Linear Layouts in Submodular Systems

Hiroshi Nagamochi

Department of Applied Mathematics and
Physics, Graduate School of Informatics,
Kyoto University, Japan,
nag@amp.i.kyoto-u.ac.jp

Abstract

Linear layout of graphs/digraphs is one of the classical and important optimization problems that have many practical applications. Recently Tamaki proposed an $O(mn^{k+1})$-time and $O(n^k)$-space algorithm for testing whether the pathwidth (or vertex separation) of a given digraph with n vertices and m edges is at most k. In this paper, we show that linear layout of digraphs with an objective function such as cutwidth, minimum linear arrangement, vertex separation (or pathwidth) and sum cut can be formulated as a linear layout problem on a submodular system (V, f) and then propose a simple framework of search tree algorithms for finding a linear layout (a sequence of V) with a bounded width that minimizes a given cost function. According to our framework, we obtain an $O(kmn^{2k})$-time and $O(n+m)$-space algorithm for testing whether the pathwidth of a given digraph is at most k.

Key words. Linear Layout, Submodular Functions, Optimization, Pathwidth, Cutwidth, Search Tree Algorithm

1 Introduction

Let $G = (V, E)$ stand for an undirected or directed graph with a set V of n vertices and a set E of m edges. Linear layout of graphs is a problem of finding a linear arrangement (a sequence of V) $\sigma = (v_1, \ldots, v_n)$ of the vertex set V of G so that a prescribed cost function $cost(\sigma)$ is minimized. The problem is one of the classical and important optimization problems that have many practical applications (e.g., see [4]). From practical point of views, there have been introduced several different choices of cost functions, among which the following ones can be described by vertex/edge-cut functions of digraphs (where we regard an undirected graph as a symmetric digraphi i.e., treat each undirected edge uv as two oppositely directed edges (u, v) and (v, u)):

- Cutwidth: $cost_{\rm CW}(\sigma) = \max\{d^+_G(\{v_1, \ldots, v_i\}) \mid 1 \le i \le n-1\}$, where $d^+_G(X)$ denotes the number of directed edges with a tail in X and a head in $V - X$;
- Minimum Linear Arrangement: $cost_{\rm MLA}(\sigma) = \sum\{d^+_G(\{v_1, \ldots, v_i\}) \mid 1 \le i \le n-1\}$;
- Vertex Separation (or Pathwidth): $cost_{\rm VS}(\sigma) = \max\{\Gamma^+_G(\{v_1, \ldots, v_i\}) \mid 1 \le i \le n-1\}$, where $\Gamma^+_G(X)$ denotes the number of out-neighbors of a subset X (the vertices $v \in X$ that have directed edges from v to a vertex in $V - X$); and
- Sum Cut: $cost_{\rm SC}(\sigma) = \sum\{\Gamma^+_G(\{v_1, \ldots, v_i\}) \mid 1 \le i \le n-1\}$.

In these functions, directed edges with the backward direction in a sequence σ are ignored. The *cutwitdh* (resp., *vertex veparation*) of G is defined to be the minimum of $cost_{\rm CW}(\sigma)$ (resp., $cost_{\rm VS}(\sigma)$) over all sequences σ of V. The vertex separation of a digraph G is equal to the "pathwidth" of G (e.g., [11]), which is a width of a path-decomposition of G.

[1]Technical report 2012-004, June 13, 2012.

Bodlaender et al. [3] showed that a class of linear layout problems including the above four can be solved (i) in $O^*(2^n)$ time and $O^*(2^n)$ space by a dynamic programming; and (ii) in $O^*(4^n)$ time and polynomial space by a search tree algorithm (where the O^*-notation suppresses factors that are polynomial in n).

When a problem is parameterized by value k of its cost function, it is known that CUTWIDTH and VERTEX SEPARATION admit faster exact algorithms. We let CUTWIDTH(k) (resp., VERTEX SEPARATION(k)) stand for the problem of testing whether a given graph/digraph G has a sequence σ of V such that $cost_{\mathrm{CW}}(\sigma) \leq k$ (resp., $cost_{\mathrm{VS}}(\sigma) \leq k$) or not. Gurari and Sudborough [8] presented an $O(n^k)$-time and exponential-space dynamic programming algorithm for CUTWIDTH(k) in undirected graphs, and Makedon and Sudborough [12] later improved the time bound to $O(n^{k-1})$. For CUTWIDTH(k) in undirected graphs with a fixed k, Fellows and Langston [6] obtained an $O(n^2)$-time algorithm, and the time bound is improved to linear by Abrahamson and Fellows [1] and Thilikos et al. [19]. For VERTEX SEPARATION(k) in undirected graphs with a fixed k, Fellows and Langston [6] designed an $O(n^3)$-time algorithm, and afterwards Bodlaender [2] gave a linear time algorithm. For undirected graphs, it is known that the graph minor theorem by Robertson and Seymour [16] implies polynomial-time algorithms for problems CUTWIDTH(k) and VERTEX SEPARATION(k) with fixed k and that the theorem, however, cannot be applied to the directed case (e.g., see [18, 19])

Recently Tamaki [18] proposed an $O(mn^{k+1})$-time and $O(n^k)$-space algorithm for testing whether the pathwidth (or vertex separation) of a given digraph with n vertices and m edges is at most k. Although it remains open whether VERTEX SEPARATION(k) in digraphs is fixed-parameter tractable or not, it is the first nontrivial step toward design of efficient exact algorithms for computing graph parameters of digraphs. His algorithm is a search tree algorithm equipped with a pruning procedure that tries to discard one of two partial sequences with the same length by a dominance relationship. It is proven that the number of all partial sequences with the same length during an execution is always $O(n^k)$, which ensures the claimed time and space complexities of the algorithm. More interestingly, although the submodularity of function Γ_G^+ is used to derive the upper bound $O(n^k)$, the mechanism of the algorithm is self-contained in the sense that it never relies on any other optimization mechanism such as submodular minimization and dynamic programming to attain the nontrivial upper bound. In fact, recently Nagamochi [13] proved that the new mechanism can be conversely used to solve the submodular minimization problem, the most representative optimization problem.

From these observations, it would be natural to find a way of applying submodular minimization to the pathwidth problem in digraphs. Our research group has implemented Tamaki's algorithm to investigate the distribution of pathwidth of chemical graphs, and it turned out that the $O(n^k)$-space algorithm easily uses up the memory allowed for graphs with over 100 vertices [9]. This is another motivation for us to develop a more space-efficient algorithm for the problem.

In this paper, we show that linear layout of digraphs with an objective function such as cutwidth, minimum linear arrangement, vertex separation (or pathwidth) and sum cut can be formulated as a linear layout problem on a submodular system (V, f) and then propose a simple framework of search tree algorithms for finding a linear layout (a sequence of V) with a bounded width that minimizes a given cost function. Also our framework can handle precedent relations such that a certain element u is required to precede some other element v in an output sequence.

The paper is organized as follows. Section 2 reviews basic results on submodular functions and introduces a layout problem in submodular systems. Section 3 presents a key property on sequences in submodular systems, based on which a search tree algorithm is designed. Section 4 analyzes the time complexity of the algorithm applied to the problem of testing whether the cutwidth/pathwidth of a given digraph is at most k. Finally Section 5 makes concluding remarks.

2 Preliminaries

Submodular Systems Let V denote a given finite set with $n \geq 1$ elements. A set function f on V is called *submodular* if $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$ for every pair of subsets $X, Y \subseteq V$. There are numerous examples of submodular set functions such as cut function of digraphs and hypergraphs, matroid rank function, and entropy function. The problem of finding a subset X that minimizes $f(X)$ over a submodular set function f is one of the most fundamental and important issues in optimization. Grotschel, Lovasz, and Schrijver gave the first polynomial time [7] algorithm for minimizing a submodular set function [7]. Schrijver [17] and Iwata, Fleischer, and Fujishige [10] independently developed strongly polynomial time combinatorial algorithms for the submodular minimization. Currently an $O(n^6 + n^5\theta)$-time minimization algorithm is obtained by Orlin [14], where $n = |V|$ and θ is the time to evaluate $f(X)$ of a specified subset X.

For two disjoint subsets $S, T \subseteq V$, an *(S,T)-separator* is defined to be a subset X such that $S \subseteq X \subseteq V - T$, and let $f_{\min}(S,T)$ denote the minimum $f(X)$ of an (S,T)-separator X, where such a set X is called a *minimum* (S,T)-separator. We denote (S,T) with $S = \{s\}$ and $T = \{t\}$ by (s,t).

We here remark that the problem of finding a subset X with minimum $f(X)$ in a submodular system (V, f) is essentially equivalent to that of finding a minimum (S,T)-separator in a submodular system. The problem of finding a subset X with minimum $f(X)$ in a submodular system (V, f) is essentially equivalent to that of finding a minimum (S,T)-separator in a submodular system in the following sense. A given submodular system (V, f) can be extended to a submodular system $(V' = V \cup \{s,t\}, g)$ by defining $g(X) = f(X - \{s,t\})$ for each subset $X \subseteq V \cup \{s,t\}$. Note that g is submodular, since for every two sets $X, Y \subseteq V \cup \{s,t\}$, $g(X) + g(Y) = f(X - \{s,t\}) + f(Y - \{s,t\}) \geq f((X - \{s,t\}) \cap (Y - \{s,t\})) + f((X - \{s,t\}) \cup (Y - \{s,t\})) = g(X \cap Y) + g(X \cup Y)$. Hence a minimum (s,t)-separator X in (V', g) gives a subset $X - \{s,t\} \subseteq V$ with minimum $f(X)$. Now we show the converse. To find a minimum (S,T)-separator in a given submodular system (V, f) and $S, T \subseteq V$, we consider a submodular system $(V' = V - S - T, g)$ by defining $g(X) = f(X \cup S)$ for each subset $X \subseteq V - S - T$. Note that g is submodular, since for every two sets $X, Y \subseteq V - S - T$, $g(X) + g(Y) = f(X \cup S) + f(Y \cup S) \geq f((X \cup S) \cap (Y \cup S)) + f((X \cup S) \cup (Y \cup S)) = g(X \cap Y) + g(X \cup Y)$. Hence a subset $X \subseteq V'$ with minimum $g(X)$ gives a minimum (S,T)-separator $X \cup S$ in (V, f).

Sequences For two integers $i \leq j$, the set of all integers h with $i \leq h \leq j$ is denoted by $[i,j]$. A sequence σ consisting of some elements in a finite set V is called *non-duplicating* if each element of V occurs at most once in σ. We denote by Σ_i the set of all non-duplicating sequences of exactly i elements in V, where Σ_0 contains only the null sequence (the sequence of length zero). We denote $\cup_{0 \leq i \leq n} \Sigma_i$ by Σ. Let $\sigma \in \Sigma$ be a sequence. We denote by $V(\sigma)$ the set of elements constituting σ and by $|\sigma| = |V(\sigma)|$ the length of σ. Let $\sigma(i)$ denote the ith element in a sequence σ, and Let σ_i be the sequence that consists of the first i elements of σ, i.e., $\sigma_i = (\sigma(1), \sigma(2), \ldots, \sigma(i))$. Given two disjoint subsets $S, T \subseteq V$, a sequence σ is called an (S,T)-sequence if $V(\sigma_{|S|}) = S$ and $V - V(\sigma_{|V-T|}) = T$. We let $\overline{X}$ denote $V - X$.

For two sequences $\alpha, \beta \in \Sigma$ such that $V(\alpha) \cap V(\beta) = \emptyset$, we denote by $\alpha\beta$ the sequence $\sigma \in \Sigma_{|\alpha|+|\beta|}$ obtained by appending β to α so that $\sigma(i) = \alpha(i)$ for $i \leq |\alpha|$ and $\sigma(i) = \beta(i - |\alpha|)$ otherwise.

For a subset $X \subseteq V$, let $\sigma[X]$ denote the sequence $\sigma' \in \Sigma_{|V(\sigma) \cap X|}$ such that $V(\sigma') = V(\sigma) \cap X$ and for every two elements $u, v \in V(\sigma')$, u precedes v in σ' if and only if u precedes v in σ.

Linear Layouts We consider a cost function *cost* on sequences $\sigma \in \Sigma$. A cost function *cost* is called *non-decreasing* if *cost*(σ) is determined only by $\{f(\sigma_1), f(\sigma_2), \ldots, f(\sigma_{\ell-1})\}$ ($\ell = |\sigma|$) and *cost*(σ) does not decrease when $f(\sigma_i)$ for some i increases, where we regard $\{f(\sigma_1), \ldots, f(\sigma_{\ell-1})\}$ as a multiset consisting of exactly ℓ numbers. For example, the following three functions are all non-decreasing:

$$f_{\min}(\sigma) = \min\{f(\sigma_i) \mid 1 \leq i \leq \ell - 1\},$$

$$f_{\max}(\sigma) = \max\{f(\sigma_i) \mid 1 \le i \le \ell - 1\},$$
$$f_{\text{sum}}(\sigma) = \sum\{f(\sigma_i) \mid 1 \le i \le \ell - 1\}.$$

We call $f_{\max}(\sigma)$ the *f-width* of σ.

For a subset X of a digraph $G = (V, E)$, let $\Gamma^-_G(X)$ denote the number of in-neighbors of a subset X (the vertices $v \in V - X$ that have directed edges from v to a vertex in X), and let $d^-_G(X) = d^+_G(V - X)$. Observe that $cost_{\text{CW}} = f_{\max}$ and $cost_{\text{MLA}} = f_{\text{sum}}$ for the edge-cut function $f = d^+_G$, and $cost_{\text{VS}} = f_{\max}$ and $cost_{\text{SC}} = f_{\text{sum}}$ for the vertex-cut function $f = \Gamma^+_G$.

We are ready formulate a general form of the problems studied in this paper:

> **Linear Layouts in Submodular Systems** Given a nonnegative submodular system (V, f) with $f(\emptyset) = f(V) = 0$ ($n = |V|$), a positive real $k > 0$ and a non-decreasing cost function *cost*, find a sequence $\sigma \in \Sigma_n$ with f-width at most k that minimizes $cost(\sigma)$ among all sequences with f-width at most k.

Note that there is a chance that a sequence $\tau \in \Sigma_n$ with f-width greater than k attains $cost(\tau)$ smaller than the minimum $cost(\sigma)$ of the above problem when *cost* is not given by $f_{\max}$. However our main result (Theorem 1) still suggests that for the problem of minimizing $cost_{\text{MLA}}$ or $cost_{\text{SC}}$, $f_{\max}$ is a useful measure to parameterize these problems, since values of these cost functions in strongly connected digraphs are not less than n are inadequate to measure the computational tractability.

Precedent Constraint In some application of arrangement of elements such as scheduling problems (e.g., see section 11.2 in [4]), an output sequence is required to meet a precedent relation among elements such that an element u precedes another element v, denoted by $u \prec v$. The set of such ordered pairs (u, v) can be given by a poset P on V, where P is represented by a set of directed edges (u, v) such that $u \prec v$ and there is no element w with $u \prec w$ and $w \prec v$ (in general P is not necessarily equal to a given digraph G itself). We can naturally include the side constraint as a penalty function into a given submodular system (V, f). Define the DAG (V, P), and let p be the submodular function on V by defining $p(X) = (k + 1)d^-_P(X)$ for each subset $X \subseteq V$, where $d^-_P(X)$ denotes the the number of directed edges of (V, P) with a tail in $V - X$ and a head in X. Clearly $(V, f' = f + g)$ remains a submodular system, and any sequence σ of V with $f'_{\max}(\sigma) \le k$ satisfies $f_{\max}(\sigma_i) = f'_{\max}(\sigma_i) \le k$ for $i = 1, 2, \ldots, n - 1$, which indicates that there is no edge $(u, v) \in P$ such that $i > j$ for $\sigma(i) = u$ and $\sigma(j) = v$, i.e., the given precedent constraint is met.

Main Result In this paper, we prove the next.

Theorem 1 *Given a submodular system (V, f), a real k, and a non-decreasing function cost, a minimum cost sequence σ with f-width at most k (if any) can be obtained by solving submodular minimization $O(n^{2\Delta(k)+2})$ times using $O(|V|)$ work space except for storage of f, where $\Delta(k)$ denotes the number of distinct values in $\{f(X) \le k \mid \emptyset \subsetneq X \subsetneq V\}$.*

In particular, when f is integer-valued and k is a positive integer, it holds $\Delta(k) \le k - \min_{\emptyset \subsetneq X \subsetneq V} f(X)$.

3 Algorithm

This section proves Theorem 1 by presenting a search tree algorithm that solves the problem. All we need to design our new algorithm is the following observation.

Lemma 2 *For a submodular system (V, f), let τ be an (S, T)-sequence $\tau \in \Sigma_n$ of V. For a minimum (S, T)-separator A in (V, f), let $\sigma = \tau[A]\tau[\overline{A}] \in \Sigma_n$, and ψ be a bijection on $[|S| + 1, n - |T|]$ such that $\psi(i)$ is the index j such that $\sigma(i) = \tau(j)$. Then*

$$f(\sigma_i) \le f(\tau_{\psi(i)}) \text{ for all } i \in [|S| + 1, n - |T|]. \qquad (1)$$

Pproof. Fix $i \in [|S|+1, n-|T|]$, and let $j = \psi(i)$. Since $V(\tau_j) \cup A$ and $V(\tau_j) \cap A$ are (S,T)-separators, we have $f(A) = f_{\min}(S,T) \le \min\{f(V(\tau_j) \cup A), f(V(\tau_j) \cap A)\}$. Hence by the submodularity of f, it holds $f(A) + f(\tau_j) \ge f(V(\tau_j) \cap A) + f(V(\tau_j) \cup A)$, from which we have $f(\tau_j) \ge \max\{f(V(\tau_j) \cap A), f(V(\tau_j) \cap A)\}$.

We first consider the case where $|S|+1 \le i \le |A|$. In this case it holds $V(\sigma_i) = V(\tau_j) \cap A$ and we have $f(\sigma_i) = f(V(\tau_j) \cap A) \le f(\tau_j)$. On the other hand ($|A|+1 \le i \le n-|T|$), it holds $V(\sigma_i) = V(\tau_j) \cup A$ and we have $f(\sigma_i) = f(V(\tau_j) \cup A) \le f(\tau_j)$. ∎

Note that (1) implies that $cost(\sigma[V-S-T]) \le cost(\tau[V-S-T])$ for any non-decreasing cost function *cost*.

Fix a nonnegative submodular system (V,f) with $f(\emptyset) = f(V) = 0$ and a real number k, an instance of our problem is specified by an ordered pair (S,T) of disjoint subsets $S, T \subseteq V$, to which we wish to find an (S,T)-sequence σ such that the f-width of the subsequence $\sigma[V-S-T]$ is at most k and $cost(\sigma[V-S-T])$ is minimized among all such sequences σ. Such an (S,T)-sequence σ is called a *solution* to the instance (S,T).

To find a solution to a given instance (S,T) by a search tree algorithm, we introduce branch operations based on Lemma 2.

For every two elements $s, t \in V$ in a given submodular system (V,f), we first genetare an instance $(S = \{s\}, T = \{t\})$. There are at most n^2 such instances. Let $f^* = \min\{f(X) \mid \emptyset \subsetneq X \subsetneq V\}$.

An instance (S,T) with $|V-S-T| \le 1$ is trivial since it has a uniqe solution (if any). Let $|V-S-T| \ge 2$ Compute $f_{\min}(S,T)$ invoking submodular minimization on f. Assume that $f_{\min}(S,T) \le k$, since otherwise there is no (S,T)-sequence σ such that the f-width of $\sigma[V-S-T]$ is at most k.

Case 1. $f_{\min}(S,T) = k$: In this case, we can reduce (S,T) into trivial one. Choose an arbitrary element $u \in V-S-T$ such that $f(S \cup \{u\}) = k$ (if no such element $u \in V-S-T$ exits then the instance (S,T) has no solution either). By Lemma 2, a solution to (S,T) can be obtained by combining solutions to $(S, T' = V-S-\{u\})$ and $(S' = S \cup \{u\}, T)$. Since $(S, T' = V-S-\{u\})$ has a unique solution, this reduces the current instance (S,T) to $(S' = S \cup \{u\}, T)$, where $f_{\min}(S', T) = k$ still holds. Hence we can apply the above procedure until the intance becomes trivial (or we find out infeasibility of (S,T)).

Case 2. $f_{\min}(S,T) < k$: We furhter test whether there is a minimum (S,T)-separator A with $S \subsetneq A \subsetneq V-T$ (this can be done by computing $f_{\min}(S \cup \{u\}, T \cup \{v\})$ for all pairs $u, v \in V-S-T$, thus $O(|V-S-T|^2)$ times of submodular minimization).

Case 2a. A minimum (S,T)-separator A with $S \subsetneq A \subsetneq V-T$ exists: We split the current instance into two instances $(S, T' = V-A)$ and $(S' = A, T)$. By Lemma 2, a solution to (S,T) can be obtained by combining solutions to $(S, T' = V-A)$ and $(S' = A, T)$.

Case 2b. No minimum (S,T)-separator A with $S \subsetneq A \subsetneq V-T$ exists; i.e., only S or $V-T$ is a minimum (S,T)-separator:

(i) Exactly one of S and $V-T$, say S is a minimum (S,T)-separator: We branch into $|V-S-T|$ instances $I_u = (S_u = S \cup \{u\}, T)$, $u \in V-S-T$, and select an (S,T)-sequence with minimum *cost* among solutions to I_u, $u \in V-S-T$ as a solution to (S,T). Note that $f_{\min}(S_u, T) > f_{\min}(S,T)$.

(ii) Both of S and $V-T$ are minimum (S,T)-separators: We branch into $|V-S-T|(|V-S-T|-1)$ instances $I_{uv} = (S_u = S \cup \{u\}, T_v = T \cup \{v\})$, $u, v \in V-S-T$, and select an (S,T)-sequence with minimum *cost* among solutions to I_{uv}, $u, v \in V-S-T$ as a solution to (S,T). Note that $f_{\min}(S_u, T_v) > f_{\min}(S,T)$.

The above branching rules give our search tree algorithm. See Appendix A for an entire description of the algorithm.

We now analyze the time and space complexities of our algorithm.

For each instance (S,T), we solve submodular minimization $O(|V-S-T|^2)$ times to generate a set of instances in Case 2. Let $\Delta(a,b)$ denote the number of distinct values in $\{f(X) \mid a \le f(X) < b,\ X \subseteq V\}$. It is not difficult to see that the number of instances in the search tree rooted at an instance (S,T) is at most $|V-S-T|^{2\Delta(f_{\min}(S,T),k)}$ since the number of branches is $|V-S-T|$ and the depth of the rooted tree is $\Delta(f_{\min}(S,T),k)$. Hence we have the next.

Lemma 3 *From an instance (S,T), at most $|V-S-T|^{2\Delta(f_{\min}(S,T),k)}$ instances that invoke submodular minimization will be generated.*

Proof. From an instance (S,T), at most $|V-S-T|^{2\Delta(f_{\min}(S,T),k)}$ instances that invoke submodular minimization will be generated.

In Case 1, the current instance (S,T) will be reduced to a trivial instance without branching. Thus an instance (S,T) with $f_{\min}(S,T)=k$ can be solved without generating an instance that invokes submodular minimization. Hence the claim holds since $\Delta(f_{\min}(S,T),k)=0$.

In Case 2a, the current instance (S,T) is split into two instances. It holds $1+|V-A-T|^{2\Delta(f_{\min}(S,T'),k)}+|A-S|^{2\Delta(f_{\min}(S',T),k)} \leq |V-S-T|^{2\Delta(f_{\min}(S,T),k)}$.

We consider Case 2b(ii) (Case 2b(i) can be treated analogously). The current instance (S,T) is split into $|V–S–T|(|V–S–T|-1)$ instances (S_u,T_v) $(u,v\in V–S–T)$ with $f_{\min}(S_u,T_v)>f_{\min}(S,T)$. Hence it holds $1+\sum_{u,v\in V–S–T}|V-S_u-T_v|^{2\Delta(f_{\min}(S_u,T),k)} \leq |V–S–T|^2\cdot|V–S–T|^{2(\Delta(f_{\min}(S,T),k)-1)} \leq |V-S-T|^{2\Delta(f_{\min}(S,T),k)}$.

From the above argument, the lemma holds. ∎

It always holds $f_{\min}(S,T)\geq f^*$ for any generated instances (S,T). By Lemma 3, our algorithm generates from each instance $(S=\{s\},T=\{t\})$, at most $n^{2\Delta(f^*,k)}=n^{2(\Delta(k)-1)}$ instances that invokes submodular minimization, where $\Delta(k)=|\{f(X)\leq k \mid \emptyset\subsetneq X\subsetneq V\}|$. Since there are at most n^2 pairs of (s,t) and each instance invokes at most n^2 submodular minimization, the number of times for solving submodular minimizations is at most $n^2 n^{2\Delta(k)-2}n^2=n^{2\Delta(k)+2}$. This proves Theorem 1.

4 Digraph Case

In this section, we consider layout of a digraph $G=(V,E)$ with cost functions $cost_{\mathrm{CW}}$, $cost_{\mathrm{MLA}}$, $cost_{\mathrm{VS}}$ and $cost_{\mathrm{SC}}$, and analyze upper bounds on the time and space complexities of our algorithm applied to these problems using flow technique. We consider the problem of finding a minimum cost of an (S,T)-sequence σ with f-width at most k.

4.1 Cutwidth and Minimum Linear Arrangement

We here show how to find a minimum cost layout of a digraph under a fixed cutwidth. First consider the case where there is no precedent constraint, i.e., we set $f=d_G^+$; we assume that G is connected and $m\geq n-1$. Let λ denote the edge-connectivity of G, i.e., $\lambda=\min_{\emptyset\subsetneq X\subsetneq V} d^+(X)$. In this case, $f_{\min}(S,T)$ and a minimum (S,T)-separator can be obtained by computing a maximum (s',t')-flow φ in a directed network G' obtained from G by contracting S and T into single vertices s' and t', where the capacity of each directed edge is 1. From a maximum (s',t')-flow φ, we can find a minimum (S,T)-separator A with $S\subsetneq A\subsetneq V-T$ in G in linear time (if any) by constructing a DAG representation of all minimum (S,T)-separators in linear time [15] without newly solving $O(n^2)$ minimization problems. Hence for each instance (S,T), we need to solve a single maximum flow problem, which takes $O(k(m+n))$ time and $O(n+m)$ space [5], where we do not need to find any minimum (S,T)-separator when the flow value exceeds k. Since the total number of instances to be generated is at most $n^2 n^{2\Delta(k)-2}\leq n^2 n^{2(k-\lambda+1)-2}$, the entire time complexity is $O(kmn^{2(k-\lambda+1)})$.

Theorem 4 *Given a digraph $G=(V,E)$ with n vertices and m edges and an integer $k\geq 1$, whether there is a sequence of V with pathwidth at most k can be tested in $O(kmn^{2(k-\lambda+1)})$ time and $O(n+m)$ space. When such a sequence exists, a sequence $\sigma\in\Sigma_n$ with pathwidth at most k that minimizes $cost_{\mathrm{CW}}$ can be found in the same time and space complexities.*

We next consider the case where a precedent constraint is imposed as a poset $P \subseteq V \times V$ i.e., we set $f = d^+_G + (k+1)d^-_P$ (note that f-width at most k is equal to d^+_G-width in any sequences). In this case, let $\overline{P} = \{(v,u) \mid (u,v) \in P\}$, and augment G by adding all edges $(v,u) \in \overline{P}$ to obtain a directed network, where the capacity of each directed edge in E is 1 and we treat each $(v,u) \in \overline{P}$ as $k+1$ multiple edges with capacity 1. Hence the number m' of edges in the augmented multigraph is at most $m+(k+1)|P|$. For a given (S,T), we can obtain $f_{\min}(S,T)$ and a minimum (S,T)-separator in a similar way; we compute a maximum (s',t')-flow in the directed network after contracting S and T into single vertices s' and t', taking $O(km'n^{2(k-\lambda+1)}) = O(k(m+n+k|P|)n^{2(k-\lambda+1)})$ time and $O(n+m+|P|)$ space.

Theorem 5 *Given a digraph $G = (V,E)$ with n vertices and m edges, a poset $P \subseteq V \times V$ and an integer $k \geq 1$, whether there is a sequence of V with pathwidth at most k which meets the precedent constraint by P can be tested in $O(k(m+n+k|P|)n^{2(k-\lambda+1)})$ time and $O(n+m+|P|)$ space. When such a sequence exists, a sequence $\sigma \in \Sigma_n$ with pathwidth at most k that minimizes $cost_{\mathrm{CW}}$ under the precedent constraint by P can be found in the same time and space complexities.*

For the layout of digraphs with sum cut $cost_{\mathrm{CW}}$, the same statements of Theorems 4 and 5 hold by replacing $cost_{\mathrm{CW}}$ with $cost_{\mathrm{MLA}}$.

4.2 Vertex Separation and Sum Cut

We here show how to find a minimum cost layout of a digraph under a fixed pathwidth (or vertex separation). We consider the case where a precedent constraint where a precedent constraint is imposed as a poset $P \subseteq V \times V$ i.e., we set $f = \Gamma^+_G + (k+1)d^-_P$ (note that f-width at most k is equal to Γ^+_G-width in any sequences).

In this case, we can compute a minimum (S,T)-separator by computing a maximum flow applying the standard technique of converting vertex-cuts into edge-cuts (however $\min_{\emptyset \subsetneq X \subsetneq V} \Gamma^+(X) \leq \min_{v \in V} \Gamma^+(V - \{v\}) \leq 1$ is not the vertex-connectivity of G). For this, we construct a digraph $G_P = (V' \cup V'', A_E \cup A_V \cup A_{\overline{P}})$ as follows. Let $\overline{P} = \{(v,u) \mid (u,v) \in P\}$. Replace each vertex $v \in V$ with two copies v' and v'' with a new directed edge (v',v''), and let $A_V = \{(v',v'') \mid v \in V\}$. For each directed edge $(u,v) \in E$, we set a directed edge (u'',v') in G_P, and let $A_E = \{(u'',v') \mid (u,v) \in E\}$. For each directed edge $(v,u) \in \overline{P}$, we set a directed edge (v'',u''), and let $A_{\overline{P}} = \{(v'',u'') \mid (v,u) \in \overline{P}\}$, where we treat each edge (v'',u'') in G_P as $k+1$ multiple edges. The number m' of edges in the multigraph G_P is at most $m+n+(k+1)|P|$. The next lemma verifies that we can obtain a minimum (S,T)-separator A with $\Gamma^+(A) \leq k$ (if any) by computing a minimum $(\hat{S},\hat{T})$-separator in G_P for $\hat{S} = \{u',u'' \mid u \in S\}$ and $\hat{T} = \{u'' \mid u \in T\}$.

Lemma 6 *For the vertex-cut function Γ^+_G of a digraph $G = (V,E)$, and the penalty function $p = (k+1)d^-_P$ defined by a poset P on V, let $f = \Gamma^+_G + p$ be a set function on V. Let g be the edge-cut function $d^+_{G_P}$ of $G_P = (V' \cup V'', A_E \cup A_V \cup A_{\overline{P}})$ defined from (G,P,k) in the above. Given two disjoint subsets $S,T \subseteq V$, let $\hat{S} = \{u',u'' \mid u \in S\}$ and $\hat{T} = \{u'' \mid u \in T\}$. Then $f_{\min}(S,T) > k$ if and only if $g_{\min}(\hat{S},\hat{T}) > k$; and if $f_{\min}(S,T) \leq k$, then $g_{\min}(\hat{S},\hat{T}) = f_{\min}(S,T)$.*

Proof. (i) First we show that $f_{\min}(S,T) \leq k$ implies $f_{\min}(S,T) \geq g_{\min}(\hat{S},\hat{T})$. Let X be an (S,T)-separator attaining $f(X) = f_{\min}(S,T) \leq k$. Since $f(X) \leq k$, there is no edge $(v,u) \in \overline{P}$ such that $v \in X$ and $u \in V - X$. Hence $f(X) = \Gamma^+_G(X)$. From X, we construct a subset $X' \subseteq V' \cup V''$ by $X' = \{v',v'' \mid v \in X\} \cup \{v' \mid (u,v) \in E,\ u \in X,\ v \in V - X\}$. Then the set of directed edges outgoing from X' in G_P is $\{(v',v'') \mid v \in V - X$ with some $u \in X$ and $(u,v) \in E\}$, where $S' \subseteq X'$ and $X' \cap T' = \emptyset$. Hence $f_{\min}(S,T) = f(X) = \Gamma^+_G(X) = g(X')$. Since X' is an $(\hat{S},\hat{T})$-separator in G_P, we have $f_{\min}(S,T) = g(X') \geq g_{\min}(\hat{S},\hat{T})$, as required.

(ii) Next we show that $g_{\min}(\hat{S},\hat{T}) \leq k$ implies $g_{\min}(\hat{S},\hat{T}) \geq f_{\min}(S,T)$. Let Y be an $(\hat{S},\hat{T})$-separator attaining $g(Y) = g_{\min}(S',T') \leq k$. Since $g(Y) \leq k$, there is no edge $(v'',u'') \in A_{\overline{P}}$ such that $v'' \in Y$ and $u'' \notin Y$. From Y, we construct a subset $Y' \subseteq V$ by $V_Y = \{v \mid v'' \in Y\}$, where $S \subseteq V_Y$ and $V_Y \cap T = \emptyset$. Note that there is no edge $(v,u) \in \overline{P}$ such that $v \in Y'$ and $u \notin Y'$, since otherwise there would be an edge $(v'',u'') \in \overline{P}$ with $v'' \in Y$ and $u'' \notin Y$ which leaves from Y in G_P, a contradiction. Hence $\Gamma^+(V_Y) = f(V_Y)$. For each edge $(u'',v') \in A_E$ outgoing from V_Y, $(u,v) \in E$ will be an edge outgoing from V_Y in G, where the edge $(u'',v') \in A_E$ corresponds to the out-neighbor v of V_Y. For each edge $(v',v'') \in A_V$ outgoing from V_Y, there is an edge $(u'',v') \in A_E$ (otherwise $Y - \{v'\}$ would satisfy $g(Y - \{v'\}) < g(Y)$), and $(u,v) \in E$ will be an edge outgoing from V_Y in G, where the edge $(v',v'') \in A_V$ corresponds to the out-neighbor v of V_Y. Since any edge outgoing from Y corresponds to an out-neighbor of V_Y, we have $g(Y) \geq \Gamma^+(V_Y)$. Therefore $g_{\min}(\hat{S},\hat{T}) = g(Y) \geq \Gamma^+(V_Y) = f(V_Y) \geq f_{\min}(S,T)$, as required.

Note that (i) implies that if $g_{\min}(\hat{S},\hat{T}) > k$ then $f_{\min}(S,T) > k$, while (ii) means that if $f_{\min}(S,T) > k$ then $g_{\min}(\hat{S},\hat{T}) > k$. Hence $f_{\min}(S,T) > k$ if and only if $g_{\min}(\hat{S},\hat{T}) > k$. Assume that $f_{\min}(S,T) \leq k$, which now implies $g_{\min}(\hat{S},\hat{T}) \leq k$. Hence from (i) and (ii), we have $g_{\min}(\hat{S},\hat{T}) \leq f_{\min}(S,T)$ and $g_{\min}(\hat{S},\hat{T}) \geq f_{\min}(S,T)$; i.e., $g_{\min}(\hat{S},\hat{T}) = f_{\min}(S,T)$. This completes the proof. ∎

Since a minimum $(\hat{S},\hat{T})$-separator in G_P can be obtained by computing a maximum (s',t')-flow after contracting $\hat{S}$ and $\hat{T}$ into single vertices s' and t'. The single maximum flow problem can be solved in $O(km') = O(k(m+n+|P|))$ time and $O(n+m') = O(m+n+|P|)$ space analogously with the case of cutwidth. Hence the time bound is $O(k(m+n+|P|)n^2 n^{2\Delta(k)-2}) = O(k(m+n+|P|)n^2 n^{2(k+1)-2})$. In particular, when no precedent constraint is imposed, we can assume that G is strongly connected (otherwise a solution is easily obtained) and we can set $\Delta(k) \leq k$ and $|P| = 0$ in these bounds. Therefore we obtain the following results.

Theorem 7 *Given a digraph $G = (V,E)$ with n vertices and m edges and an integer $k \geq 1$, whether there is a sequence of V with pathwidth at most k can be tested in $O(kmn^{2k})$ time and $O(n+m)$ space. When such a sequence exists, a sequence $\sigma \in \Sigma_n$ with pathwidth at most k that minimizes $cost_{\rm VS}$ can be found in the same time and space complexities.*

Theorem 8 *Given a digraph $G = (V,E)$ with n vertices and m edges, a poset $P \subseteq V \times V$ and an integer $k \geq 1$, whether there is a sequence of V with pathwidth at most k which meets the precedent constraint by P can be tested in $O(k(m+n+k|P|)n^{2k+2})$ time and $O(n+m+|P|)$ space. When such a sequence exists, a sequence $\sigma \in \Sigma_n$ with pathwidth at most k that minimizes $cost_{\rm VS}$ under the precedent constraint by P can be found in the same time and space complexities.*

For the layout of digraphs with sum cut $cost_{\rm SC}$, the same statements of Theorems 7 and 8 hold by replacing $cost_{\rm VS}$ with $cost_{\rm SC}$.

5 Concluding Remarks

In this paper, we introduced a linear layout in submodular systems (V,f), which includes several linear layout problems in graphs/digraphs, defining non-decreasing cost functions and f-width. We proposed a framework for search tree algorithms of finding a minimum cost layout with a bounded f-width. In particular, we obtained $O(kmn^{2k})$-time and $O(n+m)$-space algorithms for testing whether the cutwidth/pathwidth of a given digraph is at most k or not. Our result in contrast to Tamak's algortihm has a similar trade-off between the $O^*(2^n)$-time and space algorithm and the $O^*(4^n)$-time and polynomial-space algorithms; reducing the space complexity to polynomial one increases the time complexity up to the square of it (the work complexity). Theorem 1 would indicate that f-width is a useful parameter to investigate the tractability of layout problems with cost functions whose value is as large as n.

Acknowledgment

The author would like to thank Prof. Hisao Tamaki for useful discussions.

References

[1] K. Abrahamson, M. Fellows. Finite automata, bounded treewidth and well-quasiordering. in: Contemp. Math., 147: Amer. Math. Soc., Providence, RI, 539–563, 1993.

[2] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. 25:(6) 1305–1317, 1996.

[3] H. L. Bodlaender, F. V. Fomin, A. M. C. A. Koster, D. Kratsch, D. M. Thilikos. A note on exact algorithms for vertex ordering problems on graphs. Theory Comput. Syst. 50: 420–432, 2012.

[4] J. Díaz, J Petit, M. Serna. A survey of graph layout problems. ACM Computing Surveys (CSUR) 34(3): 313–356, 2002.

[5] S. Even, R. E. Tarjan. Network flow and testing graph connectivity. SIAM J. Comput., 4: 507–518, 1975.

[6] M. R. Fellows, M. A. Langston. Layout permutation problems and well-partially ordered sets. In Advanced Research in VLSI. MIT Press, Cambridge, MA, 315–327, 1988.

[7] M. Grötschel, L. Lovász, A. Schrijver. The ellipsoid algorithm and its consequences in combinatorial optimization. Combinatorica 1: 499–513, 1981.

[8] E. Gurari, I. H. Sudborough. Improved dynamic programming algorithms for the bandwidth minimization and the mincut linear arrangement problem. J. Algorithm 5, 531–546, 1984.

[9] M. Ikeda, H. Nagamochi. A method for computing the pathwidth of chemical graphs. The 15th Japan-Korea Joint Workshop on Algorithms and Computation, July 10 - 11, 2012, National Institute of Informatics, Tokyo, Japan (to appear).

[10] S. Iwata. L. Fleischer, S. Fujishige. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions. J. ACM 48: 761–777, 2001.

[11] N. G. Kinnersley. The vertex separation number of a graph equals its path-width. Inf. Proc. Lett. 42: 345-350, 1992.

[12] F. Makedon, I. H. Sudborough. On minimizing width in linear layouts. Dis. Appl. Math. 23:(3), 243–265, 1989.

[13] H. Nagamochi. Submodular minimization via pathwidth. TAMC2012, LNCS 7287, 584–593, 2012.

[14] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. Math. Program., Ser. A 118: 237–251, 2009.

[15] J. -C. Picard, M. Queyranne. On the structure of all minimum cuts in a network and applications, Math. Prog. Study, 13: 8–16, 1980.

[16] N. Robertson, P. Seymour. Graph Minors. XX. Wagner's conjecture. J. Combin. Theory Ser. B 92(2): 325–35, 2004.

[17] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. J. Combin. Theory Ser. B 80: 346–355, 2000.

[18] H. Tamaki. A polynomial time algorithm for bounded directed pathwidth. LNCS 6986, 331-342, 2012.

[19] D. M. Thilikos, M. Serna, H. L. Bodlaender. Cutwidth I: A linear time fixed parameter algorithm. J. of Algorithms 56: 1–24, 2005.

Appendix A

We here present of an entire description of our algorithm stated in Section 3. Let (V, f) be a nonnegative submodular system with $f(\emptyset) = f(V) = 0$, k be a positive number and $s, t \in V$ be and two distinct elements. We execute the following algorithm Arrange(S,T) with $S = \{s\}$ and $T = \{t\}$ to find an (s,t)-sequence σ with f-width at most k that minimizes a given non-decreasing cost function *cost* among (s,t)-sequences with f-width at most k.

Algorithm Arrange(S,T)
Input: An ordered disjoint subsets (S,T) of V.
Output: An (S,T)-sequence σ with $f_{\max}(\sigma) \leq k$ that minimizes *cost* among all such (S,T)-sequences, or a message "none" when there is no (S,T)-sequence σ with $f_{\max}(\sigma) \leq k$.

  if $|V - S - T| \leq 1$ **then**
    Halt returning the unique (S,T)-sequence if $f(S), f(\overline{T}) \leq k$ or
    returning message "none" otherwise
  endif;
  /* $|V - S - T| \geq 2$ */
  if $f_{\min}(S,T) > k$ **then** halt returning message "none"
  endif;
  /* $f_{\min}(S,T) \leq k$ */
  if $f_{\min}(S,T) = k$ **then**
    Let η be the null sequence;
    while $|S| + |\eta| + |T| < |V|$ **do**
      Choose any element $u \in V - S - T$ such that $f(S \cup V(\eta u)) = k$; $\eta := \eta u$
      (if no such u exists then halt returning message "none")
    endwhile;
    Halt returning the (S,T)-sequence η
  endif;
  /* $f_{\min}(S,T) < k$ */
  if There is a minimum (S,T)-separator A with $A \neq S, \overline{T}$ **then**
    $S' := A$; $T' := \overline{A}$;
    if one of Arrange(S,T') and Arrange(S',T) is "none" **then**
      halt returning message "none"
    else
      $\sigma :=$Arrange(S,T'); $\tau :=$Arrange(S',T);
      Halt returning $\eta := \sigma[A]\tau[\overline{A}]$
    endif
  else /* only S or $\overline{T}$ is a minimum (S,T)-separator */
    if $\overline{T}$ is not a minimum (S,T)-separator **then**
      for each $u \in V - S - T$ **do**
        $S_u := S \cup \{u\}$; $c_u := cost($Arrange$(S_u, T))$
        (let $c_u := \infty$ if Arrange$(S_u,T) =$"none")

```
    endfor;
    Halt returning η :=ARRANGE(S_{u*}, T) for
    u* = argmin{c_u < ∞ | u ∈ V − S − T}
    or returning "none" if no such u* exists
  elseif S is not a minimum (S, T)-separator then
    for each v ∈ V − S − T do
      T_v := T ∪ {v}; c_v := cost(ARRANGE(S, T_v))
      (let c_v := ∞ if ARRANGE(S, T_v) ="none")
  endfor;
  Halt returning η :=ARRANGE(S, T_{v*}) for
  v* = argmin{c_v < ∞ | v ∈ V − S − T}
  or returning "none" if no such v* exists
else /* both of S and T̄ are a minimum (S, T)-separators */
  for each ordered pair (u, v) with u, v ∈ V − S − T do
    S_u := S ∪ {u}; T_v := T ∪ {v}; c_uv := cost(ARRANGE(S_u, T_v))
    (let c_uv := ∞ if ARRANGE(S_u, T_v) ="none")
  endfor;
  Halt returning η :=ARRANGE(S_{u*}, T_{v*}) for
  (u*, v*) = argmin{c_uv < ∞ | u, v ∈ V − S − T}
  or returning "none" if no such (u*, v*) exists
endif
```