

深層強化学習を用いた複数拠点の配送計画

佐藤凜太郎[†] 辻康考[‡]

九州大学工学部[†] 九州大学大学院工学研究院[‡]

1 はじめに

物流業界は、営業収入が約 25 兆円、就業者数が全産業就業者数の約 4%を占めるが、近年の少子高齢化による運転手不足とともに、都市部では宅配需要の急増、逆に過疎地では積載効率の低下が大きな課題となっている[1]。配送車の積載量を満たしながら顧客の需要に応じて配送する経路を生成する問題（使用車両台数あるいは総走行距離の最小化）は、配送経路問題（VRP: Vehicle Routing Problem）と呼ばれ、多くの数値モデル・解法が提案されてきた。

近年、多くの分野の課題に対して深層学習を用いた研究が活発に試みられている。VRP に対しても深層強化学習（DRL: Deep Reinforcement Learning）による解法が提案されている[2]-[4]。本稿では、より現実に近い問題である、配送拠点が複数で利用可能な配送車両の台数が指定された問題：複数拠点の配送経路問題（MDVRP: Multi-Depot Vehicle Routing Problem）を DRL の枠組みで解く手法について提案する。

2 関連研究

VRP に対しては、既に DRL を用いた手法がいくつか提案されている[2]-[4]。

特に Kool らは[4]、Multi-Head Attention 層（MHA 層）[5]を取り入れたモデルを用いて、VRP を含む種々の経路問題に対して解の精度・計算効率を向上させている。

3 提案手法

3.1 配送車の選択

複数拠点の場合、各時点でどの拠点に配置されている配送車を動かすか選択しなければならない。各時刻において、グラフ中のノードの座標や需要、各配送車の位置・積載量を考慮して、動かす配送車および訪れるノードを逐次的に選択する手法を提案する。

3.2 モデル構造

Kool らが用いたモデル構造同様、グラフの情報を保持して伝達する Encoder と、伝達された情報を元に、確率分布を各時刻で出力する Decoder の 2 つで構成される。

i 番目 ($i \in \{1, \dots, n\}$) のノードの特徴量ベクトル x_i は、Encoder にて、高次元のベクトル空間に埋め込まれ、続く N 個の MHA 層を通過し、隠れ状態ベクトル h_i^N として出力される。Decoder では、この Encoder から伝達された h_i^N を各配送車で共有する。Decoder において、グラフ全体の情報を保持する隠れ状態ベクトルの平均 $\bar{h}^N = \sum_{i=1}^n h_i^N$ 、 m 番目 ($m \in \{1, \dots, M\}$) の配送車が、時刻 $t-1$ で位置していたノードの隠れ状態ベクトル $h_{\pi_{t-1}, m}^N$ 、時刻 t での積載量 $D_{t, m}$ を結合してコンテキストベクトル $h_{C, t}^N$ を次式により生成する。

$$h_{C, t}^N = [\bar{h}^N; h_{\pi_{t-1}, 1}^N; D_{t, 1}; \dots; h_{\pi_{t-1}, M}^N; D_{t, M}] \quad (1)$$

時刻 t で生成されたコンテキストベクトル $h_{C, t}^N$ を MHA 層に入力して、ベクトル $u_{i, m, t}$ を出力させる。

グラフ $s = \{x_i\}_{i=1}^n$ 、時刻 1 から $t-1$ までの経路 $\pi_{1:t-1}$ 及び使用した車両 $y_{1:t-1}$ が与えられたときの、時刻 t で動かす配送車および訪れるノードを選択する条件付き確率分布 $p(\pi_t = i, y_t = m | s, \pi_{1:t-1}, y_{1:t-1})$ へと、 $u_{i, m, t}$ をソフトマックス関数を用いて次式により変換する。

$$p(\pi_t = i, y_t = m | s, \pi_{1:t-1}, y_{1:t-1}) = \text{softmax}(u_{i, m, t}) \quad (2)$$

3.3 学習手法

経路 π を生成終了後、ノード間の重みの総和であるコスト $L(\pi, y | s)$ を算出する。また、各時点で求めた確率分布が互いに独立として、配送車 y を使用して経路 π を生成するときの確率分布 $p_\theta(\pi, y | s)$ を次式により表現する。

$$p_\theta(\pi, y | s) = \prod_t p(\pi_t = i, y_t = m | s, \pi_{1:t-1}, y_{1:t-1}) \quad (3)$$

Solving The Multi-Depot Vehicle Routing Problem using Deep Reinforcement Learning

[†] Rintaro Sato, School of Engineering, Kyushu University

[‡] Yasutaka Tsuji, Graduate School of Engineering, Kyushu University

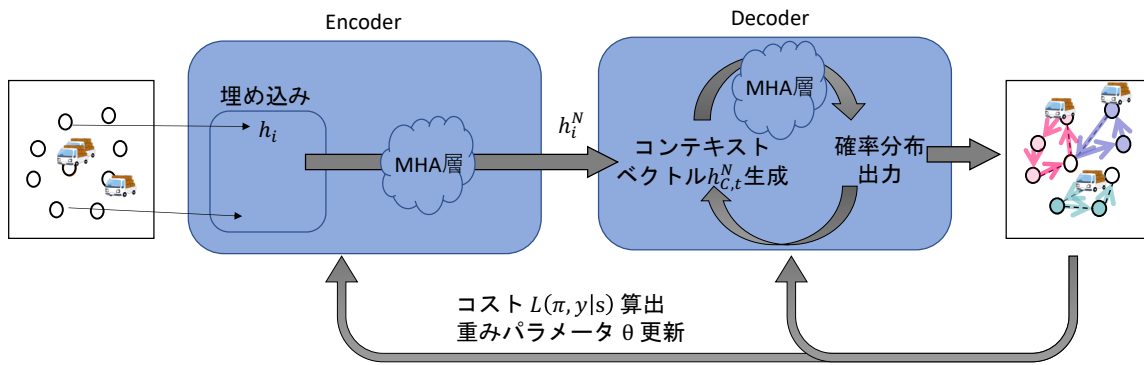


図 1 MHA 層を用いた複数拠点に対応するモデル構造及び学習方法

損失関数 $E[L(\pi, y|s)]$ の勾配を、REINFORCE アルゴリズム [6] の近似により、 $p_\theta(\pi, y|s)$ と $L(\pi, y|s)$ を用いて推定する。推定された勾配を用いて、Encoder 及び Decoder の重みパラメータ θ を勾配降下法で更新して学習させる (図 1)。

4 実験

顧客数を 100, 拠点数を 3, 各拠点で使用可能な車両台数を 5, 全ての配送車の積載量を 2.0 として、性能比較を行った (表 1)。表 1 における MHA を本研究で提案するモデル構造とし、経路生成後、局所探索法である 2-opt を各経路に適用した (図 2)。表 1 の時間の項目の括弧内は、学習済みモデルの読み込み時間を含めた処理時間を示す。比較手法として、遺伝的アルゴリズム (GA), 数理最適化ソルバーである Google OR-Tools [7] を用いた。全ての手法は、共通して Python で実装した。MHA の実装に関しては、深層学習ライブラリである PyTorch [8] を用いた。

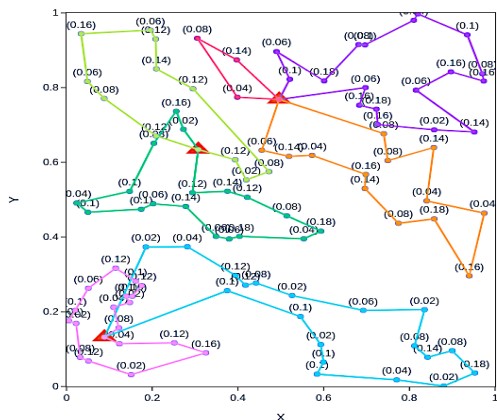


図 1 表 1 の MDVRP の解の描画例 (顧客ノードの括弧内は、需要を表す)

5 おわりに

本稿では、DRL を用いた MDVRP の解法を示した。2-opt による解の改善を図ることで、GA と同精度

の解を高速に求められた。複数拠点だけでなく、単一拠点の配送計画に適用できるため、既存の Kool らの手法の拡張といえる。配達時間を考慮した時間制約も、今後導入する予定である。

表 1 実験結果

	MHA+2-opt	GA	OR-Tools
時間 (s)	1.388(4.766)	35.19	1.306
コスト	10.27	10.27	9.924

参考文献

- [1] “物流を取り巻く現状について 国土交通省 物流政策課 2018年10月11日 資料2.”
- [2] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, “Neural combinatorial optimization with reinforcement learning,” in *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings*, pp. 1–15.
- [3] M. Nazari, A. Oroojlooy, M. Takáč, and L. V. Snyder, “Reinforcement learning for solving the vehicle routing problem,” *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, pp. 9839–9849, 2018.
- [4] W. Kool, H. van Hoof, and M. Welling, “Attention, Learn to Solve Routing Problems!,” *arXiv*, 2018.
- [5] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-Decem, no. Nips, pp. 5999–6009.
- [6] R. J. Willia, “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning,” *Mach. Learn.*, vol. 8, no. 3, pp. 229–256, 1992.
- [7] Inc. Google., “Google’s optimization tools(Or-Tools),” [URL] <https://developers.google.com/optimization/routing>. 2016.
- [8] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, “Automatic differentiation in PyTorch.” 2017.