An Efficient Algorithm for Image Segmentation, Markov Random Fields and Related Problems

DORIT S. HOCHBAUM

University of California, Berkeley, Berkeley, California

Abstract. Problems of statistical inference involve the adjustment of sample observations so they fit some a priori rank requirements, or order constraints. In such problems, the objective is to minimize the *deviation cost* function that depends on the distance between the observed value and the modify value. In Markov random field problems, there is also a pairwise relationship between the objects. The objective in Markov random field problem is to minimize the sum of the deviation cost function and a penalty function that grows with the distance between the values of related pairs—*separation function*.

We discuss Markov random fields problems in the context of a representative application—the *image segmentation* problem. In this problem, the goal is to modify color shades assigned to pixels of an image so that the penalty function consisting of one term due to the deviation from the initial color shade and a second term that penalizes differences in assigned values to neighboring pixels is minimized. We present here an algorithm that solves the problem in polynomial time when the deviation function is convex and separation function is linear; and in strongly polynomial time when the deviation cost function is linear, quadratic or piecewise linear convex with few pieces (where "few" means a number exponential in a polynomial function of the number of variables and constraints). The complexity of the algorithm for a problem on n pixels or variables, m adjacency relations or constraints, and range of variable values (colors) U, is $O(T(n, m) + n \log U)$ where $T(n, m)$ is the complexity of solving the minimum s, t cut problem on a graph with n nodes and m arcs. Furthermore, other algorithms are shown to solve the problem with convex deviation and convex separation in running time $O(mn \log n \log nU)$ and the problem with nonconvex deviation and convex separation in running time $O(T(nU, mU))$. The nonconvex separation problem is NP-hard even for fixed value of U.

For the family of problems with convex deviation functions and linear separation function, the algorithm described here runs in polynomial time which is demonstrated to be fastest possible.

Categories and Subject Descriptors: F.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.1.6 [**Numerical Analysis**]: Optimization—*convex programming; integer programming*; G.2.2 [**Discrete Mathematics**]: Graph Theory; G.2.3. [**Discrete Mathematics**]: Applications; I.4.6 [**Image Processing and Computer Vision**]: Segmentation

General Terms: Algorithms

Additional Key Words and Phrases: Convex optimization, Markov random fields, parametric minimum cut, strongly polynomial algorithms

---

The research was supported in part by National Science Foundation awards DMI 97-13482 and DMI 00-84857.

Author's address: Haas School of Business and Dept. of IE&OR, Etcheverry Hall, University of California, Berkeley, Berkeley, CA 94720-1777, e-mail: hochbaum@ieor.berkeley.edu.

1. *Introduction*

Many problems of statistical inference, or Markov random field problems, involve sample observations that do not conform to prior information about the properties of the data. In the problem of image segmentation the image is transmitted and degraded by noise. The goal is to reset the values of the colors to the pixels so as to minimize the penalty for the deviation from the observed colors, and furthermore, so that the discontinuity in terms of separation of colors between adjacent pixels is as small as possible. In other statistical inference problems (see Barlow et al. [1972] for an extensive survey), the objective is to adjust the observed values so that the assumed order and ranking relationship is satisfied.

We consider the image segmentation problem as a prototype for the class of Markov Random Fields problems studied here. This class of problems is also known as the *metric labelling* problem [Kleinberg and Tardos 1999].

Consider an image constituting of a set of pixels each with a given color and a neighborhood relation between pairs of pixels. In the image segmentation problem, each pixel gets a color assignment that may be different from the given color of the pixel so that neighboring pixels will tend to have the same color assignment. The aim is to modify the given color values as little as possible while penalizing changes in color between neighboring pixels. The penalty function thus has two components: the deviation cost that accounts for modifying the color assignment of each pixel, and the separation cost that penalizes pairwise discontinuities in color assignment for each pair of neighboring pixels. This problem has been studied over the past two decades.[1] The case of primary interest for image segmentation has the separation cost nonconvex, which is an NP-hard problem as noted in the sequel.

Although the image segmentation problem has the pixels embedded in the plane, the results described are applicable to any arbitrary graph that is not necessarily Euclidean or planar.

Representing the image segmentation problem as a graph problem, we let the pixels be nodes in a graph and the pairwise neighborhood relation be indicated by edges between neighboring pixels. Each pairwise adjacency relation $\{i, j\}$ is replaced by a pair of two opposing arcs (i, j) and (j, i) each carrying a capacity representing the penalty function for the case that the color of j is greater than the color of i and vice versa. The set of directed arcs representing the adjacency (or neighborhood) relation is denoted by A. We denote the set of neighbors of i, or those nodes that have pairwise relation with i, by $N(i)$. Thus, the problem is defined on a graph $G = (V, A)$.

Let each node j have a value g_j associated with it—the observed color. The problem is to assign an integer value x_j to each node j so as to minimize the penalty function.

Let the K color shades be a set of ordered values $\mathcal{L} = \{q_1, q_2, \ldots, q_K\}$. Denote the assignment of a color q_p to pixel j by setting the variable $x_j = p$. Each pixel j is permitted to be assigned any color in a specified range $\{q_{\ell_j}, \ldots, q_{u_j}\}$.

[1] See, for example, Blake and Zisserman [1987], Boykov et al. [1988], Ishikawa and Geiger [1998], Geiger and Girosi [1991], and Geman and Geman [1984].

For $G()$ the *deviation cost* function and $F()$ the *separation cost* function the problem is,

$$\operatorname*{Min}_{u_i \geq x_i \geq \ell_i} \sum_{i \in V} G_i(g_i, x_i) + \sum_{i \in V} \sum_{j \in N(i)} F_{ij}\,(x_i - x_j).$$

This formulation is equivalent to the following constrained optimization problem, referred to as (IS) (standing for Image Segmentation):

$$\begin{array}{llr}
\text{(IS)}\quad \text{Min} & \displaystyle\sum_{j \in V} G_j(g_j, x_j) + \sum_{(i,\, j) \in A} F_{ij}\,(z_{ij}) & \\
\text{subject to} & x_i - x_j \leq z_{ij} & \text{for } (i, j) \in A \\
 & u_j \geq x_j \geq \ell_j & j = 1, \ldots, n \\
 & z_{ij} \geq 0 & (i, j) \in A.
\end{array}$$

Now the constraints of (IS) have several interesting properties. First, the coefficients of the constraints form a totally unimodular matrix. Second, the set of constraints are those of the linear programming dual of the minimum cost network flow. For the dual of minimum cost network flow problem, a generic constraint would be of the type

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

Third, the constraints of the problem are monotone, as defined in Hochbaum [1998a]. That is, each constraint is of the form $ax_i - bx_j \leq c_{ij} + z_{ij}$ for $a, b \geq 0$. For each of these classes of problems, there are known algorithms that are thus applicable to the (IS) problem.

The (IS) problem has several known special cases.

(1) *The Statistical Inference Problem.* The problem is to adjust observed values g_i to values x_i so that the adjusted values conform to rank requirements of the type $x_i \leq x_j$. The objective function is to minimize the deviation penalty of $|x_i - g_i|$. This is an (IS) with the functions $F()$ and the variables z_{ij} omitted. (Or, alternatively, when $F_{ij}\,(z_{ij}) = \infty$.) Such convex problems have been addressed by Hochbaum and Queyranne [2000], where the problem is called *the convex closure problem*.

(2) *The Minimum Cut Problem.* If the functions $F()$ are linear, the functions $G()$ are identically 0 and the variables x_j are binary, the problem (IS) is the minimum-cut problem. More precisely, by setting $x_s = 1$ and $x_t = 0$, a feasible solution to this (IS) problem is a partition of the set of nodes corresponding to the variables to $(S, \bar{S})$ with S containing s and $\bar{S}$ containing t and the sum of arc capacities of arcs in $(S, \bar{S})$ is minimum.

(3) *The Minimum s-Excess Problem.* For binary variables x_j, the problem becomes the so-called *s-excess* problem in Hochbaum [1998b]. This problem is defined on a capacitated graph with node weights. The goal is to find a subset of nodes S so that the total weight of the nodes in the set plus the capacity of the cut separating the set from the rest of the graph $C(S, \bar{S})$ is minimum. This problem plays a crucial role in the solution to the (IS) problem, as discussed in Section 3.1. The s-excess problem is equivalent to the (IS) problem with a set of colors restricted to two colors. This binary (IS) problem was first identified as equivalent to the minimum-cut problem by Greig et al. [1989].

(4) *The Minimum Closure Problem*. A closed set has the property that all successors of nodes in the set are included in the set. For a directed graph $G = (V, A)$ with node weights w_i, $i \in V$ the minimum closure problem is to find a closed set of total minimum weight. (Note that weights should be positive or negative; otherwise, the problem is trivial to solve.) For each arc (i, j), we have the constraint $x_i \leq x_j$. When node i is in the set, $x_i = 1$ and node j, its successor, is in the set as well. This is an (IS) problem with binary variables x_j and no variables z_{ij}. This problem was found by Picard [1976] to be solved by a minimum cut procedure on an associated graph.

(5) *Unconstrained Minimization of Convex Functions.* When the set A is empty and there are no constraints other than the box constraints, then the problem is to minimize each of the convex functions in integers on the given interval. That task cannot be performed in strongly polynomial time as explained in Section 2. More precisely, it is provably impossible for nonlinear and nonquadratic optimization problems over linear constraints to be solved in strongly polynomial time.

(6) *The Multiway Cut Problem.* The problem is defined for a graph with given edge weights and a set of K terminals specified among the vertices. The problem is to partition the graph to K subsets so that each terminal belongs to one of the subsets and so that the weight of the edges that have two endpoints in two different sets is minimum. This problem is special case of (IS) when $G_j() = 0$ and the functions F_{ij} are so-called δ-functions:

$$F_{ij}\,(x_i - x_j) = \begin{cases} u_{ij} & \text{if} \quad x_i \neq x_j \\ 0 & \text{if} \quad x_i = x_j. \end{cases}$$

This demonstrates that the (IS) problem with δ functions for $F_{ij}()$ is NP-hard as it is at least as hard as the multiway cut problem [Boykov et al. 1999b].

The main result here is an algorithm for $G()$ convex and for $F()$ linear, the complexity of which is the *sum* of the complexity of a minimum cut procedure *plus* the complexity of minimizing in integers the n convex functions G_j. Since the (IS) problem generalizes both the minimum-cut and the convex minimization problems, this complexity is the best that can be achieved.

1.1. RESULTS TO DATE. Most literature reviewed here provides algorithms for problems generalizing the (IS) problem.

The (IS) problem is a case of convex separable optimization over linear constraints. As such, it was first shown to be solved in polynomial time by Hochbaum and Shanthikumar [1990]. Among the results in Hochbaum and Shanthikumar [1990], it was proved that any continuous or integer convex separable minimization over linear constraints is solved in time polynomial in the number of variables and the constraints and the absolute value of the largest subdeterminant. For (IS) the constraint matrix is totally unimodular and thus the largest subdeterminant's absolute value is 1.

An inequality is said to be *monotone* if it is of the form $ax_i - bx_j \leq c_{ij} + z_{ij}$ with $a, b \geq 0$. Problem (IS) is a special case with $a = b = 1$. In Hochbaum [1998a], it was shown that any nonlinear separable problem with $G()$ arbitrary nonlinear functions and $F()$ convex functions, on monotone constraints is solved in integers in time that is polynomial in the number of x_j variables n, and the number of constraints (or variables z_{ij}) m, and in the range of the x_j

variables $U = \{\max_j u_j - \ell_j\}$. The optimization problem over monotone constraints is cast as a minimum cut problem on a graph with nU nodes and mU arcs. The complexity of that algorithm is, due to the presence of the factor U, *pseudopolynomial*.

Ishikawa and Geiger [1998] provide a pseudopolynomial algorithm for the (IS) problem. That algorithm casts the problem as a minimum cut on a graph of pseudopolynomial size.

Another problem generalizing the (IS) problem is the convex dual of the minimum cost network flow. Consider a minimum cost network flow problem defined on a network $G = (V, A)$ with flow variables denoted by y_{ij}. Let the sum of supplies be equal to the sum of demands of the nodes, $\sum_{i=1}^{n} w_i = 0$. Thus, we can write the problem's flow balance constraints as inequalities.

$$\begin{array}{ll} \text{Min} & \sum_{ij \in A} c_{ij} y_{ij} \\ \text{subject to} & \sum_j y_{ij} - \sum_j y_{ji} \geq w_i \quad i \in V \\ & \bar{u}_{ij} \geq y_{ij} \geq 0 \quad (i, j) \in A. \end{array}$$

The dual of this problem is

$$\begin{array}{lll} \text{Min} & \sum_{j \in V} w_j x_j + \sum_{(i,j) \in A} \bar{u}_{ij} z_{ij} & \\ \text{subject to} & x_i - x_j \leq c_{ij} + z_{ij} & \text{for } (i, j) \in A \\ & x_j \geq 0 & j = 1, \ldots, n \\ & z_{ij} \geq 0 & (i, j) \in A. \end{array}$$

Although there are no explicit upper bounds on the variables x_j, it is easy to see that for $C = \max_{(i,j) \in A} c_{ij}$, $x_j \leq nC$ for all $j \in V$. Thus, the problem is equivalent to a problem with bounded variables in a range of length $U = nC$. The constraints of (IS) are the constraints of the dual of minimum cost network flow problem with 0 costs, $c_{ij} = 0$.

Recently Ahuja et al. [1999a, 1999b] devised two different polynomial-time algorithms for the convex dual of the minimum-cost network flow. In Ahuja et al. [1999b], it was shown how to solve the convex dual of the minimum-cost network flow with an algorithm of complexity $O(mn \log n \log nU)$. The earlier paper Ahuja et al. [1999a] describes an algorithm that solves this problem by making $\log U$ calls to a minimum cut procedure on a graph on $O(n^2)$ nodes, where n is the number of variables x_i in the problem, and $O(nm)$ arcs, where m is the size of A or the number of variables z_{ij}.

The fact that the (IS) problem is solvable in pseudopolynomial time has been noted again recently by Chekuri et al. [2000]. They describe a new linear programming formulation of (IS) that has pseudopolynomial size, the solution to which is integral. The number of variables in their formulation is $O(n^2 U^2)$.

The instances of the (IS) problem with the function $F()$ nonconvex are NP-hard even for fixed value of $U \geq 3$ (since this problem generalizes the multiway cut problem on U terminals.) These intractable problems have been a recent subject of a great deal of work in approximation algorithms.[2] All these and other

[2] See, for example, Boykov et al. [1988; 1999a; 1999b], Kleinberg and Tardos [1999], Gupta and Tardos [2000], and Chekuri et al. [2000].

approximation algorithms known for the problem run in pseudopolynomial time and depend on U (or K, in the context of the number of colors).

1.2. OUR RESULTS. Let the (IS) problem involve n pixels (variables) and m adjacency relations (arcs). Let $T(n, m)$ be the complexity of solving the minimum s, t cut problem on a graph with n nodes and m arcs. Our main result is an algorithm that solves the problem for $G()$ convex functions and $F()$ linear functions in time $O(T(n, m) + n \log U)$. This complexity expression is composed of the time required to solve a minimum cut problem plus the time required to find the minima of n convex functions. Since the (IS) problem generalizes both these problems this time complexity is the best time complexity achievable. Any improvement in the run time of algorithms to identify the integer minima of convex functions or to find a minimum (parametric) cut would immediately translate into improvements of the run time of our algorithm.

A summary of our results for solving the (IS) problem include algorithms, depending on the functional form of the functions F and G.

(1) For $G()$, $F()$ convex functions, the problem is solved in polynomial time. An algorithm that runs in $\log U$ calls to a minimum cut procedure with complexity $O(\log U \cdot T(n^2, mn))$ is reported in Ahuja et al. [1999a]. Another, more efficient, algorithm for this problem runs in $O(mn \log n \log nU)$ [Ahuja 1999b]. Both these algorithms have been devised for the more general problem of the convex dual of minimum cost network flow.

(2) For $G()$ convex functions and $F()$ linear deviation functions, the algorithm reported here has the complexity of the minimum cut problem, $T(n, m)$, plus the complexity of finding the integer minima in intervals of up to n convex functions of the form of G_j. The linear deviation functions are of the form:

$$F_{ij}(x_i - x_j) = \begin{cases} u_{ij} & \text{if} \quad x_i > x_j \\ 0 & \text{if} \quad x_i = x_j \\ u_{ji} & \text{if} \quad x_i < x_j. \end{cases}$$

The complexity of this problem is $O(T(n, m) + n \log U)$.

(3) For $G()$ arbitrary *nonlinear* functions and $F()$ convex functions the algorithm of Hochbaum [1998a] and Ahuja et al. [1999a] runs in pseudopolynomial time required to find a minimum cut on a graph with nU nodes and mU edges, $T(nU, mU)$.

Our results imply that a specific type of the multiway cut problem—where the weight of each edge in the cut is a linear function of the absolute distance between its endpoints' subset terminal values—is polynomial time solvable.

2. *A Complexity Model for Convex Functions*

The challenge of convex optimization for nonlinear and nonquadratic problems is that searching for a minimum of a convex function involves an unavoidable factor such as $\log U$ in the running time, for $U = \max_i\{u_i - \ell_i\}$ [Hochbaum 1994]. Although one can replace the length of the interval by other parameters that depend on the variability of the functions, the running time cannot be made strongly polynomial using the arithmetic complexity model (see Hochbaum [1994] for details). The algorithm presented here differs from previous algorithms in that the search for the minima of the convex functions is disjoint from the rest of the algorithm and conducted as a post-processing step. Thus, if the functions have a special structure that allows to find the minima more efficiently, this structure can be used to improve the complexity of the algorithm. For instance, for the minima of quadratic functions can be found in $O(1)$. Also, for piecewise linear functions with few pieces (such as absolute value deviation), the minima can be found in strongly polynomial time. “Few” in this context means that the number of pieces or breakpoints is N where $\log N$ is a polynomial function of n and m, $p(n, m)$. So “few” can be for instance $O(2^{p(n,m)})$.

The structure of the convex functions here is not restricted, so it is necessary to define the complexity model used. In dealing with convex functions, we use the unit cost complexity model: This assumes the existence of an oracle returning function values for every argument input in $O(1)$. We will only be interested with arguments that are integers, or lie on a grid of ϵ granularity when we consider the continuous problem. Any arithmetic operation or comparison involving functions values is assumed to executed in unit time. We further assume that the sum of two convex functions can be determined in $O(1)$. Derivatives, or rather subgradients, are also required. For the problems considered we let the subgradient of the function $f()$ be,

$$f'_j(x) = f_j(x+1) - f_j(x).$$

The negative result in Hochbaum [1994] is not applicable to the quadratic case. Thus, it may be possible to solve constrained quadratic optimization problems in strongly polynomial time. Yet, very few quadratic optimization problems are known to be solvable in strongly polynomial time. For instance, it is not known how to solve the minimum quadratic cost network flow problem in strongly polynomial time. For the (IS) problem with the functions $G()$ quadratic convex and the functions $F()$ linear, our result adds to the limited repertoire of quadratic problems solved in strongly polynomial time.

3. *A Reduction to the Binary Case—The Threshold Theorem*

The main idea of the algorithm is to reduce the problem to a number of calls to the binary version of the problem (IS). We prove next a key theorem that demonstrates that for a given threshold value α, and a suitably selected binary version of the problem, the solution to the binary problem separates the x_j variables into two sets: One set of variables that have value above α in an optimal solution, and the second that has the variables values below α in an optimal solution.

3.1. THE S-EXCESS PROBLEM. As noted in the introduction the s-excess problem is the binary version of the (IS) problem. The formal statement of the problem is

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 u_{ij} for all $(i, j) \in A$.*

Optimization Problem: *Find a subset of nodes $S \subseteq V$ such that $\sum_{i\in S} w_i + \sum_{i\in S,\, j\in \bar{S}} u_{ij}$ is minimum.*

The s-excess problem is formulated as the binary optimization problem

$$
\begin{array}{llll}
\text{(s-excess)} & \text{Min} & \sum_{j \in V} w_j x_j + \sum u_{ij} z_{ij} & \\
\text{subject to} & & x_i - x_j \leq z_{ij} & \text{for } (i, j) \in A \\
& & 1 \geq x_j \geq 0 & \text{integer } j = 1, \ldots, n \\
& & 1 \geq z_{ij} \geq 0 & \text{integer } (i, j) \in A.
\end{array}
$$

The following lemma was proved in Greig et al. [1989] and in Hochbaum [1998b].

LEMMA 3.1. *Solving the minimum s-excess problem is equivalent to solving the minimum-cut problem on an appropriately defined graph.*

PROOF. Let the s-excess problem be defined on a graph $G = (V, A)$. Define a graph $G_{st} = (V \cup \{s, t\}, A_{st})$: The set of nodes of the graph is the set V appended by two nodes s and t. There is an arc between each node of negative weight j and the source carrying the capacity $-w_j$. There is an arc between each node of positive weight i and the sink carrying the capacity w_i.

We claim that S is the source set of a minimum cut in G_{st} if and only if it is a set of minimum s-excess capacity in the graph G. Let $C(A, \bar{A})$ be the sum of capacities of arcs with tails in A and heads in $\bar{A}$.

Noting that the capacities of arcs adjacent to source are the negative of the respective node weights, we have that the s-excess weight of a set S is the sum of capacities: $-C(\{s\}, S) + C(S, \{t\})$. We rewrite the objective function in the minimum s-excess problem:

$$
\begin{aligned}
\min_{S \subseteq V} [-C(\{s\}, S) \; + \; C(S, \bar{S} \cup \{t\})] \\
&= \min_{S \subseteq V} -[C(\{s\}, V) - C(\{s\}, \bar{S})] + C(S, \bar{S} \cup \{t\}) \\
&= -C(\{s\}, V) + \min_{S \subseteq V} [C(\{s\}, \bar{S}) + C(S, \bar{S} \cup \{t\})].
\end{aligned}
$$

In the last expression, the term $-C(\{s\}, V)$ is a constant. The expression minimized is precisely the sum of capacities of arc in the cut $(S \cup \{s\}, \bar{S} \cup \{t\})$. Thus, the set S minimizing the s-excess is also the source set of a minimum cut and, vice versa—the source set of a minimum cut also minimizes the s-excess. □

The Convex s-excess problem is a generalization of the s-excess problem with node weights $f_j()$ that are convex functions.

$$
\begin{array}{llll}
\text{(Convex s-excess)} \; P(\mathbf{x}) = P(\mathbf{x}, \mathbf{z}) = & \text{Min} & \sum_{j \in V} f_j(x_j) + \sum e_{ij} z_{ij} & \\
\text{subject to} & & x_i - x_j \leq z_{ij} & \text{for } (i, j) \in A \\
& & u_j \geq x_j \geq \ell_j & j = 1, \ldots, n \\
& & z_{ij} \geq 0 & (i, j) \in A.
\end{array}
$$

Obviously, the Convex s-excess problem is identical to the (IS) problem. Note that the arc weights e_{ij} are of interest only when positive. Otherwise, the problem is unbounded.

A feasible solution to the problem is specified in terms of the values of the vector $\mathbf{x}$ and $\mathbf{z}$. It is however sufficient to specify the values of $\mathbf{x}$ alone, since the values of $\mathbf{z}$

are uniquely determined by $\mathbf{x}$. That is, in an optimal solution $z_{ij} = \max\{x_i - x_j, 0\}$. It is therefore sufficient to only specify the values of $\mathbf{x}$ in an optimal solution.

3.2. THE THRESHOLD THEOREM. We define the s-excess problem on a graph G_α for α an integer scalar. Let the weights of nodes in G_α be $w_i = f_i'(\alpha)$ where $f_i'(\alpha)$ are the subgradients of $f_i()$ at α, $f_i(\alpha) - f_i(\alpha - 1)$. Let the subgradient value of function $f_i(x)$ to be equal to M at values of $x > u_i$, and to $-M$ for values $x < \ell_i$, for M a suitably large value. With this extension, the box constraints are uniform for all variables, $u \geq x_j \geq \ell$ and we choose $\alpha \in (\ell, u)$. Let the arc weights be $u_{ij} = e_{ij}$.

In a graph with arc capacities u_{ij} and two disjoint sets of nodes $A, B \subset V$, we use the notation $C(A, B)$ to denote the total capacity of arcs from the A to B, $\sum_{i \in A, j \in B} u_{ij}$.

In case there are several optimal s-excess sets, we term a *minimal* minimum s-excess set to be a minimum s-excess set that does not contain other minimum s-excess sets. Similarly, we term a *maximal* minimum s-excess set as a minimum s-excess set that is not contained in another minimum s-excess set.

The threshold theorem establishes that all elements i in the minimum s-excess set S^* in G_α satisfy that in an optimal solution to the convex s-excess problem, $\mathbf{x}^*$, $x_i^* \leq \alpha$, and all elements j in the complement of S^* satisfy that $x_j^* > \alpha$. The theorem is an extension of the threshold theorem of Hochbaum and Queyranne [2000] used in the context of *convex closure problems.*

THEOREM 3.1. *Let S^* be the maximal minimum s-excess set in the graph G_α. Then there is an optimal solution $\mathbf{x}^*$ to the corresponding convex s-excess problem satisfying, $x_i^* \geq \alpha$ if $i \in S^*$ and $x_i^* < \alpha$ if $i \in \bar{S}^*$.*

PROOF. Let S^* be a maximal minimum s-excess set, and suppose it violates the theorem. Then for every optimal solution $\mathbf{x}^*$ there is a subset $S^\circ \subseteq S^*$ such that $x_j^* < \alpha$ for all $j \in S^\circ$, or there is a subset $S^1 \subseteq \bar{S}^*$ such that $x_j^* \geq \alpha$ for all $j \in S^1$.

Suppose there exists a subset S° then select among all optimal solutions for which the violation in S^* is on a subset of S° the one for which $\sum_{j \in S^\circ} x_j^*$ is maximum. Since S^* is a minimum s-excess set, the contribution of adding the set S° to the set $S^* \setminus S^\circ$ must be nonpositive

$$\Delta^\circ = \sum_{j \in S^\circ} w_j + C(S^\circ, \bar{S}^*) - C(S^* \backslash S^\circ, S^\circ) \leq 0.$$

Now, for $\epsilon > 0$, consider a solution $\mathbf{x}'$ defined as follows:

$$x_i' = \begin{cases} x_i^* & \text{if} \quad i \notin S^\circ \\ x_i^* + \epsilon & \text{if} \quad i \in S^\circ. \end{cases}$$

Then, $P(\mathbf{x}') \leq P(\mathbf{x}^*) + \epsilon \Delta^\circ \leq P(\mathbf{x}^*)$. (Note that the first inequality may be strict since the term $C(S^\circ, \bar{S}^*)$ may not be fully accounted for some arcs in $(S^\circ, \bar{S}^*)$ that have $x_i^* < x_j^*$, and then the corresponding value of $z_{ij} = 0$ does not change for $\mathbf{x}'$ for ϵ small enough. That proves that this situation is impossible since a strict inequality contradicts optimality.) Therefore, the solution $\mathbf{x}'$ is optimal, the violation in S^* is only in a subset of S^*, and $\sum_{j \in S^\circ} x_j' > \sum_{j \in S^\circ} x_j^*$, which is a contradiction to the maximality of $\mathbf{x}^*$.

We conclude that in every optimal solution $\mathbf{x}^*$, $x_i^* \geq \alpha$ for $i \in S^*$.

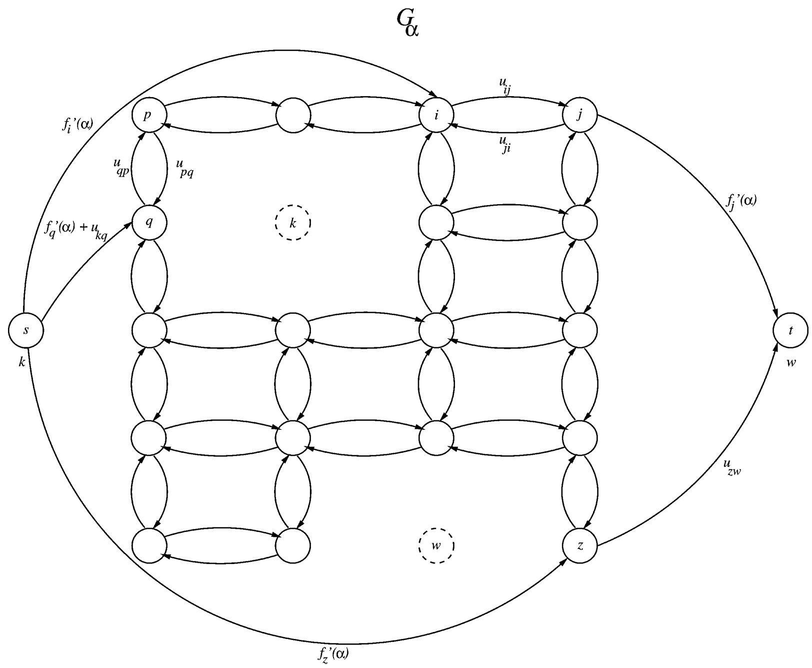

FIG. 1. The graph G_α.

Now suppose there exists a set $S^1 \subseteq \bar{S^*}$ as above in an optimal solution $\mathbf{x}^*$. Since S^1 is not in a maximal minimum s-excess set its contribution to the objective must be positive

$$\Delta^1 = \sum_{j \in S^1} w_j + C(S^1, \bar{S^*} \backslash S^1) - C(S^*, S^1) > 0.$$

Now, for $\delta = \min_{j \in S^1}(x_j^* - \alpha) + \epsilon > 0$, let a solution $\mathbf{x}''$ be defined as follows:

$$x_i'' = \begin{cases} x_i^* & \text{if} \quad i \notin S^1 \\ x_i^* - \delta & \text{if} \quad i \in S^1. \end{cases}$$

In the solution $\mathbf{x}''$ all arcs $(i, j) \in (S^1, \bar{S^*} \backslash S^1)$ have $x_i' > x_j'$ and thus the corresponding values of z_{ij} are positive for all these arcs. Therefore, the term $\sum_{(i,j) \in (S^1, \bar{S^*} \backslash S^1)} u_{ij} z_{ij}$ is reduced by $\delta C(S^1, \bar{S^*} \backslash S^1)$. The term $\sum_{(i,j) \in (S^*, S^1)} u_{ij} z_{ij}$ may go up, but by no more than $\delta C(S^*, S^1)$.

Thus, $P(\mathbf{x}'') \leq P(\mathbf{x}^*) - \delta \Delta^1 < P(\mathbf{x}^*)$, which contradicts the optimality of $\mathbf{x}^*$. $\square$

3.3. AN ILLUSTRATED EXAMPLE. Consider a grid of pixels in the plane with pixels considered neighbors if they are adjacent horizontally or vertically. Thus, each pixel has up to four neighbors. Consider the graph G_α where node k having been shrunk with the source and node w shrunk with the sink. The process of shrinking subsets of nodes with source or sink is part of the parametric algorithm that applies to the graph G_α. The arcs between nodes that have been shrunk and other nodes in the graph appear as arcs between source or sink and the respective nodes. In the example illustrated in the graph in Figure 1, let $f_z'(\alpha)$, $f_i'(\alpha)$, $f_q'(\alpha) < 0$

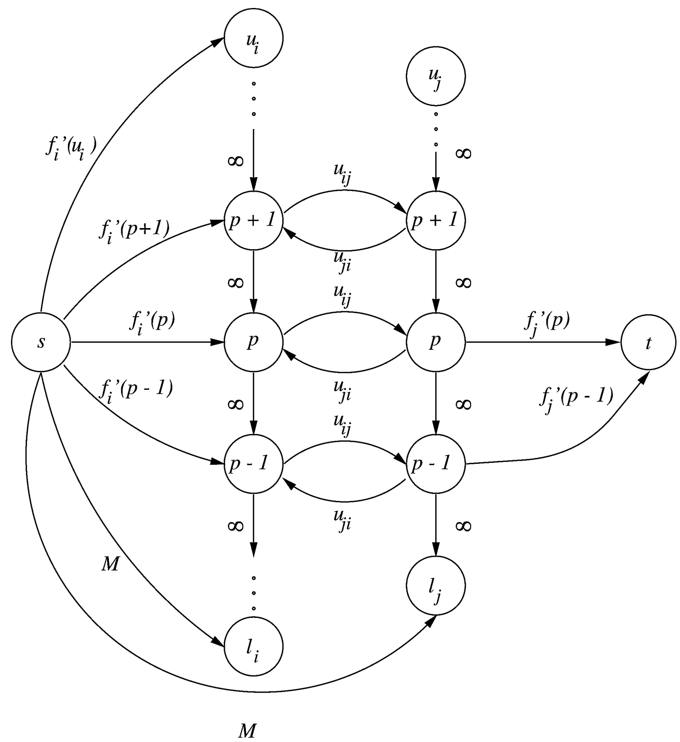

FIG. 2. A schematic description of the part of the network on $O(nU)$ nodes and $O(mU)$.

and $f_j'(\alpha) > 0$. The algorithm we present has the complexity of solving for the minimum cut on the graph in Figure 1.

In case the functions $F()$ are not linear, a larger network is required. For each pair of adjacent nodes, the network contains construction as given in Figure 2. The network contains $u_i - \ell_i + u_j - \ell_j$ nodes and a similar number of arcs for each pair of adjacent pixels i and j. For simplicity of illustration in this graph, we let $f_i'(k) < 0$ for all $\ell_i \le k \le u_i$ and $f_j'(k) > 0$ for all $\ell_j \le k \le u_j$.

4. *The Algorithm*

One obvious method of using the threshold theorem for solving the convex s-excess (or (IS)) problem is to perform a search by calling for the solution of the s-excess problem for all integer values of α in the interval (ℓ, u). When done, the output of such process is a partition of the set of variables V into q sets, and the interval into q disjoint intervals, so that all variables in the same set have their optimal value lie in the same interval. The goal would be to find, for each variable x_j, the largest of value of α for which it is still in the source set and the smallest value of α for which it is no longer in the source set. With this information, we narrow down the value of x_j at an optimal solution to an interval defined by these values. We later show that once these intervals are identified, all variables assigned to the same interval assume the same value in that interval, and that value can be determined in polynomial time. One drawback of the approach just described is that it makes U calls to a minimum-cut procedure, and is thus pseudopolynomial.

It is easy to see that a binary-search-type approach could be used to implement the procedure of identifying the intervals to a polynomial time procedure. Next, we show that one can do better still by implementing the process of identifying the set and interval partitioning in strongly polynomial time and in the complexity of solving a single minimum-cut problem.

4.1. THE PARAMETRIC GRAPH G_λ. We create a graph with parametric capacities $G_\lambda = (V, A)$. Each node $j \in V$ has an arc from s going into the node with capacity $-\min\{0, f_j'(\lambda)\}$, and an arc from j to the sink t with capacity $\max\{0, f_j'(\lambda)\}$. The capacities of the arcs adjacent to source in this graph are monotone nonincreasing as a function of λ, and the arcs adjacent to sink are all with capacities that are monotone nondecreasing as a function of λ. Note that each node is connected with a positive capacity arc, either to source or to sink, but not to both. Denote the source set of a minimum cut in the graph G_λ by S_λ. Let all arcs (i, j) that are neither adjacent to source nor to sink carry the capacity u_{ij}.

Let ℓ be the lowest lower bound on any of the x_j variables and u the largest upper bound. Consider varying the value of λ in the interval $[\ell, u]$. Since for all variables $j \in V$ and the value ℓ, $f_j'(\ell) = -M$, $S_\ell = V$, and at u, $f_j'(u) = M$ and $S_u = \emptyset$. As the value of λ increases, the sink set becomes larger and contains the previous sink sets corresponding to smaller values of λ. The source set is becoming smaller with increasing values of λ and each contains the source sets that are generated for larger values of λ. Our goal is to identify for each variable the largest value of λ, $\underline{\lambda}$ so that the variable is still in the minimum s-excess set, and the smallest value of λ, $\bar{\lambda}$, so the variable is not in the s-excess set. Then, the optimal value of this variable is in $[\underline{\lambda}, \bar{\lambda})$.

For larger values of λ, additional nodes join the sink set of the cut. As λ grows, the value of the cut changes as a function of λ that is the sum of the capacities of the nodes adjacent to source that are in the sink set, and nodes adjacent to sink that are in the source set. That function has a breakpoint, called *node shifting breakpoint*, where a node switches from the source set to the sink set. The value of $\underline{\lambda}$ for node x_j is the value of the breakpoint when node j joins the source set of the minimum cut. The value of $\bar{\lambda}$ is the next node shifting breakpoint.

Since the source set of a minimum cut can only shrink with increasing values of λ, then

$$S_k \supseteq S_{k+1} \qquad \text{for all} \quad k.$$

Let the output of the parametric search process be a set of up to n node-shifting breakpoints, $\ell = \lambda_1 < \lambda_2 < \cdots < \lambda_q < u$. Then,

$$V = S_{\lambda_1} \supset S_{\lambda_2} \supset \cdots \supset S_{\lambda_q} \supset \emptyset = S_{\lambda_q + 1}.$$

The total number of breakpoints cannot exceed n, $q < n$, since each is associated with at least one node shifting from source set to sink set. Let $S(i) = S_{\lambda_i} - S_{\lambda_{i+1}}$, for $i = 1, \ldots, q$.

LEMMA 4.1. *For $j \in S(k)$, the value of x_j at an optimal solution, x_j^*, is the argument minimizing,*

$$\min_{x\in[\lambda_k,\lambda_{k+1})} \sum_{i\in S(k)} f_i(x) + x \cdot \sum_{p=k+1}^{q} C(S(k), S(p)) - x \cdot \sum_{p<k} C(S(p), S(k)).$$

PROOF. First, we show that all variables in each set $S(k)$ assume the same value in an optimal solution. Next, we will demonstrate that this value is the one minimizing the function given in the lemma statement.

According to the threshold theorem, λ_k is the largest value so that for $j \in S(k)$, $x_j \geq \lambda_k$, and λ_{k+1} is the smallest value so that $x_j < \lambda_{k+1}$. Thus, all variables x_i, with $i \in S(k)$, have their optimal value lie in the interval $[\lambda_k, \lambda_{k+1})$. Suppose two variables in $S(k)$ assume different optimal values in the interval $x^*_{i_1} < x^*_{i_2}$. Then there exists a value $\lambda \in [x^*_{i_1}, x^*_{i_2}]$ so that the source set S_λ contains the node i_2 but not i_1. But there is no breakpoint in $[\lambda_k, \lambda_{k+1})$, which is a contradiction.

Let $\alpha_k \in [\lambda_k, \lambda_{k+1})$ be the common value of the variables in $S(k)$, for $k = 1, \ldots, q$. Then $\alpha_1 < \alpha_2 < \cdots < \alpha_q$. Now the objective value of the convex s-excess problem is,

$$\min \sum_{k=1}^{q} \sum_{i\in S(k)} f_i(\alpha_k) + \sum_{k=2}^{q} \sum_{p=1}^{k-1} \sum_{i\in S(k),\, j\in S(p)} e_{ij}(\alpha_k - \alpha_p) = \min \sum_{k=1}^{q} L_k(\alpha_k).$$

This expression is separable into functions of α_k where,

$$L_k(\alpha_k) = \sum_{i\in S(k)} f_i(\alpha_k) + \alpha_k \cdot \sum_{p=1}^{k-1} C(S(k), S(p)) - \alpha_k \cdot \sum_{p=k}^{q} C(S(p), S(k)). \qquad \square$$

4.2. IDENTIFYING AN INTEGER NODE-SHIFTING BREAKPOINT. Let the maximal minimum cut source set in G_λ be denoted by $S^{\max}_\lambda$ and the minimal minimum cut source set, by $S^{\min}_\lambda$.

The source set of a minimum cut of G_λ remains invariant for λ varying in the range $[k-1, k)$ for k integer although the value of the cut may change. This is since the objective function can be assumed to be piecewise linear convex with integer breakpoints. Thus, in order to verify that λ is a node-shifting breakpoint, it suffices to compare $S^{\min}_\lambda$ with $S^{\max}_{\lambda+\epsilon}$ for $\epsilon > 0$ small enough. In our case we only consider integer values of λ, and $\epsilon = 1$ is a small enough value: If $S^{\min}_\lambda \supset S^{\max}_{\lambda+1}$, then λ is a node-shifting breakpoint.

The existence of a breakpoint in an interval (λ_1, λ_2) is confirmed if and only if $S^{\min}_{\lambda_1} \supset S^{\max}_{\lambda_2}$.

4.2.1. *Parametric Analysis.* We assume henceforth that our minimum cut algorithm delivers as output both $S^{\min}_\lambda$ and $S^{\max}_\lambda$. We further assume that the procedure **min-cut** (G_λ) returns both the minimal and maximal source sets of minimum cuts (if different), $S^{\min}_\lambda$, $S^{\max}_\lambda$, and $S^{\max}_{\lambda+1}$.

For a given interval (λ_1, λ_2), where arc capacities do not contain a breakpoint b_j where a node switches from being connected to source to being connected to sink, we can find all node-shifting breakpoints by using the procedure **parametric**.

The procedure uses the operation "contract" of a set of nodes with the source or with the sink. All arcs that were adjacent to the contracted set become, after the contract operation, adjacent to the node—source or sink—that the set was contracted with.

Procedure parametric $(\lambda_1, \lambda_2, S^{\min}_{\lambda_1}, S^{\max}_{\lambda_2})$
Contract: $s \leftarrow s \cup S^{\max}_{\lambda_2}$, $t \leftarrow t \cup S^{\min}_{\lambda_1}$.
  If $V = \{s, t\}$, or, if $\lambda_2 - \lambda_1 \leq 1$, halt "no breakpoints".
Else, let $\lambda^* = \lfloor(\lambda_1 + \lambda_2)/2\rfloor$.
Call **min-cut**(G_{λ^*}) for the output $S^{\min}_{\lambda^*}$, $S^{\max}_{\lambda^*}$
If λ^* is a breakpoint, output λ^*. Else continue,
  Call **parametric** $(\lambda_1, \lambda^*, S^{\min}_{\lambda_1}, S^{\max}_{\lambda^*})$
  Call **parametric** $(\lambda^*, \lambda_2, S^{\min}_{\lambda^*}, S^{\max}_{\lambda_2})$
end

The choice of λ^* as the median in the interval (λ_1, λ_2) is unimportant. Any integer value interior to the interval will work correctly. Verifying whether λ^* is a breakpoint is equivalent to one of the conditions being satisfied:

(i) $S^{\min}_{\lambda_1} \supset S^{\max}_{\lambda^*}$ and $\lambda^* - \lambda_1 \leq 1$, or
(ii) $S^{\min}_{\lambda^*} \supset S^{\max}_{\lambda_2}$ and $\lambda_2 - \lambda^* \leq 1$.

The analysis of the complexity of the procedure follows arguments used in Gallo et al. [1989] for the push-relabel algorithm. This analysis applies also for the pseudoflow algorithm in all its variants [Hochbaum 1998b]: For a given interval, where we search for breakpoints, we run the algorithm twice. Once from the lower endpoint of the interval where the maximal source set of the cut obtained at that value shrunk into the source, and a second time from the highest endpoint of the interval where the maximal sink set of the cut is shrunk into the sink. The runs proceed for the graph and reverse graph till the first one is done. The newly found cut subdivides the graph into source set and sink set of sizes (n_1, m_1) and (n_2, m_2) for the number of nodes and arcs in each. One of these induced subgraphs is smaller in terms of the number of nodes, say $n_1 \leq (1/2)n$. In that smaller subgraph, two new runs are initiated from both endpoints. In the larger interval, however, we *continue* the previous runs using two properties:

—*Reflectivity*. The complexity of the algorithm remains the same whether running it on the graph or reverse graph.
—*Monotonicity*. Running the algorithm on a monotone sequence of parameter values has the same complexity as a single run.

Under these properties, one run is "reflected" to the opposite endpoint (thus viewed as monotone continuation), and the other run continues as a monotone continuation). Let $m_1 + m_2 \leq m$, $n_1 + n_2 \leq n$ and $n_1 \leq (1/2)n$. The running time $\mathcal{T}(m, n)$ is the running time required by the algorithm to solve the problem on a graph with m arcs and n nodes. Let Q be a constant, then using the push-relabel algorithm for **min-cut** (G_{λ^*}) the running time function satisfies the recursive equation,

$$\mathcal{T}(m, n) = \mathcal{T}(m_1, n_1) + \mathcal{T}(m_2, n_2) + 2Qm_1 n_1 \log \frac{n_1^2}{m_1}.$$

The solution to the recursive equation with the minimum-cut algorithm implemented as push-relabel algorithm is $\mathcal{T}(m, n) = O(mn \log n^2/m)$.

4.3. THE FORMAL DESCRIPTION OF THE ALGORITHM. Let ℓ be the lowest lower bound on any of the variables x_j and u the largest upper bound. Let $U = u - \ell$.

procedure minimum s-excess $(G, f_j, j = 1, \ldots, n)$

Step (1). Call **parametric** $(\ell, u, \emptyset, V)$.
Let the output be a set of up to n breakpoints $\lambda_1 < \lambda_2 < \cdots < \lambda_q$.
Let $S(k) = S_{\lambda_k} \setminus S_{\lambda_k+1}$.
Step (2). For $k = 1, \ldots, q$ find the integer minimum of the convex function,

$$\min_{x \in [\lambda_k, \lambda_{k+1})} L_k(x) = \min_{x \in [\lambda_k, \lambda_{k+1})} \sum_{i \in S(k)} f_i(x) + x \cdot \sum_{p=k+1}^{q} C(S(k), S(p)) - x \cdot \sum_{p<k} C(S(p), S(k)).$$

Let the argument of the minimum be α_k, for $k = 1, \ldots, q$.
Step (3). Output the optimal solution $\mathbf{x}^*$ where, $x_j^* = \alpha_k$ for $j \in S(k)$.

In Step (2), identifying the minimum argument α_k of a convex function, amounts to searching, in the sorted array of derivative values $L_k'(x) = \sum_{i \in S(k)} f_i'(x) + \sum_{p=k+1}^{q} C(S(k), S(p)) - \sum_{p<k} C(S(p), S(k))$, for a last integer value q' so that $L_k(q')$ is negative. Then, compare $L_k(q')$ and $L_k(q'+1)$ and the lower is the desired minimum. This can be accomplished using binary search in $O(\log(\lambda_{k+1} - \lambda_k))$ time per function and no more than $O(n \log U)$ total complexity.

4.4. THE COMPLEXITY OF THE ALGORITHM. The parametric search generating all breakpoints in Step (1) is implemented in time $T(n, m)$, which is, for example, $O(mn \log n^2/m)$. The second step of the algorithm requires finding all the integer minima of the convex functions for a total complexity of $O(n \log U)$. Step (3) is accomplished in $O(n)$ time. The total run time of the algorithm is thus,

$$O\left(mn \log \frac{n^2}{m} + n \log U\right).$$

5. *Conclusions*

We present here a particularly efficient algorithm for image segmentation problem with convex deviation cost and linear separation cost. It is further shown that the problem can be solved in polynomial time, but not as efficiently, when the separation cost is convex as well, and in pseudopolynomial time, $T(nU, mU)$ when the deviation cost is nonconvex and the separation cost is convex. When the separation cost is noncovex the problem is NP-hard even for fixed U. It would be of interest to investigate whether NP-hard instances of the problem, with nonlinear separation costs, can be addressed by relaxing the problem to the polynomially solvable version solved here. Specifically, there may be a potential for using the formulation here in achieving alternative approximation results for these problems.

REFERENCES

AHUJA, R. K., HOCHBAUM, D. S., AND ORLIN, J. B. 1999a. A cut-based algorithm for the convex dual of the minimum cost network flow problem. UC Berkeley manuscript, Oct.

AHUJA, R. K., HOCHBAUM, D. S., AND ORLIN, J. B. 1999b. Solving the convex cost integer dual network flow problem. In *Proceedings of Symposium on Integer Programming and Combinatorial Optimization*

(*IPCO99*), G. Cornuejols, R.E. Burkard and G.J. Woeginger, Eds. Lecture Notes in Computer Science, vol. 1610. Springer-Verlag, New York, pp. 31–34.

AHUJA, R. K., MAGNANTI, T. L., AND ORLIN, J. B. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, N.J.

BARLOW, R. E., BARTHOLOMEW, D. J., BREMER, J. M., AND BRUNK, H. D. 1972. *Statistical Inference Under Order Restrictions*. Wiley, New York.

BLAKE, A., AND ZISSERMAN, A. 1987. *Visual Reconstruction*. MIT Press, Cambridge, Mass.

BOYKOV, Y., VEKSLE, O., AND ZABIH, R. 1998. Markov random fields with efficient approximations. In *Proceedings of 6th IEEE International Conference on Computer Vision and Pattern Recognition* (Santa Barbara, Calif.). IEEE Computer Society Press, Los Alamitos, Calif., pp. 648–655.

BOYKOV, Y., VEKSLE, O., AND ZABIH, R. 1999a. Fast approximate energy minimization via graph cuts. In *Proceedings of the 7th IEEE International Conference on Computer Vision*. IEEE Computer Society Press, Los Alamitos, Calif., pp. 377–384.

BOYKOV, Y., VEKSLE, O., AND ZABIH, R. 1999b. A new algorithm for energy minimization with discontinuities. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 205–220.

CHEKURI, C., KHANNA, S., NAOR, J., AND ZOSIN, L. 2000. Approximation algorithms for the metric labeling problem via a new linear programming formulation. Manuscript, July.

GALLO, G., GRIGORIADIS, M. D., AND TARJAN, R. E. 1989. A fast parametric maximum flow algorithm and applications. *SIAM. J. Comput. 18*, 1, 30–55.

GEIGER, D., AND GIROSI, F. 1991. Parallel and deterministic algorithms for MRFs: Surface reconstruction. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI-13*, 401–412.

GEMAN, S., AND GEMAN, D. 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI-6*, 721–741.

GOLDBERG, A. V., AND TARJAN, R. E. 1988. A new approach to the maximum-flow problem. *JACM 35*, 4 (Oct.), 921–940.

GREIG, D. M., PORTEOUS, B. T., AND SEHEULT, A. H. 1989. Exact maximum a posteriori estimation for binary images. *J. Roy. Statist. Soc., Seri. B 51*, 2, 271–279.

GUPTA, A., AND TARDOS, E. 2000. A constant factor approximation algorithms for a class of classification problems. In *Proceedings of the 32nd Annual Symposium on Theory of Computing* (Portland, Ore., May 21–23). ACM, New York, pp. 652–658.

HOCHBAUM, D. S. 1994. Lower and upper bounds for allocation problems. *Math. Oper. Res. 19*, 2, 390–409.

HOCHBAUM, D. S. 1998a. Extended abstract: Instant recognition of half integrality and 2-approximations. In *Proceedings of the Approximations Algorithms* (*APPROX 98*). Lecture Notes in Computer Science, vol. 1444. Springer-Verlag, New York, pp. 99–110.

HOCHBAUM, D. S. 1998b. Extended abstract: The pseudoflow algorithm and the pseudoflow-based simplex for the maximum flow problem. In *Proceedings of 6th International Conference on Integer Programming and Combinatorial Optimization*. Lecture Notes in Computer Science, vol. 1412. Springer-Verlag, New York, pp. 325–337.

HOCHBAUM, D. S., AND QUEYRANNE, M. 2000. Extended abstract: Minimizing a convex cost closure set. In *Proceedings of the 8th Annual European Symposium Algorithms* (*ESA 2000*) (Saarbrucken, Germany). Lecture Notes in Computer Science, vol. 1879. Springer-Verlag, New York, pp. 256–267. Full version to appear: *SIAM J. of Discrete Math.*

HOCHBAUM, D. S., AND SHANTHIKUMAR, J. G. 1990. Convex separable optimization is not much harder than linear optimization. *JACM 37*, 4, 843–862.

ISHIKAWA, H., AND GEIGER, D. 1998. Segmentation by grouping junctions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR98*). IEEE Computer Society Press, Los Alamitos, Calif., pp. 125–131.

KLEINBERG, J., AND TARDOS, E. 1999. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *Proceedings of the IEEE Symposium on Foundations of Computer Science.* IEEE Computer Society Press, Los Alamitos, Calif., pp. 14–23.

PICARD, J. C. 1976. Maximal closure of a graph and applications to combinatorial problems. *Manage. Sci. 22*, 1268–1272.

RECEIVED SEPTEMBER 2000; REVISED JANUARY 2001; ACCEPTED JANUARY 2001