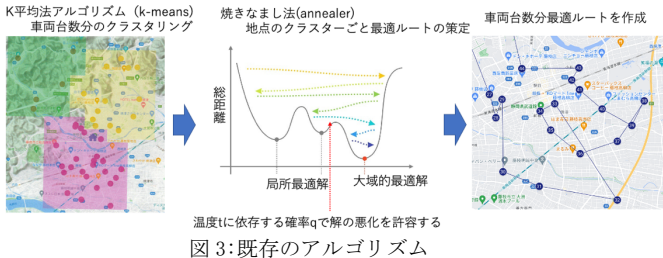


各地点間に関し、API(Google Maps Distance Matrix api)で、季節、時間帯、曜日、混雑度を指定しない通常モードで求めた地点間の移動距離と地点間のユークリッド距離との比較を行った(図3を参照)。図3から明らかな通り、どの地点間においてもユークリッド距離で求めた地点間の距離が短いことがわかる。

また、システムのもう一つの問題点は、その解法にある。SBS 情報システムの初期のごみ収集管理システムでは、車両台数をKとするk平均法アルゴリズム(k-means)でクラスタリングを行った後、クラスターごとに焼きなまし法を用いてルート策定を行っていた。

しかしそれでは、クラスターの隣接近くにある地点に関しては、経路によっては他のクラスターに属した方が効率的になる場合も多い。またクラスタリングの距離としてユークリッド距離(直線距離)を採用しているため、道路状況を考慮していない。そのため、実際にはクラスタリングの分類が間違えているため、解が最適ではない場合がある。



2.2 評価指標

今回は求める地点間のコスト精度の評価指標として、平均絶対値誤差(MAE)を用いた。(式(1))

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (1)$$

平均絶対値誤差は各データに対して「API(評価データの値を真の値とし、ユークリッド距離のような予測した値との差(誤差)の絶対値を計算し、その総和をデータ数で割った値(平均値)を求めたものである。

図2では、APIとユークリッド距離で求めた地点間の距離の平均絶対値誤差(MAE)は1535.71mと非常に大きな値になった。以上から、正確なコスト取得のためには、ユークリッド距離ではなくAPIで求めた距離を使用すべきである。

2.3 Google APIの課題

「Google Maps Distance Matrix API」を使用し実際の値に近い時間行列、距離行列を作成したが、APIは利用量に応じて料金が課金される従量課金制である。そのため地点数が増えると使用料金が増加し採算が取れないため、現実的に利用することが困難となる(図4参照)。事実、F市の45地点の双方向距離行列を求めるのに約2000円程掛かった。ごみ収集システムを人口規模が約5倍の静岡市で展開した場合、1日の収集数が単純計算でF市の約5倍の200が予想され、その費用は単方向行列で3万円、双方向行列でその倍の6万円が平均的にかかる。ごみ収集を週1回と

すると、毎回違った家庭から収集依頼が来るとすると、双方向行列取得の費用として、月24万円、1年には288万円の費用が想定され、現実的ではない。

そこで本論文では、各地点間の実移動時間を機械学習によって予測を行い、各地点間の移動コストの予測を行った(4節)。

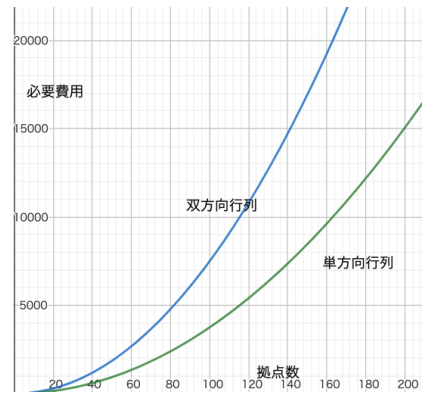


図4:拠点数に応じた利用コスト(円)

3. コストの違いによるルート策定後の解の比較

3.1 使用するソルバー

VRPに関しては数多くの研究が行われているため、多数の近似解を求めるアルゴリズムが研究開発されている。本研究の目的は、現実に近い地点間距離をできるだけ正確に求め、ごみ収集システムで実際に利用することであるので、ルート作成に関しては既存のソルバーを採用した。近似解を求めるツールとして本論文では多数の制約条件にも対応できる「OR-tools (Google OR-tools)」を使用した。「Google OR-tools」^[5]とは、Googleのオープンソース数理最適化ツールであり、車両ルーティング、フロー、整数および線形計画法、制約計画法などの解が非常に多く存在する問題に対する最良の解決策を見つけようとするソルバーである。

3.2 ユークリッド距離と実距離(Google API)の比較

「OR-tools」を用いて、ユークリッド距離とAPIで求めた移動コスト(地点間の実際に近い距離)を使用して車両1台分のTSPを解き、ユークリッド距離の場合とAPIによる移動コストのそれぞれの場合について、最適な移動順番を求めた。その結果から、ユークリッド距離と実距離(APIによる実際に近い距離)のそれぞれについて、総移動距離、総移動時間を求めた。以下がその結果である(表1)。

	ユークリッド距離	実距離
総移動距離	97.188Km	90.206Km
総移動時間	14756秒	13513秒

表1 ユークリッド距離と実距離

比較の結果、移動距離は6.982Km(7%)、移動時間は1243秒(9%)改善する結果となった。結果、ユークリッド距離では正確な実際の距離がわからないため、実距離をコストとして採用することが良いことが確かめられた。

3.3 SBS 初期システムとの違い (改善)

次に、SBS 初期システムで採用していた当初のアルゴリズム (K 平均方+焼きなまし法) と「Google OR tools」を使用した場合に差がでるかを検討した。結果を表 2 に示す。

	当初のアルゴリズム	Google OR tools
総移動距離	115.4Km	97.2Km
総移動時間	16500 秒	14756 秒

表 2 SBS 初期システムとの近い (改善)

「Google OR tools」と当初のアルゴリズム (K 平均方+焼きなまし法) を比較すると、総移動距離は 18.2Km(16%) 改善、総移動時間は 1744 秒(11%) 改善した。

4.OSM を用いた API 利用回数の削減

API の利用には使用料金が掛かり、地点数が増えると利用料金がかさみ、API を営利的に利用することは現実的ではない。API の利用回数を削減するため、API の最小限の利用のもと、地点間の最短経路を求めることにより、地点間の実際に近い移動距離と移動時間の推定値を求める。

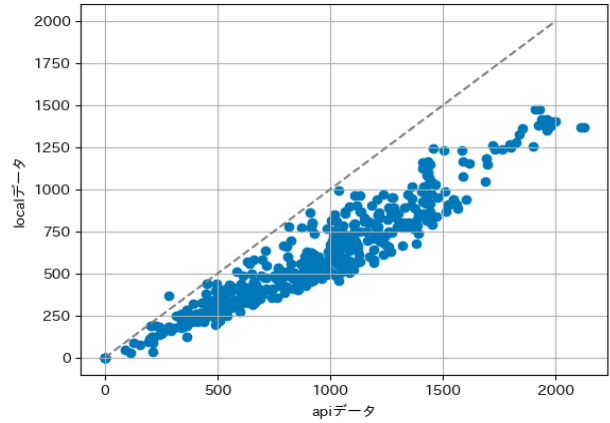


図 5:API と OSM の各拠点の所要時間の比較

地図データとして、OSM (open street map) [5]を使用した。OSM とは、誰でも自由に地図を使えるようにオープンデータの地理情報を作るプロジェクトの総称であり、オープンソースで地図データが公開されている。[6]実験では、F 市の地図データをダウンロードし、そのデータを元にしてダイクストラ法を用いて地点間の最短経路を求めた。地点間の最短経路から求めた移動経路から、道路に付随している最大速度を使用して、地点間の推定移動時間を算出し

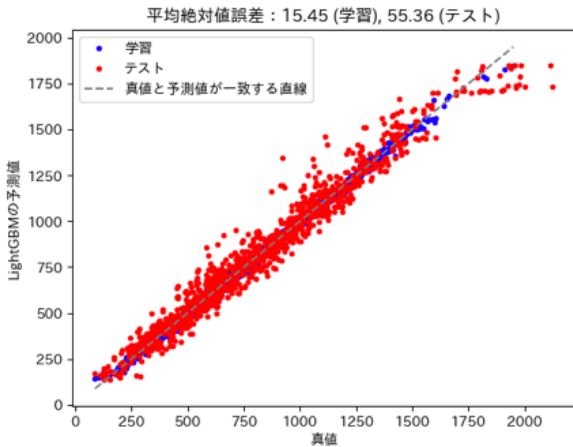


図 6:LightGBM の予測値と真値の比較

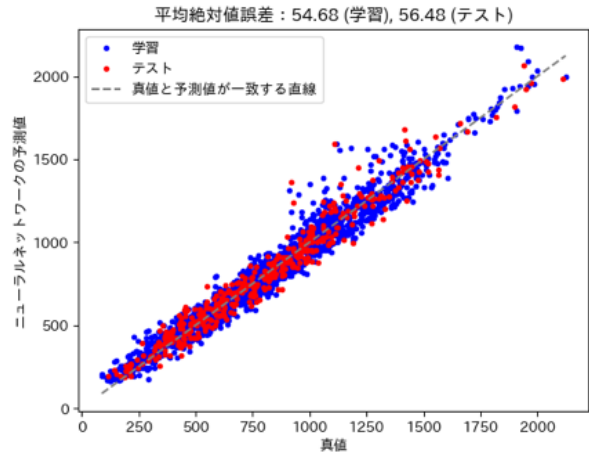


図 7:ニューラルネットワークの予測値と真値の比較

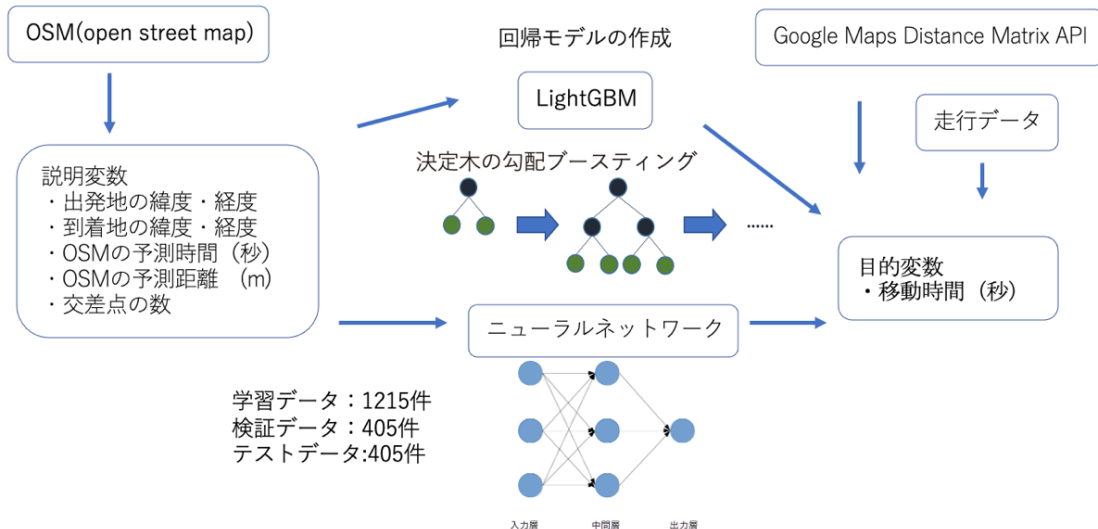


図 8:学習モデルの構築について

た。OSM から予測した地点間の所要時間 (local データ) と API による地点間の所要時間の散布図を図 5 に示す。

本来であれば OSM と API の値が同じになる必要があるが、必ずしも同じにはならない。平均絶対値誤差が 311 秒と大きな誤差が生じていて、改善を行う必要がある。

5. 機械学習を用いた誤差の削減

5.1 地点間コストという教師データの学習

OSM で求めた予測時間と実際の時間 (API による時間) の階差が大きいため、機械学習を用いて、OSM を使用して求めたデータが API を利用して求めた予測時間に回帰するモデルの作成を行った。機械学習には、**決定木の勾配ブースティング機械学習手法 LightGBM^[7]**を用いた予測 (図 6) と **フィードフォワード型ニューラルネットワーク**を用いた予測 (図 7) を構築し、比較を行なった。目的変数として移動時間 (秒)、説明変数として (出発地の緯度・経度、到着地の緯度・経度、OSM の予測時間、OSM の予測距離、交差点の数) を使用して学習した。

5.2 学習結果

LightGBM で学習した結果は平均絶対値誤差が 55 秒と、API と OSM の値を比べた場合の誤差 311 秒に比べて、256 秒も誤差が少ない結果を得ることができた。学習データの平均絶対値誤差は 15.4 秒と高い精度で学習を行えていることが分かった。

また、更なる精度の向上のため「Keras」^[8]を用いて 3 層構造の単純なフィードフォワード型のニューラルネットワークを作成し、検証を行った (図 7)。テストデータでの平均誤差は 56.48 となり、「LightGBM」での平均誤差 55.36 と近く、重回帰分析とニューラルネットワーク分析の学習誤差には大きな差は生まれなかった。

作成したモデルを図 5 との平均絶対値誤差を比較すると、「LightGBM」を用いた場合は 255.82 秒、「ニューラルネットワーク」を用いた場合は 253.9 秒と、元データに比べ値が小さくなった。2 つの回帰モデルの差は約 1 秒とほぼ差がないため、システム上で高速かつ少ない計算量で動作する「LightGBM」を採用した。

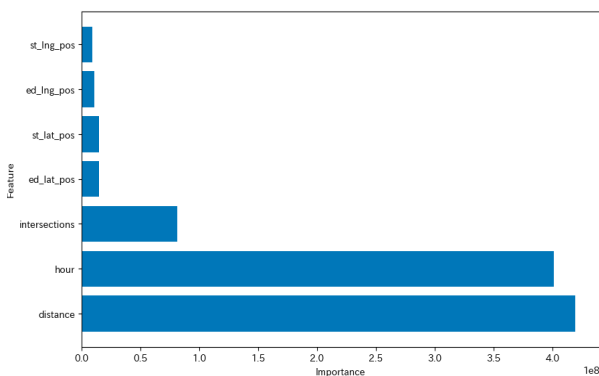


図 9: 特徴量重要度の可視化

また学習を行なった際に、説明変数が目的変数にどの程度の影響を与えたかの比較を行うため、特徴量重要度の可視化を行った (図 9)。影響度が高い項目としては上から OSM で求めた予測距離、予測時間、交差点の数が挙げられ

る。影響度が低い項目としては出発場所と到着場所の緯度経度の情報は少ないことが挙げられた。

5.3 最終結果

API で取得した距離行列と OSM を入力データとして重回帰分析を使用して求めた時間行列データをもとに OR-tools を用いて最短経路の近似解を求めた。

表 3 機械学習アルゴリズムの比較

	API を使用して求めた距離行列	LightGBM を使用して求めた距離行列
総移動距離	90.648Km	89.784Km
総移動時間	13164 秒	13059 秒

API のデータをコストとして用いた場合と「LightGBM」を用いて作成したコストから「OR-tools」を用いて最適経路の作成を行い、経路の総移動距離の比較 (表 3) を行うと 864m 総移動時間にして 105 秒の差があった。これは総移動時間移動距離から比べるとわずかな差であり OSM を用いた重回帰分析の学習結果の精度の高さを確認することができた。

6. システムの実装

6.1 実システムの実装

本論文では現実問題におけるゴミ収集における配送計画問題の解の改善を行なったが、これらのアルゴリズムを web アプリケーションとして実装を行う必要がある。そのために以下のサーバーレス環境の構成を AWS 上で行なった (図 10)。特徴としてサーバーレス環境で構築を行なったため運用コストを抑えることができ、可用性を高めることができる。また単体で動作が可能なマイクロサービスを構築したため、アプリケーションとして呼び出すことが可能となる。地点の情報、トラックの情報などの Json データを post することで計算され、最適化された経路情報が結果としてレスポンスされる構造になっている。

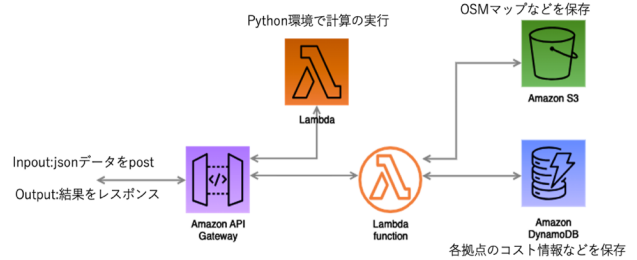


図 10: AWS のシステム構成図

7. まとめ

本研究では現実の配送最適化問題に対して、主に地点間のコスト予測による解の改善を行なった。

実験の結果ユークリッドを用いたコストでは、実距離を使用する場合に比べて、実際の解が悪化することが確認できた。実距離コストの作成に「API(Google Maps Distance Matrix API)」を使用することで、現実に近い距離、時間行列を作成することができる。しかし、ゴミの収集場所は一つの市内に数千箇所存在しているため、API の利用費用が

指数関数的に増加する。そのため営利的に API を利用してコストを求めることは現実的ではない。

7.1 本論文のアイデアの採用

その問題を解決するため、「OSM(open street map)」よりダウンロードした map データからダイクストラ法を用いて 2 点間の最短経路を求めた。そこから予想到着距離と所要時間を求め拠点間の移動コストを作成した。しかし API を使用して求めたコスト行列と比較すると、平均絶対値誤差が 311 秒と誤差が大きい。誤差削減の為、機械学習を行い誤差 55 秒まで削減することができた。それらのシステムを web アプリケーションとして実装を行うため AWS を用いて実装を行い、汎用的かつ高い可用性を持つシステムの設計を行なった。そのためこれらのシステムはごみ収集だけでなく、配送、送迎、配達などさまざまな分野への応用が期待でき、CO₂ の排出削減や作業の効率化などさまざまな分野への応用が期待できる。

8. 今後の展開

8.1 システムの活用

今後は、実際に「戸別ゴミ収集システム」を利用している自治体と連携を行いながら、実証実験を推し進める。それらの実験で得られた走行データを、機械学習の学習データとしてシステムにフィードバックすることで、さらなる最適化を目指す。具体的には走行時間やごみの量などのデータを、事前に予測したデータと比較し、それらの差分を学習することにより、予測精度の向上を目指す。

8.2 本提案の汎用性と今後の計画

本論文で提案したアルゴリズムや手法は汎用的であり、ごみ収集だけでなく、送迎や災害時など他の分野にも応用可能である。そのため開発したシステムを応用して、災害が発生した場合に出る災害廃棄物の推定と、廃棄物の回収に関わる環境負荷 (CO₂ の排出量) の推定を行っている。各車両の走行順や走行距離、容量の推定を行い、そこから CO₂ の排出量の推定と環境負荷の削減を目指す。その結果を、自治体の災害対応や災害廃棄物の廃棄場所の意思決定に利用する予定である。

参考文献

- [1] 物流の 2024 年問題とは: 解決に向けた全日本トラック協会の取り組み (特集 2024 年問題をめぐる対策), 月間下水道, Vol.47, No.5, 2024 年 4 月.
- [2] P. Toth and D. Vigo: Vehicle Routing: Problems, Methods, and Applications, SIAM(2014).
- [3] 令和 3 年度オープンイノベーション推進事業の実証実験を完了 藤枝市「戸別ごみ収集業務管理システム」, <https://www.sbs-infosys.co.jp/news/information/98.html>
- [4] Google Maps Platform: Build awesome apps with Google's knowledge of the real world. <https://developers.google.com/maps?hl=en>
- [5] Google OR-Tools. <https://developers.google.com/optimization?hl=en>
- [6] OpenStreetMap. <https://www.openstreetmap.org/>
- [7] LightGBM. <https://lightgbm.readthedocs.io/en/stable/>
- [8] Keras. <https://keras.io/>
- [9] Y. Dumas, J. Desrosiers, E. Gelinas, M. M. Solomon, An optimal algorithm for the traveling salesman problem with time windows, Operations Research 43 (2) (1995) 367-371.
- [10] 坂上知英, 吉澤慎, 太田義勝, 大山口通夫, 巡回セールスマン問題の近似アルゴリズムについて, vol.25, pp.81-96(2000)
- [11] Abdoun Otman, Jaafar Abouchabaka, A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem, International Journal of Computer Applications 31(11):49-57(2011)
- [12] 李雅婕, 時間枠付き巡回回路問題に対する実験的解析 (東京海洋大学 2022-08-12)
- [13] 橋本英樹, 胡艶楠, 配送計画問題に対する発見的解法 (Jstage Vol. 64, No. 6, pp. 218-223, 2020)