

域内トラック輸送の効率化を目的とした Ambient Calculus による物流シミュレーターに関する研究

A Study on the Logistics Simulator with Ambient Calculus for optimization of Intra-regional Truck Transportation

鳥山 颯斗[†]
Hayato Toriyama

加藤 暢[†]
Toru Kato

1. 研究背景

現在物流業において時間外労働の上限が月 45 時間・年 360 時間に制限される「働き方改革関連法」が 2024 年 4 月 1 日に適用される「2024 年問題」に直面している。現在のトラックドライバーの超過実労働時間は月 31~35 時間であり年間で換算すると 372 時間~420 時間となり大幅に基準値を超えている。その一方で現在の中ロット貨物の都市地域ブロックにおける輸送では、回転率を重視し、集荷配達後にトラックが空のまま営業所に戻っており、そのため積載率が 50%を上回ることがない。トラック資源を有効活用し、積載効率を高める有効な策として検討されているのが「フィジカル・インターネット構想」である。これは物流各社が共有する集配拠点（ハブ）、規格化された容器（パッケージ）、配送上の約束事（プロトコル）を取り決め、デジタルインターネットに倣い物流の飛躍的な効率化を目指す構想である。これを実現するために様々な AI・アルゴリズムが研究され、コスト最小化、最適積載効率、最短ルートなど、輸送手段の最適化が目指されている。しかしこのような最適化を真に実現するためには、解の存在確認と組み合わせ最適化、この 2 点で組み合わせ爆発規模の問題への対処能力が必要となるが未だ有効な手段は確立されていない。

2. 研究目的

国内貸切トラック輸送の 8 割強が関東域内、関西域内といった「地域ブロック内」で動いている[1]。地域ブロックをまたぐ長距離輸送は 1~2 割程度の貨物の偏りがある上に、比較的往復実車が進んでいる分野でもあるため、本研究では域内中ロット貨物輸送の効率化を目的としたシミュレーションシステムを構築する。

図 1 は、複数のトラックが個別に配送を行っている様子を示している。トラック 1 とトラック 2 は近隣の営業所から出発し、それぞれの荷物を同一地区に配送している。トラック 3 はトラック 1 が向かう先の地区からトラック 1 の発地付近に配送している。現状ではこれらのトラックは、それぞれの配達が終われば空荷の状態でも営業所まで戻っている。これに対し、たとえばトラック 1 がトラック 2 の集荷地にも一旦立ち寄って同一地区に配送し、さらにトラック 3 の集荷地にも立ち寄って集荷し、もとの発地付近に配達するような計画が立てられれば積載率の改善がみこまれる。

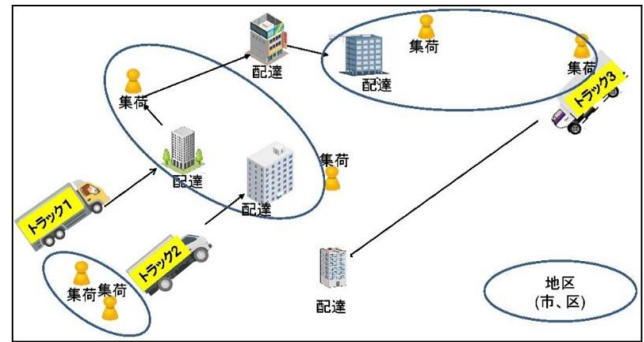


図 1 域内中ロット貨物輸送の現状と集約

もちろん時間や品目、トラックの大きさなど様々な制約が存在するため一概にこのような集約が可能な訳ではないが、本研究では、多数の配送の中から様々な条件を満足する配送計画を自動的に立案できる仕組みを、Ambient Calculus による物流システムのモデル化手法を用いて提案する。

Ambient Calculus(AC) [2] とは、Microsoft 研究所の Cardelli 等によって提案されたプロセス代数の一種である。ACにおいて名前のついた[括弧]は ambient と呼ばれ、現実世界の対象物やその階層構造を入れ子で表現する。これにより動的に変換する階層構造を持ち並行に動作する対象物を数理的にモデル化し、モデル上で対象物の性質を解析するために用いられる。この性質は特に物流システムをモデル化するために適している。例えば、図 5 の①にある Chiyoda[Truck[...]]は千代田区を表す Chiyoda ambient の中にトラックを表す Truck ambient が存在しており、これにより現実世界の階層構造を表現している。さらに現実の物流システムのモデル化に必要な時間的制約を表現するために AC を時間拡張したプロセス代数である混合型時間 AC(HTAC) が文献 [3] で提案されている。本研究では域内貨物輸送をモデル化するために AC の文法をさらに拡張して用いる。

3. 域内物流モデル化のための AC の拡張

本章では、AC を用いて現実の域内貨物配送を正確にモデル化するために行った AC の文法拡張及び処理系の実装について述べる。

3.1 条件付き Enter Capability

中ロット貨物の具体的な品目として、書籍・水産品・電気機械など様々な貨物が運搬されている。ただし様々な貨物を混載する場合に一定の条件を満たす必要がある。例えば水産品を搭載したトラックに書籍を載せることは、衛生的に不可能であるため条件次第でトラックに搭載させるか

[†] 近畿大学 KINDAI UNIVERSITY

判断する機能が必要である。そこで様々な条件に一致する `ambient` にのみ遷移する、条件付き `Enter Capability` の `qin` を新たに導入する。`qin` を用いた具体的なプロセス式を図 2 に示す。①・②・③はトラックを表しており、その中の `Type` は積載できる種別、`MaxWeight` は積載容量、`Weight` は現在の積載量を表している。④は貨物を表しており、その中の `Type` は貨物の種別、`Weight` は貨物の重量を表している。通常の `in Capability` は単に引数にある名前と同じ `ambient` に入るという動作を表すが、`qin Truck` は種別と積載量の条件が満たされる場合のみトラックに入るという動作を表す。

また混載で考慮すべき条件として、すでに積まれている荷物と同方面に向かう荷物のみを積み込み可とする必要がある。図 3 は千代田区に荷物を運送し豊島区行きの荷物を持つトラックを示している。この時、千代田区内にある荷物の内同方向である新宿行きの荷物は積み込み可とし、逆方向の台東区行きの荷物は積み込み不可とする必要がある。現在地とそれぞれの貨物の緯度経度を元にして、適切な方面に向かう荷物のみを抽出する機能を `qin` に持たせることで、方角的にも混載可能な荷物を積載するという動作を表すプロセス式とすることができる。

```
Truck#1[Type[<Fish>]|MaxWeight[<3000>]|...] ①
Truck#2[Type[<Machine>]|MaxWeight[<3000>]|Weight[<2000>]|...] ②
Truck#3[Type[<Machine>]|<Car>]|MaxWeight[<4000>]|Weight[<1000>]|...] ③
Ca[qin Truck|Type[<Machine>]|Weight[<2500>]] ④
```

図 2 条件付き Enter Capability の具体例

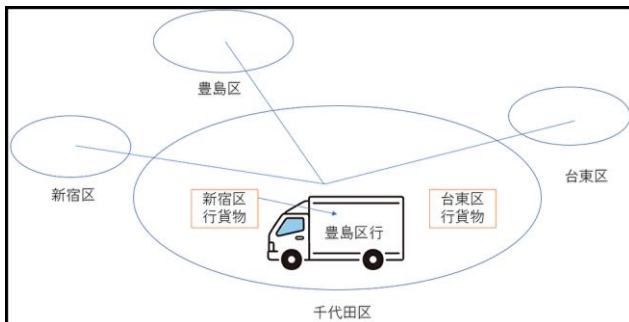


図 3 条件付き Enter Capability の拡張例

3.2 制御 ambient 及び capability の優先

非選択的決定など遷移できる `capability` が複数存在するようなプロセス式の場合、そのすべての遷移パターンを実行する全パターン実行機能を我々は実装した。図 4 は物流シミュレーターで扱う域内トラック輸送のプロセス式を簡易的に表したものである。①は実際のトラックの動作を表す `capability` の `out Area1.in Area2` によりトラックが移動す

```
Area1[
  Truck#1[out Area1.in Area2]|...] ①
  |Truck#2[load[...]|open load[...]] ②
]|
Area2[|]
```

図 4 域内トラック輸送の簡易的なプロセス式

ることを表している。これに対し、②は実際の動作ではなく、積み込みが完了したかどうかを制御しており `open load` により `load[...]` が展開されることで積み込みの完了を表している。この式を遷移させる場合、常に遷移できる `capability` が動作と制御の 2 パターン存在しているので、全パターン実行を行う際にはパターン数が指数関数的に増加してしまう。そこで動作 `ambient` は頭文字を大文字で表記、制御 `ambient` は頭文字を小文字で表記を行い、制御動作を優先的に遷移させることにより、遷移のパターンを大幅に削減することが可能となる。

3.3 データベースから最適な展開を行う Open Capability

目的地の異なる複数の荷物をトラックが積載するというプロセス式を遷移させる場合、単純な実装では、次の目的地がランダムに決まってしまう。全パターン実行によりすべての行き先を網羅することは可能ではあるが、現実の配送をモデル化するためには、最も近い目的地を持つ荷物を先に届けるようなプロセス式にする必要がある。本研究では、データベースに記録された各地の緯度経度から各荷物の目的地までの距離を計算し、最も近い目的地を表す `ambient` を展開し選択を行う `capability` である `openDB` を新たに導入する。図 5 に具体例を示す。①・④・⑤は千代田区・豊島区・新宿区を表しており、千代田区にトラックが存在している。②・③は貨物の目的地を示している。`openDB` により②・③の `dest` 内の式のそれぞれの目的地を読み取り、千代田から豊島と新宿の距離を計算し、より近い新宿が選ばれることで `dest#2` が展開され、トラックが新宿へと向かうという動作を表す仕組みになっている。

```
Chiyoda[Truck[openDB dest ①
  |dest#1[out Chiyoda.in Toshima] ②
  |dest#2[out Chiyoda.in Shinjyuku]|...] ③
Toshima[] ④
Shinjyuku[] ⑤
```

図 5 openDB の例

4. おわりに

本研究では域内物流モデル化のための AC 文法拡張を提案した。この中で組み合わせ爆発の対応もある程度行ったが十分ではない。我々は、本研究とは別に独自の強化学習アルゴリズムを用い、膨大な組み合わせの中から最適解を発見する ϕ 法 (ファイ法) と呼ばれる手法を提案している。 ϕ 法は現実的な時間で他の手法よりもより最適に近い組み合わせを発見することが可能な技術である。本研究で提案した域内貨物輸送のモデル化手法と ϕ 法を組み合わせることにより、複数の貨物の混載をする際に発生する組み合わせ爆発問題に対応し、トラックの台数や走行距離などの観点でより効率の良い配送計画が立てられるシミュレーターを提案する予定である。

参考文献

- [1] 国土交通省.全国貨物純流動調査(物流センサス)報告書 (2017).
- [2] L. Cardelli and A.D. Gordon. Mobile Ambients. LNCS, Vol. 1387, pp. 140–155 (1998).
- [3] 藤坂吉秀, 稲森啓太, 樋口昌宏, 加藤暢. 物流記述のための混合型時間アンビエント計算. 情報処理学会論文誌プログラミング (PRO), Vol. 10, No. 4, pp. 12–27 (2017).