

分散探索戦略に基づく配送経路問題の一解法

函館高専 ○伊藤 圭 浜 克己

要 旨

本研究の目的は、有軌道空間における配送経路問題を効率的に解決することである。本研究では、マルチエージェントシステムを採用し、競売等の処理を導入して、計算負荷の軽減を図るとともに柔軟性を持った分散的なシステムの開発を試みる。

1. 緒 言

配送経路問題は、運送業界において輸送コストの軽減や顧客サービスの向上などに関わり、その重要性は高く、従来から多くの研究が行われている^{1) 2)}。

本研究では、有軌道空間における配送経路問題について、配送先の選択および配送経路の設計を行うとともに、負荷の軽減や柔軟性を持たせるために、それらの作業を各トラックが分散的に行う方式について検討し、計算機実験を通じてその有効性を調査する。

2. 問題設定

対象となる空間を2次元の平面と仮定し、トラックエージェント（以下 TA）に以下の前提を置く。

- 1) 空間内は有軌道であり、TA はその軌道に沿って移動しなければならない。
- 2) 各 TA には、最大移動可能距離に制限がある。対象問題 P は、以下の三項組で表現する。

$$P = \{S, Q, A\} \quad (1)$$

S : 2次元空間内の状態集合

$$S = \{(x, y, s) \mid 0 \leq x \leq X, 0 \leq y \leq Y, s \in \{0, 1\}\}$$

X : 空間の X サイズ, Y : 空間の Y サイズ
s : 移動の可否 (0 : 移動可, 1 : 移動不可)

Q : 拠点の状態集合

$$Q = \{(Q_i, q_i) \mid i = 0, 1, \dots, N_Q\} \quad N_Q : \text{店舗数},$$

Q_i : 拠点番号 ($i = 0$: センター, $i \neq 0$: 店舗),
 $q_i \in S$: 拠点位置

A : TA の状態集合

$$A = \{(A_i, a_i, r_i, t_i, T_i) \mid i = 1, \dots, N_A\}$$

N_A : 最大 TA 数, A_i : TA 番号
 $a_i \in S$: 現在位置, r_i : 配送店舗数
 t_i : 配送移動距離

$$T_i = (Q_0, \dots, Q_j, \dots, Q_0) : \text{配送経路}$$

各 TA は、配送センターから出発して、配送センターに戻ることをとする。

評価関数 E は以下のように設定する。

$$E = w_1 \cdot N_r + w_2 \cdot \sum_{i=1}^{N_r} t_i \quad (2)$$

N_r : 使用 TA 台数, w_1, w_2 : 重み係数

以上より、配送経路問題は以下のように定式化される。

$$\text{Find } T_i \text{ for } \forall i \text{ such as minimizing } E$$

$$\text{subject to } t_i \leq t_{\max} \quad (3)$$

3. マルチエージェントシステムの適用

本研究では、対象問題の解法としてマルチエージェントシステム (MAS) を導入する。

(1) 初期経路の設定

各 TA は、最大移動可能距離の範囲内で配送する店舗をランダムに決定し、配送すべき店舗がなくなるまでこの処理を続ける。この際、各店舗の位置は既知であるが、一般的な配送経路問題と異なり、本研究では有軌道空間を対象とするため、店舗間の経路および距離を直接求めることはできない。したがって、各 TA は周囲 4 方向の状態を検出できるセンサを有し、その状態に応じて移動できる方向へランダムに移動する。開始店舗であるセンターから出発し、このようにして現在の店舗から候補となる次の店舗までの経路が求まるたびに、同様の方法でその店舗からセンターまでの経路も求め、それら 2 つの距離を現在の店舗までの距離に付加した総和が最大移動可能距離を越えなければ、TA はその候補の店舗を自らの管轄に加えるものとする。距離の総和が制限を越えた場合には、その候補の店舗は、再びセンターから出発する別の TA の候補となる。

このように、目的地である店舗に到達するまでの経路が一つの部分解となるが、一度通過した道を再び通過した場合、その間の経路はループと見なし省略する。移動に関して、以下の 2 つを優先している。

1) TA の目的地である店舗の方向。

2) 過去に探索したことのある同じ 2 店舗間の経路。ただし、以前に探索した経路よりも短い経路が得られない限り後者の優先度は更新されない。

すべての店舗が選出されたら、TA 毎に経路の最適化処理を行うが、計算時間の短縮のために、一度探索した店舗間の経路と距離は記憶しておき、最適化処理で再び同じ 2 店舗間の経路と距離が必要になった時は、探索をせずにこの記憶した結果を呼び出して使用する。2 店舗間の経路探索は、以後もランダムに指定した 2 店舗に対して行われ、よい結果が得られれば更新あるいは追加される。

(2) 競売

必要な TA の台数を減らすために、TA 間で店舗に関する競売を行う。TA は、現在所有する店舗から 1 店舗

を除いた状態で経路の最適化処理を行う。このとき、所要距離が最も減少するような店舗を競りに出す。

この作業には、TA が管轄している店舗の数だけ経路の最適化処理が必要となるが、前述のように一度計算した2店舗間の経路と距離はそのまま使用できるので、それによって処理時間を軽減することが可能である。また、このようにして競りに出された店舗は、各 TA が実際にその店舗を取り入れた状態で評価を行い、最も所要距離の増加が少ない TA の元に管轄が移動する。ただし、結果次第では、競りに出した TA の元に戻ることもある。経路の最適化処理には、m-step 2-opt 法を使用している。

<m-step 2-opt 法>

$P \cdot P+1, P+2+m \cdot P+3+m$ と $P \cdot P+2+m, P+1 \cdot P+3+m$ の経路長を比較して、より短い方を適用する方法である。図1に、基点となる店舗 $P=1$ 、 $m=1$ の時の初期状態と処理後の状態を示す。

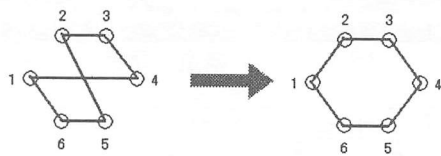


Fig.1 Example of m-step 2opt method (m=1,P=1)

4. 計算機実験

本研究では、以上の方法論にしたがい計算機実験を行った。実験条件は以下の通りである。また実験に使用する対象空間を図2に示す。

<実験条件>

- ・対象空間のサイズは 41×61 で配送センターはその中央にあり、配送先の店舗数は50とする。
- ・各 TA の最大移動可能距離は200とする。
- ・評価関数の重み係数を、 $w_1=20$ 、 $w_2=20$ とする。

初期経路を決定した後に、各 TA に競売を1回ずつ行わせ、その後ランダムに2店舗を選び、それらの間で経路探索を行い、改善された場合にその経路と距離を更新する。競売からここまでの流れを1ステップとして、50ステップまで繰り返す。

図2にこれらの処理を適用し、10回の試行中の最良結果に対して得られた経路を図3に、また、TA の台数、経路長、評価値の変化を図4にそれぞれ示す。図4より、3つの項目全てが同じ傾向で減少し、約20ステップで解がほぼ収束していることがわかる。10回の試行結果に対し、50ステップ後の必要な TA 台数は3~5とばらついた。その原因として、初期経路や初期 TA 台数の違いなどもあるが、最小値となった TA 3台の時の結果を調べたところ、途中で2店舗間の経路が改善され、距離が減少していた。これによって、収束しかけた競売の処理が再び活発になり、結果的に TA 台数の減少に繋がったと考えられる。

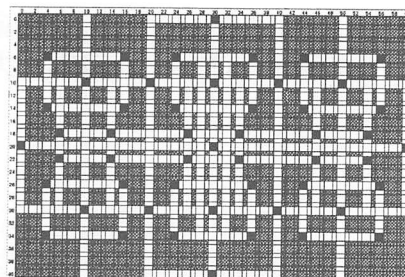


Fig.2 Object space for simulation

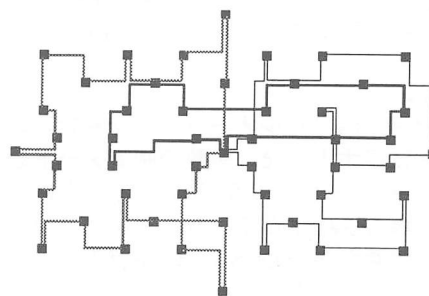


Fig.3 Obtained vehicle routes

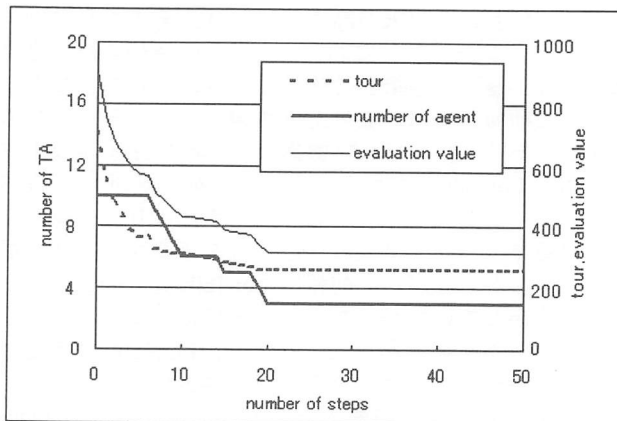


Fig.4 Transition of each state

5. 結言

本研究では、各 TA がそれぞれの状況を考慮した分散システムを構築した結果、ある程度の規模の問題まで対応できることが確認された。今後は、競売処理を強化するために TA 間の店舗の交換などを考慮するほか、現在競売に出す店舗を決定する処理に大きな負荷が掛かっているのを、それを軽減するための新しい処理を考案し、大規模問題にも対応できるようにする予定である。

参考文献

- 1) Thangiah, S. R. et. al., An Agent Architecture for Vehicle Routing Problems, SAC2001, 517-521, 2001
- 2) Gambardella, L. M. et. al., MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows, Technical Report, IDSIA-06-99, 1999