Noname manuscript No.
(will be inserted by the editor)

Dijkstra's algorithm and L-concave function maximization

Kazuo Murota · Akiyoshi Shioura

Received: date / Accepted: date

Abstract Dijkstra's algorithm is a well-known algorithm for the single-source shortest path problem in a directed graph with nonnegative edge length. We discuss Dijkstra's algorithm from the viewpoint of discrete convex analysis, where the concept of discrete convexity called L-convexity plays a central role. We observe first that the dual of the linear programming (LP) formulation of the shortest path problem can be seen as a special case of L-concave function maximization. We then point out that the steepest ascent algorithm for L-concave function maximization, when applied to the LP dual of the shortest path problem and implemented with some auxiliary variables, coincides exactly with Dijkstra's algorithm.

Keywords shortest path problem · Dijkstra's algorithm · discrete concave function · steepest ascent algorithm

Mathematics Subject Classification (2010) 90C27 · 68Q25

1 Introduction

The single-source shortest path problem in a directed graph with nonnegative edge length is a classical combinatorial optimization problem formulated as follows: given a directed graph $G = (V, E)$ with edge length $\ell(e) \geq 0$ ($e \in E$) and a vertex $s \in V$ called a source, we want to compute the shortest path

Kazuo Murota
Graduate School of Information Science and Technology
University of Tokyo, Tokyo 113-8656, Japan
E-mail: murota@mist.i.u-tokyo.ac.jp

Akiyoshi Shioura
Graduate School of Information Sciences
Tohoku University, Sendai 980-8579, Japan
Tel./Fax: +81-22-795-4753
E-mail: shioura@dais.is.tohoku.ac.jp

length from the source vertex s to each vertex $v \in V$. Among many algorithms for the shortest path problem, Dijkstra's algorithm [3] described below is most fundamental (see, e.g., [1,16]).

Dijkstra's Algorithm

Step 0: Set $U := V$. Set $\pi(s) := 0$, $\pi(v) := +\infty$ $(v \in V \setminus \{s\})$.

Step 1: Set $W := \arg\min\{\pi(v) \mid v \in U\}$ and $X := U \setminus W$.

Step 2: If $X = \emptyset$, then stop; for $v \in V$, $\pi(v)$ is the shortest path length from s to v.

Step 3: For $v \in X$, set

$$\pi(v) := \min\left[\pi(v), \min\{\pi(u) + \ell(u,v) \mid (u,v) \in E,\ u \in W\}\right].$$

Set $U := X$. Go to Step 1. □

In this note, we discuss Dijkstra's algorithm from the viewpoint of discrete convex analysis. Discrete convex analysis is a theoretical framework for well-solved combinatorial optimization problems introduced by Murota (see [10]; see also [12]), where the concept of discrete convexity called L$^\natural$-convexity plays a central role. We observe first that the dual of the linear programming (LP) formulation of the shortest path problem can be seen as a special case of L$^\natural$-concave function maximization (see Section 2 for the definition of L$^\natural$-concavity). We then point out that the steepest ascent algorithm for L$^\natural$-concave function maximization, when applied to the LP dual of the shortest path problem and implemented with some auxiliary variables, coincides exactly with Dijkstra's algorithm.

Besides Dijkstra's algorithm there are some other existing algorithms that can be recognized as a special case of the steepest ascent algorithm for L$^\natural$-concave function maximization. We explain such connection to the dual algorithm of Hassin [6] for the minimum cost flow problem in Section 5.1, and to the dual algorithm of Chung and Tcha [2] for the minimum cost submodular flow problem in Section 5.2. An application to computer vision, where the steepest descent algorithm for L$^\natural$-convex function minimization is used for the panoramic image stitching problem, can be found in Kolmogorov and Shioura [9].

2 Review of L$^\natural$-convexity

In this section we review the concepts of L$^\natural$-convex sets and L$^\natural$-concave functions, and present some useful properties. See [12] for more account of these concepts.

2.1 L$^\natural$-convex Sets

Let V be a finite set. A set $S \subseteq \mathbb{Z}^V$ of integral vectors is said to be *L$^\natural$-convex* if it is nonempty and satisfies the following condition:

$$\text{if } p, q \in S, \text{ then } \left\lceil \frac{p+q}{2} \right\rceil, \left\lfloor \frac{p+q}{2} \right\rfloor \in S, \tag{1}$$

where for $x \in \mathbb{R}^V$, $\lceil x \rceil$ and $\lfloor x \rfloor$ denote, respectively, the integer vectors obtained from x by component-wise round-up and round-down to the nearest integers. The condition (1) is called the *discrete midpoint convexity* for a set.

Discrete midpoint convexity (1) implies the following property. For $p, q \in \mathbb{R}^V$, we denote by $p \vee q$ and $p \wedge q$, respectively, the vectors of component-wise maximum and minimum of p and q, i.e.,

$$(p \vee q)(v) = \max\{p(v), q(v)\}, \quad (p \wedge q)(v) = \min\{p(v), q(v)\} \quad (v \in V).$$

Proposition 1 (cf. [12, Chapter 5]) *Let $S \subseteq \mathbb{Z}^V$ be an L$^\natural$-convex set. If $p, q \in S$, then it holds that $p \vee q, p \wedge q \in S$.*

This property implies, in particular, that a maximal vector in a bounded L$^\natural$-convex set is uniquely determined.

The following proposition gives a polyhedral description of L$^\natural$-convex sets.

Proposition 2 (cf. [12, Chapter 5]) *A set $S \subseteq \mathbb{Z}^V$ is an L$^\natural$-convex set if and only if S is a nonempty set represented as*

$$S = \{p \in \mathbb{Z}^V \mid p(v) - p(u) \le a(u,v) \ (u, v \in V, u \neq v), \\ b(v) \le p(v) \le c(v) \ (v \in V)\}$$

with some $a(u,v) \in \mathbb{Z} \cup \{+\infty\}$ $(u, v \in V,\ u \neq v)$, $b(v) \in \mathbb{Z} \cup \{-\infty\}$ $(v \in V)$, and $c(v) \in \mathbb{Z} \cup \{+\infty\}$ $(v \in V)$.

2.2 L$^\natural$-concave Functions

Let $g : \mathbb{Z}^V \to \mathbb{R} \cup \{-\infty\}$ be a function defined on the integer lattice points, and denote $\operatorname{dom} g = \{p \in \mathbb{Z}^V \mid g(p) > -\infty\}$. We say that g is an *L$^\natural$-concave function* if $\operatorname{dom} g \neq \emptyset$ and it satisfies the following condition:

$$g(p) + g(q) \le g\left(\left\lceil \frac{p+q}{2} \right\rceil\right) + g\left(\left\lfloor \frac{p+q}{2} \right\rfloor\right) \qquad (\forall p, q \in \operatorname{dom} g).$$

In the maximization of an L$^\natural$-concave function $g : \mathbb{Z}^V \to \mathbb{R} \cup \{-\infty\}$, a maximizer of g can be characterized by a local optimality. For $X \subseteq V$, we denote by $\chi_X \in \{0, +1\}^V$ the characteristic vector of X.

Theorem 1 *Let $g : \mathbb{Z}^V \to \mathbb{R} \cup \{-\infty\}$ be an L$^\natural$-concave function. A vector $p \in \operatorname{dom} g$ is a maximizer of g if and only if $g(p) \ge g(p + \chi_X)$ $(\forall X \subseteq V)$ and $g(p) \ge g(p - \chi_X)$ $(\forall X \subseteq V)$.*

A maximizer of g can be computed by the following steepest ascent algorithm. We suppose that an initial vector $p_0 \in \mathrm{dom}\, g$ is given in advance.

Algorithm 0

(Steepest Ascent Algorithm for L$^\natural$-concave Function Maximization)
Step 0: Set $p := p_0$.
Step 1: Find $\varepsilon \in \{+1, -1\}$ and $X \subseteq V$ that maximize $g(p + \varepsilon\chi_X)$.
Step 2: If $g(p) \geq g(p + \varepsilon\chi_X)$, then stop; p is a maximizer of g.
Step 3: Set $p := p + \varepsilon\chi_X$. Go to Step 1. $\square$

It is noted that Step 1 can be done in (strongly) polynomial time by using a polynomial-time algorithm for submodular set function minimization [7,15] since the set functions $\rho^+, \rho^- : 2^V \to \mathbb{R} \cup \{+\infty\}$ defined by

$$\rho^+(X) = g(p) - g(p + \chi_X), \quad \rho^-(X) = g(p) - g(p - \chi_X) \quad (X \subseteq V)$$

are submodular functions with $\rho^+(\emptyset) = \rho^-(\emptyset) = 0$.

The steepest ascent algorithm above terminates in a finite number of iterations if $\mathrm{dom}\, g$ is a finite set. The obtained vector p is indeed a maximizer of g by Theorem 1. See [9,13] for the time complexity of the algorithms of this type.

3 Shortest Path Problem and L$^\natural$-convexity

We show the connection of the single-source shortest path problem with L$^\natural$-convex sets and L$^\natural$-concave functions. We assume that edge length $\ell(e)$ is integer-valued for all $e \in E$, while the real-valued edge length is considered in Section 4.3.

A linear programming formulation of the single-source shortest path problem is given as follows:

$$\textbf{(P)} \left|\begin{array}{l}\text{Minimize } \displaystyle\sum_{(u,v)\in E} \ell(u,v)x(u,v) \\ \text{subject to } \sum\{x(u,s) \mid (u,s) \in E, u \in V\} \\ \qquad\qquad - \sum\{x(s,u) \mid (s,u) \in E,\ u \in V\} = -(n-1), \\ \qquad\qquad \sum\{x(u,v) \mid (u,v) \in E, u \in V\} \\ \qquad\qquad - \sum\{x(v,u) \mid (v,u) \in E,\ u \in V\} = 1 \quad (v \in V \setminus \{s\}), \\ \qquad\qquad x(u,v) \geq 0 \qquad ((u,v) \in E). \end{array}\right.$$

This LP can be seen as a minimum-cost flow problem, where a unit of flow is sent from the source vertex s to each vertex $v \in V \setminus \{s\}$, and the flow cost on edge $(u,v) \in E$ is given by $\ell(u,v)$.

The LP dual of (P) is given as follows:

$$\left|\begin{array}{l}\text{Maximize } \displaystyle\sum_{v\in V\setminus\{s\}} \{p(v) - p(s)\} \\ \text{subject to } p(v) - p(u) \leq \ell(u,v) \qquad ((u,v) \in E), \\ \qquad\qquad p(v) \in \mathbb{R} \qquad (v \in V). \end{array}\right.$$

In this LP, we can fix $p(s) = 0$ without loss of generality. Moreover, we may assume that $p(v)$ is integer-valued since edge length $\ell(e)$ is integer-valued (see, e,g., [16]). Then, we obtain the following problem:

$$
\textbf{(D)} \left|
\begin{array}{ll}
\text{Maximize} & \displaystyle\sum_{v \in V \setminus \{s\}} p(v) \\
\text{subject to} & p(v) - p(u) \le \ell(u,v) \qquad ((u,v) \in E), \\
& p(s) = 0, \\
& p(v) \in \mathbb{Z} \qquad (v \in V \setminus \{s\}).
\end{array}
\right.
$$

This problem will be the main object of our discussion.

We denote by $S \subseteq \mathbb{Z}^V$ the feasible region of (D), i.e.,

$$
S = \{p \in \mathbb{Z}^V \mid p(v) - p(u) \le \ell(u,v) \ ((u,v) \in E),\ p(s) = 0\}. \tag{2}
$$

By Proposition 2, S is an L$^\natural$-convex set. Hence, the problem (D) can be seen as maximization of a linear function with positive coefficients over an L$^\natural$-convex set.

We assume that there exists a directed path from s to every $v \in V \setminus \{s\}$. Then, (P) has a feasible (and optimal) solution, and the optimal value of (D) is finite. Hence, the set S is bounded from above, and Proposition 1 implies that S has a unique maximal vector p_*, which is an optimal solution of (D). It is also noted that the zero vector $\mathbf{0}$ is contained in S since $\ell(u,v) \ge 0$ for $(u,v) \in E$.

We define a function $g_{\mathrm{D}} : \mathbb{Z}^V \to \mathbb{R} \cup \{-\infty\}$ by

$$
g_{\mathrm{D}}(p) = \begin{cases} \displaystyle\sum_{v \in V \setminus \{s\}} p(v) & (\text{if } p \in S), \\ -\infty & (\text{otherwise}). \end{cases} \tag{3}
$$

We see that the maximization of g_{D} is equivalent to the problem (D). Since S satisfies the discrete midpoint convexity (1), g_{D} satisfies the inequality

$$
g_{\mathrm{D}}(p) + g_{\mathrm{D}}(q) \le g_{\mathrm{D}}\left(\left\lceil \frac{p+q}{2} \right\rceil\right) + g_{\mathrm{D}}\left(\left\lfloor \frac{p+q}{2} \right\rfloor\right)
$$

for all $p, q \in \mathrm{dom}\, g_{\mathrm{D}}$; in fact, the inequality above holds with equality. This means that g_{D} is an L$^\natural$-concave function. Hence, the problem (D) can be seen as a special case of L$^\natural$-concave function maximization.

4 Dijkstra's Algorithm and Steepest Ascent Algorithm

4.1 Steepest Ascent Algorithm Applied to Shortest Path Problem

We apply the steepest ascent algorithm (Algorithm 0) in Section 2 to the maximization of the L$^\natural$-concave function g_{D} in (3) associated with the shortest path problem, where the zero vector $\mathbf{0} \in S$ is used as the initial vector p_0. Then, we observe the following properties.

Proposition 3
(i) *The condition* $g_{\mathrm{D}}(p) \geq g_{\mathrm{D}}(p - \chi_Y)$ $(\forall Y \subseteq V)$ *holds in each iteration, and therefore we may assume* $\varepsilon = +1$ *in Step 1.*
(ii) *In Step 1, we have*

$$X \in \arg\max\{|Y| \mid Y \subseteq V,\ p + \chi_Y \in S\}, \tag{4}$$

and such X *is uniquely determined. In particular,* $g_{\mathrm{D}}(p) \geq g_{\mathrm{D}}(p + \chi_X)$ *holds in Step 2 if and only if* $X = \emptyset$.
(iii) *Denote by* X_k *the set* X *found in Step 1 of the* k*-th iteration. Then, it holds that* $X_k \subseteq X_{k-1}$ *for all* $k \geq 2$.

Proof [Proof of (i)] The vector p is always contained in S in each iteration. If $p - \chi_Y \notin S$, then $g_{\mathrm{D}}(p - \chi_Y) = -\infty < g_{\mathrm{D}}(p)$. If $p - \chi_Y \in S$, then

$$g_{\mathrm{D}}(p - \chi_Y) = \sum_{v \in V \setminus \{s\}} p(v) - |Y \setminus \{s\}| \leq g_{\mathrm{D}}(p).$$

[Proof of (ii)] Since X satisfies $p + \chi_X \in S$, we have $s \notin X$. For $Y \subseteq V \setminus \{s\}$ with $p + \chi_Y \in S$, it holds that

$$g_{\mathrm{D}}(p + \chi_Y) = \sum_{v \in V \setminus \{s\}} p(v) + |Y| = g_{\mathrm{D}}(p) + |Y|.$$

Hence, the equation (4) follows. The uniqueness of X in (4) follows from Proposition 1. The latter statement is obvious from the equation (4).

[Proof of (iii)] For a fixed $k \geq 2$, let $p' = \sum_{i=1}^{k-2} \chi_{X_i}$. Since p' and $p' + \chi_{X_{k-1}} + \chi_{X_k}$ are in S, the discrete midpoint convexity (1) for S implies that $p' + \chi_{X_{k-1} \cup X_k} \in S$. By the choice of X_{k-1}, we have $|X_{k-1} \cup X_k| = |X_{k-1}|$ (see the claim (ii)), implying that $X_k \subseteq X_{k-1}$. □

From the observation above, the steepest ascent algorithm in Section 2 applied to the function g_{D} in (3) can be rewritten as follows with a variable U and a step size λ.

Algorithm 1 (Steepest Ascent Algorithm for (D))
Step 0: Set $p := \mathbf{0}$, $U := V$.
Step 1: Let X be the unique maximal subset of U such that $p + \chi_X \in S$.
Step 2: If $X = \emptyset$, then stop; p is an optimal solution of (D).
Step 3: Set $\lambda := \max\{\mu \in \mathbb{Z}_+ \mid p + \mu\chi_X \in S\}$, $p := p + \lambda\chi_X$, and $U := X$. Go to Step 1. □

It is noted that if $v \in U$, the value $p(v)$ may possibly be incremented in the following iterations, and if $v \in V \setminus U$, the value $p(v)$ remains the same in the following iterations. We also have $s \notin U$ in each iteration, except for the first iteration. It is easy to see that the following property holds, which will be used in the next section.

Proposition 4 *Let $X \subseteq V$ be the set computed in Step 1 of some iteration of Algorithm 1, and $\tilde{p} \in \mathbb{R}^V$ be the vector p after the update in Step 3 of the same iteration. Then, the values $\tilde{p}(v)$ $(v \in X)$ are the same.*

Remark 1 Algorithm 1 can be applied to the following more general problem:

$$\text{Maximize } \sum_{v \in V} w(v)p(v) \text{ subject to } p \in S,$$

where $w \in \mathbb{R}^V$ is a positive vector and $S \subseteq \mathbb{Z}^V$ is an L$^\natural$-convex set containing the zero vector. □

4.2 Implementation with Auxiliary Variables

We present an implementation of Algorithm 1 by using auxiliary variables. This reveals the connection between the steepest ascent algorithm for L$^\natural$-concave function maximization and Dijkstra's algorithm. To avoid complications of degeneracy we assume, to the end of this section, that edge length $\ell(e)$ is a positive integer for every $e \in E$; this assumption will be used only in the proof of Proposition 8.

In Step 3 of Algorithm 1, we need to compute the step size λ. This can be done by using auxiliary variables $\pi(v)$ $(v \in X)$ given by

$$\pi(v) = \min\{p(u) + \ell(u,v) \mid (u,v) \in E,\ u \in V \setminus X\} \qquad (v \in X). \tag{5}$$

Proposition 5 *Let $p \in S$ and $X \subseteq V \setminus \{s\}$. For $\pi(v)$ $(v \in X)$ in* (5) *we have*

$$\max\{\mu \in \mathbb{Z}_+ \mid p + \mu\chi_X \in S\} = \min\{\pi(v) - p(v) \mid v \in X\}. \tag{6}$$

Proof Put $p_\mu = p + \mu\chi_X$. Since $p \in S$ and $s \notin X$, we have $p_\mu \in S$ if and only if

$$p_\mu(v) - p_\mu(u) \le \ell(u,v) \qquad (\forall (u,v) \in E,\ u \in X,\ v \in V \setminus X); \tag{7}$$

for other edges (u,v), it holds that $p_\mu(v) - p_\mu(u) \le p(v) - p(u) \le \ell(u,v)$. Since $p_\mu(v) - p_\mu(u) = (p(v) + \mu) - p(u)$ holds in (7), the condition (7) is rewritten as $\mu \le \pi(v) - p(v)$ $(\forall v \in X)$. Therefore, the equation (6) follows. □

By Proposition 5, Step 3 of Algorithm 1 is rewritten as follows:

Step 3: For $v \in X$, set

$$\pi(v) := \min\{p(u) + \ell(u,v) \mid (u,v) \in E,\ u \in V \setminus X\}.$$

Set $\lambda := \min\{\pi(v) - p(v) \mid v \in X\}$, $p := p + \lambda\chi_X$, and $U := X$. Go to Step 1.

The proof of Proposition 5 shows the following facts, which are useful in computing the step direction X in Step 1 of Algorithm 1.

Proposition 6 *Let $p \in \mathbb{R}^V$ and $X \subseteq V$ be the vector and the set at the end of Step 1 of some iteration, and $\pi(v)$ $(v \in X)$ and λ be the values computed in Step 3 of the same iteration. Put $W = \arg\min\{\pi(v) - p(v) \mid v \in X\}$ and $\tilde{p} = p + \lambda\chi_X$, i.e., $\tilde{p}$ is the vector p after the update.*
(i) *If $v \in W$, then $\tilde{p}(v) = \pi(v)$ and $\tilde{p} + \chi_Y \notin S$ $(\forall Y \subseteq X$ with $v \in Y)$.*
(ii) *If $v \in X \setminus W$, then $\tilde{p}(v) < \pi(v)$.*

Proposition 7 *Let $p \in \mathbb{R}^V$, $X \subseteq V$, and $\pi(v)$ $(v \in V)$ be as in Proposition 6. Then, it holds that*

$$\arg\min\{\pi(v) - p(v) \mid v \in X\} = \arg\min\{\pi(v) \mid v \in X\}. \tag{8}$$

Proof By Proposition 4, the values $p(v)$ $(v \in X)$ are the same. Hence, the equation (8) follows. □

The step direction X in Step 1 of Algorithm 1 can be found easily by using the values $\pi(v)$ computed in Step 3 of the previous iteration.

Proposition 8 *Suppose that $\ell(e)$ is a positive integer for every $e \in E$.*
(i) *In Step 1 of the first iteration, we have $X = V \setminus \{s\}$.*
(ii) *Let X and $\pi(v)$ $(v \in X)$ be as in Proposition 6. Then, the set $\tilde{X}$ computed in Step 1 of the next iteration is given by*

$$\tilde{X} = X \setminus \arg\min\{\pi(v) \mid v \in X\}. \tag{9}$$

Proof [Proof of (i)] Since $\chi_X \in S$, we have $s \notin X$. To prove $X = V \setminus \{s\}$, it suffices to show that $\chi_{V\setminus\{s\}} \in S$. Putting $q = \chi_{V\setminus\{s\}}$, we have $q(v) - q(u) \leq 1 \leq \ell(u,v)$ for every $(u,v) \in E$ since q is a 0-1 vector. Hence, we have $q \in S$.

[Proof of (ii)] Let p, λ, and $\tilde{p}$ be as in Proposition 6. Put

$$W = \arg\min\{\pi(v) \mid v \in X\} = \arg\min\{\pi(v) - p(v) \mid v \in X\},$$

where the latter equality is by Proposition 7. By Proposition 6 (i), we have $\tilde{X} \subseteq X \setminus W$ since $\tilde{p} + \chi_{\tilde{X}} \in S$. To prove $\tilde{X} = X \setminus W$, we show that $\tilde{p} + \chi_{X\setminus W} \in S$. Since $\tilde{p} \in S$, we have $\tilde{p} + \chi_{X\setminus W} \in S$ if

$$\tilde{p}(v) - \tilde{p}(u) < \ell(u,v) \quad (\forall (u,v) \in E, v \in X \setminus W,\ u \in (V \setminus X) \cup W). \tag{10}$$

If $u \in W$, then we have $u, v \in X$, and therefore Proposition 4 implies that $\tilde{p}(v) - \tilde{p}(u) = 0 < 1 \leq \ell(u,v)$. If $u \in V \setminus X$, then Proposition 6 (ii) implies that

$$\tilde{p}(v) < \pi(v) \leq p(u) + \ell(u,v) = \tilde{p}(u) + \ell(u,v)$$

since $\pi(v) = \min\{p(r) + \ell(r,v) \mid (r,v) \in E,\ r \in V \setminus X\}$ and $\tilde{p}(u) = p(u)$. Hence, (10) follows. □

Based on Proposition 8, Algorithm 1 can be implemented by using auxiliary variables $\pi(v)$ $(v \in V)$ as follows.

Algorithm 2 (Implementation of Algorithm 1 with auxiliary variables)
Step 0: Set $p := \mathbf{0}$, $U := V$. Set $\pi(s) := 0$, $\pi(v) := +\infty$ $(v \in V \setminus \{s\})$.

Step 1: Set $W := \arg\min\{\pi(v) \mid v \in U\}$ and $X := U \setminus W$.
Step 2: If $X = \emptyset$, then stop; p is an optimal solution of (D).
Step 3: For $v \in X$, set

$$\pi(v) := \min\{p(u) + \ell(u,v) \mid (u,v) \in E,\ u \in V \setminus X\}.$$

Set $\lambda := \min\{\pi(v) - p(v) \mid v \in X\}$, $p := p + \lambda\chi_X$, and $U := X$. Go to Step 1. □

Proposition 9 *At the termination of Algorithm 2, $p(v) = \pi(v)$ holds for every $v \in V$.*

Proof In Step 1, $p(v) = \pi(v)$ holds for $v \in W$ by Proposition 6 (i), and the elements in W are deleted from U. Note that the values $p(v)$ and $\pi(v)$ do not change in the following iterations if v is deleted from U. Since every $v \in V$ is deleted (i.e., contained in W) in some iteration, the claim follows. □

Proposition 10 *For $v \in X$, the value $\pi(v)$ computed in Step 3 of Algorithm 2 satisfies*

$$\pi(v) = \min\left[\pi'(v), \min\{\pi'(u) + \ell(u,v) \mid (u,v) \in E,\ u \in W\}\right],$$

where $\pi'(r)$ $(r \in \{v\} \cup W)$ are the values of $\pi(r)$ at the beginning of Step 3.

Proof It suffices to show that

$$\begin{aligned}&\min\{p(u) + \ell(u,v) \mid (u,v) \in E,\ u \in V \setminus X\}\\ &= \min\left[\pi'(v), \min\{\pi'(u) + \ell(u,v) \mid (u,v) \in E,\ u \in W\}\right]. \end{aligned} \tag{11}$$

In the first iteration, we have $X = V \setminus \{s\}$ and $W = \{s\}$ by Proposition 8 (i). Therefore, (11) holds.

In the k-th iteration with $k > 1$, it holds that

$$\pi'(v) = \min\{p'(u) + \ell(u,v) \mid (u,v) \in E,\ u \in V \setminus U\},$$

where p' is the vector p in Step 1 of the $(k-1)$-st iteration. Hence, we obtain (11) as follows:

$$\begin{aligned}&\min\left[\pi'(v), \min\{\pi'(u) + \ell(u,v) \mid (u,v) \in E,\ u \in W\}\right]\\ &= \min\left[\min\{p'(u) + \ell(u,v) \mid (u,v) \in E,\ u \in V \setminus U\},\right.\\ &\qquad\left.\min\{\pi'(u) + \ell(u,v) \mid (u,v) \in E,\ u \in W\}\right]\\ &= \min\{p(u) + \ell(u,v) \mid (u,v) \in E,\ u \in V \setminus X\},\end{aligned}$$

where we use the fact that $p'(u) = p(u)$ for $u \in V \setminus U$ and $\pi'(u) = p(u)$ for $u \in W$ (see Proposition 6 (i)). □

Steps 2 and 3 of Algorithm 2 can be rewritten as follows by using Propositions 9 and 10, respectively.

Algorithm 3

Step 0: Set $p := \mathbf{0}$, $U := V$. Set $\pi(s) := 0$, $\pi(v) := +\infty$ $(v \in V \setminus \{s\})$.
Step 1: Set $W := \arg\min\{\pi(v) \mid v \in U\}$ and $X := U \setminus W$.
Step 2: If $X = \emptyset$, then stop; π $(= p)$ is an optimal solution of (D).
Step 3: For $v \in X$, set

$$\pi(v) := \min\left[\pi(v), \min\{\pi(u) + \ell(u,v) \mid (u,v) \in E,\ u \in W\}\right].$$

Set $\lambda := \min\{\pi(v) - p(v) \mid v \in X\}$, $p := p + \lambda\chi_X$, and $U := X$. Go to Step 1. □

We see that the variables p and λ are not needed to compute an optimal solution of (D), and therefore can be eliminated from Algorithm 3. The resulting algorithm coincides with Dijkstra's algorithm described in Section 1. That is, Dijkstra's can be recognized as an algorithm which implicitly computes an optimal solution of the L$^\natural$-concave maximization problem (D) in a greedy way.

Remark 2 We have demonstrated that Dijkstra's algorithm can be derived from the L$^\natural$-concave maximization algorithm when the edge lengths are positive integers. Even if some edges have zero lengths, Algorithm 2 as well as Algorithm 3 works well, although a degeneracy with $\lambda = 0$ may occur in Step 3. In the case of nonnegative integer edge length, the set X in Step 1 of Algorithm 2 satisfies $X \supseteq \hat{X}$, where $\hat{X}$ is the unique maximal subset of U such that $p + \chi_{\hat{X}} \in S$, and all the propositions in Section 4.2, except for Proposition 8, remain to be true. □

4.3 Real-Valued Edge Length

We have shown that the steepest ascent algorithm for L$^\natural$-concave function maximization coincides exactly with Dijkstra's algorithm by assuming that edge length is a nonnegative integer. For the general case of real-valued edge length, we can show the same statement by using the concept of polyhedral L$^\natural$-concave function in real variables as follows.

A polyhedral concave function $g : \mathbb{R}^V \to \mathbb{R} \cup \{-\infty\}$ is said to be *L$^\natural$-concave* [14] if $\mathrm{dom}\, g = \{p \in \mathbb{R}^V \mid g(p) > -\infty\}$ is nonempty and g satisfies the inequality

$$g(p) + g(q) \leq g((p + \lambda) \wedge q) + g(p \vee (q - \lambda\mathbf{1}))$$

for every $p, q \in \mathrm{dom}\, g$ and $\lambda \in \mathbb{R}_+$, where $\mathbf{1} \in \mathbb{R}^V$ is the vector all components equal to one. Note that an L$^\natural$-concave function on the integer lattice points defined in Section 2 is characterized by the same inequality, where p and q are restricted to integral vectors and λ is a nonnegative integer.

It can be shown that the following steepest ascent algorithm finds a maximizer of a polyhedral L$^\natural$-concave function $g : \mathbb{R}^V \to \mathbb{R} \cup \{-\infty\}$ with bounded $\mathrm{dom}\, g$. For $p \in \mathrm{dom}\, g$ and $q \in \mathbb{R}^V$, we define

$$g'(p; q) = \lim_{\alpha \downarrow 0} \frac{g(p + \alpha q) - g(p)}{\alpha},$$

which is the directional derivative of a polyhedral concave function g.

Steepest Ascent Algorithm for Polyhedral L$^\natural$-concave Function on $\mathbb{R}^V$

Step 0: Set $p := p_0$, where p_0 is an initial vector chosen from $\mathrm{dom}\, g$.
Step 1: Find $\varepsilon \in \{+1, -1\}$ and $X \subseteq V$ that maximize $g'(p; \varepsilon\chi_X)$.
Step 2: If $g'(p; \varepsilon\chi_X) \le 0$, then stop; p is a maximizer of g.
Step 3: Set

$$\lambda := \max\{\mu \in \mathbb{R}_+ \mid g(p + \mu\,\varepsilon\chi_X) - g(p) = \mu\, g'(p; \varepsilon\chi_X)\}.$$

Set $p := p + \lambda\,\varepsilon\chi_X$. Go to Step 1. □

Note that the maximum in Step 3 always exists since g is assumed to be a polyhedral concave function with bounded $\mathrm{dom}\, g$.

In the case of real-valued edge length, the LP dual of the shortest path problem is obtained from (D) by removing the integrality constraint $p(v) \in \mathbb{Z}$ ($v \in V \setminus \{s\}$). We can define a function $g : \mathbb{R}^V \to \mathbb{R} \cup \{-\infty\}$ associated with the LP dual as follows, in a similar way as in (3):

$$g(p) = \begin{cases} \displaystyle\sum_{v \in V \setminus \{s\}} p(v) & \text{(if } p \text{ is a feasible solution to the LP dual)}, \\ -\infty & \text{(otherwise)}. \end{cases}$$

Then, g is a polyhedral L$^\natural$-concave function. By applying the steepest ascent algorithm to this function g and implementing the algorithm with some auxiliary variables, we obtain Dijkstra's algorithm, in a similar way as in Section 4.

Remark 3 Dijkstra's algorithm is invariant under scaling of length of edges by a positive real number. The steepest ascent algorithm for polyhedral L$^\natural$-concave function maximization shares this invariance. Let g_α denote the function g for the edge length multiplied by $\alpha > 0$. Then we have $g_\alpha(\alpha p) = \alpha g(p)$ and the steepest ascent algorithm above applied to g_α with the initial vector αp_0 produces the same sequence of p's up to scaling by α. □

5 Concluding Remarks

We have revealed a close relationship between Dijkstra's algorithm for the shortest path problem and the steepest ascent algorithm for L$^\natural$-concave function maximization. This is not the only instance of the relationship between the L$^\natural$-concave function maximization algorithm and existing combinatorial optimization algorithms. Two other such instances are explained below: Hassin's dual algorithm for the minimum cost flow problem [6] and Chung–Tcha's dual algorithm for the minimum cost submodular flow problem [2].

In this connection it would be worth mentioning that the steepest ascent algorithm for another kind of discrete concave functions, called M$^\natural$-concave functions [5,12], has also a close relationship to classical combinatorial optimization algorithms. For example, Kalaba's algorithm [8] (see also [16]) for the minimum spanning tree problem can be understood as a special case of the steepest descent algorithm for M$^\natural$-convex functions given in [11,13].

5.1 Connection to Hassin's Algorithm

A dual algorithm for the minimum cost flow problem is proposed by Hassin [6]. We show that this algorithm coincides with the steepest ascent algorithm in Section 4.3 applied to the dual of the minimum cost flow problem.

For a directed graph $G = (V, E)$, nonnegative edge capacity $c(e)$, and edge cost $k(e)$ for $e \in E$, the minimum cost flow problem is formulated as follows:

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{(u,v)\in E} k(u,v)x(u,v) \\ \text{subject to} & \partial x(u) = 0 \quad (u \in V), \\ & 0 \le x(u,v) \le c(u,v) \quad ((u,v) \in E), \end{array}$$

where

$$\partial x(u) = \sum_{v:(u,v)\in E} x(u,v) - \sum_{v:(v,u)\in E} x(v,u) \qquad (u \in V).$$

The dual problem is given as

$$\begin{array}{ll} \text{Maximize} & g(p) = \displaystyle\sum_{(u,v)\in E} c(u,v)\min\{0, p(v) - p(u) + k(u,v)\} \\ \text{subject to} & p(v) \in \mathbb{R} \quad (v \in V). \end{array}$$

It can be shown that the objective function $g : \mathbb{R}^V \to \mathbb{R} \cup \{+\infty\}$ is a polyhedral L$^\natural$-concave function (see, e.g., [12,14]). Moreover, the function g has the property[1] of being constant in the direction of $\mathbf{1} = (1, 1, \ldots, 1)$:

$$g(p + \alpha\mathbf{1}) = g(p) \qquad (\forall p \in \mathbb{R}^V,\ \forall \alpha \in \mathbb{R}). \tag{12}$$

Suppose that the steepest ascent algorithm in Section 4.3 is applied to this function g. Due to (12), we can always choose $\varepsilon = +1$ in Step 1. Moreover, $g'(p; \chi_X)$ in Steps 1 and 2 is given as $g'(p; \chi_X) = I(p, X)$ with

$$I(p, X) = \sum_{(u,v)\in E^<_{\mathrm{in}}(p,X)} c(u,v) - \sum_{(u,v)\in E^\le_{\mathrm{out}}(p,X)} c(u,v), \tag{13}$$

$$E^<_{\mathrm{in}}(p, X) = \{(u,v) \in E \mid p(v) - p(u) + k(u,v) < 0,\ u \in V \setminus X, v \in X\},$$

$$E^\le_{\mathrm{out}}(p, X) = \{(u,v) \in E \mid p(v) - p(u) + k(u,v) \le 0,\ u \in X, v \in V \setminus X\},$$

and λ in Step 3 is expressed as $\lambda = \lambda(p, X)$ with

$$\lambda(p, X) = \min\big\{|p(v) - p(u) + k(u,v)| \ \big|\ (u,v) \in E^<_{\mathrm{in}}(p,X) \cup E^>_{\mathrm{out}}(p,X)\big\}, \tag{14}$$

$$E^>_{\mathrm{out}}(p, X) = \{(u,v) \in E \mid p(v) - p(u) + k(u,v) > 0,\ u \in X, v \in V \setminus X\}.$$

Hence, the steepest ascent algorithm can be rewritten as follows:

[1] A polyhedral L$^\natural$-concave function satisfying the condition (12) is called a polyhedral L-concave function (see, e.g., [12]).

Step 0: Set $p := p_0$, where p_0 is an initial vector chosen from $\mathbb{R}^V$.
Step 1: Find $X \subseteq V$ that maximizes $I(p, X)$.
Step 2: If $I(p, X) \leq 0$, then stop; p is a maximizer of g.
Step 3: Set $p := p + \lambda(p, X)\, \chi_X$. Go to Step 1. □

This is nothing but Hassin's dual algorithm.

A similar connection can be established for the discrete version with integral dual variable $p \in \mathbb{Z}^V$, where we assume that edge costs are integral; note that for such a case there exists an integral dual optimal solution (see, e.g., [16]). Such an observation to connect Hassin's algorithm to L$^\natural$-concave maximization leads to a new technical result that Hassin's algorithm, when combined with a standard scaling approach, runs in weakly polynomial time (see [12,13]), although no polynomial bound is not shown in [6].

5.2 Connection to Chung–Tcha's Algorithm

A dual algorithm for the minimum cost submodular flow problem is proposed by Chung and Tcha [2]. We show that this algorithm coincides with the steepest ascent algorithm in Section 4.3 applied to the dual of the minimum cost submodular flow problem.

For a directed graph $G = (V, E)$, nonnegative edge capacity $c(e)$ and edge cost $k(e)$ for $e \in E$, and a submodular function $\rho : 2^V \to \mathbb{R}$ with $\rho(\emptyset) = \rho(V) = 0$, the minimum cost submodular flow problem is formulated as follows:

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{(u,v)\in E} k(u,v)x(u,v) \\ \text{subject to} & \displaystyle\sum_{u\in Y} \partial x(u) \leq \rho(Y) \quad (Y \subseteq V), \\ & 0 \leq x(u,v) \leq c(u,v) \quad ((u,v) \in E). \end{array}$$

The linear programming dual is given as

$$\begin{array}{ll} \text{Maximize} & \displaystyle -\sum_{(u,v)\in E} c(u,v)s(u,v) - \sum_{Y\subseteq V} \rho(Y)t(Y) \\ \text{subject to} & \displaystyle -s(u,v) + \sum_{Y:u\in Y} t(Y) - \sum_{Y:v\in Y} t(Y) \leq k(u,v) \quad ((u,v) \in E), \\ & s(u,v) \geq 0 \qquad ((u,v) \in E), \\ & t(Y) \geq 0 \quad (Y \subseteq V). \end{array}$$

It is known that there exists a vector $p \in \mathbb{R}^V$ such that $s_p(u,v)$ $((u,v) \in E)$ and $t_p(Y)$ $(Y \subseteq V)$ defined by

$$\begin{aligned} s_p(u,v) &= -\min\{0, p(v) - p(u) + k(u,v)\} \qquad ((u,v) \in E), \\ t_p(Y) &= \begin{cases} \tilde{p}_i - \tilde{p}_{i+1} & (\text{if } Y = L_i,\ 1 \leq i \leq k-1), \\ 0 & (\text{otherwise}) \end{cases} \end{aligned} \tag{15}$$

provide an optimal solution of the dual problem, where

$$\tilde{p}_1 > \tilde{p}_2 > \cdots > \tilde{p}_k \text{ are distinct values of components of } p,$$
$$L_i = \{v \in V \mid p(v) \geq \tilde{p}_i\} \qquad (i = 1, 2, \ldots, k-1)$$

(see [2,4]; see also Theorem 5.6 and its proof in [5]). We use this fact to rewrite the dual problem.

We define a function $\hat{\rho} : \mathbb{R}^V \to \mathbb{R}$ by

$$\hat{\rho}(p) = \sum_{Y \subseteq V} \rho(Y) t_p(Y) = \sum_{i=1}^{k-1} (\tilde{p}_i - \tilde{p}_{i+1}) \rho(L_i) \qquad (p \in \mathbb{R}^V).$$

Note that the function $\hat{\rho}$ is the so-called Lovász extension of submodular function ρ (see, e.g., [5]). Then, the dual problem is rewritten as follows:

$$\begin{array}{ll} \text{Maximize} & g(p) = \displaystyle\sum_{(u,v) \in E} c(u,v) \min\{0, p(v) - p(u) + k(u,v)\} - \hat{\rho}(p) \\ \text{subject to} & p(v) \in \mathbb{R} \quad (v \in V). \end{array}$$

The Lovász extension of a submodular function, in general, is an L$^\natural$-convex function with the property (12), and therefore the objective function $g : \mathbb{R}^V \to \mathbb{R} \cup \{+\infty\}$ is also a polyhedral L$^\natural$-concave function satisfying the property (12) (see, e.g., [12,14]).

Suppose that the steepest ascent algorithm in Section 4.3 is applied to this function g. Due to (12), we can always choose $\varepsilon = +1$ in Step 1. Moreover, we have

$$g'(p; \chi_X) = I(p, X) - \hat{\rho}'(p; \chi_X)$$

in Steps 1 and 2 with $I(p, X)$ in (13), and

$$\lambda = \min\{\lambda(p, X), \mu(p, X)\}$$

in Step 3 with $\lambda(p, X)$ in (14) and

$$\mu(p, X) = \min\{\tilde{p}_i - \tilde{p}_{i+1} \mid 1 \leq i \leq k-1,\ (L_{i+1} \setminus L_i) \cap X \neq \emptyset,\ (L_i \setminus L_{i-1}) \setminus X \neq \emptyset\},$$

where L_0 is defined to be an empty set[2]. Hence, the steepest ascent algorithm can be rewritten as follows:

Step 0: Set $p := p_0$, where p_0 is an initial vector chosen from $\mathbb{R}^V$.
Step 1: Find $X \subseteq V$ that maximizes $I(p, X) - \hat{\rho}'(p; \chi_X)$.
Step 2: If $I(p, X) \leq \hat{\rho}'(p; \chi_X)$, then stop; p is a maximizer of g.
Step 3: Set $p := p + \min\{\lambda(p, X), \mu(p, X)\} \chi_X$. Go to Step 1. □

[2] The value of $\hat{\rho}'(p; \chi_X)$ admits an explicit formula, which is omitted here for simplicity of the description.

This coincides with the dual algorithm by Chung and Tcha.

A similar connection can be also established for the discrete version with integral dual variable $p \in \mathbb{Z}^V$ in the case of integral edge costs; note that for such a case there exists an integral dual optimal solution (see, e.g., [16]). Such an observation to connect Chung–Tcha's algorithm to L$^\natural$-concave maximization leads to a new technical result that Hassin's algorithm, when combined with a standard scaling approach, runs in weakly polynomial time (see [12, 13]), although no polynomial bound is not shown in [2].

Acknowledgements This research is partially supported by KAKENHI (21360045, 21740060) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), the Global COE program "The Research and Training Center for New Development in Mathematics," and the Aihara Project, the FIRST program from the Japan Society for the Promotion of Science (JSPS).

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Upper Side River (1993).
2. Chung, N.-k., Tcha, D.-w.: A dual algorithm for submodular flow problems. Oper. Res. Letters **10**, 489–495 (1991).
3. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**, 269–271 (1959).
4. Edmonds, J.,Giles, R.: A min-max relation for submodular functions on graphs. Ann. Discrete Math. **1**, 185–204 (1977).
5. Fujishige, F.: Submodular Functions and Optimization, Second Edition. Elsevier, Amsterdam (2005).
6. Hassin, R.: The minimum cost flow problem: a unifying approach to dual algorithms and a new tree-search algorithm. Math. Programming **25**, 228–239 (1983).
7. Iwata, S. , Fleischer, L., Fujishige, S.: A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions. J. ACM **48**, 761–777 (2001).
8. Kalaba, R.: On some communication network problems. In: Bellman, R., Hall, M., Jr, (eds.) Proc. Symposia in Applied Mathematics **10**, pp. 261–280, American Mathematical Society, Providence (1960).
9. Kolmogorov, V., Shioura, A.: New algorithms for convex cost tension problem with application to computer vision. Discrete Optim. **6**, 378–393 (2009).
10. Murota, K.: Discrete convex analysis. Math. Program. **83**, 313–371 (1998).
11. Murota, K.: Algorithms in discrete convex analysis. IEICE Trans. Systems and Information **E83-D**, 344–352 (2000).
12. Murota, K.: Discrete Convex Analysis. SIAM, Philadelphia (2003).
13. Murota, K.: On steepest descent algorithms for discrete convex functions. SIAM J. Optim. **14**, 699–707 (2003).
14. Murota, K., Shioura, A: Extension of M-convexity and L-convexity to polyhedral convex functions. Adv. Appl. Math. **25**, 352–427 (2000).
15. Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. J. Combin. Theory Ser. B **80**, 346–355 (2000).
16. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Springer, Berlin (2003).