Flow Is Best, Fast and Scalable: The Incremental Parametric Cut for Maximum Density and Other Ratio Subgraph Problems

Dorit S. Hochbaum[a]

Department of Industrial Engineering and Operations Research, University of California, Berkeley, U.S.A.
dhochbaum@berkeley.edu

Keywords: Densest Subgraph, Graph Structures, Monotone Integer Programming, Breakpoints Algorithm, Conductance.

Abstract: The maximum density subgraph, or densest subgraph, problem has numerous applications in analyzing graph and community structures in social networks, DNA networks and financial networks. The densest subgraph problem has been the subject of study since the early 80s and polynomial time flow-based algorithms are known, yet research in the last couple of decades has been focused on developing heuristic methods for solving the problem claiming that flow computations are computationally prohibitive. We introduce here a new polynomial time algorithm, the *incremental parametric cut* algorithm (IPC) that solves the maximum density subgraph problem and many other max or min ratio problems in the complexity of a single minimum-cut. A characterization of all these efficiently solvable ratio problems is given here as problems with monotone integer programming formulations. IPC is much more efficient than the parametric cut algorithm since instead of generating all *breakpoints* it explores only a tiny fraction of those breakpoints. Compared to the heuristic methods, IPC not only guarantees optimality, but also runs orders of magnitude faster than the heuristic methods, as shown in an accompanying experimental study.

1 INTRODUCTION

We introduce here a new efficient algorithm for the maximum density (MD), or densest, subgraph problem and many other ratio problems. The maximum density subgraph problem is to identify a subset of nodes in the graph that maximizes the density, defined as the ratio of the weights of the edges with both endpoints in the subset, divided by the sum of weights of the nodes in the subgraph. The densest subgraph has played a central role in analyzing network structures since the 1970's. The more recent applications of the problem are in the context of very large scale networks, such as identifying emerging cyber-communities (Kumar et al., 1999), DNA motif finding (Fratkin et al., 2006), and real-time story identification (Angel et al., 2014).

The maximum density problem was studied since the late 70's. (Picard and Queyranne, 1982) are likely the first to study the problem and recognize its link to the max-flow min-cut problem. Their method was based on a general "linearization" approach that applies for any ratio optimization problem, reducing it to the λ-*question*, defined next, which they proposed to solve with a min-cut procedure on a related graph.

[a] https://orcid.org/0000-0002-2498-0512

A general ratio problem $\max_{\mathbf{x}\in\mathcal{F}} \frac{f(\mathbf{x})}{g(\mathbf{x})}$ can be reduced to a sequence of calls to an oracle that provides a yes/no answer to the λ-*question*:
Is there a feasible solution $\mathbf{x} \in \mathcal{F}$ such that $\frac{f(\mathbf{x})}{g(\mathbf{x})} > \lambda$?
Or equivalently "Is there a feasible solution $\mathbf{x} \in \mathcal{F}$ such that $f(\mathbf{x}) - \lambda g(\mathbf{x}) > 0$?"
To answer this λ-question it is sufficient to solve:

$$(\lambda\text{-problem}) \quad \max_{\mathbf{x}\in\mathcal{F}} f(\mathbf{x}) - \lambda g(\mathbf{x}).$$

If the maximum value is greater than 0 then there is a feasible solution of ratio value strictly greater than λ. Otherwise the answer is no. Specifically, if the maximum value is strictly less than 0, then there is no feasible solution of ratio value great or equal to λ. If the answer is 0 then the respective optimal solution for the λ-question has a ratio value of λ which is the maximum ratio.

Therefore, any ratio problem that has the corresponding λ-problem polynomial time solvable, and the log of the number of possible values of the ratio bounded by a polynomial quantity, is solvable in polynomial time by applying binary search on the value of the parameter λ.

(Picard and Queyranne, 1982) showed that the λ-

Hochbaum, D.
Flow Is Best, Fast and Scalable: The Incremental Parametric Cut for Maximum Density and Other Ratio Subgraph Problems.

In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2024) - Volume 1: KDIR*, pages 275-282
ISBN: 978-989-758-716-0; ISSN: 2184-3228

problem for MD can be solved as a min-cut (minimum s,t-cut) on a related graph, the construction of which appeared ad-hoc. Their method was essentially a predecessor of our IPC algorithm, showing that the λ-problem for MD would be solved up to n times, where n is the number of nodes in the graph. Here we show a systematic method that maps any optimization (and ratio) problem that is a *monotone integer* program to an associated graph and therefore all these problems are solvable in polynomial time, which as proved here, is the complexity of one min-cut procedure.

For the maximum density problem, a follow up paper by (Goldberg, 1984) improved on the algorithm of Picard and Queyranne, by using binary search on the λ-problem making multiple call to a min-cut procedure, up to $\log n$ times for the edge-unweighted node-unweighted problem. A major breakthrough, the *parametric flow* procedure, was introduced in (Gallo et al., 1989), identifying the solutions for *all* values of the parameter λ that correspond to all possible solutions to the λ-problem, and in the complexity of a single min-cut procedure. This parametric procedure used the push-relabel algorithm of (Goldberg and Tarjan, 1988). Later (Hochbaum, 1998; Hochbaum, 2008) showed a parametric cut procedure using HPF (Hochbaum PseudoFlow) also with the complexity of a single min-cut. We will refer to this parametric procedure also as fully parametric, to differentiate it from "simple" parametric, reviewed in Section 2.2.

Despite its theoretical efficiency, the parametric flow procedure has never been used to solve the densest subgraph problem, to the best of our knowledge. One contributing factor for the lack of use is that there is no implementation available for the parametric push-relabel version proposed by (Gallo et al., 1989). (However, for HPF there is a parametric flow/cut implementation publicly available, (Hochbaum, 2020a).) Instead, flow algorithms have been employed using multiple calls to min-cut in a binary search process, resulting in high running times. This perceived inefficiency gave rise to current state-of-the-art algorithms for the maximum density problem that are based on greedy heuristics that do not guarantee optimality, (Charikar, 2000), (Boob et al., 2020), (Harb et al., 2022). A recent justification for not using the polynomial time flow algorithms is that "flow computations are expensive" (Boob et al., 2020).

Our main contribution here is a new polynomial time algorithm, the *incremental parametric cut* (IPC) algorithm, that solves optimally and efficiently the densest subgraph problem and many other minimum or maximum ratio problems. We also provide an easy characterization of the ratio problems that are solvable with this procedure, as those that can be formulated as monotone integer programming problems. For those problems we describe the respective s,t-graph construction that follows from the formulation.

In a separate experimental study (Hochbaum et al., 2024) we show that the number of breakpoints IPC generates is in the range of $2-13$ even for datasets on millions of nodes and hundreds of million edges, which is typically less than 1% of the total number of breakpoints. This results in very fast running times that are orders of magnitude faster than those of the parametric flow procedure and recent state-of-the-art heuristics that do not produce optimal solutions.

To summarize, the main contributions here are:
1. The incremental parametric cut algorithm IPC that solves "monotone" ratio optimization problems in the complexity of a single min-cut.
2. A new, previously unknown, formulation of densest subgraph problem and its generalizations, that uses half of the number of arcs as compared to the known formulation.
3. An easy characterization of all ratio problems that are solved by IPC. Examples are given in Table 1.

1.1 Ratio Problems Solved with IPC

Notation. We consider the graph representation of the problems, firstly for undirected graphs corresponding to symmetric problems. Let $G=(V,E)$ denote an undirected graph with n denoting the number of nodes in V, and m denoting the number of edges in E. Every edge $[i,j]\in E$ has an associated weight $w_{ij}\geq 0$. Let the *weighted degree* of node $i\in V$ be $d_i=\sum_{[i,j]\in E} w_{ij}$. For $B_1,B_2\subseteq V$, let $C(B_1,B_2)=\sum_{[i,j]\in E,\ i\in B_1,\ j\in B_2} w_{ij}$ be the sum of weights of the edges between nodes in the set B_1 and those in set B_2. Let q_i denote a nonnegative cost value associated with each node, and u_i, or u'_i denote two types of values associated with each node, which could be positive or negative. Let the *degree volume* of a set of nodes S be $d(S)=\sum_{i\in S} d_i$, $q(S)=\sum_{i\in S} q_i$ and $U(S)=\sum_{i\in S} u_i$.

Some ratio problems are defined on directed graphs, $G=(V,A)$, where each arc $(i,j)\in A$ has an associated weight $w_{ij}\geq 0$. The weighted *outdegree* of a node i is $d^+_i=\sum_{j|(i,j)\in A} w_{ij}$, and the outdegree volume of a set of nodes S is $d^+(S)=\sum_{i\in S} d^+_i$.

A sample list of some of the ratio problems solved here is given in Table 1. The *Max density* problem is defined with weighted edges but unit weight on the nodes. This name refers more often to the special case of the unweighted problem where both edges weights are 1 and node weights are 1.

Many ratio problems appear in contexts where the size of optimal set is bounded. For example, the expansion ratio of a graph problem is $\min_{|S|\leq\frac{n}{2}}\frac{C(S,\bar{S})}{|S|}$. This added *size restriction* turns the problem NP-hard. The Cheeger constant problem is typically presented as $\min_{S\subset V}\frac{C(S,\bar{S})}{\min\{d(S),d(\bar{S})\}}$, which is equivalent to the size restricted ratio problem $\min_{d(S)\leq\frac{1}{2}d(V)}\frac{C(S,\bar{S})}{d(S)}$. The conductance problem is $\min_{\pi(S)\leq\frac{1}{2}\pi(V)}\frac{C(S,\bar{S})}{\pi(S)}$ where π_i is interpreted as the stationary probability of node i. We add here the $*$ to the name of the problem to indicate that there is no size restriction, and then the problem is polynomial time solvable. For the minimization problems, the entire set of nodes V is often the optimal solution of value 0. To avoid that trivial solution, the problem is typically solved on a subgraph of nodes V_1. For example Metis (Karypis and Kumar, 1998) has been used to identify a subgraph which is likely to contain the optimal solution for these problems and then the minimization is subject to $\emptyset\subset S\subseteq V_1$.

Table 1: A list of some of the ratio problems solved with the incremental parametric cut. *No size restriction.

Problem name	Objective
Max density	$\max_{S\subseteq V}\frac{C(S,S)}{\lvert S\rvert}$
Weighted max density	$\max_{S\subseteq V}\frac{C(S,S)}{q(S)}$
Ratio quadratic Knapsack	$\max_{S\subseteq V}\frac{C(S,S)+U(S)}{q(S)}$
HNC	$\max_{\emptyset\subset S\subset V}\frac{C(S,S)}{C(S,\bar{S})}$
HNC-equivalent	$\max_{\emptyset\subset S\subset V}\frac{d(S)}{C(S,\bar{S})}$
Max HNC-extension	$\max_{\emptyset\subset S\subset V}\frac{U(S)}{C(S,\bar{S})+U'(S)}$
Expansion ratio*	$\min_{\emptyset\subset S\subset V}\frac{C(S,\bar{S})}{\lvert S\rvert}$
Cheeger*/HNC	$\min_{\emptyset\subset S\subset V}\frac{C(S,\bar{S})}{d(S)}$
Conductance*	$\min_{\emptyset\subset S\subset V}\frac{C(S,\bar{S})}{q(S)}$

The problem HNC (Hochbaum Normalize Cut), also named NC' or SNC, was presented in (Sharon et al., 2006) as an NP-hard problem identical to the Normalized Cut (Shi and Malik, 2000), but shown polynomial time solvable in (Hochbaum, 2010). The same mistake was repeated in (Fortunato, 2010), who stated that Cheeger*/HNC, equation (22), $\min_{\emptyset\subset S\subset V}\frac{C(S,\bar{S})}{d(S)}$, is the normalized cut problem and NP-hard.

2 THEORETICAL BACKGROUND

2.1 Characterization of Polynomial Time Solvability: Monotone Ratio Problems

If the linearized problem can be formulated as monotone integer programming, IPM[1], then it is solvable with a min-cut procedure on an associated s,t graph, where the graph construction is uniquely mapped from the formulation, (Hochbaum, 2002).

IPM problems are classified as *monotone IP2* and *monotone IP3* where IP3 generalizes IP2. An integer program is a monotone IP2 if each constraint contains at most two of the variables that appear with opposite sign coefficients. An integer program is a monotone IP3 if each constraint contains at most two of the variables that appear with opposite sign coefficients and a third variable that appears in that constraint only. (There is an additional requirement that the "third variables" must have nonnegative coefficients in a minimization objective function, and nonpositive coefficients in a maximization objective function.) It is thus easy to recognize whether a formulation is monotone.

The formulation of monotone integer program for a set of n x-variables and a set of constraints involving a collection of pairs of variables A and a respective set of z-variables is,

$$
\begin{array}{lll}
\text{(IPM)}\ \max & \sum_{i=1}^{n} w_i x_i - \sum_{(i,j)\in A} e_{ij} z_{ij} & \\
\text{s.t.} & a_{ij}x_i - b_{ij}x_j \leq c_{ij} + z_{ij} & \forall\,(i,j)\in A \\
 & \ell_i \leq x_i \leq u_i,\ \text{integer} & \forall\, i\in V \\
 & z_{ij}\geq 0,\ \text{integer} & \forall\,(i,j)\in A.
\end{array}
$$

Here there is a restriction that the coefficients of e_{ij} in the objective function are nonnegative for maximization and non-positive for minimization.

Any IPM problem is equivalent to the following binary *s-excess* problem which is formulated on the variables $x_i = 1$ iff node i is in the optimal set S:

$$
\begin{array}{lll}
(s\text{-excess})\ \ \max & \sum_{j\in V} w_i x_i - \sum_{(i,j)\in A} u_{ij} z_{ij} & \\
\text{subject to} & x_i - x_j \leq z_{ij} & \text{for } (i,j)\in A \\
 & x_j \quad \text{binary } j=1,\ldots,n & \\
 & z_{ij} \quad \text{binary } (i,j)\in A. &
\end{array}
$$

The respective graph G_{st} is constructed as follows, (Hochbaum, 2002): We add nodes s and t to the graph G, with an arc from s to every positive weight node i,

[1]We use the acronym IPM rather than MIP so as not to confuse it with Mixed Integer Programming

of capacity $u_{si} = w_i$, and an arc from every negative weight node j to t of capacity $u_{jt} = -w_j$. Let this added set of arcs, adjacent to s and t (source node and sink node respectively) be denoted by A_{st}. The arcs of A each carry the capacity u_{ij} which is infinite if the constraint has only two variables. The graph G_{st} is then $(V \cup \{s,t\}, A \cup A_{st})$. The proof of the following lemma is given in (Hochbaum, 2002) and omitted here.

Lemma 1. *S^* is a set of maximum s-excess capacity in the original graph G if and only if S^* is the source set of a minimum s,t-cut in the associated graph G_{st}.*

We say that a ratio problem is a monotone integer program (IPM), if the corresponding λ-problem is IPM. For the λ-problem, the corresponding flow graph G_λ has arc capacities that are functions of the parameter λ. An s,t-graph is said to be a *parametric flow graph* if it has source-adjacent capacities that are monotone non-increasing with the parameter λ and the sink-adjacent capacities that are monotone non-decreasing with λ (or vice versa). For a λ-problem represented as a parametric flow graph, G_λ, the parametric cut procedure solves the λ-problem, for all values of the parameter. This is the case for all the problems listed in Table 1 and many more.

2.2 Parametric Cut, Nestedness and the "Continue" Property

Let the minimum cut for graph G_λ be $(S_\lambda, \bar{S}_\lambda)$ with S_λ the "source set" of the minimum cut and $\bar{S}_\lambda$ the "sink set". A property of the parametric flow graph is that as the values of λ are increasing, the source sets of the minimum cuts can only decrease, each a subset of the previous. Formally, for a monotone increasing sequence of p λ values, $\lambda_1 < \lambda_2 < \ldots < \lambda_p$, the corresponding optimal solutions, the source sets of the minimum cuts in the graph G_λ, satisfy $S_{\lambda_1} \supseteq S_{\lambda_2} \supseteq \ldots \supseteq S_{\lambda_p}$, and the respective sink sets satisfy $\emptyset = \bar{S}_{\lambda_0} \subseteq \bar{S}_{\lambda_1} \subseteq \ldots \subseteq \bar{S}_{\lambda_p}$. This property is called *nestedness* and is proved as a corollary of the parametric flow algorithms of (Gallo et al., 1989; Hochbaum, 1998; Hochbaum, 2008). As the value of the parameter λ increases, the respective cut solutions change when the sink set strictly increases. The values of the parameter where the change occurs are called *breakpoints*. Because of the nestedness the solution set changes by adding at least one node to the sink set, and therefore there are at most n breakpoints. For ℓ breakpoints, $\lambda'_1 < \lambda'_2 < \ldots < \lambda'_\ell$, the respective sink sets are strict subsets of each other: $\bar{S}_{\lambda'_1} \subset \bar{S}_{\lambda'_1} \subset \ldots \subset \bar{S}_{\lambda'_\ell}$.

There are two variants of the parametric cut procedure. The *fully parametric* variant generates all the breakpoints (see (Hochbaum, 2020a)). The *simple parametric* variant takes as input a sequence of values of λ, or a sequence of source adjacent capacities and sink adjacent capacities that are monotone non-increasing on one side, and monotone non-decreasing on the other, (Hochbaum, 2020b), and outputs the minimum cut solution for each of them. A property required of a min-cut max-flow algorithm in order for either the fully or simple parametric cut to work in the complexity of a single min-cut procedure, $T(n,m)$, is the *continue* property: Once an optimal solution has been found for one setting of the capacities, it is used as the initial solution for the new problem with updated, monotone, capacities. This is done while maintaining the labels and the invariant structure of the algorithm, which for HPF is called *normalized tree*. To-date, only push-relabel and HPF are max-flow min-cut algorithms that have the continue property. For HPF the routine **HPF-para-continue**(λ, S) is the part that takes a solution, which is the subset S in the related graph, and updated capacities corresponding to λ to find the optimal solution for the updated problem which is a subset of S.

The *continue* operation for HPF using monotonicity is referred to as **HPF-para-continue** and takes as input the solution source set for the value of λ previously used, that is guaranteed to contain the optimal ratio solution (because of nestedness, and the new value λ).

2.3 The Concave Envelope of the Breakpoints

For a general maximum ratio problem $\max_{\mathbf{x} \in \mathcal{F}} \frac{f(\mathbf{x})}{g(\mathbf{x})}$, we consider the graph that maps any value of $g(\mathbf{x}) = B$, so-called "budget", to the maximum value of $f(\mathbf{x}_B) = \arg\max_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x}) | g(\mathbf{x}) \leq B$, referred to as the "benefit". Finding those maximum benefits is in general NP-hard.

Consider the lower envelope of all the lines that have the entire collection of optimal solutions below them. This envelope, shown in red line segments in Figure 1, is concave piecewise linear and the points at which the line segment changes, are called *breakpoints* (marked by boxes in Figure 1).

The ratio value corresponding to each optimal point is the slope of the line connecting it to the origin. Hence the first, leftmost, breakpoint is also the optimal solution to the maximum ratio problem.

The properties of the concave envelope were studied, in the context of the dynamic evolution problem, in (Hochbaum, 2009). These properties include:

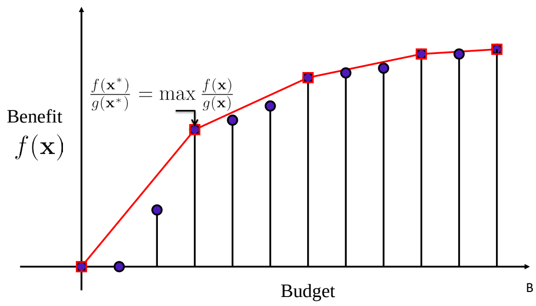

Figure 1: The concave envelope, the breakpoints and the ratio maximizing solution.

- The concave envelope and breakpoints are found with fully parametric cut procedure, (Hochbaum, 2020a).
- The breakpoints of the envelope correspond to the breakpoints of the respective parametric cut solutions, and the left derivative at the ith breakpoint is equal to the ith parameter breakpoint value λ_i'.
- At the breakpoints of the envelope the solutions are optimal.
- The first breakpoint – the smallest positive budget breakpoint – corresponds to the solution which attains the largest ratio of the benefit to the budget.
- The breakpoints correspond to solutions that are *nested* and their number is at most n, the number of variables, or nodes, in the respective graph.

For the respective minimization problems the envelope of the breakpoints is *convex*, and the first breakpoint corresponds to the solution that attains the smallest ratio of benefit to budget, see e.g. Figure 5.

2.4 Incremental Parametric Cut Procedure

Consider the general ratio maximization problem $\max_{\mathbf{x}\in\mathcal{F}} \frac{f(\mathbf{x})}{g(\mathbf{x})}$ where any feasible vector $\mathbf{x}'$ is associated with a subset of nodes in the associated graph, $S' = \{i \in V | x_i' = 1\}$.

The procedure starts with a set of nodes S^0 that is to contain the optimal ratio solution, which for the maximum density problem can be the entire graph, $S^0 = V$. The initial value of the parameter is $\lambda_0 = \frac{f(S^)}{g(S^0)}$. Solving the λ_0-problem either provides a solution with strictly higher ratio value, that is also a breakpoint solution, or else its value is 0 and therefore it is the maximum ratio solution. Because of the nested property, each subsequent solution set is strictly contained in the previous iteration's solution set. The value of the ratio is then updated and used as λ in the next iteration. Let S^0 be an initial feasible solution.

PROCEDURE INCREMENTAL PARAMETRIC $(f(), g(), S^0 \subseteq \mathcal{F}$,k=0).

Step 1: $\lambda_k = \frac{f(S^k)}{g(S^k)}$.

Step 2: **HPF-para-continue**(λ_k, S^k) to solve *improve*$(\lambda_k) = \max_{S\subseteq\mathcal{F}\cap S^k} f(S) - \lambda_k g(S)$.
Let $S^{k+1} = \arg\max_{S\subseteq\mathcal{F}\cap S^k} f(S) - \lambda_k g(S)$.

Step 3: If $\{improve(\lambda_k) > 0\}$ let $k := k+1$. Go to step 1, *else* stop. Output $S^* = S^k$.

We now prove the correctness of the procedure in that it visits a sequence of budget-decreasing breakpoints.

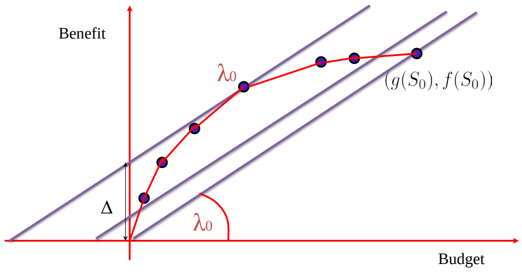

Figure 2: Identifying a breakpoint with $\lambda_0 = \frac{f(S_0)}{g(S_0)}$ subgradient, skipping over several breakpoints.

Lemma 2. *The optimal solution to* $\max_{\mathbf{x}\in\mathcal{F}} f(\mathbf{x}) - \lambda_0 g(\mathbf{x})$, $\mathbf{x}^1$, *is either a breakpoint on the concave envelope at a budget* $< g(\mathbf{x}^0)$ *and with strictly larger ratio than that of* $\mathbf{x}^0$, *or* $\mathbf{x}^1 = \mathbf{x}^0$ *and it is the maximum ratio solution.*

Proof. Consider the line equation $f(\mathbf{x}) = \lambda_0 g(\mathbf{x}) + \Delta$ where Δ the intercept of the line, of slope λ_0, on the vertical axis, as in Figure 2. Maximizing Δ is equivalent to $\max_{\mathbf{x}\in\mathcal{F}} f(\mathbf{x}) - \lambda_0 g(\mathbf{x}) = \Delta^*$. Therefore the line $f(\mathbf{x}) = \lambda_0 g(\mathbf{x}) + \Delta^*$ lies above all feasible solutions and is tangent to the concave envelope at breakpoint $\mathbf{x}^1$, where $\mathbf{x}^1 = \arg\max_{\mathbf{x}\in\mathcal{F}} f(\mathbf{x}) - \lambda_0 g(\mathbf{x})$. $\mathbf{x}^1$ is a breakpoint with a left subgradient equal to λ_ℓ and right subgradient equal to λ_r, such that $\lambda_\ell \geq \lambda_0 \geq \lambda_r$. □

The complexity of the incremental parametric cut procedure is that of a single min-cut HPF procedure on the graph, $T(n,m)$. More precisely, the complexity is $T(n,m) + O(qn)$ where q is the number of breakpoints visited. [2] As noted in the introduction, this number is very small in practice.

[2](Hochbaum, 2023) mistakenly stated that such a procedure visits adjacent breakpoints.

3 THE METHOD FOR WEIGHTED MAX DENSITY

Let the weighted maximum density problem, WMD, be given on a graph $G = (V,E)$ with positive edge weights u_{ij} and node weights q_i, $\max_{S\subseteq V} \frac{C(S,S)}{q(S)}$. The standard integer programming formulation of the problem has binary variables for each node $i \in V$: $x_i = 1$ if node i is selected in S and 0 otherwise, and binary variables for each edge $[i, j] \in E$, $y_{ij} = 1$ if both i and j are in S, and 0 otherwise. With this notation the formulation of WMD is,

$$
\begin{array}{lll}
\text{(WMD)} & \max & \frac{\sum_{[i,j]\in E} u_{ij} y_{ij}}{\sum_{j\in V} q_i x_i} \\
\text{subject to} & x_i \le y_{ij} & \text{for } [i,j] \in E \\
 & x_j \le y_{ij} & \text{for } [i,j] \in E \\
 & x_j & \text{binary } j \in V \\
 & y_{ij} & \text{binary } [i,j] \in E
\end{array}
$$

The graph corresponding to this IP2 formulation has $m+n$ nodes, one for each variable. We next present a general procedure for generating an equivalent compact (monotone) formulation for WMD and other ratio problems. Let d_i denote the weighted degree of node i in G: $d_i = \sum_{j|[i,j]\in E} u_{ij}$, and $d(S) = \sum_{i\in S} d_i$. It is easy to see that for any non-empty subset of nodes $S \subset V$, we have the identity $d(S) = 2C(S,S) + C(S,\bar{S})$. Therefore, $C(S,S) = \frac{1}{2}(d(S) - C(S,\bar{S}))$.

Hence, $\max_{S\subseteq V} \frac{C(S,S)}{q(S)} = \frac{1}{2} \max_{S\subseteq V} \frac{d(S)-C(S,\bar{S})}{q(S)}$ which is formulated as monotone integer program as well, with up to 3 variables per inequality using the same x-variables as in WMD, and "cut" variables z_{ij} that are equal to 1 if $i \in S$ and $j \in \bar{S}$ and zero otherwise:

$$
\begin{array}{lll}
\text{(WMD-compact)} & \max & \frac{\sum_{j\in V} d_i x_i - \sum_{[i,j]\in E} u_{ij} z_{ij}}{\sum_{j\in V} 2 q_i x_i} \\
\text{subject to} & x_i - x_j \le z_{ij} & \text{for } [i,j] \in E \\
 & x_j - x_i \le z_{ji} & \text{for } [i,j] \in E \\
 & x_j & \text{binary } j \in V \\
 & z_{ij}, z_{ji} & \text{binary } [i,j] \in E.
\end{array}
$$

The graph associated with the linearized problem, λ-WMD-compact, has one node for each x_i variable and two arcs for each edge in E resulting in a compact formulation on $n+2$ nodes and $2m+2n$ arcs.

Improved Formulation and Smaller Associated Graph. For WMD as well as for any ratio problem that includes only $C(S,S)$ along with linear terms, there is an even more efficient formulation that includes only one z_{ij} variable for every pair that has positive utility, instead of two. This results in a graph with $n+2$ nodes and $m+2n$ arcs which is about half of the number of arcs as compared to the formulation above.

The key is to observe that the problem can be represented on a directed graph $G = (V,A)$ where for each pair i and j with positive utility and $i < j$ there is one arc $(i, j) \in A$ from i to j.

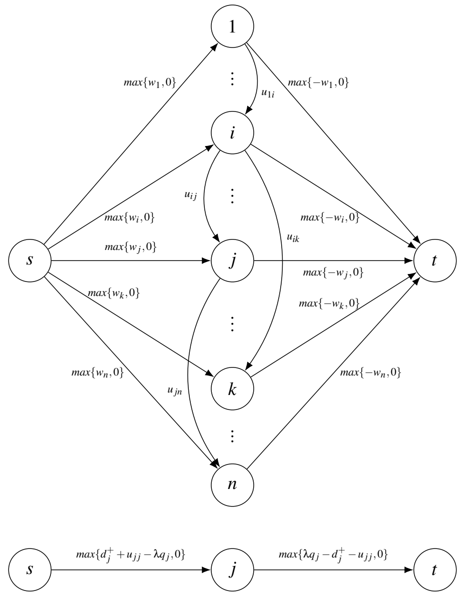

Figure 3: The flow graph G_λ for λ-WMD-compact1.

Let d_i^+ be the weighted out-degree of node i in G: $d_i^+ = \sum_{j|(i,j)\in A} u_{ij}$. Then, for any subset of nodes $S \subset V$, $d^+(S) = C(S,S) + C(S,\bar{S})$. Therefore (WMD-compact1) is an IPM formulation of WMD:

$$
\begin{array}{lll}
\text{(WMD-compact1)} & \max & \frac{\sum_{j\in V} d_i^+ x_i - \sum_{(i,j)\in A} u_{ij} z_{ij}}{\sum_{j\in V} q_i x_i} \\
\text{subject to} & x_i - x_j \le z_{ij} & \text{for } (i,j) \in A \\
 & x_j & \text{binary } \forall j \in V \\
 & z_{ij} & \text{binary } \forall (i,j) \in A.
\end{array}
$$

The objective function of the linearized ratio problem for the λ-question of (WMD-compact1) is, (λ-WMD) max $\sum_{j\in V} d_i^+ x_i - \sum_{(i,j)\in A} u_{ij} z_{ij} - \lambda \sum_{j\in V} q_i x_i$. The associated graph for this λ-WMD is given in Figure 3 which is obviously a parametric flow graph.

We conclude with an example of finding the densest subgraph with IPC, reported in (Hochbaum et al., 2024), in the dataset COM-YOUTUBE with $n = 1134890$ $m = 2987624$ from (Leskovec and Krevl, 2014). The running time of IPC for this dataset is 1.892 sec. The concave envelope of the breakpoints is shown in Figure 4.

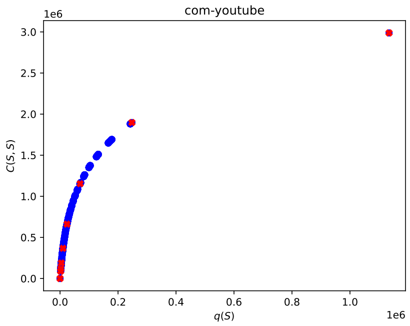

Figure 4: The concave envelope of all 1253 breakpoints, in blue, versus 9 breakpoints explored by IPC, in red. (Courtesy: A. Irribarra-Cortés).

4 APPLICATIONS OF IPC

We consider here the three ratio problems: HNC $\max_{\emptyset \subset S \subset V} \frac{C(S,S)}{C(S,\bar{S})}$, Cheeger's* $\min_{\emptyset \subset S \subset V} \frac{C(S,\bar{S})}{d(S)}$ and conductance*/HNC-extension $\min_{\emptyset \subset S \subset V} \frac{C(S,\bar{S})}{q(S)}$.

We first show, directly from the problem statement, that HNC is an IPM ratio problem. Then provide a transformation showing that HNC is equivalent to Cheeger*'s, and obviously conductance is a slight generalization of both. We then give the formulation for all three problems that leads to the parametric flow graph that is solved with IPC.

We first comment on the use of the constraint $\emptyset \subset S \subset V$ in ratio problems involving the cut $C(S,\bar{S})$. For such problems, unlike WMD, if unrestricted the solution will be the entire graph with cut value 0. In general that means that to solve such problems it is necessary to use *seeds* which are subsets of nodes so at least one belongs to the sink set and at least one belongs to the source set. For these problems, when they have size constraint, such as for Cheeger's, of the form $d(S) \leq \frac{d(V)}{2}$, the problems are NP-hard. To address the issue of the seeds and to solve the size restricted ratio problems heuristically one can choose to first identify a subset of the graph where the optimal subgraph may reside. This was done for example using the Metis graph partitioning heuristic of (Karypis and Kumar, 1998) by (Lang and Rao, 2004). Once the subgraph satisfying the size restriction is found, say V', the problem becomes $\min_{\emptyset \subset S \subset V'} \frac{C(S,\bar{S})}{d(S)}$.

Consider the integer programming formulation of HNC $\max_{\emptyset \subset S \subset V} \frac{C(S,S)}{C(S,\bar{S})}$ with edge weights w_{ij} and binary variables x_i, y_{ij} and z_{ij}. Let $x_i = 1$ if $i \in S$, $y_{ij} = 1$ if both i and j in S and $z_{ij} = 1$ if $i \in S$ $j \in \bar{S}$. The following is the linearized formulation λ-HNC:

$$
\begin{array}{lll}
(\lambda\text{-HNC}) & \max & \sum_{[i,j]\in E} w_{ij} y_{ij} - \lambda \sum_{j \in V} w_{ij} z_{ij} \\
\text{subject to} & & x_i \leq y_{ij} \quad \text{for } [i,j] \in E \\
& & x_j \leq y_{ij} \quad \text{for } [i,j] \in E \\
& & x_i - x_j \leq z_{ij} \quad \text{for } [i,j] \in E \\
& & x_j - x_i \leq z_{ji} \quad \text{for } [i,j] \in E \\
& & x_j \quad \text{binary } j \in V \\
& & z_{ij}, z_{ji}, y_{ij} \quad \text{binary } [i,j] \in E
\end{array}
$$

This monotone integer program maps into an associated graph on $m+n+2$ nodes and $2m+2n$ arcs. A compact formulation of HNC, equivalent to Cheeger's*, is given in the next lemma (proof omitted for lack of space):

Lemma 3. *The following two problems are equivalent and have the same optimal solutions:* $\max_{\emptyset \subset S \subset V} \frac{C(S,S)}{C(S,\bar{S})}$, *and* $\min_{\emptyset \subset S \subset V} \frac{C(S,\bar{S})}{d(S)}$.

Therefore solving HNC-extension, or conductance*, provides solutions to all three problems since setting $q_i = d_i$ is HNC or Cheeger's* problem. The problem $\min_{\emptyset \subset S \subset V} C(S,\bar{S}) - \lambda q(S)$ is formulated as follows.

$$
\begin{array}{lll}
(\lambda\text{-HNC-extension}) & \min & \sum_{[i,j]\in E} u_{ij} z_{ij} - \lambda \sum_{j \in V} q_i x_i \\
\text{subject to} & & x_i - x_j \leq z_{ij} \quad \text{for } [i,j] \in E \\
& & x_j - x_i \leq z_{ji} \quad \text{for } [i,j] \in E \\
& & x_j \quad \text{binary } j \in V \\
& & z_{ij}, z_{ji} \quad \text{binary } [i,j] \in E.
\end{array}
$$

The graph associated with this monotone integer program has $n+2$ nodes and $2m+2n$ arcs which improves on the number of nodes $m+n+2$ in the λ-HNC formulation.

To conclude we provide an example of solving Cheeger's* on a subgraph V' delivered by the Metis procedure, $\min_{\emptyset \subset S \subset V'} \frac{C(S,\bar{S})}{d(S)}$ applied to the dataset EGO-GPLUS of size $n = 107614$ $m = 12238285$, from (Leskovec and Krevl, 2014) (reported in (Hochbaum et al., 2024)). The convex envelope shown in Figure 5 illustrates the difference between the set of all breakpoints, generated with the fully parametric cut procedure, versus the set of points explored by IPC.

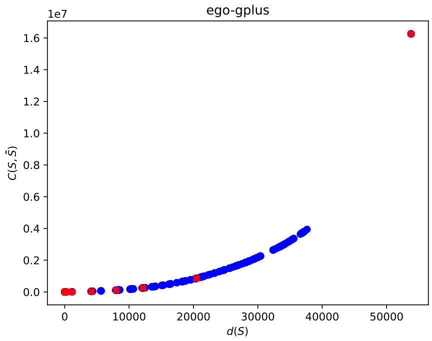

Figure 5: The convex envelope of all 291 breakpoints, in blue, versus 11 breakpoints explored by IPC, in red. (Courtesy: A. Irribarra-Cortés).

ACKNOWLEDGEMENTS

This research was supported in part by the AI4OPT institute NSF award 2112533.

REFERENCES

Angel, A., Koudas, N., Sarkas, N., Srivastava, D., Svendsen, M., and Tirthapura, S. (2014). Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *The VLDB journal*, 23:175–199.

Boob, D., Gao, Y., Peng, R., Sawlani, S., Tsourakakis, C., Wang, D., and Wang, J. (2020). Flowless: Extracting densest subgraphs without flow computations. In *Proceedings of The Web Conference 2020*, pages 573–583.

Charikar, M. (2000). Greedy approximation algorithms for finding dense components in a graph. In *International workshop on approximation algorithms for combinatorial optimization*, pages 84–95. Springer.

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.

Fratkin, E., Naughton, B. T., Brutlag, D. L., and Batzoglou, S. (2006). Motifcut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics*, 22(14):e150–e157.

Gallo, G., Grigoriadis, M. D., and Tarjan, R. E. (1989). A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55.

Goldberg, A. V. (1984). Finding a maximum density subgraph. *UC Berkeley manuscript*.

Goldberg, A. V. and Tarjan, R. E. (1988). A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940.

Harb, E., Quanrud, K., and Chekuri, C. (2022). Faster and scalable algorithms for densest subgraph and decomposition. *Advances in Neural Information Processing Systems*, 35:26966–26979.

Hochbaum, D. S. (1998). The pseudoflow algorithm and the pseudoflow-based simplex for the maximum flow problem. In *Integer Programming and Combinatorial Optimization: 6th International IPCO Conference Houston, Texas, June 22–24, 1998 Proceedings 6*, pages 325–337. Springer.

Hochbaum, D. S. (2002). Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. *European Journal of Operational Research*, 140(2):291–321.

Hochbaum, D. S. (2008). The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations research*, 56(4):992–1009.

Hochbaum, D. S. (2009). Dynamic evolution of economically preferred facilities. *European Journal of Operational Research*, 193(3):649–659.

Hochbaum, D. S. (2010). Polynomial time algorithms for ratio regions and a variant of normalized cut. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):889–898.

Hochbaum, D. S. (2020a). Hpf - hochbaum's pseudoflow. Accessed: May 28, 2022, https://riot.ieor.berkeley.edu/Applications/full-para-HPF/pseudoflow-parametric-cut.html.

Hochbaum, D. S. (2020b). Pseudoflow (simple) parametric maximum flow solver version 1.0. Accessed: May 28, 2022, https://riot.ieor.berkeley.edu/Applications/Pseudoflow/parametric.html.

Hochbaum, D. S. (2023). Unified new techniques for np-hard budgeted problems with applications in team collaboration, pattern recognition, document summarization, community detection and imaging. *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 1:365–372.

Hochbaum, D. S., Irribarra-Cortés, A., and Asín-Achá, R. (2024). Fast and optimal incremental parametric procedure for the densest subgraph problem: An experimental study. *UC Berkeley manuscript*.

Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999). Trawling the web for emerging cyber-communities. *Computer networks*, 31(11-16):1481–1493.

Lang, K. and Rao, S. (2004). A flow-based method for improving the expansion or conductance of graph cuts. In *Integer Programming and Combinatorial Optimization: 10th International IPCO Conference, New York, NY, USA, June 7-11, 2004. Proceedings 10*, pages 325–337. Springer.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Picard, J.-C. and Queyranne, M. (1982). A network flow solution to some nonlinear 0-1 programming problems, with applications to graph theory. *Networks*, 12(2):141–159.

Sharon, E., Galun, M., Sharon, D., Basri, R., and Brandt, A. (2006). Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.