

複数オーダーの同時積み付けを考慮した ケミカルタンカーの船舶スケジューリング

瀬田 剛 広^{†1}

陸上に於いて配送計画問題が OR の重要な問題として存在するように、海に於いても船舶のスケジューリング問題が存在し、重要な問題となっている。本報告では、そのような問題の 1 つの例である、ケミカルタンカーの船隊のスケジューリング問題を扱う。本問題は、既存研究において、既に解法が提案されているが、本報告で提案する定式化の見直しを行うことにより、複数オーダーの同時積み付け法に関してより細かく考慮することが可能となる。実際、実績値と比較して、5.2% の航行距離が削減される、という結果が得られ、本報告の提案する手法を既存研究の手法と組み合わせることにより、実用的な速度を維持しながらも、さらに良いスケジュールが立案出来る可能性が示された。

Scheduling Chemical Tankers, Considering Multi Order Pairing

SETA TAKAHIRO^{†1}

In the same way there exists the VRP as an important problem on land, there also exists the ship scheduling problem as an important problem on the sea. In this report, we consider a scheduling problem of chemical tanker fleet. The formulation we propose can treat how to pair orders in a more detailed way than an existing algorithm. Using this formulation, total travelling distance is reduced by 5.2% compared to the actual result. It is shown that the existing algorithm together with our formulation will make better schedules keeping a practical speed.

1. はじめに

配送計画問題といえば、OR の研究者なら誰も知っているような、一大テーマである。海の世界でも陸と同様、船舶スケジューリング問題（配船計画問題）が存在する。しかし、残念ながらその研究は陸における配送計画問題ほど進んでいるとは言いがたい。配送計画問題と全く同じで面白みが無い訳ではなく、船舶独自の特徴も多く存在するのにも関わらず、である。特に日本は周囲を海に囲まれ、造船等に於ける一流の技術を持ちながら、この分野の研究の事例は非常に少ない様である。（配船分野の研究動向については、サーベイ論文^{1),3)}が存在し、分かり易くまとめられている。いくつかの例について、比較的详细にまとめられた、日本語の文献⁵⁾も存在する。）

だが、環境問題を考えても、効率的な船舶の運航が行われることは、単に運航効率の面だけでなく、モーダルシフトの推進による環境負荷減少という二重の効果が期待され、社会的意義も大きい。また、幾つかの船会社の配船の様子を見ても、人手に頼る部分が非常に多く、近年の OR 分野の研究成果の反映余地が大き

い分野である。

本報告では、日本国内のタンカー船会社を例として船舶スケジューリングについて紹介するとともに、実際にそのスケジューリングを複数のオーダーの同時積み付け法も考慮しながら解くアルゴリズムを提案する。この問題については、すでに久保・小林⁴⁾が階層的積み木法と列生成法とを利用した手法を提案している。しかし、筆者には複数オーダーの同時積み付け法を定める部分の定式化については改善の余地があるように見受けられた。本報告はその手法を踏襲しながらも、定式化の形を多少変更し、複数オーダーの同時積み付け法をより細かく扱えるようにすることを目指すものである。

2. 問題の概要

本報告では、日本国内のとある不定期ケミカルタンカーの船会社を対象に船舶スケジューリングを行った。ここで簡単に船舶スケジューリングとはどのような問題か、またタンカー及び今回扱った問題にはどのような特徴があるかについて述べる。

まず、船舶スケジューリングとは、ある船会社の支配下の複数の船舶（船隊と呼ぶ）のある期間の運航計画を立てるものである。対象とする期間などでいく

^{†1} 独立行政法人 海上技術安全研究所（海技研, NMRI）
National Maritime Research Institute; Japan

つかに分類される¹⁾が、中でも本報告で扱った事例は「不定期船」を対象とし、「オーダー式」のものである。これは、船ごとに決まった航路に従って長期間運航を行うモデルではなく、荷主から、「どのような荷物(What)を、どれくらいの量(How much)、どこからどこまで(Where)、いつからいつまで(When)に運んで欲しい」という要望(オーダー)を受け、それに従って支配下の船舶の運航を定めるものである。問題設定にもよるが、要望を全て受けきれない場合は多少割高になるが、他の船会社等から船の融通(本報告で扱われる「スポット備船」と呼ばれる形などがある)を受けたり、逆に船舶が余っている場合は、他社に貸し出したり、本来の輸送対象の荷物以外のものを運んだりというようなこともある。このような問題の場合、計画期間としては数日から長くても1ヶ月程度が対象となる。なお、船舶の場合、1航海に数日を要する。また、船舶の運航には気象・海象の影響による遅延・早着が多発する。そのため、時刻の指定を行う場合も人間が配船を行う場合は、細かくても数時間単位、今回対象とした船会社では、1日を午前と午後と2つに分けるといった程度の粗い指定となる。同時に、一度立てた計画も頻繁に更新することが求められる。もう1つ船舶の特徴として、基本的に船舶は高価な物で、またオーダーメイドで造られるため、基本的には船隊内の船それぞれ、性能が異なるということがある。そのため、計画時は船を1隻1隻区別して扱うことが求められる。

次に、本報告で扱うケミカルタンカーの特徴について述べる。ケミカルタンカーは、船体内に「ホールド」と呼ばれる区画(タンク)を複数持っている。今回扱った例の場合、1隻あたり4または5区画のホールドを持っていた。船の形状が単純な直方体では無いということもあり、同じ船のホールドでも一般に容量は異なる。一般的にケミカルタンカーが運ぶ製品(揮発性の油製品など)は液体であり一度混ざると分離出来ないため、異なる品種を同じホールドには積載することは出来ない。また、同じ製品でも、荷主や向け先の違いなどがあれば、同様に混ぜることは出来ず、別の製品として扱うことが求められる。本報告のテーマからはやや外れるので詳細は割愛するが、他にも、揮発性の油という危険物を扱うということから、荷役作業は日中にしか行えないように法律で決まっている、というような制約条件がいくつか存在し、スケジューリングを行うアルゴリズムには制約条件を柔軟に取り込める性質が強く求められる。

最後に、特に本報告で扱う問題設定について述べる。本報告では、ある船会社の約1ヶ月分の実際の船の運航実績を元にテストデータを作成した。運航実績で同じ船舶で輸送された複数の製品については、1つ1つ分離し、それぞれをオーダーとし、この分離されたオー

ダーの同時積み付け法(ペアリング方法)を計算する部分も問題の一部として問題設定を行った。実際の運航においては船が傾くことが無い様、荷の積み付け位置にも注意を払うが、それは除外した。オーダーが要求する積日・揚日については、本来の要求が幅を持っていたことも考えられるが、実績の積日・揚日が要求されたものとしてテストデータを作成した。実際の運航においては、全く同じ時間に2隻の船が同時に同じ場所で荷役を行うことは許されないが、現実世界においても、荷役順序の調整は直前になって行うケースが多い様であることもあり、また、積日・揚日を固定したことから、実績でかち合わなかった船が配船結果でかち合うことも少ないと考えられるため、本報告の問題設定では考慮不要とした。

3. アルゴリズムと定式化

対象とする問題を解くにあたり、本報告では、久保・小林⁴⁾と同様、それぞれの船舶ごとに可能な運航経路を生成し、集合分割問題として全体のスケジュールを定める手法を採用した。本手法は船舶ごとの運航経路を生成する段階で主要な制約条件のチェックを全て行うことが出来るため、制約条件の追加が容易に行えるという特徴も持ち、船会社ごとに制約条件がいろいろ異なる船舶スケジューリングの問題に対し、望ましい手法であると考えられる。

船舶ごとに可能な運航経路を生成する部分については、集合分割問題の子問題として、列生成法を利用した。さらに、各船舶ごとに可能な運航経路を生成する部分の前段階として、オーダーの同時積み付け法(ペア)を求める部分(ペアリング問題)については、提案手法の特徴である同時積み付け法の考慮部分の効果を最大限発揮させるため、また、問題のサイズがそれ程大きくならなかったため、全列挙の手法を採用した。さらにペアリング問題の前段階である、オーダーのホールドへの割当て方法を求める部分についても全列挙の手法を採用した。

集合分割問題の定式化は以下に示す様になる。

$$\begin{aligned} \min. & \sum_{v \in V} \sum_{r \in R} C_r x_{rv} + \sum_{o \in O} F_o y_o \\ \text{s.t.} & \sum_{(r,v): o \in O_r} x_{rv} + y_o = 1 \quad \forall o \in O \\ & \sum_{r \in R_v} x_{rv} \leq 1 \\ & x_{rv} \in \{0, 1\} \quad \forall r \in R_v, v \in V \\ & y_o \in \{0, 1\} \quad \forall o \in O \end{aligned}$$

目的関数はそれぞれの船舶 v が選択した運航経路 r のコスト C_r の和に、さらに支配下船舶では処理出来なかったオーダー o をスポット備船で処理した場合のコスト F_o の和を加えたものが、総コストであり、それを最小化したいことを表す。現実にはスポット備船にか

かるコストが大きいことから、 F_r は C_r と比較して大きな値となり、スポット備船は出来るだけ少なくするように計算することになる。1番目の制約式は、全てのオーダー o が、支配下船がそのオーダーを含む経路 r を選ぶことによって、もしくは、スポット備船 y_o によって処理されることを表す。 O_r は運航経路 r に含まれるオーダーの集合である。運航経路自体はペアの列であり、ペアはオーダーの積または揚の列として構成される。2番目の制約式は、それぞれの船舶 v が選べる経路 r は高々1つということを表す。

既存手法⁴⁾との大きな違いは1番目の制約条件をオーダー単位で立式している所である。これにより、スポット備船を表す変数 y_o を同時に輸送するオーダーのペアの単位ではなく、それぞれのオーダー単位に持つことが出来る様になり、1つのオーダーを含むペアを複数用意することが可能となっている。

次に、ペアリング問題の部分であるが、この部分はオーダーの積と揚との列を空の状態から始め、1つずつオーダーの積荷役と揚荷役とを、現在の列に挿入していくという手法を用いた。オーダーの荷役をどの順番で行うかも重要な情報であるため、どのオーダーを挿入するかだけでなく、どの位置に挿入するかによっても分岐を行った。また、それぞれのオーダーのホールドへの割当て方法は複数あり、その方法も影響するため、ホールド割当て部分でも分岐を行った。オーダーを挿入する段階においては、荷の量がホールドに積載可能か、指定された時刻に間に合うかなどの制約条件を組み入れてペアを生成した。また、基本的には全列挙ではあるが、それが終わった段階、または途中の段階において、たとえば同じ港での揚荷役の順序の入れ替わりやホールド積み付けの方法等、以降の計算で等価になるものについては、どちらか一方のみを残すような処理を行った。

最後に、それぞれのオーダーのホールドへの割当て方法を求める部分についてだが、オーダーと船との全ての組合せについて、極小なホールド割当てを列挙し、ビットパターンの形で保存する方法をとった。基本的には全列挙ではあるが、やはり、極小なホールド割当てのみが意味を持つため、それだけを保存するようにした。本報告が扱った例の場合、ホールドの数は最も多い船舶でも5区画であったため、どのような方法で列挙しても問題なく全列挙可能であるが、極小な割当て方法のみを求めるため、ホールドを容量の降順にソートし、大きなホールドから順に使うか使わないかを決めていくことにより、列挙を行った。

4. 数値実験とその結果

この節では、船会社より頂いた、実際の運航データを用いて行った実験の結果を示す。

実際の運航の実績値、同時積み付け法を実際の運航

の組合せに固定した場合の結果、そして、本報告が提案している、複数オーダーの同時積み付けを計算する手法による結果の3つを比較した。コストとしては総運航距離を用いた。燃料消費量は船速や船の性能によっても変化するため、完全には運航距離に比例しない。しかし、ほぼ比例関係と近似しても良く、本報告の使った目的関数は、燃料消費量、すなわち運航コストの最小化を目的としていると言える。利用したテストデータに含まれる船舶は7隻、オーダーは全部で111件、期間は約1ヶ月(最初の積日から最後の揚日までで32日間)であった。船舶1隻あたりのホールドの数は前述の通り最大5区画、1航海あたりの期間は1日から4日まで幅があるが、平均して3日程度、同時に運ぶオーダー数も幅があるが最大4つであった。なお、計算環境としては、CPUがIntel Pentium4 3.4GHz、RAMが4GByteのWindows XPマシンに混合整数計画ソルバーとして、Dash Optimization社のXPRESS MP(2007B)を用い、C++でプログラムを作成、Microsoft社のVisual Studio 2005 Professional EditionのReleaseビルドでコンパイルした。

実験結果は表1に示すとおりである。元データとして実績データを用いたが、実績データをシミュレーションにかけた所、ヒアリングで得られた制約条件が厳しいものがあつたためか、1隻処理出来ないオーダーが発生する結果となった。提案手法は、ペアリングを再計算するため、このオーダーも処理することが可能であり、その結果を表の最下行に書いたが、その他の結果は、そのオーダーを除いた総航行距離となっている。結果として提案手法は実績値と比べて総運航距離で5.2%の削減を達成することが出来た。ペアを固定した場合は3.7%であり、提案手法の有効性が確認されたと言える。また、ペアを組替えることにより、実績値で必要になった備船を削減し、さらに総航行距離を減らすことにも成功している。

方法	総航行距離	実績値比	計算時間	備船数
実績値	32630	—	—	1
ペア固定	31423	-3.7%	32秒	1
提案手法	30932	-5.2%	927秒	1
提案手法	31342	-4.0%	524秒	0

計算に要した時間について言えば、15分または10分であり、他の作業をしていれば過ぎてしまう程度の時間なので、実際の運用を目的とした場合、1ヶ月分の計算をするためにかかる時間と考えれば、人間の作業よりはるかに高速であり、十分許容範囲と言える程度である。しかし、条件を変更して1日に複数回動作させることを考えたり、今後、業務が拡張された場合まで考えると高速化が求められる水準とも言える。

なお、全列挙により得られたペアの数は、不要なペ

アを削減した後の段階で 363 組であり、列生成によって作成された列の総数は船舶 7 隻の合計で、4648 列であった。

5. 既存手法との融合

本報告で行った実験では、ペアリング問題を解く際、この手法を使った場合の最大の効果を検証する目的があったため、また、問題がそれ程大きいものではなく、実行時間の問題も起きなかったため、ペアを全数生成した。その結果、計算時間の面に於いてはやや遅く、問題の大きさが大きくなれば限界が来ることが容易に予想される物となっている。それに対し、久保・小林の手法⁴⁾はペアリング問題に列生成法を利用することで、効率的にペアリング問題を解く方法を提案している。しかし、ペアリング問題を効率的に解くとともに、下に示す式の 2 行目が表す様に、1つのオーダーに対し、それを含むペアは 1 通りに限定するという形で制限を入れており、その制限が妥当であるかについては検証が必要であると考えられる。

そこで、本報告では今後の発展としてこれら 2 つの方法を融合したアルゴリズムを提案する。本報告に於いては、時間の都合上、実装し、実験を行う部分まで手が回らなかったが、これらの手法は以下の様にして、お互いの長所を生かしつつ、融合出来ると考えられる。

1つのオーダーに対し、たとえば、10 個程度のペアを認めるなら、既存手法において、各オーダーに対して、それを含む効率的なペアを求める定式化

$$\begin{aligned} \min. & \sum_{p \in P} C_p x_p + \sum_{k \in K} F_k y_k \\ \text{s.t.} & \sum_{p: k \in K_p} x_p + y_k = 1 \quad \forall k \in K \\ & x_p \in \{0, 1\} \quad \forall p \in P, y_k \in \{0, 1\} \quad \forall k \in K \end{aligned}$$

において、全てのオーダー k についてそれを含むペアがただ 1 つであることを示す 1 番目の制約式を次の様に変更する。

$$\sum_{p: k \in K_p} x_p + y_k \geq 10, \quad 0 \leq y_k \leq 10 \quad \forall k \in K$$

これにより、それぞれのオーダー k に対し、それを含むペア x_p を 10 個以上残すことが可能となる。10 個以上としたが、目的関数の最小化の条件があるため、ほぼ、10 個になると予想される。もちろん、 \geq の代わりに $=$ を用いても構わない。 y_k がバイナリ変数で無くなることで多少、計算時間が増大することも考えられるが、基本的に y_k は、 x_p で足りない分を埋めるための変数であり、整数制約さえ不要な変数なので、速度低下は軽微であると期待される。また、列生成のための子問題にも小さな変更が必要になるが、基本的な効率性を失わずに、ペアを複数生成することが出来ると考えられる。もしくは、列生成の時点ではペアをただ 1 つ選ぶ、元の定式化を利用して行うことも考え

られ、全く変更を行わない事も可能と考えられる。

6. まとめ

本報告では、船舶スケジューリング問題について紹介するとともに、特に複数オーダーの同時積み付けを扱う場合の定式化について提案し、実データを用いた実験結果を示すことにより、その有効性を確認した。また、本報告の定式化と既存手法とそれぞれの長所を組み合わせることで、さらに良いアルゴリズムが構築される可能性について指摘した。

今後の課題としては、5 節で指摘した既存手法との融合を実装・実証すること、また、数値実験についてはより多くのデータを用いて行うことが挙げられる。今後、これらを行い、その結果については改めて報告したいと考えている。

謝辞 この研究は独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) の先導研究開発「内航船の環境調和型運航支援計画支援システムの研究開発」の一部として行ったものである。また、貴重なデータを提供して下さいた船会社の方には、この場を借りて改めて感謝を申し上げる。

参考文献

- 1) Christiansen, M., Fagerholt, K. and Ronen, D.: Ship Routing and Scheduling: Status and Perspectives, *Transportation Science*, Vol.38, No.1, pp.1-18 (2004).
- 2) Kim, S.-H. and Lee, K.-K.: An Optimization-based Decision Support System for Ship Scheduling, *Computers and Industrial Engineering*, Vol.33, No.3-4, pp.689-692 (1997).
- 3) Ronen, D.: Ship scheduling: The last decade, *European Journal of Operational research*, Vol.71, pp.325-333 (1993).
- 4) 久保幹雄, 小林和博: 階層的積木法と列生成法の融合 - 輸送・船舶スケジューリングを例として -, 計測と制御, Vol.47, No.6, pp.519-524 (2008).
- 5) 小林和博, 久保幹雄: 船舶スケジューリング, 第 20 回 RAMP シンポジウム論文集, pp. 61-75 (2008).