Coordinated Route Planning

Atharva Suhas Mulay
B.Tech. Computer Science
Pre-final year
IIT Ropar
2021csb1076@iitrpr.ac.in

Jayrajsinh Chavda
B.Tech Computer Science
Pre-final year
IIT Ropar
2021csb1078@iitrpr.ac.in

Nishad Dhuri
B.Tech Computer Science
Pre-final year
IIT Ropar
2021csb1116@iitrpr.ac.in

***Abstract*—This paper presents a novel algorithm for coordinating large-scale routes, leveraging graph attention reinforcement learning and Monte Carlo Tree Search (MCTS). The road network is partitioned into regions, treating each as a player in a Markov game. A bilevel optimization framework is employed, with region planners coordinating route choices for vehicles and a global planner evaluating the generated strategies. The algorithm aims to balance user fairness and system efficiency, address dimensionality challenges, and enhance route planning by simulating traffic dynamics. Experimental results on real-world road networks demonstrate the algorithm's effectiveness in minimizing travel time and efficiently managing large-scale routes, contributing to congestion alleviation**

I. Introduction

With the advent of AI autonomous driving vehicles, there's a possibility that most cars will be driven by AI very soon. With the urbanisation, we see that traffic control has become a serious issue for the urban planners. Traffic congestion takes away the precious time of millions of people who live in cities. With the advent of autonomous driving vehicles and, in general, AI Maps technology, we have the opportunity to plan special paths that try to reduce traffic congestion. Recently, drivers are increasingly leveraging real-time information through GPS devices and online routing tools (e.g., Google maps) to move faster using vehicular networks. This trend offers a new tool to guide drivers to make choices for the benefit of the city, thus creating a more optimal traffic configuration. Consequently, coordinated route planning (CRP) stands out as a promising and popular way to overcome the traffic congestion problem, i.e. considering the whole system welfare and assigning routes to users in such a way that the congestion problem is solved or, at least kept at a minimum level. The problem is a multiagent path planning problem with very high dimensionality, and no promising solution is known yet. In recent years, interest in this domain has been revived. There are some general solutions to the problem, like M* search. There is current research being done using Nash-equilibrium methods to generate plans to reduce traffic congestion. Some of the current research uses Deep Learning techniques to analyse Traffic Congestion from satellite images.

II. Problem Definition

The problem can be formally defined as follows:

Given a road network, denoted as a connected and directed weighted graph $G = (V, E)$, where V represents the set of road intersections, E denotes the set of road segments, $W : E \rightarrow \mathbb{R}$ defines the edge weights indicating distances between intersections, $T : E \rightarrow \mathbb{R}$ signifies static traffic on each road segment, and $N(S, D)$ represents a set of drivers with respective source (S_i) and destination (D_i) information, the objective is to formulate a coordinated route planning algorithm. This algorithm seeks to yield optimal routes for all drivers, minimizing traffic congestion while accounting for traffic conditions and minimizing the overall travel time.

The Coordinated Route Planning (CRP) problem is computationally intricate and exhibits high stochasticity. Consequently, strategies that systematically consider all potential routes for each driver and leverage historical patterns for predicting traffic congestion are often employed to derive approximate solutions. The problem can be modeled as a multiagent path planning problem.

III. Literature Review

In this section, we review existing methodologies, particularly focusing on the **Multi-Agent Path Planning (MAPP)** problem, to provide a foundation for the method proposed in this paper. The primary challenge addressed by these methodologies is the optimization of route planning for multiple agents operating in a shared environment. While most MAPP approaches assume *hard collisions*, prohibiting simultaneous use of the same resource, the **SC-M*** algorithm introduces the concept of *soft collisions*, allowing agents to share resources at the expense of a reduced solution quality.

SC-M Algorithm:*

SC-M* (*Soft-Collision M**) is an extension of the **M*** MAPP algorithm. Unlike traditional approaches, **SC-M*** acknowledges the possibility of *soft collisions*, where agents can coexist at the same location, albeit with a penalty. This aligns with the problem addressed in our paper, where some traffic is permitted on every road with associated penalties. **SC-M*** tracks collision scores for each agent, placing those exceeding defined thresholds into a *soft-collision set* for sub-dimensional expansion. This modification enhances scalability for handling a larger number of agents while maintaining bounded collision probabilities.

Challenges in Previous Methodologies:

1) **Gap between User Equilibrium (UE) and System Optimum (SO):** There exists a challenge in balancing

fairness (*UE*) and efficiency (*SO*) in the context of Cooperative Route Planning (**CRP**). The User Equilibrium ensures selfish route selection, but it does not minimize total travel time. On the other hand, the System Optimum prioritizes efficiency but is unstable and unfair. Bridging the gap between *UE* and *SO*, often referred to as the "price of anarchy," remains a practical challenge in large-scale **CRP** systems.

2) **Complexity in Coordinating Massive Routes:** Coordinating routes for a large number of drivers poses a computational challenge. The number of potential route combinations grows exponentially with the number of drivers and potential routes. Pruning techniques are applied to reduce the search space. For instance, *Li et al.* (2021) implemented a depth-first search with pruning for early termination, while *Chen et al.* (2020) developed an approximate matching algorithm with local search and pruning techniques. However, the complexity remains a concern given the scale of hundreds of thousands of drivers in a city.
3) **Dynamic Impact on Successive Traffic Situations:** Addressing the spatio-temporal dynamics of transportation networks is crucial. Existing approaches rely on historical data to model and predict congestion, but this is not conducive to real-time optimization. Achieving real-time optimization in the face of evolving congestion patterns poses a significant challenge in **CRP** systems.

In summary, the review highlights the advancements in **MAPP** algorithms, with a specific focus on the **SC-M*** algorithm, and outlines the challenges faced by previous methodologies in Cooperative Route Planning, setting the stage for the proposed method in this paper.

IV. Bi-level Optimisation Framework

We first partition the road networks into M regions and propose a bilevel optimization framework, consisting of M region planners and a global planner. The region planner considers user equilibrium and coordinates the route choices for vehicles within the region. The global planner is built on a value function that adopts graph attention networks for evaluating each state (a combination of strategies) and Monte Carlo Tree Search (MCTS), which can dynamically search the state space and simulate the traffic for sufficiently long steps.

Fundamental Concepts

Fundamental concepts used in CRP Algorithm are briefly described as follows:

1) **Road Network Partition:** The road network $G(V, E)$ is spatially partitioned into M parts/regions, where $G = G_1 \cup G_2 \cup \ldots \cup G_M$, and each $G_i = (V_i, E_i)$. $V = V_1 \cup V_2 \cup \ldots \cup V_M$, $E = E_1 \cup E_2 \cup \ldots \cup E_M$, and $V_i \cap V_j = \emptyset$ if $i \neq j$.
2) **Regional Planner:** For each region m, a region planner coordinates the route choices of vehicles within this region N_m, eventually outputting several possible strategies, where each strategy contains the route choices of all vehicles in N_m. This planner takes user fairness into consideration.
3) **Global Planner:** This planner balances and evaluates the inter-region influence and exploits simulations over the future to aid decision-making. Since each region has several possible strategies, this planner evaluates the combination strategies of all regions. However, the number of combination states is exponential with the number of regions and depth of simulations. Therefore, we exploit the Monte Carlo Tree Search algorithm to achieve higher efficiency.

Our proposed approach for CRP integrates regional and global planning, leveraging the strengths of both components to provide an efficient solution to the challenges inherent in large-scale cooperative route planning.

Challenges and Solutions

1) **Bridging Gap Between User Equilibrium (UE) and System Optimum (SO):** Exploiting the region planner to make coordinated traffic assignments in each region can help achieve user equilibrium. Then, a global planner can evaluate the combinations of strategies obtained by the regions, thus optimizing the total travel time. Consequently, congestion problems are solved or, at least, kept at a minimum level.
2) **Alleviating the Curse of Dimensionality:** Our proposed approach reformulates CRP as a modified Markov game, significantly reducing the state-action space since each region planner is treated as a player. A region planner generates several strategies under different assignment objectives, where each strategy forms the action space for that region. Consequently, the action and state space are significantly decreased.
3) **Simulation Over the Future to Capture Traffic Dynamics:** It can simulate and evaluate the future traffic for sufficiently long steps and apply the simulation results of several steps to make an informed and effective decision. It exploits the MCTS algorithm to dynamically evaluate states for narrowing down the search space.

A. Modified Markov Game

On our road network G, we have N drivers whose routes we need to coordinate. We model this problem as a Markov game where each region is treated as a player, represented by a tuple $(S, M, \{A_1, A_2, \ldots, A_M\}, F, \{r_1, r_2, \ldots, r_M\})$. M is the set of all regions. S is the status of the road network and all the drivers. $\{A_1, A_2, \ldots, A_M\}$ is the joint action space representing all possible actions for each region at the state S, where $A_i = \bigcup_{i \in N_m} P_i$, P_i being all possible actions for a driver in a region. F is the transition probability of going from the current state s_t to s_{t+1} after some action $\{a_1, a_2, \ldots, a_m\}$, where a_i are the particular actions taken by all the drivers in region i. r_i is the reward for each region, where $r_i = q_1 \cup q_2 \cup \ldots \cup q_N$, where q_i is the reward obtained by a driver in the region after taking some action.

Algorithm 1 Modified Markov Game

```
Require:
  Road network graph $G = (V, E)$
  Number of drivers $N$
  Set of regions $M$
  Initial state $S$ of road network and drivers
  Joint action space $A_i$ for each region $i$
  Transition probability function $F$
  Reward function $r_i$ for each region $i$
Ensure:
  Optimal joint actions for cooperative route planning
  Procedure Markov Game
  Initialization:
  Define the road network graph $G = (V, E)$
  Set the number of drivers $N$
  Define the set of regions $M$
  Set initial state $S$ of road network and drivers
  Define joint action space $A_i$ for each region $i$
  Set transition probability function $F$
  Define the reward function $r_i$ for each region $i$
  repeat
    for each time step $t$ do
      for each region $i$ in $M$ do
        Choose action $a_i$ from $A_i$ based on some strategy or policy
      end for
      Update the state based on the chosen actions:
      Apply the transition probability function $F$
      Calculate rewards for each driver and update the global reward:
      for each region $i$ in $M$ do
        Calculate $q_i$ for each driver based on the chosen actions in region $i$
        Update the global reward $r_i = q_1 \cup q_2 \cup \ldots \cup q_N$
      end for
      Update the state of the road network and drivers
    end for
  until convergence or a predefined number of iterations =0
```

V. Methodology

The methodology articulated in the proposed model involves a series of procedural steps. Initially, the road network $G(V, E)$ undergoes partitioning into multiple regions through the application of established graph partitioning algorithms. Subsequently, a regional planner, tasked with coordinating the route choices of vehicles within each designated region N_m, and a global planner, responsible for balancing and assessing inter-region influences, are devised. The subsequent sections of our proposed model expound upon the intricacies of designing the regional and global planners.

A. Proposed Model

Regional planner: The local planner m aims to coordinate the route choices of vehicles N_m, which can be described as a game $(G_m, N_m, (P_i)_{i \in N_m}, (g_i)_{i \in N_m})$. g_i is the payoff (cost) function. The strategy space P_i of a player $i \in N_m$ is the set of all simple paths in G from o_{0i} to d_{0i}, where o_{0i} and d_{0i} are the mapped locations of o_i and d_i on G_m, respectively. Denote a strategy profile as $P = (p_1, p_2, \ldots, p_{|N_m|})$, where $p_i \in P_i$. The payoff function can be set by considering different objectives and is introduced as follows:

- Objective A: minimum free-flow travel latency
- Objective B: minimum travel distance
- Objective C: minimum number of traffic lights
- Objective D: travel time with mutual influence

For objectives A, B, and C, the cost function $g_i(P)$ only correlates with p_i. $\min_{P_i \in N_m} g_i(P)$ can also be rewritten as $\min_{p_i \in P_i} g_i(p_i)$. For player i, it chooses p_i which minimizes the cost function, i.e., $p_i = \arg\min_{p_i \in P_i} g_i(p_i)$. If there exists another strategy p_{0i} that could let i spend a lower cost, then $g_i(p_{0i}) < g_i(p_i)$, which means that p_i is not minimum and is contradictory.

Similarly, for objective D, the cost function is the travel time which depends on the number of vehicles using the road segments, and for such a game, existence and uniqueness of the user equilibrium have been proved. **Therefore, the game has a unique user equilibrium for each defined objective.**

Global planner: We use Monte Carlo Tree Search(reinforcement learning model) on our super nodes to coordinate the path planning.

The region planner $m \in M$ applies different objectives for coordinating the route choices of vehicles N_m, eventually obtaining several actions, where each action is a user equilibrium under an objective. The possibility of each action is difficult to define. Therefore, we ignore the prior probability of action and only consider the value function for each state in the lookahead search. So, in our MDP, transition probabilities are assumed to be uniformly distributed.

The Value of each state is defined as the summation of $V(s_t)$ (exploitation factor or goodness factor) and $u(s_t)$ (exploration factor).

$$a_t = \text{argmax}_a(V(s_{t+1}) + u(s_{t+1}))$$

Each action in MCTS refers to one step in our plan for all active drivers. Action state contains augmented actions of each region.

The MCTS algorithm centers around iterative manipulation of a decision tree, happening in four phases: selection, expansion, simulation, and backpropagation. At a high level, MCTS repeatedly samples episodes from this generative model to approximate the expected reward down the various paths recorded in the decision tree.

Selection: Selection begins at the root of the search tree and finishes when the simulation reaches a leaf node. Before the leaf node, an action is selected according to the statistics in the search tree,

$$a = \text{argmax}(V(s) + u(s))$$

level of exploration. This search control strategy initially prefers states with equal probability and low visit count but asymptotically prefers states with high value.

$$u(s_t) = \sqrt{2 \cdot \log(N(s_t)) / \sum N(s_{t+1})}$$

Expansion: When the visit count $N(s)$ of a state s exceeds a threshold n, i.e., $N(s) > n$, state s will be expanded. We fully expand the leaf node s_L to s_{L+1}, i.e., iterate over all possible combinations of actions generated by each region planner.

Evaluation: After expansion, we rollout to the terminal state based on some policy based on heuristics. Rollout begins with the leaf state, all drivers select from the first three neighbors with the shortest distance to the destination with some probability for each of three. The rollout continues until the end of the game (all drivers have reached their destination). When the game reaches a terminal state, the reward is inversely related to the depth of the tree (distance traveled) and average traffic value in all states.

Backup: At the end of the simulation, the rollout statistics are updated in a backward pass through the leaf state to the root state. For each state that was visited in the selection process, we update

$$R = A \cdot \frac{1}{\sqrt{\text{depth}}} + B \cdot \text{Average Traffic Quality in rollout}$$

where A and B are hyperparameters that need to be changed for fine-tuning.

The Average Traffic Quality in rollout is calculated as,

$$\text{Average Traffic Quality in rollout} = \frac{\sum T(s_t)}{\text{depth of rollout}}$$

where,

$$T(s_t) = \frac{1}{\text{var(drivers across road network)}}$$

.

We add the reward value R up the tree from the leaf to the root and update

$$N(s_t) + = N(s_t) + 1$$

.

Algorithm 2 Monte Carlo Tree Search

```
procedure MONTECARLOTREESEARCH
    $\tau_0 \leftarrow$ MakeTree($x_0$) {Initialize decision tree with root state $x_0$}
    repeat{Main, iterative loop of the algorithm}
        $\tau_{i,r} \leftarrow$ TreePolicy($\tau_0$) {Exploration, returning new leaf and current reward}
        $x_i \leftarrow \tau_i$.state
        $r \leftarrow$ DefaultPolicy($x_i$) {Simulation, completing episode and returning total reward}
        Backpropagate($\tau_i, r$) {Update node statistics along the exploration path}
    Timeout()
    return BestAction($\tau_0$) {Pick the approximately optimal root-level action}

procedure TREEPOLICY($\tau$) {Decision policy for exploration}
    $r \leftarrow 0$
    $x \leftarrow \tau$.state
    while Nonterminal($x$) do
        $a \leftarrow$ BestAction(x) {Heuristically select an action from $A$}
        $x' \leftarrow$ Transition$P(x, a)$ {Sample transition from the generative model}
        $r \leftarrow r + R(x, a, x')$
        if $\tau$.children[$a$][$x'$] = null then
            $\tau$.children[$a$][$x'$] $\leftarrow$ MakeTree($x'$) {Initialize a leaf node for state $x'$}
                return $\tau$.children[$a$][$x'$], $r$ {Move on to the simulation phase}
        end if
        $\tau \leftarrow \tau$.children[$a$][$x'$]
        $x \leftarrow x'$
    end while
    return $\tau, r$
end procedure
procedure DEFAULTPOLICY($x, r$) {Decision policy for simulation}
    while Nonterminal($x$) do
        $a \leftarrow$ RandomAction(x) {Randomly select an action from $A$}
        $x' \leftarrow$ TransitionP($x, a$) {Sample transition from the generative model of $P$}
        $r \leftarrow r + R(x, a, x')$
    end while
    return $r$
end procedure
procedure BACKPROPAGATE($\tau, r$) {Update statistics along the path to this tree node}
    repeat
        $\tau$.reward $\leftarrow \tau$.reward $+ r$
        $\tau$.count $\leftarrow \tau$.count $+ 1$
        $\tau \leftarrow \tau$.parent
    until $\tau$ = null
    return
end procedure
procedure BESTACTION($\tau$) {Pick the approximately best action at this tree node}
    return $\arg\max_{a \in A} \left( \frac{\sum_{\tau' \in \tau.\text{children}[a][\cdot]} \tau'.\text{reward}}{\sum_{\tau' \in \tau.\text{children}[a][\cdot]} \tau'.\text{count}} \right)$
end procedure=0
```

VI. Future Work

we have used statistics method to extract the traffic congestion information from the road network of a given state. We can use Graph Attention Network, A deep learning model used to analyse graph. Graph Attention Network can be trained to extract the measure of traffic congestion. This will provide better reward values.

VII. Conclusions

The present study introduces an algorithm for the coordination of extensive route networks, utilizing a value function and the Monte Carlo tree search algorithm. This algorithm for CRP demonstrates notable efficiency and effectiveness in large-scale route coordination, exhibiting the ability to: 1) reconcile user fairness with system efficiency, 2) attain heightened search efficiency by mitigating the curse of dimensionality, and 3) facilitate proficient and informed route planning through future traffic dynamics simulation. This paper contributes a viable approach to address the Coordinated Route Planning (CRP) problem, emphasizing the significance of integrating user-centric and system-oriented considerations for enhanced large-scale route coordination.

References

[1] Chen, L.; Shang, S.; Yao, B.; and Li, J. 2020. Pay your trip for traffic congestion: Dynamic pricing in traffic-aware road networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 582–589

[2] Belov, A.; Mattas, K.; Makridis, M.; Menendez, M.; and Ciuffo, B. 2021. A microsimulation based analysis of the price of anarchy in traffic routing: The enhanced Braess network case. Journal of Intelligent Transportation Systems, 1–16.

[3] C¸ olak, S.; Lima, A.; and Gonzalez, M. C. 2016. Understand- ´ ing congested travel in urban areas. Nature communications, 7(1): 1–8.

[4] Colini-Baldeschi, R.; Cominetti, R.; Mertikopoulos, P.; and Scarsini, M. 2020. When is selfish routing bad? The price of anarchy in light and heavy traffic. Operations Research, 68(2): 411–434.

[5] Delarue, A.; Anderson, R.; and Tjandraatmadja, C. 2020. Reinforcement learning with combinatorial actions: An application to vehicle routing. Advances in Neural Information Processing Systems, 33: 609–620

[6] Frank, M. 1981. The braess paradox. Mathematical Programming, 20(1): 283–302.

[7] Fu, Z.-H.; Qiu, K.-B.; and Zha, H. 2021. Generalize a small pre-trained model to arbitrarily large tsp instances. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 7474–7482.

[8] https://courses.cs.washington.edu/courses/cse599i/18wi/resources/lecture19/lecture19.pdf

[9] https://www.mdpi.com/2076-3417/9/19/4037