widehat{X}^{[1]} \in \mathring{\mathcal{N}}_1(\bar{X}) \cap \mathrm{dom}(\Phi)$, the second from (32), and the strict inequality from (29). Thus, $Q_c(\bar{X} \mid \bar{X}) < Q_c(Y \mid \bar{X})$ for every $Y \in \mathring{\mathcal{N}}_1(\bar{X})$. Therefore, $\bar{X}$ is a one-flip optimum of $Q_c(\cdot \mid \bar{X})$. □

D Detailed incremental updates for quadratic-fractional graph-cut objectives

This appendix provides the details of the incremental implementation of TSPG for the graph-cut objectives considered in Subsection 3.4. For these objectives, the function Φ has the form

$$\Phi(X) = \sum_{k=1}^{K} \frac{f(X_k)}{g(X_k)}. \tag{33}$$

Here, both the numerator and denominator are quadratic functions, given by

$$f(x) = x^\top A x + a^\top x + r_f, \qquad g(x) = x^\top B x + b^\top x + r_g, \tag{34}$$

where $A, B \in \mathbb{R}^{N\times N}$ are symmetric. The initial point $X^{(0)} \in \mathcal{I}^{N\times K}$ is assumed to satisfy

$$g(X_k^{(0)}) \neq 0, \qquad k = 1, \ldots, K, \tag{35}$$

so that the objective value and gradient are well-defined at the initial point. The graph-cut objectives in (2) both take this form. For these objectives, the condition is automatically satisfied when the similarity matrix satisfies $w_{ii} > 0$ for all i and the initial point $X^{(0)}$ has no zero column.

The implementation below follows TSPG line by line, using the one-flip formulas to incrementally update the quantities required for gradient computation and inner-search trial evaluation.

Maintained quantities. At the beginning of the t-th outer iteration of TSPG, write $\bar{\ell} := \ell^{(t)}$ and $\bar{X} := X^{(t)}$. For every $k \in \mathbb{Z}_K^+$, maintain

$$f_k := f(\bar{X}_k), \qquad F_k := \nabla f(\bar{X}_k), \qquad g_k := g(\bar{X}_k), \qquad G_k := \nabla g(\bar{X}_k). \tag{36}$$

Since the maintained denominators g_k are nonzero, the outer-iteration gradient $V = \nabla\Phi(\bar{X})$ is obtained columnwise using the quotient rule:

$$V_k = g_k^{-2}(g_k F_k - f_k G_k), \qquad k \in \mathbb{Z}_K^+. \tag{37}$$

The alternative labels and thresholds used by TSPG are then computed from V exactly as in (13) and (11), followed by the ordering in (15).

Incremental evaluation of an inner-search trial. Consider the i-th inner step, and let f_k, F_k, g_k, G_k, $k \in \mathbb{Z}_K^+$, denote the maintained quantities corresponding to the last accepted trial point $\widehat{X}^{[i-1]}$. Set

$$\tau := \pi_i, \qquad \theta := \bar{\ell}_\tau, \qquad \iota := \widetilde{\ell}_\tau. \tag{38}$$

By Theorem 1(i), we have

$$\widehat{X}^{[i]} = \big(\widehat{X}^{[i-1]}\big)_{\tau\to\iota} \in \mathring{\mathcal{N}}_1\big(\widehat{X}^{[i-1]}\big). \tag{39}$$

Following (23), the function values corresponding to the next trial point $\widehat{X}^{[i]}$ differ only in columns θ and ι, and their updated values are

$$\begin{aligned} \widetilde{f}_\theta &:= f_\theta - (F_\theta)_\tau + A_{\tau\tau}, & \widetilde{f}_\iota &:= f_\iota + (F_\iota)_\tau + A_{\tau\tau}, \\ \widetilde{g}_\theta &:= g_\theta - (G_\theta)_\tau + B_{\tau\tau}, & \widetilde{g}_\iota &:= g_\iota + (G_\iota)_\tau + B_{\tau\tau}. \end{aligned} \tag{40}$$

Since (39) gives $\|\widehat{X}^{[i]} - \widehat{X}^{[i-1]}\|_F^2 = 2$, and all unchanged columnwise ratios cancel from the two sides of (17), the acceptance condition for a trial point can be written as

$$\widetilde{g}_\theta \widetilde{g}_\iota \neq 0 \quad \text{and} \quad \frac{\widetilde{f}_\theta}{\widetilde{g}_\theta} + \frac{\widetilde{f}_\iota}{\widetilde{g}_\iota} + c \leq \frac{f_\theta}{g_\theta} + \frac{f_\iota}{g_\iota}, \tag{41}$$

where the first condition ensures that the two updated denominators are nonzero, and hence that the trial point is within the domain of the objective function. The second condition is evaluated only when the first condition holds, and the trial point $\widehat{X}^{[i]}$ is accepted if both conditions are satisfied. Once the trial point is accepted, the candidate function values in (40) are committed. Applying (23) to the maintained gradients gives

$$\begin{aligned} F_\theta &\leftarrow F_\theta - 2A_\tau, & F_\iota &\leftarrow F_\iota + 2A_\tau, \\ G_\theta &\leftarrow G_\theta - 2B_\tau, & G_\iota &\leftarrow G_\iota + 2B_\tau. \end{aligned} \tag{42}$$

Then, the scalar quantities are replaced by

$$(f_\theta, f_\iota, g_\theta, g_\iota) \leftarrow (\widetilde{f}_\theta, \widetilde{f}_\iota, \widetilde{g}_\theta, \widetilde{g}_\iota). \tag{43}$$

If the trial fails, the inner search terminates and none of the updates in (42)–(43) is performed.

The resulting incremental implementation of TSPG is summarized in Algorithm 2.

Algorithm 2 Incremental TSPG for quadratic-fractional objectives

Require: $X^{(0)} \in \mathcal{I}^{N\times K}$ satisfying (35), and $c > 0$.

Initialize $\ell^{(0)} \in \mathcal{M}(X^{(0)})$ and $\{f_k, F_k, g_k, G_k\}_{k=1}^K$ at $X^{(0)}$.
for $t = 0, 1, 2, \ldots$ **do**
  Set $\bar{\ell} \leftarrow \ell^{(t)}$ and compute $V \leftarrow \nabla\Phi(X^{(t)})$ by (37).
  Compute $\widetilde{\ell}$, $\{\bar{\rho}_i\}_{i=1}^N$, and $\{\pi_i\}_{i=1}^N$ according to (13), (11), and (15).
  Set $\widehat{\ell}^{[0]} \leftarrow \bar{\ell}$ and $m_t \leftarrow 0$.
  for $i = 1, \ldots, N$ **do**
    Set (τ, θ, ι) according to (38).
    Compute $\widetilde{f}_\theta, \widetilde{f}_\iota, \widetilde{g}_\theta, \widetilde{g}_\iota$ according to (40).
    if (41) holds **then**
      Update $F_\theta, F_\iota, G_\theta, G_\iota$ according to (42).
      Update $f_\theta, f_\iota, g_\theta, g_\iota$ according to (43).
      Set $\widehat{\ell}^{[i]} \leftarrow \widehat{\ell}^{[i-1]}$, $\widehat{\ell}^{[i]}_\tau \leftarrow \iota$, and $m_t \leftarrow i$.
    else
      break.
    end if
  end for
  If $m_t > 0$, set $\ell^{(t+1)} \leftarrow \widehat{\ell}^{[m_t]}$; **otherwise, stop**.
end for