

最小費用流アルゴリズムを用いたタンク繰りスケジューリング構成法

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 NAGAMUCHI 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

国内に数多く存在している石油化学コンビナートでは、原油を蒸留することで生じる中間留分であるナフサと呼ばれる化学原料から、エチレンプラントにおいていくつかの工程を経ることで、エチレン、プロピレン、ブタジエンといった種々の石油化学原料が生産される。ナフサの購入は主に海外からの船舶輸送による購入や、近隣の石油会社からのパイプラインを通じて行われる。購入したナフサは直接エチレンプラントにフィードされるわけではなく、送給量を調節するために一時的に複数のタンクに保存され、エチレンプラントにはタンクからほぼ一定の送給速度で断続的にフィードされる。

原料として購入するナフサは、原油の蒸留方法の違いなどによってその構成成分が大きく異なる。そのため、異なる構成成分を持つナフサそれぞれを銘柄名を与えて区別している。

エチレンプラントによって生産される化学原料の収率は、基本的に原料として与えられるナフサの構成成分によって決まる。したがって、タンク回り配管の弁やポンプを操作することによって、複数のタンク間によるナフサの移液や混合を行うことで、適切な構成成分を持つナフサをプラントに送給することが可能となる。

タンク繰りの作業スケジュールの立案においては、エチレンプラントに送給するナフサの量・構成成分を考慮し、生産される化学原料の収率の最大化を図ることが重要な課題となる。

本稿では、昭和電工 (株) における事例を基に、一か月分のタンク繰りスケジュールを計算するアルゴリズムを提案する。昭和電工 (株) のケースでは、原料となるナフサの購入は、主にナフサ船舶からシーバース (港) を通じての購入と、石油会社からパイプラインを通じての購入の 2 種類となる。ここではシーバースおよびパイプラインを合わせてポートと呼ぶことにする。またエチレンプラントへのフィードラインを単純にプラントと呼ぶことにする。このプラントは複数存在し、それぞれのプラントへ送給することの出来るタンクは限られている。

2 タンク繰りスケジューリング問題

タンク繰りスケジューリング問題は、ナフサタンク配管網、購入するナフサの銘柄および日付の情報、各銘柄ごとの月始めおよび月終わりのタンク内の在庫、各日ごとの各プラントに対するフィード量を入力として与えたとき、プラント、タンク、ポンプなどにおける制約を全て満たした実行可能な一ヶ月分のタンク繰りスケジュールを出力とする。この際、異なる銘柄が経路の一部を共有することは基本的に許されない。ポート、タンク、プラントの関係を示す図を図 1 に示す。

また、この問題の実行可能スケジュールとして満たすべき制約条件としては、各タンクの容量の制約、ポンプによる送給量上下限の制約、ナフサを流す経路に関する制約、プラントへの連続的送給の制約、シーバースからのナフサ給入に関する制約、

ナフサの構成成分によって使用できるタンクが限定される制約、タンク内に複数の銘柄のナフサが入った場合の挙動に関する制約などが挙げられる。

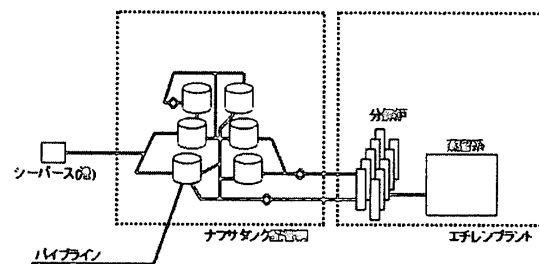


図 1: ポート、タンク、プラントの関係を表した図

ナフサを流す経路については、独立な 1 本の経路か、せいぜい途中で枝分かれしたり、逆に途中で合流するといったフォーク状の経路のみが許される。シーバースからの給入時の制約については、給入するナフサを受入れるタンクにおいて、受入れを始める一定時間前にはそのタンクに関する全ての操作を終わらせていなければならない制約と、受入れ終了後一定時間の間他の操作を行わず静置しなければならないという制約が存在する。

一番最後に挙げた、タンク内に複数の銘柄のナフサが入った場合の制約は、この問題特有のものといえる。ある銘柄のナフサがタンクに入っているとすると、このとき、その銘柄よりも比重の大きい銘柄のナフサがタンクに受入れられた場合は、受入れられた銘柄が下層、最初から入っていた銘柄が上層というように、タンク内で複数の層を形成する。この場合、タンクから取り出せるのは一番下の層のみとなる。逆に、タンク内の銘柄よりも比重の小さい銘柄のナフサがタンクに受入れられた場合は、2 種類の銘柄はタンク内で混合して 1 種類の新たなナフサとなる。このナフサの成分は 2 種類の銘柄の構成成分の加重平均となる。

この混合分離に関する制約が、この問題を非常に難しいものとしている。なぜならナフサが混合されるということは、その組み合わせおよび混合の比率によって、異なった構成成分を持つ無数の新しい銘柄のナフサが生成され得るからである。したがって、この問題は整数計画問題に定式化することすら困難であり、仮に定式化できたとしても、変数量が膨大な量となり、計算が不可能となることが容易に想像できる。

3 フローアルゴリズム

この問題をグラフ問題として定式化するために、まずはナフサタンク配管網をグラフ構造に置き換える。配管網の構成要素は、タンク、ポート、プラント、ナフサが通過する配管、ナフサをせき止めるための弁、逆止弁（一方向弁）、ナフサを送給するために必要なポンプから成る。このうち、タンク、ポート、プラント、配管を節点集合、弁、逆止弁、ポンプを枝集合に対応付ける。枝の張り方は、弁およびポンプについては両方向に1本ずつ、逆止弁については一方向に1本のみとなる。 $G = (V, E)$ を、 V を節点集合、 E を枝集合とする有向グラフとする。我々の提案したアルゴリズムは、ナフサの流れをネットワークフローで表現し、最小費用フローアルゴリズム [1] をベースとしている。

問題 1 (最小費用フロー問題)

入力: 有向グラフ $G = (V, E)$ 、各枝 $e \in E$ に対する、正の実数の容量 $cap(e)$ 、非負の実数の費用 $cost(e)$ 、入口(ソース)の集合 $S \subset V$ 、出口(シンク)の集合 $T \subset V$ 、 $s \in S$ に対する正の実数の送出量 $v(s)$ 、 $t \in T$ に対する正の実数の受入量 $v(t)$ 、を持つネットワーク $N = (G, cap, cost, S, T, v)$

出力: 各枝 $e \in E$ の流量が $cap(e)$ 以下となる、 S から T への費用最小となるフロー □

この最小費用フローを用いたアルゴリズムでは、与えられた銘柄集合に対する実行可能なスケジュールを得るために以下の方法を取る。

扱う b 種の銘柄を比重の大きな順にソートする。次に最初の銘柄から順番に各 i 番目の銘柄 m_i に対し、ナフサタンク配管網のグラフ $G = (V, E)$ 、各タンク h の月始めの在庫量 T_h^0 、月終わりの在庫量 T_h^{D+1} 、 d 日目の受け入れ口 k からの購入量 v_k^d 、 d 日目のプラントへフィード口 k への供給量 F_k^d 、ペナルティ情報 φ_i を用いて、ネットワーク N_i を構成し銘柄 m_i の最小費用フロー f_i を計算する (図 3 参照)。

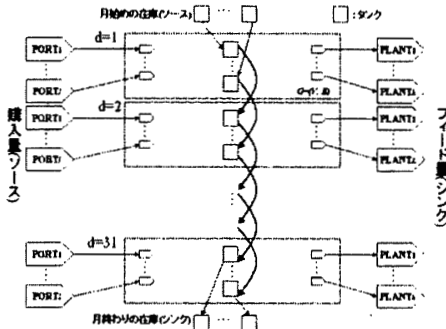


図 2: ネットワークモデル

この中のペナルティ情報 φ_i は入力として与えられるものではなく、 m_1, m_2, \dots, m_{i-1} の銘柄に対して求めた最小費用フロー f_1, f_2, \dots, f_{i-1} によって設定されるもので、経路の独立性やその他の様々な制約を満足させるために、フロー f_i の流れを制御するためにネットワーク N_i の構築時に各枝 e の $cost(e)$ に加算される。

これを一通りの全銘柄について行うことを、ひとつのラウンドと定義する。提案したアルゴリズムでは、ペナルティ情報を随時更新しつつこのラウンドを繰り返すことで、解 $\{f_1, f_2, \dots, f_b\}$ を少しずつ最適な実行可能解に近づけていく。ペナルティ情報は、異なる銘柄の流れがぶつからないように考慮しなければならないため、ラウンドを繰り返すことで次第に与えられた制約をクリアするように構築する。したがって、最初のラウンドで

は $cost(e)$ に加算されるペナルティコストは非常に小さく設定し、ラウンドを繰り返すにしたがって次第に大きくしていく。また、最後のラウンドでは、制約を確実に満足させるために、ペナルティ情報を基に枝の削除も行う。

アルゴリズム FindTankSchedule

/* M は銘柄の集合、 R はラウンド数の上限値を表す。

MakeNetwork はネットワークモデルを構築する関数、

MinCostFlow は最小費用フローを解く関数とする。

f_i には N_i 上の銘柄 m_i の最小費用フローが格納される。*/

ステップ 1: 与えられた銘柄集合 M を比重の大きい順に m_1, m_2, \dots, m_b とする。 $i := 1, r := 1$ 。

ステップ 2: $m_i \in M$ に対して

$N_i := \text{MakeNetwork}(G, T_i^0, T_i^{D+1}, v_i, F_i, \varphi_i)$ を実行。

ステップ 3: $f_i := \text{MinCostFlow}(N_i)$ 。

ステップ 4: f_i より P_e を更新。 $i \neq b$ なら $i := i + 1$ してステップ 2 へ。

ステップ 5: $r \neq R$ なら $i := 1, r := r + 1$ してステップ 2 へ。

ステップ 6: $\{f_1, \dots, f_b\}$ を経路情報に分解してスケジュール時間を割り当てる。

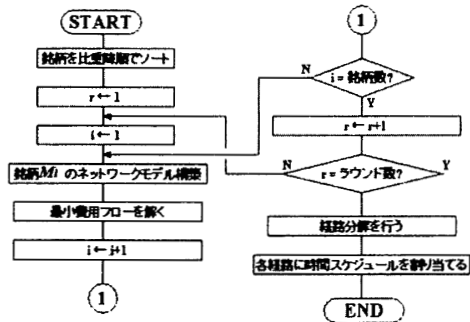


図 3: フローチャート

4 アルゴリズムの機能

前述した制約のうち、混合に関する制約以外は、ペナルティ情報をうまく設定することで、これらを満足するスケジュールリングを生成させることが出来る。また、タンク内の銘柄の層が余りに増えすぎるのは実用上望ましくなく、タンク内に生じる層の数を指定した層数 k 以下とする制約を加えてある。

混合に関する制約を解決するため、現段階では以下の仕様となっている。まず、基本的には混合が行われないようなスケジュールが計算される。混合を行う場合は、入力として混合を行う日付、タンク、各銘柄の種類および量を与えることで、指定箇所でのみ混合を行い、それ以外の箇所では混合が行われないスケジュールリングを計算する。

5 まとめ

フローアルゴリズムは実行可能解を見つけることが困難な例題に対して解を見逃すこともあり得るが、人手で運用してきた実績に対しては常に可能解を得ることができた。しかし、上述した通り、ナフサの混合に関しては混合が行われるタンクや日にちを問題の入力として指定する方法を取っている。実用上はこの部分も決定変数とする問題が扱えるほうが望ましい。現在、最適な混合条件をアルゴリズム側で自動的に選択させる方法を研究中である。

参考文献

- [1] 浅野孝夫, 情報と構造, 日本評論社, 1994.