Polynomial Time Algorithms for Ratio Regions and a Variant of Normalized Cut

Dorit S. Hochbaum

Abstract—In partitioning, clustering, and grouping problems, a typical goal is to group together similar objects, or pixels in the case of image processing. At the same time, another goal is to have each group distinctly dissimilar from the rest and possibly to have the group size fairly large. These goals are often combined as a ratio optimization problem. One example of such a problem is a variant of the normalized cut problem, another is the ratio regions problem. We devise here the first polynomial time algorithms solving optimally the ratio region problem and the variant of normalized cut, as well as a few other ratio problems. The algorithms are efficient and combinatorial, in contrast with nonlinear continuous approaches used in the image segmentation literature, which often employ spectral techniques. Such techniques deliver solutions in real numbers which are not feasible to the discrete partitioning problem. Furthermore, these continuous approaches are computationally expensive compared to the algorithms proposed here. The algorithms presented here use as a subroutine a minimum s,t-cut procedure on a related graph which is of polynomial size. The output consists of the optimal solution to the respective ratio problem, as well as a sequence of nested solutions with respect to any relative weighting of the objectives of the numerator and denominator.

Index Terms—Grouping, image segmentation, graph theoretic methods, partitioning.

1 Introduction

A major challenge in the field of imaging is vision grouping, or segmentation. The purpose of grouping and segmentation is to recognize and delineate, automatically, the salient objects in an image. Image segmentation is equivalent to partitioning the set of pixels forming the image, or to clustering, its pixels. High quality clustering is often defined by multiple attributes. As an optimization problem, this requires attaining more than one objective. The motivation for studying the ratio problems here is as examples of setting an optimization criterion involving two different goals. Our primary focus is on the ratio regions problem and the variant of normalized cut problem as both of these problems have been perceived to be NP-hard. We demonstrate here that these problems are in fact polynomially solvable. We then seek to provide insights as to what differentiates easy ratio problems from hard ones.

All problems addressed here address two different goals in the objective. The first goal in the variant of normalized cut, discussed in the paper by Sharon et al. [27], is to select a group of pixels that is as dissimilar to the remainder of the image as possible. The second goal is to maximize the similarity of the pixels within the group. These two objectives are combined in [27] as a minimization of the ratio of the first function to the second. For the ratio regions problem, discussed in the paper of Cox et al. [7], the first goal is the same—to have the selected group's pixels as dissimilar to the remainder of the image as possible. The second goal is to maximize the number of pixels within the selected set. Here as well, the combined objective is presented as a ratio of the two functions [7].

The reason why these two problems, as well as other bipartitioning optimization problems such as ratio cuts and normalized cuts, employ two goals in the objective function is that a direct and efficient method, based on the single goal of minimum cut, that selects a set that is as dissimilar as possible from the rest of the graph, does not work well in practice. The optimal solution tends to consist of a small subset of the graph—frequently a singleton node. This phenomenon was noted by Shi and Malik [25] and others. The second goal seeks to attain a group of pixels which is relatively large in size by giving weight to the similarity within the set, or the number of nodes within the set. A number of criteria were devised in order to compensate and correct for the phenomenon of small segments—the notions of *normalized cut* [25], variant normalized cut [27], and ratio regions [7] are some of them.

1.1 A Graph Representation of Image Segmentation

A graph theoretical framework is suitable for representing image segmentation and grouping problems. The image segmentation problem is presented on an undirected graph $G = (V, E)$, where V is the set of pixels and E are the pairs of adjacent pixels for which similarity information is available. Typically, one considers a planar image with pixels arranged along a grid. The four-neighbor setup is a commonly used adjacency rule with each pixel having four neighbors—two along the vertical axis and two along the horizontal axis. This setup forms a planar grid graph. The eight-neighbor arrangement is also used, but then the planar structure is no longer preserved, and complexity of various algorithms increases, sometimes significantly. Planarity is also not satisfied for three-dimensional images, and in general

- *The author is with the Department of Industrial Engineering and Operations Research, Etcheverry Hall, University of California, Berkeley, CA 94720. E-mail: hochbaum@ieor.berkeley.edu.*

Manuscript received 6 Sept. 2008; revised 17 Feb. 2009; accepted 30 Mar. 2009; published online 3 Apr. 2009.
Recommended for acceptance by N. Paragios.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-09-0599.
Digital Object Identifier no. 10.1109/TPAMI.2009.80.

clustering problems there is no grid structure and thus the respective graphs are not planar. The algorithms presented here do not assume any specific property of the graph G—they work for general graphs.

The edges in the graph representing the image carry *similarity* weights. There is a great deal of literature on how to generate similarity weights, and we do not discuss this issue here. We only use the fact that similarity is inversely increasing with the difference in attributes between the pixels. In terms of the graph, each edge $[i,j]$ is assigned a similarity weight w_{ij} that increases as the two pixels i and j are perceived to be more similar. Low values of w_{ij} are interpreted as dissimilarity. However, in some contexts, one might want to generate *dissimilarity* weights independently. In that case, each edge has two weights, w_{ij} for similarity, and $\hat{w}_{ij}$ for dissimilarity.

1.2 Notations and Preliminaries

Consider an undirected graph $G = (V, E)$. We use the common notation of $n = |V|$ the number of nodes and $m = |E|$ the number of edges in the graph G.

Let the weights of the edges in the graph be w_{ij} for $[i,j] \in E$. If the edges have two sets of weights, these will be denoted by $w_{ij}^{(1)}$ and $w_{ij}^{(2)}$.

A bipartition of the graph is called a *cut*, $(S, \bar{S}) = \{[i,j] | i \in S, j \in \bar{S}\}$, where $\bar{S} = V \setminus S$. We define the *capacity of a cut* $(S, \bar{S})$ as $C(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} w_{ij}$. More generally, for any pair of sets $A, B \subseteq V$ we define the set of edges going between these two sets as $(A, B) = \{[i,j] | i \in A, j \in B\}$. And the capacity of (A, B) is $C(A, B) = \sum_{i \in A, j \in B} w_{ij}$. We define the *capacity of a set* $A \subset V$ to be $C(A) = C(A, A) = \sum_{i,j \in A} w_{ij}$. For inputs with two sets of edge weights, $w_{ij}^{(1)}$ and $w_{ij}^{(2)}$, we let $C_1(A, B) = \sum_{i \in A, j \in B} w_{ij}^{(1)}$ and $C_2(A, B) = \sum_{i \in A, j \in B} w_{ij}^{(2)}$.

Given a partition of a graph into k disjoint components, $\{V_1, \ldots, V_k\}$ the *k-cut* value is $C(V_1, \ldots, V_k) = \frac{1}{2} \sum_{i=1}^{k} C(V_i, \bar{V}_i)$. The problem of partitioning a graph to k nonempty components that minimize the k-cut value is called the minimum k-cut. (This problem is polynomial time solvable for fixed k [12].)

Let $d_i = \sum_{[i,j] \in E} w_{ij}$ denote the sum of edge weights adjacent to node i. The weight of a subset of nodes $B \subseteq V$ is denoted by $d(B) = \sum_{j \in B} d_j$ referred to as the *volume* of B. Note that, with the notation above, $d(B) = C(B, V)$.

Let D be a diagonal matrix $n \times n$, with $d_{ii} = d_i = \sum_{[i,j] \in E} w_{ij}$. Let W be the weighted node-node adjacency matrix of the graph where $W_{ij} = W_{ji} = w_{ij}$. The matrix $D - W$ is called the Laplacian of the graph and is known to be positive semidefinite [14].

1.3 Overview of Results

Two applications of a methodology that generates efficient algorithms for ratio problems are illustrated: One is for the problem of "*normalized cut variant*," which is to minimize the ratio of the similarity between the set of objects and its complement and the similarity within the set of objects. The second problem is that of "*ratio regions*," which is to minimize the ratio of the similarity between the set of objects and its complement and the number (or weight) of the objects within the set. The algorithms described provide, in addition to an optimal solution to the ratio problem, the set of *all* solutions corresponding to all possible relative weighting of the two objectives. These solutions are often more informative than the optimal solution to the ratio problem alone, as illustrated, for example, in Fig. 6.

A list of problems which are amenable to the methodology paradigm presented here are summarized in Table 1.

TABLE 1
Ratio Optimization Problems in Image Segmentation

Problem name	Objective	Reference
Normalized cut′	$\min_{S \subset V} \frac{C(S,\bar{S})}{C(S,S)}$	[25], [27]
"Density"	$\min_{S \subset V} \frac{C(S,S)}{\|S\|}$	[24]
Ratio regions	$\min_{S \subset V} \frac{C(S,\bar{S})}{\|S\|}$	[7]
Weighted ratio regions	$\min_{S \subset V} \frac{C(S,\bar{S})}{\sum_{i \in S} q_i}$	[7]

In this table, q_i denotes an arbitrary weight assigned to node i.

We discuss here also hard ratio problems, including the Cheeger constant, the graph expander, the ratio regions, and the normalized cut problems. We provide some insight as to the distinguishing features that make these problems hard to solve.

The paper is organized as follows: We begin with a discussion of a collection of ratio problems (and sum of two ratios problems) and their relationship to each other, in Section 2. In Section 3, we present the general purpose solution technique for ratio problems that have "monotone" formulations, and apply it to the variant normalized cut problem. In Section 5.1, we apply the technique for the ratio regions problem.

2 Several Ratio Problems

We describe here four types of ratio problems: the normalized cut and its relation to the variant normalized cut problem, the graph expander problem and its relation to the ratio regions problem, the densest subgraph problem, and the "ratio cut" problem.

2.1 The Normalized Cut and Normalized Cut′ Problems

Shi and Malik addressed in their work on segmentation [25] an alternative criterion to replace minimum cut procedures. This is because the minimum cut in a graph with edge similarity weights creates a bipartition that tends to have one side very small in size. To correct for this unbalanced partition they proposed several types of objective functions, one of which is the *normalized cut*, which is a bipartition of V, $(S, \bar{S})$, minimizing:

$$\min_{S \subset V} \frac{C(S, \bar{S})}{d(S)} + \frac{C(S, \bar{S})}{d(\bar{S})}. \quad (1)$$

In such an objective function, the one ratio with the smaller value of $d()$ will dominate the objective value—it will always be at least $\frac{1}{2}$ of the objective value. Therefore, this type of objective function drives the segment S and its complement to be approximately of equal size. Indeed, like the balanced cut problem, the problem was shown to be NP-hard [25] by reduction from set partitioning.

A variant of the bipartition problem which discourages a very small set in the output is the quantity $h_G = \min \frac{C(S, V \setminus S)}{\min\{d(S), d(\bar{S})\}}$, also known as the *Cheeger constant* [5], [6]. Variants of the Cheeger constant have the denominator equal to $\min\{|S|, |V \setminus S|\}$ or $\min\{C(S,S), C(\bar{S},\bar{S})\}$. The Cheeger constant is approximated by the second largest eigenvalue of a certain related adjacency matrix of the graph. This eigenvalue λ_1 is related to the Cheeger constant by the inequalities: $2h_G \geq \lambda_1 \geq h_G^2/2$. Computing the value of the Cheeger constant is NP-hard as it is closely related to finding the expander ratio of a graph. Again it drives to a roughly equal or balanced partition of the graph. A minimization of the Cheeger constant (via a relaxation) has also been employed in an image segmentation context by Grady and Schwartz [13].

Instead of the sum of two ratios objective, there are other related optimization problems used for image segmentation. Sharon et al. [27] define a variant of the normalized cut as

$$\min_{S \subset V} \frac{C(S, \bar{S})}{C(S, S)}.$$

This objective function is equivalent to minimizing one term in (1). To see this, note that:

$$\frac{C(S, \bar{S})}{C(S, S)} = \frac{C(S, \bar{S})}{d(S) - C(S, \bar{S})} = \frac{1}{\frac{d(S)}{C(S,\bar{S})} - 1}.$$

Therefore, minimizing this ratio is equivalent to maximizing $\frac{d(S)}{C(S,\bar{S})}$ which in turn is equivalent to minimizing the reciprocal quantity $\frac{C(S,\bar{S})}{d(S)}$, which is the first term in (1). The optimal solution in the bipartition S will be the one set for which the value of the similarity within, $C(S,S)$, is the greater between the set and its complement.

Sharon et al. [27] state that:

> A salient segment in the image is one for which the similarity across its border is small, whereas the similarity within the segment is large (for a mathematical description, see Methods). We can thus seek a segment that minimizes the ratio of these two expressions. Despite its conceptual usefulness, minimizing this normalized cut measure is computationally prohibitive, with cost that increases exponentially with image size.

One of our contributions here is to show that the problem of minimizing this ratio, called here the $normalized\ cut'$ problem, is in fact solvable in polynomial time and with a combinatorial algorithm.

The typical solution approach used when addressing optimization problems for image segmentation is to approximate the problem objective by a nonlinear (quadratic) expression for which the eigenvectors of an associated matrix form an optimal solution. Let binary variables x_i for $i \in V$ be defined so that $x_i = 1$ if node i in the selected side of the cut—the segment. The following nonlinear formulation has been used by Sharon et al. and others [27], [28], [25]:

$$\min_{x_i \in \{0,1\}} \frac{\sum w_{ij}(x_i - x_j)^2}{\sum w_{ij} x_i \cdot x_j} = \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T W \mathbf{x}},$$

where L is the Laplacian matrix of the graph and W is the edge weight matrix as defined in Section 1.2. Once the constraints on the discrete values of $\mathbf{x}$, $x_i \in \{0,1\}$, are relaxed, the relaxed formulation's optimal solution is an eigenvector and thus can be solved optimally with spectral techniques. The use of spectral techniques involves several computational hurdles: The Laplacian matrices are of size $n \times n$, where n is the number of pixels. These are very large matrices for which the computation of eigenvector solution, even for small images and, even with the use of advanced sparse matrix computation techniques, is extremely challenging. Moreover, spectral techniques engage real number computations which trigger numerical difficulties. Finally, the exact solution to the nonlinear problem is a vector of real numbers, whereas the original problem is discrete and binary. So, there is a further heuristic step of how to convert the continuous solution into binary values.

This normalized cut' problem (which does not include the "balanced" requirement) is, however, polynomial time solvable. We show an algorithm solving the problem in the same complexity as a single minimum s,t-cut on a related graph on $O(n+m)$ nodes and $O(n+m)$ edges for n the number of pixels in the image, and m adjacency pairs, which in images is typically $O(n)$. Another approach in [20] generates a different, slightly smaller, graph.

2.2 Ratio Regions and Expanders

Consider the objective function $\min_{|S| \leq \frac{n}{2}} \frac{C(S,\bar{S})}{|S|}$. This value of the optimal solution, for a graph G, is known as the expansion ratio of G. This problem is NP-hard as the limit on the size of $|S|$ makes it difficult and drives the solution toward a *balanced* cut, which, as noted above, is a known NP-hard problem. The objective function can also be written as $\min_{S \subset V} \frac{C(S,\bar{S})}{\min\{|S|,|\bar{S}|\}}$.

A variant of this problem was addressed under the name *ratio regions* by Cox et al. [7]. The ratio region problem is motivated by seeking a segment, or region, where the boundary is of low cost and the segment itself has high node weight:

$$\min_{S \subset V} \frac{C(S, \bar{S})}{|S|}. \quad (2)$$

Note that this formulation does not contain the constraint $|S| \leq \frac{n}{2}$ which is present in the expander ratio problem. Hence, the segments' sizes are not necessarily balanced. The ratio regions problem studied by Cox et al. is restricted to planar graphs and thus, in the context of images, to planar grid images with four neighbors only. In the case of 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. This observation is key to the algorithm in [7]. For graph nodes of weight q_i, the problem is generalized to

$$\min_{S \subset V} \frac{C(S, \bar{S})}{\sum_{i \in S} q_i}.$$

Cox et al. [7] showed how to solve the weighted problem on planar graphs where all node weights are positive.

This weighted problem, for *any* general graph and for arbitrary weights (positive or negative), is shown here to be polynomial time solvable in the complexity of a single minimum cut. This result is not new as the problem is, in fact, equivalent to a binary and linear version of the Markov Random Fields problem, called the maximum s-excess problem in [16]. It is interesting to note that the pseudoflow algorithm in [16] is set to solve the maximum s-excess problem directly.

2.3 Densest Subgraph

Sarkar and Boyer [24] defined the problem $\min_{S\subset V}\frac{C(S,S)}{|S|}$. This objective is of interest for weights that reflect dissimilarity. In that case, the goal is to minimize the dissimilarity within the selected segment while the size of the segment tends to be large. For similarity weights, the objective would be to maximize this quantity. This problem has been known for a long time as the *maximum density subgraph*. The density of a subgraph induced by the subset of nodes D is $\frac{C(D,D)}{|D|}$. The maximum density subgraph is the subgraph induced by S that maximizes

$$\max_{S\subset V}\frac{C(S,S)}{|S|}.$$

Both this minimization problem and its maximization version were shown to be solvable in polynomial time by Goldberg [10]. Gallo et al. [9] showed how the problem would be solved as a parametric minimum s,t-cut in the complexity of a single s,t-cut.

A node weighted version of the problem is $\max_{S\subset V}\frac{C(S,S)}{q(S)}$. This problem is solved by a minor extension of the densest subgraph approach in the same runtime.

2.4 "Ratio Cuts"

This problem was introduced by Wang and Siskind [29]. In the ratio cut problem, each edge has two weights associated with it. Wang and Siskind studied the case where $w_{ij}^{(1)}$ are positive and $w_{ij}^{(2)}$ are equal to 1 for all $[i,j]\in E$. So, this objective is to find a cut minimizing the cut value divided by the number of edges in the cut. The rationale for this objective is to try and increase the number of edges in the cut and, hence, the size of the cluster/segment. The goal is to minimize the ratio

$$\min_{S\subset V}\frac{C_1(S,\bar{S})}{C_2(S,\bar{S})}.$$

This problem was shown in [29] to be at least as hard as the sparsest cut problem and therefore NP-hard. On the other hand, for planar graphs, Wang and Siskind demonstrated that the problem is solvable in polynomial time. The algorithm makes multiple calls to a nonbipartite matching algorithm that runs in polynomial time but is not efficient: For planar graphs, the λ-question, discussed and defined in Section 3.2, is solved by finding a maximum weight nonbipartite matching in a related graph. The procedure of [29] makes repeated calls to solving nonbipartite matching problems, where, for each value of λ, another graph has to be constructed. This is the main source of inefficiency of the technique, that is, to have to construct a separate graph for each call without being able to take advantage of the solution for the previous calls. This contrasts with other polynomial time algorithms reported here.

3 The Solution Approach

3.1 Monotone Integer Programming Formulation

The solution approach begins with a formulation of the problem as an integer linear programming problem with *monotone* inequalities constraints. It was shown in [18] that any integer programming formulation on monotone constraints has a corresponding graph where the minimum cut solution maps to an optimal solution to the integer programming problem. Thus, the formulation is solvable in polynomial time.

To convert the ratio objective to a linear objective, we utilize a well-known reduction of the ratio problem to a linearized optimization problem.

3.2 Linearizing Ratio Problems

A general approach for maximizing a fractional (or, as it is sometimes called, geometric) objective function over a feasible region $\mathcal{F}$, $\min_{x\in\mathcal{F}}\frac{f(x)}{g(x)}$, 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 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 $O(\log(UF))$ times in order to solve the problem, where U is an upper bound on the value of the numerator and F an upper 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. Note that the number of calls to the linear optimization is not strongly polynomial but rather, if binary search is employed, depends on the logarithm of the magnitude of the numbers in the input. In some cases, however, there is a more efficient procedure. Several examples are the densest subgraph problem which has an efficient parametric procedure [9], the *Fast-PD MRF* that exploits information coming not only from the original MRF problem, but also from a dual problem [23], and the dynamic graph cuts approach that suggests using the flow obtained during the computation of the max-flow corresponding to a particular problem instance for solving a similar (or subsequent) instance of the problem by dynamically updating the solution of the previous one [21].

It is important to note that *not all* ratio problems are solvable in polynomial time. One prominent example is the *ratio cuts* discussed in Section 2.4. For that problem, the linearized version is NP-hard by reduction from maximum cut.

It is also important to note that linearizing is not always the right approach to use for a ratio problem. For example, the ratio problem of finding a partition of a graph to k components minimizing the k-cut between components for $k\geq 2$ divided by the number of components k always has an optimal solution with $k=2$ which is attained by a, polynomial time, minimum two-cut algorithm. On the other hand, the linearized problem is NP-hard to solve since it is equivalent to solving the minimum k-cut problem. (It can, however, be solved in polynomial time for fixed k by the algorithm in [12].)

4 The Normalized Cut Formulation

We first provide a formulation for the problem,

$$\min_{S \subset V} \frac{C_1(S, \bar{S})}{C_2(S, S)}.$$

This is a slight generalization of normalized cut' 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,

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

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

With these variables, the following is a valid formulation (NC) of the normalized cut problem:

$$\begin{array}{ll} \text{(NC) min} & \dfrac{\sum w_{ij} z_{ij}}{\sum w'_{ij} y_{ij}}, \\ \text{subject to} & x_i - x_j \le z_{ij} \quad \text{for all } [i,j] \in E, \\ & x_j - x_i \le z_{ji} \quad \text{for all } [i,j] \in E, \\ & y_{ij} \le x_i \quad \text{for all } [i,j] \in E, \\ & y_{ij} \le x_j, \\ & 1 \le \sum_{[i,j] \in E} y_{ij} \le |E| - 1, \\ & x_j \text{ binary } j \in V, \\ & z_{ij} \text{ binary } [i,j] \in E, \\ & y_{ij} \text{ binary } i, j \in V. \end{array}$$

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. With the constraints, z_{ij} cannot be 0 unless both end points i and j are in the same set. On the other hand, y_{ij} cannot be equal to 1 unless both end points i and j are in S.

The sum constraint ensures that at least one edge is in the segment 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. Instead of including explicitly the sum constraint in the formulation, we replace it by setting some edge in the feature to serve as "seed" and some edge in the background to serve as "seed." Adding seeds is an approach often used in segmentation, see, e.g., [3], [13]. The latter demonstrated that the segmentation solution is largely independent of the choice of the "seed" nodes. Moreover, it suggested automatically choosing the nodes with maximum degree since a node of high degree will be in the interior of a region or in an area of uniform intensity in the context of image processing. Since, for both the feature and its complement, the cut value is the same, the solution will always be in terms of the larger segment in the bipartition that is likely to contain higher total similarity weights. We thus replace the sum constraint by *automatically* setting $y_{i^*j^*} = 1$ and $y_{i'j'} = 0$ for some pair of edges in the feature and the background, respectively.

Excluding the sum constraint, the problem formulation (NC) is easily recognized as a monotone integer programming with up to three variables per inequality according to the definition provided in Hochbaum's work [18]. Such linear optimization problems were shown there to be solvable as a minimum cut problem on a certain associated graph.

4.1 Linearizing the Objective Function

Consider the ratio objective function for the normalized cut' $\min_{S \subset V} \frac{C_1(S,\bar{S})}{C_2(S,S)}$. Because this objective function is a ratio, we first "linearize" the problem. A linearization procedure is to set it as calls to the *λ-question* which asks whether $\min_{S \subset V} \frac{C_1(S,\bar{S})}{C_2(S,S)} < \lambda$. The *λ-question* for the normalized cut' problem can be stated as the following linear optimization question:

Is there a feasible subset $V' \subset V$ such that

$$\sum_{[i,j] \in E} w_{ij} z_{ij} - \lambda \sum_{[i,j] \in E} w'_{ij} y_{ij} < 0?$$

One possible approach for using the solution to the λ-question in order to solve the ratio problem is to utilize a binary search procedure that calls for the λ-question $O(\log(UF))$ times in order to solve the problem, where $U = \sum_{[i,j] \in E} w_{ij}$ and $F = \sum_{[i,j] \in E} w'_{ij}$ for the weights at the denominator w'_{ij}.

With the construction of the graph, we observe that one can instead use a parametric approach which is significantly more efficient. Several applications of parametric maximum flow, or minimum cut, in computer vision have been suggested in [17] and [22]. We note that the λ-question is the following *monotone* optimization problem:

$$\begin{array}{ll} (\lambda\text{-NC})\text{min} & \sum_{[i,j] \in E} w_{ij} z_{ij} - \lambda \sum_{[i,j] \in E} w'_{ij} y_{ij}, \\ \text{subject to} & x_i - x_j \le z_{ij} \quad \text{for all } [i,j] \in E, \\ & x_j - x_i \le z_{ji} \quad \text{for all } [i,j] \in E, \\ & y_{ij} \le x_i \quad \text{for all } [i,j] \in E, \\ & y_{ij} \le x_j, \\ & y_{i^*j^*} = 1 \text{ and } y_{i'j'} = 0, \\ & x_j \text{ binary } j \in V, \\ & z_{ij} \text{ binary } [i,j] \in E, \\ & y_{ij} \text{ binary } i, j \in V. \end{array}$$

If the optimal value of this problem is negative, then the answer to the λ-question is yes; otherwise the answer is no. This problem (λ-NC) is an integer optimization problem on a totally unimodular constraint matrix. That means that we can solve the linear programming relaxation of this problem and get a basic optimal solution that is integer. Instead, we will use a much more efficient algorithm described in [18] which relies on the monotone property of the constraints.

There is an alternative formulation to the linearized optimization problem which does not introduce the vari-

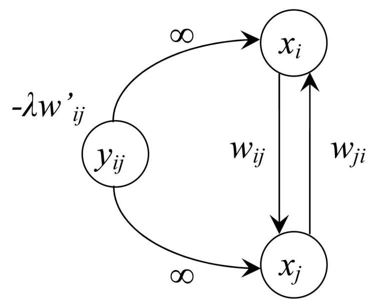

Fig. 1. The basic gadget in the graph representation.

ables z_{ij} and removes the constraints. This formulation can be thought of as a special case of MRF, as presented in [17], with piecewise linear "energy" (or "separation") functions:

$$f^{(1)}(\phi) = \begin{cases} 0, & \text{if } \phi \leq 0, \\ \phi, & \text{if } \phi > 0, \end{cases}$$

$$f^{(2)}(\phi) = \begin{cases} 0, & \text{if } \phi \leq 0, \\ M \cdot \phi, & \text{if } \phi > 0, \end{cases}$$

where $M \gg 1$. A sufficiently large value of M is $M = \sum_{ij} w_{ij} + \lambda \sum_{ij} w'_{ij}$.

The problem is then the following MRF problem:

$$\begin{array}{ll} \text{(MRF-}\lambda\text{-NC)} & \\ \quad \min & \sum_{[i,j]\in E} w_{ij}\big[f^{(1)}(x_i - x_j) + f^{(1)}(x_j - x_i)\big] \\ & + [\sum_{[i,j]\in E} f^{(2)}(y_{ij} - x_i) + f^{(2)}(y_{ij} - x_j)] \\ & - \lambda \sum_{ij} w'_{ij} \cdot y_{ij}, \\ \text{subject to} & x_j \text{ binary } j \in V, y_{ij} \text{ binary } i,j \in V. \end{array}$$

This MRF formulation was shown in [17] to be solvable in polynomial time, through solving a parametric min-cut. The resulting construction is identical to the one presented next.

4.2 Solving the λ-Question with a Minimum Cut Procedure

We construct 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 end point nodes x_i and x_j have infinite capacity. Fig. 1 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-NC. 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-NC. 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 end points nodes are effectively shrunk with s. The node $y_{i'j'}$ and its end points nodes are analogously shrunk with the sink t. We denote this graph illustrated in Fig. 2, G'_{st}.

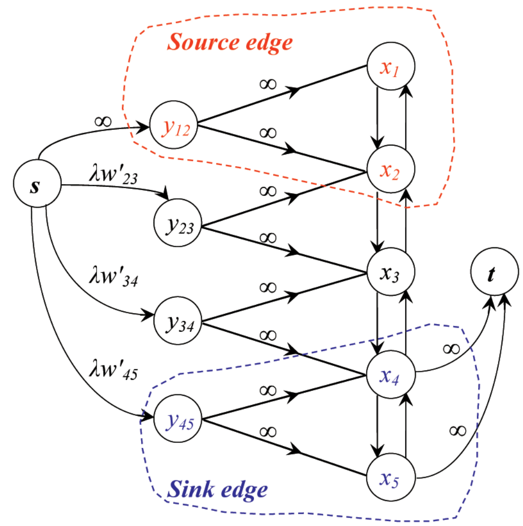

Fig. 2. The graph G'_{st} with edge $[1,2]$ as source seed and edge $[4,5]$ as sink seed.

Theorem 4.1. *A minimum s,t-cut in the graph G'_{st}, (S,T), corresponds to an optimal solution to λ-NC by setting all of the variables whose nodes belong to S to 1 and zero otherwise.*

Proof. Note that, whenever a node y_{ij} is in the sink set T, the arc connecting it to the source is included in the cut. Let the set of x variable nodes be denoted by V_x and the set of y variable nodes, excluding $y_{i^*j^*}$, be denoted by V_y. Let (S,T) be any finite cut in G'_{st} with $s \in S$ and $t \in T$ and capacity $C(S,T)$.

$$\begin{aligned} C(S,T) &= \sum_{y_{ij}\in T\cap V_y} \lambda w'_{ij} + \sum_{i\in V_x\cap S, j\in V_x\cap T} w_{ij} \\ &= \sum_{v\in V_y} \lambda w'_v - \sum_{y_{ij}\in S\cap V_y} \lambda w'_{ij} + \sum_{x_i\in V_x\cap S, x_j\in V_x\cap T} w_{ij} \\ &= \lambda W' + \left[\sum_{i\in V_x\cap S, j\in V_x\cap T} w_{ij} - \sum_{y_{ij}\in S\cap V_y} \lambda w'_{ij}\right]. \end{aligned}$$

This proves that, for a fixed constant $W' = \sum_{v\in V_y} w'_v$, the capacity of a cut is equal to a constant $W'\lambda$ plus the objective value corresponding to the feasible solution. Hence, the partition (S,T) minimizing the capacity of the cut minimizes also the objective function of λ-NC. □

4.3 A Parametric Procedure for Solving Normalized Cut′

The capacities of the source-adjacent arcs in the graph G'_{st} are monotone increasing with λ. As the value of λ increases, the source set of the respective minimum cuts is nested. This is called the nestedness lemma in [16]. Although the capacity of the cut is increasing with an increase in the value of λ, the set of nodes in the source set can be incremented only $n' = |V'|$ times. We call the values of λ where the source set expands by at least one node the *break points* of the parametric cut. Let the break points be $\lambda_1 > \lambda_2 > \cdots > \lambda_\ell$, with corresponding *minimal source sets*,

(a) (b) (c)

Fig. 3. Normalized cut segmentation. (a) The input to the normalized cut procedure. (b) The output using Shi's implementation, and (c) the output using the algorithm described here.

(a) (b) (c)

Fig. 4. Normalized cut segmentation. (a) The input to the normalized cut procedure. (b) The output using Shi's implementation, and (c) the output using the algorithm described here.

$S_1 \subset S_2 \subset \cdots \subset S_\ell$.[1] As a result of the nestedness lemma, $\ell \leq n'$ for a graph on n' nodes since there can be no more than n' different nested source sets. The capacity value of the minimum cut is increasing as a function of increasing values of λ along a piecewise linear concave curve.

Theorem 4.2. *All break points of the density graph can be found by solving a parametric minimum cut problem where the source-adjacent capacities of arcs are linear functions of the parameter, λ.*

Gallo et al. [9] showed how to find all of the break points and the corresponding minimum cuts in the same complexity as that required to solve a single minimum s,t-cut problem with the push-relabel algorithm of [11]. The pseudoflow algorithm for maximum flow and minimum cut (see [16]) also finds the parametric break points in the complexity of a single minimum s,t-cut. Once all of the break points are generated, we search for the largest value of λ among the break points so that the optimal value of λ-NC is negative, or equivalently, the minimum s,t-cut value that is strictly less than $\lambda W'$.

To summarize, let $T(n,m)$ be the running time required to solve the minimum cut problem on a graph with n nodes and m arcs. In the graph G'_{st}, the number of nodes is $n' = n + m$, where m is the number of adjacencies or edges in the image graph. The number of arcs in G'_{st}, $m' = |A'|$, is $O(m)$. For a general graph, this running time is $O(m^2 \log m)$ with either the pseudoflow algorithm or the push-relabel algorithm. The degree of each node is constant for imaging applications so, for that context, $m' = O(n)$ and n' is $O(n)$ and the running time is $O(n^2 \log n)$.

Theorem 4.3. *The normalized cut problem is solvable in the running time of a minimum s,t-cut problem, $T(n', m')$.*

Remark. It may be desirable to solve (λ-NC) without specifying a source and a sink. The problem is then to partition the graph G'_{st} to two nonempty components so that the cut separating them is minimum. This problem is the directed minimum two-cut problem. It was shown by Hao and Orlin [15] that the directed minimum two-cut problem is solved in the same complexity as a single minimum s,t-cut problem, with the push-relabel algorithm. (This was shown to hold also for the pseudoflow algorithm.)

1. A source set S is *minimal* if there is no other minimum cut with a source set strictly contained in S.

In order to solve the normalized cut' problem, the algorithm produces a sequence of nested solutions for all possible values of the parameter λ. Each such solution represents a different weighting of the cut objective versus the similarity objective. As the value of the λ grows, the similarity part of the objective is more prominent and the optimal solution S_λ expands. Although the normalized cut ratio problem's optimal solution comprises of a single connected component, the sequence of optimal solutions to the range of parameter values is not necessarily formed of a single connected component. Such solutions could be more meaningful in medical images, for instance, where lesions are the features sought, but they often appear as disjoint components in the image. In fact, we show in the experiments in Section 5.1 that choosing a break point which does not correspond to the optimal ratio, produces visually best results as compared to the segmentation corresponding to the minimum ratio.

4.4 Experiments with Normalized Cut

The normalized cut procedure was implemented using the pseudoflow algorithm [16], which solves the minimum s,t-cut problem (and the maximum flow problem.) The pseudoflow algorithm is fast in theory and in practice [4], and is currently the fastest algorithm on standard and nonstandard benchmark problems. The code and its parametric version are available for download at http://riot.ieor.berkeley.edu/riot/Applications/Pseudoflow. The segmentation was performed on a 32-bits Windows Vista (SP1) machine with 2.00 GHz Intel Core 2 Duo CPU ($T7300$) and 2 GBytes of memory. The results presented in Figs. 3c and 4c were computed in 0.413 and 0.291 seconds, respectively. The segmentation results obtained using the pseudoflow algorithm were compared to the results obtained when the images were segmented with the implementation provided by Shi [26] and is based on [25].

Figs. 3 and 4 depict the NC-segmentation results for the two different implementations ([25] and [4]). Figs. 3a and 4a are the original images, the images in Figs. 3b and 4b are the normalized cut segmentations resulting from using Shi and Malik's algorithm, and the images in Figs. 3c and 4c are the normalized cut' segmentations through our algorithm. The similarity weight for two adjacent nodes i,j is a function of the distance between the input color (or gray levels) v_i, v_j of the two nodes, $d(v_i, v_j)$, $e^{-\alpha d(v_i, v_j)}$ for a constant α.

Since the algorithm of Shi and Malik seeks to optimize the normalized cut criterion, as defined by (1), we choose

TABLE 2
Normalized Cut Measure for Image Segmentation

Optimization Algorithm	Shi & Malik	Algorithm Here
Figure 3	$127 \cdot 10^{-4}$	$1.466 \cdot 10^{-4}$
Figure 4	$35 \cdot 10^{-4}$	$1.702 \cdot 10^{-4}$

here the value of the objective in (1) as a performance measure as to the quality of the segmentation. The values of that objective for the segmented images are given in Table 2. Although the algorithm here solves, optimally, the normalized cut' and not the normalized cut problem it gives better (lower) results in terms of the *normalized cut* objective.

5 The Ratio Regions Formulation

5.1 Solving the Ratio Regions Problem

The linearization of the weighted ratio regions problem, is an instance of the s-excess problem in [16]. For this reason, we provide here only a sketch of the algorithm for solving the problem.

As before, we formulate the problem first. Let

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

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

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

Let the similarity weight on each edge be w_{ij} and the weight of node (pixel) j be d_j. With these parameters, the ratio regions (RR) problem formulation becomes

$$\begin{array}{lll} \text{(RR) min} & \frac{\sum w_{ij} z_{ij}}{\sum q_j x_j}, & \\ \text{subject to} & x_i - x_j \le z_{ij} & \text{for all } [i,j] \in E, \\ & x_j - x_i \le z_{ij} & \text{for all } [i,j] \in E, \\ & x_j \text{ binary } j \in V, & \\ & z_{ij} \text{ binary } [i,j] \in E. & \end{array}$$

The corresponding λ-question is

$$\begin{array}{lll} (\lambda\text{-RR})\text{min} & \sum_{[i,j]\in E} w_{ij} z_{ij} - \sum_{j \in V} \lambda q_j x_j, & \\ \text{subject to} & x_i - x_j \le z_{ij} & \text{for all } [i,j] \in E, \\ & x_j - x_i \le z_{ij} & \text{for all } [i,j] \in E, \\ & x_j \text{ binary } j \in V, & \\ & z_{ij} \text{ binary } [i,j] \in E. & \end{array}$$

Again, an alternative definition of the linearized problem as an MRF problem is using the definitions of $f^{(1)}$ and M as in Section 4.1. The corresponding MRF-λ-RR formulation is given by:

$$\begin{array}{ll} (\text{MRF-}\lambda\text{-RR}) & \\ \quad\text{min} & \sum_{[i,j]\in E} w_{ij} \left[f^{(1)}(x_i - x_j) + f^{(1)}(x_j - x_i) \right] \\ & -\lambda \sum_{j\in V} \lambda q_j x_j, \\ \text{subject to} & x_j \text{ binary } j \in V. \end{array}$$

The graph constructed for that problem is of the same size as the original graph G. Each node representing a variable x_j has an arc going to sink node with capacity λq_j.

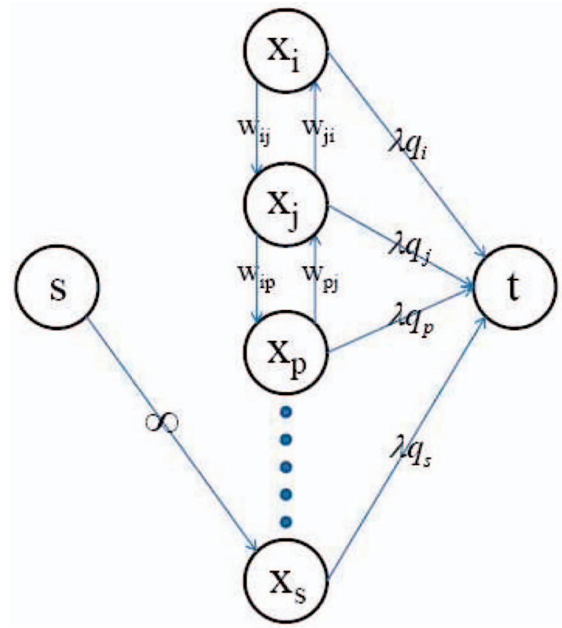

Fig. 5. The graph G'_{st} for the ratio regions problem with node x_s serving as source seed.

One variable node, x_s, is selected arbitrarily as corresponding to a source "seed."

The graph G'_{st} shown in Fig. 5 has $O(n)$ nodes and $O(m)$ arcs. The algorithm solving the problem is then a simple parametric cut algorithm in that graph, with runtime $T(n,m)$. Furthermore, the parametric s,t-cut algorithm delivers the sequence of optimal nested solutions for all values of λ, as well as the optimal solution to the ratio problem, in runtime $T(n,m)$.

5.2 Experiments with Ratio Regions

Similarly to the normalized cut procedure, the Ratio Region algorithm was implemented using the pseudoflow algorithm [16]. The running time for segmenting Figs. 6c and 6d were 1.838 and 2.242 seconds, respectively. The segmentation results obtained using the pseudoflow algorithm were compared to the results reported by Cox et al. [7].

Fig. 6 presents the RR-segmentation results. Fig. 6a is the original image; Fig. 6b is the segmentation reported in [7]. Figs. 6c and 6d are the segmentation results, utilizing the algorithm here, for two different λ values. The λ value for Fig. 6c is the optimal value of λ for the minimum ratio region objective. The λ value used for achieving the segmentation depicted in Fig. 6d was chosen to be $4 \cdot 10^{-7}$. The similarity weights w_{ij} used here are the same as defined in [7] and are based on the histogram of the absolute gradient, $|x_i - x_j|$. If X is the random variable of the absolute gradient of all the edges, than w_{ij} is defined by:

$$w_{ij} = Prob\big(X > |x_i - x_j|\big)$$

We use the ratio region objective value, as defined in (2), as a performance measure for evaluating the ratio regions results. This measure for the segmentation results in Figs. 6b, 6c, and 6d are given in Table 3.

In evaluating the results, it should be noted that the algorithm in this paper aims at giving an exact solution for a given segmentation criterion, rather than suggesting better

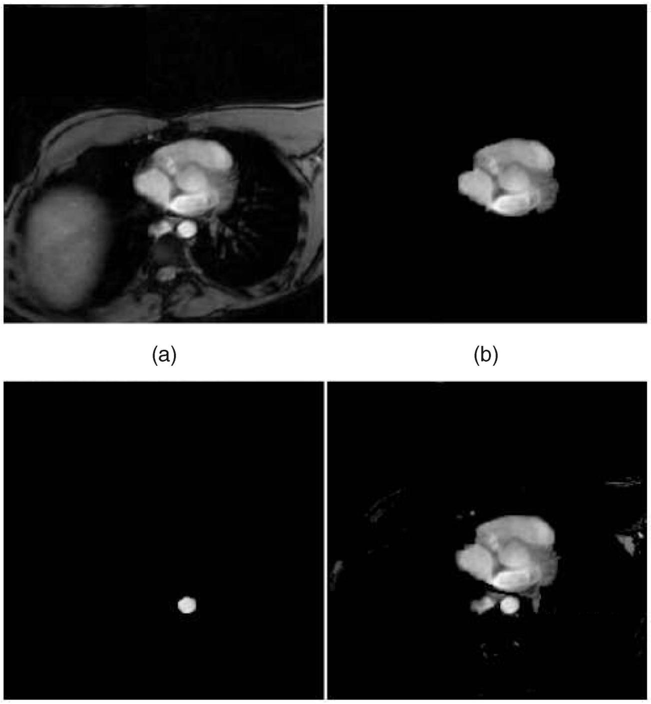

Fig. 6. Ratio Regions segmentation. (a) The input. (b) The output using Cox's implementation. (c) The optimal ratio region segmentation using the algorithm here. (d) A segmentation using the linearized problem for $\lambda = 4 \cdot 10^{-7}$.

segmentation scheme. Therefore, while the visual quality of the segmentation through minimization of the ratio regions measure in Fig. 6c is not representative of what might be perceived as the "feature," it is the optimal solution and, in that regard, casts doubt on the use of the ratio regions criterion as stated. On the other hand, the linear combination of the numerator and denominator with the choice of λ as in Fig. 6d provides a good quality visual for the image here.

6 The Maximum Density Problem

For formulating the maximum density problem, as before let x_i and y_{ij} be:

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

and

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

Then, the linearized maximum density problem corresponding λ question is

TABLE 3
Ratio Regions Measure for Image Segmentation

Cox et. al Figure 6(b)	minimum ratio regions $\lambda = 8 \cdot 10^{-3}$ Figure 6(c)	$\lambda = 4 \cdot 10^{-7}$ Figure 6(d)
$8.1349 \cdot 10^{-5}$	$7.6291 \cdot 10^{-9}$	$17 \cdot 10^{-4}$

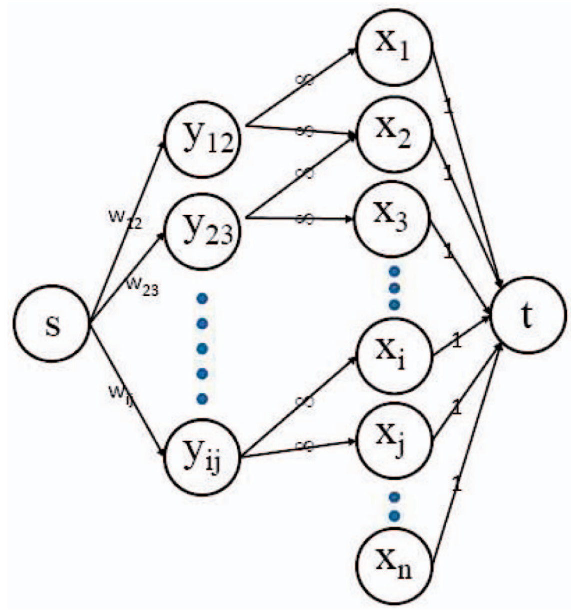

Fig. 7. The graph for solving the densest Region Problem.

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

The corresponding MRF formulation of the linearized version, with $f^{(2)}$ and M as in Section 4.1, is

$$\begin{array}{ll} (\text{MRF-}\lambda\text{-MD}) \min & \sum_{j \in V} \lambda x_j \\ + \sum_{[i,j]\in E} w_{ij} [\sum_{[i,j]\in E} & f^{(2)}(y_{ij} - x_i) + f^{(2)}(y_{ij} - x_j)], \\ & x_j \text{ binary } j \in V, \\ & y_{ij} \text{ binary } i, j \in V. \end{array}$$

The λ-question is presented as a minimum s,t-cut problem on an unbalanced bipartite graph G^b. As illustrated in Fig. 7, the graph G^b is constructed so that 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 on 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 corresponding node on G^b to the edge y_{23} on G is connected to the nodes in G^b that correspond to the nodes x_2 and x_3 on G (details are available in [19]). That bipartite graph has $m + n$ nodes, and $m' = O(m)$ arcs. 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 as discussed next.

The number of iterations required by the push-relabel algorithm or the pseudoflow algorithm is bounded by a function of the length of the longest residual path in the graph—$O(m'n')$—where m' is the number of arcs in the bipartite graph and n' is the maximum residual path length. In the λ-network constructed for the λ-question, this length n' is at most $2n + 2$ as each path alternates between the two sets in the partition.

This fact is used by Ahuja et al. [1], who devised improved push-relabel algorithms for unbalanced bipartite graphs. Among those, the most efficient for parametric minimum cut is an adaptation of the parametric push-relabel algorithm of

Gallo et al. with runtime $O(m'n'\log(\frac{n'^2}{m'}+2))$. This runtime translates to $O(mn\log(\frac{n^2}{m}+2))$ for the parametric problem solving the minimum density problem on a general graph, improving on the $O(m^2\log m)$ complexity for a direct application of the parametric cut algorithm.

7 Conclusions

We show here that several ratio problems used in image segmentation can be solved in polynomial time using a parametric minimum cut procedure. This is done here particularly efficiently with the pseudoflow algorithm. Algorithms for solving a variant of normalized cut, ratio regions, and the maximum density problem are detailed.

The segmentation results presented are competitive with other algorithms for normalized cut and ratio regions. Moreover, these solutions have much better (lower) objective value compared to other algorithms.

Acknowledgments

This is to thank Barak Fishbain for his contributions on running the experiments and Yarden Fishbain for gracefully posing for the photo in Fig. 3.

References

[1] R.K. Ahuja, J.B. Orlin, C. Stein, and R.E. Tarjan, "Improved Algorithms for Bipartite Network Flow," *SIAM J. Computing,* vol. 23, no. 5, pp. 906-933, 1994.

[2] A. Blake and A. Zisserman, *Visual Reconstruction.* MIT Press, 1987.

[3] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," *Proc. Int'l Conf. Computer Vision,* vol. I, pp. 105-112, 2001.

[4] B. Chandran and D.S. Hochbaum, "A Computational Study of the Pseudoflow and Push-Relabel Algorithms for the Maximum Flow Problem," *Operations Research,* vol. 57, pp. 358-376, Mar. 2009.

[5] J. Cheeger, "A Lower Bound for the Smallest Eigenvalue of the Laplacian," *Problems in Analysis,* R.C. Gunning, ed., pp. 195-199, Princeton Univ. Press, 1970.

[6] F.R.K. Chung, *Spectral Graph Theory.* Am. Math. Soc., 1997.

[7] I.J. Cox and S.B. Rao, Y. Zhong, "'Ratio Regions': A Technique for Image Segmentation," *Proc. Int'l Conf. Pattern Recognition,* vol. 2, pp. 557-564, Aug. 1996.

[8] W.H. Cunningham, "Optimal Attack and Reinforcement of a Network," *J. ACM,* vol. 32, pp. 549-561, 1985.

[9] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan, "A Fast Parametric Maximum Flow Algorithm and Applications," *SIAM J. Computing,* vol. 18, no. 1, pp. 30-55, 1989.

[10] A.V. Goldberg, "Finding a Maximum Density Subgraph," Technical Report UCB/CSD/84/171, Univ. of California, Berkeley, 1984.

[11] A.V. Goldberg and R.E. Tarjan, "A New Approach to the Maximum Flow Problem," *J. ACM,* vol. 35, pp. 921-940, 1988.

[12] O. Goldschmidt and D.S. Hochbaum, "A Polynomial Algorithm for the K-Cut Problem," *Math. Operations Research,* vol. 19, no. 1, pp. 24-37, 1994.

[13] L. Grady and E. Schwartz, "Isoperimetric Graph Partitioning for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 3, pp. 469-475, Mar. 2006.

[14] K.M. Hall, "An R-Dimensional Quadratic Placement Algorithm," *Management Science,* vol. 17, no. 3, pp. 219-229, 1970.

[15] J. Hao and J.B. Orlin, "A Faster Algorithm for Finding the Minimum Cut in a Directed Graph," *J. Algorithms,* vol. 17, no. 3, pp. 424-446, 1994.

[16] D.S. Hochbaum, "The Pseudoflow Algorithm: A New Algorithm for the Maximum Flow Problem," *Operations Research,* vol. 58, no. 4, pp. 992-1009, July/Aug. 2008, extended Abstract in "The Pseudoflow Algorithm and the Pseudoflow-Based Simplex for the Maximum Flow Problem," *Proc. Int'l Conf. Integer Programming and Combinatorial Optimization,* pp. 325-337, June 1998.

[17] D.S. Hochbaum, "An Efficient Algorithm for Image Segmentation, Markov Random Fields and Related Problems," *J. ACM,* vol. 48, no. 4, pp. 686-701, 2001.

[18] D.S. Hochbaum, "Solving Integer Programs over Monotone Inequalities in Three Variables: A Framework for Half Integrality and Good Approximations," *European J. Operational Research,* vol. 140, no. 2, pp. 291-321, 2002.

[19] D.S. Hochbaum, "Dynamic Evolution of Economically Preferred Facilities," *European J. Operational Research,* vol. 193, pp. 649-659, 2009.

[20] D.S. Hochbaum, "Replacing Spectral Techniques for the Normalized Cut Problem by Polynomial Time Algorithms," manuscript Univ. of California, Berkeley, June 2008.

[21] P. Kohli and P.H.S. Torr, "Dynamic Graph Cuts for Efficient Inference in Markov Random Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 12, pp. 2079-2088, Dec. 2007.

[22] V. Kolmogorov, Y. Boykov, and C. Rother, "Applications of Parametric Maxflow in Computer Vision," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1-8, 2007.

[23] N. Komodakis, G. Tziritas, and N. Paragios, "Fast, Approximately Optimal Solutions for Single and Dynamic MRFs," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2007.

[24] S. Sarkar and K.L. Boyer, "Quantitative Measures of Change Based on Feature Organization: Eigenvalues and Eigenvectors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 478-483, 1996.

[25] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.

[26] MATLAB Normalized Cuts Segmentation Code, http://www.cis.upenn.edu/jshi/software/, 2008.

[27] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and Adaptivity in Segmenting Visual Scenes," *Nature,* vol. 442, pp. 810-813, Aug. 2006.

[28] D.A. Tolliver and G.L. Miller, "Graph Partitioning by Spectral Rounding: Applications in Image Segmentation and Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1053-1060, 2006.

[29] S. Wang and J.M. Siskind, "Image Segmentation with Ratio Cut," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 6, pp. 675-690, June 2003.

Dorit S. Hochbaum received the PhD degree from the Wharton School of Business at the University of Pennsylvania. She is a full professor at the University of California, Berkeley. She is a professor in the Department of Business Administration and in the Department of Industrial Engineering and Operations Research (IEOR). Her research interests are in the design and analysis of efficient algorithms, approximation algorithms, network flow related problems, efficient utilization of resources, and discrete and continuous optimization. Her recent application areas of research relate to developing efficient methodologies for image segmentation, clustering, customer segmentation, prediction, ranking, group decision making, and data mining. She has also investigated inverse problems with applications varying from medical prognosis and error correction to financial risk assessment and prediction. She is currently serving as a co-PI on the DOmestic NUclear Threat Security (DONUTS) project for the Department of Homeland Security, for which she is developing pattern recognition and detection of hidden features methodologies. She served as the chair of the Manufacturing and Information Technology Group at Haas School of Business. She is the founder and director of the UC Berkeley Supply Chain Initiative. She is the founder and codirector of the RIOT project. She is the author of more than 130 papers that appeared in the operations research, management science, and theoretical computer science literature. She serves as the department editor for Management Science Department of *Optimization and Modelling* and on the editorial board of *Networks* and on the advisory board of *Algorithms and Operations Research*. Previously, she served on the editorial boards of *Operations Research* and *Operations Research Letters*. In 2004 she received an honorary doctorate of sciences from the University of Copenhagen for her work on approximation algorithms. In 2005, she was awarded the title of INFORMS fellow.