Received: 19 January 2020 | Revised: 17 July 2020 | Accepted: 30 July 2020

DOI: 10.1002/net.21983

SPECIAL ISSUE ARTICLE

Applications and efficient algorithms for integer programming problems on monotone constraints

Dorit S. Hochbaum

Department of Industrial Engineering and Operations Research, University of California, Berkeley, California

Correspondence
Dorit S. Hochbaum, Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA.
Email: hochbaum@ieor.berkeley.edu

Funding information
This research was supported by the National Science Foundation, Grant/Award Number: CMMI-1760102.

Abstract

We present here classes of integer programming problems that are solvable efficiently and with combinatorial flow algorithms. The problems are characterized by constraints that have either at most two variables per inequality that appear with opposite sign coefficients, or have in addition a third variable that appears only in one constraint. Such integer programs, referred to here as monotone IP2 or IP3, are shown to be solvable in polynomial time for polynomially bounded variables. This article demonstrates the vast applicability of IP2 and IP3 as models for integer programs in multiple scenarios. Since the problems are easily recognized, the knowledge of their structure enables one to determine easily that they are efficiently solvable. The variety of applications, that previously were not known to be solved as efficiently, underlies the importance of recognizing this structure and, if appropriate, formulating problems as monotone IP2 or IP3. Additionally, if there is flexibility in the modeling of an integer programming problem, the formulation choice as monotone IP2 or IP3 leads to efficient algorithms, whereas slightly different modeling choices would lead to NP-hard problems.

KEYWORDS

image segmentation, integer programming, monotone inequalities, network flow, parametric cut, ratio problems

1 | INTRODUCTION

In modeling integer programs for problems that arise in applications, it is often the case that there is flexibility in the choice of the model and formulation. We present a class of integer programs that is powerful in that it can capture a vast range of models, and in that, it is solvable in polynomial time and with efficient combinatorial optimization techniques. We call these integer programming classes *monotone IP2* and *monotone IP3*. 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 IP2 or IP3. We demonstrate here, via a number of case examples, that monotone IP2 and IP3 are ubiquitous integer programming formulations in that they have multiple applications. Some of these applications are described in Sections 5, 6, 7, 8, and 9, showing how all of these are solved with one unified algorithm based on the minimum s, t-cut problem. These samples the vast spectrum of applications where some were previously known to be polynomial time solvable, but with ad-hoc techniques devised for the specific case, some were previously thought to be NP-hard, and others were previously formulated as NP-hard problems whereas an alternative formulation as a monotone integer program is a better model and is moreover efficiently solvable. Therefore, awareness of the potential of modeling problems as IP2 and IP3 leads to polynomial time algorithms and substantial advantages in terms of computational efficiency and eliminating the need for ad-hoc devised algorithms.

The origin of the term "monotone" relates to the lattice property of feasible solutions to a monotone integer program: For any two feasible solution vectors, the maximum (component-wise) and the minimum (component-wise) of the two vectors is feasible as well.

The formulation of (monotone) IP2 on n variables and a set of constraints related to a collection of pairs of variables A, where repetition of pairs is permitted, is

$$
\begin{aligned}
\text{(IP2)} \ \max \quad & \sum_{i=1}^{n} w_i x_i \\
\text{s.t.} \quad & a_{ij}x_i - b_{ij}x_j \geq c_{ij} \quad \forall \ (i,j) \in A \\
& \ell_i \leq x_i \leq u_i, \quad \text{integer} \quad \forall \ i \in V,
\end{aligned}
$$

where $a_{ij} \geq 0$, $b_{ij} \geq 0, \forall (i, j) \in A$. The objective function coefficients are unrestricted in sign, and therefore the problem can be written equivalently as a minimization problem.

We can assume without loss of generality that for a_{ij}, b_{ij} nonnegative the inequality relationship is "$\geq$" in all monotone inequalities: It is easy to see that the monotone inequality $ax - by \leq c$ can be transformed into $by - ax \geq -c$, and the monotone inequality $ax - by \geq c$ with $a \leq 0$, $b \leq 0$ can be transformed into $(-b)y - (-a)x \geq c$ with $-b \geq 0, -a \geq 0$.

A monotone IP2 is an NP-hard problem. This was proved by Lagarias [25] for the simultaneous diophantine approximation problem which is a special case of monotone IP2. However, Hochbaum and Naor showed, in Reference [12], that monotone IP2 is solved in pseudo-polynomial time that depends only on the range of the variables. A monotone IP2 on n variables and m constraints and $U = \max_{i=1}^{n} (u_i - \ell_i)$ was proved to be solved in the running time of the minimum cut problem on a graph on nU vertices, and $O(mU)$ arcs. This running time depends on the choice of the minimum cut procedure, which for the pseudoflow algorithm of [16], would be $O\left(mnU^2 \log \frac{n^2U}{m}\right)$. This means that monotone IP2 is *weakly* NP-hard. Note that in all interesting applications discovered to date the variables have small range - typically the variables are binary - and therefore $U = 1$. The same algorithm works with the same complexity for a nonlinear separable objective function. That is, for any nonlinear functions $w_j(x_j)$, the IP2 problem with the objective function max $\sum_{i=1}^{n} w_i(x_i)$, is solved with the same algorithm in $O\left(mnU^2 \log \frac{n^2U}{m}\right)$ time.

The formulation of (monotone) IP3 for a set of nx-variables and a set of constraints involving a collection of pairs of variables A and a respective set of z-variables is

$$
\begin{aligned}
\text{(IP3)} \ \max \quad & \sum_{i=1}^{n} w_i x_i - \sum_{(i,j)\in A} e_{ij} z_{ij} \\
\text{s.t.} \quad & a_{ij}x_i - b_{ij}x_j \leq c_{ij} + z_{ij} \quad \forall \ (i,j) \in A \\
& \ell_i \leq x_i \leq u_i, \ \text{integer} \quad \forall \ i \in V \\
& z_{ij} \geq 0, \ \text{integer} \quad \forall \ (i,j) \in A.
\end{aligned}
$$

Here there is a restriction that the coefficients of e_{ij} in the objective function are nonnegative. The algorithm that is presented for IP3 works also for a nonlinear extension where each $w_i x_i$ may be replaced by any nonlinear function $w_i(x_i)$, and each term $e_{ij} z_{ij}$ may be replaced by $e_{ij}(z_{ij})$ for $e_{ij}()$ a convex nondecreasing function. With this extension, IP3 is solved as a minimum cut problem on a related graph [1], as shown in Section 4.

Throughout we will omit the adjective of "monotone" for IP2 and IP3, and refer to these monotone problems simply as IP2 or IP3. We will also assume that the maximum range of the variables, $U = \max_{i=1}^{n} (u_i - \ell_i)$, is a polynomial bounded quantity.

We first present here the unified algorithm which relies on the equivalence of IP2 to a maximum closure problem [12], and the equivalence of IP3 to the dual of the minimum cost network flow problem [15]. It is then shown that both IP2 and IP3 are solved as a minimum cut problem on a related graph. Next, the effectiveness of the use of IP2 or IP3 modeling is demonstrated for selected applications.

One application is the maximum density subgraph problem presented in Section 5. This problem was previously known to be solvable in polynomial time [8], but the approach devised to solve it was ad-hoc. For this problem, as well as for several other applications presented, the objective function is a ratio of linear functions. It is shown in Section 5 that with "parametric" minimum cut, the same overall method applies, and with the same running time as for the linear objective function monotone IP2 or IP3. A second problem, the co-segmentation problem, that arises in the computer vision context, is discussed in Section 8. The formal problem is to identify and segment, for two separate images, a feature that they have in common. The formulation of the problem as IP3, in Reference [23], presented a significant leap in the accuracy and efficiency of solving the co-segmentation problem. The ratio region problem was investigated in Reference [5], in the context of image segmentation, and was shown to be polynomial only for the special case of planar graphs, with an ad hoc methodology. It is shown in Section 6, and in Reference [19], that the problem is IP3 and therefore solvable in polynomial time for all graphs, and more efficiently than the method devised for the case of planar graphs in Reference [5]. A clustering problem (HNC), that was incorrectly thought to be NP-hard, in Reference [29], was recognized to be efficiently solvable as a special case of IP3 in Reference [19]. This is discussed in Section 7. We conclude with an application that arose in the context of detecting potential nuclear threats and generating alerts, described in Section 9. The running time of the procedure, as IP3, is fast enough to enable its deployment in real time, as opposed to alternative techniques that used intractable integer programming models for the problem [9, 18].

1.1 | Outline of the paper

Preliminaries and notation are provided in Section 2. In Section 3, we present how to solve IP2 with the maximum closure problem, and in Section 4, how to solve IP3 with the *s*-excess problem. We then proceed to provide illustrations for the use of the IP2 and IP3 models for various challenging problems. In Section 5, on the maximum density problem, we provide background discussion on how to solve ratio problems with linearization, and in Section 5.3, how to solve the parametric cut or parametric flow problems that arise from the linearization without additional complexity. The problems of ratio region, HNC, co-segmentation, and nuclear threat detection are discussed in Sections 6, 7, 8, and 9, respectively. We conclude with final summary remarks in Section 10.

2 | PRELIMINARIES AND NOTATION

Let $G = (V, A)$ denote a directed graph with $n = |V|$ nodes and $m = |A|$ arcs. Every arc $(i, j) \in A$ has a capacity (or weight) u_{ij}. Let the set $S \subset V$ be a non-empty set of nodes. Its complement is $\bar{S} = V \backslash S$. The sets $(S, \bar{S})$ form a bi-partition referred to as a *cut*. The *capacity* of a cut $(S, \bar{S})$ is $C(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} u_{ij}$, the sum of arc capacities going from a node in S to a node in $\bar{S}$. Given a source node $s \in V$ and a sink node $t \in V$, an s, t-cut is a cut $(S, \bar{S})$ where $s \in S$ and $t \in \bar{S}$. The set S of an s, t-cut $(S, \bar{S})$ is known as the *source set* of the cut, and the set $\bar{S}$ is known as the *sink set* of the cut. The minimum s, t-cut problem on G is to find an s, t-cut that minimizes $C(S, \bar{S})$.

Let G_{st} be a graph (V_{st}, A_{st}), where $V_{st} = V \cup \{s, t\}$ and $A_{st} = A \cup A_s \cup A_t$ in which A_s and A_t are the source-adjacent and sink-adjacent arcs respectively. A *flow* vector $f = \{f_{ij}\}_{(i,j) \in A_{st}}$ is said to be *feasible* if it satisfies:

1 Flow balance constraints: for each $j \in V$, $\sum_{(i,j) \in A_{st}} f_{ij} = \sum_{(j,k) \in A_{st}} f_{jk}$ (ie, inflow(j) = outflow(j)), and
2 Capacity constraints: the flow on an arc is between the lower bound and upper bound capacity of the arc, that is, $0 \leq f_{ij} \leq u_{ij}$.

A *maximum flow* is a feasible flow f^* that maximizes the flow out of the source s (or into the sink), called the *value of the flow*. The value of the maximum flow is $\sum_{(s,i) \in A_s} f^*_{si}$. An s, t-cut in G_{st} (or *cut* for short) is a partition of V_{st} into $(S \cup \{s\}, \bar{S} \cup \{t\})$. The capacity of the cut is $C(S \cup \{s\}, \bar{S} \cup \{t\})$. A minimum s, t-cut is a cut of minimum capacity, referred to here as *min-cut*. It is well known (Ford-Fulkerson [6]) that the maximum value of the flow is equal to the capacity of a min-cut. Every algorithm known to date that solves the min-cut problem, also solves the maximum flow problem first. There are many known algorithms for the max-flow min-cut problems. We will denote the complexity of solving the max-flow min-cut problem on a graph with n nodes and m arcs by $T(n, m)$. One of the leading algorithms that will be used here is the pseudoflow algorithm HPF, [16], with complexity of $T(n, m) = O\left(mn \log \frac{n^2}{m}\right)$. That algorithm also solves, in the same complexity, the parametric max-flow min-cut problem (discussed in Section 5.3) that is at the core of several of the applications presented here.

3 | SOLVING IP2 AS A MAXIMUM/MINIMUM CLOSURE PROBLEM

3.1 | The maximum closure problem

We begin by defining a closed set on a graph.

Definition 3.1. Given a directed graph $G = (V, A)$, a subset of the nodes $D \subseteq V$ is closed, if for every node in D, its successors are also in D.

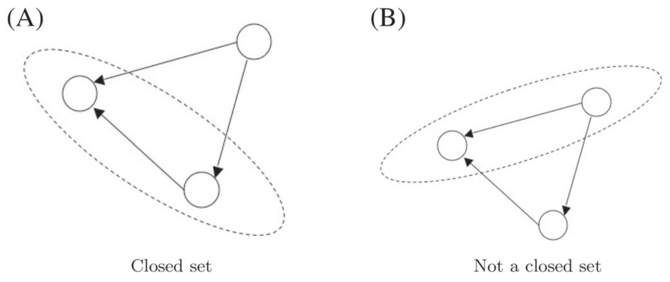

FIGURE 1 Closed set vs nonclosed set

For an illustration of a closed set versus nonclosed set see Figure 1. Consider a directed graph $G = (V, A)$ where every node $i \in V$ has a corresponding weight w_i (Figure 1). The *maximum closure problem* is to find a closed set $V' \subseteq V$ with maximum total weight. That is, the maximum closure problem is:

Problem Name: *Maximum closure*
Instance: *Given a directed graph $G = (V, A)$, and node weights (positive or negative) w_i for all $i \in V$.*
Optimization Problem: *find a closed subset of nodes $V' \subseteq V$ such that $\sum_{i \in V'} w_i$ is.*
maximum.

The maximum closure problem is formulated as an integer programming problem with the binary variables x_i, for each node $i \in V$ that takes the value 1 if node i is in the maximum closure, and 0 otherwise:

$$\max \quad \sum_{i \in V} w_i x_i$$

$$\text{s.t.} \quad x_i \leq x_j \quad \forall (i,j) \in A$$

$$x_i \in \{0, 1\} \quad \forall i \in V$$

The first set of constraints imposes the requirement that for every node i included in the set, its successor is also in the set. Observe that since every row has exactly one 1 and one -1, the constraint matrix is totally unimodular. Therefore, its linear relaxation formulation results in integer solutions. This structure also indicates that the problem is the dual of a flow problem.

It is noted that Johnson [24] seems to be the first researcher who demonstrated the connection between the maximum closure problem and the selection problem (ie, maximum closure on bipartite graphs), and showed that the selection problem is solvable by a max flow algorithm. Picard [26] demonstrated that a minimum cut algorithm on a related graph solves the maximum closure problem.

The closure problem is solved with a minimum cut procedure on a related graph. Let $V^+ \equiv \{j \in V | w_j > 0\}$, and $V^- \equiv \{i \in V | w_j \leq 0\}$. We construct an s, t-graph G_{st} as follows. Given the graph $G = (V, A)$ we set the capacity of all arcs in A equal to ∞. We add a source s, a sink t, set A_s of arcs from s to all nodes $i \in V^+$ (with capacity $u_{s,i} = w_i$), and set A_t of arcs from all nodes $j \in V^-$ to t (with capacity $u_{j,t} = |w_j| = -w_j$). The graph $G_{st} = \{V \cup \{s, t\}, A \cup A_s \cup A_t\}$ is a *closure graph* (a closure graph is a graph with a source, a sink, and with all finite capacity arcs adjacent only to either the source or the sink.) This construction is illustrated in Figure 2.

Claim 3.2. *If $(\{s\} \cup S, \{t\} \cup T)$ is a finite s, t-cut on $G_{s,t}$, then S is a closed set on G.*

Proof. Assume by contradiction that S is not closed. This means that there must be an arc $(i, j) \in A$ such that $i \in S$ and $j \in T$. This arc must be in the cut (S, T), and by construction $u_{i,j} = \infty$, which is a contradiction to the cut being finite. ■

Theorem 3.3. *If $(\{s\} \cup S, \{t\} \cup T)$ is an optimal solution to the minimum s, t-cut problem on $G_{s,t}$, then S is a maximum closed set on G.*

Proof.

$$\begin{aligned} C(\{s\} \cup S, \{t\} \cup T) &= \sum_{(s,i) \in A_{st}, i \in T} u_{s,i} + \sum_{(j,t) \in A_{st}, j \in S} u_{j,t} \\ &= \sum_{i \in T \cap V^+} w_i + \sum_{j \in S \cap V^-} -w_j \end{aligned}$$

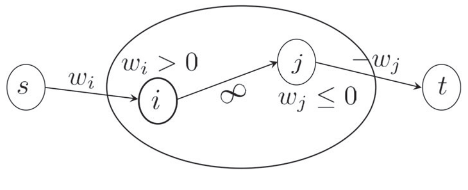

FIGURE 2 A visual representation of G_{st}

$$\begin{aligned} &= \sum_{i\in V^+} w_i - \sum_{i\in S\cap V^+} w_i - \sum_{j\in S\cap V^-} w_j \\ &= W^+ - \sum_{i\in S} w_i \end{aligned}$$

(Here $W^+ = \sum_{i\in V^+} w_i$, which is a constant.) This implies that minimizing $C(\{s\} \cup S, \{t\} \cup T)$ is equivalent to minimizing $W^+ - \sum_{i\in S} w_i$, which is in turn equivalent to $\max_{S\subseteq V} \sum_{i\in S} w_i$.

Therefore, any source set S that minimizes the cut capacity also maximizes the sum of the weights of the nodes in S. Since by Claim 3.2 any source set of an s, t-cut in $G_{s,t}$ is closed, we conclude that S is a maximum closed set on G. ■

3.2 | Variants/special cases

- In the minimum closure problem, we seek to find a closed set with minimum total weight. This can be solved by negating the weights on the nodes in G to obtain G^-, constructing $G^-_{s,t}$ just as before, and solving for the maximum closure. Under this construction, the source set of a minimum s, t-cut on $G_{s,t}$ is a minimum closed set on G.
- The selection problem is a special case of the maximum closure problem where the graph is bipartite.

3.3 | Monotone IP2 as a closure problem

The following process of "binarization" serves to convert the IP2 problem into an equivalent closure problem. Binarization begins by replacing each variable x_i in our original problem, by a sum of binary variables, $x_i = \ell_i + \sum_{p=\ell_i+1}^{u_i} x_i^{(p)}$, and imposing the restriction that if $x_i^{(p)} = 1$ then $x_i^{(p-1)} = 1$ for all $p = \ell_i + 1, \dots, u_i$. This is achieved by the inequalities $x_i^{(p)} \leq x_i^{(p-1)}$.

Next, for any monotone inequality $a_{ij}x_i - b_{ij}x_j \geq c_{ij}$ with $a_{ij}, b_{ij} \geq 0$,

$$a_{ij}x_i - b_{ij}x_j \geq c_{ij} \Leftrightarrow x_i \geq \frac{c_{ij} + b_{ij}x_j}{a_{ij}} \overset{x_i \text{ integer}}{\Longrightarrow} x_i \geq \left\lceil \frac{c_{ij} + b_{ij}x_j}{a_{ij}} \right\rceil.$$

Equivalently, if $x_j \geq p$, we must have $x_i \geq q(p) = \left\lceil \frac{c_{ij}+b_{ij}p}{a_{ij}} \right\rceil$. In terms of the newly defined binary variables, this is further equivalent to the inequalities $x_j^{(p)} \leq x_i^{(q(p))}$.

With these binary variables we rewrite IP2:

$$\begin{aligned} \max \quad & \sum_{i\in V} w_i\ell_i + \sum_{i\in V} w_i \sum_{p=\ell_i+1}^{u_i} x_i^{(p)} \\ \text{subject to} \quad & x_j^{(p)} \leq x_i^{(q(p))} \quad \forall\ (i,j) \in E \quad \text{for } p = \ell_j + 1, \dots, u_j \\ & x_i^{(p)} \leq x_i^{(p-1)} \quad \forall\ i \in V \quad \text{for } p = \ell_i + 1, \dots, u_i \\ & x_i^{(p)} \in \{0,1\} \quad \forall\ i \in V \quad \text{for } p = \ell_i + 1, \dots, u_i, \end{aligned}$$

where $q(p) \equiv \left\lceil \frac{c_{ij}+b_{ij}p}{a_{ij}} \right\rceil$.

Inequalities $x_i^{(p)} \leq x_i^{(p-1)}$ guarantee the restriction that $x_i^{(p)} = 1 \Rightarrow x_i^{(p-1)} = 1$ for all $p = \ell_i + 1, \dots, u_i$; Inequalities $x_j^{(p)} \leq x_i^{(q(p))}$ enforce that the monotone inequalities in the original problem are satisfied if $x_j^p = 1 \Rightarrow x_i^{q(p)} = 1$.

This resulting IP2 is the maximum closure problem on an s, t-graph G_{st} defined as follows. For $V^+ \equiv \{i \in V | w_i > 0\}$ and $V^- \equiv \{j \in V | w_j \leq 0\}$, the sets of nodes and arcs are constructed below.

Set of nodes:
Add a source s, a sink t, and nodes $x_i^{(\ell_i)}, x_i^{(\ell_i+1)}, \dots, x_i^{(u_i)}$ for each $i \in V$.
Set of arcs:

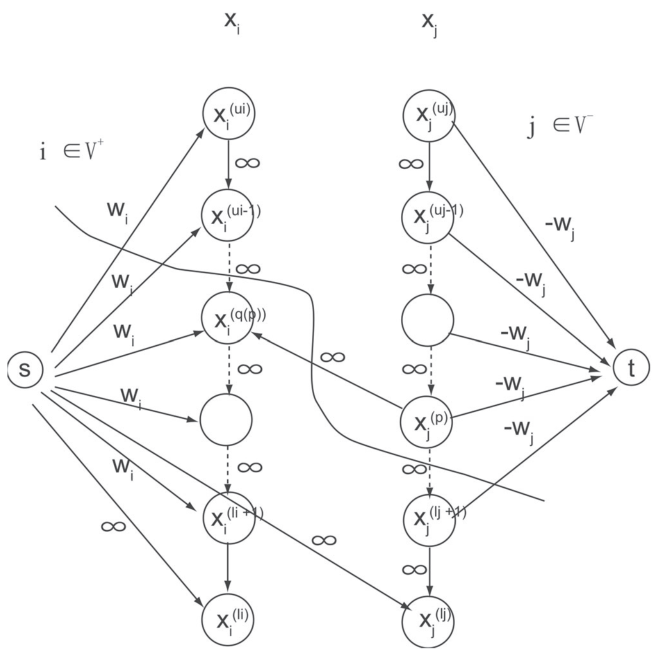

FIGURE 3 Illustration of G_{st} for Monotone IP2 and example of a finite s, t-cut

1. For any $i \in V^+$, connect s to $x_i^{(p)}, p = \ell_i + 1, \dots, u_i$, by an arc with capacity w_i.
2. For any $i \in V^-$, connect $x_i^{(p)}, p = \ell_i + 1, \dots, u_i$, to t by an arc with capacity $|w_i| = -w_i$.
3. For any $i \in V$, connect $x_i^{(p)}$ to $x_i^{(p-1)}$, $p = \ell_i + 1, \dots, u_i$, by an arc with capacity ∞, and connect s to $x_i^{(\ell_i)}$ by an arc with capacity ∞. We call this "path" of infinite capacities construction, the x_i-chain.
4. For any $(i, j) \in E$, connect $x_j^{(p)}$ to $x_i^{q(p)}$ by an arc with capacity ∞ for all $p = \ell_i + 1, \dots, u_j$. (Note that for situations where $q(p) > u_i$, we must have $x_j^{(p)} = 0$. Therefore, we can either remove the node $x_j^{(p)}$ by redefining a tighter upper bound for x_j, or simply fix $x_j^{(p)}$ to be zero by introducing an arc from $x_j^{(p)}$ to t with capacity ∞.)

This construction is illustrated in Figure 3.

We refer to the transformation of IP2 to an equivalent closure problem "binarizing," since it replaces the real coefficients by 0, 1 and −1 coefficients.

3.4 | Remarks

- It should be noted that the maximum closure problem is defined on an s, t-graph with $2 + \sum_{i \in V}(u_i - \ell_i)$ nodes. The size of the graph is not a polynomial function of the length of the input. Therefore, the maximum closure problem corresponding to the monotone IP2 is solved in pseudo-polynomial time.
- For the minimization version of IP2, we can construct the s, t-graph in the same way and define closure with respect to the sink set instead.
- An IP2 on a *nonlinear* separable objective is solved with the same algorithm, in the same complexity. For a nonlinear objective, min $\sum_{j=1}^{n} w_j(x_j)$, the only modification required is in defining the weight associated with $x_j^{(p)}$ to be $w_j(p) - w_j(p-1)$. This weight could be positive or negative.

4 | SOLVING IP3 AS AN *s*-EXCESS PROBLEM

The s-excess problem is a generalization of the closure problem in that it allows us to violate the closure constraints, with penalties. The process of solving IP3 relies on an analogous binarizing process which transforms an IP3 problem to an s-excess problem, which in turn, is also solvable as a minimum s, t-cut on a related graph. The process of binarizing leads to the following theorem:

Theorem 4.1. [15] *A monotone* IP3 *is equivalent to an integer program in $O(nU)$ binary variables over $O(mU)$ totally unimodular constraints.*

In Reference [15], it is also shown how to solve, with the same complexity, also the more general convex monotone IP3 problem with an objective function which is convex and separable in the variables x_j and z_{ij}.

The process of "Binarizing" when applied to an IP3 results in an equivalent problem in binary variables with constraint coefficients that are in $\{0, -1, 1\}$ and inequalities of the form:

$$x_i - x_j \leq 0 \quad \text{or}$$

$$x_i - x_j \leq z_{ij}.$$

More precisely, "Binarizing" converts any monotone IP3 to an instance of the *minimum s-excess problem*. That problem was introduced in Reference [13] in the context of the pseudoflow algorithm. The problem is defined as follows:

Problem Name: *Minimum s-Excess*
Instance: *Given a directed graph $G = (V, A)$, node weights (positive or negative) w_i for.*
all $i \in V$, and nonnegative arc weights e_{ij} for all $(i, j) \in A$.
Optimization Problem: *Find a subset of nodes $S \subseteq V$ such that.*
$\sum_{i \in S} w_i + \sum_{i \in S, j \in \bar{S}} e_{ij}$ is minimum.

The minimum s-excess problem is a generalization of the maximum closure, or rather the respective minimum closure, problem. The constraints of the s-excess problem are a relaxation of the closure constraints, in that a violation of the closure constraint $x_i \leq x_j$ is permitted at a penalty of e_{ij} per unit violation. Indeed, when the quantities e_{ij} are infinite, the problem reduces to the closure problem. The s-excess problem is formulated as a binary optimization problem

$$\begin{aligned} \text{(s-excess)} \quad & \min \sum_{j \in V} w_j x_j + \sum_{(i,j) \in A} e_{ij} z_{ij} \\ \text{subject to} \quad & x_i - x_j \leq z_{ij} \quad \text{for } (i,j) \in \text{A} \\ & x_j \ \text{binary} \ \ j = 1, \ldots, n \\ & z_{ij} \ \text{binary} \ \ (i,j) \in A. \end{aligned}$$

Although the constraints of the type $x_i - x_j \leq 0$ do not seem to appear in this formulation, these are written as $x_i - x_j \leq z_{ij}$ where the cost coefficient e_{ij} of the respective variable z_{ij} (and the capacity of the corresponding arc) is infinite. It is shown next to that the minimum s-excess set in a graph G is the source set of a minimum cut in an associated graph G_{st}, defined as follows. We add nodes s and t to the graph G, with an arc from s to every negative weight node i (with capacity $u_{si} = -w_i$), and an arc from every positive weight node j to t (with capacity $u_{jt} = w_j$).

Lemma 4.1. *S^* is a set of minimum s-excess capacity in the original graph G if and only if S^* is the source set of a minimum cut in an associated graph G_{st}.*

Proof. Recall the notation, $V^+ \equiv \{i \in V | w_i > 0\}$, and $V^- \equiv \{j \in V | w_j < 0\}$. Let $(\{s\} \cup S, \{t\} \cup T)$ define an s, t-cut on G_{st}. Then the capacity of this cut is given by

$$\begin{aligned} C(\{s\} \cup S, \{t\} \cup T) &= \sum_{(s,i) \in A_{st}, i \in T} u_{s,i} + \sum_{(j,t) \in A_{st}, j \in S} u_{j,t} + \sum_{i \in S, j \in T} e_{ij} \\ &= \sum_{i \in T \cap V^-} - w_i + \sum_{j \in S \cap V^+} w_j + \sum_{i \in S, j \in T} e_{ij} \\ &= \sum_{i \in V^-} - w_i - \sum_{j \in S \cap V^-} - w_i + \sum_{j \in S \cap V^+} w_j + \sum_{i \in S, j \in T} e_{ij} \\ &= W^- + \sum_{j \in S} w_j + \sum_{i \in S, j \in T} e_{ij}, \end{aligned}$$

where W^- is the sum of all negative weights in G, which is a constant. Therefore, minimizing $C(s \cup S, t \cup T)$ is equivalent to minimizing $\sum_{j \in S} w_j + \sum_{i \in S, j \in T} e_{ij}$, and we conclude that the source set of a minimum s, t-cut on G_{st} is also a minimum s-excess set of G. ∎

It was shown, in the previous section, how to represent a monotone inequality with two variables, $ax_i - bx_j \leq c$, as a graph in which any feasible finite cut corresponds to a feasible solution. Similarly, for a monotone inequality on three variables, $ax_i - bx_j \leq c_{ij} + z_{ij}$, the construction of the x_i-chain is exactly the same as in the case of IP2. The difference is in the setting of the arcs and the capacities assigned to these arcs. The algorithm presented here applies for the more general nonlinear extension of IP3 where each $w_i x_i$ may be replaced by any nonlinear function $w_i(x_i)$, and each term $e_{ij}z_{ij}$ may be replaced by $e_{ij}(z_{ij})$ for $e_{ij}()$ a convex nondecreasing function.

Consider the monotone inequality of the type $ax_i - bx_j \leq c_{ij} + z_{ij}$. For notational convenience and without risk of ambiguity we let, in the following discussion, $c = c_{ij}$ and $z = z_{ij}$. Let $e_{ij}(z)$ be the (convex nondecreasing) cost function associated with z, and $0 \leq z \leq \gamma_{ij}$.

The inequality is equivalent to the set of implications:

$$\text{If } x_i \geq p \text{ then } x_j \geq \left\lceil \frac{ap - c - z}{b} \right\rceil.$$

We denote

$$j(p, z) \equiv \left\lceil \frac{ap - c - z}{b} \right\rceil.$$

Several more notational conventions will streamline the exposition: Instead of listing the inequalities, we list the capacities of the arcs in the associated graph G. We let the capacity of arc (p, q) be denoted by $c(p, q)$. Stating that $c(p, q) = \infty$ will be taken to be equivalent to generating an inequality

$$x_i^{(p)} \leq x_j^{(q)}.$$

Stating that $c(p, q) = K$ will be taken to be equivalent to generating an inequality

$$x_i^{(p)} \leq x_j^{(q)} + z(p, q)$$

where the cost of the binary variable $z(p, q)$ in the objective function is K. Therefore the objective function term corresponding to the z-variables is

$$\sum_{(i,j)\in E_1} \sum_{p_i=\ell_i}^{u_i} \sum_{p_j=\ell_j}^{u_j} c(p_i, p_j) z(p_i, p_j).$$

We will demonstrate that the capacities and costs indeed correspond to the correct cost of the z variables.

The description is restricted to the case where $a \leq b$. The procedure for $a > b$ follows similar principles and will be omitted. This assumption allows us to conclude that when the value of x_i is increased by 1, the value of $j(p, z)$ can increase by one unit at most.

4.1 | z is binary

We first describe the procedure for generating inequalities and arcs for z binary, that is, $\gamma_{ij} = 1$. The challenge here is to track whether the setting of z to be equal to 1 does indeed relax the inequality. In general, there should be an arc $(p, j(p, 0))$ of cost $e_{ij} = e_{ij}(1)$. We need however to avoid the situation when $j(p, 0) = j(p-1, 0)$ and charge twice the value e_{ij} when $x_i = p$ and $x_j = j(p, 0) - 1$. In this case both $x_i^{(p)}$ and $x_i^{(p-1)}$ are in the source set of the cut and $x_i^{(j(p,0))}$ is in the sink set. So only arc $(p-1, j(p-1, 0))$ exists and the charge of e_{ij} is applied once.

```
Generate arcs ($ax_i - bx_j \leq c + z$, 1, $e_{ij}()$)
Let $p$ be smallest so that $p \geq \ell_i$ and $j(p, 0) \geq \ell_j + 1$.
If $p > u_i$, then stop: Output “constraint is always satisfied”.
If $j(p, 1) > u_j$, then stop: “problem is infeasible”.
Set $c(p, j(p, 1)) = \infty$.
If $j(p, 0) > j(p, 1)$ then $c(p, j(p, 0)) = e_{ij}$.
while $p \leq u_i - 1$, do
    $p \leftarrow p + 1$.

    $c(p, j(p, 1)) = \infty$.

    If $j(p, 0) > j(p, 1)$ and $j(p, 0) > j(p-1, 0)$ then $c(p, j(p, 0)) = e_{ij}$.
    Else, continue
end
```

Proof of correctness. Let $x_i = p$ and $x_j = q$ in some solution.

If $q < j(p, 1)$ then the solution is infeasible as it violates the inequality $ax_i - bx_j \leq c + z$. The arc $(p, j(p, 1))$ is in the cut and it has infinite capacity, as required. Thus such a solution has infinite s-excess value.

If $q = j(p, 1) < j(p, 0)$ then $c < ap + b \leq c + 1$. Therefore, the value of z must be equal to 1 with a charge of e_{ij} to the objective function. The only arc in the cut between the x_i and the x_j chains is from the lowest node $p' \leq p$ in the x_i-chain with $j(p', 1) < j(p', 0) = j(p, 0)$. This arc is of capacity e_{ij} and thus $\sum_{p_i=\ell_i}^{u_i} \sum_{p_j=\ell_j}^{u_j} c(p_i, p_j) z(p_i, p_j) = e_{ij}$.

If $q > j(p, 1)$ then, since $a \leq b$, $j(p, 0) = j(p, 1) + 1$, $q \geq j(p, 0)$ in which case, there is no arc in the cut associated with this inequality. ■

4.2 | z is integer

The generation of arcs/inequalities for the case of z integer obviously generalizes the case of z binary.

Let $x_i = p$ and define q_p to be the smallest value such that $j(p, q) = j(p, 0) - 1$. That is,

$$\left\lceil \frac{ap - c - q_p}{b} \right\rceil = \frac{ap - c - q_p}{b} = \left\lceil \frac{ap - c}{b} \right\rceil - 1.$$

We will introduce arcs adjacent to $x_i = p$ sequentially for $p = \ell_i, \ldots, u_i$. The nodes in the x_j-chain will have at any point in the process a subset of arcs from nodes $p = \ell_i, \ldots, p - 1$ with certain incapacity already generated. We denote the total incumbent sum of incapacity into node p' by $C(p')$.

```
(Generate arcs $(ax_i - bx_j \leq c + z, \gamma_{ij}, e_{ij}())$
Let $p$ be smallest so that $p \geq \ell_i$ and $j(p, 0) \geq \ell_j + 1$.
If $p > u_i$, then stop: Output "constraint is always satisfied".
If $j(p, \gamma_{ij}) > u_j$, then stop: "problem is infeasible".
while $p \leq u_i$, do.
Let $q_p$ be smallest so that $j(p, q_p) = j(p, 0) - 1$. Set $c(p, j(p, 0)) = e_{ij}(q_p) - C(j(p, 0))$, $k = 1$
    until $j(p, 0) - k = \max\{\ell_j, j(p, \gamma_{ij})\} + 1$, do:
        set $c(p, j(p, 0) - k) = e_{ij}(kb + q_p) - e_{ij}((k-1)b + q_p) - C(j(p, 0) - k)$.
        $k \leftarrow k + 1$.

    end
Let $c(p, j(p, \gamma_{ij})) = \infty$.
        $p \leftarrow p + 1$.

end
```

Since the functions $e_{ij}()$ are convex and monotone nondecreasing then it is guaranteed that every capacity generated is nonnegative. Another property of the generated capacities is that the total sum of capacities in a cut between x_i and x_j is precisely $e_{ij}(ax_i - bx_j - c)$.

Proof of correctness: We first establish the nonnegativity of the capacities.

When done with the assignment of arc capacities adjacent to node $x_i = p$ then the new total capacities are

$$\begin{aligned}
C(j(p, 0)) &= e_{ij}(q_p) \\
C(j(p, 0) - 1) &= e_{ij}(b + q_p) - e_{ij}(q_p) \\
C(j(p, 0) - 2) &= e_{ij}(2b + q_p) - e_{ij}(b + q_p) \\
&\cdot \\
&\cdot \\
&\cdot \\
&\cdot \\
C(j(p, 0) - k) &= e_{ij}(kb + q_p) - e_{ij}((k-1)b + q_p).
\end{aligned}$$

First we show the nonnegativity of the capacities of all arcs. Suppose this is true by induction for all arcs adjacent to nodes $x_i = \ell_i, \ldots, p$ and consider the capacities of arcs adjacent to node $x_i = p + 1$. There are two cases:

$$j(p, 0) = j(p + 1, 0), \quad \text{or}$$

$$j(p, 0) < j(p + 1, 0).$$

In the first case there are arcs of total capacity $e_{ij}(q_p)$ adjacent to $j(p, 0)$ when we assign the capacity to the arc $(p + 1, j(p + 1, 0))$. Necessarily $q_{p+1} > q_p$ (the difference is in fact a units) thus recalling that $e_{ij}()$ are monotone nondecreasing,

$e_{ij}(q_p) \le e_{ij}(q_{p+1})$.Therefore $e_{ij}(q_{p+1}) \ge C(j(p, 0))$. Also, for any k,

$$C(p, j(p,0) - k) = e_{ij}(kb + q_p) - e_{ij}((k-1)b + q_p) \le e_{ij}(kb + q_{p+1}) - e_{ij}((k-1)b + q_{p+1}).$$

Thus all arc capacities are nonnegative.

Consider now the second case where $j(p, 0) < j(p+1, 0)$. The incumbent capacity adjacent to $j(p+1, 0)$ is 0. Also $q_{p+1} + b > q_p$ thus for all k,

$$C(p, j(p,0) - k) \le e_{ij}((k-1)b + q_p) - e_{ij}((k-2)b + q_p) \le e_{ij}(kb + q_{p+1}) - e_{ij}((k-1)b + q_{p+1}),$$

Thus all arc capacities are nonnegative.

Let $x_i = p$ and $x_j = q$ in some solution.

If $q < j(p, \gamma_{ij})$ then the solution is infeasible as it violates the inequality in its most relaxed form $ax_i - bx_j \le c + \gamma_{ij}$. The arc $(p, j(p, \gamma_{ij}))$ is then in the cut and it has infinite capacity, as required. Thus such a solution has infinite s-excess value.

If $q \ge j(p, 0)$ then the inequality is satisfied and there is no arc associated with this inequality (and with z_{ij}) in the cut.

So suppose that $j(p, 0) > q \ge j(p, \gamma_{ij})$. In this case, the value of z_{ij} in an associated optimal assignment is $\lceil ap - bq - c \rceil$.

Claim 4.2. $z_{ij} = q_p + b(j(p, 0) - q - 1)$.

Proof. Assume for convenience that c is integer, then $q_p = ap - c - b(j(p, 0) - 1)$. Thus,

$$q_p + b(j(p,0) - q - 1) = ap - c - bq.$$

■

Since the total capacity adjacent to nodes $x_j = q+1, \ldots, j(p, 0)$ is $C(p, x_j)$, the total charge for arcs in the cut is $\sum_{x_j=q+1}^{j(p,0)} C(p, x_j)$. This sum is a telescopic sequence adding up to $e_{ij}(q_p + b(j(p, 0) - q - 1))$:

$$\begin{aligned}\sum_{x_j=q+1}^{j(p,0)} C(p, x_j) &= e_{ij}(q_p)\\ &+ e_{ij}(b + q_p) - e_{ij}(q_p)\\ &+ e_{ij}(2b + q_p) - e_{ij}(b + q_p)\\ &.\\ &.\\ &.\\ &.\\ &+ e_{ij}((j(p,0) - q - 1)b + q_p) - e_{ij}((j(p,0) - q - 2)b + q_p)\\ &= e_{ij}((j(p,0) - q - 1)b + q_p).\end{aligned}$$

So the total charge for arcs in the cut is precisely $e_{ij}(z_{ij})$ as required.

4.2.1 | An example

We present an example of the case when $a_{ij} = b_{ij} = 1$.

We define recursively the excess unit increments of the functions e_{ij} (a discrete equivalent of the second derivative):

$$\begin{aligned}\Delta_{ij}(0) &= 0\\ \Delta_{ij}(1) &= e_{ij}(1) - e_{ij}(0)\\ \Delta_{ij}(2) &= e_{ij}(2) - e_{ij}(1) - \Delta_{ij}(1)\\ &.\\ &.\\ &.\\ &.\\ \Delta_{ij}(\gamma_{ij}) &= e_{ij}(\gamma_{ij}) - e_{ij}(\gamma_{ij} - 1) - \Delta_{ij}(\gamma_{ij} - 1).\end{aligned}$$

Figure 4 depicts the generated graph for an inequality $x_i - x_j \le 2 + z_{ij}$ where $\gamma_{ij} = 3$. The illustration shows the arcs associated with the lowest nodes in the x_i chain. Note that the arcs for the first node, ℓ_i, follow a different pattern from that of the other

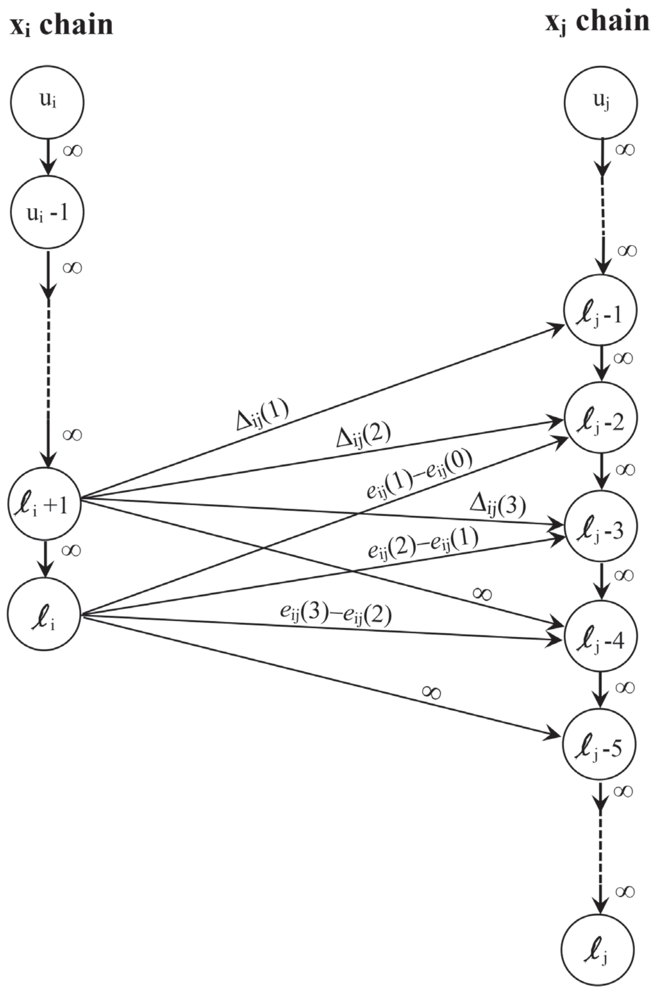

FIGURE 4 Arcs and their capacities when $e_{ij}()$ are convex

nodes. That is because the higher valued nodes in the chain are guaranteed that when they are in the source set, then so are all the nodes under them and thus the arcs adjacent to a particular valued node in the x_j chain is also originating from lower valued nodes. Thus such an arc contributes only to the incremental cost.

In this example:

$c(\ell_i, \ell_i - 2) = e_{ij}(1) - e_{ij}(0)$. If this arc is in the cut then $z_{ij} \geq 1$.

$c(\ell_i, \ell_i - 3) = e_{ij}(2) - e_{ij}(1)$. If this arc is in the cut then $z_{ij} \geq 2$.

$c(\ell_i, \ell_i - 4) = e_{ij}(3) - e_{ij}(2)$. If this arc is in the cut then $z_{ij} \geq 3$.

$c(\ell_i, \ell_i - 5) = \infty$. It is infeasible for x_j to be $\leq \ell_i - 5$.

$c(\ell_i + 1, \ell_i - 1) = \Delta_{ij}(1) = e_{ij}(1) - e_{ij}(0)$. If this arc is in the cut then $z_{ij} \geq 1$.

$c(\ell_i + 1, \ell_i - 2) = \Delta_{ij}(2)$. If this arc is in the cut then also the arc $(\ell_i, \ell_i - 2)$ and the arc $(\ell_i + 1, \ell_i - 1)$ are in the cut and $z_{ij} \geq 2$. The total capacity of these three arcs is $\Delta_{ij}(1) + \Delta_{ij}(2) + e_{ij}(1) - e_{ij}(0) = e_{ij}(2) - e_{ij}(1) + e_{ij}(1) - e_{ij}(0) = e_{ij}(2) - e_{ij}(0)$ as required.

$c(\ell_i + 1, \ell_i - 3) = \Delta_{ij}(3)$. If this arc is in the cut then $z_{ij} \geq 3$.

$c(\ell_i + 1, \ell_i - 4) = \infty$. It is infeasible for $x_j \leq \ell_i - 4$ if $x_i \geq \ell_i + 1$.

5 | MAXIMUM DENSITY SUBGRAPH

The *density* of a graph is the ratio of the number of edges to the number of nodes in the graph. Given a graph $G = (V, E)$ the maximum density subgraph problem is to identify a non-empty subset of nodes $H \subseteq V$ such that $\frac{|E(H)|}{|H|}$ is maximized, where $E(H) \subseteq E$ is the set of edges with both endpoints in H. To simplify the notation, let $n = |V|$, the number of nodes in G, and let $m = |E|$, the number of edges in G. This problem was shown to be solvable in polynomial time by Goldberg [8] with an ad-hoc algorithm. Here we show that formulating the problem as an integer program yields a monotone IP2 and hence the respective polynomial time algorithm. Generalizations of the problem with weighted edges and weighted nodes are likewise solvable in polynomial time.

One issue that arises in the maximum density subgraph problem is that the objective function is a *ratio*. This does not affect the solution method as monotone IP2 or IP3. It does however add the use of *parametric cut* as we see here and with other ratio problems such as the problem of HNC (Hochbaum Normalized Cut) in Section 7. We first address the *linearization* of ratio problems.

5.1 | Linearizing ratio problems

A general approach for maximizing a fractional (or as it is sometimes called, geometric) objective function over a feasible region $\mathcal{F}$, $\min_{x\in F}\frac{f(x)}{g(x)}$, is to reduce it to a sequence of calls to an oracle that provides the yes/no answer to the *λ-question*:

Is there a feasible subset $x \in \mathcal{F}$ such that $(f(x) - \lambda g(x) < 0)$?

If the answer to the λ-question is *yes* then the optimal solution to the original problem has a value smaller than λ. Otherwise, the optimal value is greater than or equal to λ. A standard approach is then to utilize a binary search procedure that calls for the λ-question $O\left(\log\frac{u}{\ell}\right)$ times in order to solve the problem, where u is an upper bound on the value of the numerator and ℓ a lower bound on the value of the denominator.

Therefore, if the linearized version of the problem, that is the λ-question, is solved in polynomial time, then so is the ratio problem. One way of solving the λ-question is to solve the respective optimization problem $\min_{x\in\mathcal{F}}(f(x) - \lambda g(x))$. One can then employ a binary search procedure that makes calls to this linear optimization. Note that the number of calls to this linear optimization is not strongly polynomial but rather depends on the logarithm of the magnitude of the numbers in the input. We show that if the linearized optimization problem is a monotone integer program IP2, or IP3 with the λ parameter appearing only in the coefficients of the x variables, then instead of using binary search we employ a *parametric cut* procedure which is much more efficient and requires the same complexity as a single minimum cut procedure.

5.2 | Solving the maximum density subgraph problem

We formulate here a generalized version of the maximum density problem on a graph $G = (V, E)$ in which the edges have weights w_{ij} for $[i, j] \in E$ and the nodes have weight d_j for $j \in V$. For $S \subset V$ let x_i and y_{ij} be the binary variables:

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \in \bar{S}. \end{cases}$$

and

$$y_{ij} = \begin{cases} 1 & \text{if } i,j \in S \\ 0 & \text{otherwise.} \end{cases}$$

With this notation the objective function is max $\frac{\sum_{[i,j]\in E} w_{ij} y_{ij}}{\sum_{j\in V} d_j x_j}$. The formulation of the linearized maximum density problem corresponding to the λ question is

$$\begin{aligned}
(\lambda\text{-MD}) \max \; & \sum_{[i,j]\in E} w_{ij} y_{ij} - \sum_{j\in V} \lambda d_j x_j \\
\text{subject to } \; & y_{ij} \le x_i \quad \text{for all } [i,j] \in E \\
& y_{ij} \le x_j \quad \text{for all } [i,j] \in E \\
& x_j \;\; binary \;\; j \in V \\
& y_{ij} \;\; binary \;\; i,j \in V.
\end{aligned}$$

This formulation is a monotone IP2. The λ-question is hence solvable as a minimum s, t-cut problem on an unbalanced bipartite graph G^b. As illustrated in Figure 5, the graph G^b is constructed so nodes representing the *edges* y_{ij} of the graph G are on one side of the bipartition and nodes representing the *nodes* x_i of G are on the other. Each node of G^b representing an edge of G is connected with infinite capacities arcs to the nodes representing its end nodes on G. For example, the node in G^b corresponding to the edge y_{23} in G is connected to the nodes in G^b that correspond to the nodes x_2 and x_3 in G. That bipartite graph has $m + n$ nodes, and $m' = O(m)$ arcs. This is actually a selection problem (a bipartite version of the closure problem) where the edges y_{ij} are the set of selections and the nodes x_i are the elements. If the solution of the selection problem is the empty set, the answer of the λ-question is "No." The complexity of a single minimum s, t-cut in such graph is therefore $O(m^2 \log m)$. This complexity however can be improved to $O\left(mn\log\left(\frac{n^2}{m}+2\right)\right)$.

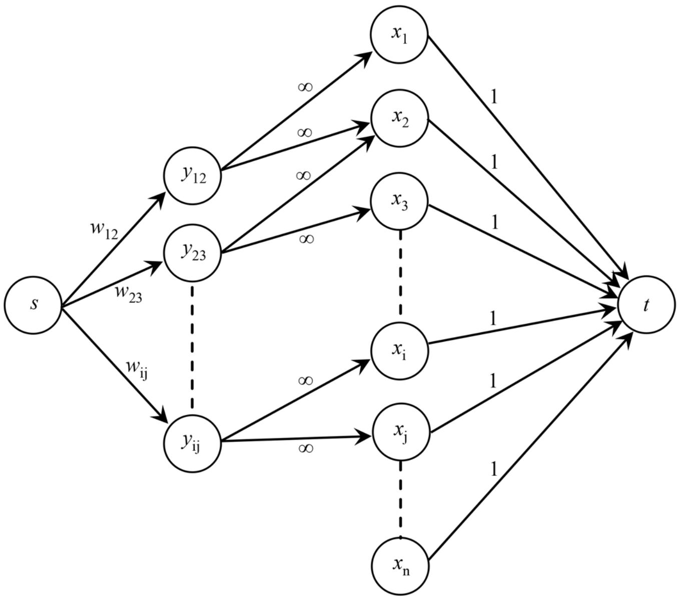

FIGURE 5 The graph for solving the 1-question for the max density subgraph problem with node weights equal to 1

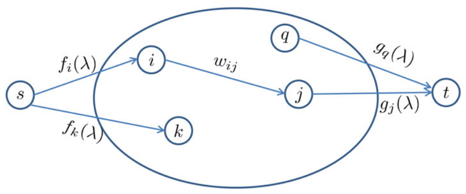

FIGURE 6 Parametric capacity graph [Color figure can be viewed at wileyonlinelibrary.com]

Once we know the answer of a λ-question, we know we can increase or decrease the value of λ to find the maximum density subgraph. Employing binary search on the value of λ, we can solve the maximum density subgraph problem in polynomial time (and the unweighted version in strongly polynomial time since the objective can only take rational values with integer numerator in $[1, m]$ and integer denominator in $[1, n]$).

A much more efficient algorithm is attained by employing the parametric cut problem, discussed next. A parametric graph has the source adjacent and sink adjacent capacities a function of a parameter, so that the source adjacent capacities are monotone non-decreasing with the parameter and the sink adjacent capacities are monotone non-increasing with the parameter. If these functions are linear then the parametric cut problem is solved for all values of the parameter λ in the same complexity as a single minimum cut procedure [7, 16]. If the functions are nonlinear then there is an added term to the complexity, $O(n \log U)$, for U the range of the parameter which is provably unavoidable [16]. Since the capacities for the maximum density problems are linear in λ the problem is solved in the complexity of a minimum cut on the graph, for example, $O\left(mn \log\left(\frac{n^2}{m}+2\right)\right)$.

5.3 | Parametric min-cut max-flow problem

Given is a "parametric flow" graph $G(\lambda) = (V \cup \{s, t\}, A)$ where each node $j \in V$ has an incoming arc from s with weight $f_j(\lambda)$, and an outgoing arc to the sink t with weight $g_j(\lambda)$. Suppose $f_j(\lambda)$ is monotone nondecreasing in λ and $g_j(\lambda)$ is monotone nonincreasing in λ (or $f_j(\lambda)$ is monotone nonincreasing in λ and $g_j(\lambda)$ is monotone nondecreasing in λ). All other arcs in the graph $(i, j) \in A$ have fixed capacities w_{ij} that are independent of the parameter. The min-cut (or max-flow) problem on such a graph depends on the parameter. A parametric graph is illustrated in Figure 6.

It was shown in References [7] and [16] that all the solutions to the parametric minimum cut of the maximum flow problem can be attained efficiently, in the complexity of a single minimum cut procedure, with the added term $O(n \log U)$ if the monotone capacity functions are nonlinear. The source sets of the minimum cuts in a parametric flow graph as a function of λ are nested:

Lemma 5.1. (Hoc08). *Let $S^*(\lambda)$ denote the source set of a minimum s, t-cut in the parametric flow graph $G(\lambda)$. Then, for $\lambda_1 < \lambda_2$:*

$$S^*(\lambda_1) \subseteq S^*(\lambda_2).$$

Although the capacity of the cut increases with increased λ, the source set of the minimum cut is strictly incremented at most n times. This implies that there are at most n distinct values of the parameter λ, $0 \leq \lambda_1 < \lambda_2 < \cdots < \lambda_\ell$, for which the source set is augmented by at least one node. We refer to such λ-values as *breakpoints*. Let the sets $S_1^* \subset S_2^* \subset \cdots \subset S_\ell^*$ denote the associate minimal source sets of the minimum cuts, where S_1^* is the source set of the minimum cut for $\lambda \in [0, \lambda_1]$.

Gallo et al. [7] showed that the breakpoints and their associated source sets are found in the complexity of a single minimum cut with the push-relabel algorithm [7]. Hochbaum [16] showed that the HPF (Hochbaum's PseudoFlow) algorithm can also be used to identify all breakpoints in the complexity $T(n, m)$ of solving for a single minimum cut. These are the only two known algorithms that solve the parametric minimum cut in the complexity of a single minimum cut. For both push-relabel and HPF the complexity is, for example, $T(n,m) = O\left(mn\frac{n^2}{m}\right)$. (Other complexity expressions depend on the implementation of the respective algorithm.)

With the breakpoints λ_i and associated source sets S_i^* for $i = 1, \ldots, \ell$ we are able to solve the ratio version of monotone IP2 or IP3 as well. Recall that these problems are equivalent to finding the smallest λ such that the linearized objective value is non-positive. Since the source sets only change at the breakpoints, it is sufficient to find the smallest λ-breakpoint where the linearized objective value, of (λ-MD) for the maximum density problem, is non-positive. The source set for this breakpoint is the optimal solution for the respective ratio problem.

Since we are considering here integer linear programs, the capacity functions of the parameter λ resulting from the linearization are always linear. We will call a monotone IP3 parametric, if the objective function depends on the parameter only in the coefficients of the x-variables.

Theorem 5.2. *If the linearized version of a ratio problem is a monotone IP2 or parametric IP3 then the ratio problem is solved optimally with a parametric minimum cut procedure in the complexity $T(n, m)$ of determining a max-flow/min-cut.*

We conclude that the maximum ratio problem on a graph $g = (V, E)$ with n nodes and m edges is solved in the complexity of minimum cut on a bipartite graph with m and n nodes in the two parts of the graph, and $O(m)$ edges. We note that the running time of both the HPF and the push-relabel algorithms is more efficient for bipartite graphs than for general graphs of the same number of nodes and edges, and in particular for those graphs that have the smaller side of the bipartition of size n. The largest distance label is then at most $O(n)$ which improves the complexity of the algorithms. We omit further details of the complexity discussion.

6 | RATIO REGIONS

The *ratio region* problem, studied by Cox et al. [5], was motivated by seeking a segment, or region, where the boundary is of low cost and the segment itself has large number of nodes. The problem is formalized as a graph problem on graph $G = (V, E)$, where the nodes of V correspond to pixels of the image, and the edges connect pairs of adjacent pixels. The edge weights represent the similarity between the pixels of the endpoints of the edge. The problem is to find a subset of nodes $S \subset V$, so that the cost of the edges on the boundary, which is the cut $C(S, \overline{S})$, is small while having the number of nodes in S relatively large. This choice of objective was to address a general difficulty that arises when by solving for a set S, the argument of the minimum cut $C(S, \overline{S})$ in the graph. Most often the minimum cut solution results in a partition that has a singleton on one side and all the other nodes of the graph on the other. The additional objective of having the size of the set as large as possible is designed to compensate for this phenomenon. The ratio region problem formulation combines these two objectives as a ratio

$$\min_{S \subset V} \frac{C(S, \overline{S})}{|S|} \tag{1}$$

This ratio region problem was addressed by Cox et al. only for graphs that are planar and thus, in the context of images, to planar grid images with 4 neighbors setup. A key to their algorithm is the observation that for planar graphs, the length of the path along the boundary of the region is the same as the capacity of a cut in the dual graph.

A generalized, weighted version of the problem has graph nodes i having weights q_i. The formulation of this weighted ratio region problem is

$$\min_{\emptyset \subset S \subseteq V} \frac{C(S, \overline{S})}{\sum_{i \in S} q_i}. \tag{2}$$

For planar graphs and node weights that are all positive, Cox et al. [5] devised a polynomial time algorithm. Hochbaum showed in Reference [19] that the weighted ratio region problem (1), for *any* general graph and for any arbitrary weights, is polynomial time solvable and with a minimum cut technique (or rather, parametric cut). This follows immediately from the formulation of the problem as a monotone IP3. To see how this problem is a monotone IP3, let $G = (V, E)$ be a graph with edge weights w_{ij} and node weights q_i, and define the following binary variables:

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \in \overline{S}, \end{cases}$$

$z_{ij} = 1$ if exactly one of i or j is in S:

$$z_{ij} = \begin{cases} 1 & \text{if } i \in S, j \in \overline{S}, \text{or } i \in \overline{S}, j \in S \\ 0 & \text{if } i,j \in S \text{ or } i,j \in \overline{S}. \end{cases}$$

With these variables, the following is an integer programming formulation of the weighted ratio region problem:

$$\text{(WRR)} \quad \min \frac{\sum_{[i,j]\in E} w_{ij} z_{ij}}{\sum_{i\in V} q_i x_i} \tag{3}$$

$$\text{subject to } x_i - x_j \le z_{ij} \quad \text{for all } [i,j] \in E \tag{4}$$

$$x_j - x_i \le z_{ji} \quad \text{for all } [i,j] \in E \tag{5}$$

$$x_j \ \text{binary} \ j \in V \tag{6}$$

$$z_{ij} \ \text{binary} \ [i,j] \in E. \tag{7}$$

For the respective linearized optimization problem, we select a "seed" $s \in V$ among the nodes and set $x_s = 1$ to ensure that the set selected is non-empty. Similarly, we select a seed $t \in V$ that will be excluded from the set selected, to ensure that the complement is not empty. The formulation of the λ-question is

$$\text{(linearized} - \text{WRR)} \quad \min \sum_{[i,j]\in E} w_{ij} z_{ij} - \sum_{i\in V} \lambda q_i x_i \tag{8}$$

$$\text{subject to } x_i - x_j \le z_{ij} \quad \text{for all } [i,j] \in E \tag{9}$$

$$x_j - x_i \le z_{ji} \quad \text{for all } [i,j] \in E \tag{10}$$

$$x_s = 1 \tag{11}$$

$$x_j \ \text{binary} \ j \in V \tag{12}$$

$$z_{ij} \ \text{binary} \ [i,j] \in E. \tag{13}$$

This linearized-WRR is a monotone IP3 problem and the parameter λ multiplies only the coefficients of the x variables. The associated graph constructed has nodes for the x-variables, that are connected to the source node s if $\lambda q_i > 0$ and to the sink node t if $\lambda q_i < 0$. This is a parametric graph and therefore, the weighted ratio region problem is solvable in the complexity $T(n, m)$.

7 | HNC: A VARIANT OF NORMALIZED CUT PROBLEM

HNC [19,21] is a graph-based clustering model that trades-off two objectives: the homogeneity of the cluster and the distinctness of the cluster from the remaining objects. HNC (Hochbaum's Normalized Cut) is a variant of the NP-hard Normalized Cut Problem (NC) which was mistakenly stated to be identical to NC in Reference [29], as explained in Section 7.1. Although we discuss HNC here on an undirected graph $G = (V, E)$, the model and algorithm for solving it apply to directed graphs as well.

The node set of the graph V is the set of objects, and the edges in E represent pairwise similarity relations between objects. The pairwise similarity on edge $[i, j] \in E$ is measured with a similarity weight $w_{ij} \geq 0$. The objects i and j are considered more similar to the value of w_{ij} increases.

The cluster of interest in HNC is a non-empty set $S \subset V$. We represent the homogeneity of the cluster S with $C(S, S)$, the total sum of similarities between objects inside the cluster. We measure the distinctness of the cluster from its complement with the capacity of the cut $C(S, \bar{S})$. The HNC model balances the two objectives of the maximization of $C(S, S)$ and the minimization of $C(S, \bar{S})$.

To enable the choice of a cluster it is required that S contains one or more seed objects as a representative of what property is sought for the cluster. Without such seeds, the optimal solution is to select $S = V$ with $C(S, S) = \sum_{[i,j]\in E} w_{ij}$ and $C(S, \bar{S}) = 0$. Let $S_P \subset V$ denote a non-empty set of *positive seeds* that must be contained in the cluster S. Similarly, $S_N \subset V$ denotes the non-empty set of *negative seeds* that are in $\bar{S}$. These seed sets guarantee that both S and $\bar{S}$ are non-empty. In a supervised context, these seed sets contain the labeled objects in the training data.

The HNC model is to optimize the ratio of distinctness and homogeneity:

$$\min_{\substack{\emptyset \subset S \subset V \\ S_P \in S, S_N \in \bar{S}}} \frac{C(S, \bar{S})}{C(S, S)}.$$

This objective was shown in Reference [19] to be equivalent to minimizing the cut capacity $C(S, \bar{S})$ divided by the sum of the weighted degrees of the nodes in S:

$$\min_{\substack{\emptyset \subset S \subset V \\ S_P \in S, S_N \in \bar{S}}} \frac{C(S, \bar{S})}{\sum_{i\in S} d_i},$$

as shown next.

Lemma 7.1. ([19]) *The set of optimal solutions to (14) and (15) are identical.*

Proof.

$$\frac{C(S, \bar{S})}{C(S, S)} = \frac{2C(S, \bar{S})}{\sum_{i\in S} d_i - C(S, S')} = \frac{2}{\frac{\sum_{i\in S} d_i}{C(S,\bar{S})} - 1}.$$

These two problems are thus equivalent, since they have equivalent objectives and the same set of feasible solutions. ■

We demonstrate here that both these problems have monotone IP3 formulations and are hence solvable in polynomial time.

7.1 | Relation to normalized cut

HNC is a close relative of the normalized cut (NC) problem. Its use in the context of image segmentation was popularized by Shi and Malik [30]. The NC problem is defined as

$$\min_{\emptyset \subset S \subset V} \frac{C(S, \bar{S})}{\sum_{i\in S} d_i} + \frac{C(S, \bar{S})}{\sum_{i\in \bar{S}} d_i} \tag{NC}$$

Compared to (15), the NC problem has an additional term in the objective. Although this change may seem minor, it has profound impact on the complexity of the problem. In fact, Shi and Malik showed that the Normalized Cut problem is NP-hard [30] whereas there is a practical, polynomial-time algorithm for the HNC problem [19]. In previous work [29], it was mistakenly stated that the HNC problem is identical to Normalized Cut and NP-hard.

To generate good feasible solutions to the Normalized Cut problem Shi and Malik [30] proposed a continuous relaxation of the problem that is solved by finding an eigenvector to a matrix related to the Laplacian of the graph. This method is called *spectral clustering*. HNC was shown in References [19] and [21] to be a form of discrete relaxation of normalized cut. Yet, HNC is solved more efficiently than the spectral clustering method. Furthermore, an extensive empirical comparison of HNC to spectral clustering on image segmentation benchmark instances, provided in Reference [11], indicated that the HNC solutions are superior, both visually and in terms of approximating the objective of the Normalized Cut problem, to the spectral clustering algorithm.

7.2 | The formulation of HNC

A formulation for the problem $\min_{S\subset V}\frac{C_1(S,\bar{S})}{C_2(S,S)}$ which generalizes HNC is provided first. This is a slight generalization of the normalized cut variant problem in permitting different similarity weights for the numerator, w_{ij}, and denominator, w_{ij}'.

We begin with an integer programming formulation of the problem. Let

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \in \bar{S}. \end{cases}$$

We define two additional sets of binary variables: $z_{ij} = 1$ if exactly one of i or j is in S; $y_{ij} = 1$ if both i or j are in S. Thus,

$$z_{ij} = \begin{cases} 1 & \text{if } i \in S, j \in \bar{S}, \text{or } i \in \bar{S}, j \in S \\ 0 & \text{if } i,j \in S \text{ or } i,j \in \bar{S}, \end{cases}$$

and

$$y_{ij} = \begin{cases} 1 & \text{if } i,j \in S \\ 0 & \text{otherwise.} \end{cases}$$

We further need to ensure that at least one edge is in the cluster S and at least one edge is in the complement-the background. Otherwise, the ratio is undefined in the first case, and the optimal solution is to choose the trivial solution $S = V$ in the second. To enforce that we pre-select an edge $[i^*, j^*]$ to be in S and another edge $[i', j']$ in the complement. This is done by setting $y_{i^*j^*} = 1$ and $y_{i'j'} = 0$. With these variable definitions, the following is a valid formulation (HNC):

$$\text{(HNC)} \quad \min \frac{\sum w_{ij} z_{ij}}{\sum w'_{ij} y_{ij}} \tag{16}$$

$$\text{subject to } x_i - x_j \le z_{ij} \quad \text{for all } [\text{i, j}] \in \text{E} \tag{17}$$

$$x_j - x_i \le z_{ji} \quad \text{for all } [\text{i, j}] \in \text{E} \tag{18}$$

$$y_{ij} \le x_i \quad \text{for all } [\text{i, j}] \in \text{E} \tag{19}$$

$$y_{ij} \le x_j \tag{20}$$

$$y_{i^*j^*} = 1 \text{ and } y_{i'j'} = 0 \tag{21}$$

$$x_j \text{ binary } j \in V \tag{22}$$

$$z_{ij} \text{ binary } [i,j] \in E \tag{23}$$

$$y_{ij} \text{ binary } i,j \in V. \tag{24}$$

To verify the validity of the formulation notice that the objective function drives the values of z_{ij} to be as small as possible, and the values of y_{ij} to be as large as possible. Constraints (17) and (18) ensure z_{ij} cannot be 0 unless both endpoints i and j are in the same set. On the other hand, constraints (19) and (20) ensure that y_{ij} cannot be equal to 1 unless both endpoints i and j are in S.

The problem formulation (HNC) is easily recognized as a monotone integer programming, IP3, with a ratio objective. We can thus linearize the objective function and solve the λ-question which asks whether $\min_{S\subset V}\frac{C_1(S,\bar{S})}{C_2(S,S)} < \lambda$, as a monotone IP3:

$$\begin{aligned} (\lambda - \text{HNC}) \quad \min & \sum_{[i,j]\in E} w_{ij} z_{ij} - \lambda \sum_{[i,j]\in E} w'_{ij} y_{ij} \\ \text{subject to } & x_i - x_j \le z_{ij} \quad \text{for all } [\text{i, j}] \in \text{E} \\ & x_j - x_i \le z_{ji} \quad \text{for all } [\text{i, j}] \in \text{E} \\ & y_{ij} \le x_i \quad \text{for all } [\text{i, j}] \in \text{E} \end{aligned}$$

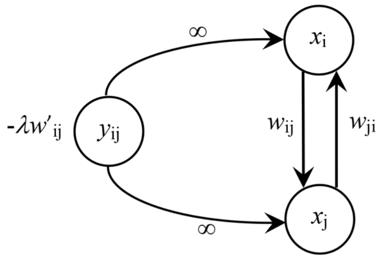

FIGURE 7 The basic gadget in the graph representation

$$\begin{aligned}
&y_{ij} \leq x_j \\
&y_{i^*j^*} = 1 \text{ and } y_{i'j'} = 0 \\
&x_j \text{ binary } j \in V \\
&z_{ij} \text{ binary } \quad [i,j] \in E \\
&y_{ij} \text{ binary } i,j \in V.
\end{aligned}$$

7.3 | Solving the λ-question with a minimum cut procedure

The formulation of (λ-HNC) is already binarized, and therefore can be solved as a minimum s-excess problem on an associated graph. That graph is a directed graph $G' = (V', A')$ with a set of nodes V' that has a node for each variable x_i and a node for each variable y_{ij}. The nodes y_{ij} carry a negative weight of $-\lambda w_{ij}$. The arc from x_i to x_j has capacity w'_{ij} and so does the arc from x_j to x_i as in our problem $w_{ij} = w_{ji}$. The two arcs from each edge-node y_{ij} to the endpoint nodes x_i and x_j have infinite capacity. Figure 7 shows the basic gadget in the graph G' for each edge $[i, j] \in E$.

We claim that any finite cut in this graph, that has $y_{i^*j^*}$ on one side of the bipartition and $y_{i'j'}$ on the other, corresponds to a feasible solution to the problem λ-HNC. Let the cut (S, T), where $T = V' \setminus S$, be of finite capacity $C(S, T)$. We set the value of the variable x_i or y_{ij} to be equal to 1 if the corresponding node is in S, and 0 otherwise. Because the cut is finite, then $y_{ij} = 1$ implies that $x_i = 1$ and $x_j = 1$.

Next, we claim that for any finite cut the sum of the weights of the y_{ij} nodes in the source set and the capacity of the cut is equal to the objective value of problem λ-HNC. Notice that if $x_i = 1$ and $x_j = 0$ then the arc from the node x_i to node x_j is in the cut and therefore the value of z_{ij} is equal to 1.

We next create a source node s and connect all y_{ij} nodes to the source with arcs of capacity $\lambda w'_{ij}$. The node $y_{i^*j^*}$ is then shrunk with a source node s and therefore also its endpoint nodes are effectively shrunk with s. The node $y_{i'j'}$ and its endpoint nodes are analogously shrunk with the sink t. We denote this graph illustrated in Figure 8 as G'_{st}.

As a special case of IP3, we have the theorem:

Theorem 7.1. *A minimum s, t-cut (S, T) in the graph G'_{st} corresponds to an optimal solution to λ-HNC by setting all the variables whose nodes belong to S to* 1 *and to* 0 *otherwise.*

The linearized version of (15) translates to another monotone IP3 for which the associated graph is smaller [21]:

$$\min \quad \sum_{[i,j]\in E} w_{ij} z_{ij} - \lambda \sum_{i \in V} d_i x_i \tag{25}$$

$$\text{s.t.} \quad x_i - x_j \leq z_{ij} \quad \forall [i,j] \in E, \quad x_j - x_i \leq z_{ji} \quad \forall [i,j] \in E,$$

$$x_{s_P} = 1 \quad \forall s_P \in S_P, \quad x_{s_N} = 0 \quad \forall s_N \in S_N,$$

$$x_i \in \{0,1\} \quad \forall i \in V, \quad z_{ij} \in \{0,1\} \quad \forall [i,j] \in E.$$

The graph associated with the formulation in (25) is smaller and contains only $O(n)$ nodes. We now show the construction of the associated graph for this IP3 [21]. Let the associated graph be a directed graph $G_A = (V \cup \{s, t\}, A)$ with the arc set A consisting of the following sets of arcs: For every edge $[i, j] \in E$, add the two arcs (i, j) and (j, i) to A. Both arcs have a capacity of $w_{ij} \geq 0$. We also add arcs (s, i) with capacity λd_i for every node $i \in V$. Finally, for every positive seed $i \in S_P$ we add an infinite capacity arc (s, i), and for every negative seed $j \in S_N$ we add an infinite capacity arc (j, t). Note that corresponding to the previous formulation that sets an edge in S and an edge in $\bar{S}$, the seed sets are $S_P = \{i^*, j^*\}$, and $S_N = \{i', j'\}$.

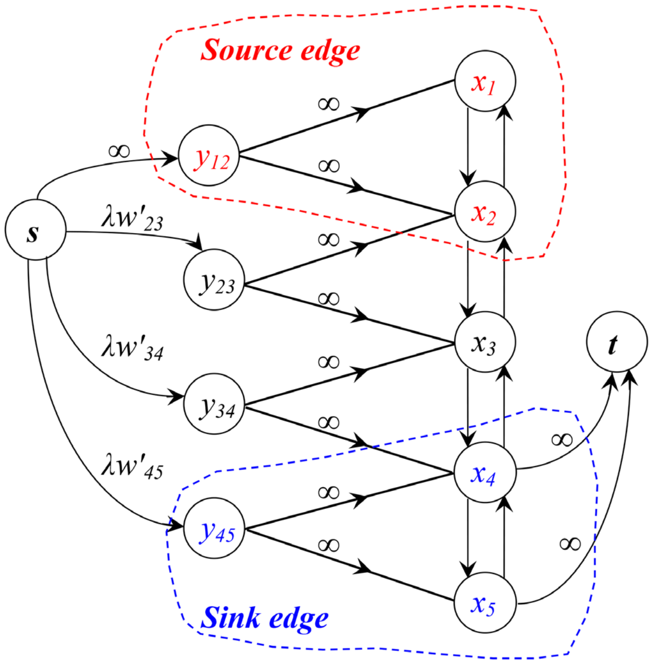

FIGURE 8 The graph G'_{st} with edge [1, 2] as source seed and edge [4,5] as sink seed [Color figure can be viewed at wileyonlinelibrary.com]

As in the general case of IP3, here also the source set of a minimum s, t-cut in the associated graph G_A is an optimal solution to the linearized HNC problem. Although it follows from the general case, we include the proof for this specific case in the following theorem.

Theorem 7.2. ([21]). *The source set S^* of a minimum s, t-cut $(S^* \cup \{s\}, \overline{S}^* \cup \{t\})$ in G_A is an optimal solution to the linearized HNC problem for a fixed value of λ.*

Proof. Let $(S \cup \{s\}, \overline{S} \cup \{t\})$ be any finite s, t-cut in G_A. The infinite capacity arcs (s, i) for $i \in S_P$ guarantee that $S_P \subseteq S$. Similarly, $S_N \subset \overline{S}$. Thus, the set S is a feasible solution. G_A always contains a finite cut when the seed sets are disjoint, $S_P \cap S_N = \emptyset$.

The capacity of cut $(S \cup \{s\}, \overline{S} \cup \{t\})$ is equal to

$$C(S \cup \{s\}, \overline{S} \cup \{t\}) = C(S,\overline{S}) + \lambda \sum_{i \in \overline{S}} d_i = C(S,\overline{S}) + \lambda \left(D - \sum_{i \in S} d_i \right) = \lambda D + C(S,\overline{S}) - \lambda \sum_{i \in S} d_i,$$

where $D = \sum_{i \in V} d_i$ denotes the sum of weighted degrees. Thus, minimizing the capacity of the cut is equivalent to minimizing $C(S,\overline{S}) - \lambda \sum_{i \in S} d_i$. ■

The associated graph has $n+2$ nodes and up to $2m+2n$ arcs, since every edge $[i,j] \in E$ has two associated arcs and every node has up to two associated arcs. One arc connecting it to the source node s with capacity λd_i, and potentially another arc that connects a seed node to either the source node s or to the sink node t.

Corollary 7.3. *The linearized HNC problem for a fixed value of λ is solvable in $O(T(n+2, 2m+2n))$ time.*

Since the associated graph is parametric, we can solve the linearized version simultaneously for all values of λ in the complexity of a single minimum cut, $T(n+2, 2m+2n)$, with a parametric minimum cut problem [7,16]. This directly implies that the ratio version of HNC in (15) is solved in the same complexity, since these problems are equivalent to finding the smallest λ such that the objective of the linearized problem is non-positive.

7.4 | Applications of HNC

HNC has been used in image segmentation. In this context, it was shown in Reference [11] how to generate good results both subjectively and from the point of view of providing a better approximation to the NC problem than the spectral technique of Shi and Malik [30].

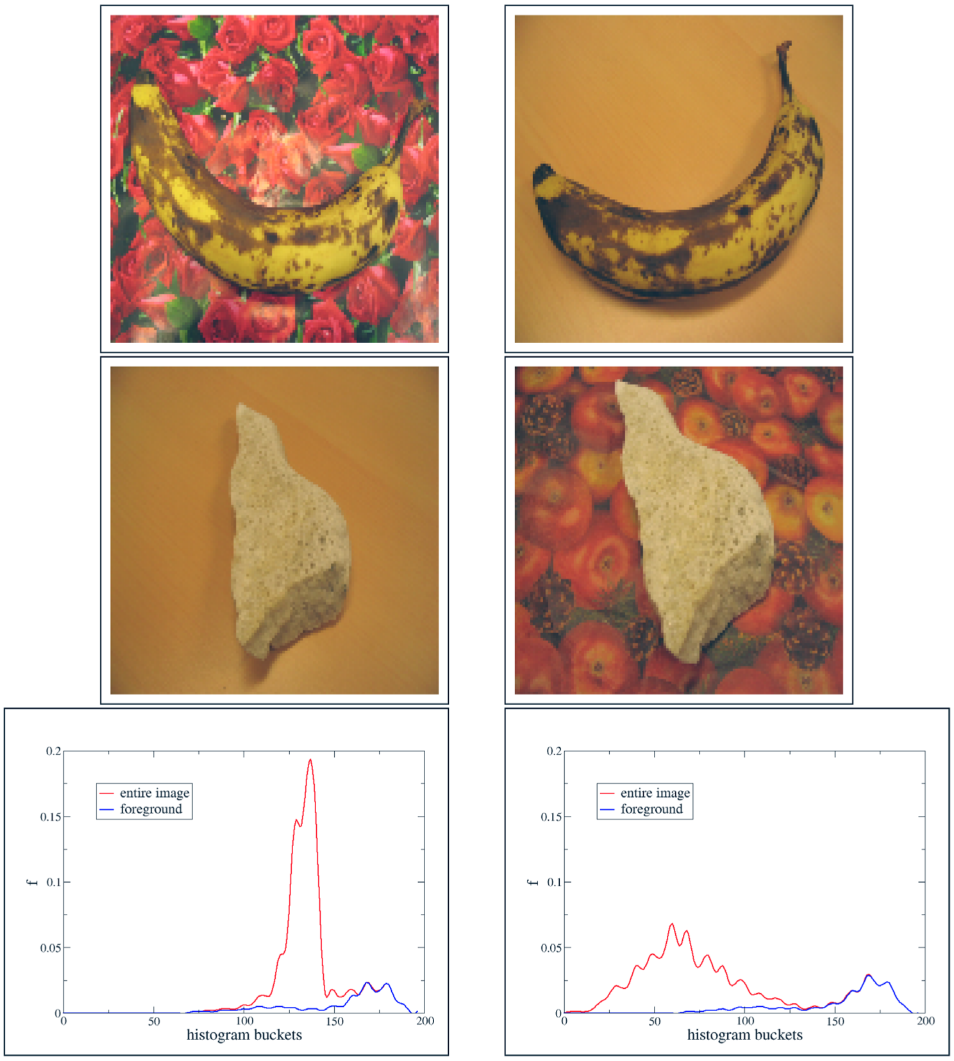

FIGURE 9 A similar object in two images in rows 1–2. The histogram of the foreground (of row 2 images) is shown in row 3 [Color figure can be viewed at wileyonlinelibrary.com]

HNC was successfully used in data mining contexts. These include the denoising of spectra of nuclear detectors [31]; ranking drugs according to their effectiveness [10]; and it has been a leading algorithm in the Neurofinder benchmark for cell identification in calcium imaging movies [2,28]. Moreover, it was tested as a generic machine learning techniques and compared to other leading machine learning methods [3]. It was shown that its use of pairwise similarities renders it a very effective machine learning method that tends to require less training than other machine learning techniques.

8 | THE CO-SEGMENTATION PROBLEM

In the co-segmentation problem, the objective is to segment a similar object from a pair of images. The background in the two images may be arbitrary; therefore, simultaneous segmentation of both images must be performed with a requirement that the appearance of the two sets of foreground pixels in the respective images are consistent. The IP3 presented model here first appeared in Reference [23].

The input to the co-segmentation problem consists of two images for segmentation $I^{(1)}$ and $I^{(2)}$, of the same size, of n pixels each. We denote the jth pixel in the qth image by $I_j^{(q)}$, for $j = 1, \ldots, n$ and $q = 1, 2$. The input consists of, in addition to the images, a representation of the images in terms of their histograms; more specifically, we are given a classification of each pixel in each image into one of K "buckets" of occurrence of the intensity values in an image. For example, see Figure 9 for two pairs of images that have a feature in common, and the histogram of the foreground of the second pair of images.

Let the histogram buckets (each bucket corresponds to an intensity range) be given as $H_1, H_2, \ldots, H_K$. For each image $I^{(q)}$, $q = 1, 2$, this may be specified in terms of a matrix $\mathbf{B}^{(q)}$ of size $n \times K$ such that for pixel j and bucket H_k:

$$B_{j,k}^{(q)} = \begin{cases} 1 & \text{if } \mathrm{I}_\mathrm{j}^{(\mathrm{q})} \in \mathrm{H}_\mathrm{k}; \\ 0 & \text{otherwise.} \end{cases}$$

That is, the entry $B_{j,k}^{(q)}$ is 1 if the intensity of pixel $I_j^{(q)}$ falls in the intensity range of bucket H_k, where q refers to either the first or the second image.

8.1 | The model

A segmentation of each image will partition the set of pixels into *foreground* vs *background* pixels. In co-segmentation the aim is to ensure that the foreground in the two images are similar. Toward this goal, the objective is to get (a) the number of pixels that are in the foreground, and (b) the number of pixels in H_k, to be approximately similar in both images. One strategy is to define *similarity* between all pairs of pixels $I_i^{(1)}$ and $I_j^{(2)}$. We can say that the pair i, j is similar if both belong to H_k and designate a similarity weight s_{ij} to be equal to 1 if that happens. Formally,

$$s_{ij} = \begin{cases} 1 & \text{if } \exists k \text{ such that } B_{i,k}^{(1)} = B_{j,k}^{(2)} = 1 \\ 0 & \text{other wise.} \end{cases}$$

Let $x_j^{(q)}$ be a binary variable indicating whether pixel $I_j^{(q)}$ is classified in the foreground:

$$x_j^{(q)} = \begin{cases} 1 & \text{if } I_j^{(q)} \text{ is classified as foreground} \\ 0 & \text{if } I_j^{(q)} \text{ is classified as background.} \end{cases}$$

The number of pixels in the foreground of $I^{(1)}$ that belong to H_k is denoted by $a_k = \sum_{j=1}^n B_{j,k}^{(1)} x_j^{(1)}$ and the number in $I^{(2)}$ that belong to H_k is denoted by $b_k = \sum_{j=1}^n B_{j,k}^{(2)} x_j^{(2)}$. Let the total number of foreground pixels in $I^{(1)}$ and $I^{(2)}$ be F_1 and F_2 respectively. We model a measure of similarity of the two foreground features as the optimal solution to

$$\begin{aligned} \max \ & \sum_{k=1}^K a_k b_k \\ \text{subject to } & \sum_{k=1}^K a_k = |F_1| \\ & \sum_{k=1}^K b_k = |F_2|. \end{aligned} \tag{27}$$

For two given features $|F_1|$ and $|F_2|$ are fixed, and the optimal solution value is the number of pairs of identical histogram classifications between the two features. We introduce a similarity variable, specified by the binary value s_{pq}, which is 1 if p and q belong to the same bucket, and 0 otherwise.

8.2 | Problem statement

Maximizing similarity of histograms as in (27) by itself is not sufficient to obtain meaningful segmentations. This is because co-segmentation must take the spatial homogeneity of the images into account also. This may be achieved by introducing the adjacency relationship between neighboring pixels as an additional bias into the maximization in (27). Another option, adopted here, is to segment both images while using the similarity in (27) as a bias term.

8.3 | MRF segmentation

We formulate the task of segmenting both images as a binary labeling of Markov Random Field (MRF) on the graphs corresponding to the input images [4,14,20]. A generic formulation of binary MRF is given next. Note that the binary MRF problem is an instance of the s-excess problem discussed in Section 4. For non-binary labels and convex objective function there are very efficient algorithms solving the MRF problem in polynomial time [14,20], also discussed in Section 4.2. Modeled as the binary MRF, the co-segmentation problem is to find an assignment of values to every pixel in an image, as either foreground or background label. This is represented by a binary variable x_j assigned to each pixel j and is equal to 1 if the pixel is assigned to the foreground. The assignment is such that the total deviation and separation penalties are minimized. The *deviation*, d_j, is charged for a pixel that is set in the foreground, although there is a priori information indicating it should be in the background.

The *separation or smoothness* penalty w_{pq} measures the cost of assigning different labels to two neighboring pixels, $p \sim q$. A commonly used setting for the separation weight is $w_{pq} = \exp(-\beta \|\ell(p) - \ell(q)\|^2)$, where $\ell(p)$ and $\ell(q)$ are the respective given labels of pixels p and q and β is a constant. The MRF formulation for one image is then

$$\begin{aligned} \text{(MRF)} \ \min \ & \sum d_j x_j + \sum_{i\sim j} w_{ij} y_{ij} \\ \text{subject to } & x_i - x_j \le y_{ij} \\ & x_j - x_i \le y_{ji}. \\ & x_i, y_{ij} \ \text{binary for } i,j = 1, \dots, n. \end{aligned} \tag{28}$$

8.4 | Co-segmentation monotone IP3 model

Our model attempts to simultaneously minimize the separation and deviation terms in the MRF model for *each* image as well as maximize the similarity (rather than minimize the difference as in Reference [27]) between the foreground features in the two images as specified in (27). As these are two conflicting and incompatible goals, we use a linear combination of the two objectives (treating the second term as a bias). Let λ be a coefficient expressing the relative weights of the two objectives [27]: when the value of λ is high, then similarity is the most important requirement, and when it is low, the MRF penalties are dominant. Let z_{ij} be a variable equal to 1 if $I_i^{(1)} \in F_1$ and $I_j^{(2)} \in F_2$. Our objective function minimizes a combination of the penalties incurred by the MRF optimization in each image, and *subtracts* the similarity measure of the number of pairs of the same histogram buckets in the resulting two foreground features. Since we have a minimization in (28), a high similarity in the foreground features serves as a reward, exactly as desired.

In this formulation, we seek an assignment of each pixel to the foreground or the background. So we may simplify the notation: $d_j^{(1)}, d_j^{(2)}$ are the deviation penalties charged for placing pixel j in the foreground of image 1 and 2 respectively. These penalties can be positive or negative. The minimization objective includes terms representing the MRF optimization in both images:

$$\sum d_j^{(1)} x_j^{(1)} + \sum_{i\sim j} w_{ij} y_{ij}^{(1)} + \sum d_j^{(2)} x_j^{(2)} + \sum_{i\sim j} w_{ij} y_{ij}^{(2)}.$$

Simultaneously, we also wish to maximize the benefit of high similarities between corresponding histogram buckets in both images represented as

$$\sum_{i\in I^{(1)}, j\in I^{(2)}} s_{ij} z_{ij}.$$

Since s_{ij} is equal to 1 only for "matching" histogram buckets, this latter term can also be written as

$$\sum_{k=1}^{K} \sum_{i\in I^{(1)}\cap H_k, j\in I^{(2)}\cap H_k} z_{ij}. \tag{29}$$

Notice that (29) is equivalent to the requirement specified in (27). Our formulation of the co-segmentation problem is then a linear combination of the MRF minimization and similarity maximization objectives which yields a monotone IP3 formulation as follows:

$$\begin{aligned} \min \ & \sum d_j^{(1)} x_j^{(1)} + \sum_{i\sim j} w_{ij} y_{ij}^{(1)} + \sum d_j^{(2)} x_j^{(2)} + \sum_{i\sim j} w_{ij} y_{ij}^{(2)} - \lambda \sum_{k=1}^{K} \sum_{i\in I^{(1)}\cap H_k, j\in I^{(2)}\cap H_k} z_{ij} \\ \text{subject to } & z_{ij} \le x_i^{(1)} \ \text{ for } i \in I^{(1)} \\ & z_{ij} \le x_j^{(2)} \ \text{ for } j \in I^{(2)} \\ \text{(Co-seg)} & x_i^{(q)} - x_j^{(q)} \le y_{ij}^{(q)} \ \text{ for } q = 1,2 \\ & x_j^{(q)} - x_i^{(q)} \le y_{ji}^{(q)} \ \text{ for } q = 1,2 \\ & x_i^{(q)}, y_{ij}^{(q)}, z_{ij} \ \text{binary for } q = 1,2, \\ & i,j = 1, \dots, n. \end{aligned}$$

Correctness. To verify correctness, observe that the first set of constraints on z_{ij} ensures that the binary variable z_{ij} can be equal to 1 only if both pixels i and j, in the first and second images respectively, are selected in the foreground. The second set of constraints is to guarantee that if adjacent pixels i and j in one of the images are assigned such that one is in the foreground and the other is in the background, then the separation penalty for that neighboring pair is charged. Notice that we make use of the objective that drives z_{ij} to be as large as possible (ie, 1) and y_{ij} to be as small as possible (ie, 0). Since the model of the

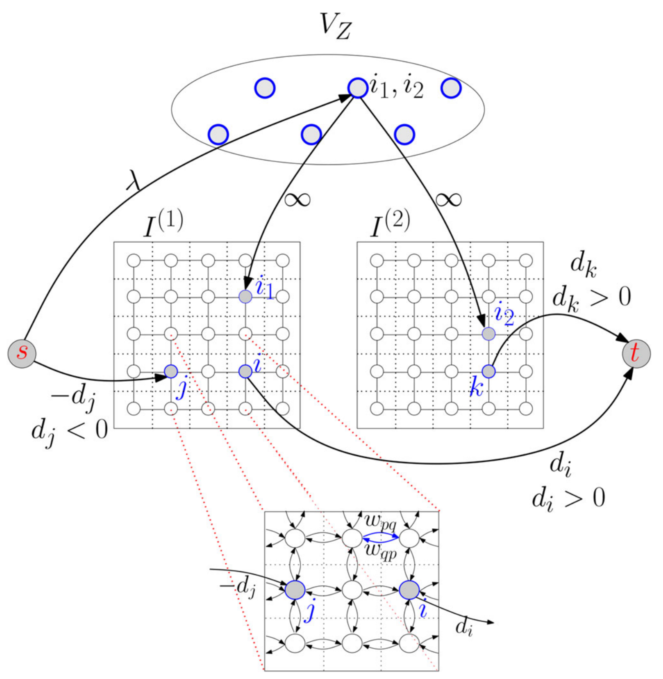

FIGURE 10 The construction of the graph G with two dummy nodes, the set of pixels in the two images $I^{(1)}$ and $I^{(2)}$, and the set of similarity nodes V_z. Some nodes and arcs are annotated to show the graph structure [Color figure can be viewed at wileyonlinelibrary.com]

(Co-seg) problem is a monotone IP3 formulation, we can make use of a construction of an s, t graph G, where the minimum s, t-cut partition will provide an optimal solution to the (Co-seg) problem.

8.5 | The graph construction

We now show the construction of the s, t graph G which will be used to solve (Co-seg): For each of the two images, the graph contains a grid of nodes, called here pixel-nodes, one corresponding to each pixel. To achieve *only* the MRF segmentation for both images specified as in Reference [28], we can use a graph construction similar to the one described in Reference [14], with either the 4-neighbor or the 8-neighbor or any other form of neighborhood topology used to describe the adjacency relationship between pixel-nodes. To make it suitable for co-segmentation, additional features and modifications are added.

We denote the pixel-nodes in the graph by V_x, as each corresponds to a variable x_i. The graph G contains the "dummy" nodes s and t. Each pixel-node j has a weight d_j associated with it, as shown in (28). If $d_j > 0$, then there is an arc (j, t) of capacity d_j. If $d_j < 0$, then there is an arc (s, j) of capacity $-d_j$. We partition V_x into $V_{x^+} \cup V_{x^-} \cup V_0$, where for each node j in V_{x^+}, $d_j > 0$, and for each node j in V_{x^-}, $d_j < 0$. For each pair of adjacent nodes i and j there is a capacity w_{ij} on both directed arcs (i, j) and (j, i).

We now outline the key modifications. In addition to the nodes for the pixels in the two images, there is a *similarity node*, or z-node, for each pair (i_k^1, i_k^2) so that $i_k^1 \in I^{(1)} \cap H_k$ and $i_k^2 \in I^{(2)} \cap H_k$. This node corresponds to the variable $z_{i_k^1, i_k^2}$ in the (Co-seg) model. We denote the set of similarity nodes by V_z, and link each such node to both i_k^1 and i_k^2 with arcs of infinite capacity. We then link this node to the source with an arc $(s, (i_k^1, i_k^2))$ of capacity λ (weight of the bias).

The constructed graph is $G = (V \cup \{s, t\}, A)$ with $V = V_x \cup V_z$ and A the set of arcs. The set of arcs A is the union of: the set of adjacency arcs in I_1, A_1; the set of adjacency arcs in I_2, A_2; the set of arcs (j, t) directed to the sink from all nodes $j \in V_{x^+}$; the set of arcs (s, j) from the source to all nodes $j \in V_{x^-}$; one arc (s, z) for each node $z \in V_z$ and two arcs from each z to the respective pixel nodes. An illustration of the graph is shown in Figure 10.

Solving the minimum cut problem on this graph provides an optimal solution to the (Co-seg) model. An extensive experimental study described in Reference [23] demonstrated high-quality co-segmentation results with lower co-segmentation error as compared to other leading algorithms. Furthermore, the running times of the minimum cut algorithm for (Co-seg) are also considerably faster than the running times of other algorithms for the problem.

9 | THE MULTI-SENSOR NUCLEAR THREAT DETECTION PROBLEM

We consider here a scenario in an urban environment facing potential nuclear threats such as "dirty bombs". One way of reducing false-positive and false-negative errors in an alerting system is by considering inputs from multiple sources. With recent technology, it has become operational and cost-effective for multiple detectors to be mounted on moving vehicles. The scenario considered is that of multiple taxi cabs each carrying a detector in an urban area such as Manhatten. The real-time detectors' positions are known in real time as these are continuously reported from GPS data. The level of detected risk is then reported from each detector at each position. The problem is to delineate the presence of a potentially dangerous source and its approximate location by identifying a small area that has higher than threshold concentration of reported risk. This problem of using spatially varying detector networks to identify and locate risks is modeled and formulated here as a monotone IP3.

The information transmitted by the detectors is to be used as input in a process which is to generate alerts triggered by identifying a "small region" with a "high concentration" of risk. This problem was initially studied with a limit on the size of the region and a bound on the concentration, which led to an NP-hard problem. The alternative modeling presented in References [9] and [18] treats those limits as soft constraints and results in a monotone integer program formulation.

The goal is to identify, at every period of time, a region within the area of interest, which is limited in size and with high concentration of alerts, or report that none has been identified. The purpose is to delineate the presence of a potentially dangerous source and its approximate location. The detectors' transmitted information, along with the geographical positioning of the collection of detectors, is to be consolidated into reliable reporting on whether nuclear threat exists, and if so, its approximate position. In case this information is deemed to indicate a high enough likelihood of real danger, the detection operations shifts to a high alert state where higher sensitivity detectors and personnel with expertise will be deployed into the region of interest with the task of pinpointing, locating and disabling the source of the threat.

The alert concentration problem is presented as an optimization problem, combining two goals: One goal is to identify a small region; another goal is to have large number of alerts, or high concentration of alerts in the region. These two goals are potentially conflicting - focusing on a large number of alerts within an area is likely to result in the entire region; on the other hand, focusing on concentration alone would result in a single block of the area containing the highest level of reported alert, thus disregarding information provided by other detectors in the adjacent area.

We formulate here the detection problem as an IP3 model which is parametrized by a weight, β, that balances the relative contribution of the two goals. The problem is then solvable in polynomial time as a minimum s, t-cut problem, and is solved for all values of the parameter using a parametric cut procedure. As shown in Reference [9], the efficiency of the procedure enables its use in real-time scenarios.

9.1 | Notation

Without loss of generality, we consider the region where the detectors are deployed to be a rectangular area subdivided in grid squares. These will be small enough to contain approximately one vehicle and up to two detectors (it is noted that this assumption plays no role in the formulation). Let V be the collection of positions (blocks or pixels of the grid) in the area considered.

In our formulation, we capture the size of the region by a measure of its boundary length. A directed graph G with the set of nodes V corresponding to the set of blocks represents the region. For each adjacent pair of blocks, if one is within the region and the other outside, the added length to the boundary is $u_{ij} = 1$. The adjacency $[i, j]$ is represented by a pair of arcs in opposite directions each of capacity 1. These arcs are referred to as the "adjacency" arcs of G, and denoted by A_a. Let $S \subset V$ be the blocks of a selected subregion. We measure the *size* of the area delineated by S using the length of its boundary, counted as the number of block sides that separate S from $\overline{S}$. The length of the boundary of a subset of grid points S is then $\sum_{i\in S, j\in \overline{S}, (i,j)\in A_a} u_{ij}$. Since the set A_a contains arcs of capacity 1, this length is equal to $C(S, \overline{S}) = |\{[i,j] | i \in S, j \in \overline{S}\}|$. Note that there is no requirement that the set S is contiguous. Indeed it can be formed of several connected components. It is proved, however (see Theorem 9.1), that there is always an optimal solution to the ratio problem formed of a single connected component.

9.2 | The input

The information captured by a detector is a spectrum of gamma-ray emissions recording the frequency at each energy level. An analysis process is utilized to convert the spectrum information to indicate whether the detected information implies the presence of a nuclear threat or not. Alternatively, a likelihood level of such presence is reported when the analysis of the detector's transmitted information delivers, for each alert reported by a detector at i, an alert confidence level of p_i. For each no-alert reported from position j, there is a no-alert confidence level of q_j.

At a given time instance, let D be the set of positions of taxis with $D^* \subseteq D$ representing the set of positions reporting alert. The set $\overline{D}^* = D \backslash D^*$ is the set of taxi positions reporting no alert. An extension of the model allows for varying levels of alert and varying levels of no-alert.

9.3 | Formulating the objective function

Since our objective involves multiple goals, we address these by setting a trade-off between the importance of the short boundary vs the high concentration of alerts. One way of trading these off is by minimizing a ratio function—of the length of the boundary divided by the concentration of alerts in the region. Another, is to use a weighted combination of the goals.

To formalize the goal of "small area" we define an area to be of small size if it is enclosed by a "short" boundary. The boundary of an area is then the number of edges that separate in-region from out-region, or the rectilinear length of the boundary. In the graph $G = (V, A_a)$ defined above the length of the boundary of a set $S \subset V$ is $C(S, \overline{S})$. Prior to proceeding, we note that this definition of length needs certain tweaking. Let the set of boundary blocks of the entire area considered be denoted by B. With the definition of the set of arcs A_a, the selection of any subset of B reduces the defined size of the region. For example, if the selected region is all of V then the length of the circumference $C(V, \emptyset)$ is equal to 0. To prevent that, we add a set of arcs A_B from the boundary nodes to an imaginary point in space. This will be quantified in a manner explained later. We let the corner blocks contribute 2 to the length of the boundary, if included in the set. The length of the boundary is thus $C(S, \overline{S}) + |B \cap S|$ where we count the corner block twice in B (instead of introducing additional notation.)

Next, we formalized the goal of identifying high concentration of alerts. One indication of the level of alert in an area is the number of alerts at higher than threshold level within the area. Let us now, for the sake of simplicity, assume that the inputs are in the form of alert or no-alert. Let $D \subseteq V$ be the set of positions occupied by vehicles. Let $D^* \subset D$ be the set of positions reporting alerts. Part of our objective is then to identify a subregion of positions containing S so that $|D^* \cap S|$ is maximized.

Maximizing the number of alerts within the selected region is an objective that has some pitfalls. For instance, if the region considered, S, contains, in addition to the alerts, also a relatively high number of no-alerts $\overline{D}^* \cap S$, for $\overline{D}^* = D \backslash D^*$, then this should diminish the significance of the alerts in the region. To address this, we add yet another minimization objective, min $|\overline{D}^* \cap S|$. This objective is then combined with the length of the boundary objective, as min $C(S, \overline{S}) + |B \cap S| + \alpha|\overline{D}^* \cap S|$. Although, in terms of the model, we do not restrict the value of α, it is reasonable that α should be a small value compared to the contribution of alert positions, as discussed below. If we choose to disregard the number of no-alerts in the region, then this is captured by setting $\alpha = 0$.

One way of combining a maximization objective $g(x)$ with a minimization objective $f(x)$ is to minimize the ratio of the two functions $\frac{f(x)}{g(x)}$. For the alert concentration problem, the ratio objective function is

$$\min_{S \subset V} \frac{C(S, \overline{S}) + |B \cap S| + \alpha|\overline{D}^* \cap S|}{|D^* \cap S|}$$

One advantage of using this ratio formulation is that it is guaranteed that an optimal solution will be a single connected component. This was proved in Hochbaum [17] for a general family of ratio problems. Formally, we define the concept of *additive functions*. For a set of connected components in the graph $A_1, \ldots, A_k, A_i \subset V$, that are pairwise disjoint, the function $f()$ is said to be additive if $f(\bigcup_{j=1}^{k} A_j) = \sum_{j=1}^{k} f(A_j)$. For additive ratio functions there is an optimal solution consisting of a single connected component:

Theorem 9.1. [17] *For additive functions f and g, there exists an optimal solution to the problem* min $\frac{f(x)}{g(x)}$ *consisting of a single connected component and its complement.*

It is easy to verify that our functions here are additive, and hence the existence of a single connected component optimal solution follows.

An alternative to the ratio presentation is to minimize a function which is a linear combination of the two objectives, which is also in the form of the "β-question" discussed in Section 5.1. Using β as a weight for the relative importance of the weights, the objective function of the concentrated alert (CA) problem is

$$\text{(CA)} \quad \min_{S \subseteq V} \quad C(S, \overline{S}) + |B \cap S| - \beta|D^* \cap S| + \alpha|\overline{D}^* \cap S|.$$

The problem (CA) has two parameters, β and α. We show how to solve the problem for all values of β provided that α is fixed, and vice versa. Each of these algorithms runs in strongly polynomial time for all values of the parameter.

9.4 | Constructing the graph

Let the region be represented by a collection of nodes V of a directed graph where each block is represented by a node. The set of nodes is appended by a dummy *sources* and a dummy *sink* node t.

An edge represents two adjacent blocks, where the adjacency can be selected to be any form of adjacency. Here we use either the *4-neighbors* adjacency or *8-neighbors* adjacency. The weight of each edge is set to 1, and each edge $[i, j]$ is replaced by two directed arcs, (i, j) and (j, i), both of capacity 1. These arcs form the set A_a.

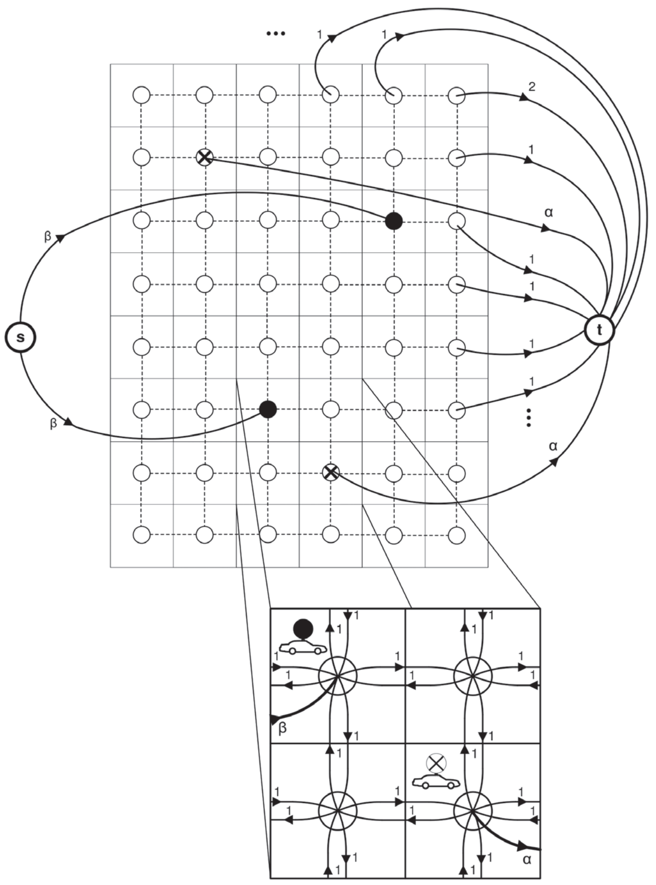

FIGURE 11 The graph G_{st} for the nuclear threats detection problem

We connect the set of arcs A_B to the sink t with capacities of 1 except for "corner" blocks that contribute 2 to the length of the boundary. Each position that contains an alert taxi in D^* is set to be adjacent to s with a directed arc of capacity β. Each position that contains a no-alert taxi, in $\overline{D}^*$, is set to be adjacent to t with a directed arc of capacity α. We denote these sets of arcs by A_β and A_α, respectively.

We therefore construct a directed s, t graph $G_{st} = (V_{st}, A)$, where $V_{st} = V \cup \{s, t\}$ and $A = A_a \cup A_B \cup A_\beta \cup A_\alpha$. The construction of the graph is illustrated in Figure 11 where an alert position is indicated by a solid circle, and a no-alert position by a crossed circle.

We now have a graph $G = (V \cup \{s, t\}, A)$ with arc capacities u_{ij} for arc $(i, j) \in A$, on which we define the minimum s, t-cut problem and show that solving it provides the optimal solution to our CA problem.

9.5 | The main theorems and algorithms

Let a cut be a partition (S, T) of V of capacity $C(S, T) = \sum_{(i,j) \in A,\, i \in S,\, j \in T} u_{ij}$.

Theorem 9.2. *The source set of a minimum cut in the graph G_{st} is the optimal region for problem* CA.

Proof. Let $(S \cup \{s\}, T \cup \{t\})$ be a partition of $V \cup \{s, t\}$ and thus an s, t-cut in G. We compute this cut's capacity:

$$\begin{aligned} C(S \cup \{s\}, T \cup \{t\}) &= |B \cap S| + |\overline{D}^* \cap S|\alpha + |D^* \cap T|\beta + \sum_{(i,j) \in A, i \in S j \in T} 1 \\ &= |B \cap S| + |\overline{D}^* \cap S|\alpha + (|D^*| - |D^* \cap S|)\beta + C(S, T) \\ &= |D^*| + |B \cap S| + C(S, T) + |\overline{D}^* \cap S|\alpha - |D^* \cap S|\beta. \end{aligned}$$

Now the first term is a constant $|D^*|\beta$. Thus minimizing $C(S \cup \{s\}, T \cup \{t\})$ is equivalent to minimizing $|B \cap S| + C(S, T) + |\overline{D}^* \cap S|\alpha - |D^* \cap S|\beta$, which is the objective of the CA problem. ■

We conclude that solving the concentrated alert problem reduces to finding a minimum s, t-cut in the graph G_{st}. The region we are seeking will then correspond to the source set S of the minimum cut $(S \cup \{s\}, T \cup \{t\})$.

The weighted version of the alert concentration problem: The information provided by the detector may be too ambiguous to translate to a simple binary statement of the form of alert or no-alert. Instead, one defines a threshold level, and within the above-threshold alert category, one creates a function that maps the alert profile transmitted from location i to a weight value p_i that is monotone increasing with the increased confidence in the significance of the alert information. Similarly, the below-threshold category of no-alert maps into a weight value q_i that is monotone increasing with the increased confidence in the significance of the no-alert information. The modified *weighted concentrated alert* problem is then to find a subregion S that optimizes the function

$$\text{(WCA)} \min_{S\subseteq V} \ C(S,\bar{S}) + |B\cap S| + \alpha \sum_{i\in \overline{D}^*\cap S} q_i - \beta \sum_{i\in D^*\cap S} p_i.$$

To solve this weighted problem, we modify the assignments of capacities to the arcs in the graph G_{st} as follows:

For each position i in D^* we let the capacity of the arc from the source to i be $u_{si} = \beta p_i$, and for each position, i in $\overline{D}^*$ we let the capacity of the arc from i to the sink be $u_{it} = \alpha q_i$. We call the graph with these modified arc capacities G^W_{st}. A weighted version of Theorem 9.2 then holds:

Theorem 9.3. *The source set of the minimum cut in the graph G^W_{st} is the optimal region for problem WCA.*

Proof. The proof is a simple generalization of Theorem 9.2. We include it here for the sake of completeness.

Let $(S\cup\{s\}, T\cup\{t\})$ be, as before, an s, t-cut in G, of capacity

$$\begin{aligned} C(S\cup\{s\}, T\cup\{t\}) &= |B\cap S| + \alpha \sum_{i\in\overline{D}^*\cap S} q_i + \beta\alpha \sum_{j\in D^*\cap T} p_j + \sum_{(i,j)\in A, i\in S j\in T} 1 \\ &= |B\cap S| + \alpha \sum_{j\in\overline{D}^*\cap S} q_i + \beta\left(\sum_{i\in V} p_i - \sum_{j\in D^*\cap S} p_j\right) + C(S,T) \\ &= \beta\sum_{i\in V} p_i + |B\cap S| + C(S,T) + \alpha \sum_{i\in\overline{D}^*\cap S} q_i - \beta \sum_{j\in\overline{D}^*\cap S} p_j. \end{aligned}$$

Since $\beta\sum_{i\in V} p_i$ is a constant, the source set of the minimum cut is also minimizing $|B\cap S| + C(S,T) + \alpha\sum_{i\in\overline{D}^*\cap S} q_i - \beta\sum_{j\in\overline{D}^*\cap S} p_j$. ∎

The solution for all values of the parameters is then attained with a parametric cut procedure. As the value of β is changing the solution changes as well. We note that the source adjacent arc capacities are monotone increasing in β and the sink adjacent arc capacities are unaffected. Therefore this is a scenario of the parametric maximum flow minimum cut problem. As discussed in Section 5.3 the complexity of solving such a problem is the same as the complexity of solving for a single minimum s, t-cut. The source code of the solver we use is available at [22], and a parametric version is available as well.

Since we can find the solution for all values of β, this leads to finding the optimal solution to the respective ratio problem which corresponds to the largest value of β where the solution value is still ≤ 0.

It is possible to examine the sensitivity of the parameter α independently from that of β. In other words, we keep β fixed and then study the effect on the solution of solving for all possible values of α.

9.6 | Numerical examples

Several instances of the problem were devised on a grid. In the figures below a full circle represents a detector position reporting an alert and a crossed circle represents a detector position reporting a no-alert. The length of the boundary was taken to be its rectilinear length. That is, the 4-neighbor adjacency was selected. The problem instances were run for $\beta = 3.99$ and $\alpha = 0.5\beta$. The reason why the value of β is just under 4 is to prevent the generation of regions consisting of singletons of alert positions.

In Figure 12 the set V is a 7×10 grid. The set of three alert positions forms the optimal solution. The optimal region is indicated by darker shade. Notice that in this example there are, on row 10, two alert positions separated by an empty position. Although these might indicate an elevated alert status for that area, the vacant grid position rules out selecting this set. The presence of vacant positions therefore should not necessarily be interpreted as diminishing the alert level. These are only the result of a random distribution of the positions.

To allow for regions to be generated even if they contain alert positions separated by a small number of empty grid points, we assign a small value of β, denoted by γ, to each vacant grid point. That means that every vacant position is interpreted as a "minor" alert position and the objective function has an extra term $-\gamma|V\backslash D|$. The modification in the graph of Figure 10 is to add arcs from source s to every unoccupied square in the grid with capacity γ each. Theorem 9.2 is easily extended for this case. In the next set of examples we set $\gamma = 0.021$.

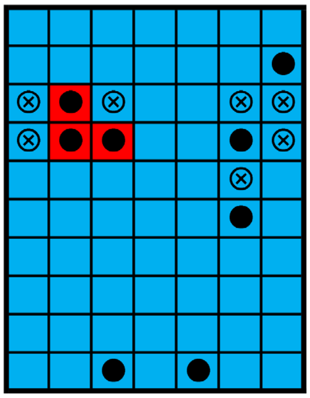

FIGURE 12 The solution for a 7 × 10 grid [Color figure can be viewed at wileyonlinelibrary.com]

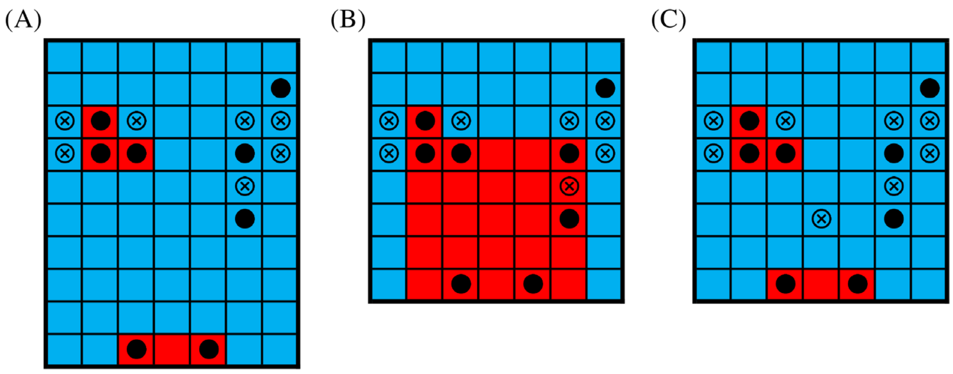

FIGURE 13 The effect of introducing γ values at vacant grid positions [Color figure can be viewed at wileyonlinelibrary.com]

As we see in Figure 13A, the addition of the γ coefficient indeed changes the optimal solution, and now we have two alert regions, one of which contains an empty position. However, if the two regions are "close", and in the given configuration they are three rows apart, as shown in Figure 13B, then the two regions merge into one. In Figure 13C, one sees that adding one no-alert position within the region has the effect of separating the two regions. The determination of which values to set and when regions should be consolidated is to be determined by nuclear detection experts and the geographical parameters of the region, as well as the density of the detectors' distribution in the region.

In conclusion, we note that further studies of simulated scenarios in Reference [9] demonstrated that the algorithm devised here is capable of detecting not only the position of a nuclear threat, but also the movement of the source, under moderate requirements on the density of the detectors.

10 | CONCLUSIONS

We present here a class of integer programming models, characterized by constraints that are monotone, that are solvable efficiently with a minimum cut procedure. The ease of recognizing this class enables the use of the same unified algorithm for a vast spectrum of applications. The sample applications presented here illustrate the versatility of this class of problems, and the computational power associated with modeling a problem as an integer program on monotone constraints.

ACKNOWLEDGMENT

Research supported in part by NSF award No. CMMI-1760102.

ORCID

Dorit S. Hochbaum https://orcid.org/0000-0002-2498-0512

REFERENCES

[1] R.K. Ahuja, D.S. Hochbaum, and J.B. Orlin, *A cut-based algorithm for the nonlinear dual of the minimum cost network flow problem*, Algorithmica **39**(3) (2004), 189–208.

[2] R.A. Achá, D.S. Hochbaum, and Q. Spaen, *HNCcorr: Combinatorial optimization for neuron identification*, Ann. Oper. Res. **289** (2020), 5–32.
[3] P. Baumann, D.S. Hochbaum, and Y.T. Yang, *A comparative study of the leading machine learning techniques and two new optimization algorithms*, Eur. J. Oper. Res. **272**(3) (2019), 1041–1057.
[4] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Trans Pattern Anal. Mach. Intell. **23**(11) (2001), 1222–1239.
[5] I.J. Cox, S.B. Rao, and Y. Zhong. "Ratio regions": A technique for image segmentation. Proceedings of the 13th International Conference on Pattern Recognition (ICPR), 1996, pp. 557–564.
[6] L.R. Ford Jr. and D.R. Fulkerson, *Maximal flow through a network*, Can. J. Math. **8** (1956), 399–404.
[7] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan, *A fast parametric maximum flow algorithm and applications*, SIAM J. Comput. **18**(1) (1989), 30–55.
[8] A.V. Goldberg, *Finding a maximum density subgraph. Technical Report UCB/CSD-84-171, EECS Department*, University of California, Berkeley, CA, 1984.
[9] D.S. Hochbaum and B. Fishbain, *Nuclear threat detection with mobile distributed sensor networks*, Ann. Oper. Res. **187**(1) (2011), 45–63.
[10] D.S. Hochbaum, C.-N. Hsu, and Y.T. Yang, *Ranking of multidimensional drug profiling data by fractional-adjusted bi-partitional scores*, Bioinformatics **28**(12) (2012), i106–i114.
[11] D.S. Hochbaum, C. Lyu, and E. Bertelli, *Evaluating performance of image segmentation criteria and techniques*, EURO J. Comput. Optim. **1**(1–2) (2013), 155–180.
[12] D.S. Hochbaum and J. Naor, *Simple and fast algorithms for linear and integer programs with two variables per inequality*, SIAM J. Comp. **23**(6) (1994), 1179–1192.
[13] D.S. Hochbaum, The pseudoflow algorithm and the pseudoflow-based simplex for the maximum flow problem. *Proc. of IPCO 98*, Lecture Notes Comput. Sci. **1412** (1998), 325–337.
[14] D.S. Hochbaum, *An efficient algorithm for image segmentation, Markov random fields and related problems*, J. ACM (JACM) **48**(4) (2001), 686–701.
[15] D.S. Hochbaum, *Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations*, Eur. J. Oper. Res. **140**(2) (2002), 291–321.
[16] D.S. Hochbaum, *The pseudoflow algorithm: A new algorithm for the maximum flow problem*, Oper. Res. **58**(4) (2008), 992–1009.
[17] D.S. Hochbaum. Polynomial time algorithms for the normalized cut problem and bi-criteria, multi-objective and ratio problems in clustering, imaging and vision grouping: Part II. unpublished, 2008.
[18] D.S. Hochbaum, "*The multi-sensor nuclear threat detection problem*," *Operations Research and Cyber-Infrastructure*, J. Chinneck, B. Kristjansson, and M. Saltzman (eds), Springer, New York, NY, 2009, pp. 389–399.
[19] D.S. Hochbaum, *Polynomial time algorithms for ratio regions and a variant of normalized cut*, IEEE Trans. Pattern Anal. Mach. Intell. **32**(5) (2010), 889–898.
[20] D.S. Hochbaum, *Multi-label Markov random fields as an efficient and effective tool for image segmentation, total variations and regularization*, Numer. Math. Theory Meth. Appl. **6**(1) (2013), 169–198.
[21] D.S. Hochbaum, *A polynomial time algorithm for Rayleigh ratio on discrete variables: Replacing spectral techniques for expander ratio, normalized cut, and Cheeger constant*, Oper. Res. **61**(1) (2013), 184–198.
[22] D.S. Hochbaum. HPF—Hochbaum's PseudoFlow, https://riot.ieor.berkeley.edu/Applications/Pseudoflow/maxflow.html. Accessed January 3, 2020.
[23] D.S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. Proc. of IEEE 12th international conference on computer vision, IEEE, 2009, pp. 269–276.
[24] T.B. Johnson, *Optimum Open Pit Mine Production Technology. PhD Thesis, Operations Research Center*, University of California, Berkeley, 1968.
[25] J.C. Lagarias, *The computational complexity of simultaneous diophantine approximation problems*, SIAM J. Comput. **14** (1985), 196–209.
[26] J.C. Picard, *Maximal closure of a graph and applications to combinatorial problems*, Manag. Sci. **22** (1976), 1268–1272.
[27] C. Rother, T. Minka, A. Blake and V. Kolmogorov. Cosegmentation of image pairs by histogram matching—Incorporating a global constraint into MRFs. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 993–1000.
[28] Q. Spaen, R.A. Achá, S.N. Chettih, M. Minderer, C. Harvey, and D.S. Hochbaum, *HNCcorr: A novel combinatorial approach for cell identification in calcium-imaging movies*, eNeuro **6** (2019), 2.
[29] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, *Hierarchy and adaptivity in segmenting visual scenes*, Nature **442**(7104) (2006), 810–813.
[30] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern. Anal. Mach. Intell. **22**(8) (2000), 888–905.
[31] Y.T. Yang, B. Fishbain, D.S. Hochbaum, E.B. Norman, and E. Swanberg, *The supervised normalized cut method for detecting, classifying, and identifying special nuclear materials*, INFORMS J. Comput. **26**(1) (2013), 45–58.

How to cite this article: Hochbaum DS. Applications and efficient algorithms for integer programming problems on monotone constraints. *Networks.* 2020;1–29. https://doi.org/10.1002/net.21983