New Algorithms for the Dual of the Convex Cost Network Flow Problem with Application to Computer Vision

Vladimir Kolmogorov
University College London, UK
vnk@adastral.ucl.ac.uk

Akiyoshi Shioura
Tohoku University, Japan
shioura@dais.is.tohoku.ac.jp

Technical Report
June 4, 2007

Abstract

Motivated by various applications to computer vision, we consider an integer convex optimization problem which is the dual of the convex cost network flow problem. In this paper, we first propose a new primal algorithm for computing an optimal solution of the problem. Our primal algorithm iteratively updates primal variables by solving associated minimum cut problems. The main contribution in this paper is to provide a tight bound for the number of the iterations. We show that the time complexity of the primal algorithm is $K \cdot T(n, m)$ where K is the range of primal variables and $T(n, m)$ is the time needed to compute a minimum cut in a graph with n nodes and m edges.

We then propose a primal-dual algorithm for the dual of the convex cost network flow problem. The primal-dual algorithm can be seen as a refined version of the primal algorithm by maintaining dual variables (flow) in addition to primal variables. Although its time complexity is the same as that for the primal algorithm, we can expect a better performance practically.

We finally consider an application to a computer vision problem called the panoramic stitching problem. We apply several implementations of our primal-dual algorithm to some instances of the panoramic stitching problem and test their practical performance.

We also show that our primal algorithm as well as the proofs can be applied to the $\mathrm{L}^\natural$-convex function minimization problem which is a more general problem than the dual of the convex cost network flow problem.

1 Introduction

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple undirected graph. We assume that if $(u, v) \in \mathcal{E}$ then $(v, u) \notin \mathcal{E}$. We consider the following optimization problem:

$$\text{(DCCF): Minimize} \quad E(\boldsymbol{x}) = \sum_{u \in \mathcal{V}} D_u(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_v - x_u) \quad \text{subject to} \quad \boldsymbol{x} \in \mathbb{Z}^{\mathcal{V}},$$

where $D_u : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$ $(u \in \mathcal{V})$ and $V_{uv} : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$ $((u,v) \in \mathcal{E})$ are convex functions such that $\operatorname{dom} D_u = \{\alpha \in \mathbb{Z} \mid D_u(\alpha) < +\infty\}$ and $\operatorname{dom} V_{uv} = \{\alpha \in \mathbb{Z} \mid V_{uv}(\alpha) < +\infty\}$ are finite intervals. The problem of this type is known as the dual of the convex cost network flow problem and extensively discussed in the literature (see, e.g., [1, 19, 29, 28]). Recently, various applications of the problem (DCCF) have been studied in the area of image processing and computer vision (see, e.g, [7, 17, 31, 20, 24, 32, 9, 8]). In this paper, we propose new primal and primal-dual algorithms for the problem (DCCF) and prove a tight bound for the number of iterations required by the algorithms.

Previous Algorithms We denote $n = |\mathcal{V}|$, $m = |\mathcal{E}|$, and K is a positive integer such that

$$|\alpha - \beta| \leq K \qquad (\forall \alpha, \beta \in \operatorname{dom} D_u,\ \forall u \in \mathcal{V}). \tag{1.1}$$

It is well known that the problem (DCCF) can be solved exactly by reducing it to a linear program whose size is pseudo-polynomial in the input size of the problem. General LP solvers, however, do not exploit the special structure of (DCCF), and therefore are not very efficient.

Ishikawa [20] and Ahuja et al. [2] reduce the problem (DCCF) to a minimum s-t cut problem in a graph with $O(nK)$ nodes and $O(mK^2)$ edges. In the important special case when the functions $V_{uv}(\cdot)$ are given by piecewise linear functions with a constant number of breakpoints, the number of edges is reduced to $O(mK)$. A disadvantage of this approach is that it needs a large amount of memory (either $O(mK^2)$ or $O(mK)$).

Algorithms in [20, 2] can be seen as primal algorithms since they directly solve the problem (DCCF). An alternative is to solve the dual problem instead. Several dual algorithms were proposed by Karzanov and McCormick [21] and Ahuja et al. [1]. The worst-case complexity of the latter algorithm is $O(nm \log(n^2/m) \log(nK))$, which is the best known for (DCCF).

It is known [3] that the problem (DCCF) can be reduced to a linear cost network flow problem on a graph with $O(rm)$ edges, where $r \leq K$ is the maximum number of breakpoints of piecewise-linear convex functions $D_u(\cdot)$ and $V_{uv}(\cdot)$. Therefore, it is possible to use any existing method for the linear cost network flow problem. One of them, the primal-dual algorithm of Ford and Fulkerson [11, 12], is related to the technique that we develop in this paper. In particular, the two algorithms are equivalent in a special case when the functions $D_u(\cdot)$ are linear and the functions $V_{uv}(\cdot)$ have one breakpoint ($r = 1$). However, if $r > 1$ then the techniques are different; our algorithm works with a graph with $O(m)$ rather than $O(rm)$ edges.

Our Contributions In this paper we propose new primal and primal-dual algorithms for the problem (DCCF). Our primal algorithm finds an optimal solution of (DCCF) by at most $2K$ computation of a minimum cut of a graph with n nodes and m edges. The algorithm is similar to the steepest descent algorithm of Murota for the minimization of $\mathrm{L}^\natural$-convex functions [26, 27, 28]. The minimization of $\mathrm{L}^\natural$-convex functions is a more general problem than (DCCF) (see Section 2 for the definition of $\mathrm{L}^\natural$-convexity). The algorithm is also similar to that of Bioucas-Dias and Valadão [4] which was originally applied to the following special case without the functions $D_u(\cdot)$:

$$\text{(DCCF}_0\text{): Minimize} \quad E_0(\boldsymbol{x}) = \sum_{(u,v)\in\mathcal{E}} V_{uv}(x_v - x_u) \quad \text{subject to} \quad \boldsymbol{x} \in \mathbb{Z}^{\mathcal{V}}.$$

Our major contribution is to provide a tight bound on the number of iterations, while bounds in [4, 28] are much weaker. Our proof is based on the analysis of the L_∞ distance between the current feasible solution and an optimal solution, and it is shown that the distance decreases monotonically in each iteration. The proof is applicable not only to the problem (DCCF), but also to the minimization of $L^\natural$-convex functions. Hence, our result also implies a better bound on the number of iterations for Murota's steepest descent algorithm.

One drawback of the primal algorithm is that it solves different min cut/max flow problems independently, although these problems are strongly related. Thus, a natural idea for speeding up computations is to use maximum flow obtained in one iteration as an initialization for the next iteration. This is a motivation of our primal-dual method. It maintains both primal and dual variables. It makes at most $2K$ calls to a maximum flow algorithm and a shortest path procedure.

The worse-case complexity of both algorithms is $O(K) \cdot T(n, m)$ where $T(n, m)$ is the running time of one maximum flow computation on a graph with n nodes and m edges. This is worse than the complexity of the algorithm in [1]. However, our techniques have a practical advantage: they rely only on a maximum flow algorithm, which is more readily available. For example, it is possible to use maximum flow algorithm that was specifically tuned to computer vision problems [6]. Experimental results of our algorithms are shown in Section 5

Although the algorithms described above are pseudo-polynomial, it is possible to apply proximity scaling technique of Hochbaum and Shanthikumar [18] to get an algorithm polynomial in $\log K$ rather than K (see [2]). In particular, combining proximity scaling technique with our algorithms yields the time complexity $O(n \log K \cdot T(n, m))$.

Other Related Work Hochbaum [17] gives a very efficient algorithm for a special case of (DCCF). Namely, if the functions V_{uv} are given as $V_{uv}(x_v - x_u) = \lambda_{uv}|x_v - x_u|$, then the technique in [17] has almost the same time complexity as that of a single maximum flow computation on a graph with n nodes and m edges. Similar ideas appear in [31, 9, 8]. The method is applicable to the problem of image restoration using total variation minimization [17, 31, 9, 8].

If the functions $D_u(\cdot)$ are arbitrary and $V_{uv}(\cdot)$ are convex, then the problem can be solved exactly in time $T(nK, mK^2)$ or $T(nK, mK)$, depending on the structure of the functions V_{uv} [20, 2]. If both $D_u(\cdot)$ and $V_{uv}(\cdot)$ are arbitrary then the problem becomes NP-hard. Boykov et al. [7], Kleinberg and Tardos [22] and Komodakis and Tziritas [23] give constant factor approximation algorithms in the case when the functions $V_{uv}(\cdot)$ are metrics. Veksler [30] used procedures UP and DOWN, described below, as a heuristic tool for minimizing functions with *Potts* interaction terms, i.e. $V_{uv}(x_v - x_u) = \lambda_{uv} \min\{|x_v - x_u|, 1\}$.

Application to Computer Vision Problems It is known that the problem (DCCF) arises in many applications in computer vision such as panoramic image stitching [24, 32], image restoration [7], minimization of total variation [9], and phase unwrapping in SAR images [4]. In such applications, the node set $\mathcal{V}$ of the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ usually corresponds to the set of pixels in a given image, and the variable x_u represents the label of the pixel $u \in \mathcal{V}$ which must belong to a finite set of integers $\{0, 1, \ldots, K-1\}$. For motion or stereo, the labels are disparities, while for image restoration they represent intensities. The functions $D_u(\cdot)$ encode unary data penalty functions, and $V_{uv}(\cdot)$ are pairwise interaction potentials. The objective

function of (DCCF) is often derived in the context of Markov random fields [14]: a minimum of E corresponds to a maximum a-posteriori labeling $\boldsymbol{x}$.

In this paper, we consider the panoramic image stitching problem which inspired our work. Given different portions of the same scene with some overlap, the goal of panoramic image stitching is to generate an output image which is similar to the original images and does not have a visible seam. The approach of [24, 32] is to compute the image whose gradients match the gradients of the two input images, which can be done by solving an instance of (DCCF).

We apply our proposed algorithms to some instances of (DCCF) arising from actual panoramic image stitching problems, and test the empirical performance of our algorithms. In particular, we implement several versions of our primal-dual algorithms and compare their running time for the panoramic image stitching problems.

Outline In Section 2, we review the concept of $\mathrm{L}^\natural$-convex function. We also discuss the equivalence between the problems (DCCF) and (DCCF_0). In Section 3 we describe a primal algorithm and prove a bound on the number of iterations. In Section 4 we review the dual problem and present a primal-dual algorithm. In Section 5 we compare the speed of several algorithms on the panoramic image stitching problem.

2 Preliminaries

2.1 Review of Fundamental Results on $\mathrm{L}^\natural$-convex Functions

Our problem (DCCF) is closely related to the concepts of $\mathrm{L}^\natural$-convex functions introduced by Fujishige and Murota [13]. In this section, we review some fundamental results on $\mathrm{L}^\natural$-convex functions.

A function $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ with $\operatorname{dom} E \neq \emptyset$ is called L-convex if it satisfies the following properties:

(LF1) $E(\boldsymbol{x}) + E(\boldsymbol{y}) \geq E(\boldsymbol{x} \wedge \boldsymbol{y}) + E(\boldsymbol{x} \vee \boldsymbol{y}) \qquad (\forall \boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} E),$
(LF2) $\exists r \in \mathbb{R}$ such that $E(\boldsymbol{x} + \lambda \mathbf{1}) = E(\boldsymbol{x}) + \lambda r \quad (\forall \boldsymbol{x} \in \operatorname{dom} E,\ \forall \lambda \in \mathbb{Z}),$

where $\operatorname{dom} E = \{\boldsymbol{x} \in \mathbb{Z}^V \mid E(\boldsymbol{x}) < +\infty\}$, the vectors $\boldsymbol{x} \wedge \boldsymbol{y}, \boldsymbol{x} \vee \boldsymbol{y} \in \mathbb{Z}^{\mathcal{V}}$ are defined by

$$(\boldsymbol{x} \wedge \boldsymbol{y})_u = \min\{x_u, y_u\}, \quad (\boldsymbol{x} \vee \boldsymbol{y})_u = \max\{x_u, y_u\} \quad (u \in \mathcal{V}),$$

and $\mathbf{1} \in \mathbb{Z}^{\mathcal{V}}$ is the vector with all components equal to one. Throughout the paper, we assume that the value r in the property (LF2) is zero. Note, without this condition an L-convex function E does not have a minimum.

A function $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ with $\operatorname{dom} E \neq \emptyset$ is called $\mathrm{L}^\natural$-convex if the function $\tilde{E} : \mathbb{Z}^{\tilde{\mathcal{V}}} \to \mathbb{R} \cup \{+\infty\}$ defined by

$$\tilde{E}(x_0, \boldsymbol{x}) = E(\boldsymbol{x} - x_0 \mathbf{1}) \quad (x_0 \in \mathbb{Z}, \boldsymbol{x} \in \mathbb{Z}^{\mathcal{V}}) \tag{2.1}$$

is L-convex, where 0 denotes a new element not in $\mathcal{V}$ and $\tilde{\mathcal{V}} = \{0\} \cup \mathcal{V}$.

$\mathrm{L}^\natural$-convex functions are conceptually equivalent to L-convex functions, but the class of $\mathrm{L}^\natural$-convex functions contains that of L-convex functions as a proper subclass. $\mathrm{L}^\natural$-convexity is equivalent to integral convexity by Favati–Tardella [10] (see [27] for details).

By the equivalence between $\mathrm{L}^\natural$-convexity and L-convexity, all properties stated for $\mathrm{L}^\natural$-convex functions can be rephrased for L-convex functions, and vice versa. Hence, we primarily show properties of $\mathrm{L}^\natural$-convex functions below.

The next property shows that the objective function of the problem (DCCF) is $\mathrm{L}^\natural$-convex. That is, the problem (DCCF) is a special case of the minimization of an $\mathrm{L}^\natural$-convex function.

Proposition 2.1 (cf. [27])**.**
(i) *The objective function $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ of the problem* (DCCF) *is $L^\natural$-convex.*
(ii) *The objective function $E_0 : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ of the problem* (DCCF_0) *is L-convex with $r = 0$ in* (LF2).

Due to its definition, $\mathrm{L}^\natural$-convex functions satisfy submodular inequality.

Proposition 2.2. *Let $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ be an $L^\natural$-convex function. Then,*

$$E(\boldsymbol{x}) + E(\boldsymbol{y}) \geq E(\boldsymbol{x} \wedge \boldsymbol{y}) + E(\boldsymbol{x} \vee \boldsymbol{y}) \qquad (\forall \boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} E).$$

For any subset X of $\mathcal{V}$, we define $\chi_X \in \{0,1\}^{\mathcal{V}}$ by

$$(\chi_X)_u = \begin{cases} 1 & (u \in X), \\ 0 & (u \in \mathcal{V} \setminus X). \end{cases}$$

$\mathrm{L}^\natural$-convexity of a function defined over the integer lattice can be characterized by the following properties.

Theorem 2.3 ([27])**.** *Let $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ be a function with $\operatorname{dom} E \neq \emptyset$.*
(i) *E is $L^\natural$-convex if and only if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^{\mathcal{V}}$ with $\{u \in \mathcal{V} \mid x_u > y_u\} \neq \emptyset$, we have*

$$E(\boldsymbol{x}) + E(\boldsymbol{y}) \geq E(\boldsymbol{x} - \chi_W) + E(\boldsymbol{y} + \chi_W),$$

where $W = \arg\max\{x_u - y_u \mid u \in \mathcal{V}\}$.
(ii) *E is $L^\natural$-convex if and only if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^{\mathcal{V}}$ and $\lambda \in \mathbb{Z}_+$, we have*

$$E(\boldsymbol{x}) + E(\boldsymbol{y}) \geq E((\boldsymbol{x} - \lambda\mathbf{1}) \vee \boldsymbol{y}) + E(\boldsymbol{x} \wedge (\boldsymbol{y} + \lambda\mathbf{1})).$$

We show some properties for minimizers of an $\mathrm{L}^\natural$-convex function. Given a function $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$, let

$$\arg\min E = \{\boldsymbol{x} \in \operatorname{dom} E \mid E(\boldsymbol{x}) \leq E(\boldsymbol{y}) \ (\forall \boldsymbol{y} \in \mathbb{Z}^{\mathcal{V}})\}.$$

Minimizers of an $\mathrm{L}^\natural$-convex function can be characterized by local optimality.

Theorem 2.4 ([27])**.** *Let $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ be an $L^\natural$-convex function and $\boldsymbol{x} \in \operatorname{dom} E$. Then, $E(\boldsymbol{x}) \leq E(\boldsymbol{y})$ for all $\boldsymbol{y} \in \operatorname{dom} E$ if and only if $E(\boldsymbol{x}) \leq E(\boldsymbol{x} + \chi_X)$ for all $X \subseteq \mathcal{V}$ and $E(\boldsymbol{x}) \leq E(\boldsymbol{x} - \chi_X)$ for all $X \subseteq \mathcal{V}$.*

Proposition 2.5 (cf. [27])**.** *Let $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ be an $L^\natural$-convex function. For any $\boldsymbol{x}, \boldsymbol{y} \in \arg\min E$, we have*

$$\left\lceil \frac{\boldsymbol{x} + \boldsymbol{y}}{2} \right\rceil \in \arg\min E, \qquad \left\lfloor \frac{\boldsymbol{x} + \boldsymbol{y}}{2} \right\rfloor \in \arg\min E,$$

where $\lceil (\boldsymbol{x}+\boldsymbol{y})/2 \rceil$ and $\lfloor (\boldsymbol{x}+\boldsymbol{y})/2 \rfloor$ denote, respectively, the integer vectors obtained from $(\boldsymbol{x}+\boldsymbol{y})/2$ by component-wise round-up and round-down to the nearest integers.

Input: initial feasible solution $\boldsymbol{x} := \boldsymbol{x}^\circ \in \operatorname{dom} E$.
Step 1: Set SuccessUp := false, SuccessDown := false.
Step 2: Do UP or DOWN in any order until SuccessUp = SuccessDown = true:
UP (do only if SuccessUp is false):
- Compute $X^+ \in \arg\min\{E(\boldsymbol{x} + \chi_X) \mid X \subseteq \mathcal{V}\}$.
- If $E(\boldsymbol{x} + \chi_{X^+}) = E(\boldsymbol{x})$, set SuccessUp := true; otherwise set $\boldsymbol{x} := \boldsymbol{x} + \chi_{X^+}$.
DOWN (do only if SuccessDown is false):
- Compute $X^- \in \arg\min\{E(\boldsymbol{x} - \chi_X) \mid X \subseteq \mathcal{V}\}$.
- If $E(\boldsymbol{x} + \chi_{X^-}) = E(\boldsymbol{x})$, set SuccessDown := true; otherwise set $\boldsymbol{x} := \boldsymbol{x} - \chi_{X^-}$.
Step 3: Output $\boldsymbol{x}$ and stop.

Figure 1: Primal algorithm

Proposition 2.6 (cf. [27]). *Let $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ be an $L^\natural$-convex function, and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}^{\mathcal{V}}$ be vectors with $\{\boldsymbol{x} \in \operatorname{dom} E \mid a_u \leq x_u \leq b_u \ (u \in \mathcal{V})\} \neq \emptyset$. Then, $\arg\min\{E(\boldsymbol{x}) \mid a_u \leq x_u \leq b_u \ (u \in \mathcal{V})\}$ contains the unique minimal and maximal minimizers.*

2.2 Equivalence between the Problems (DCCF) and (DCCF$_0$)

We discuss the equivalence between the two problems (DCCF) and (DCCF$_0$). While the problem (DCCF$_0$) is a special case of (DCCF), it is known that (DCCF) can be reduced to the problem (DCCF$_0$), as shown below. Hence, the two problems (DCCF) and (DCCF$_0$) are essentially equivalent to each other, and any algorithm for the one problem can be adapted to the other.

Let $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ be an objective function of the problem (DCCF), and we define a function $\tilde{E} : \mathbb{Z}^{\tilde{\mathcal{V}}} \to \mathbb{R} \cup \{+\infty\}$ by (2.1). Then, we have

$$\begin{aligned} \tilde{E}(x_0, \boldsymbol{x}) &= E(\boldsymbol{x} - x_0\mathbf{1}) \\ &= \sum_{u \in \mathcal{V}} D_u(x_u - x_0) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_v - x_u) \quad (x_0 \in \mathbb{Z}, \boldsymbol{x} \in \mathbb{Z}^{\mathcal{V}}). \end{aligned}$$

We put $\tilde{\mathcal{E}} = \{(0, u) \mid u \in \mathcal{V}\} \cup \mathcal{E}$ and define a function $V_{0u} : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$ $(u \in \mathcal{V})$ by

$$V_{0u}(\alpha) = D_u(\alpha) \qquad (\alpha \in \mathbb{Z}).$$

Then, it holds that

$$\tilde{E}(x_0, \boldsymbol{x}) = \sum_{(u,v) \in \tilde{E}} V_{uv}(x_v - x_u) \qquad (x_0 \in \mathbb{Z}, \boldsymbol{x} \in \mathbb{Z}^{\mathcal{V}}).$$

Hence, we obtain an objective function of the problem (DCCF$_0$). This shows that the problem (DCCF) can be reduced to (DCCF$_0$).

3 Primal Algorithm

Our primal algorithm is described in Fig. 1. The primal algorithm can be applied to the minimization of L$^\natural$-convex function, which is a more general problem than (DCCF). We will

Input: initial feasible solution $\boldsymbol{x} := \boldsymbol{x}^\circ \in \operatorname{dom} E$.
Step 1: Compute $X^+ \in \arg\min\{E(\boldsymbol{x} + \chi_X) \mid X \subseteq \mathcal{V}\}$.
Step 2: Compute $X^- \in \arg\min\{E(\boldsymbol{x} - \chi_X) \mid X \subseteq \mathcal{V}\}$.
Step 3: If $E(\boldsymbol{x}) = \min\{E(\boldsymbol{x} + \chi_{X^+}), E(\boldsymbol{x} - \chi_{X^-})\}$, then output x and stop.
Step 4: If $E(\boldsymbol{x} + \chi_{X^+}) \leq E(\boldsymbol{x} - \chi_{X^-})$, then set $\boldsymbol{x} := \boldsymbol{x} + \chi_{X^+}$; otherwise set $\boldsymbol{x} := \boldsymbol{x} - \chi_{X^-}$.
Step 5: Go to Step 1.

Figure 2: Murota's steepest descent algorithm for the minimization of an $\mathrm{L}^\natural$-convex function

assume throughout this section that $E : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ is an $\mathrm{L}^\natural$-convex function, and consider the minimization of the function E. Any vector in the set $\operatorname{dom} E$ is said to be a feasible solution of the problem.

Our primal algorithm is very similar to the steepest descent algorithm of Murota [27, 28] for minimizing an $\mathrm{L}^\natural$-convex function (see Fig. 2). Murota's algorithm can be seen as a specialized implementation of our primal algorithm. While our algorithm has a flexibility in the choice of the procedures UP and DOWN, Murota's algorithm computes both of X^+ and X^- and chooses a better one by comparing the function values of $E(\boldsymbol{x} + \chi_{X^+})$ and $E(\boldsymbol{x} - \chi_{X^-})$. We note that the algorithm described in Fig. 2 is slightly different from Murota's original algorithm in the choice of $\tilde{X}^+$ and $\tilde{X}^-$; in Murota's original algorithm X^+ is the unique minimal set in $\arg\min\{E(\boldsymbol{x}+\chi_X) \mid X \subseteq \mathcal{V}\}$ and X^- is the unique maximal set in $\arg\min\{E(\boldsymbol{x}-\chi_X) \mid X \subseteq \mathcal{V}\}$.

Our primal algorithm is also similar to Murota's steepest descent algorithm for minimizing an L-convex function [26, 27, 28]. The algorithm uses only the procedure UP and therefore can be applied only to L-convex function minimization and its special case (DCCF_0). The algorithm of Bioucas-Dias and Valadão for (DCCF_0) [4] can be seen as a specialized version of Murota's algorithm.

3.1 Analysis of the Primal Algorithm

We prove the validity of our primal algorithm and analyze the number of iterations.

To analyze the number of iterations, we define values $\rho^+(\boldsymbol{x})$ and $\rho^-(\boldsymbol{x})$ for a vector $\boldsymbol{x} \in \operatorname{dom} E$ as

$$\rho^+(\boldsymbol{x}) = ||\hat{\boldsymbol{x}} - \boldsymbol{x}||_\infty, \qquad \rho^-(\boldsymbol{x}) = ||\check{\boldsymbol{x}} - \boldsymbol{x}||_\infty,$$

where $\hat{\boldsymbol{x}}$ denotes the unique minimal vector in $\arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \geq \boldsymbol{x}\}$ and $\check{\boldsymbol{x}}$ denotes the unique maximal vector in $\arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \leq \boldsymbol{x}\}$. We note that the existence of such $\hat{\boldsymbol{x}}$ and $\check{\boldsymbol{x}}$ is implied by Proposition 2.6.

Proposition 3.1. *For $\boldsymbol{x} \in \operatorname{dom} E$, if $\rho^+(\boldsymbol{x}) = \rho^-(\boldsymbol{x}) = 0$ then $\boldsymbol{x} \in \arg\min E$.*

Proof. If $\rho^+(\boldsymbol{x}) = \rho^-(\boldsymbol{x}) = 0$, then we have $E(\boldsymbol{x}) \leq E(\boldsymbol{x}+\epsilon\chi_X)$ for all $\epsilon \in \{+1, -1\}$ and $X \subseteq V$. This implies that x is a minimizer of E by Theorem 2.4. □

Note that $\rho^+(\boldsymbol{x}) = 0$ (resp. $\rho^-(\boldsymbol{x}) = 0$) alone implies $\boldsymbol{x} \in \arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \geq \boldsymbol{x}\}$ (resp. $\boldsymbol{x} \in \arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \leq \boldsymbol{x}\}$), but does not imply $\boldsymbol{x} \in \arg\min E$ in general.

Each iteration of the primal algorithm increases neither of $\rho^+(\boldsymbol{x})$ nor $\rho^-(\boldsymbol{x})$ and decreases strictly at least one of $\rho^+(\boldsymbol{x})$ and $\rho^-(\boldsymbol{x})$.

Proposition 3.2. *In each iteration of the primal algorithm, we have the following:*
(i) *If* $\rho^+(\boldsymbol{x}) > 0$, *then* $\rho^+(\boldsymbol{x} + \chi_{X^+}) = \rho^+(\boldsymbol{x}) - 1$ *and* $\rho^-(\boldsymbol{x} + \chi_{X^+}) \leq \rho^-(\boldsymbol{x})$.
(ii) *If* $\rho^-(\boldsymbol{x}) > 0$, *then* $\rho^-(\boldsymbol{x} - \chi_{X^-}) = \rho^-(\boldsymbol{x}) - 1$ *and* $\rho^+(\boldsymbol{x} - \chi_{X^-}) \leq \rho^+(\boldsymbol{x})$.

The proof is given at the end of this section.

Given a function E, we define

$$K_\infty = \max\{||\boldsymbol{x} - \boldsymbol{y}||_\infty \mid \boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} E\}.$$

It is easy to see that $\rho^+(\boldsymbol{x}) \leq K_\infty$ and $\rho^-(\boldsymbol{x}) \leq K_\infty$ for any $\boldsymbol{x} \in \operatorname{dom} E$. We note that if the function E is given as the objective function of (DCCF), then the integer K given by (1.1) satisfies $K \geq K_\infty$.

Theorem 3.3.
(i) *The output* $\boldsymbol{x}$ *of the primal algorithm satisfies* $\boldsymbol{x} \in \arg\min E$.
(ii) *The number of iterations of the primal algorithm is bounded by* $\rho^+(\boldsymbol{x}^\circ) + \rho^-(\boldsymbol{x}^\circ) + 2$, *which is further bounded by* $2K_\infty + 2$.

Proof. The claim (ii) is immediate from Proposition 3.2. We then prove (i). We see from Proposition 3.2 that once $\rho^+(\boldsymbol{x})$ (or $\rho^-(\boldsymbol{x})$) becomes zero, then it is kept until the termination of the algorithm. This implies that $\rho^+(\boldsymbol{x}) = \rho^-(\boldsymbol{x}) = 0$ holds at the end of the algorithm, and therefore the vector $\boldsymbol{x}$ is a minimizer of the $\mathrm{L}^\natural$-convex function E by Proposition 3.1. □

Clearly, this bound is tight in some cases. For example, consider the following problem:

$$\text{Minimize} \quad D_1(x_1) + D_2(x_2) \quad \text{subject to} \quad (x_1, x_2) \in \mathbb{Z}^2,$$

where k is a positive integer and $D_1, D_2 : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$ are functions defined by

$$D_1(\alpha) = \begin{cases} \alpha & (0 \leq \alpha \leq k), \\ +\infty & (\text{otherwise}), \end{cases} \qquad D_2(\alpha) = \begin{cases} -\alpha & (0 \leq \alpha \leq k), \\ +\infty & (\text{otherwise}). \end{cases}$$

This problem is a special case of (DCCF) with $K_\infty = K = k$ and $(0, k)$ is the unique optimal solution. If we start the primal algorithm with $\boldsymbol{x}^\circ = (k, 0)$, then the algorithm requires $2k + 2 = 2K_\infty + 2$ iterations.

We now prove Proposition 3.2. Recall that for any vector $\boldsymbol{x} \in \operatorname{dom} E$, the vector $\hat{\boldsymbol{x}}$ denotes the unique minimal vector in $\arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \geq \boldsymbol{x}\}$ and $\check{\boldsymbol{x}}$ denotes the unique maximal vector in $\arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \leq \boldsymbol{x}\}$.

Lemma 3.4. *Let* $\boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} E$.
(i) *Suppose that* $\boldsymbol{x} \leq \boldsymbol{y}$ *and* $E(\boldsymbol{y}) = \min\{E(\boldsymbol{z}) \mid \boldsymbol{x} \leq \boldsymbol{z} \leq \boldsymbol{y}\}$. *Then,* $\rho^+(\boldsymbol{y}) \leq \rho^+(\boldsymbol{x})$ *and* $\rho^-(\boldsymbol{y}) \leq \rho^-(\boldsymbol{x})$. *In particular, we have* $\hat{\boldsymbol{y}} = \hat{\boldsymbol{x}} \vee \boldsymbol{y}$.
(ii) *Suppose that* $\boldsymbol{x} \geq \boldsymbol{y}$ *and* $E(\boldsymbol{y}) = \min\{E(\boldsymbol{z}) \mid \boldsymbol{x} \geq \boldsymbol{z} \geq \boldsymbol{y}\}$. *Then,* $\rho^+(\boldsymbol{y}) \leq \rho^+(\boldsymbol{x})$ *and* $\rho^-(\boldsymbol{y}) \leq \rho^-(\boldsymbol{x})$. *In particular, we have* $\check{\boldsymbol{y}} = \check{\boldsymbol{x}} \wedge \boldsymbol{y}$.

Proof. We prove (i) only.

[Proof of "$\rho^+(\boldsymbol{y}) \leq \rho^+(\boldsymbol{x})$"] It suffices to show that $\hat{\boldsymbol{y}} = \hat{\boldsymbol{x}} \vee \boldsymbol{y}$ since it implies

$$\rho^+(\boldsymbol{y}) = ||\hat{\boldsymbol{y}} - \boldsymbol{y}||_\infty = ||(\hat{\boldsymbol{x}} \vee \boldsymbol{y}) - \boldsymbol{y}||_\infty \leq ||\hat{\boldsymbol{x}} - \boldsymbol{x}||_\infty = \rho^+(\boldsymbol{x}).$$

By Proposition 2.2, we have

$$E(\hat{\boldsymbol{x}}) + E(\boldsymbol{y}) \geq E(\hat{\boldsymbol{x}} \vee \boldsymbol{y}) + E(\hat{\boldsymbol{x}} \wedge \boldsymbol{y}). \tag{3.1}$$

Since $\boldsymbol{x} \leq \hat{\boldsymbol{x}} \wedge \boldsymbol{y} \leq \boldsymbol{y}$, we have $E(\boldsymbol{y}) \leq E(\hat{\boldsymbol{x}} \wedge \boldsymbol{y})$, which, together with (3.1), implies

$$E(\hat{\boldsymbol{x}}) \geq E(\hat{\boldsymbol{x}} \vee \boldsymbol{y}). \tag{3.2}$$

Since $\hat{\boldsymbol{y}} \geq \boldsymbol{y} \geq \boldsymbol{x}$ and $\hat{\boldsymbol{x}} \in \arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \geq \boldsymbol{x}\}$, we have

$$E(\hat{\boldsymbol{y}}) \geq E(\hat{\boldsymbol{x}}). \tag{3.3}$$

Similarly, we have

$$E(\hat{\boldsymbol{x}} \vee \boldsymbol{y}) \geq E(\hat{\boldsymbol{y}}) \tag{3.4}$$

since $\hat{\boldsymbol{x}} \vee \boldsymbol{y} \geq \boldsymbol{y}$ and $\hat{\boldsymbol{y}} \in \arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \geq \boldsymbol{y}\}$. It follows from (3.2), (3.3), and (3.4) that $E(\hat{\boldsymbol{x}}) = E(\hat{\boldsymbol{y}}) = E(\hat{\boldsymbol{x}} \vee \boldsymbol{y})$, which implies

$$\hat{\boldsymbol{y}} \in \arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \geq \boldsymbol{x}\}, \qquad \hat{\boldsymbol{x}} \vee \boldsymbol{y} \in \arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \geq \boldsymbol{y}\}.$$

It follows from the choices of $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ that $\hat{\boldsymbol{x}} \leq \hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{y}} \leq \hat{\boldsymbol{x}} \vee \boldsymbol{y}$. These inequalities and $\boldsymbol{y} \leq \hat{\boldsymbol{y}}$ imply $\hat{\boldsymbol{y}} = \hat{\boldsymbol{x}} \vee \boldsymbol{y}$.

[Proof of "$\rho^-(\boldsymbol{y}) \leq \rho^-(\boldsymbol{x})$"] We first show that $\check{\boldsymbol{x}} \leq \check{\boldsymbol{y}}$. By Proposition 2.2, we have

$$E(\check{\boldsymbol{x}}) + E(\check{\boldsymbol{y}}) \geq E(\check{\boldsymbol{x}} \vee \check{\boldsymbol{y}}) + E(\check{\boldsymbol{x}} \wedge \check{\boldsymbol{y}}). \tag{3.5}$$

Since $\check{\boldsymbol{x}} \in \arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \leq \boldsymbol{x}\}$ and $\check{\boldsymbol{x}} \wedge \check{\boldsymbol{y}} \leq \check{\boldsymbol{x}} \leq \boldsymbol{x}$, we have $E(\check{\boldsymbol{x}}) \leq E(\check{\boldsymbol{x}} \wedge \check{\boldsymbol{y}})$, which, together with (3.5), implies $E(\check{\boldsymbol{y}}) \geq E(\check{\boldsymbol{x}} \vee \check{\boldsymbol{y}})$. Since $\check{\boldsymbol{y}} \leq \check{\boldsymbol{x}} \vee \check{\boldsymbol{y}} \leq \boldsymbol{y}$ and $\check{\boldsymbol{y}}$ is the maximal vector in $\arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \leq \boldsymbol{y}\}$, we have $\check{\boldsymbol{y}} = \check{\boldsymbol{x}} \vee \check{\boldsymbol{y}}$, i.e., $\check{\boldsymbol{x}} \leq \check{\boldsymbol{y}}$.

We may assume that $\rho^-(\boldsymbol{y}) > 0$ since otherwise the inequality holds immediately. Put $\lambda = \rho^-(\boldsymbol{y})$ and $W = \{u \in \mathcal{V} \mid y_u - \check{y}_u = \lambda\}$. By Theorem 2.3, we have

$$E(\boldsymbol{y}) + E(\check{\boldsymbol{y}}) \geq E(\boldsymbol{y} - \chi_W) + E(\check{\boldsymbol{y}} + \chi_W).$$

We also have $E(\check{\boldsymbol{y}}) < E(\check{\boldsymbol{y}} + \chi_W)$ by the definition of $\check{\boldsymbol{y}}$ and the inequality $\check{\boldsymbol{y}} + \chi_W \leq \boldsymbol{y}$. Hence, it holds that $E(\boldsymbol{y} - \chi_W) < E(\boldsymbol{y})$. Since $E(\boldsymbol{y}) = \min\{E(\boldsymbol{z}) \mid \boldsymbol{x} \leq \boldsymbol{z} \leq \boldsymbol{y}\}$, if $\boldsymbol{y} - \chi_W \geq \boldsymbol{x}$ then we have $E(\boldsymbol{y} - \chi_W) \geq E(\boldsymbol{y})$, a contradiction. Hence, there exists some $u \in \mathcal{V}$ such that $x_u = y_u$ and $u \in W$. This implies that $x_u = y_u = \check{y}_u + \lambda \geq \check{x}_u + \lambda$. Therefore, $\rho^-(\boldsymbol{x}) \geq x_u - \check{x}_u \geq \lambda = \rho^-(\boldsymbol{y})$. □

Proof of Proposition 3.2. We prove (i) only; the claim (ii) can be shown in the same way.

Put $\boldsymbol{y} = \boldsymbol{x} + \chi_{X^+}$. Then, we have $E(\boldsymbol{y}) = \min\{E(\boldsymbol{z}) \mid \boldsymbol{x} \leq \boldsymbol{z} \leq \boldsymbol{y}\}$, which implies $\rho^+(\boldsymbol{y}) \leq \rho^+(\boldsymbol{x})$ and $\rho^-(\boldsymbol{y}) \leq \rho^-(\boldsymbol{x})$ by Lemma 3.4. To prove $\rho^+(\boldsymbol{y}) = \rho^+(\boldsymbol{x}) - 1$, it suffices to show that

$$S \subseteq X^+, \quad \text{where } S = \arg\max\{\hat{x}_u - x_u \mid u \in \mathcal{V}\}.$$

Then, this and Lemma 3.4 imply

$$\rho^+(\boldsymbol{y}) = ||\hat{\boldsymbol{y}} - \boldsymbol{y}||_\infty = ||(\hat{\boldsymbol{x}} \vee \boldsymbol{y}) - \boldsymbol{y}||_\infty = ||\hat{\boldsymbol{x}} - \boldsymbol{x}||_\infty - 1 = \rho^+(\boldsymbol{x}) - 1.$$

Input: initial feasible solution $\tilde{\boldsymbol{x}} := \tilde{\boldsymbol{x}}^\circ \in \operatorname{dom} \tilde{E}$.
Step 1: Compute $\tilde{X}^+ \in \arg\min\{\tilde{E}(\tilde{\boldsymbol{x}} + \chi_{\tilde{X}}) \mid \tilde{X} \subseteq \tilde{\mathcal{V}}\}$.
Step 2: If $\tilde{E}(\tilde{\boldsymbol{x}}) = \tilde{E}(\tilde{\boldsymbol{x}} + \chi_{\tilde{X}^+})$, then output $\tilde{\boldsymbol{x}}$ and stop.
Step 3: Set $\tilde{\boldsymbol{x}} := \tilde{\boldsymbol{x}} + \chi_{\tilde{X}^+}$.
Step 4: Go to Step 1.

Figure 3: Murota's steepest descent algorithm for the minimization of an L-convex function

Assume, to the contrary, that $S \setminus X^+ \neq \emptyset$. Put

$$S' = \arg\max\{\hat{x}_u - x_u - (\chi_{X^+})_u \mid u \in \mathcal{V}\} = S \setminus X^+.$$

Theorem 2.3 implies

$$E(\hat{\boldsymbol{x}}) + E(\boldsymbol{x} + \chi_{X^+}) \geq E(\hat{\boldsymbol{x}} - \chi_{S'}) + E(\boldsymbol{x} + \chi_{X^+} + \chi_{S'}).$$

Since $\chi_{X^+} + \chi_{S'} = \chi_{X^+ \cup S}$, we have $E(\boldsymbol{x} + \chi_{X^+} + \chi_{S'}) = E(\boldsymbol{x} + \chi_{X^+ \cup S}) \geq E(\boldsymbol{x} + \chi_{X^+})$, where the inequality is by the choice of X^+. Hence, we have $E(\hat{\boldsymbol{x}}) \geq E(\hat{\boldsymbol{x}} - \chi_{S'})$, a contradiction to the fact that $\hat{\boldsymbol{x}}$ is the minimal vector in $\arg\min\{E(\boldsymbol{z}) \mid \boldsymbol{z} \geq \boldsymbol{x}\}$ since $\hat{\boldsymbol{x}} - \chi_{S'} \geq \boldsymbol{x}$. □

3.2 Analysis of Murota's Steepest Descent Algorithm

Since Murota's steepest descent algorithm for L$^\natural$-convex function can be seen as a specialized implementation of our primal algorithm, Theorem 3.3 implies that Murota's algorithm terminates in $O(K_\infty)$ iterations, which is much better than the bound $O(K_1)$ shown in [28], where

$$K_1 = \max\{||\boldsymbol{x} - \boldsymbol{y}||_1 \mid \boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} E\}.$$

In this section, we compare our primal algorithm and Murota's algorithm. In particular, we show the following:

- Our primal algorithm requires the same or larger number of iterations.
- Our primal algorithm requires the same or fewer total number of calls to the minimization procedure in UP and DOWN.

Note that one iteration of Murota's algorithm makes two calls to the procedure for minimizing a submodular function, so it is roughly twice as expensive as one iteration of the primal algorithm.

3.2.1 Analysis of Steepest Descent Algorithm for L-convex Functions

In [28], Murota firstly proposes a steepest descent algorithm for L-convex functions, which is then adapted to L$^\natural$-convex functions through the relation (2.1). For the simplicity of the proof, we firstly analyze the number of iterations required by the algorithm for L-convex functions, and then restate the result in terms of L$^\natural$-convex functions.

Murota's steepest descent algorithm for L-convex functions is described in Fig. 3, where $\tilde{\mathcal{V}} = \{0\} \cup \mathcal{V}$ and $\tilde{E} : \mathbb{Z}^{\tilde{\mathcal{V}}} \to \mathbb{R} \cup \{+\infty\}$ is an L-convex function with $\operatorname{dom} \tilde{E} \neq \emptyset$. We note

that the algorithm described in Fig. 3 is slightly different from Murota's original algorithm in the choice of $\tilde{X}^+$ in Step 1; in Murota's original algorithm $\tilde{X}^+$ is the unique minimal set in $\arg\min\{\tilde{E}(\tilde{\boldsymbol{x}} + \chi_{\tilde{X}}) \mid \tilde{X} \subseteq \tilde{\mathcal{V}}\}$.

Given a vector $\tilde{\boldsymbol{x}}^\circ \in \mathbb{Z}^{\tilde{\mathcal{V}}}$, we denote by $\tilde{\boldsymbol{x}}^*$ the unique minimal vector in the set $\arg\min\{\tilde{E}(\tilde{\boldsymbol{z}}) \mid \tilde{\boldsymbol{z}} \geq \tilde{\boldsymbol{x}}^\circ\}$ and define $\tilde{\mu}(\tilde{\boldsymbol{x}}^\circ) = ||\tilde{\boldsymbol{x}}^* - \tilde{\boldsymbol{x}}^\circ||_\infty$. It should be mentioned that the definition of $\tilde{\mu}(\tilde{\boldsymbol{x}}^\circ)$ will not change if $\tilde{\boldsymbol{x}}^*$ is the unique maximal vector in $\arg\min\{\tilde{E}(\tilde{\boldsymbol{z}}) \mid \tilde{\boldsymbol{z}} \leq \tilde{\boldsymbol{x}}^\circ\}$ rather than the unique minimal vector in $\arg\min\{\tilde{E}(\tilde{\boldsymbol{z}}) \mid \tilde{\boldsymbol{z}} \geq \tilde{\boldsymbol{x}}^\circ\}$.

The property (LF2) of L-convex functions implies that

$$\tilde{\boldsymbol{x}}^* \in \arg\min\{\tilde{E}(\tilde{\boldsymbol{z}}) \mid \tilde{\boldsymbol{z}} \geq \tilde{\boldsymbol{x}}^\circ\} \subseteq \arg\min \tilde{E},$$

i.e., the vector $\tilde{\boldsymbol{x}}^*$ is a minimizer of the function $\tilde{E}$. On the other hand, it is easy to see that Murota's algorithm is the same as our primal algorithm except that the procedure DOWN is missing. Therefore, the discussion in Section 3.1 shows that Murota's algorithm outputs the vector $\tilde{\boldsymbol{x}}^*$ in $\tilde{\mu}(\tilde{\boldsymbol{x}}^\circ) + 1$ iterations.

Theorem 3.5. *The number of iterations of Murota's steepest descent algorithm for L-convex function $\tilde{E}$ is equal to $\tilde{\mu}(\tilde{\boldsymbol{x}}^\circ) + 1$.*

3.2.2 Analysis of Steepest Descent Algorithm for L$^\natural$-convex Functions

We now analyze the number of iterations required by the steepest descent algorithm for L$^\natural$-convex functions.

The behavior of the steepest descent algorithm for an L$^\natural$-convex function E with the initial vector $\boldsymbol{x}^\circ$ is essentially the same as that of the steepest descent algorithm for the L-convex function $\tilde{E}$ defined by (2.1) with the initial vector $\tilde{\boldsymbol{y}}^\circ = (0, \boldsymbol{x}^\circ) \in \mathbb{Z} \times \mathbb{Z}^{\mathcal{V}}$. The correspondence between the two steepest descent algorithms is as follows (see [28]):

L$^\natural$-convex E		L-convex $\tilde{E}$
$\boldsymbol{x} \to \boldsymbol{x} + \chi_X$	$\Longleftrightarrow$	$\tilde{\boldsymbol{y}} \to \tilde{\boldsymbol{y}} + (0, \chi_X)$
$\boldsymbol{x} \to \boldsymbol{x} - \chi_X$	$\Longleftrightarrow$	$\tilde{\boldsymbol{y}} \to \tilde{\boldsymbol{y}} + (1, \chi_{\mathcal{V}\setminus X})$

where $\tilde{\boldsymbol{y}} = (x_0, \boldsymbol{x} + x_0\mathbf{1})$ and x_0 is a nonnegative integer representing the number of iterations with "$\boldsymbol{x} \to \boldsymbol{x} - \chi_X$" so far.

For a vector $\boldsymbol{x} \in \mathbb{Z}^{\mathcal{V}}$ we define $\mu(\boldsymbol{x}) = \tilde{\mu}(0, \boldsymbol{x})$. As a corollary of Theorem 3.5 we obtain the following bound for the number of iterations.

Theorem 3.6. *The number of iterations of Murota's steepest descent algorithm for L$^\natural$-convex function E is equal to $\mu(\boldsymbol{x}^\circ) + 1$.*

We now show that our primal algorithm requires the same or larger number of iterations than Murota's algorithm.

Theorem 3.7. *The number of iterations of our primal algorithm for L$^\natural$-convex function E is at least $\mu(\boldsymbol{x}^\circ) + 1$.*

Proof. Let $\boldsymbol{x}$ be the output of our primal algorithm applied to $\boldsymbol{x}^\circ$. Denote

$$\begin{aligned} d^+ &= \max\left[0, \max\{x_u - x_u^\circ \mid u \in \mathcal{V},\ x_u > x_u^\circ\}\right], \\ d^- &= \max\left[0, \max\{x_u^\circ - x_u \mid u \in \mathcal{V},\ x_u < x_u^\circ\}\right]. \end{aligned}$$

Clearly, the primal algorithm calls procedure UP (resp., DOWN) at least $d^+ + 1$ (resp., $d^- + 1$) times. We will show next that $d^+ + d^- \geq \mu(\boldsymbol{x}^\circ)$, which will imply the theorem.

Consider vector $\tilde{\boldsymbol{y}} = (d^-, \boldsymbol{x} + d^-\mathbf{1})$. Since $\tilde{E}(\tilde{\boldsymbol{y}}) = \tilde{E}(0, \boldsymbol{x})$ and $(0, \boldsymbol{x})$ is a minimizer of $\tilde{E}$, vector $\tilde{\boldsymbol{y}}$ is also a minimizer. Furthermore, $\tilde{\boldsymbol{y}} \geq (0, \boldsymbol{x}^\circ)$. Thus, $||\tilde{\boldsymbol{y}} - (0, \boldsymbol{x}^\circ)||_\infty \geq \tilde{\mu}(0, \boldsymbol{x}^\circ)$. It remains to notice that $||\tilde{\boldsymbol{y}} - (0, \boldsymbol{x}^\circ)||_\infty = d^+ + d^-$. □

We then show that our primal algorithm requires the same or fewer total number of calls to the minimization procedure in UP and DOWN than Murota's algorithm.

Theorem 3.8. *For any feasible solution $\boldsymbol{x} \in \operatorname{dom} E$ there holds $\rho^+(\boldsymbol{x}) \leq \mu(\boldsymbol{x})$, $\rho^-(\boldsymbol{x}) \leq \mu(\boldsymbol{x})$.*

Proof. We prove only the first inequality. Let $\tilde{\boldsymbol{x}}^*$ be the minimal vector in $\arg\min\{\tilde{E}(\boldsymbol{y}) \mid \tilde{\boldsymbol{y}} \geq (0, \boldsymbol{x})\}$. Then, $\mu(\boldsymbol{x}) = ||\tilde{\boldsymbol{x}}^* - (0, \boldsymbol{x})||_\infty$. We will show next that $(0, \hat{\boldsymbol{x}}) \leq \tilde{\boldsymbol{x}}^*$. This will imply the desired inequality since $\rho^+(\boldsymbol{x}) = ||(0, \hat{\boldsymbol{x}}) - (0, \boldsymbol{x})||_\infty$.

Define vectors $\tilde{\boldsymbol{y}} = (y_0, \boldsymbol{y}) = \tilde{\boldsymbol{x}}^* \wedge (0, \hat{\boldsymbol{x}})$, $\tilde{\boldsymbol{z}} = \tilde{\boldsymbol{x}}^* \vee (0, \hat{\boldsymbol{x}})$. Clearly, $y_0 = 0$. We have

$$E(\boldsymbol{y}) = \tilde{E}(\tilde{\boldsymbol{y}}) \leq \tilde{E}(0, \hat{\boldsymbol{x}}) + [\tilde{E}(\tilde{\boldsymbol{x}}^*) - \tilde{E}(\tilde{\boldsymbol{z}})] \leq \tilde{E}(0, \hat{\boldsymbol{x}}) = E(\hat{\boldsymbol{x}}),$$

where the first inequality follows from submodularity of $\tilde{E}$, and the second inequality follows from the optimality of $\tilde{\boldsymbol{x}}^*$ and the fact that $\tilde{\boldsymbol{z}} \geq (0, \boldsymbol{x})$. Since $\boldsymbol{y} \geq \boldsymbol{x}$ and $E(\boldsymbol{y}) \leq E(\hat{\boldsymbol{x}})$, we have $\hat{\boldsymbol{x}} \leq \boldsymbol{y}$. Thus, $(0, \hat{\boldsymbol{x}}) \leq \tilde{\boldsymbol{y}} \leq \tilde{\boldsymbol{x}}^*$, as claimed. □

We note that our algorithm makes at most $\rho^+(\boldsymbol{x}^\circ) + \rho^+(\boldsymbol{x}^\circ) + 2$ calls to the procedure for minimizing a submodular function, while Murota's algorithm makes $2\mu(\boldsymbol{x}^\circ) + 2$ such calls. Thus, the theorem implies that our algorithm makes the same of fewer number of calls.

It should be mentioned that Murota's algorithm can be implemented so that it calls the procedure for minimizing a submodular function only once in each iteration. Instead of computing both of X^+ and X^- and choosing a better one, we just need to compute $\tilde{X}^+ \in \arg\min\{\tilde{E}(\tilde{\boldsymbol{x}} + \chi_{\tilde{X}}) \mid \tilde{X} \subseteq \tilde{\mathcal{V}}\}$, as in Murota's algorithm for L-convex function, and then compute X^+ or X^- by using $\tilde{X}^+$.

4 Primal-Dual Algorithm

In order to describe our primal-dual algorithm, we need to review the dual of the problem (DCCF), which is done in Section 4.1. Based on this, we then present our primal-dual algorithm in Section 4.2. It can be viewed as an extension of our primal algorithm. It also uses procedures UP and DOWN; however, during these procedures the algorithm updates not only primal variables $\boldsymbol{x}$ but also dual variables, namely flow. After describing the algorithm we show the validity of the algorithm and analyze the time complexity in Section 4.3.

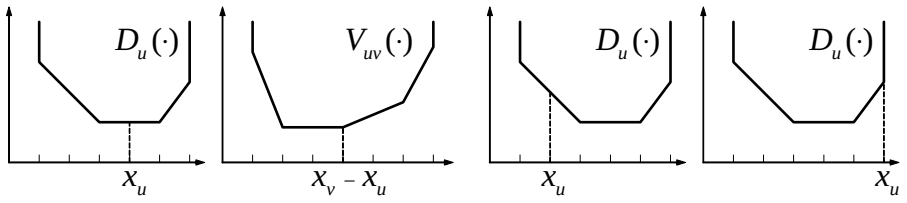

Figure 4: Optimality condition. Any optimal primal-dual pair $(\boldsymbol{x}, f)$ must satisfy the conditions illustrated in (a) and (b). During the algorithm, the condition for edges shown in (b) is always satisfied. However, this condition may be violated for some nodes (possible cases are shown in (a), (c) and (d)).

4.1 Flow and the Dual Problem

It is well known that a convex cost network flow problem can be obtained as the dual of the problem (DCCF) in the following way (see, e.g., [1, 19, 29]). We define $\mathcal{A} = \{(u, v) \mid (u, v) \in \mathcal{E}\} \cup \{(v, u) \mid (u, v) \in \mathcal{E}\}$, i.e., $(\mathcal{V}, \mathcal{A})$ is a directed graph corresponding to the undirected graph $(\mathcal{V}, \mathcal{E})$. A flow is a vector $f \in \mathbb{R}^{\mathcal{V} \cup \mathcal{A}}$ satisfying

$$\begin{array}{lcll} f_{uv} & = & -f_{vu} & ((u, v) \in \mathcal{E}) \quad \text{(antisymmetry)}, \\ f_u & = & \displaystyle\sum_{(u,v) \in \mathcal{E}} f_{uv} & (u \in \mathcal{V}) \quad \text{(flow conservation)}. \end{array}$$

Given a flow f, we define a function $E^f(\boldsymbol{x}) : \mathbb{Z}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ as follows:

$$E^f(\boldsymbol{x}) = \sum_{u \in \mathcal{V}} D_u^f(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}^f(x_v - x_u), \tag{4.1}$$

where

$$\begin{array}{lcll} D_u^f(\alpha) & = & D_u(\alpha) - f_u \cdot \alpha & (\alpha \in \mathbb{Z}), \\ V_{uv}^f(\alpha) & = & V_{uv}(\alpha) - f_{uv} \cdot \alpha & (\alpha \in \mathbb{Z}). \end{array}$$

It is not difficult to check that for any flow f functions E^f and E are the same, i.e., $E^f(\boldsymbol{x}) = E(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{Z}^{\mathcal{V}}$. Furthermore, the functions $D_u^f(\cdot)$ and $V_{uv}^f(\cdot)$ are convex.

For a flow f, let us define a function $H : \mathbb{R}^{\mathcal{V} \cup \mathcal{A}} \to \mathbb{R} \cup \{-\infty\}$ as

$$H(f) = \sum_{u \in \mathcal{V}} \min_{\alpha \in \mathbb{Z}} D_u^f(\alpha) + \sum_{(u,v) \in \mathcal{E}} \min_{\alpha \in \mathbb{Z}} V_{uv}^f(\alpha).$$

We note that $\min_{\alpha \in \mathbb{Z}} D_u^f(\alpha)$ (resp., $\min_{\alpha \in \mathbb{Z}} V_{uv}^f(\alpha)$) is a concave function in variable f_u (resp., in variable f_{uv}).

We now consider the following optimization problem:

$$\text{(CCF):} \quad \text{Maximize} \quad H(f) \quad \text{subject to} \quad f \in \mathbb{R}^{\mathcal{V} \cup \mathcal{A}},\ f \text{ is a flow.}$$

This is the dual of the problem (DCCF), and essentially the minimization of a convex objective function over flows.

Clearly, $H(f)$ is a lower bound of the function value $E^f(\boldsymbol{x}) = E(\boldsymbol{x})$:

$$H(f) \le E(\boldsymbol{x}) \qquad \text{for any flow } f \text{ and any feasible solution } \boldsymbol{x} \in \operatorname{dom} E. \tag{4.2}$$

This is a statement of weak duality. It turns out that strong duality holds as well.

Theorem 4.1 (cf. [1, 19, 29])**.**
(a) [strong duality] *Let* $f^* \in \mathbb{R}^{\mathcal{V}\cup\mathcal{A}}$ *and* $\boldsymbol{x}^* \in \operatorname{dom} E$ *be optimal solutions of the problems* (CCF) *and* (DCCF)*, respectively. Then,*

$$H(f^*) = E(\boldsymbol{x}^*).$$

(b) [optimality condition] *A flow* $f \in \mathbb{R}^{\mathcal{V}\cup\mathcal{A}}$ *and a feasible solution* $\boldsymbol{x} \in \operatorname{dom} E$ *are optimal solutions of the problems* (CCF) *and* (DCCF) *if and only if the following conditions hold:*

$$D_u^f(x_u) = \min_{\alpha\in\mathbb{Z}} D_u^f(\alpha) \qquad (\forall u \in \mathcal{V}), \tag{4.3a}$$

$$V_{uv}^f(x_v - x_u) = \min_{\alpha\in\mathbb{Z}} V_{uv}^f(\alpha) \qquad (\forall (u,v) \in \mathcal{E}). \tag{4.3b}$$

We will prove that our algorithm finds a pair $(\boldsymbol{x}, f)$ satisfying the optimality condition (4.3). The optimality condition (4.3) is illustrated in Fig. 4(a) and (b). Since the functions $D_u^f(\cdot)$ and $V_{uv}^f(\cdot)$ are convex, these conditions are equivalent to

$$\begin{aligned} &\operatorname{grad} D_u^f(x_u - 0) \le 0, \qquad \operatorname{grad} D_u^f(x_u + 0) \ge 0, \\ &\operatorname{grad} V_{uv}^f(x_v - x_u - 0) \le 0, \qquad \operatorname{grad} V_{uv}^f(x_v - x_u + 0) \ge 0, \end{aligned}$$

where we use the following notation for a function $g : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$:

$$\operatorname{grad} g(\alpha - 0) = g(\alpha) - g(\alpha - 1), \quad \operatorname{grad} g(\alpha + 0) = g(\alpha + 1) - g(\alpha) \qquad (\alpha \in \mathbb{Z}).$$

4.2 Algorithm

We are now ready to present our primal-dual algorithm. It maintains a feasible solution $\boldsymbol{x} \in \operatorname{dom} E$ and a flow $f \in \mathbb{R}^{\mathcal{V}\cup\mathcal{A}}$. An important property of the algorithm is that the condition (4.3b) is always satisfied, although the condition (4.3a) for nodes may not hold. Therefore, we have the cases shown in Fig. 4 (c) and (d). It is convenient to use the following notation for sets of nodes violating the conditions:

$$\begin{aligned} \mathcal{V}^+(\boldsymbol{x}, f) &= \{u \in \mathcal{V} \mid \operatorname{grad} D_u^f(x_u + 0) < 0\}, \\ \mathcal{V}^-(\boldsymbol{x}, f) &= \{u \in \mathcal{V} \mid \operatorname{grad} D_u^f(x_u - 0) > 0\}. \end{aligned}$$

To simplify notation, for zero flow we denote $\mathcal{V}^+(\boldsymbol{x}) = \mathcal{V}^+(\boldsymbol{x}, 0)$ and $\mathcal{V}^-(\boldsymbol{x}) = \mathcal{V}^-(\boldsymbol{x}, 0)$. Fig. 4 (c) corresponds to a node in $\mathcal{V}^+(\boldsymbol{x})$, while Fig. 4 (d) corresponds to a node in $\mathcal{V}^-(\boldsymbol{x})$. Note that since the functions $D_u^f(\cdot)$ are convex, a node cannot be in both sets simultaneously. Furthermore, the condition (4.3a) holds if and only if sets $\mathcal{V}^+(\boldsymbol{x}, f)$ and $\mathcal{V}^-(\boldsymbol{x}, f)$ are empty.

The outline of the algorithm is shown in Fig. 5. We now give details of each procedure. Some of them involve modifying flow f. For simplicity of notation we will make the following convention: after every modification of f the function E is replaced with the function E^f, and

Input: initial feasible solution $\boldsymbol{x} := \boldsymbol{x}^\circ \in \operatorname{dom} E$.

1. INITIALIZE-FLOW (updates f)
2. Set SuccessUp := false, SuccessDown := false.
3. Do UP or DOWN in any order until SuccessUp = SuccessDown = true:

   UP (do only if SuccessUp is false):
   - MAXFLOW-UP (updates $\boldsymbol{x}$ and f)
   - If $\mathcal{V}^+(\boldsymbol{x}, f) = \emptyset$, set SuccessUp := true
   - Optional: DIJKSTRA-UP (updates $\boldsymbol{x}$)

   DOWN (do only if SuccessDown is false):
   - MAXFLOW-DOWN (updates $\boldsymbol{x}$ and f)
   - If $\mathcal{V}^-(\boldsymbol{x}, f) = \emptyset$, set SuccessDown := true
   - Optional: DIJKSTRA-DOWN (updates $\boldsymbol{x}$)

4. Optional: DIJKSTRA-DOWN; set $\boldsymbol{x}^{\min} := \boldsymbol{x}$.
5. Optional: DIJKSTRA-UP; set $\boldsymbol{x}^{\max} := \boldsymbol{x}$.

Figure 5: Our primal-dual algorithm. See text for description of procedures. Upon termination $\boldsymbol{x}$ is a minimizer of E, $\boldsymbol{x}^{\min}$ is the unique minimal minimizer, $\boldsymbol{x}^{\max}$ is the unique maximal minimizer, and f is an optimal flow.

the flow is set to zero[1]. This means that at the beginning of each operation the input flow is zero.

INITIALIZE-FLOW Its goal is to set flow f so that the condition (4.3b) is satisfied for every edge $(u, v) \in \mathcal{E}$. Since $\operatorname{grad} V_{uv}^f(x_v - x_u \pm 0) = \operatorname{grad} V_{uv}(x_v - x_u \pm 0) - f_{uv}$, a necessary and sufficient condition for flow f_{uv} is

$$\operatorname{grad} V_{uv}(x_v - x_u - 0) \le f_{uv} \le \operatorname{grad} V_{uv}(x_v - x_u + 0)$$

After setting f_{uv} values f_u are computed from the flow conservation constraint.

MAXFLOW-UP This operation is similar to procedure UP of the primal algorithm, except that it modifies not only feasible solution $\boldsymbol{x}$ but also flow f. We will show later that it tries to move towards satisfying (4.3a). In particular, sets $\mathcal{V}^+(\boldsymbol{x}, f)$ and $\mathcal{V}^-(\boldsymbol{x}, f)$ will not grow. The procedure can be summarized as follows.

First, we construct a directed graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{A}})$ such that

$$\begin{aligned} \hat{\mathcal{V}} &= \mathcal{V} \cup \{s, t\}, \\ \hat{\mathcal{A}} &= \mathcal{A} \cup \{(s, u), (u, s) \mid u \in \mathcal{V} \setminus \mathcal{V}^+(\boldsymbol{x})\} \cup \{(u, t), (t, u) \mid u \in \mathcal{V}^+(\boldsymbol{x})\}, \end{aligned}$$

[1]For implementation it is also possible to keep original functions $D(\cdot)$, $V(\cdot)$ and flow f. When calling a particular procedure, we first compute functions $D^f(\cdot)$, $V^f(\cdot)$ and supply them to the procedure. If the output of the operation is flow f' then it is added to f, so $f + f'$ becomes the new flow.

where s and t are called the source and the sink, respectively. We also consider a capacity $\hat{c}_{uv}$ for $(u, v) \in \hat{\mathcal{A}}$ defined by

$$\begin{array}{lll} \hat{c}_{uv} = \operatorname{grad} V_{uv}(x_v - x_u + 0), & \hat{c}_{vu} = -\operatorname{grad} V_{uv}(x_v - x_u - 0) & \text{for } (u, v) \in \mathcal{E}, \\ \hat{c}_{su} = \operatorname{grad} D_u(x_u + 0), & \hat{c}_{us} = 0 & \text{for } u \in \mathcal{V} \setminus \mathcal{V}^+(\boldsymbol{x}), \\ \hat{c}_{ut} = -\operatorname{grad} D_u(x_u + 0), & \hat{c}_{tu} = 0 & \text{for } u \in \mathcal{V}^+(\boldsymbol{x}). \end{array}$$

We note that all of the capacities are nonnegative by the condition (4.3b).

We then solve the following maximum flow problem:

$$\begin{array}{lll} \text{Maximize} & \displaystyle\sum_{(s,u)\in\hat{\mathcal{A}}} \hat{f}_{su} & \\ \text{subject to} & \hat{f}_{uv} \le \hat{c}_{uv} & (\forall (u, v) \in \hat{\mathcal{A}}), \\ & \hat{f}_{uv} = -\hat{f}_{vu} & (\forall (u, v) \in \hat{\mathcal{A}}), \\ & \displaystyle\sum_{u:(u,v)\in\hat{\mathcal{A}}} \hat{f}_{uv} = 0 & (\forall v \in \hat{\mathcal{V}} \setminus \{s, t\}), \\ & \hat{f} \in \mathbb{R}^{\hat{\mathcal{A}}}. & \end{array}$$

Finally, we update the flow f by using $\hat{f}$ as follows:

$$\begin{array}{rcll} f_{uv} & := & f_{uv} + \hat{f}_{uv} & \text{for } (u, v) \in \mathcal{A}, \\ f_u & := & f_u + \hat{f}_{su} & \text{for } (s, u) \in \hat{\mathcal{A}}, \\ f_u & := & f_u - \hat{f}_{ut} & \text{for } (u, t) \in \hat{\mathcal{A}}. \end{array}$$

It is easy to see that the output flow f satisfies antisymmetry and flow conservation constraints, since the same holds for flow $\hat{f}$.

A new feasible solution $\boldsymbol{y}$ is computed from a minimum s-t cut of the graph $\hat{\mathcal{G}}$. An s-t cut (S, T) of the graph $\hat{\mathcal{G}}$ is a pair of subsets of $\mathcal{V}$ such that $\{S, T\}$ is a partition of $\mathcal{V}$ and $s \in S$, $t \in T$. We denote by $\operatorname{cap}(S, T)$ the capacity of an s-t cut (S, T), i.e.,

$$\operatorname{cap}(S, T) = \sum \{\hat{c}_{uv} \mid (u, v) \in \hat{\mathcal{A}},\ u \in S,\ v \in T\}.$$

A minimum s-t cut is an s-t cut (S, T) minimizing the capacity $\operatorname{cap}(S, T)$. If we obtain a minimum s-t cut (S, T), then we set $\boldsymbol{y} = \boldsymbol{x} + \chi_{X^+}$, where $X^+ = T \cap \mathcal{V}$. As we will show later, the output feasible solution $\boldsymbol{y}$ is the same as in the procedure UP of the primal algorithm, i.e., $X^+ \in \arg\min\{E(\boldsymbol{x} + \chi_X) \mid X \subseteq \mathcal{V}\}$.

MAXFLOW-DOWN This operation is the same as MAXFLOW-UP, except for the definition of the arc set $\hat{\mathcal{A}}$ and the update of $\boldsymbol{x}$. The arc set $\hat{\mathcal{A}}$ is given by

$$\hat{\mathcal{A}} = \mathcal{A} \cup \{(s, u), (u, s) \mid u \in \mathcal{V}^-(\boldsymbol{x})\} \cup \{(u, t), (t, u) \mid u \in \mathcal{V} \setminus \mathcal{V}^-(\boldsymbol{x})\},$$

Capacities $\hat{c}_{uv}$ are defined by

$$\begin{array}{lll} \hat{c}_{uv} = \operatorname{grad} V_{uv}(x_v - x_u + 0), & \hat{c}_{vu} = -\operatorname{grad} V_{uv}(x_v - x_u - 0) & \text{for } (u, v) \in \mathcal{E}, \\ \hat{c}_{su} = \operatorname{grad} D_u(x_u - 0), & \hat{c}_{us} = 0 & \text{for } u \in \mathcal{V}^-(\boldsymbol{x}), \\ \hat{c}_{ut} = -\operatorname{grad} D_u(x_u - 0), & \hat{c}_{tu} = 0 & \text{for } u \in \mathcal{V} \setminus \mathcal{V}^-(\boldsymbol{x}). \end{array}$$

A new feasible solution $\boldsymbol{y}$ is computed from a minimum s-t cut (S, T) of the graph $\hat{\mathcal{G}}$ by $\boldsymbol{y} = \boldsymbol{x} - \chi_{X^-}$, where $X^- = S \cap \mathcal{V}$.

DIJKSTRA-UP This operation is optional. It does not affect the worst-case complexity of the algorithm, but may improve empirical performance. In this procedure we fix flow and compute a maximal feasible solution $\boldsymbol{y} \geq \boldsymbol{x}$ such that functions $D_u(\cdot)$ are non-increasing on $[x_u, y_u]$ and the condition (4.3b) holds. If we denote $d_u = y_u - x_u \geq 0$, then these constraints are equivalent to

$$\begin{aligned} d_u &\leq d_u^{\max} = \max\{d \in \mathbb{Z} \mid D_u(x_u + d) \leq \ldots \leq D_u(x_u + 1) \leq D_u(x_u)\}, \\ d_v - d_u &\leq d_{uv}^{\max} = \max\{d \in \mathbb{Z} \mid V_{uv}(x_v - x_u + d) = V_{uv}(x_v - x_u)\}, \\ d_u - d_v &\leq d_{vu}^{\max} = \max\{d \in \mathbb{Z} \mid V_{uv}(x_v - x_u - d) = V_{uv}(x_v - x_u)\}. \end{aligned}$$

It is well known (see, e.g., [3]) that finding a maximal vector $\boldsymbol{d}$ satisfying these constraints can be reduced to a single-source shortest path problem and such a vector $\boldsymbol{d}$ can be computed efficiently by using Dijkstra's algorithm.

DIJKSTRA-DOWN This operation is similar to the previous one; we compute a minimal feasible solution $\boldsymbol{y} \leq \boldsymbol{x}$ such that functions $D_u(\cdot)$ are non-decreasing on $[y_u, x_u]$ and the condition (4.3b) holds. If we denote $d_u = x_u - y_u \geq 0$, then these constraints are equivalent to

$$\begin{aligned} d_u &\leq d_u^{\max} = \max\{d \in \mathbb{Z} \mid D_u(x_u - d) \leq \ldots \leq D_u(x_u - 1) \leq D_u(x_u)\}, \\ d_v - d_u &\leq d_{uv}^{\max} = \max\{d \in \mathbb{Z} \mid V_{uv}(x_v - x_u - d) = V_{uv}(x_v - x_u)\}, \\ d_u - d_v &\leq d_{vu}^{\max} = \max\{d \in \mathbb{Z} \mid V_{uv}(x_v - x_u + d) = V_{uv}(x_v - x_u)\}. \end{aligned}$$

As before, a maximal vector $\boldsymbol{d}$ (corresponding to a minimal feasible solution $\boldsymbol{y}$) can be computed using Dijkstra's algorithm.

4.3 Analysis of the algorithm

First we analyze the behavior of the algorithm without procedures DIJKSTRA-UP and DIJKSTRA-DOWN. In the theorem below we assume that the input pair $(\boldsymbol{x}, 0)$ satisfies the condition (4.3b).

Theorem 4.2.
1. *Let* $(\boldsymbol{y}, f)$ *be the output of* MAXFLOW-UP *applied to* $(\boldsymbol{x}, 0)$. *Then,*
(a) *The condition* (4.3b) *holds for* $(\boldsymbol{y}, f)$.
(b) *Any minimum s-t cut* (S, T) *of* $\hat{\mathcal{G}}$ *satisfies* $T \cap \mathcal{V} \in \arg\min\{E(\boldsymbol{x} + \chi_X) \mid X \subseteq \mathcal{V}\}$.
(c) *There holds* $\mathcal{V}^+(\boldsymbol{y}, f) \subseteq \mathcal{V}^+(\boldsymbol{x})$ *and* $\mathcal{V}^-(\boldsymbol{y}, f) \subseteq \mathcal{V}^-(\boldsymbol{x})$.
(d) *If* $\rho^+(\boldsymbol{x}) = 0$, *then* $\mathcal{V}^+(\boldsymbol{y}, f) = \emptyset$.
2. *Let* $(\boldsymbol{y}, f)$ *be the output of* MAXFLOW-DOWN *applied to* $(\boldsymbol{x}, 0)$. *Then,*
(a) *The condition* (4.3b) *holds for* $(\boldsymbol{y}, f)$.
(b) *Any minimum s-t cut* (S, T) *of* $\hat{\mathcal{G}}$ *satisfies* $S \cap \mathcal{V} \in \arg\min\{E(\boldsymbol{x} - \chi_X) \mid X \subseteq \mathcal{V}\}$.
(c) *There holds* $\mathcal{V}^+(\boldsymbol{y}, f) \subseteq \mathcal{V}^+(\boldsymbol{x})$ *and* $\mathcal{V}^-(\boldsymbol{y}, f) \subseteq \mathcal{V}^-(\boldsymbol{x})$.
(d) *If* $\rho^-(\boldsymbol{x}) = 0$, *then* $\mathcal{V}^-(\boldsymbol{y}, f) = \emptyset$.

Combining Proposition 3.2 and Theorem 4.2, we can show that the algorithm terminates in at most $2K_\infty + 2$ iterations and yields an optimal primal-dual pair $(\boldsymbol{x}, f)$ upon termination. Indeed, 1 (a) and 2 (a) of Theorem 4.2 imply that the condition (4.3b) always holds. After at most K_∞ iterations of procedure UP, the quantity $\rho^+(\boldsymbol{x})$ becomes zero, and therefore after

at most $K_\infty + 1$ iterations the set $\mathcal{V}^+(\boldsymbol{x}, f)$ becomes empty. At this point flag SuccessUp is set to true, and set $\mathcal{V}^+(\boldsymbol{x}, f)$ will remain empty. Similar argumentation holds for procedure DOWN. When the algorithm terminates, both of the sets $\mathcal{V}^+(\boldsymbol{x}, f)$ and $\mathcal{V}^-(\boldsymbol{x}, f)$ are empty, so the optimality condition (4.3) holds for the pair $(\boldsymbol{x}, f)$.

This analysis remains valid even with procedures DIJKSTRA-UP or DIJKSTRA-DOWN, as shown below.

Theorem 4.3. *Let $\boldsymbol{y}$ be the output of* DIJKSTRA-UP *or* DIJKSTRA-DOWN *applied to $(\boldsymbol{x}, 0)$. Then,*
(a) *The condition* (4.3b) *holds for $(\boldsymbol{y}, 0)$.*
(b) *There holds $\mathcal{V}^+(\boldsymbol{y}) \subseteq \mathcal{V}^+(\boldsymbol{x})$ and $\mathcal{V}^-(\boldsymbol{y}) \subseteq \mathcal{V}^-(\boldsymbol{x})$.*
(c) *There holds $\rho^+(\boldsymbol{y}) \leq \rho^+(\boldsymbol{x})$ and $\rho^-(\boldsymbol{y}) \leq \rho^-(\boldsymbol{x})$.*

It can be seen that if the procedure DIJKSTRA-UP is applied to an optimal pair $(\boldsymbol{x}, f)$ then the output $\boldsymbol{y}$ is the maximal optimal solution. Indeed, according to Theorem 4.1 a feasible solution $\boldsymbol{y}$ is optimal if and only if it satisfies

$$\begin{aligned} D_u^f(y_u) &= D_u^f(x_u) && (\forall\, u \in \mathcal{V}), \\ V_{uv}^f(y_v - y_u) &= V_{uv}^f(x_v - x_u) && (\forall\, (u,v) \in \mathcal{E}). \end{aligned}$$

For feasible solutions $\boldsymbol{y} \geq \boldsymbol{x}$ this is equivalent to saying that functions $D_u(\cdot)$ are non-increasing on $[x_u, y_u]$ and the condition (4.3b) holds. By construction, DIJKSTRA-UP finds the maximal feasible solution satisfying these conditions. Similarly, we can show that applying DIJKSTRA-DOWN to an optimal pair $(\boldsymbol{x}, f)$ yields the minimal optimal solution.

The following theorem allows to simplify slightly the algorithm's implementation.

Theorem 4.4.
1. *Let $(\boldsymbol{y}, f)$ be the output of* MAXFLOW-UP *applied to $(\boldsymbol{x}, 0)$. Then, applying* DIJKSTRA-UP *to $(\boldsymbol{x}, f)$ and to $(\boldsymbol{y}, f)$ would yield the same feasible solution $\boldsymbol{z}$.*
2. *Let $(\boldsymbol{y}, f)$ be the output of* MAXFLOW-DOWN *applied to $(\boldsymbol{x}, 0)$. Then, applying* DIJKSTRA-DOWN *to $(\boldsymbol{x}, f)$ and to $(\boldsymbol{y}, f)$ would yield the same feasible solution $\boldsymbol{z}$.*

Thus, if DIJKSTRA-UP is applied immediately after MAXFLOW-UP then it is not necessary to update variables $\boldsymbol{x}$ in MAXFLOW-UP (and similarly for DOWN); that is, MAXFLOW-UP updates only dual variables f and then DIJKSTRA-UP updates only primal variables $\boldsymbol{x}$ in this implementation.

We now turn to the proof of Theorems 4.2, 4.3, and 4.4. We omit proofs of part 2 of Theorems 4.2 and 4.4 since they are very similar to those of part 1.

Proof of Theorem 4.2, part 1(a). From the capacity constraints we get $f_{uv} \leq \operatorname{grad} V_{uv}(x_v - x_u + 0)$, $-f_{uv} = f_{vu} \leq -\operatorname{grad} V_{uv}(x_v - x_u - 0)$. Therefore, we have

$$\begin{aligned} \operatorname{grad} V_{uv}^f(x_v - x_u + 0) &= \operatorname{grad} V_{uv}(x_v - x_u + 0) - f_{uv} \geq 0, \\ \operatorname{grad} V_{uv}^f(x_v - x_u - 0) &= \operatorname{grad} V_{uv}(x_v - x_u - 0) - f_{uv} \leq 0, \end{aligned}$$

which implies that $V_{uv}^f(x_v - x_u) = \min_{\alpha \in \mathbb{Z}} V_{uv}^f(\alpha)$. Thus, if $y_v - y_u = x_v - x_u$, then the condition (4.3b) holds for edge (u, v). Let us consider the case $y_v - y_u = x_v - x_u + 1$. This can

only happen when $u \in S$ and $v \in T$, which means that edge (u, v) must be saturated. Therefore, we have $f_{uv} = \hat{f}_{uv} = \hat{c}_{uv} = \operatorname{grad} V_{uv}(x_v - x_u + 0)$, implying $\operatorname{grad} V^f_{uv}(x_v - x_u + 0) = 0$. Hence, we have

$$V^f_{uv}(y_v - y_u) = V^f_{uv}(x_v - x_u + 1) = V^f_{uv}(x_v - x_u) = \min_{\alpha \in \mathbb{Z}} V^f_{uv}(\alpha).$$

The case $y_v - y_u = x_v - x_u - 1$ can be considered similarly. □

Proof of Theorem 4.2, part 1(b). Let (S, T) be a s-t cut of the graph $\hat{\mathcal{G}}$, and put $\boldsymbol{y} = \boldsymbol{x} + \chi_{T \cap \mathcal{V}}$. Then, we have

$$\begin{aligned}
\operatorname{cap}(S,T) &= \sum\{\hat{c}_{su} \mid u \in (T \cap \mathcal{V}) \setminus \mathcal{V}^+(\boldsymbol{x})\} + \sum\{\hat{c}_{ut} \mid u \in (S \cap \mathcal{V}) \cap \mathcal{V}^+(\boldsymbol{x})\} \\
&\quad + \sum\{\hat{c}_{uv} \mid (u,v) \in \mathcal{E},\ u \in S,\ v \in T\} + \sum\{\hat{c}_{vu} \mid (u,v) \in \mathcal{E},\ u \in T,\ v \in S\} \\
&= \sum\{\operatorname{grad} D_u(x_u + 0) \mid u \in T \cap \mathcal{V}\} - \sum\{\operatorname{grad} D_u(x_u + 0) \mid u \in \mathcal{V}^+(\boldsymbol{x})\} \\
&\quad + \sum\{\operatorname{grad} V_{uv}(x_v - x_u + 0) \mid (u,v) \in \mathcal{E},\ u \in S,\ v \in T\} \\
&\quad - \sum\{\operatorname{grad} V_{uv}(x_v - x_u - 0) \mid (u,v) \in \mathcal{E},\ u \in T,\ v \in S\} \\
&= E(\boldsymbol{y}) - E(\boldsymbol{x}) - \sum\{\operatorname{grad} D_u(x_u + 0) \mid u \in \mathcal{V}^+(\boldsymbol{x})\}.
\end{aligned}$$

This equation shows that (S, T) is a minimum s-t cut if and only if $T \cap \mathcal{V} \in \arg\min\{E(\boldsymbol{x} + \chi_X) \mid X \subseteq \mathcal{V}\}$. □

Proof of Theorem 4.2, part 1(c). We consider two possible cases.

[Case 1: $u \in \mathcal{V}^+(\boldsymbol{x})$, i.e. $\operatorname{grad} D_u(x_u + 0) < 0$] We need to show that $u \notin \mathcal{V}^-(\boldsymbol{y}, f)$. This holds since

$$\begin{aligned}
\operatorname{grad} D^f_u(y_u - 0) &\le \operatorname{grad} D^f_u((x_u + 1) - 0) \\
&= \operatorname{grad} D^f_u(x_u + 0) = \operatorname{grad} D_u(x_u + 0) - f_u \le 0,
\end{aligned}$$

where the first inequality follows from $y_u \le x_u + 1$ and convexity of $D^f(\cdot)$, and the last inequality follows since $-f_u = \hat{f}_{ut} \le \hat{c}_{ut} = -\operatorname{grad} D_u(x_u + 0)$.

[Case 2: $u \notin \mathcal{V}^+(\boldsymbol{x})$, i.e. $\operatorname{grad} D_u(x_u + 0) \ge 0$] We have $0 \le f_u = \hat{f}_{su} \le \operatorname{grad} D_u(x_u + 0)$. The fact that $u \notin \mathcal{V}^+(\boldsymbol{y}, f)$ follows from

$$\operatorname{grad} D^f_u(y_u + 0) \ge \operatorname{grad} D^f_u(x_u + 0) = \operatorname{grad} D_u(x_u + 0) - f_u \ge 0.$$

Now suppose that $u \notin \mathcal{V}^-(\boldsymbol{x})$, i.e., $\operatorname{grad} D_u(x_u - 0) \le 0$. We need to show that $u \notin \mathcal{V}^-(\boldsymbol{y}, f)$. If $y_u = x_u$, then this follows from

$$\operatorname{grad} D^f_u(y_u - 0) = \operatorname{grad} D_u(x_u - 0) - f_u \le 0.$$

If $y_u = x_u + 1$, then $u \in T$, implying that the edge (s, u) must be saturated, i.e., $f_u = \hat{f}_{su} = \operatorname{grad} D_u(x_u + 0)$. Therefore, we have

$$\operatorname{grad} D^f_u(y_u - 0) = \operatorname{grad} D^f_u(x_u + 0) = \operatorname{grad} D_u(x_u + 0) - f_u = 0.$$

□

Proof of Theorem 4.2, part 1(d). We show that $u \notin \mathcal{V}^+(\boldsymbol{y}, f)$ for all $u \in \mathcal{V}$. For nodes $u \notin \mathcal{V}^+(\boldsymbol{x})$ this follows from part (c). Let us consider a node $u \in \mathcal{V}^+(\boldsymbol{x})$. The condition $\rho^+(\boldsymbol{x}) = 0$ means that $\emptyset \in \arg\min\{E(\boldsymbol{x} + \chi_X) \mid X \subseteq \mathcal{V}\}$. Therefore, according to part 1(b), the cut $(\mathcal{V} \cup \{s\}, \{t\})$ is a minimum s-t cut of the graph $\hat{\mathcal{G}}$. Thus, the edge (u, t) must be saturated, i.e., $f_u = -\hat{f}_{ut} = -\hat{c}_{ut} = \operatorname{grad} D_u(x_u + 0)$. This implies that

$$\operatorname{grad} D_u^f(y_u + 0) \geq \operatorname{grad} D_u^f(x_u + 0) = \operatorname{grad} D_u(x_u + 0) - f_u = 0,$$

implying $u \notin \mathcal{V}^+(\boldsymbol{y}, f)$, as desired. □

Proof of Theorem 4.3. We consider only the procedure DIJKSTRA-UP; the proof for the procedure DIJKSTRA-DOWN is completely analogous. The statements (a) and (b) follow directly from the definition of $\boldsymbol{y}$. The definition of $\boldsymbol{y}$ also implies that $E(\boldsymbol{y}) = \min\{E(\boldsymbol{z}) \mid \boldsymbol{x} \leq \boldsymbol{z} \leq \boldsymbol{y}\}$. Hence, we have (c) by Lemma 3.4. □

Proof of Theorem 4.4, part 1. Let us show that (i) $D_u^f(y_u) \leq D_u^f(x_u)$ for all nodes u, and (ii) $V_{uv}^f(y_v - y_u) = V_{uv}^f(x_v - x_u)$ for all edges (u, v). The theorem will then follow from the description of DIJKSTRA-UP.

If $y_u = x_u$ for node u then the fact (i) is trivial. Suppose that $y_u = x_u + 1$; we need to show that $\operatorname{grad} D_u^f(y_u - 0) = \operatorname{grad} D_u^f(x_u + 0) \leq 0$. If $u \in \mathcal{V}^+(\boldsymbol{x})$ then this follows since $u \notin \mathcal{V}^-(\boldsymbol{y}, f)$ by Theorem 4.2, part 1(c). If $u \notin \mathcal{V}^+(\boldsymbol{x})$, then $f_u = \hat{f}_{su} = \hat{c}_{su} = \operatorname{grad} D_u(x_u + 0)$ since $u \in T$ and edge (s, u) is saturated. Therefore, $\operatorname{grad} D_u^f(x_u + 0) = \operatorname{grad} D_u(x_u + 0) - f_u = 0$.

Finally, the fact (ii) was shown earlier (see the proof of Theorem 4.2, part 1(a)). □

5 Experiments on the Panoramic Image Stitching Problem

We tested the speed of several algorithms on the panoramic image stitching application. Given two input images I^1 and I^2 defined on overlapping domains $\mathcal{V}^1$ and $\mathcal{V}^2$, the goal of the panoramic image stitching is to compute an output image without a visible seam. Levin et al. [24, 32] proposed several techniques for this problem. One of them, GIST1 algorithm under l_1 norm, was shown to outperform many other stitching methods. It involves minimizing the following function for each color channel:

$$E(\boldsymbol{x}) = \sum_{(u,v) \in \mathcal{E}} w_{uv}^1 |(x_v - x_u) - (I_v^1 - I_u^1)| + w_{uv}^2 |(x_v - x_u) - (I_v^2 - I_u^2)|,$$

where $(u, v) \in \mathcal{E}$ if and only if u and v are neighboring pixels. In other words, we want the gradient of image $\boldsymbol{x}$ to match gradients of images I^1 and I^2. Weights w_{uv}^1 and w_{uv}^2 were determined as follows. For edges whose endpoints belong to $\mathcal{V}^1 \setminus \mathcal{V}^2$ we set $w_{uv}^1 = 2, w_{uv}^2 = 0$. For edges in $\mathcal{V}^2 \setminus \mathcal{V}^1$ we set $w_{uv}^1 = 0, w_{uv}^2 = 2$. For all other edges we set $w_{uv}^1 = w_{uv}^2 = 1$.

It is easy to see that an optimal solution for the minimization of the function E is determined only up to an additive constant. Similar to [24, 32], we computed this constant so that median intensity of I^1 in $\mathcal{V}^1$ matches that of the output image. This does not uniquely determines the solution, however, since there may be multiple optimal solutions $\boldsymbol{x}$ satisfying this requirement. Levin et al. do not discuss how to choose between them. We propose the following technique. We put constraints $x_u \in [0, K - 1]$ on the variables, where K is sufficiently large (e.g., 512).

Figure 6: Results of panoramic stitching. First two columns: input images (courtesy of A. Zomet). Rectangles show the area of overlap. Last three columns: results corresponding to $\boldsymbol{x}^{\min}$, $\boldsymbol{x}^{\text{av}}$, and $\boldsymbol{x}^{\max}$, respectively (note that images are cropped). The additive constant is chosen as described in the text.

We then compute the minimal optimal solution $\boldsymbol{x}^{\min}$, the maximal optimal solution $\boldsymbol{x}^{\max}$, and their average $\boldsymbol{x}^{\text{av}} = \lfloor(\boldsymbol{x}^{\min} + \boldsymbol{x}^{\max})/2\rfloor$ which is also an optimal solution by Proposition 2.5. Furthermore, these optimal solutions have the minimum possible range defined as $\max_u\{x_u\} - \min_u\{x_u\} + 1$. In our experiments it was very close to 256. Having a small range may be advantageous since intensities must be mapped to interval [0,255]; if the range is too large then some regions may become too dark or saturated.

Fig. 6 shows panoramas corresponding to feasible solutions $\boldsymbol{x}^{\min}$, $\boldsymbol{x}^{\text{av}}$, and $\boldsymbol{x}^{\max}$. It can be seen that the solution $\boldsymbol{x}^{\text{av}}$ looks significantly better than the other two. The overlap area is too dark in $\boldsymbol{x}^{\min}$ and too bright in $\boldsymbol{x}^{\max}$.

Algorithms tested We compared the speed of three different algorithms. The first two are the primal-dual method without Dijkstra and with Dijkstra computations. Procedure DOWN is applied only after SuccessUp becomes true. We used the max flow algorithm of Boykov and Kolmogorov [6] available at http://www.cs.cornell.edu/People/vnk/software.html (version 3.0).

The third technique that we tried is as follows. We converted the original problem to the linear cost network flow problem. (Note that we did not enforce constraints $x_u \in [0, K-1]$). We then applied an algorithm of Goldberg [15] available at http://www.avglab.com/andrew/soft.html (version 4.0). It has one free parameter, namely scaling factor; we set it to 32 (results for other factors were faster by at most one percent). The problem (DCCF) can be converted to the linear cost network flow problem in many different ways. We used a transformation with the following property: if the initial feasible solution satisfied the optimality condition, then so did the resulting linear cost network flow problem. In all codes we used 32-bit integers.

We note that we did not test the cost scaling algorithm of Ahuja et al. [1]. It uses ideas similar to [15], but it works with a smaller graph. Therefore, [1] could potentially be faster than

converting the problem to a linear cost network flow problem and then applying the algorithm in [15]. In our application, however, graph sizes would differ only slightly, and we argue that direct implementation of the technique in [1] is unlikely to beat the implementation in [15]. Indeed, the latter is highly optimized and includes many heuristics which significantly improve the empirical performance.

Initialization and two-stage procedure Algorithms were initialized with the following feasible solution $\boldsymbol{x}^\circ$: $x_u^\circ = I_u^1$ in region $\mathcal{V}^1 \setminus \mathcal{V}^2$, $x_u^\circ = I_u^2$ in $\mathcal{V}^2 \setminus \mathcal{V}^1$, and $x_u^\circ = \lfloor (I_u^1 + I_u^2)/2 \rfloor$ in $\mathcal{V}^1 \cap \mathcal{V}^2$. Besides applying an algorithm directly to the original problem, we also tested the following two-stage procedure. First we solve the problem for a subgraph induced by subset $\mathcal{V}'$ obtained by the erosion of the set $\mathcal{V}^1 \cap \mathcal{V}^2$ by one pixel. In other words, we fix nodes in $\mathcal{V} \setminus \mathcal{V}'$ by adding terms $C|x_u - x_u^\circ|$ to the objective function for nodes $u \in \mathcal{V} \setminus \mathcal{V}'$, where C is a sufficiently large constant. (In implementation nodes which are not connected to nodes in $\mathcal{V}'$ can be safely omitted.) Then we apply the algorithm to the whole problem using the optimal solution and the flow obtained in the first stage as an initialization.

Experiments We used three datasets D0, D1, and D2 shown in Fig. 6. Their dimensions are 449×193 for D0 and 577×257 for D1 and D2. The percentages of overlap area are 4.9%, 10.0% and 6.9%, respectively. We also used scaled-down datasets D0-s, D1-s and D2-s (both X and Y dimensions are reduced by 2 times). Note that results for scaled-down images visually look worse.

The table below shows running times in seconds (we measure the total time for 3 color channels). The tests were performed on a machine with Intel Celeron 1.4GHz processor in Microsoft Windows XP environment, using Microsoft Visual Studio 7.0 C++ compiler.

	D0-s	D1-s	D2-s	D0	D1	D2
primal-dual, no Dijkstra, 1 stage	14.39	28.63	32.47	68.31	164.52	178.21
primal-dual, no Dijkstra, 2 stages	4.02	16.33	17.63	25.43	111.78	127.67
primal-dual with Dijkstra, 1 stage	3.02	6.88	8.71	16.53	28.76	51.26
primal-dual with Dijkstra, 2 stages	0.58	0.81	0.93	2.20	3.57	3.61
linear cost network flow, 1 stage	2.48	3.68	2.39	9.22	21.69	21.17
linear cost network flow, 2 stages	1.01	1.50	2.16	11.39	21.17	18.23

The two-stage procedure with the primal-dual algorithm with Dijkstra computations is a clear winner.

References

[1] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin. Solving the convex cost integer dual network flow problem. *Management Science*, 49:7:950–964, 2003.

[2] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin. A cut based algorithm for the convex dual of the minimum cost network flow problem. *Algorithmica*, 39:3:189–208, April 2004.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[4] J. Bioucas-Dias and G. Valadão. Phase unwrapping via graph cuts. In *Pattern Recognition and Image Analysis: 2nd Iberian Conference (IbPRIA)*, LNCS. Springer, June 2005.

[5] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, November 2002.

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9), September 2004.

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11), November 2001.

[8] A. Chambolle. Total variation minimization and a class of binary MRF models. In *5th International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, LNCS 3757. June 2005, 136–152.

[9] J. Darbon and M. Sigelle. A fast and exact algorithm for total variation minimization. In *Pattern Recognition and Image Analysis: 2nd Iberian Conference (IbPRIA)*, LNCS. Springer, June 2005.

[10] P. Favati and F. Tardella. Convexity in nonlinear integer programming. Ricerca Operativa 53:3–44, 1990.

[11] L. R. Ford and D. R. Fulkerson. A primal-dual algorithm for the capacitated hitchcock problem. *Naval Research Logistics Quarterly*, 4:47–54, 1957.

[12] L. R. Ford and D. R. Fulkerson. *Flows in networks*. Princeton Univ. Press, 1962.

[13] A. Fujishige and K. Murota. Notes on L-/M-convex functions and the separation theorems. *Math. Programming*, 88:129–146, 2000.

[14] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[15] A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *J. Algorithms*, 22:1–29, 1997.

[16] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. *J. ACM*, 35:921–940, 1988.

[17] D. S. Hochbaum. An efficient algorithm for image segmentation, Markov Random Fields and related problems. *J. ACM*, 48:2:686–701, July 2001.

[18] D. S. Hochbaum and J. G. Shanthikumar. Convex separable optimization is not much harder than linear optimization. *J. ACM*, 37:843–862, 1990.

[19] M. Iri. *Network Flow, Transportation and Scheduling—Theory and Algorithms*. Academic Press, 1969.

[20] H. Ishikawa. Exact optimization for Markov Random Fields with convex priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, October 2003.

[21] A. B. Karzanov and S. T. McCormick. Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM J. Computing*, 4:1245–1275, 1997.

[22] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov Random Fields. *Foundations of Computer Science*, pages 14–23, 1999.

[23] N. Komodakis and G. Tziritas. A new framework for approximate labeling via graph cuts. In *Intl. Conf. on Computer Vision*, October 2005.

[24] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *European Conf. on Computer Vision*, May 2004.

[25] K. Murota. Discrete convex analysis. *Math. Programming* 83:313–371, 1998.

[26] K. Murota. Algorithms in discrete convex analysis. *IEICE Transactions on Systems and Information*, E83-D:344–352, 2000.

[27] K. Murota. *Discrete Convex Analysis*. SIAM Monographs on Discrete Mathematics and Applications, Vol. 10, 2003.

[28] K. Murota. On steepest descent algorithms for discrete convex functions. *SIAM J. Optimization*, 14(3):699–707, 2003.

[29] R. T. Rockafellar. *Convex Analysis*. Princeton Univ. Press, 1970.

[30] O. Veksler. Efficient graph-based energy minimization methods in computer vision. PhD thesis, Cornell University, Dept. of Computer Science, Ithaca, NY (1999)

[31] B. Zalesky. Network flow optimization for restoration of images. *J. of Applied Mathematics*, 2(4):199–218, 2002.

[32] A. Zomet, A. Levin, S. Peleg, and Y. Weiss. Seamless image stitching by minimizing false edges. *IEEE Trans. Image Processing*, 15(4):969–977, 2006.