

自動車運搬船における貨物積載プランニングの席割問題

竹田 陽

2021 年 4 月 21 日

Abstract. Our research considers the situation in which cars are transported to several ports of various countries by a carrier ship for cars (pure car carrier, PCC). Two stages of work called *stowage planning* and *simulation planning* are performed before loading a collection of cars onto a PCC. The stowage planning stage considers assigning cars to one of the areas called *holds*, a few of which compose each level of the ship. The simulation planning stage places the cars assigned to each area in the stowage planning one by one considering the orientation of cars. In our research, as the first step of fully automating the stowage planning and simulation planning, the goal is to output stowage plans efficiently with a computer using mathematical optimization technology.

1 はじめに

複数の港で荷物を船に積み、複数の港で降ろす運搬船を考える。積荷の種類としては燃料、原料など多岐に渡るが本研究では自動車を運搬する船を考える。一般的に自動車運搬船は、自動車を船の一定間隔で区切られたホールドと呼ばれるスペースにどの自動車を何台割り当てるかを考える席割作業を行い、その後席割作業で割り当てられた自動車に対して向きと場所考慮して一台ずつ船内の領域に配置するシミュレーション作業をする。現状自動車を輸送する会社はこの作業を手で行っているが少なくとも席割作業に 3 時間、シミュレーション作業に 4 時間かかることから、多大な人件費と時間をかける必要があり直前の追加注文等に対応が出来ないという問題点がある [1]。本研究ではこの問題を解決するために、席割作業とシミュレーション作業それぞれについて数理最適化技術を用いることで、コンピューター計算により短時間で有効な席割とシミュレーションを出力することを目標とする。ただし本稿では研究の第一段階として、席割作業とシミュレーション作業のうち席割作業の自動化に対して検証した結果を記す。

本研究で扱う席割作業の概要について述べる。入力としてプリウスやハイエースなどの乗用車や、クレーン車やブルドーザーのような建機等の様々な種類の

車を港 A から港 B まで輸送せよというような積載自動車の注文リストを受け取る。注文リストを受け取ったプランナーと呼ばれる席割を作成する作業者は好ましい席割になるように、注文の船内スペースへの割当を考える。席割作成における前提条件について説明する。例えば船内の特定の領域に積まれている自動車よりも奥に積まれている自動車が、ある港 C で降ろされて空きスペースが発生したとする。この場合次の港 D で積む自動車があればその自動車の積みやすさのために、船内に積んだ自動車を奥側に詰めて手前の領域を確保するというような既に積まれた自動車の航海中の移動は原則しない。また自動車は全てのホールドに容易にアクセスすることは出来ない。例えば本稿の実験で扱う自動車運搬船は 12 階構造の内 5 階のみに外部から自動車を入れるスロープがあり、船内の奥側ホールドにアクセスしたい場合はそのホールドにアクセスする際に通過するホールドに十分な空きスペースがないとアクセスすることが出来ない。

本稿では第 2 章で席割問題に対する詳細な問題設定や、本研究で扱う自動車輸送航海における専門用語について定義する。第 3 章では注文に含まれる自動車一つ一つをどの領域に割り当てるかを最適化する数理モデルと、そのモデルに登場する本研究独自の制約や好ましい席割作成のための目的関数を説明する。第 3 章で紹介する数理モデルは二種類あり、一つ目は

効果的な席割作成に対する考慮事項を全て目的関数で表したモデルである。二つ目は一つ目で提案したモデルについて、求解時間を抑えるために一つ目のモデルの目的関数の一部を制約化したものである。入力として与えられる自動車の数膨大になる場合があり、その場合これらのモデルでは入力情報が増加した場合に短い時間で求解できない可能性がある [2]。第 4 章では入力情報が膨大な場合でも十分短い時間で求解ができるように、入力で与えられた積載自動車の注文リストをグループ化したものに対して船内スペースへの割り当てを最適化するモデルを提案する。第 5 章では提案した数理モデルと商用ソルバーを用いて、実際の過去の航海データを基に求解実験を行う。またソルバーで得られた結果を席割作業を行なっている方々に評価してもらうことで、提案した数理モデルの有効性について考察する。

2 問題説明

様々な国で複数の港を経由し、自動車を輸送する運搬船を想定する。このとき乗用車 100 台を港 A から港 B、トラック 30 台を港 C から港 D というような各注文を、運搬船の階層毎に一定の広さで区切られたスペースへの割当てを計画する。この作業を席割作業 (stowage plan)[1] という。本稿では席割作業の自動化の初期段階として、単純な船の構造データを用いて実験する。また実際の席割業務では積載する自動車の種類が多様であるため、ショベルカーやブルドーザーなどの建機は自動車の高さや重さが乗用車とは大きく異なるので特定の領域にしか収容出来ないという問題がある。これに対して本稿では自動車を全て乗用車とし、船内の任意領域においてどの自動車も積載が可能と言う条件で席割を作成する。この章では一般的な船への自動車積載における専門用語の定義、席割作業における入力情報や出力情報の説明をする。

2.1 用語定義

本研究で扱う船の航海等に関する専門用語の定義をする。

- 席割
注文一つ一つを船のホールド割り当てる作業。
- シミュレーション
席割で決まった自動車を一台ずつホールド内の領域に貼り付ける作業。
- プランナー
席割やシミュレーションを考える作業。
- ギャング
港で自動車をホールドまで運転して積み降ろしをする作業。
- デッキ
船の内部の階層。
- ホールド
各デッキ内を一定間隔の領域で仕切られた空間。
- ランプ
上下デッキに移動するために各デッキの特定ホールドについているスロープ。
- 注文
乗用車 100 台を港 A から港 B へ輸送、トラック 30 台を港 C から港 D へ輸送というような積載自動車の情報とそれらの積み地と揚げ地に関する情報。
- 積み地
注文における自動車を積む港。
- 揚げ地
注文における自動車を降ろす港。
- RT (revenue ton)
基準となる車一台の面積に対する各注文の車の面積の割合。本研究ではこれを資源要求量として扱う [1]。
- ユニット数
各注文に含まれる自動車の台数。

2.2 入力情報

本研究で扱う二種類の情報について述べる。

- 注文情報
本稿では注文内に含まれる情報のうち注文番号、積み地と揚げ地、ユニット数、積載自動車の RT、

自動車の重量情報を扱う.

- 船体情報

輸送に使う船のデッキやホールド番号, 各ホールドへアクセスするために通るホールドや各ホールドの許容収容量などの船の内部構造に関する情報.

2.3 出力

入力情報を元に様々な条件を考慮して注文番号 1 番を 3 デッキ 1 ホールドに 30 台中 30 台割り当てる, 注文番号 2 番を 2 デッキ 2 ホールドに 200 台中 40 台割り当てる, というようなエクセルファイル形式の席割を出力する.

3 定式化

この章では各注文に含まれる自動車を, どのホールドに何台割り当てるかを最適化する複数の数理モデルを提案する. モデルは二種類ありプランナー側の要望を全て目的関数で表したモデル SP1-1 と,

4 記号の定義

本研究における変数と定数記号の定義をする. 本研究の変数の定義を表??に, 定数の定義を表??に記載する.

4.1 制約

モデル SP1-1 で考慮する本研究の独自の制約について記す.

(a) 自動車移動経路に関する制約

ある港において注文が運搬船内の特定のホールドで積み降ろしがあるとき, 自動車がホールドへ到達するためには船の入口とホールドの間で通過する任意のホールドに対して自動車が通るための道を作る必要がある. 従って, 本研究では各ホールドに対して自動車の移動通路が十分確保出来る許容充填率を定義し, 任意の港で各ホールドに注文を積むときや各ホールドから注文を降ろすときには, 入口とそのホールドの間で通過する全てのホールドで, その港における充填率が許容充填率以下であることが必要である. 本研究ではこれを制約にすることで, 現実的に自動車の積み降ろしが不可能になる席割を禁止する.

(b) 船内重量バランスに関する制約

運搬船で自動車を運ぶ際に船の前方や後方へ自動車を詰め込みすぎたり, 船の上方へ自動車を詰め込みすぎると船が傾いたときにバランスを崩しそのまま横転してしまう可能性がある [5]. 本研究ではこの問題を考慮するため, 過去 10 回分の航海データを解析することによりその航海内での運搬船の重量バランスを数値化した. その結果得られた 10 回分の航海データの中で船内重量バランスが最も偏ったときを閾値とし, 運搬船の上下方向と前後方向それぞれに対して, 各港において運搬船から降ろす注文を全て降ろしたとき, または港で積む注文を全て積んだときについて船内重量バランスが閾値を超えてしまうような自動車の割当を禁止する.

4.2 目的関数

席割に関するプランナーやギャングとの意見交換や席割作業を実際に体験することで, プランナーは席割を考える上で注文の積み間違いや降ろし間違いが起こらないような席割や, ギャングにとって効率が良いような席割を目指していることが分かった. 本研究では, このような席割を実現するための 6 つの目的関数のパラメータを提案する.

(a) 一つのホールド内の降ろし地を揃える

ある港においてホールドから注文を降ろす場合, 同じホールド内に揚げ地の異なる注文が複数存在していると注文の降ろし間違いが発生する可能性がある. 本研究では揚げ地において各ホールド内で揚げ地を通過する注文があるときに, その揚げ地で降ろす注文があるとペナルティが発生する. またこのとき更に降ろす注文の中で積み地が異なる注文が複数含まれていたとする. この場合席割作業の次にシミュレーション作業を行う際, プランナーは降ろし間違いを防ぐため揚げ地が同じ注文は出来るだけホールド内の一つの領域に行き渡るようにホールド内のスペース配分などを考慮しなければならない. このようなプランナーの負担を考え, この場合は降ろす注文の中に含ま

れる相異なる積み地の分だけ更にペナルティが加算される。

(b) 船内で注文の積み降ろし地を揃える

ギャング達とのヒアリングで、積み間違いや降ろし間違いを防ぐためになるべく積み地や揚げ地が同じ注文は船内の同じ場所で固めておきたいという要望があった。本研究ではこの要望に応えるために隣接ホールドペアを用意した。隣接ホールドペアとは隣同士に位置するホールド同士のペアと、ランプで繋がったホールド同士のペアのことを指す。各隣接ホールドペアに対して積み地や揚げ地が異なっている注文が存在していると、その数分ペナルティが発生する。ただし、プランナーの要望で積み地がバラけてしまっても、揚げ地はなるべく固めたいとの意見をいただいたため、目的関数 (b) のペナルティ重みは揚げ地のペナルティ重みを積み地よりも重く設定している。

(c) 作業効率充填率を考慮する

ある港で注文の積み降ろしのために自動車をホールドから出すときやホールドに自動車を入れるときに、そのホールドと入口の間で通過するホールドにはその港を通過する注文が一定数存在する。この注文がホールドの領域に対して一定量以上の場合、プランナーはそのホールドを自動車が通れるための道を作る必要がありホールド内における自動車のスペース配分を考える必要がある。またギャングは作られた道に自動車を通さなければいけないため、自動車の移動に慎重になり全体の作業効率が落ちる可能性が高い。本研究ではこのように作業効率が落ちないための各ホールド毎の作業効率充填率を設定し、これを上回ったホールドを自動車が通過する場合にペナルティが発生する。

(d) デッドスペースを作らない

注文の合計資源要求量が船の合計収容量より小さい場合、特定のホールドに空きスペースが出来

る。このスペースが入口付近にあれば追い積みという港での突然の追加注文を積むことに対応出来るが、空きスペースが出来たホールドに到達するまでに通るホールドが許容充填率を上回っている場合は、このスペースに追い積みをする事は出来ない。このような空きスペースのことをデッドスペースと言う。目的関数 (d) ではランプで繋がった船のホールドが許容充填率を超え特定デッキ間の移動ができない状態になったとき、そのホールドよりも奥にあるホールドの中で、指定された最大許容デッドスペースよりも大きいデッドスペースがあるホールドが存在するときに大きいペナルティが発生する。

(e) 残容量をなるべく入口付近に寄せる

船の入口に近いホールドにまだ自動車を入れることのできるスペースがあると船への追い積みに対応できる。また、研究で扱う船は追い積みに対応しやすい5デッキに残容量をなるべく残したいとのプランナーから要望があった。目的関数 (e) では、積載自動車を全て積み終わる最後の積み地をチェックポイントとし、チェックポイントで5デッキや入口に近いホールドに対して残容量が多いほど全体目的関数のペナルティを減らす利得が発生する。

(f) 注文の分割ルールを守る

注文の資源要求量は、あるホールドに割り当てる注文の車のユニット数とその自動車一台のRTの積で表現する。従って注文に含まれる自動車のユニット数が膨大な場合、どのホールドにもその注文を割り当てることが出来ないという問題が発生する。この場合は注文の容量を適当な大きさに分割して複数ホールドへ収容する必要がある。実際の席割プランニング上ではこの場合、まず注文に含まれる合計ユニット数が100台以上の大きい注文と小さい注文に分類する。小さい注文については、分割して異なるホールドに入れるとギャングが自動車を積むときに混乱を招きやすいため基本的には分割をすることはないが、隣りあ

たホールド同士にちょうど良いスペースがあるときだけは分割してそれぞれのホールドに入れる. 目的関数 (f) では小さい注文に関して, 隣り合ったホールド同士以外に分割して注文を入れる場合は大きいペナルティが発生する. 大きい注文については一つのホールドに全てを入れることは不可能なため分割が必要であるが, 100 台以下の小さいサイズに分割してしまうとホールドの挿入先が増えてギャングの負担が増えるため, 100 台以下で分割した場合はペナルティが発生する. 更に分割した注文を二つの異なるデッキ内のホールドに分割した場合, 同デッキ内の隣接しないホールドに分割した場合, 同デッキ内の隣接ホールドに分割した場合の順に大きいペナルティがかかる.

参考文献

- [1] Kris Braekers, An Caris, and Gerrit K Janssens. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, Vol. 67, pp. 166–186, 2014

表 1 変数の定義

変数	変数の説明
n_i	ホールド i の残容量が b'_i のとき $n_i = b'_i$
v_{ij}	注文 j をホールド i に n 台 ($0 \leq n \leq u_j$) 割り当てるとき $v_{ij} = n$
x_{ij}	注文 j をホールド i に割り当てるとき $x_{ij} = 1$ そうでないとき $x_{ij} = 0$
c_{ij}	ホールド i に大きい注文 j を 100 台以下に分割して積むとき $c_{ij} = 1$ そうでないとき $c_{ij} = 0$
$c_{i_1 i_2 j}^1$	ホールドペア (i_1, i_2) に 1. 大きい注文 j_1 を分割して積むとき $c_{i_1 i_2 j_1}^1 = p_{\text{large}}^{\text{c1}}$ 2. 小さい注文 j_2 を分割して積むとき $c_{i_1 i_2 j_2}^1 = p_{\text{small}}^{\text{c1}}$ 3. それ以外るとき $c_{i_1 i_2 j}^1 = 0$
$c_{i_1 i_2 j}^2$	ホールドペア (i_1, i_2) に 分割した大きい注文 j_1 を 1. 隣のホールド同士に積むとき $c_{i_1 i_2 j_1}^2 = n \cdot p_{\text{large}}^{\text{c2}}$ 2. 同デッキの隣ではないホールド同士に積むとき $c_{i_1 i_2 j_2}^2 = s \cdot p_{\text{large}}^{\text{c2}}$ 分割した小さい注文 j_2 を 3. 隣のホールド同士に積むとき $c_{i_1 i_2 j_2}^2 = n \cdot p_{\text{small}}^{\text{c2}}$ 4. 同デッキの隣ではないホールド同士に積むとき $c_{i_1 i_2 j_2}^2 = s \cdot p_{\text{small}}^{\text{c2}}$ 5. それ以外るとき $c_{i_1 i_2 j}^2 = 0$
m_{it}	ホールド i が港 t で作業効率充填率を超えているとき $m_{it} = 1$ そうでないとき $m_{it} = 0$
m_{ijt}	港 t で注文 j をホールド i から積み降ろしを行う際, n 個の通過ホールドで作業効率充填率を超えそこを v 台車が通るとき $m_{ijt} = vn$