

時間制約付き混載輸送ネットワーク最適化問題 Less-Than Truckload Network Design Problem with Leadtime

岡野 裕之 柳沢 弘揮 依田 邦和
Hiroyuki Okano Hiroki Yanagisawa Kunikazu Yoda

日本アイ・ビー・エム株式会社 東京基礎研究所
Tokyo Research Laboratory, IBM Japan

1623-14 Shimotsuruma, Yamato, Kanagawa 242-8502, Japan
{okanoh, yanagis, yoda}@jp.ibm.com

概要

混載型の輸送ネットワーク設計問題を取り上げる。対象問題は、全国に配置された物流拠点で荷物を集約し、目的拠点の異なる荷物を混載して輸送する、宅配便や郵便に代表される物流サービスにおける計画問題である。出発拠点と目的拠点のペアごとに最大到達時間(リードタイム)が規定されており、その制約下で最も運用コストが小さくなるように、荷物の輸送経路とキャリアの運行スケジュールを決定する。この問題に対し、荷物の経路問題とキャリアの割当問題に分割するアプローチを提案し、その有効性を議論する。

Keywords: LTLT, 多品種フローネットワーク設計問題, 列生成, POP, 局所探索

1 はじめに

本論文は広域的な物流ネットワークを効率化する最適化問題について議論する。対象は、宅配便や郵便に代表される、全国の拠点間で荷物を輸送する物流サービス事業者が抱える問題である。筆者らはこの問題に対し、日本全国規模の問題に適用できる実用的なアルゴリズムを開発した。本論文の目的は、対象問題の最適化問題としての性質を明らかにすること、実用的なヒューリスティックアルゴリズムを提案することの2点である。最適化問題としての性質を議論するために、列生成による標準的な解法がどの程度の実用性を持つかを検証する。また、最適解からの距離と目的関数値の関係から、対象問題において局所探索に都合のよい Proximate Optimality Principle (POP) と呼ばれる性質が成り立つことを示す。これらの議論に基づき、対象問題を荷物の経路問題とキャリアの割当問題の二つに分割し、時間展開したグラフ上でこれらを交互に解く実用的な解法を提案する。最後に提案手法の評価のために、ある事例に基づいた、日本全国規模の人工的なテストインスタンスによる数値実験の結果を示す。

1.1 対象問題の背景

陸上の貨物輸送は、一つの拠点と近隣の複数地点を結ぶ地域的な配送と、複数の拠点間で荷物を輸送する広域的なものに分類できる。後者は、出発拠点から目的拠点まで荷物を載せ換えずに輸送する直送型 (full-truckload) と、中継地で荷物を載せ換えながら、途中区間では異なる目的拠点宛ての荷物を混載して輸送する、混載型 (less-than-truckload) の2つに分類され

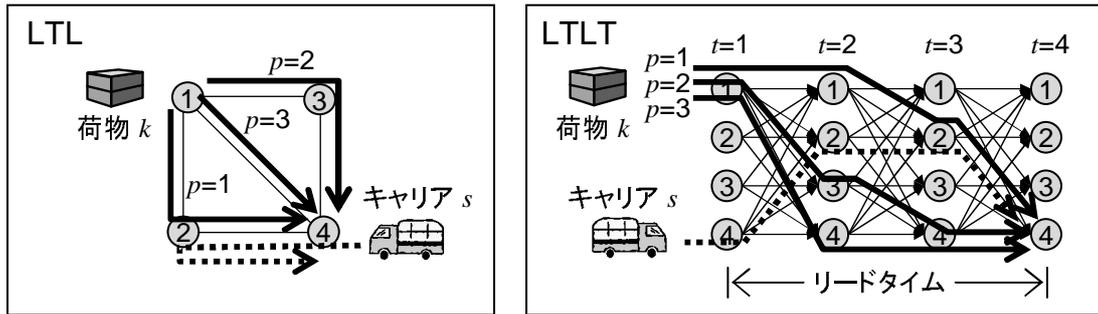


図 1: 2 種類の混載輸送ネットワーク最適化問題

LTL がリードタイムを考慮しない問題．LTLT がリードタイムを考慮する問題である．この例では荷物 k のリードタイムは 3 であり，待ち時間の位置の違いで 9 通りのパスが存在するが，3 本のパスだけを図示している．右のようなグラフを時間展開グラフと呼ぶ．別時刻における同一拠点を結ぶアークは待ちを表している．

る．これら広域的な輸送は，陸運だけでなく，航空や内航海運も組み合わせた形で行われる．本論文で扱う問題は混載型の輸送問題である．

混載型を採用する物流事業者は，拠点の運営，キャリアの運行計画，荷物の輸送計画などを行う．実際のキャリアの運行については自社資産で行う場合もあるが，別の物流事業者を利用することが多い．(トラック・鉄道貨物・航空機・船舶を総称してキャリアと呼ぶ．) いずれの場合も，運行経路ごとの負荷予測に基づいて，ある期間のサービスを外部調達したり，自社資産を配置する必要があり，この精度が運用コストを左右する．運行経路ごとの負荷予測を行うには，拠点間の荷量予測に加えて，荷物の輸送経路(以下，荷物経路)とキャリアの運行スケジュールを決める必要がある．荷量予測が与えられた時に，荷物経路とトラックの運行スケジュールを求める問題が，混載輸送ネットワーク最適化問題 (*Less-than truckload network design problem, LTL*) である．

物流事業者は顧客に対し，「地域 A をある日の午前 10:00 に出発した荷物は，地域 B には翌日の午前中に届く」といった最大到達時間(リードタイム)に関する目標を提示している．出発拠点と目的拠点のペアごとのリードタイムはサービスレベルと呼ばれ，物流事業者間の競争における重要な指標である．したがって，リードタイムの設定値を満たす範囲で最適な荷物経路およびキャリアの運行スケジュールを決定することが，物流事業者にとっての重要な課題となっている．混載輸送ネットワーク最適化問題において，特にリードタイム制約を加味したものが本論文で対象とする問題であり，本論文ではこれを「時間制約付き混載輸送ネットワーク最適化問題」 (*Less-than truckload network design problem with leadtime, LTLT*) と呼ぶ．

1.2 LTLT の困難性

前節で述べたような背景から，LTLT ではキャリアの運行スケジュールの決定が大きな要素となる．(鉄道，航空機，船舶の運行スケジュールは所与のため，主な考慮対象はトラック．) 例えば，ある区間の 1 日の荷量がトラック 1 台分としても，その区間を輸送する荷物のリードタイム制約を満たすために，2 便以上のトラックを運行しなければならない場合がある．したがって，区間ごとの荷量だけを扱い時間軸を考慮しない定式化では，任意の誤差が生じてしま

い実用にならない．拠点をノード，輸送区間をアーク（有向辺）とする入力グラフに対し，拠点を時刻ごとに別ノードとして展開（時間展開）し，そのグラフを入力として読み替えれば，時間軸を考慮しない定式化でも間接的に時間軸を考慮することができる（図 1）．しかしこの方法では入力データが大きくなるため，明示的に時間軸を考慮した解法でない限り実用には耐えないと思われる．

先行研究の多くは，時間軸を考慮せず区間ごとの荷量からトラック便数を決めている [10, 3, 7, 8, 11]．区間ごとのキャリアの運行スケジュールまで考慮した研究として文献 [1] がある．文献 [1] は，荷物経路を決定変数とし，キャリアの運行スケジュールを決定する子問題を繰り返し解くヒューリスティック解法を提案している．この解法が機能するのは，キャリアの運行区間が一つの場合だけを仮定しており，ピンパッキングに対する First Fit Decreasing のようなアルゴリズムが子問題に対しよい近似解を与えるからである．これに対し本論文が対象とする LTLT は，キャリアが複数区間の経路を運行することを前提とするため，キャリアの運行スケジュールを陽に主問題の中で扱う必要がある．

1.3 本論文の構成

2章において列生成解法の実用性や POP の議論を示す．3章では LTLT に対する実用的な局所探索アルゴリズムを提案し，実問題への応用を示す．最後に 4章でまとめを示す．

2章では，荷物の大きさや種類が均一で分割可能であり，拠点の処理容量（拠点容量と呼ぶ）が無制限という単純化したモデルを対象とする．この単純化したモデルは混合整数計画として定式化できるため，小規模のインスタンスについて最適解を求めることが可能である．また限定された単純化のため，実問題への応用に対して意味のある知見が得られる．3章では，荷物は分割不可能である（整数単位でしか輸送できない）とし，輸送単位へのまとめ方が異なる 2 種類の荷物があるとする．また，拠点容量制約を取り入れる．物流拠点において荷物を方面別に仕分ける能力には，自動仕分け機や人手の限界により上限が存在する．これを無視して計画してしまうと計画どおりに運用できない（拠点で余計な待ち時間が発生する）．そのため，3章では，拠点容量を時間展開グラフに埋め込んで表現する方法を提案する．

2 LTLT の最適化問題としての性質

荷物の候補経路と調達可能なキャリアの運行スケジュールを数え上げることにより，LTL および LTLT は次のような混合整数計画問題として表現できる：

$$(LTLT \text{ 主問題}) \quad \min_{x,y} \sum_{s \in S} c_s y_s \quad (1a)$$

$$\text{s.t.} \quad \sum_{p \in P^k} x_p^k = 1 \quad \forall k \in K \quad (1b)$$

$$\sum_{k \in K} \sum_{p \in P_a^k} v^k x_p^k - \sum_{s \in S_a} u_s y_s \leq 0 \quad \forall a \in A \quad (1c)$$

$$0 \leq x_p^k \leq 1 \quad (1d)$$

$$y_s \in \mathbb{Z}^* \quad (1e)$$

(1e) を $y_s \geq 0$ と線形緩和した問題に対して，以下のような双対問題を定義できる：

$$\text{(LTLT 双対問題)} \quad \min_{\sigma, \theta} \sum_{k \in K} \sigma^k \quad (2a)$$

$$\text{s.t.} \quad \sigma^k + \sum_{a \in A_p^k} v^k \theta_a \leq 0 \quad \forall k \in K, \forall p \in P^k \quad (2b)$$

$$\sum_{a \in A_s} -u_s \theta_a \leq c_s \quad \forall s \in S \quad (2c)$$

$$\theta_a \leq 0 \quad (2d)$$

ここで，各変数および定数の意味は以下である：

- x_p^k : 荷物 k の p 番目の候補経路で輸送する荷物 k の割合
- y_s : キャリア s の台数
- c_s : キャリア s のコスト
- u_s : キャリア s の積載容量
- v^k : 荷物 k の輸送すべき総量
- K : 荷物の集合
- A : アークの集合．ただし別時刻における同一拠点を結ぶアークは除く．
- S : キャリアの候補集合
- P^k : 荷物 k の経路候補集合．個々の経路を荷物経路と呼ぶ．
- P_a^k : 荷物 k の荷物経路の内，アーク a を含むものの集合
- S_a : レグ a を運行するキャリアの集合．レグはキャリアから見たアークの意味．
- A_s : キャリア s が運行するレグ (アーク) の集合．キャリア経路と呼ぶ．
- A_p^k : 荷物 k の p 番目の候補経路が含むアークの集合
- σ^k : (1b)に関する双対変数
- θ_a : (1c)に関する双対変数

上記の定式化では，荷物 k は，同一の出発拠点・目的拠点・出発時刻・リードタイムを共有する荷物全体を表す．荷物が分割可能であることに注意する．また，上記定式化では時間軸が明示的には表現されていないが，荷物経路およびキャリア経路は時間展開グラフ上のパスであり，間接的に時間軸が表現されている．以下，キャリアの運行スケジュールと (時間展開グラフ上の) キャリア経路を同義として用いる．

別の定式化として，荷物経路の候補を持たず，各アークに 0-1 変数を持たせて流量保存則を導入する定式化も一般的である．ただし実問題へ応用する場合，荷物経路に副次的な制約が考慮できることが望ましいことから，上記のような定式化を採用した．同様にキャリア経路についても，既設路線の考慮や副次的な制約の考慮のために，上記のように複数レグを明示的に入力する定式化とした．キャリアが複数レグを持てる点 (つまり $|A_s| \geq 1$) は，LTLT の特徴である．アークごとの輸送量に独立にコストがかかる場合は， $|A_s| = 1$ とすればよく，その場合の LTLT は多品種フローネットワークデザイン問題と等価になる．

2.1 列生成解法の実用性

ある程度以上の拠点数になると，荷物経路集合 P^k およびキャリア集合 S について全列挙は不可能である．これに対処する方法としてまず考えられるのが列生成である．列生成では，

表 1: 列生成解法による問題規模および整数ギャップ

期間は時間展開グラフの高さ，行数は $|K| + |A|$ ，荷物経路は $\sum_k |P^k|$ ，キャリアは $|S|$ を表す．COIN-OR[9] を用いて計算した．—は制限時間内に列生成が終了しなかったことを示す．

拠点数	期間	行数	荷物経路	キャリア	整数ギャップ (%)
7	6	228	290	331	12.98
19	12	1692	3249	2095	2.51
37	18	5904	14447	7641	11.28
61	24	14808	43013	17475	7.45
91	30	30780	106374	35160	—

小さいサイズの集合について主問題の線形緩和を解き，双対問題において (2b),(2c) に違反を起こさせるような，まだモデルに入っていない制約を見つけ，それに対応して主問題に新しい列 (荷物経路およびキャリア) を挿入する．荷物経路を生成する場合 (2b) を利用し，時間展開グラフの各アーク a のコストに $-\theta_a$ を設定して最短路計算を行い， $-\sum_{a \in A_p^k} v^k \theta_a - \sigma^k < 0$ を満たすパスを探索する．キャリアを生成する場合 (2c) を利用し， $c_s + \sum_{a \in A_s} u_s \theta_a < 0$ を満たす経路 A_s とコスト c_s を持つキャリアを生成する．これらの条件を満たす列が見つからなくなった時点で列生成を終え，混合整数計画法を適用して許容解を得る．

テストインスタンスを用いて列生成の実用性を検証した．拠点数は，六角形の頂点とその六角形の中心点の計 7 点を拠点として配置したインスタンス，さらにその外周の三角格子に 12 拠点配置した計 19 拠点をインスタンス，同じ要領で 37 点，61 点，91 点のインスタンスを生成し，アークは三角格子の各辺とした．グラフの直径方向の 1 本のパス沿いには大きな容量のキャリアを配置し，その他のアークには小さな容量のキャリアを配置した．スケジュールの期間はグラフの直径を 1 日分とみなして 3 日間とした．荷物は 1 日目と 2 日目に，全点間にランダムな量だけ発生させた．荷物のリードタイムは最短路の 1.2 倍程度，キャリアのレグ数は最大 2，レグ数 2 のキャリアのコストはレグ数 1 のコストの 1.5 倍とした．以上の設定で実験した結果が表 1 である．これは容易に想像できた結果だが，小さい拠点数でも問題規模が爆発して混合整数計画が解けなくなる (実用的な時間で許容解が見つからない) ため，列生成解法は LTLT に対して実用的ではないと言える．

ただし，以上のような単純化したモデルを対象とする場合，別の定式化や別解法によってよい結果が得られる可能性はある．この章の議論は，あくまでも実用的な解法を設計するための知見を得ることが目的である．

2.2 POP (Proximate Optimality Principle) に関する議論

POP とは，いくつかの最適化問題について経験的に成り立つ原理で，Glover[6] によって提唱された．それは「良い解の近傍には別の良い解が存在する」という性質であり，良い解の近傍を集中的に探索することが有益であることを示唆する．直感的には

- 良い解どうしは互いに多くの共通の構造を持っている

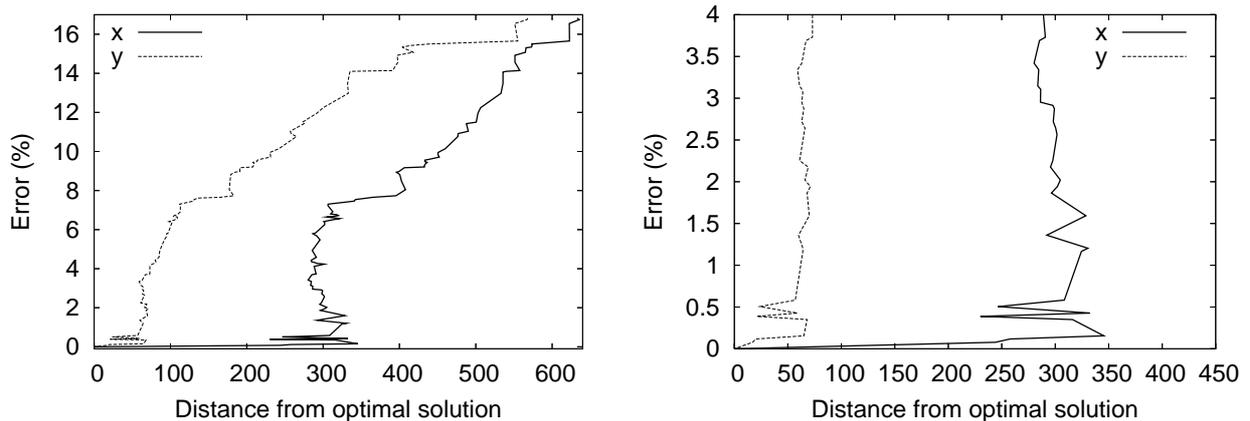


図 2: 混合整数計画法の途中段階で得られる解のプロット

拠点数 19 のあるインスタンスに対する混合整数計画法の実行途中で見つかったそれまでの最良解を、変数 x (荷物経路選択) と変数 y (キャリア選択) のそれぞれについてプロットしたもの (左側) と、その誤差 4% までの範囲を拡大したもの (右側) . 混合整数計画法には COIN-OR[9] の分枝限定法 (CBC) を用いた .

とすることができる . これまで POP が成り立つと報告された最適化問題には以下が含まれる: Traveling Salesman Problem (TSP) [5], MAXSAT [12], Quadratic Assignment Problem (QAP) [4], Job Shop Scheduling (JSS) [2] . 例えば文献 [12] では, MAXSAT に対してランダムに生成した解を使って, 最適解からの距離と最適値からの誤差が強い相関を持つことを実証している . LTLT についても POP が成り立てば, この問題に対して局所探索が有効であると判断できる .

LTLT は目的関数が $\sum cy$ のため, 変数 y については POP が成り立つことが予想できる . つまり, y の微小な変化は目的関数を微小にしか変化させないため, よい解の周辺には同程度のよい解が見つかる可能性が高い . しかし, x については目的関数に組み込まれていないため, y の変化に対して x を大きく変化させないと同程度の解が得られない可能性がある . そこで, 混合整数計画法の途中で得られるそれまでの最適解 (\hat{x}, \hat{y}) について, 最終的に得られる最適解 (x^*, y^*) との, x, y それぞれの距離 $\sum |\hat{x} - x^*|, \sum |\hat{y} - y^*|$ と誤差 $(\sum c\hat{y} / \sum cy^* - 1) \times 100$ をプロットした . 図 2 は, 拠点数 19 のあるインスタンスについて混合整数計画法の探索の軌跡をプロットしたものである . 図 2(左側) から, 全体として, x と y の双方について距離と誤差が相関を持つことが分かる .

この実験は, あるアルゴリズムが「よい解の近傍のさらによりよい解を探索している」という事実を示しているだけで, 最適解からの距離と誤差との関係については何ら示していない . 例えば, ある距離に位置する解が, ある程度の誤差の範囲に分布していることが図から読み取れる (図 2 右側) . アルゴリズムが探索しなかった解も合わせてプロットすると, 解がさらに広い範囲に分布し, 図のような明確な相関関係が現れない可能性はある . しかし少なくとも, y の微小な変化に伴い x が大きく値を変化させるようなことはなく, x, y の双方について POP が成り立つと仮定してアルゴリズムを設計すること, つまり x, y に関する局所探索に基づくアルゴリズムを用いることで, よい解が得られる可能性が高いと判断できる .

2.3 解法の方針

2.1節の議論から、荷物経路とキャリア経路を数え上げておくことは無理であることが分かった。また、2.2節の議論から、荷物経路とキャリア経路のそれぞれについて、局所探索に基づくアルゴリズムが有効であることが分かった。そこでキャリアについては、運行時間に余裕を持たせた(前後にずらせる)表現をすることで時間軸方向の候補数を削減し、 $S_1 = \{s \mid y_s = 1, s \in S\}$ と $S_0 = \{s \mid y_s = 0, s \in S\}$ の要素の入れ換え操作を行う局所探索を採用する。 $y_s \in \{0, 1\}$ とすることで局所探索の実装を単純化する。荷物経路については、荷物を分割不可能な単位に展開して K を再定義し、各 $k \in K$ について常に一つの荷物経路を保持して局所探索を適用する。ただし、LTLTの目的関数は荷物ではなくキャリアにかかっているため、荷物経路の局所探索では、積載物がなくなったキャリア・レグの数を目的関数とみなす。また、荷物経路の局所探索の中で荷物を別のキャリアに載せかえる操作も行う。詳細については次章で述べる。

3 実問題への応用

実問題においては拠点容量の考慮や、仕分け方法の異なる荷物の種類を扱うことが必要となる。そこで時間展開グラフを用いてこれらの制約を表現する方法を提案する。さらに、前章で示した解法の方針に基づき、候補キャリアの中から使用するキャリアを選択し、それらの運行スケジュール(時間展開グラフ上のキャリア経路)を決定する局所探索アルゴリズムと、時間展開グラフ上の荷物経路を決定する局所探索アルゴリズムを提案する。

3.1 時間展開グラフによるモデル化

時間制約をグラフで表現するために図3(右側)のような時間展開したネットワークで考える。例えば、「時刻 $t = 0$ における拠点1」のノードから「時刻 $t = 1$ における拠点2」へのアークは、時刻 $t = 0$ に拠点1を出発すれば時間1で拠点2に到達できることを表す。このように各拠点を時刻付きのノードで表現することにより、時間制約を満たすかどうかの判定が容易になる。例えば、ある時刻に出発した荷物が目的拠点の締切時刻までに到着できるかどうかは、その荷物が通った経路・時刻に沿って時間展開グラフ上にパスが存在するかどうかで判別可能となる。

物流拠点において荷物を方面別に仕分ける能力には限界が存在するため、多くの荷物が一度に拠点に集中しないように工夫する必要がある。そのため、図3(左側)のように、時間展開グラフ上の各ノードを、各時刻 t について到着用ノード・中間ノード・出発用ノードの三つに分解する。他の拠点から入ってくるアークは全て到着用ノードに接続し、他の拠点へ出て行くアークは全て出発用ノードに接続する。そして、ノード間には図3のようにアークを張る。ここで、中間ノードと出発用ノード間のアークを通る荷物の総量を制限することで、単位時間あたりに処理できる荷物の数を制限することができる。他のアーク(中間ノードと出発用ノードを直接結ぶアーク以外)は無制限に荷物を流せるものとする。

図3において、到着用ノードから出るアークには、荷物の種別に関する制約を設けている。これは、以下で述べる理由によるものである。通常、一つひとつの荷物は、仕分け機を利用して輸送単位ごとに容器に入れ、まとめて輸送する。(各キャリアは複数の輸送単位を積載する。)

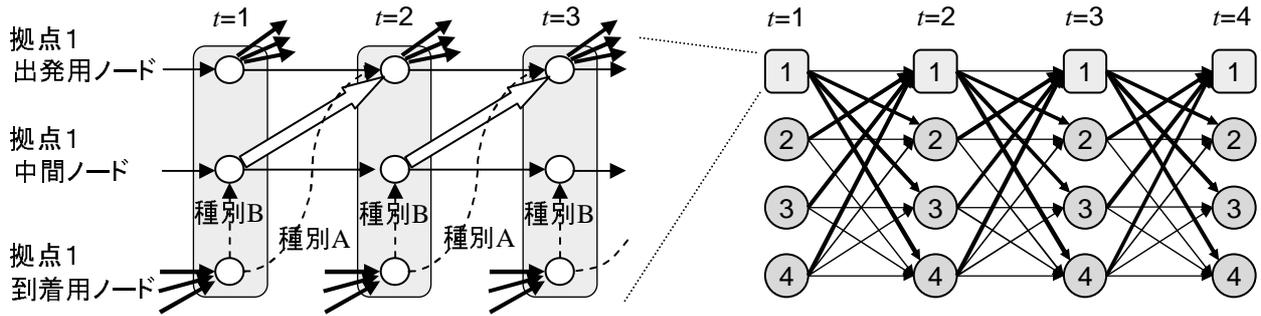


図 3: 時間展開グラフにおける拠点容量の表現

右側が時間展開グラフ．この時間展開グラフでは簡単のために全ての移動時間が 1 になっているが，実際には移動距離に応じて異なる移動時間を設定する．左側が拠点容量制約のグラフにおける表現．種別 A は仕分け不要の荷物，種別 B は仕分けが必要な荷物．白抜きのアークで容量を制約する．

荷物には大きく分けて 2 種類あり，一つは比較的荷量が多く，出発拠点で一つの輸送単位にまとめて目的拠点までそのまま輸送するもの（種別 A）．もう一つは比較的荷量が少ないため，異なる宛先への荷物も一つの輸送単位にまとめて中継拠点に輸送し中継拠点で毎回仕分けするもの（種別 B）である．この両者の扱いの違いを吸収するため，到着用ノードに到達した荷物は，種別 A のものは全て出発用ノード行きのアークに載せ，種別 B のものは全て中間ノード行きのアークに載せるものとする．このようにすることで，種別 A の荷物が中継拠点で再仕分け処理を必要としないことが表現でき，種別 B の荷物は必ず中継拠点で再仕分け処理されることが表現できる．また，種別 A の荷物も種別 B の荷物も共に，最初の出発拠点では中間ノードから出発することにより，種別 A が最初の拠点では仕分け機を通す必要があることが表現できる．

3.2 アルゴリズムの概要

提案するアルゴリズムは大きく分けて，(1) 荷物経路・時刻を固定した状態でキャリアを選択し運行スケジュールを決定するアルゴリズム（キャリア割当）と，(2) キャリアの運行スケジュールを固定した状態で荷物経路・時刻を決定するアルゴリズムの二つから構成される．この二つのアルゴリズムを交互に動作させることにより，最適解を探索する．初期解は，基本的には，単位容量あたりのコストが最小の荷物経路を用いて作成する．拠点の容量あふれが発生する場合は，それらの荷物を他の適当なパスで輸送して実行可能解を作成する．

3.3 キャリアの運行スケジュールを決定するアルゴリズム（キャリア割当問題）

この節では，時間展開グラフにおいて各荷物の出発拠点から目的拠点までの経路が与えられている場合に，最小のコストで全ての荷物を輸送するキャリアを選択し，選択したキャリアの運行スケジュールを決定する局所探索アルゴリズムについて述べる．

まず， A を時間展開グラフのアーク集合とし，各アーク $a \in A$ からはそこを通る荷物の量が取得できるものとする．キャリアの候補集合を S とする．各候補キャリア $s \in S$ には，モード（鉄道・航空機・トラックなどの種別），積載容量，経路，経路上の各拠点（経由拠点）からの出発時刻の時間枠（最早出発時刻，最遅出発時刻）が与えられているとする．キャリアの経路に

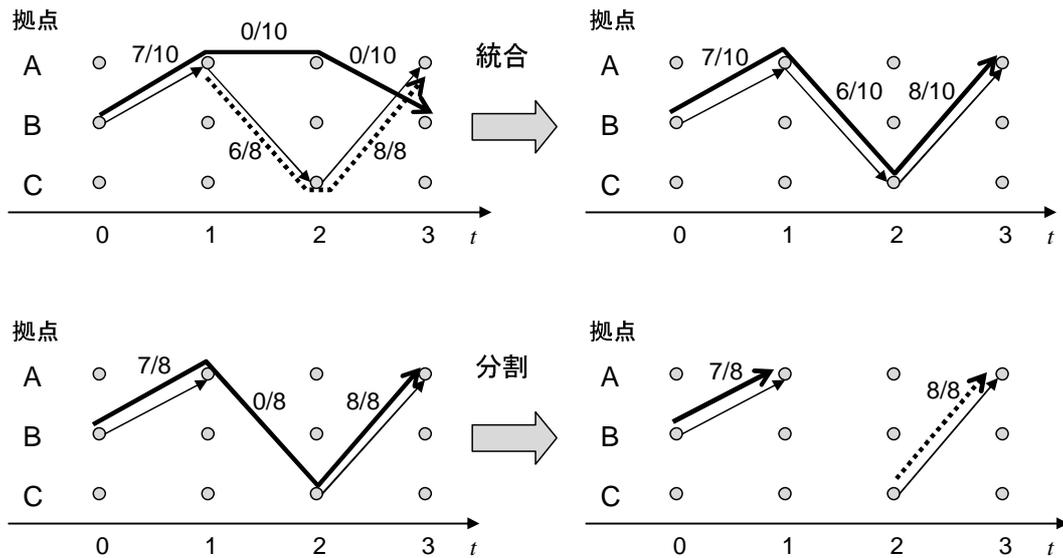


図 4: キャリアの交換による統合と分割

含まれるアークをレグと呼ぶ。なお、2 拠点間の移動時間は別に与えられるとする。(例えば、鉄道では時刻表、トラックでは発着拠点についての時間行列など。) 鉄道や航空機など時刻表に従って運行するキャリアでは、最早出発時刻と最遅出発時刻は同値である。キャリアの運行スケジュールを決定するとは、次のことを考慮しながら、候補集合 S の中から部分集合 $S_1 = \{s \mid y_s = 1, s \in S\}$ をコスト $\sum_{s \in S_1} c_s$ が最小になるように選び出すことである:

- 各 $s \in S_1$ につき積載容量制約を満たしながら各経由拠点で積載する荷物の量を決定する
- 各 $s \in S_1$ につき時間枠の範囲で各経由拠点からの出発時刻を決定する
- 各アーク上の全ての荷物はそれぞれどれかのキャリアに積載される

キャリアのコスト c_s は、モード、積載量、経由拠点、発着時刻などに依存したルールによって計算される。

3.3.1 局所探索におけるキャリアの交換

局所探索における 1 回の移動を、実行可能解 S_1 に含まれるキャリアと $S_0 = S \setminus S_1$ に含まれる候補キャリアの交換とする。実装ではパラメータ a, b を用いて、 S_1 の中から a 台のキャリアを取り出し、 S_0 の中から最大 b 台のキャリアを S_1 へ入れて、あふれた荷物を積み換える操作を行う。 S_1 から取り出すキャリア数 a は固定であるが、 S_1 へ新たに入れるキャリア数 b は最大値である。(荷物が全て載せ換えられた時点でそれ以上のキャリアを必要としないため。) 例えば $(a, b) = (1, 0)$ ならば不要なキャリアを抜き取る操作であり、 $(a, b) = (1, 1)$ ならば不要なキャリアを抜き取るか、または 1 台のキャリアを別の 1 台と交換する操作にあたる。現実のサイズの問題では a が大きくなると実行時間が大きくなるため $a \leq 3$ が限界である。

これらの交換操作によって図 4 のように統合や分割が行われる。図 4 は荷物の存在するアークとキャリアのレグを重ねて表示してあり、レグ上の i/j という数字は、そこでは荷量 i の荷

物が容量 j のキャリアによって運ばれることを意味する． $b'(\leq b)$ を S_1 に新たに入る候補キャリア数とすると， $a > b'$ ならば統合， $a < b'$ ならば分割に相当する． $a = b'$ の場合は S_1 のキャリア数は変わらないが，使用するキャリアは変化して合計のコストは下がる．

統合では，積載率の低い多数のキャリアで運ばれていた荷物を少数のキャリアに集めて積載率を上げ，合計のコストを下げる．図4上部の例では， $B(t=0) \rightarrow A(t=1)$ に荷量7， $A(t=1) \rightarrow C(t=2)$ に荷量6， $C(t=2) \rightarrow A(t=3)$ に荷量8が与えられている．これらの荷量を $B \rightarrow A \rightarrow B$ という容量10のキャリアと， $A \rightarrow C \rightarrow A$ という容量8のキャリアで運んでいる．統合によって， $B \rightarrow A \rightarrow C \rightarrow A$ という容量10のキャリア1台で運ぶことができ，合計のコストが下がる．分割では，積載率の低い少数のキャリアで運ばれていた荷物を多数のキャリアへ分けて積載率を上げ，合計のコストを下げる．図4下部に統合の場合と同様にして例を挙げた．

3.3.2 キャリア割当問題アルゴリズムの概要

荷物を載せ換える場合，時刻・拠点が全く重ならないキャリアどうしでは載せ換えを試す意味がないため，同じレグを受け持てるキャリアを検索するデータ構造を用意する．各アーク $a \in A$ に， a をレグとして受け持てる全てのキャリアを登録する．また選択中のキャリア $s \in S_1$ には，経路上の各レグに対応するアークへのリンクを持たせる．これにより，ある一つのキャリア $s \in S_1$ があったときに， s と一緒に解集合から抜き取ることが意味のあるキャリア，また S_0 から S_1 への移動を試すことが意味のあるキャリアを効率的に選び出すことができる．キャリア $s \in S_0$ を解集合 S_1 に加える際，各拠点での出発時刻と積載する荷物の量を決めなければならない．これには，動的計画法を用いて「 s に最も多く荷物を積載するような出発時刻」を求める問題を解いてそれぞれを決定した．

パラメータ (a, b) についての局所探索のアルゴリズムを以下に示す：まず，解集合 S_1 の全キャリアをヒープ H に入れる． H からはコストが高く積載率の低いキャリアが順に取り出される．各 $x \in H$ に対して，時刻・拠点に共通部分を持つ解の部分集合 $NS = \{s \mid s \in S_1, A_s \cap A_x \neq \phi\}$ を選び出す． NS から大きさ a の部分集合 X を重複なく順に取り出していく．各 $x \in X$ が積載している荷物をアークへ未積載荷物として戻し， S_1 から X を抜き取る． X と時間・拠点に共通部分を持つキャリア集合の部分集合 $NC = \{s \mid s \in S_0, A_s \cap A_x \neq \phi, x \in X\}$ を選び出し， NC からキャリアを順に取り出して，未積載荷物の一部を積載させて解 S_1 に入れる．全ての制約を満たしてコストが下がれば成功で， S_1 を更新して上記の処理を続ける（ヒープ H を初期化するところに戻る）．全ての組合せを試してもコストを下げるようなキャリアの交換がなければ，局所探索を終了する．

3.4 荷物の輸送スケジュールを決定するアルゴリズム (荷物経路問題)

荷物の輸送スケジュールを決定するアルゴリズムでは，初期解として何らかの実行可能解が与えられるものとする．つまり，全ての荷物が，いずれかのキャリアによって正しく輸送されているものとする．ここで述べるアルゴリズムは，与えられた解に対して，いくつかの荷物経路を変更することで，使用しないキャリア・レグを見つけ出すことである．つまり，目的関数はコストの最小化ではなく，使用しないキャリア・レグをなるべく多く見つけることである．アルゴリズムは次に述べる三つの局所探索ルーチンから構成されている．

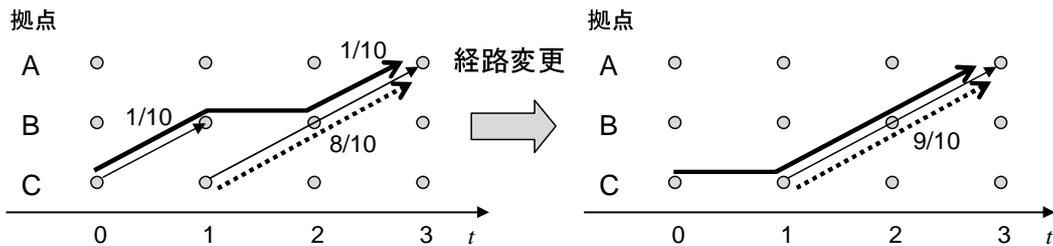


図 5: 荷物経路の変更によるキャリア数削減

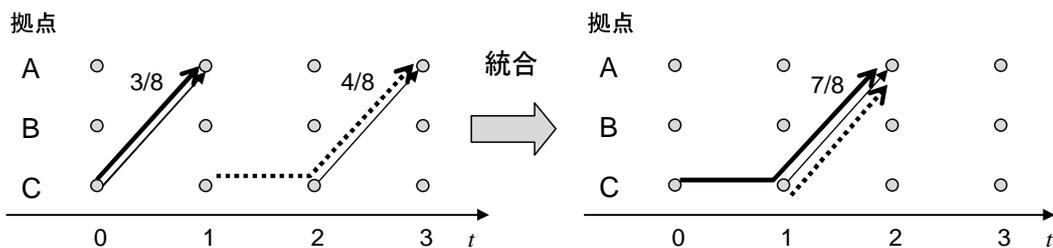


図 6: 荷物経路の変更に伴うキャリアの統合

輸送経路の変更 満載になっていないキャリアに載っている荷物は、単位荷物あたりの輸送コストが高くなっている。そのため、積載量が少ないキャリアに積載されている荷物を全て別のキャリアを用いて輸送できれば、キャリアを使用する必要がなくなるためコストが削減できる。具体的には、積載量が少ないキャリアについて、そのキャリアに積載されている全ての荷物について、荷物経路とキャリアの変更を試みる。全ての可能性を全探索して、他に空きのある経路があればそれに変更する。

例えば、図 5(左側) は、時刻 $t = 0$ に拠点 C を出発し拠点 B を経由して時刻 $t = 3$ に拠点 A に到達する荷物が一つと、時刻 $t = 1$ に拠点 C を出発して時刻 $t = 3$ に拠点 A に到達する荷物が 8 あることを示している。明らかに、時刻 $t = 0$ に拠点 C を出発する荷物は、時刻 $t = 1$ に拠点 C を出発する荷物と一緒に輸送したほうが効率的なので、右側のように輸送してコスト削減を図ることができる。

キャリアの統合 同じ区間を異なる時刻に出発するキャリアが複数ある場合で、両者に載っている荷物を混載して一つのキャリアで輸送できるとき、それらのキャリアを統合する。

例えば、図 6(左側) は、時刻 $t = 0$ に拠点 C を出発して時刻 $t = 1$ に拠点 A に到達する荷物と、時刻 $t = 2$ に拠点 C を出発して時刻 $t = 3$ に拠点 A に到達する荷物があることを示している。ここで、時刻 $t = 1$ に拠点 C を出発して時刻 $t = 2$ に拠点 A に到達するキャリアがあれば、両者をこのキャリアに統合する。これでキャリア数を半分にできたので、コスト削減効果が得られる。この局所探索は、全ての区間について、「出発時刻が隣り合うキャリアどうしを統合してキャリアの数を減らせるかどうかを判定し、減らせる場合は統合する」という操作からなる。

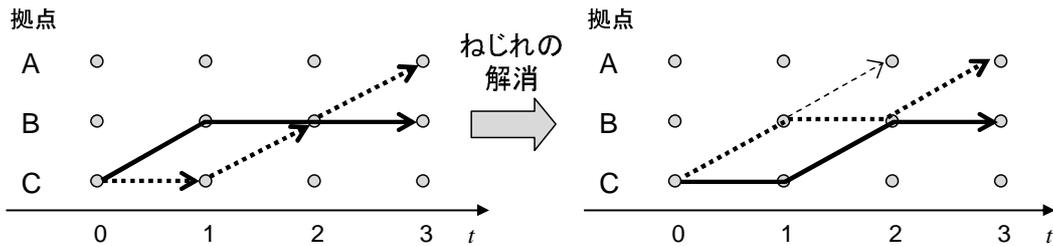


図 7: 荷物経路におけるねじれの解消

ねじれの解消 上の二つの局所探索を繰り返し適用すると、ある区間を通る荷物で、発着時刻にねじれ関係(すなわち、早く到着しなければいけない荷物が、遅く到着してもよい荷物より後に出発するケース)が発生することがある。このとき、両者の出発時刻を入れ換えれば、遅く出発していた荷物の次区間での出発時刻に余裕ができるため、上の二つの局所探索をより促進させることにつながる。

例えば、図 7(左側)は、時刻 $t = 0$ に拠点 C を出発する荷物と時刻 $t = 1$ に拠点 C を出発する荷物があり、ともに拠点 B に到着する。先に拠点 C を出発した荷物は時刻 $t = 3$ になっても拠点にとどまるが、一方で、後で拠点 C を出発した荷物は時刻 $t = 2$ に拠点 B を出発し拠点 A に向かっている。この二つの荷物は、ねじれ関係にあるため、右側のような関係になることが望ましい。つまり、先に拠点 C を出発した荷物は先に拠点 B を出発し、後で拠点 C を出発した荷物が後で拠点 B を出発するように変更する。このようにすれば、拠点 B を時刻 $t = 2$ に出発していた荷物は、時刻 $t = 1$ に拠点 B を出発するという選択肢が増えることになる。こうして、スケジュールの自由度が上がれば、他の二つの局所探索の適用を促進することにつながる。

3.5 実験データ

日本を約 10 の地域に分割し、各地域にいくつかの拠点を用意して、合計約 70 拠点用意した。各地域内には「主要拠点」が 1 ~ 3 拠点あり、その他は通常の拠点である。時間展開グラフで表現するにあたり、各拠点を 480 個 (15 分を 1 単位として 5 日分) の単位時間で展開した。キャリアは、鉄道・航空機・トラックの 3 種類を用意した。鉄道・航空機に関しては、現実に行われているダイヤ情報を使用した。鉄道はコンテナ (容量 = 7 輸送単位) ごとに積み込む。航空機は 1 輸送単位で積み込む。トラックに関しては、地域内便については、全ての 2 拠点間に最大 2 レグの範囲で設定できるようにした。地域間便については、主要拠点どうしを結ぶ全ての 2 拠点間について、直行便と単純往復便を設定した。地域内・地域間ともに、トラック A (容量 = 12 輸送単位)、トラック B (容量 = 18 輸送単位)、トラック C (容量 = 21 輸送単位) の 3 種類を各 28 台ずつ作成した。入力候補キャリア数は約 69 万台である。荷物は、全ての出発拠点・目的拠点ペア ($70 \times 70 = 4900$ ペア) について、最初の 3 日分の荷物を生成した。荷物の締切時刻は、現実のダイヤに合わせ最大約 2 日間程度の現実的な設定をした。荷物の量は、国内荷物の輸送量の分布データに基づき、全体で 6 万個の輸送単位に相当する荷物を用意した。

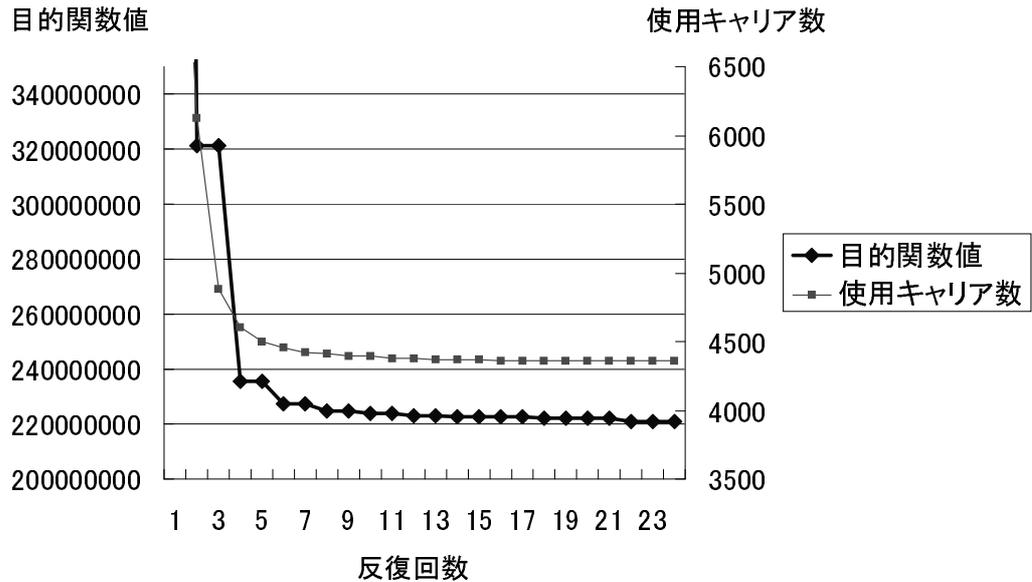


図 8: 実験結果

3.6 実験結果

上記のデータに対して最適化アルゴリズムを適用した結果，図 8 のような結果になった．各反復では，キャリア割当アルゴリズムと荷物経路アルゴリズムの結果が交互に示されている．キャリア割当アルゴリズム適用時には，キャリア使用に伴うコストが最適化されるため，目的関数値が変化する．荷物経路アルゴリズム適用時には，（荷物を一つも運ばなくなって不要になったキャリアも除去しない実装にしたため）目的関数値は変化しないが，荷物を一つも運んでいないキャリアの数は減少している．この結果から，二つのアルゴリズムを交互に実行することで徐々に解を改善し，局所最適解を見つけられることが分かる．この実験に要した計算時間は，Windows XP, Pentium 4 (3.2GHz), 3GB RAM のマシンで 53 分であった．この実行時間は，LTLT のような資源調達あるいは資源配置のための計画問題としては十分実用に耐える範囲である．

本論文では実際の事例の公表はできないが，ある事例において提案手法を用いて現状再現シミュレーションと最適化シミュレーションを行い，これらの比較から，提案手法により高い効果が得られることを確認している．

4 おわりに

陸運業において重要な問題である時間制約付き混載輸送ネットワーク最適化問題 (Less-than truckload network design problem with leadtime, LTLT) を取り上げ，混合整数計画による定式化を用いた実験により問題の性質を調べた．その結果，使用した混合整数計画による定式化ではモデルが大きくなり過ぎ，実用規模の問題には適用できないことが分かった．また，この問題の小規模なインスタンスに対する混合整数計画法の探索の軌跡をプロットする実験により，LTLT が局所探索に向いた性質を持つ，つまり Proximate Optimality Principle (POP) が成り立つと仮定できることを示した．

以上の考察を踏まえ、LTLT に対する局所探索ベースの実用的なアルゴリズムを提案した。提案した手法は、(1) キャリアの候補集合からの選択と運行スケジュールの決定を行う局所探索、(2) 荷物の経路決定とキャリアへの積み込みを行う局所探索の二つからなる。これらを、各時刻における拠点を別ノードとして展開した時間展開グラフを用いて、効率的に実装した。また、拠点における処理容量（仕分け能力の上限）の制約と仕分け条件の異なる複数種類の荷物を扱うために、これらを時間展開グラフ上で表現する方法を提案した。日本全国の規模のテストインスタンスを用いた実験により、提案手法が実用的な実行時間で局所最適解を見つけ出すことを示した。

今後の課題としては、初期解構築アルゴリズムの改良、キャリア割当アルゴリズムと荷物経路アルゴリズムの連携動作、メタヒューリスティックスの適用などが挙げられる。

参考文献

- [1] M. Amano, T. Yoshizumi, and H. Okano, *The modal-shift transportation planning problem and its fast steepest descent algorithm*, Proceedings of the 2003 Winter Simulation Conference, 2 (2003), 1720–1728.
- [2] S. Binato, W. Henry, D. Loewenstern, and M. Resende, *A GRASP for job shop scheduling*, in Essays and surveys on metaheuristics, P. Hansen and C. Ribeiro, eds., Kluwer Academic Publishers, 2001.
- [3] T. G. Crainic and J. Roy, *Design of regular intercity driver routes for the LTL motor carrier industry*, Transportation Science, 26 (1992), 280–295.
- [4] C. Fleurent and F. Glover, *Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory*, INFORMS Journal on Computing, 11 (1999), 198–204.
- [5] C. Fonlupt, D. Robilliard, P. Preux, and E. Talbi, *Fitness landscape and performance of meta-heuristics*, in Meta-Heuristics – Advances and Trends in Local Search Paradigms for Optimization, S. Vos, S. Martello, I. Osman, and C. Roucairol, eds., Springer, 1998, 255–266.
- [6] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Boston, USA, 1997.
- [7] B. Hoppe, E. Z. Klampfl, C. McZeal, and J. Rich, *Strategic Load-Planning for Less-Than-Truckload Trucking*, Technical Report CRPC-TR99812-S, Center for Research on Parallel Computation, Rice University, 1999.
- [8] N. Katayama and S. Yurimoto, *The Load Planning Problem for Less-than-Truckload Motor Carriers and a Solution Approach*, Proceedings of the 7th International Symposium on Logistics, (2002).
- [9] R. Lougee-Heimer, *The Common Optimization INterface for Operations Research*, IBM Journal of Research and Development, 47 (2003), 57–66. <http://www.coin-or.org>.

- [10] W. B. Powell and Y. Sheffi, *Design and implementation of an interactive optimization system for network design in the motor carrier industry*, Operations Research, 37 (1989), 12–29.
- [11] 片山直登, 共同輸送ネットワーク設計問題に対する *Lagrange* 緩和法, 流通情報学部紀要 2, 流通経済大学, 2002.
- [12] 柳浦睦憲, 茨木俊秀, 組合せ最適化問題に対するメタ戦略について, 電子情報通信学会論文誌, J83-D-I (2000), 3–25.