

直並列グラフの全域配送木

川端 真生^{1,a)} 西関 隆夫^{1,b)}

概要：グラフ G にはソースが1つだけあるとし、そのソース w には供給量と呼ばれる正整数が割り当てられ、ソース以外の点は全てシンクであり、需要量と呼ばれる非負整数が割り当てられている。各辺には容量と呼ばれる正整数が割り当てられているとする。 G の全域木 T が全域配送木と呼ばれるのは、ソース w から各シンク v へその需要量だけのフローを T 上の w から v へ行く道に沿って流したときに、 T の各辺 e に流れるフローの値が e の辺容量以下である時である。全域配送木問題とは、与えられたグラフ G に全域配送木が存在するかどうか判定する問題である。本文では、まず全域配送木問題が直並列グラフに対してすら NP 完全であることを示す。次に直並列グラフに対し全域配送木問題を解く擬多項式時間アルゴリズムを与える。

キーワード：全域配送木，直並列グラフ，ネットワークフロー，辺容量

1. まえがき

図1に示すように、グラフ $G = (V, E)$ は連結であり、ソース（供給点）が1つだけあるとし、そのソース $w \in V$ には供給量と呼ばれる正整数 $\text{sup}(w)$ が割り当てられている。 G のソース w 以外の全ての点はシンク（需要点）であるとして一般性を失わない。各シンク $v \in V$ には需要量と呼ばれる非負整数 $\text{dem}(v)$ が割り当てられている。また、 G の各辺 $e \in E$ には正整数の辺容量 $\text{cap}(e)$ が与えられている。図1においてソースは正方形で、シンクは丸で描かれており、それらの中にある整数は供給量あるいは需要量であり、各辺に付けられた整数はその辺容量である。

ソース w から各シンク v へ $\text{dem}(v)$ だけフローを流したい。ただし、 G の全域木 T を1つ選び、どのシンク v に対しても T 上の w から v へ行く道に沿ってフローを $\text{dem}(v)$ だけ流すことにする。しかも T の各辺 e を流れるフローの値を、 e の容量 $\text{cap}(e)$ 以下にしたい。むしろ $\text{sup}(w)$ は需要量の合計以上でなければならない。このような全域木 T を G の全域配送木と呼ぶ。図1のグラフの全域配送木 T の一例が太線で描かれている。 T の各辺 e に付けられた括弧の中の整数は e を流れているフローの値である。 T を w を根とする根付き木とみなしたとき、 T の辺 $e = (u, u')$ の端点 u が根付き木 T において u' の親であるとき、辺 e を流れるフローの値は、 u' および u' の子孫の需要量の合計

に等しい。与えられたグラフ G に全域配送木が存在するかどうか判定する問題を 全域配送木問題 と呼ぶ。

全域配送木問題は単一ソース、多重シンクのネットワークフロー問題に似ているが、各シンク v へ行くフローが1本の道フローになっていなければならない、即ち “unsplittable flow” [2], [3], [10] でなければならない。しかし、どのシンク v への道フローも G の同じ全域木 T 上の w から v へ行く道でなければならない点が unsplittable flow とは異なる。

全域配送木問題は電力網の配電計画問題やインターネットの Server-client 配送問題等によく現れる [1], [5], [6], [7], [8], [9], [11], [14]。ソースが1個とは限らず複数個ある場合には、ソースが1個の場合と同様にして全域配送林が定義できる。与えられたグラフ G が木であるときには、辺容量がない場合 [7] および辺容量がある場合 [8] に全域配送林を線形時間で求めるアルゴリズムが知られている。木より広いグラフのクラス、例えば直並列グラフに対して全域配送木問題を効率よく解くアルゴリズムを開発することが望まれている。スタイナー木問題など多くの組み合わせ問題は直並列グラフに対して線形時間で解けることが知られているが [13]、直並列グラフに対してすら NP-完全である問題もいくつか知られている。その1つに辺素道問題がある。グラフ G が与えられ、 G のいくつかの点対が指定されているときに、それらの点対を結ぶ道で互いに共通な辺を持たないものが G に存在するかどうかというのが 辺素道問題 である。この辺素道問題は直並列グラフに対してすら NP-完全であることが知られている [12]。

¹ 関西学院大学

2-1 Gakuen, Sanda, 669-1337 Japan

a) masaki.kawabata.kwansei@gmail.com

b) nishi@kwansei.ac.jp

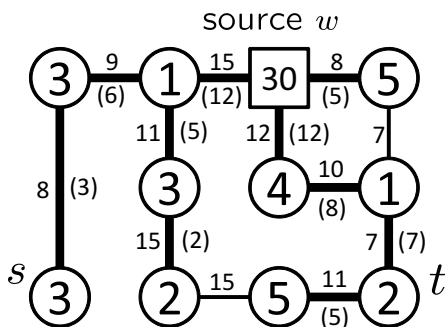


図 1 直並列グラフ G とその全域配送木 T

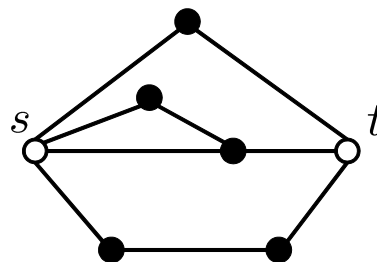


図 3 直並列グラフ

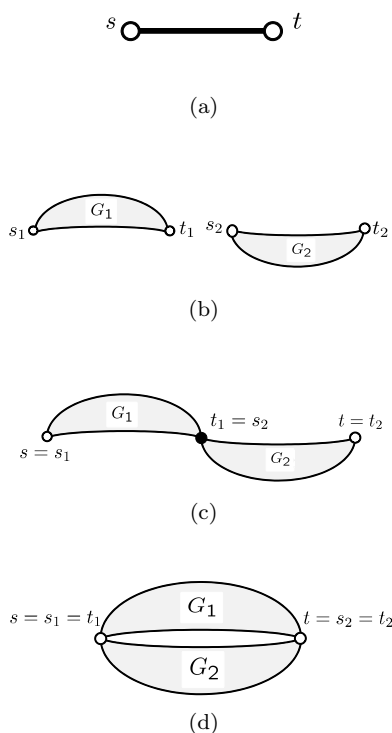


図 2 直並列グラフの定義

全域配送木問題は辺素道問題とも似てはいるが、道フローが全域木を誘導しなければならない点異なる。

本文では、まず全域配送木問題は直並列グラフに対してすら NP 完全である数少ない問題であることを示す。次に直並列グラフの全域配送木問題を解く擬多項式時間アルゴリズムを与える。与えられた直並列グラフ G の点数を n とし、需要量の総和を D とすると、本文のアルゴリズムの計算時間は $O(D^4 n)$ である。したがって、 D が定数ならば線形時間で、 D が n の多項式ならば多項式時間で走る。

2. 直並列グラフの全域配送木問題の NP 完全性

直並列グラフは次のように再帰的に定義される [13]。

1. 図 2(a) のように 1 つの辺 (s, t) からなるグラフ G は直並列グラフである。その辺の両端点 s, t はグラフ G の

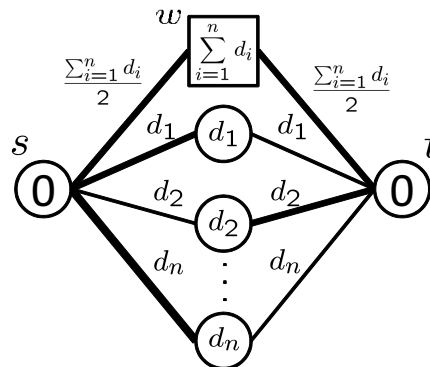


図 4 全域配送木問題への多項式時間帰着

端子と呼ばれる。

2. 図 2(b) のような端子 s_1, t_1 を持つ直並列グラフ G_1 と端子 s_2, t_2 を持つ直並列グラフ G_2 を次のように接続して得られるグラフ G は直並列グラフである。

- (a) 図 2(c) のように t_1 と s_2 を同一視して得られるグラフ G 。ただし、 G の端子は $s = s_1$ と $t = t_2$ である。この接続を直列接続と呼ぶ。

- (b) 図 2(d) のように s_1 と s_2 を同一視し、 t_1 と t_2 を同一視して得られるグラフ G 。ただし G の端子は $s = s_1 = s_2$ と $t = t_1 = t_2$ である。この接続を並列接続と呼ぶ。

図 1 および 3 のグラフは直並列グラフである。

本節では全域配送木問題は直並列グラフに対してすら NP 完全であることを示す。全域配送木問題は明らかに NP に属するので、既に NP 完全な問題として知られている集合分割問題 [4], p.47, が直並列グラフの全域配送木問題に多項式時間で帰着できることを示せばよい。ここで集合分割問題とは、与えられた n 個の正整数 d_1, d_2, \dots, d_n からなる集合 A を 2 つの部分集合 A_1 と A_2 に分割し、集合 A_1 内の整数の和が A_2 内の整数の和と等しくなるようにできるかどうかを判定する問題である。この集合分割問題の問題例 $A = \{d_1, d_2, \dots, d_n\}$ から直並列グラフ G を図 4 のように構成する。 G には 1 個のソース w と $n+2$ 個のシンクがある。 w の供給量 $\text{sup}(w)$ は $\sum_{i=1}^n d_i$ であるとする。また、整数 d_1, d_2, \dots, d_n の各々を需要量とする n 個のシンクと需要量 0 の 2 個のシンク s と t が G にある。点 s や t とソース w とを容量 $\sum_{i=1}^n d_i/2$ の辺で結ぶ。さらに $1 \leq i \leq n$ なる各 i について需要量 d_i のシンクを点 s や

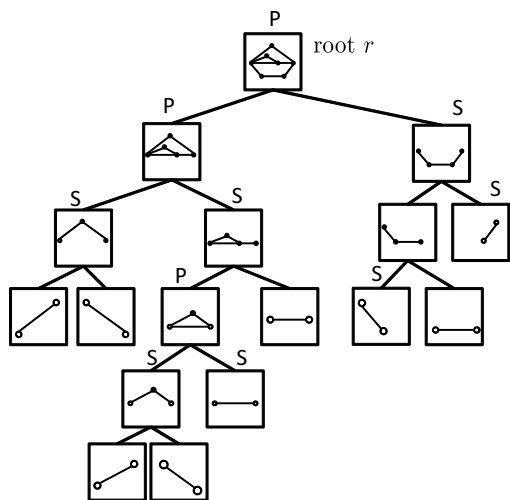


図5 図3の直並列グラフの二分分解木 T_{BD}

t と容量 d_i の辺で結ぶ．こうして得られた直並列グラフが G である．明らかに分割問題に解があるとき，かつその時に限り G に全域配送木が存在する．このようにして次の定理が得られる．

定理1 全域配送木問題は直並列グラフに対してすら NP 完全である．

このようにして， $P \neq NP$ ならば，直並列グラフの全域配送木問題を多項式時間で解くアルゴリズムは存在しない．

3. 擬多項式時間アルゴリズム

本節では，直並列グラフ G の全域配送木問題を解く擬多項式時間アルゴリズムを与える． n をグラフの点数とし，需要量の合計を D とすると，そのアルゴリズムの計算時間は $O(D^4 n)$ である．

3.1 直並列グラフの二分分解木とアルゴリズムの概要

直並列グラフ G は二分分解木 T_{BD} により表現することができる．図3の直並列グラフの二分分解木を図5に示す． T_{BD} の各点 u は G の部分グラフ G_u に対応する． u が T_{BD} の葉ならば， G_u は G の1本の辺から誘導される部分グラフである． T_{BD} の内点 u に付けられたラベル s, p はそれぞれ直列接続と並列接続を表している． u が T_{BD} の内点であり，その子が u_1 と u_2 であり， u のラベルが s (あるいは p) ならば， G_{u_1} と G_{u_2} を直列接続 (あるいは並列接続) して得られたグラフが G_u である．なお，直並列グラフの二分分解木は線形時間で構成できる [13]．

本文のアルゴリズムは二分分解木に基づいた動的計画法を利用する．直並列グラフ G の二分分解木 T_{BD} の内点を u, u' とし，図6のように u, u' に対応するグラフ G の部分グラフをそれぞれ $G_u, G_{u'}$ とすると， G の任意の全域配送木 T はグラフ $G_u, G_{u'}$ の林 $F_u, F_{u'}$ を誘導する．図6で T は太線で描かれている．この例では，林 F_u は木であり，ソース w は G_u に含まれており， F_u は G_u の全域配送木で

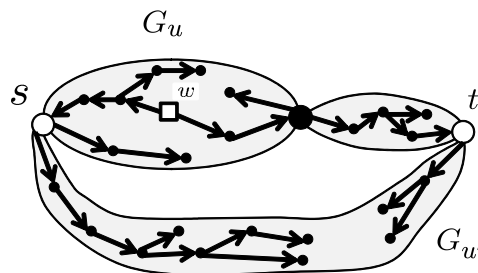


図6 直並列グラフ G の全域配送木

もある．一方，林 $F_{u'}$ は2本の木からなる林であり， $G_{u'}$ の全域二木である．本文のアルゴリズムは， G の二分分解木 T_{BD} の葉から根に向かって動的計画法を適用し，このような全域配送木あるいは“全域配送二木”を求めていく．

グラフ G には丁度2個のソース w_1 と w_2 があり，それらの供給量を $\text{sup}(w_1)$ ， $\text{sup}(w_2)$ とする． G の全域部分グラフ F が全域配送二木と呼ばれるのは， F が2本の点素な木 T_1, T_2 からなり， $i = 1$ および 2 について次の (a) および (b) が成立するときである．

(a) 木 T_i はソース w_i を含み， T_i 内の需要量の合計は $\text{sup}(w_i)$ 以下である．

(b) ソース w_i から T_i の各点 $v \in \text{dem}(v)$ だけのフローを木 T_i 上の w_i から v へ行く道に沿って流したときに， T_i の各辺 e のフローの値は $\text{cap}(e)$ 以下である．

具体的には全域配送木や全域配送二木そのものではなく，それらが存在するかどうかを表現する6つの関数を計算する． G_u がソース w を含むとき， G_u の全域配送木あるいは全域配送二木は G_u の端子からフローを出力できる．一方， G_u がソース w を含まないとき，端子からフローを入力してもらわないといけない．こうした出力量と入力量 x, y を変数とする6個の関数 $f_{s,t}, f_s^t, f_t^s, g_s^s, g_s^t, g^{s,t}$ を各部分グラフ G_u に対して定義して，動的計画法により直並列グラフの二分分解木 T_{BD} の葉から根に向けてこれら6個の関数を計算する．関数名 f や g の下添字は出力端子名を，上添字は入力端子名を表している．

3.2 関数の定義

需要量の合計 D は $\text{sup}(w)$ 以下であるとしてよい．むしろ，どの辺を流れるフローも D 以下である． $0 \leq z \leq D$ なる全ての整数 z からなる集合を \mathbb{Z}_D と書く．各部分グラフ G_u に関して定義される6個の関数の変数 $x, y \in \mathbb{Z}_D$ は，各々端子 s および t からのフローの入出力量を表す．一方，関数の値は1または0であり， G_u に全域配送木や全域配送二木が存在するかどうかを表す．詳しくは次のように定義される．なお $f_{s,t}, f_s^t, f_t^s$ はソース w を含むグラフ G_u に対して定義されて， $g_s^s, g_s^t, g^{s,t}$ はソース w を含まないグラフ G_u に対して定義される．

(i) $f_{s,t}(G_u, x, y)$

図7(a)のように G_u にはソース w が含まれるとし， G_u

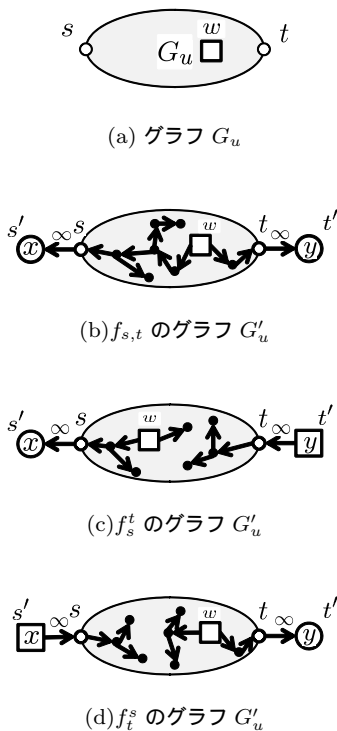


図 7 $f_{s,t}$, f_s^t , f_t^s の説明図

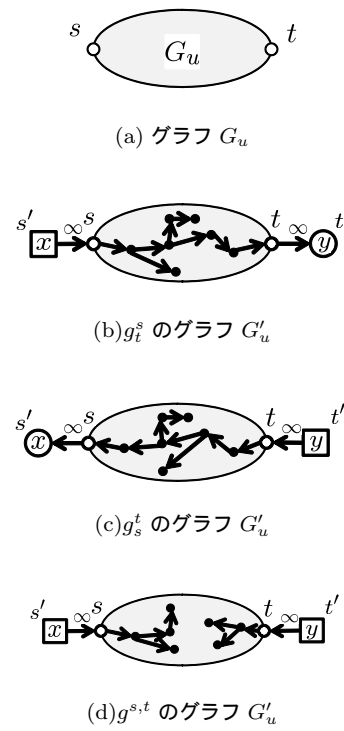


図 8 g_t^s , g_s^t , $g^{s,t}$ の説明図

の端子を s, t とする．図 7(b) のように， G_u に仮のシンク s' と t' を付加し，それらの需要量を各々 $x, y \in \mathbb{Z}_D$ とし，各々 G_u の端子 s, t と辺で結び，それら 2 本の辺の容量を ∞ とする．こうして得られた（直並列）グラフ G'_u に全域配送木 T が存在するならば， $f_{s,t}(G_u, x, y) = 1$ であり，存在しないならば $f_{s,t}(G_u, x, y) = 0$ であると関数 $f_{s,t}$ を定義する．直感的に言えば， $f_{s,t}(G_u, x, y) = 1$ は， G_u の端子 s および t からフローを各々 x および y だけ出力できる全域配送木が G_u に存在することを意味する．なお， $x \notin \mathbb{Z}_D$ あるいは $y \notin \mathbb{Z}_D$ のときには，便宜上 $f_{s,t}(G_u, x, y) = 0$ と定義する．他の 5 つの関数も同様である．

(ii) $f_t^s(G_u, x, y)$

図 7(c) のように， G_u に需要量が x の仮のシンク s' を付け加え，端子 s と容量 ∞ の辺で結ぶ．また供給量が y の仮のソース t' を付け加え，容量 ∞ の辺で端子 t と結ぶ．こうして得られたグラフを G'_u とすると， G'_u にはソースが 2 個ある． G'_u の全域配送二木で，一つの木が点 w, s, s' を含み，もう一つの木が点 t, t' を含むようなものが存在するならば， $f_t^s(G_u, x, y) = 1$ であり，存在しないならば $f_t^s(G_u, x, y) = 0$ であると関数 f_t^s を定義する．直感的に言えば， $f_t^s(G_u, x, y) = 1$ は，端子 t からフローを y だけ入力されれば端子 s から x だけ出力できるような G_u の全域配送二木が存在することを意味する．

(iii) $f_t^s(G_u, x, a, b)$

f_t^s の定義で端子 s と t の役割を交換したものが f_t^s である．詳しくは，図 7(d) のように， G_u に需要量が y の仮の

シンク t' を付け加え，端子 t と容量が ∞ の辺で結ぶ．また供給量が x の仮のソース s' を付け加え，容量 ∞ の辺で端子 s と結ぶ．こうして得られたグラフ G'_u の全域配送二木で，一つの木が点 s, s' を含み，もう一つの木が点 w, t, t' を含むものが存在するならば， $f_t^s(G_u, x, y) = 1$ であり，存在しないならば $f_t^s(G_u, x, y) = 0$ であると f_t^s を定義する．

(iv) $g_t^s(G_u, x, y)$

図 8(a) のように G_u にソース w はないとする．図 8(b) のように， G_u に仮のソース s' とシンク t' を付加し， s' の供給量を x とし， t' の需要量を y とする． s' と t' の各々を G_u の端子 s, t と辺で結び，それらの容量を ∞ とする．こうして得られたグラフ G'_u に全域配送木が存在するならば， $g_t^s(G_u, x, y) = 1$ であり，存在しないならば $g_t^s(G_u, x, y) = 0$ であると g_t^s を定義する．直感的に言えば， $g_t^s(G_u, x, y) = 1$ は，端子 s からフローを x だけ入力されれば，端子 t から y だけ出力できるような配送全域木が G_u にあることを意味する．

(v) $g_s^t(G_u, x, y)$

図 8(c) のように， g_t^s の端子 s と t の役割を交換したものが g_s^t である．

(vi) $g^{s,t}(G_u, x, y)$

図 8(d) のように， G_u に仮のソース s' と t' を付加し，それらの供給量を x と y とし，それぞれ端子 s と t と容量 ∞ の辺で結ぶ．こうして得られたグラフ G'_u に，点 s' と s を含む木と点 t' と t を含む木からなる全域配送二木が存在するならば， $g^{s,t}(G_u, x, y) = 1$ であり，存在しないならば

$g^{s,t}(G_u, x, y) = 0$ であると $g^{s,t}$ を定義する。直感的に言えば、 $g^{s,t}(G_u, x, y) = 1$ は、端子 s からフローを x だけ入力され、端子 t から y だけ入力されれば、 G_u に全域配送二木が存在することを意味する。

3.3 アルゴリズム

グラフ G_u の端子 s や t はソース w であるかもしれないし、需要量が正のシンクであるかもしれないが、どちらも需要量が 0 であるシンクであるとした仮想的グラフを G_u^* と書くことにする。記述の便宜上、本文のアルゴリズムは二分分解木 T_{BD} の葉から根に向かって、この仮想的グラフ G_u^* に対して 6 つの関数を計算する。なお G_u^* に対する関数から G_u に関する関数は容易に計算できる。

(a) 全域配送木の存在判定方法

図 5 のように、与えられた直並列グラフ G の二分分解木 T_{BD} の根を r とすると、 $G = G_r$ である。 G_r^* に対して 6 つの関数が計算されているとき、 G に全域配送木があるかどうかは次のように判断できる。

$G (= G_r)$ の端子 s と t がどちらもシンクのときには、 $f_{s,t}(G^*, \text{dem}(s), \text{dem}(t)) = 1$ ならばグラフ G には全域配送木が存在し、 $f_{s,t}(G^*, \text{dem}(s), \text{dem}(t)) = 0$ ならばグラフ G には全域配送木は存在しないと判断できる。また、 s と t の片方、例えば s がソース w のときには、 $g_t^s(G^*, \text{sup}(w), \text{dem}(t)) = 1$ ならばグラフ G には全域配送木が存在し、 $g_t^s(G^*, \text{sup}(w), \text{dem}(t)) = 0$ ならばグラフ G には全域配送木は存在しないと判断できる。

(b) 二分分解木の葉 u での関数の計算方法

グラフ G の二分分解木 T_{BD} の点 u が葉であるとき、 G_u^* に対して 6 つの関数を計算する方法を示す。図 2(a) のように G_u は 1 本の辺 $e = (s, t)$ からなる。 G_u^* にはソースが含まれないので、 $f_{s,t}, f_s^t, f_t^s$ は G_u^* に対して定義されない。 $x, y \in \mathbb{Z}_D$ に対して $g_t^s, g_s^t, g^{s,t}$ は次のように計算できる。

$$g_t^s(G_u^*, x, y) = \begin{cases} 1 & \text{if } y \leq x \text{ and } y \leq \text{cap}(e); \\ 0 & \text{otherwise;} \end{cases} \quad (1)$$

$$g_s^t(G_u^*, x, y) = \begin{cases} 1 & \text{if } x \leq y \text{ and } x \leq \text{cap}(e); \\ 0 & \text{otherwise;} \end{cases} \quad (2)$$

$$g^{s,t}(G_u^*, x, y) = 1. \quad (3)$$

(c) T_{BD} の内点 u が並列接続に対応しているときの関数の計算方法

次にグラフ G の二分分解木 T_{BD} の内点 u が並列接続に対応しているときに 6 つの関数を計算する方法を示す。 u の子を u_1, u_2 とし、 u_1 に対応する直並列グラフを G_1 とし、 u_2 に対応する直並列グラフを G_2 とする。このとき図 2(d) のように G は G_1 と G_2 を並列接続して得られる。

(i) まず $f_{s,t}(G_u^*, x, y)$ の計算方法を示す。 G_u^* にソース w が含まれていて、 w は G_u^* の端子 s, t ではないとしてよい。 w は G_1 に含まれるとしよう。(w が G_2 に含まれる場合も同様である。) 関数 $f_{s,t}(G_u^*, x, y)$ を定義するときのグラフ G_u^* の全域配送木を T とすると、むしろ T には閉路がないので、 T のフローのパターンはある整数 $a, b \in \mathbb{Z}_D$ に対して次の 3 つの場合がある。

- (a) 図 9(a) のように、 G_1 の外に端子 s と t から各々 $x+a$ と $y+b$ だけフローが出力する。また、 G_2 の中に端子 s と t から各々 a と b だけフローが入力される。
- (b) 図 9(b) のように、 G_1 の外に端子 s から $x+a$ だけフローが出力し、 G_1 の中に端子 t から b だけ入力する。また、 G_2 には端子 s から a だけ入力し、端子 t から $y+b$ だけフローが出力する。
- (c) 上の (b) で端子 s と t の役割を交換した場合である。詳しくは図 9(c) のように、 G_1 の外に端子 t から $y+b$ だけ出力し、 G_1 の中に端子 s から a だけ入力する。また、 G_2 には端子 t から b だけ入力し、端子 s から $x+a$ だけフローが出力する。

したがって、次 (a),(b) あるいは (c) が成り立つような $a, b \in \mathbb{Z}_D$ が存在するとき、かつそのときに限り $f_{s,t}(G_u^*, x, y) = 1$ と計算できる。

$$(a) \quad f_{s,t}(G_1^*, x+a, y+b) = 1 \text{ and } g^{s,t}(G_2^*, a, b) = 1 \quad (4)$$

$$(b) \quad f_s^t(G_1^*, x+a, b) = 1 \text{ and } g_t^s(G_2^*, a, y+b) = 1 \quad (5)$$

$$(c) \quad f_t^s(G_1^*, a, y+b) = 1 \text{ and } g_s^t(G_2^*, x+a, b) = 1 \quad (6)$$

(ii) 次に $f_s^t(G_u^*, x, y)$ の計算方法を示す。 w は G_1 に含まれるとしよう。図 10 のように、 $f_s^t(G_u^*, x, y)$ を定義するときのグラフ G_u^* の全域配送二木においては、ある $a, b \in \mathbb{Z}_D$ に対し、 G_1 の外に端子 s から $x+a$ のフローが出力し、 G_1 の中に端子 t から $y-b$ だけ入力する。また、 G_2 の中に端子 s と t から各々 a と b だけフローが入力される。したがって、ある $a, b \in \mathbb{Z}_D$ に対し次式が成り立つとき、かつそのときに限り $f_s^t(G_u^*, x, y) = 1$ と計算できる。

$$f_s^t(G_1^*, x+a, y-b) = 1 \text{ and } g^{s,t}(G_2^*, a, b) = 1 \quad (7)$$

(iii) $f_t^s(G_u^*, x, y)$ は $f_s^t(G_u^*, x, y)$ と同様にして計算できる。

(iv) 次に $g_t^s(G_u^*, x, y)$ の計算方法を示す。 G_u^* に w が含まれていないとしてよい。 $g_t^s(G_u^*, x, y)$ を定義するときのグラフ G_u^* の全域配送木 T は、ある $a, b \in \mathbb{Z}_D$ に対し次の 2 つの場合がある。

- (a) 図 11(a) のように、 G_1 の中に端子 s から $x-a$ だけのフローが入力され、 G_1 の外に端子 t から $y+b$ だけの

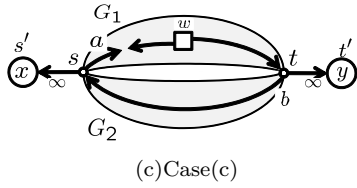
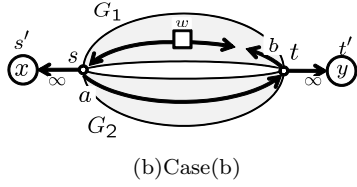
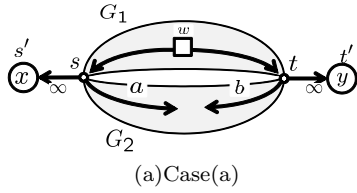


図 9 $f_{s,t}(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送木

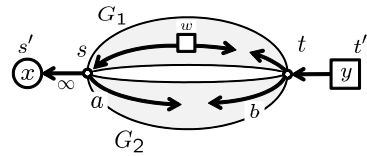


図 10 $f_s^t(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送二木

フローが出力される．一方， G_2 には端子 s から a だけのフローが入力され，端子 t から b だけのフローが入力される．

- (b) 図 11(b) のように， G_2 の中に端子 s から $x - a$ だけのフローが入力され， G_2 の外に端子 t から $y + b$ だけのフローが出力される．一方， G_1 には端子 s から a だけのフローが入力され，端子 t から b だけのフローが入力される．

したがって，ある $a, b \in \mathbb{Z}_D$ に対し次の (a) あるいは (b) が成り立つとき，かつそのときに限り $f_{s,t}(G_u^*, x, y) = 1$ と計算できる．

$$(a) \quad g_t^s(G_1^*, x - a, y + b) = 1 \text{ and } g^{s,t}(G_2^*, a, b) = 1 \quad (8)$$

$$(b) \quad g_s^t(G_1^*, a, b) = 1 \text{ and } g_t^s(G_2^*, x - a, y + b) = 1 \quad (9)$$

(v) $g_s^t(G_u^*, x, y)$ は $g_t^s(G_u^*, x, y)$ と同様にして計算できる．

(vi) 次に $g^{s,t}(G_u^*, x, y)$ の計算方法を示す． $g^{s,t}(G_u^*, x, y)$ を定義するときのグラフ $G_u^{*'}$ の全域配送二木を図 12 に示す．ある $a, b \in \mathbb{Z}_D$ に対し次式が成り立つとき，かつそのときに限り $g^{s,t}(G_u^*, x, y) = 1$ と計算できることは明らかである．

$$g^{s,t}(G_1^*, a, b) = 1 \text{ and } g^{s,t}(G_2^*, x - a, y - b) = 1 \quad (10)$$

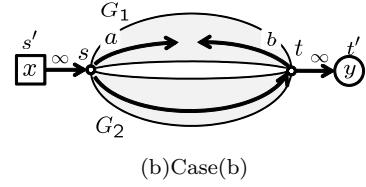
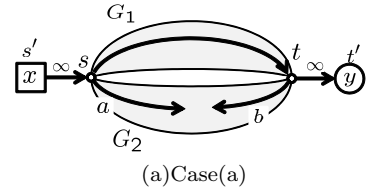


図 11 $g_t^s(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送木

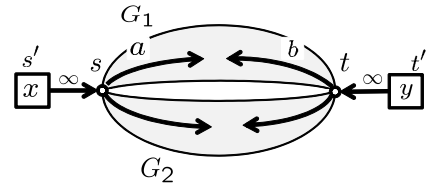


図 12 $g^{s,t}(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送二木

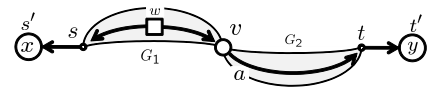


図 13 $f_{s,t}(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送木

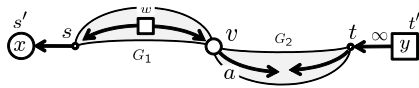
(d) T_{BD} の内点 u が直列接続に対応しているときの関数の計算方法

二分分解木 T_{BD} の内点 u が直列接続に対応しているときの計算方法を示す． u の子を u_1, u_2 とし， u_1 に対応する直並列グラフを G_1 とし， u_2 に対応する直並列グラフを G_2 とすると，図 2(c) のように G は G_1 と G_2 を直列接続して得られる． G_1 の端子 t_1 と G_2 の端子 s_2 を同一視して得られた点を v とすると， v は G においてシンクあるいはソースである．まず， G において接続点 v はシンクである場合を考える． G_u^* においては， v は端子でないで，むしろ v はシンクであり，その需要量は $\text{dem}(v)$ であり，必ずしも 0 ではない．

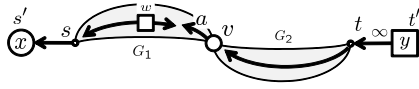
(i) まず $f_{s,t}(G_u^*, x, y)$ の計算方法を示す． G_u^* にソース w が含まれているとしてよい． w は G_1 に含まれるとしてよい．(w が G_2 に含まれる場合も同様である．) しかも $w \neq s, v$ である． $f_{s,t}(G_u^*, x, y)$ を定義するときのグラフ $G_u^{*'}$ とその全域配送木 T を図 13 に示す．ある $a \in \mathbb{Z}_D$ に対し次式が成り立つとき，かつそのときに限り $f_{s,t}(G_u^*, x, y) = 1$ と計算すればよい．

$$f_{s,t}(G_1^*, x, \text{dem}(v) + a) = 1 \text{ and } g_t^s(G_2^*, a, y) = 1 \quad (11)$$

(ii) 次に $f_s^t(G_u^*, x, y)$ の計算方法を示す． w は G_1 に含まれるとしよう． $f_s^t(G_u^*, x, y)$ を定義するときのグラフ $G_u^{*'}$ の全域配送二木は，ある $a \in \mathbb{Z}_D$ に対し図 14 (a) および



(a) Case(a)



(b) Case(b)

図 14 $f_s^t(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送二木

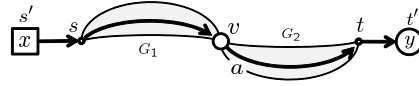


図 15 $g_t^s(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送木

(b) に示す 2 つの場合がある . したがって , ある $a \in \mathbb{Z}_D$ に対し次の (a) あるいは (b) が成り立つとき , かつそのときに限り $f_s^t(G_u^*, x, y) = 1$ と計算できる .

$$(a) \quad f_{s,t}(G_1^*, x, \text{dem}(v) + a) = 1 \text{ and } g^{s,t}(G_2^*, a, y) = 1 \quad (12)$$

$$(b) \quad f_s^t(G_1^*, x, a) = 1 \text{ and } g_s^t(G_2^*, a + \text{dem}(v), y) = 1 \quad (13)$$

(iii) $f_t^s(G_u^*, x, y)$ は $f_s^t(G_u^*, x, y)$ と同様にして計算できる .

(iv) 次に $g_t^s(G_u^*, x, y)$ の計算方法を示す . G_u^* に w が含まれていないとしてよい . $g_t^s(G_u^*, x, y)$ を定義するときのグラフ $G_u^{*'}$ とその全域配送木を図 15 に示す . ある $a \in \mathbb{Z}_D$ に対し次式が成り立つとき , かつそのときに限り $g_t^s(G_u^*, x, y) = 1$ と計算すればよい .

$$g_t^s(G_1^*, x, \text{dem}(v) + a) = 1 \text{ and } g_t^s(G_2^*, a, y) = 1 \quad (14)$$

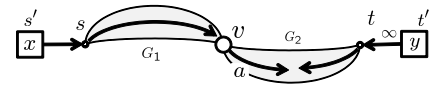
(v) $g_t^s(G_u^*, x, y)$ は $g_t^s(G_u^*, x, y)$ と同様にして計算できる .

(vi) 次に $g^{s,t}(G_u^*, x, y)$ の計算方法を示す . G_u^* にソース w が含まれていないとしてよい . $g^{s,t}(G_u^*, x, y)$ を定義するときのグラフ $G_u^{*'}$ の全域配送二木は , ある $a \in \mathbb{Z}_D$ に対し図 16 (a) および (b) に示す 2 つの場合がある . したがって , ある $a \in \mathbb{Z}_D$ に対し次の (a) あるいは (b) が成り立つとき , かつそのときに限り $g^{s,t}(G_u^*, x, y) = 1$ と計算できる .

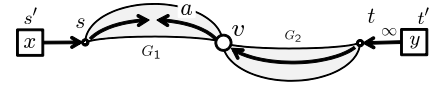
$$(a) \quad g_t^s(G_1^*, x, \text{dem}(v) + a) = 1 \text{ and } g^{s,t}(G_2^*, a, y) = 1 \quad (15)$$

$$(b) \quad g^{s,t}(G_1^*, x, a) = 1 \text{ and } g_s^t(G_2^*, \text{dem}(v) + a, y) = 1 \quad (16)$$

最後に G_1 と G_2 の接続点 v がソース w であるときの計算方法を示す .



(a) Case(a)



(b) Case(b)

図 16 $g^{s,t}(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送二木

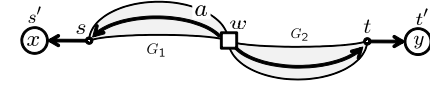


図 17 $f_{s,t}(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送木

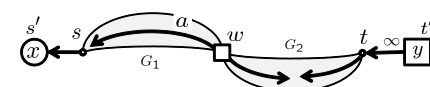


図 18 $f_s^t(G_u^*, x, y)$ のグラフ $G_u^{*'}$ の全域配送二木

(i) まず $f_{s,t}(G_u^*, x, y)$ の計算方法を示す . $f_{s,t}(G_u^*, x, y)$ を定義するときのグラフ $G_u^{*'}$ の全域配送木を図 17 に示す . ある $a \in \mathbb{Z}_D$ に対し次式が成り立つとき , かつそのときに限り $f_{s,t}(G_u^*, x, y) = 1$ と計算すればよい .

$$g_t^s(G_1^*, x, a) = 1 \text{ and } g_t^s(G_2^*, \text{sup}(w) - a, y) = 1 \quad (17)$$

(ii) 次に $f_s^t(G_u^*, x, y)$ の計算方法を示す . $f_s^t(G_u^*, x, y)$ を定義するときのグラフ $G_u^{*'}$ の全域配送二木を図 18 に示す . ある $a \in \mathbb{Z}_D$ に対し次式が成り立つとき , かつそのときに限り $f_s^t(G_u^*, x, y) = 1$ と計算すればよい .

$$g_t^s(G_1^*, x, a) = 1 \text{ and } g^{s,t}(G_2^*, \text{sup}(w) - a, y) = 1 \quad (18)$$

(iii) $f_t^s(G_u^*, x, y)$ は $f_s^t(G_u^*, x, y)$ と同様にして計算することができる .

3.4 計算時間

直並列グラフ G の二分分解木 T_{BD} の任意の葉 u に対して , 関数 $g_t^s, g_s^t, g^{s,t}$ は式 (1)–(3) を用いて $O(|\mathbb{Z}_D|^2) = O(D^2)$ 時間で計算することができる . ここで D は需要量の合計である . G は n 点からなり , G は単純直並列グラフであるとしてよいので , G の辺は高々 $2n - 3$ 本しかなく , T_{BD} には葉が高々 $2n - 3$ 枚しかない . したがって , T_{BD} の全ての葉に対して 6 つの関数を計算するのは $O(D^2 n)$ 時間でできる .

T_{BD} の任意の内部点 u に対して , 6 つの関数は式 (4)–(18) を用いて $O(|\mathbb{Z}_D|^4) = O(D^4)$ 時間で計算することができる . 二分木 T_{BD} には高々 $2n - 4$ 個の内点しかないので , T_{BD} の全ての内点に対し 6 つの関数を計算するのは $O(D^4 n)$ 時間でできる . このようにして次の定理が得られる .

定理 2 n 点からなる直並列グラフに対し全域配送木問題は $O(D^4n)$ で解ける .

4. 結論

本文では、まず全域配送木問題を定式化し、直並列グラフに対してすら NP 完全であることを示した . 次に点数を n 、需要量の合計を D としたときに、直並列グラフの全域配送木問題を $O(D^4n)$ 時間で解く擬多項式時間アルゴリズムを与えた . D が n の多項式以下ならば、そのアルゴリズムの計算時間は n の多項式であり、 D が定数ならば線形時間である . なお、本文のアルゴリズムを修正すれば、全域配送木を具体的に見つけることもできる .

本文のアルゴリズムを拡張すれば、ソースが 1 個とは限らず複数個ある直並列グラフに対する全域配送木問題も擬多項式時間で解くことができる . また、与えられたグラフ G が部分 k 木、即ち木幅が定数 k 以下であるときにも [6]、全域配送木問題や全域配送木問題を擬多項式時間で解くことができる .

与えられたグラフ G に全域配送木がないときには、ソース w を含む配送木 T で、 T に含まれるシンクの需要量の合計が最大なもの、即ち最大配送木を求めたい . 無論この最大配送木問題は直並列グラフに対してすら NP-困難である . ソースが複数ある場合には、同様にして最大配送木が定義される . G が木である場合には、辺容量がないとき [7] および辺容量があるとき [8] に対し、最大配送木を近似的に求める FPTAS が与えられている . また、辺容量がない直並列グラフや部分 k 木に対しては、最大配送木が擬多項式時間で求まる [6] . 今後は辺容量がある直並列グラフや部分 k 木に対して最大配送木問題や最大配送木問題を近似的に解く FPTAS を求めることが望まれる . なお、辺容量がない直並列グラフの最大配送木問題に対しては FPTAS が与えられている [5] .

謝辞 全域配送木問題の NP 完全性の証明に関し有益なコメントを頂いた今川廣二氏に感謝します . 本研究の一部は文部科学省私立大学戦略的基盤形成支援事業により実施された .

参考文献

- [1] N. G. Boulaxis and M. P. Papadopoulos, "Optimal feeder routing in distribution system planning using dynamic programming technique and GIS facilities," *IEEE Trans. on Power Delivery* 17, 1(2002), pp. 242-247.
- [2] C. Chekuri, A. Ene and N. Korula, "Unsplittable flow in paths and trees and column-restricted packing integer programs," *Proc. APPROX-RANDOM 2009*(2009), pp. 42-55.
- [3] Y. Dinitz, N. Garg and M. X. Goemans, "On the single-source unsplittable flow problem," *Combinatorica* 19, 1 (1999), pp. 17-41.
- [4] M. R. Garey and D. S. Johnson, "Computers and Intractability, A Guide to the Theory of NP-Completeness

- (Twenty-second printing)," W.H. FREEMAN AND COMPANY (2000), pp. 90-91.
- [5] T. Ito, E. D. Demaine, X. Zhou and T. Nishizeki, "Approximability of partitioning graphs with supply and demand," *Journal of Discrete Algorithms*, 6 (2008), pp. 627-650.
 - [6] T. Ito, X. Zhou and T. Nishizeki, "Partitioning graphs of supply and demand," *Discrete Applied Math.*, 157(2009), pp. 2620-2633.
 - [7] T. Ito, X. Zhou and T. Nishizeki, "Partitioning trees of supply and demand," *IJFCS* 16, 4(2005), pp. 803-827.
 - [8] M. Kawabata and T. Nishizeki, "Partitioning trees with supply, demand and edge-capacity," *Proc. of ISORA 2011, Lecture Notes in Operation Research*, Vol. 14 (2011), pp. 51-58, also *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science*, to appear.
 - [9] M. S. Kim, S. S. Lam and D.-Y. Lee, "Optimal distribution tree for internet streaming media," *Proc. 23rd Int. Conf. on Distributed Computing System(ICDCS '03)*, (2003), pp. 116-125.
 - [10] J. M. Kleinberg, "Single-source unsplittable flow," *Proc. of 37th FOCS* (1996), pp.68-77.
 - [11] A. B. Morton and I. M. Y. Mareels, "An efficient brute-force solution to the network reconfiguration problem," *IEEE Trans. on Power Delivery* 15, 3(2000), pp. 996-1000.
 - [12] T. Nishizeki, J. Vygen and X. Zhou, "The edge-disjoint paths problem is NP-complete for series-parallel graphs," *Discrete Applied Math.*, 115(2001), pp. 177-186.
 - [13] K. Takamizawa, T. Nishizeki and N. Saito, "Linear-time computability of combinatorial problems on series-parallel graphs," *J. Assoc. Comput. Mach.* 29 (1982), pp.6 23-641.
 - [14] J-H. Teng and C-N. Lu, "Feeder-switch relocation for customer interruption cost minimization," *IEEE Trans. on Power Delivery* 17, 1(2002), pp. 254-259.