日本オペレーションズ・リサーチ学会
2004年春季研究発表会

タンク繰りにおける経路探索法

豊橋技術科学大学工学部情報工学科 * 西垣 豊 NISHIGAKI Yutaka
豊橋技術科学大学工学部情報工学科 高橋 健吾 TAKAHASHI Kengo
02004044 豊橋技術科学大学工学部情報工学科 石井 利昌 ISHII Toshimasa
01403794 豊橋技術科学大学工学部情報工学科 永持 仁 NAGAMOCHI Hiroshi
昭和電工（株）石油化学事業部門 武田 真人 TAKEDA Makoto

1 はじめに

石油化学事業は，エチレンプラントの製品であるエチレン，プロピレン等が有機や合成樹脂等の誘導品プラントの原料となって，製品を産み出していく連産型の事業である．その中心であるエチレンプラントの競争力が石油化学事業全体へ及ぼす影響が最も大きいことから，エチレンプラントの運転や生産計画の最適化は重要な課題の一つである．

エチレンプラントは，原料であるナフサから熱分解，前蒸留，圧縮，蒸留の各工程を経て，エチレン，プロピレンなど種々の留分を生産する．これらの製品収率を決める最大の要因は，プラントに供給されるナフサの性状である．ナフサには多くの種類があり，その性状・購入価格は種類により異なる．そのため，製品価格，ナフサ価格，設備ネック等を考慮した，効率的な生産活動を行うために適切なナフサ調達計画，プラントへのナフサ供給計画の最適化が要求されている．

購入されたナフサは，プラントに供給される前に，一時的に複数のタンク (ナフサタンク) に受け入れられる．上記の通り，プラントに供給されるナフサの性状が，生産計画において重要な要因であるため，これらのナフサタンク間で異なる種類のナフサの移し変え（移液），混合などの操作を行うことで，適切な性状のナフサをプラントに供給できるようにすることが要求される．このとき，例えば，ある計画では，タンク A にあるナフサをタンク B に移し変え，タンク C にあるナフサをプラントに供給し，外部から購入されたナフサをタンク D に受け入れる，という操作を同時に行うという場合が想定される．実際に，この計画を遂行する際には，ナフサタンク配管網において，互いに独立な経路を確保する必要がある．

本稿では，昭和電工 (株) における事例を基に，このナフサタンク配管網における経路探索問題を解くアルゴリズムを提案する．我々は，ナフサタンク配管網を有向グラフに変換することで，この問題を，与えられた有向グラフと 2 点対集合 $\{\{s_i, t_i\} \mid i = 1, 2, \ldots, p\}$ に対して，互いに途中の節点を共有しない，s_i から t_i へのパス P_i の集合を求める問題として定式化する．しかし，このような独立なパス集合を（一通り）求める問題ですら，$p = 2$ の場合でも NP 困難である [1]．パスを全探索しながらパス集合を構築する分岐列挙アルゴリズムでは少し問題の規模が大きくなると莫大な計算時間を要する．そこで，経路木保存アルゴリズムと呼ばれる解法を提案する．このアルゴリズムでは，前処理として，各 2 点対 $\{a_i, b_i\}$ について，a_i から b_i への全てのパスを経路木と呼ばれる一つの木構造に保存しておき，独立なパス集合を構築する際に，これら経路木上でパスを再訪問しながらパスを集めてくる．この解法では，経路木上だけの探索をすればよいので，直接グラフ上でパスを見つけるより高速に解が見つかるという利点がある．本稿では，素朴な解法である分岐列挙アルゴリズムと提案する経路木保存アルゴリズムとの性能比較実験を行い，その有効性を検証する．

2 グラフ問題への定式化

ナフサタンク配管網は，ナフサタンク，プラント，購入されたナフサを受け入れるためのポート，ナフサが通る配管，弁，逆止弁 (一方向へしか通すことを許さないための弁)，ポンプから成る (図 1 参照)．配管網と，それを表わすグラフの節点集合，

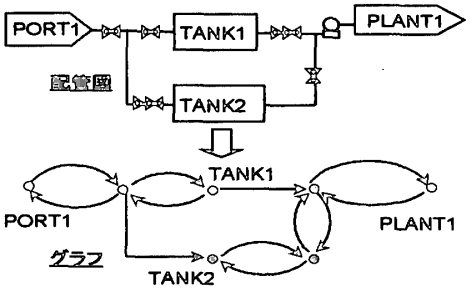

図 1: 配管網とそれを表わすグラフ

枝集合の対応関係は，それぞれ次の通りである．節点に対応づけられるのは，ナフサタンク，プラント，ポート，配管で，枝に対応づけられるのは，弁，逆止弁，ポンプである．このとき，弁，ポンプに対しては，接する二つの配管に対応する 2 節点を結ぶ有向枝が，両方向にそれぞれ 1 本ずつ対応づけられ，逆止弁に対しては，ナフサが流れる向きにのみ有向枝が 1 本対応づけられる．

$G = (V, E)$ を，V を節点集合，E を枝集合とする有向グラフとする．本稿では，以下の互いに独立なパス集合を求める問題を考える．

問題 1 (経路集合探索問題) **入力**: 有向グラフ $G = (V, E)$，始点終点のペアの集合 $\{\{s_i, t_i\} \subseteq V \mid i = 1, 2, \ldots, p\}$．
出力: 次の性質 $(i)(ii)$ を満たすパスの p 対 $(P_1, P_2, \ldots, P_p)$ の全ての集合 $\mathcal{P}$．
(i) 各 P_i は，s_i から t_i へのパスである．
(ii) どの 2 つのパス P_i, P_j も互いに途中の節点を共有しない．
□

配管網上で，始点，終点の候補となるのは，ナフサタンク，プラント，ポートである．問題 1 の出力は，(i),(ii) を満たすパス集合のすべての列挙としているが，実際には，現場での使用する弁や経路の取り方に優先度に照らした評価関数のもとで最良の１０集合を出力させる．

3 経路探索アルゴリズム

分岐列挙アルゴリズム，経路木保存アルゴリズムのいずれの方法も以下のアルゴリズム FindPath に基づいている．入出力は，問題 1 と同様である．FindPath は，まず P_1 から順に，$P_2, P_3, \ldots$ と順番にパスを固定していく．パス $P'_1, \ldots, P'_{i-1}$ が既に固定されたと仮定すると，パス P_i については，$P'_1, \ldots, P'_{i-1}$ の節点を通らないように，s_i から深さ優先探索を用いて見つける．また，$P'_1, \ldots, P'_{i-1}$ を含む実行可能解を全て見つけるまで，$P'_1, \ldots, P'_{i-1}$ を固定するという方法で，全ての実行可能解を列挙する．

アルゴリズム FindPath
Path$(p, s_1, t_1, P_1 = \{s_1\}, E_1 = \emptyset)$ を呼び出す．□

アルゴリズム Path$(p, s_i, t_i, P_i = \{u_i^1 = s_i, u_i^2, \ldots, u_i^j\}, E_i)$
/* パス P_i は，節点の順列で表わされ，節点集合としても扱う．
E_i は，探索済みの枝集合を表わす．*/
ステップ 1: 節点 u_i^j から出てる未探索の枝 $(u_i^j, u) \notin E_i$ を探索．$u \notin \cup_{k=1}^{i} P_k$ なる未探索枝があれば，ステップ 2 へ．なければ，ステップ 3 へ．
ステップ 2: $E_i := E_i \cup \{(u_i^j, u)\}$, $u_i^{j+1} := u$, $P_i := P_i \cup \{u^{j+1}\}$.
(2-1) $u_i^{j+1} \neq t_i$ なら，$j := j + 1$ としてステップ 1 へ．
(2-2) $u_i^{j+1} = t_i$ かつ $i \neq p$ なら，Path$(p, s_{i+1}, t_{i+1}, P_{i+1} = \{s_{i+1}\}, E_{i+1} = \emptyset)$ を呼び出す．
(2-3) $u_i^{j+1} = t_i$ かつ $i = p$ なら，$\mathcal{P} := \mathcal{P} \cup \{(P_1, P_2, \ldots, P_p)\}$, $P_p := P_p - \{u_p^{j+1} (= t_p)\}$ として，ステップ 1 へ．
ステップ 3: (3-1) $j = i = 1$ ならば，$\mathcal{P}$ を出力して終了．
(3-2) $j \neq 1$ ならば，$P_i := P_i - \{u_i^j\}$, $j := j - 1$ としてステップ 1 へ．
(3-3) $j = 1$ かつ $i \neq 1$ ならば，Path$(p, s_{i-1}, t_{i-1}, P_{i-1} - \{t_{i-1}\}, E_{i-1})$ を呼び出す．□

分岐列挙アルゴリズム 分岐列挙アルゴリズムは，FindPath を直接入力グラフ G に適用する解法である．

経路木保存アルゴリズム 我々の提案する経路木保存アルゴリズムは，直接グラフ G 上を探索しない．まず，前処理として，各ペア $\{s_i, t_i\}$ に対して，アルゴリズム Path$(p = 1, s_i, t_i, P_i = \{s_i\}, E_i = \emptyset)$ を実行し，始点 s_i から終点 t_i への全てのパスを表わす経路木 T_i を作成する．T_i の根は s_i に対応し，葉節点は全て t_i に対応する．T_i 上において s_i から葉節点へたどることで得られる節点の順列が，G 上の 1 つの s_i から t_i へのパスに一対一対応する．また，Path が深さ優先探索に基づいていることから，始点 s_i から途中までの順列が共通なパス同士をまとめて T_i に保持することができる．経路木保存アルゴリズムは，FindPath を前処理で得られた経路木集合上で適用することにより所望のパス集合を求める．

分岐列挙アルゴリズムでは，その時点で固定しているパス集合のみを保持しているだけで所望のパス集合を計算できる．これに対し，経路木保存アルゴリズムは，経路木集合を読み込んでから探索を行うため，各ペア間に存在するパス数が多ければ多いほど領域量を消費するという欠点があるが，一旦，経路木構造が格納できれば，余分な情報を前処理で取り除いているため，高速に所望のパス集合を構築できる利点がある．

4 実験結果

経路木保存アルゴリズムと分岐列挙アルゴリズムの性能を比較するため，昭和電工 (株) の配管図から変換した有向グラフを用いて計算機実験を行った．グラフの節点数，枝数は，ともにおよそ 100 程度である．始点終点ペア数を 1〜6 に設定した問題例をそれぞれ 10 例作り，これらに対する計算時間，メモリ消費量の平均を取った．これ以上のペア数の計測は分岐列挙アルゴリズムの計算に莫大な時間がかかった．また，経路木保存アルゴリズムはあらかじめ各ペアのパスを保持する経路木集合を生成してある．経路木集合を生成するのに要した時間は 4 秒程度であった．図 2 は，計算時間の比較を表わしている．

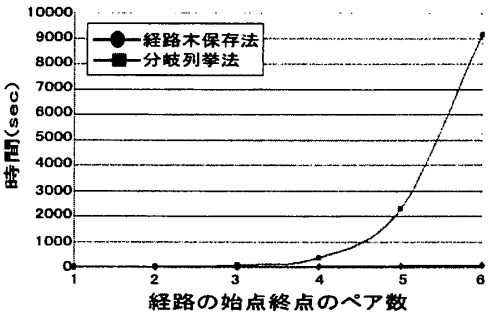

図 2: 計算時間の比較

ペア数が増加するにつれて，分岐列挙法の計算時間が指数関数的に増えていることが分かる．一方で，経路木探索法は，ペア数が増えても計算時間の増分は極めて少ない．図 3 は，計算時

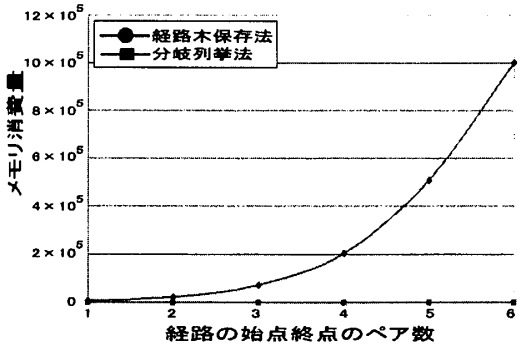

図 3: メモリ消費量の比較

間の比較を表わしている．縦軸のメモリ消費量は，木集合に使用されている節点数で評価している．経路木保存アルゴリズムのメモリ消費量は，ペア数が増えていくに従って，指数関数的に増加しているのが見てとれる．一方で，分岐列挙アルゴリズムでは，メモリ消費量の変動はほとんど見られない．

5 まとめ

今回，ページ数の都合上割愛したが，格子状グラフ，疎密グラフと呼ばれるグラフでも実験を行った．その結果，上と同様に，分岐列挙アルゴリズムでは計算時間，経路木保存アルゴリズムでは領域量が，それぞれ指数関数的に増加するという結果が得られた．今後の課題としては他の一般的なグラフに対しての性能評価，また計算時間と領域量のトレードオフを考えたアルゴリズムの開発などが挙げられる．

参考文献

[1] Y. Perl and Y. Shiloach, *Finding two disjoint paths between two pairs of vertices in a graph*, Journal of the Association for Computing Machinery, Vol. 25, No. 1, 1978, pp. 1–9.