ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11463516

OPTIMAL QUASI-CLIQUE DETECTION VIA LASRY-LIONS DOUBLE ENVELOPES

Emmanouil Kariotakis, Alexander Bodard, Panagiotis Patrinos, Aritra Konar

KU Leuven, Department of Electrical Engineering (ESAT-STADIUS),
Kasteelpark Arenberg 10, 3001 Leuven, Belgium
{emmanouil.kariotakis, alexander.bodard, panos.patrinos, aritra.konar}@kuleuven.be

ABSTRACT

Extracting large and dense subgraphs is a key task in graph mining applications. Pre-existing formulations are either overly restrictive (the maximum clique) or yield sparsely connected subgraphs (the densest subgraph problem). The optimal quasi-clique (OQC) problem allows extraction of large cliques and near-cliques of different sizes via a tunable parameter, but is NP-hard, with no principled optimization algorithm known. We propose a novel homotopy-based framework for OQC based on the Lasry-Lions (LL) double envelope that enables efficient approximation via a sequence of smooth subproblems. Experiments on real-world graphs show that our method outperforms the state-of-the-art greedy baseline for OQC, extracting larger and denser (quasi)-cliques that the baseline fails to detect.

***Index Terms*—** Dense Subgraph Discovery, Optimal Quasi-cliques, Lasry-Lions Smoothing, Homotopy-based optimization

1. INTRODUCTION

Dense subgraph discovery (DSD) is a key primitive in graph mining that extracts tightly interconnected subsets of vertices from an undirected graph. The problem has numerous applications ranging from mining trending topics in social media, finding functional modules in gene co-expression networks, and locating communities in social networks (see recent survey [1] and references therein). In recent years, DSD has also found application in spotting fraudulent behavior in e-commerce and financial transaction networks [2–4].

The broad applicability of DSD has spurred the development of several algorithmic techniques, which we survey in brief (see [1] for a more in-depth overview). The archetype of a dense subgraph is the maximum clique. However, extracting the maximum clique is an NP–hard problem [5], and non-trivial approximation in polynomial-time is also difficult, under standard complexity-theoretic conjectures [6]. Additionally, the requirement that every pair of vertices comprising a clique be connected can prove to be restrictive, since in practice, large quasi-cliques can also be of interest. As a result, relaxed notions of dense subgraphs have been considered, a key example being the Densest Subgraph (DSG) problem [7], which outputs the vertex subset that maximizes the average induced degree. The problem can be solved exactly in polynomial-time via maximum-flow, and also admits a $1/2$-factor approximation via a linear-time greedy algorithm [8]. Recently, improved algorithms with better approximation guarantees were devised by leveraging tools from convex optimization [9–11] to solve a tight linear-programming relaxation of DSG proposed in [8]. Notwithstanding such developments, DSG can output sparsely interconnected subgraphs of large sizes on real-world graphs [12]. This is because by design, DSG is endowed with a unique solution that is maximal in size [11]. Since there is no flexibility in selecting subgraphs of different sizes, DSG need not be ideal for mining large cliques and near-cliques.

Here, we consider an alternative formulation better suited for extracting large and dense subgraphs - the optimal quasi-clique (OQC) problem of [12]. OQC maximizes an objective function which assigns large rewards to vertex subsets that are highly interconnected while not being very large in size. The total reward assigned to a subset can be controlled via a single parameter $\alpha \in (0, 1)$. The follow-up work of [13] established that for a fixed α value, the maximizers of OQC correspond to the densest subgraph of a particular size, with larger α yielding smaller, denser subgraphs. Hence, variation of α in OQC allows extraction of dense subgraphs of different sizes. In practice, the OQC formulation yields denser subgraphs compared to DSG on real-world datasets [12]. Despite these advantages over DSG, OQC is NP–hard [13, 14], and at present, no principled optimization algorithm is known for the problem. The current state-of-the-art is the greedy heuristic GREEDYOQC [12], which does not feature optimality guarantees. Consequently, the full potential of the OQC framework in mining dense subgraphs remains unrealized.

In this paper, our goal is to rectify this issue via a principled optimization framework for OQC. First, we recast the problem as minimizing a binary quadratic optimization prob-

A.B. and P.P. are partially supported by the Research Foundation Flanders (FWO) research projects G081222N, G033822N, G0A0920N and the Research Council KU Leuven C1 project with ID C14/24/103. E.K. and A.K. are supported by KU Leuven Special Research Fund BOF/STG-22-040.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on June 24,2026 at 09:46:40 UTC from IEEE Xplore. Restrictions apply.

lem. In light of the problem's intrinsic difficulty, we adopt a homotopy-based approach for computing high-quality approximate solutions; an idea that has recently gained traction for tackling NP–hard optimization problems [15–18]. Our approach relies on approximating the indicator function of the binary constraints (which constitute a non-convex and disconnected set) via a continuously differentiable surrogate known as the Lasry-Lions (LL) double envelope [19]. The LL envelope features a pair of controllable parameters (λ, μ) which can be adjusted to vary the quality of approximation relative to the indicator function of the binary constraints. By appropriately varying (λ, μ); i.e., starting from "loose" to tighter approximations of the binary constraints, we obtain a sequence of differentiable optimization subproblems, each of which can be solved at scale using fast Quasi-Newton methods [20]. The solution of each subproblem is then used to warm-start the succeeding one, and the sequence of global optimizers to these subproblems can be shown to converge to the global optimizers of the combinatorial OQC problem. Experiments on real-world graphs illustrate the promising performance of the approach, extracting a broader spectrum of large dense subgraphs compared to GREEDYOQC.

2. THE OPTIMAL QUASI-CLIQUE PROBLEM

Consider an undirected, unweighted graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ on n vertices with m edges. Given a subset of vertices $\mathcal{S} \subseteq \mathcal{V}$, let $e(\mathcal{S})$ denote the number of edges in the subgraph $\mathcal{G}_\mathcal{S}$ induced by $\mathcal{S}$. The *edge-density* of $\mathcal{S}$ is defined as $\delta(\mathcal{S}) := e(\mathcal{S})/\binom{|\mathcal{S}|}{2}$, which measures how close $\mathcal{G}_\mathcal{S}$ is to a clique on $|\mathcal{S}|$ vertices. In particular, $\delta(\mathcal{S}) = 1$ if and only if $\mathcal{S}$ is a clique. However, maximizing $\delta(\mathcal{S})$ directly need not yield a satisfactory solution, since even a set of two vertices (i.e., a single edge) attains maximum density. Therefore, we adopt the *optimal quasi-clique* (OQC) formulation introduced in [12], which seeks the maximizer of the following objective function

$$f_\alpha(\mathcal{S}) := e(\mathcal{S}) - \alpha\binom{|\mathcal{S}|}{2}. \tag{1}$$

The first term encourages the subgraph induced by $\mathcal{S}$ to have a large number of edges while the second term penalizes large subgraph sizes. The regularization parameter $\alpha \in (0, 1)$ plays a balancing act in trading off subgraph density for size. The objective function admits the following interpretation: the second term can be viewed as the number of edges that appear in expectation in an Erdős-Rényi random graph defined on the vertex subset $\mathcal{S}$, where $\alpha \in (0, 1)$ denotes the probability of an edge connecting a pair of vertices. Thus, $f_\alpha(\mathcal{S})$ assigns a greater reward to subgraphs $\mathcal{G}_\mathcal{S}$ which exhibit a large surplus of edges with respect to the random subgraph model. Overall, OQC aims to solve the following optimization problem

$$\max_{\mathcal{S}\subseteq\mathcal{V}} f_\alpha(\mathcal{S}) \tag{2}$$

The choice of the parameter α affects the size and density of the extracted solution. Intuitively, selecting a small value of α allows large, non-dense subgraphs to exhibit a large edge surplus. As the value of α is increased, dense subgraphs of smaller size are favored. The work of [13] demonstrated that for a fixed α, the global maximizers of OQC correspond to the subgraphs of a particular size with the largest density. Hence, from a conceptual standpoint, through continuous variation of α, the maximizers of OQC yield the densest subgraph of every size, including the maximum clique as a special case. On the computational side, OQC was shown [14] to be NP–hard for undirected graphs on n vertices for all $\alpha \in [1 - 1/n^2, 1)$. Later, [13] showed that OQC is NP–hard for an expanded range of $\alpha \in [0.75, 1)$, irrespective of n. At present, the main algorithm for obtaining approximate maximizers of (2) is GREEDYOQC [12].

3. PROPOSED APPROACH

Problem Reformulation: Our objective is to develop a more principled approach for computing (approximate) maximizers of problem (2). First, we reformulate (2) as a binary quadratic optimization problem. Note that for any vertex subset $\mathcal{S} \subseteq \mathcal{V}$, we can associate a binary vector $\mathbf{x} \in \{0, 1\}^n$ such that $u \in \mathcal{S} \Leftrightarrow x_u = 1$. In terms of $\mathbf{x}$, we can express (1) as

$$\begin{aligned} f_\alpha(\mathbf{x}) &= \mathbf{x}^\top\mathbf{A}\mathbf{x} - \alpha\mathbf{x}^\top(\mathbf{e}\mathbf{e}^\top - \mathbf{I})\mathbf{x} \\ &= \mathbf{x}^\top(\mathbf{A} + \alpha\mathbf{I} - \alpha\mathbf{e}\mathbf{e}^\top)\mathbf{x}, \end{aligned} \tag{3}$$

where $\mathbf{A}$ is the $n \times n$ symmetric adjacency matrix of $\mathcal{G}$ and $\mathbf{e}$ is the $n \times 1$ vector of all ones. Consequently, problem (2) is

$$\min_{\mathbf{x}} \left\{-f_\alpha(\mathbf{x}) + \mathcal{I}_{\{0,1\}^n}(\mathbf{x})\right\}, \tag{4}$$

where the second term in the objective function denotes the indicator function of the binary constraint set.

Although problem (4) is no easier to solve compared to (2), it is in a form that is suitable for approximation using the framework of Lasry-Lions (LL) double envelopes [19]. In formal terms, given a proper, lower semi-continuous function $h : \mathbb{R}^n \to \bar{\mathbb{R}}$ and parameters $\lambda > \mu > 0$, its λ-Moreau envelope is $h^\lambda(\mathbf{x}) = \inf_\mathbf{y}\{h(\mathbf{y}) + (1/2\lambda)\|\mathbf{x} - \mathbf{y}\|_2^2\}$, and its (λ, μ)-Lasry-Lions double envelope is

$$h^{(\lambda,\mu)}(\mathbf{x}) = -(-h^\lambda)^\mu(\mathbf{x}) = \sup_\mathbf{y}\{h^\lambda(\mathbf{y}) - \frac{1}{2}\mu\|\mathbf{x}-\mathbf{y}\|_2^2\}. \tag{5}$$

It can be shown that the LL envelope $h^{(\lambda,\mu)}(\cdot)$ is Lipschitz smooth [16, Proposition II.3] and constitutes a tighter underestimator of h relative to its counterpart h^λ [16, Fact II.2(i)]. Leveraging this technique, we can replace $\mathcal{I}_{\{0,1\}^n}$ via its smooth LL envelope to arrive at the following smooth optimization problem

$$\min_{\mathbf{x}} \left\{\phi_{\alpha,\lambda,\mu}(\mathbf{x}) := -f_\alpha(\mathbf{x}) + h^{(\lambda,\mu)}(\mathbf{x})\right\}, \tag{6}$$

where $h^{(\lambda,\mu)}$ denotes the LL double envelope of $\mathcal{I}_{\{0,1\}^n}$ with parameters (λ, μ).

Authorized licensed use limited to: KU Leuven University Library. Downloaded on June 24,2026 at 09:46:40 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 LL double envelope homotopy approach.

1: Choose $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and sequences $\tau^{(k)}, \lambda^{(k)}, \mu^{(k)} > 0$ such that $\tau^{(k)} \searrow 0$ and $0 < \mu^{(k)} < \lambda^{(k)} \searrow 0$ with $\lambda^{(k)} - \mu^{(k)} \searrow 0$
2: Set $k = 1$
3: **while** stopping criterion is not satisfied **do**
4: Starting at $\mathbf{x}^{(k-1)}$, use a descent method to find $\mathbf{x}^{(k)}$ such that $\|\nabla \phi_{\alpha,\lambda^{(k)},\mu^{(k)}}(\mathbf{x}^{(k)})\| \leq \tau^{(k)}$
5: $k \leftarrow k + 1$
6: **end while**

Let us now compute the LL double envelope explicitly. First, remark that h is separable, i.e., $h(\mathbf{x}) = \sum_{i=1}^n \bar{h}(x_i)$ where $\bar{h} = \mathcal{I}_{\{0,1\}}$ is a scalar function. From [16, Fact II.2(vi)] it follows that

$$h^{(\lambda,\mu)}(\mathbf{x}) = \sum_{i=1}^n \bar{h}^{(\lambda,\mu)}(x_i).$$

It remains to compute $\bar{h}^{(\lambda,\mu)}$. After some algebra, we can show that the LL double envelope $\bar{h}^{(\lambda,\mu)}$ is given by

$$\bar{h}^{(\lambda,\mu)}(x) := \begin{cases} \frac{x^2}{2(\lambda-\mu)}, & x < \frac{(\lambda-\mu)}{2\lambda} \\ \frac{(x-1)^2}{2(\lambda-\mu)}, & x > \frac{\mu}{\lambda} + \frac{(\lambda-\mu)}{2\lambda} \\ \frac{1}{8\lambda} - \frac{1}{2\mu}(x - \frac{1}{2})^2, & \text{otherwise} \end{cases}. \quad (7)$$

For smoothing parameters $\lambda > \mu > 0$, this LL double envelope is a piecewise quadratic function with full domain and Lipschitz-continuous gradient.

The smooth problem (6) (with fixed parameters (λ, μ)) can be solved using a scalable implementation of the L-BFGS algorithm [20]. By varying (λ, μ) appropriately, we can control the quality of approximation of the LL envelope relative to $\mathcal{I}_{\{0,1\}^n}(\cdot)$. For example, as $\lambda > \mu \to \infty$ and $\lambda - \mu \searrow 0$, $h^{(\lambda,\mu)}$ converges pointwise to the indicator function of the unit interval $[0, 1]^n$, which is the convex hull of $\mathcal{I}_{\{0,1\}^n}$. On the other hand, as $0 < \mu < \lambda \searrow 0$, $h^{(\lambda,\mu)}$ converges pointwise to $\mathcal{I}_{\{0,1\}^n}$ itself, in a smooth fashion. Moreover, the global minimizers of the smooth problem (6) converge to the global minimizers of (4) as the approximations become tighter [16, Theorem 3.1]. However, tighter approximations produce more difficult optimization problems to solve.

Homotopy-based Approach: To address this issue, we adopt a homotopy approach (similar to the one put forth in [16]). The main idea is to generate a sequence of smooth minimization problems of the form (6), starting from an "easy" problem with a loose approximation of $\mathcal{I}_{0,1^n}(\cdot)$ and progressively refining the parameters (λ, μ) so that the approximation becomes more accurate. This adaptive adjustment allows us to obtain high-quality solutions to (4) while avoiding the difficulties of directly solving the tightest approximation from the outset. The detailed procedure is summarized in Algorithm 1.

DATASET	n	m	ω
GITHUB [22]	$38K$	$289K$	24
ROUTER-CAIDA [23]	$192K$	$609K$	17
WEB-STANFORD [24]	$282K$	$1.99M$	61

Table 1. Summary of dataset statistics: number of vertices (n), number of edges (m), maximum clique size (ω).

α	GITHUB	WEB-STANFORD	ROUTER-CAIDA
0.60	0	0	0
0.70	0	$2.61 \cdot 10^{-5}$	$0.38 \cdot 10^{-5}$
0.80	0	$3.93 \cdot 10^{-5}$	0
0.90	0	$1.95 \cdot 10^{-5}$	0
0.95	0	$1.08 \cdot 10^{-5}$	0

Table 2. Integrality Error of the final solution of the Densest One-Hop + LL method, for different datasets.

4. EXPERIMENTAL EVALUATION

Experimental Setup: We evaluated the proposed approach on real-world graph datasets mentioned in Table 1. As a baseline, we employed the state-of-the-art GREEDYOQC algorithm [12]. Since OQC is a difficult problem, the choice of initialization is important for the double envelope framework to find good solutions. One choice is the solution of GREEDYOQC itself - another is the densest one-hop neighborhood in the graph. This choice is based on prior work [21] which demonstrated that real-world graphs can harbor dense one-hop neighborhoods of non-trivial sizes. Here, among all one-hop neighborhood subgraphs, we pick the one with the highest average induced degree, and use it as initialization. Random initialization was tested but consistently led to degenerate, empty solutions and was thus omitted.

We set the initial parameters of Algorithm 1 as follows: $\tau^{(1)} = 10^{-1}$, $\lambda^{(1)} = 10^5$, and $\mu^{(1)} = \lambda^{(1)}/2$. The maximum number of outer iterations k was fixed to 20. At each iteration, all parameters were reduced by 10%, i.e., $\lambda^{(k+1)} = 0.9\lambda^{(k)}$, $\mu^{(k+1)} = 0.9\mu^{(k)}$, and $\tau^{(k+1)} = 0.9\tau^{(k)}$. For the descent method, we employed L-BFGS [20] due to its efficiency and low memory requirements. All experiments were conducted on a single machine equipped with an Intel i7-13700K CPU (3.4GHz) and 128GB of main memory, running Python 3.12.0. The implementation is publicly available on Github [1].

Testing Integrality: Although we initialize Algorithm 1 with a subgraph, it need not output one; i.e., an integral solution. This is because the LL double envelope does not exactly enforce the binary constraints for a fixed pair $(\lambda^{(k)}, \mu^{(k)})$; only asymptotically the minimizers are guaranteed to correspond.

Nevertheless, in Table 2, we show how close the final output of the Densest One-Hop + LL method is to being integral, for varying values of α across different datasets. The integrality error (IE) is computed using the formula

[1] https://github.com/ekariotakis/OQC_LL_icassp26

Authorized licensed use limited to: KU Leuven University Library. Downloaded on June 24,2026 at 09:46:40 UTC from IEEE Xplore. Restrictions apply.

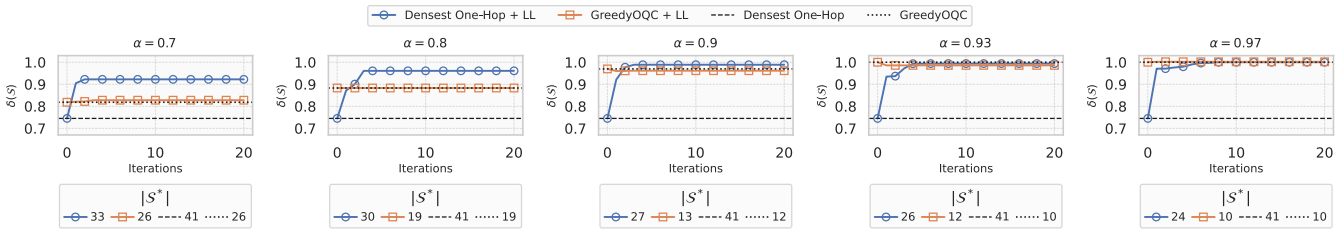


Fig. 1. Edge-density $\delta(\mathcal{S})$ vs outer iterations k of the double envelope algorithm with different initializations (densest one-hop and GREEDYOQC) for the GITHUB dataset. Legend in each subfigure displays the size of the final solution $\mathcal{S}^*$ returned by each method. Larger sizes and higher edge-density are better.

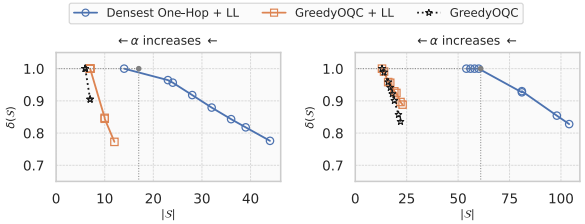


Fig. 2. Edge-density $\delta(\mathcal{S})$ vs solution size $|\mathcal{S}|$ of the extracted subgraphs for different values of $\alpha \in \{0.6, 0.65, \ldots, 0.9, 0.95\}$, comparing our method (LL) with different initializations and GREEDYOQC, on the ROUTER-CAIDA (left) and WEB-STANFORD (right) datasets. Larger sizes and higher edge-density are better. The gray dot in each figure indicates the max clique of each dataset.

$\text{IE}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_i^2)$. It is evident that the algorithm yields integral solutions in most cases. Even when it does not, it is very close to being integral. For instance, for WEB-STANFORD with $\alpha = 0.8$, the error is approximately $4 \cdot 10^{-5}$ - out of $282K$ entries, only ~ 11 were non-integral. In such cases, we performed a final rounding step on the non-integral entries by projecting them onto the $\{0, 1\}$ set.

Larger and Denser Solutions: Figure 1 depicts the edge-densities of the subgraphs extracted as a function of the number of outer iterations k for the GITHUB dataset across different values of α. Our method, when initialized with the densest one-hop neighborhood, consistently produces larger subgraphs with higher edge-density than both GREEDYOQC and our method with GREEDYOQC as initialization. This demonstrates the effectiveness of the one-hop neighborhood initialization in extracting *larger* and denser solutions. As α increases, the extracted subgraph increasingly resembles a clique. In particular, for $\alpha = 0.97$, our method with densest one-hop neighborhood initialization is able to extract the graph's maximum clique. In contrast, GREEDYOQC also identifies a clique, but a smaller one containing only 10 vertices. This illustrates that our approach is not only competitive for near-clique extraction, but can also recover the maximum clique in instances where GREEDYOQC falls short.

Exploring the Edge-Density vs Subgraph-Size Frontier: In Figure 2, we present our results on the ROUTER-CAIDA and WEB-STANFORD datasets. We show the edge-density $\delta(\mathcal{S})$ and solution size $|\mathcal{S}|$ of the extracted subgraphs for $\alpha \in \{0.6, 0.65, \ldots, 0.95\}$. For each distinct α, our method with densest one-hop neighborhood initialization is successful in extracting a large and dense (near)-clique. Hence, the method is able to produce a wider range of dense subgraphs. In contrast, in many cases, GREEDYOQC and its refinement with the double envelope return the same subgraph for different α values. This means that even when different α values should yield different subgraphs, GREEDYOQC often outputs the same result. In addition, applying the refinement with the double envelope does not bring any marked improvement in the quality of the solution. Furthermore, the subgraphs identified are also smaller with lower density. To sum up, when the double envelope is combined with the densest one-hop neighborhood initialization, we obtain a significantly better approximation to the edge-density vs subgraph size frontier, identifying not only large quasi-cliques but also cliques which are closer to the maximum clique (gray solid dot).

5. CONCLUSION

In this work, we considered the problem of extracting large and dense subgraphs by focusing on the optimal quasi-clique (OQC) formulation. Unlike prior approaches that are either overly restrictive, or favor large and sparse subgraphs, OQC enables extraction of large cliques and near-cliques of different sizes via a single tunable parameter. However, no principled algorithms are currently known for this NP–hard problem. Our work addresses this issue by putting forth a homotopy-based optimization scheme that leverages the Lasry-Lions double envelope to solve a sequence of smooth subproblems which progressively approximate OQC. As the approximations become tighter, the global optimizers of the smooth subproblems coincide with that of OQC. This principled optimization strategy enables exploration of the subgraph size-density frontier and leads to the discovery of larger and denser (near)-cliques on real-world graphs relative to the existing state-of-the-art greedy baseline, thereby highlighting its effectiveness. Since initialization plays a key role in the final outcome of our method, future work will explore alternative initialization strategies.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on June 24,2026 at 09:46:40 UTC from IEEE Xplore. Restrictions apply.

6. REFERENCES

[1] Tommaso Lanciano, Atsushi Miyauchi, Adriano Fazzone, and Francesco Bonchi, "A survey on the densest subgraph problem and its variants," *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–40, 2024.

[2] X. Li, Shenghua Liu, Zifeng Li, Xiaotian Han, Chuan Shi, Bryan Hooi, He Huang, and Xueqi Cheng, "Flowscope: Spotting money laundering based on graphs," in *Proc. of AAAI*, 2020, vol. 34, pp. 4731–4738.

[3] Tianyi Chen and Charalampos Tsourakakis, "Antibenford subgraphs: Unsupervised anomaly detection in financial networks," in *KDD*, 2022, pp. 2762–2770.

[4] B. Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos, "Fraudar: Bounding graph fraud in the face of camouflage," in *KDD*, 2016, pp. 895–904.

[5] Richard M Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*, pp. 85–103. Springer, 1972.

[6] Johan Håstad, "Clique is hard to approximate within n 1- ε," *Acta Mathematica*, vol. 182, no. 1, pp. 105–142, 1999.

[7] Andrew V Goldberg, *Finding a maximum density subgraph*, Technical report, UC Berkeley, CA, 1984.

[8] Moses Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Int. Worksh. on Approx. Alg. for Comb. Opt.* Springer, 2000, pp. 84–95.

[9] Digvijay Boob, Yu Gao, Richard Peng, Saurabh Sawlani, Charalampos Tsourakakis, Di Wang, and Junxing Wang, "Flowless: Extracting densest subgraphs without flow computations," in *Proc. Web Conf.*, 2020, pp. 573–583.

[10] C. Chekuri, K. Quanrud, and M. R. Torres, "Densest subgraph: Supermodularity, iterative peeling, and flow," in *Proc. of SODA*. SIAM, 2022, pp. 1531–1555.

[11] Elfarouk Harb, Kent Quanrud, and Chandra Chekuri, "Faster and scalable algorithms for densest subgraph and decomposition," *NeurIPS*, vol. 35, pp. 26966–26979, 2022.

[12] Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli, "Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees," in *KDD*, 2013, pp. 104–112.

[13] Aritra Konar and Nicholas D Sidiropoulos, "Optimal quasi-clique: hardness, equivalence with densest-k-subgraph, and quasi-partitioned community mining," in *Proc. of AAAI*, 2024, vol. 38, pp. 8608–8616.

[14] Charalampos Tsourakakis, "Streaming graph partitioning in the planted partition model," in *ACM COSN*, 2015, pp. 27–35.

[15] Mingjie Shao and Wing-Kin Ma, "Binary mimo detection via homotopy optimization and its deep adaptation," *IEEE Trans. Signal Process.*, vol. 69, pp. 781–796, 2020.

[16] Miguel Simões, Andreas Themelis, and Panagiotis Patrinos, "Lasry-lions envelopes and nonconvex optimization: A homotopy approach," in *29th EUSIPCO*. IEEE, 2021, pp. 2089–2093.

[17] Junbin Liu, Ya Liu, Wing-Kin Ma, Mingjie Shao, and Anthony Man-Cho So, "Extreme point pursuit—part i: A framework for constant modulus optimization," *IEEE Trans. Signal Process.*, 2024.

[18] Junbin Liu, Ya Liu, Wing-Kin Ma, Mingjie Shao, and Anthony Man-Cho So, "Extreme point pursuit—part ii: Further error bound analysis and applications," *IEEE Trans. Signal Process.*, 2024.

[19] Hédy Attouch and Dominique Aze, "Approximation and regularization of arbitrary functions in hilbert spaces by the lasry-lions method," in *Annales de l'Institut Henri Poincaré C, Analyse non linéaire*. Elsevier, 1993, vol. 10, pp. 289–312.

[20] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM TOMS*, vol. 23, no. 4, pp. 550–560, 1997.

[21] Aritra Konar and Nicholas D Sidiropoulos, "Mining large quasi-cliques with quality guarantees from vertex neighborhoods," in *KDD*, 2020, pp. 577–587.

[22] Benedek Rozemberczki, Carl Allen, and Rik Sarkar, "Multi-scale attributed node embedding," *Journal of Complex Networks*, vol. 9, no. 2, pp. cnab014, 2021.

[23] Ryan A. Rossi and Nesreen K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. of AAAI*, 2015.

[24] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

Authorized licensed use limited to: KU Leuven University Library. Downloaded on June 24,2026 at 09:46:40 UTC from IEEE Xplore. Restrictions apply.