

アニーリングマシンを用いたライドシェアアプリケーションの概念実証 —ユーザー体験を損なわない逐次的な全体最適化—

1. 背景

アニーリングマシンは、組合せ最適化問題を高精度、高効率に解くことが期待されているマシンである。組合せ最適化問題は、様々な問題が社会に内在することが知られている。一方で、アニーリングマシンの社会における認知度は低いという課題がある。この課題に対して我々は、よりユーザーにとって身近である消費者向けのアプリケーションにアニーリングマシンを用いることを考えた。その中で選んだ題材として、タクシーやライドシェアにおけるユーザーと車両のマッチング問題である。

タクシーやライドシェアにおけるマッチング問題とは、現在地からある目的地へ移動したいユーザーとそれを実際に運ぶ車両とをマッチングする問題である。この問題は、単にユーザーに最も近いタクシーを割り当てる場合には局所最適化が起き、他のユーザーも含めた配車までの時間が長くなるという課題がある。また、ユーザーの嗜好に合わせた車両や、ドライバーのキャンセル率等を考慮したマッチングを行う必要があるため、困難な問題と知られている。加えて、この問題は社会課題に密接に関連した問題である。

日本では、タクシー不足が社会課題として挙げられている。これは、過疎地域をはじめとして、高齢者の移動の不足に直結する。このような課題への対策として、「日本版ライドシェア」や「公共版ライドシェア」といったライドシェアの規制緩和が進んでいる。さらに、既存のタクシーを利用した相乗りを可能にするアプリなどがリリースされている。

2. 目的

本プロジェクトでは、アニーリングマシンを用いたライドシェアアプリケーションの開発を行う。具体的には、オンデマンド型と乗合型のライドシェアについて、ユーザー体験を損なわないマッチングアルゴリズムの開発と、それを用いたアプリケーション開発、アルゴリズムのベンチマーク手法の3点を行う。これにより、ライドシェアのマッチング問題において、アニーリングマシンを実践的に使用する手法の開発を行い、アニーリングマシンの適用範囲を拡大することができると期待される。また、ユーザーにとってより身近な消費者アプリケーションにおいて、アニーリングマシンを使用する先駆けとなると考えられる。

3. ソフトウェア開発内容

本ソフトウェアは、アニーリングマシンを用いて、タクシーやライドシェアにおけるユーザーと車両の配車マッチング問題を解決することを目的としている。開発したアプリケーションは、フロントエンド、バックエンド、アルゴリズムサービスの3層からなるアーキテクチャを採用した。開発したアプリケーションのアーキテクチャと使用した技術スタックを図1に示す。このアーキテクチャの特徴は、ユーザーのリク

エストに対する非同期処理とバッチ処理の導入、およびタスクスケジューラによるアニーリングマシンとの通信である。ユーザーとサーバー間に WebSocket を用いた双方向通信を導入することで、リクエストに対する非同期的な処理を実現した。また、Nest.js の Scheduled Task を用いて 5 秒間隔で配車リクエストとタクシー情報を自動的に集約（バッチ化）し、これを Task Scheduler (Celery) を通じてアニーリングマシンへ並列的かつ分散的に送信・求解させる構成となっている。これにより、ユーザーのリクエストに対して逐次的な全体最適化を実現した。

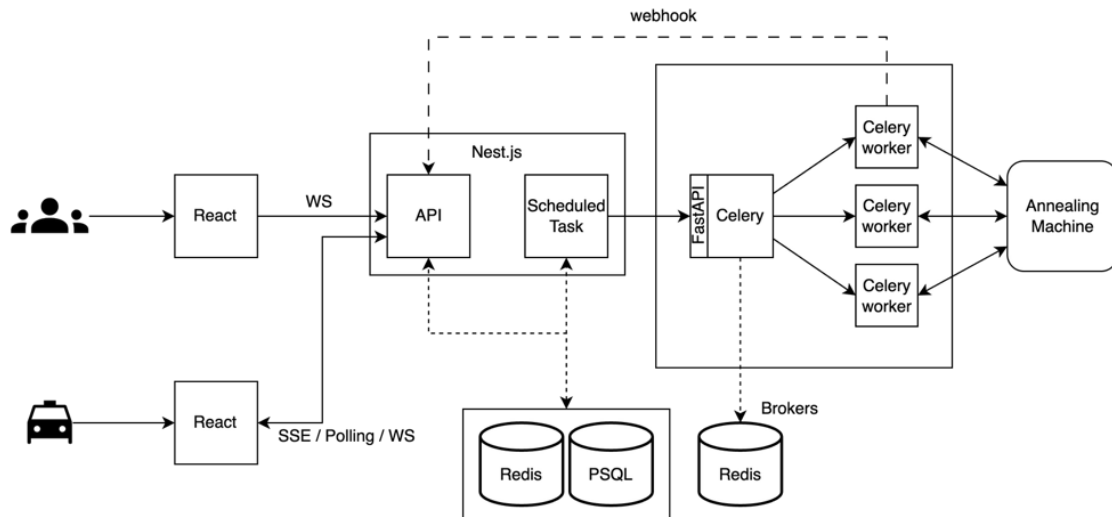


図1 アプリケーションのアーキテクチャ図と使用した技術スタック

フロントエンドでは React を使用し、ユーザーがマップ上で乗車地と目的地を指定してオンデマンド型および乗合型の配車リクエストを行える UI を実装した。バックエンドでは Nest.js を採用し、Prisma と PostgreSQL を用いてユーザーやタクシー、乗降経路などの状態管理を行うとともに、WebSocket ゲートウェイを構築することでフロントエンドとの双方向通信を可能にしている。アルゴリズムサービスでは、FastAPI と Celery を用いた非同期 API サーバーを実装し、アニーリングマシンには Fixstars Amplify 社の Annealing Engine (Amplify AE) を利用した。

本ソフトウェアは、主に以下の3つの機能を備えている。1つ目は、配車マッチング機能（オンデマンド型・乗合型）である。利用客に1台のタクシーを割り当てる「オンデマンド型」と、最大2組までの相乗りを許容する「乗合型」の2つの方式に対応している。実際にオンデマンド型・乗合型ライドシェアを使用する際の画面の遷移の一部を図2, 3に示す。本機能の実装にあたり、各ライドシェアのマッチングを組合せ最適化問題として定義し、QUBO 定式化を行うことでアニーリングマシンによって解ける形式で表現した。2つ目は、ユーザー向けのデモアプリケーション機能である。地図上で乗車地点・目的地・人数・ライドシェアの方式（ライドタイプ）を指定可能とし、マッチング完了後には配車されたタクシーの位置情報や、乗合モード時の他ユーザーの経由地を含む乗降経路を表示する機能を実装した。3つ目は、ベンチマー

ク用動的シミュレータである。システム全体の動作テストおよびアルゴリズム評価のため、実際の道路ネットワーク (OpenStreetMap) 上で複数台の仮想タクシーを自律走行させ、位置情報をリアルタイム送信するタクシーシミュレーション機能を実装した。あわせて、指定レートに従って配車リクエストを動的に生成しデータベースへ保存する、ユーザー配車リクエストのシミュレーション機能も開発した。

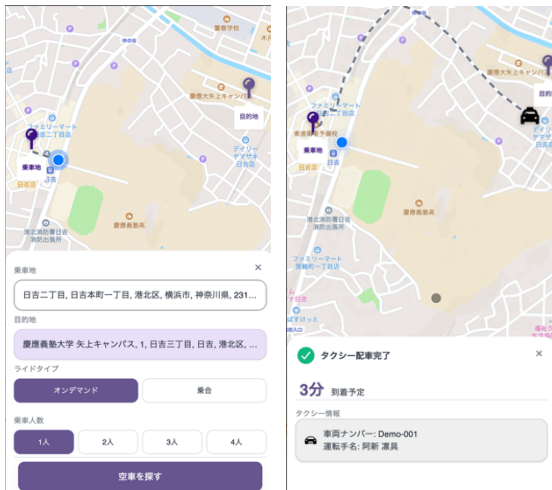


図 2 オンデマンド型ライドシェアを使用する際のアプリケーションの遷移。

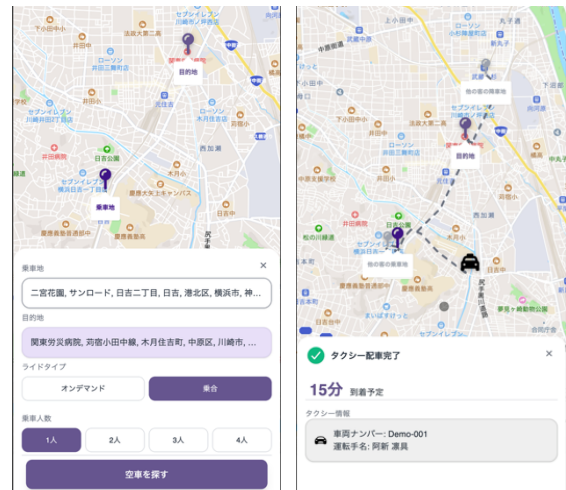


図 3 乗合型ライドシェアを使用する際のアプリケーションの遷移。

4. 新規性・優位性

本プロジェクトの新規性は、従来は静的な問題設定で用いられることが多かったアニーリングマシンを、逐次的に配車リクエストが発生するライドシェア環境へ実装し、リアルタイム性を考慮した最適化アーキテクチャとして構築した点にある。非同期処理、バッチ化、タスクスケジューラを組み合わせることで、動的に変化するユーザーおよび車両の状態を扱いながら最適化計算を実行可能とし、消費者向けアプリケーションに量子アニーリングを統合する実用的な枠組みを提示した。

また、本手法の優位性として、単純な古典的割当手法と比較したベンチマークにおいて、マッチングに要する時間の増加を数秒程度に抑えつつ、タクシーの総走行距離を最大 26.4%削減できることを示した点が挙げられる。さらに、QUBO 定式化に基づく設計により、ユーザーの嗜好などの追加制約を柔軟に組み込むことが可能である。また、ライドシェアに限らず、逐次的な最適化を必要とする他分野への応用可能性も有している。

5. 期待されるユーザー価値と社会へのインパクト

本プロジェクトでは、逐次的に送信されるリクエストに対してアニーリングマシンで最適化を行う仕組みを提案し、従来の「一度だけ最適化する」用途とは異なる新しい活用方法を示した。特に、時間とともに変化する乗客やタクシーの位置を扱うライドシェアの実環境を想定し、既存研究とのギャップを埋めている。本成果はライド

シェアに限らず，連続的な最適化が必要なアプリケーションにも応用可能である．また，既存手法を置き換えること自体を目的とはせず，ユーザー体験を損なわない範囲で，全体最適による走行距離の短縮を実現できることを示した．

6. 氏名（所属）

井出 駿太（慶應義塾大学理工学部物理情報工学科）

福原 博樹（慶應義塾大学理工学部物理情報工学科）