Pareto-Optimal Transit Route Planning with Multi-Objective Monte-Carlo Tree Search

Di Weng, Ran Chen, Jianhui Zhang, Jie Bao, Yu Zheng, Yingcai Wu

***Abstract*—Planning ideal transit routes in the complex urban environment can improve the performance and efficiency of public transportation systems effectively. However, finding such routes is computationally difficult due to the huge solution space constituted by billions of possible routes. Considering the limited scalability of exact search methods, heuristic search methods were proposed to boost the efficiency and incorporate flexible constraints. Nevertheless, the existing methods conceal multiple criteria in an objective, and thus evaluating the performance of the generated route becomes challenging due to the lack of comparable alternatives. Inspired by the prior study, we formulate the definition of pareto-optimal transit routes based on multiple criteria. However, extracting these routes remains challenging because: A) the sheer volume of possible transit routes; and B) the sparsity of pareto-optimal routes. We address these challenges by developing an efficient search framework: for challenge A, a random search method is developed based on Monte Carlo tree search where the unproductive solution subspaces are pruned progressively to reduce the search cost; and for challenge B, an estimation method is derived to guide the search process by assessing the value for each solution subspace. The superior effectiveness of our approach in approximating the pareto-optimal transit routes was demonstrated by the comprehensive evaluation based on the real-world data.**

***Index Terms*—Transit Route Planning, Heuristic Search**

I. Introduction

PUBLIC transit networks (e.g., buses and rapid transit lines) are developed in many countries to alleviate the severe traffic pressure and air pollution problems caused by the increasing number of automobiles [1]. Most of the transit routes in these networks, each of which comprises a series of stations, are typically updated every 3–5 years to match the shifting travel demand [2]. Nevertheless, finding cost-effective transit routes remains a challenging task because 1) searching for an optimal solution will require examining a huge solution space that consists of massive combinations of stations; and 2) various decision factors need to be integrated into the search process, such as the service length of transit routes, the number of intermediate stops, and the estimated travel demands satisfied by the proposed routes [3].

This problem is known as the classic Transit Network Design Problem (TNDP) [1]. Many methods have been proposed to facilitate the generation of transit routes. Traditional approaches [4], [5] employ optimization and exact search methods based on mathematical models, such as linear planning, to obtain optimal transit routes under the given constraints, but these approaches do not scale well with the number of stations in large cities. To improve the scalability and incorporate flexible constraints, recent approaches [6], [7] adopt heuristic search methods (e.g., simulated annealing and genetic algorithms) to find a feasible solution approximately in reasonable time for large cities. However, although TNDP is a multi-objective optimization problem, most of these approaches attempt to obtain solutions with a summary objective function. For example, Fan et al. [7] balances the user costs, the operator costs, and the travel demands of the generated routes with a weighted sum approach in their objective function. It is not only challenging to find a reasonable method to combine various objectives and adapt to new ones flexibly, but also difficult for transportation experts to ascertain the value of the generated routes due to the lack of comparable alternatives.

Inspired by the skyline operator [8], the idea of *skyline routes* [9] was proposed to search for pareto-optimal transit routes directly with multiple objectives. Chen et al. developed a greedy search algorithm called *probability-based spreading* (PBS), which was applied to search for possible routes between a pair of given origin and destination stations and generate a pareto-set of skyline routes. This pareto-set comprises the routes that are not outperformed by any other route in the same set with respect to two criteria, total service time and demand satisfaction. The skyline route sets enable transportation experts to evaluate the performance criteria of the promising routes directly and perform informed tradeoffs among these alternatives based on their expertise. However, limitations are observed in the prior study where A) the definition of skyline routes and the PBS algorithm are tailored specifically for two aforementioned criteria, and B) the performance of the PBS algorithm is limited to producing poor pareto fronts frequently due to its greedy characteristic.

These limitations motivate us to extend the aforementioned skyline route definition to include multiple criteria and propose a refined search framework for approximating the pareto-optimal transit routes efficiently. In particular, two major challenges were identified in the development of such framework:

Volume of transit routes. Finding a pareto-set of transit routes is a difficult combinatorial optimization problem. In a large city, numerous routes could be feasible between a given pair of origin and destination stations because of the sheer volume of potential intermediate stations. However, the PBS algorithm simply repeats greedy stochastic search without considering the structure of the solution space, thereby rendering the acquisition of the optimal pareto-sets almost

D. Weng, R. Chen, Y. Wu were with State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. E-mail: {dweng, crcrcry, ycwu}@zju.edu.cn.

J. Zhang was with Hong Kong University of Science and Technology, Hongkong, China. Email: istarzjh@gmail.com.

J. Bao, Y. Zheng were with Urban Computing Business Unit, JD Finance, Beijing, China. Email: jiebao1985@gmail.com, msyuzheng@outlook.com.

Manuscript received April 19, 2005; revised August 26, 2015.

impossible. Hence, the proposed search framework must be highly efficient in scanning and extracting feasible pareto-optimal routes, and ideally the resulted approximation should be gradually improved with more computational resources.

Sparsity of pareto-optimal routes. Pareto-optimal routes are particularly difficult to find because the number of these routes are much fewer compared to that of all feasible routes. The PBS algorithm adopts a simple search strategy that greedily selects the subsequent stations in a candidate route by maximizing passenger flows, which neither generates satisfactory pareto-optimal routes nor generalizes well for multi-criteria scenarios. Without effective heuristics to narrow down possible pareto-optimal routes, the search process may always return an inferior pareto-set. Hence, such heuristics must be developed and integrated in the proposed search method in order to find the pareto-optimal routes effectively.

To address the aforementioned challenges, this study proposes a novel random search method on the basis of the Monte-Carlo tree search (MCTS) framework [10]. For the first challenge, we propose a fast random search process to locate pareto-optimal routes efficiently and prune the unproductive route subspaces progressively to reduce the search cost. For the second challenge, we introduce a new estimation method that precomputes the average criterion gains for each search subspace on the station graph and guides the subsequent search process heuristically in selecting the most promising subspaces. By tightly integrating subspace estimation with random search, our method can extract significantly better pareto-optimal routes more efficiently compared with the prior method. Our contributions are summarized as follows.

- We propose a heuristic method for estimating the average gain of each route subspace on the station graph;
- We derive a novel random search method for approximating the pareto-optimal route sets efficiently;
- We evaluate our approach on the real world data compared with the state-of-the-art baseline.

II. Related Work

This section summarizes the relevant studies in two aspects, namely, transit network design and Monte-Carlo tree search.

Transit network design problem (TNDP) is one of the most classic problem in the transportation research area. A systematic survey on this topic has been given in [1]. In addition, the survey of the data-driven approaches is presented in [11]. Generally, the data-driven planning of transit networks is achieved in two steps. First, travel demand in the planning area is determined from surveys [12], traffic data [9], communication data [13], etc., and a number of candidate stations are extracted from the travel demand based on various clustering methods, such as grid-based clustering [9], DBSCAN [14], and CFSFDP [15]. Then, mathematical (e.g., linear programming [4], [5]) or heuristic (e.g., random search [9], tabu search [16], genetic algorithm [7], [17], [18], simulated annealing [19], and ant colony algorithm [12]) methods are applied to find feasible transit routes based on the extracted stations. Among these studies, some methods [7], [17] further incorporate the settings of bus route frequencies. Public transit routing [20], which attempts to find a set of pareto-optimal travelling routes between two locations, is another problem similar to TNDP. However, the public transit routing problem is merely a specialized case of transit route planning, which involves more complex criteria computed in pairwise, such as the travel demand. In order to assist domain experts in iteratively assessing the performance of similar routes and determine the best one while planning a transit route, a fast and effective approach for extracting alternative routes is yet to be proposed. Inspired by [9], our study focuses on generating a set of interchangeable pareto-optimal routes based on flexible performance criteria between a given pair of origin and destination stations.

Monte-Carlo tree search (MCTS) is a random search framework which has demonstrated its effectiveness in various application domains, such as games [21], [22], allocation [23], chemistry [24], etc. By repeating four search stages, namely, selection, expansion, simulation, and backpropagation, MCTS can efficiently explore the huge solution space and decide an optimal action to take at the current state. A extensive survey of MCTS methods was conducted by Browne et al. [10]. Powley et al. [25] and Perez et al. [26] applied MCTS to physical travelling salesman problem, planning routes for agents in real-time to find the shortest paths. However, their methods did not consider multi-criteria scenarios, such as transit route planning. Furthermore, MCTS was also extended to handle multi-objective optimization problems [27], [28], where pareto-optimal policies were obtained with tailored UCB functions [29]. Inspired by these prior studies, we derived a new search method based on MCTS framework. Our method efficiently extracts pareto-optimal transit routes with respect to multiple performance criteria by integrating new heuristics and adapting to the exploration of station graphs. To the best of our knowledge, our method is the first study that applies MCTS in the multi-criteria transit route planning scenario.

III. Problem Statement

This section formalizes the definition of pareto-optimal transit routes and characterizes the route planning problem that will be solved in this paper.

We assume that a set of feasible stations $S = \{s_1, s_2, ...\}$, from which stops can be selected, has been either predefined or mined from the travel demand data [14]. Each station in S is identified by its spatial location. Given a pair of origin and destination stations $s_o, s_d \in S$, we denote a transit route between these two stations as $R = (r_1, r_2, ..., r_n) \in P_{od}$, where the first stop is the origin $r_1 = o$, the last stop is the destination $r_n = d$, and $s_{r_1}, s_{r_2}, ... \in S$. Specifically, we use the indices of stations in the definition of transit routes for better presentation clarity. All feasible routes between s_o and s_d constitute a *route space* P_{od}. We also assume that a set of criteria $C = \{c_1, c_2, ...\}$ has been defined for these routes to measure the performance of each route. For each route $R_i \in P_{od}$ and each performance criterion $c_k \in C$, a criterion value $c_k(R_i) \in Z_{c_k}$ can be obtained, where Z_{c_k} contains all possible values for c_k. A total order relation $\leq_{c_k}$ can be further defined on the set Z_{c_k} for each criterion c_k:

$c_k(R_j) \leq_{c_k} c_k(R_i)$ iff the route R_j is worse than ($<_c$) or equal to ($=_c$) R_i w.r.t. the criterion c_k. That is, R_j is dominated by R_i w.r.t c_k iff $c_k(R_j) <_{c_k} c_k(R_i)$. Finally, we define the *pareto-optimal transit route set* as follows.

Definition 1 ***(Pareto-optimal transit route set)*** *A transit route set $P \subseteq P_{od}$, given a pair of origin and destination stations s_o, s_d, is a pareto-optimal transit route set iff for every route $R \in P$, there does not exist any other route $R' \in P \setminus \{R\}$ that satisfies the dominance relationship: 1) $\forall c \in C : c(R) \leq_c c(R')$ and 2) $\exists c \in C : c(R) \neq_c c(R')$.*

We denote the routes in a pareto-optimal transit route set as the *pareto-optimal transit routes*. Many route subsets of P_{od} where each route is not dominated by any other route can be the pareto-optimal transit route sets. The set that comprises all these route subsets is denoted as $\Gamma_{od} = \{P_1, P_2, ...\}$, where $P_i \subseteq P_{od}$. To compare between two pareto-optimal transit route sets, we define a binary relation $d_\Gamma(P_i, P_j)$ as follows to obtain the *pareto-optimal transit route set difference* between P_i and P_j for every $(P_i, P_j) \in \Gamma_{od}^2$.

Definition 2 ***(Pareto-optimal transit route set difference)*** *Given two non-empty pareto-optimal transit route sets P_i and P_j, the unified pareto-optimal transit route set P' is obtained from $P_i \cup P_j$ by removing the dominated routes that conflict with Definition 1. Thereafter, the pareto-optimal transit route set difference between P_i and P_j is defined as $d_\Gamma(P_i, P_j) = (|P_i \cap P'| - |P_j \cap P'|)/|P'|$.*

For any $P_i, P_j \in \Gamma_{od}$, the pareto-optimal transit route set difference $d_\Gamma(P_i, P_j)$ falls in $[-1, 1]$, where $d_\Gamma(P_i, P_j) < 0$ indicates P_j is better; otherwise, P_i is better. It is not possible to exhaustively iterate over the solution space $\mathcal{P}(P_{od})$ (i.e., the power set of the route space P_{od}) to find the optimal solution P_m that $d_\Gamma(P_m, P_i) \geq 0$ for any route set P_i since $|\mathcal{P}(P_{od})|$ can be very huge. Therefore, our objective is to generate an approximation of P_m with random search, given feasible stations S, a pair of origin and destination stations s_o, s_d, and a set of performance criteria C. The idea of *skyline routes* introduced in the prior study [9] is a specialized case of the pareto-optimal transit routes with two criteria, namely, route service time and demand satisfaction.

IV. Methodology

This section presents our approach in generating the pareto-optimal transit route sets. First, for the *sparsity* challenge, we attempt to estimate the value of each route subspace on a prebuilt station graph to guide the subsequent search process in finding pareto-optimal transit routes effectively. Second, for the *volume* challenge, we further employ a tailored MCTS method based on the estimated values of route subspaces to extract pareto-optimal transit routes from the huge solution space efficiently.

A. Building Station Graphs

Inspired by the prior study [9], we first determine the feasibility of the routes between the stations with the station graphs. In addition, we also propose a novel method that estimates the average gains for choosing a route subspace, which comprises a subset of candidate transit routes, based on a classification of the given performance criteria to guide the subsequent random search process.

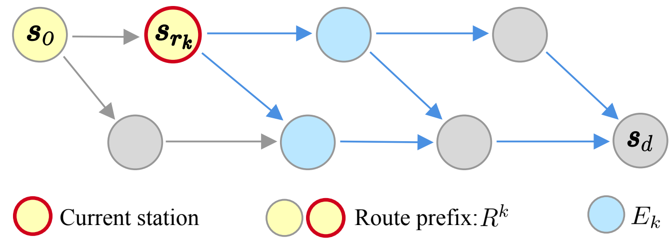


Fig. 1: The estimation of the average gain in each criterion is performed on the station graph. Blue directed edges indicate all possible routes between the current station s_{r_k} and the destination station s_d, which constitute the route subspace $P_{od}(R^k)$.

1) Basic Station Graph Construction: Given the origin and destination stations s_o, s_d, we follow the approach outlined in the prior study [9] to build a directed acyclic station graph $G_{od} = (S, E_{od})$, where the nodes comprise all potential stations in S, and each edge $(s_i, s_j) \in E_{od}$ indicates the feasibility to construct a transit route that passes s_i and s_j consecutively. Five intuitive criteria were leveraged to determine the feasibility of the transit line construction:

1) s_i must be at most δ meters away from s_j;
2) s_j must be farther from the origin s_o and closer to the destination s_d than s_i along the direction of $s_o \rightarrow s_d$;
3) s_j must be farther from the origin s_o than s_i;
4) s_j must be closer to the destination s_d than s_i;
5) s_i should be closer to s_j than any other station before s_i to avoid zigzags.

Criteria 1–4 can be easily determined during the graph construction, whereas Criterion 5 needs to be tested dynamically in the search process. In this study, we set $\delta = 3$km and adopt the road distances to generate the station graphs instead of the Euclidean distances to emulate a more realistic setting. After the graph is built, the nodes from which the destination cannot be reached are removed from the graph.

2) Route Subspace Value Estimation: Imagine a heuristic search process that produces a potential pareto-optimal transit route $R = (r_1, r_2, ..., r_n)$ by determining each station s_{r_i} passed by route R one by one starting from the origin station s_{r_1}. After first k stations $s_{r_1}, s_{r_2}, ..., s_{r_k}$ in route R, denoted by the route prefix $R^k = (r_1, r_2, ..., r_k)$, have been determined, all remaining feasible transit routes that begin with these k stations constitute a route subspace $P_{od}(R^k)$. The formal definition of the route subspace is presented as follows.

Definition 3 ***(Route subspace)*** *Given a route prefix $R^k = (r'_1, r'_2, ..., r'_k)$ of length k, the route subspace $P_{od}(R^k)$ is a subset of route space P_{od}, defined as $P_{od}(R^k) = \{R = (r_1, r_2, ...) \in P_{od} \mid \forall i \in [1, k] : r_i = r'_i\} \subseteq P_{od}$.*

Intuitively, a route subspace $P_{od}(R^k)$ comprises all feasible transit routes that pass stations $s_{r_1}, s_{r_2}, ..., s_{r_k}$ and s_d, as illustrated in Figure 1. Suppose we have a route prefix $R^k = (r_1, r_2, ..., r_k)$. We denote the indexes of all adjacent stations that the k-th station s_{r_k} in R is connected to as $E_k = \{e_1, e_2, ... \mid (s_{r_k}, s_{e_i}) \in E_{od}\}$. To determine which adjacent station s_{e_i} will be selected as the $(k+1)$-th station in R, we must evaluate all subsequent route subspaces derived from route prefixes $R^{k+1}_{e_i} = (r_1, ..., r_k, e_i)$ for every $e_i \in E_k$. To this end, we propose an estimation method that computes the average gain in the given performance criteria of transit routes if a subsequent route subspace is chosen (i.e., the $(k+1)$-th station is decided).

I. Classification of the performance criteria: We first categorize the performance criteria that measure the quality of the generated transit routes into two major types, namely, the independent and dependent criteria.

The *independent criteria* (e.g., route service time, route length, and station construction costs) are not affected by the station selections made at the previous $k-1$ search steps. For example, route service time is defined as

$$T_R = h \times (|R| - 2) + \sum_{i=1}^{|R|-1} T(s_{r_i}, s_{r_{i+1}}),$$

where $R = (r_1, r_2, ...)$ is a transit route, h is the average stopping time per station, and $T(s_i, s_j)$ indicates the estimated travel time between the stations s_i and s_j. By selecting s_n as the next station, route service time increases by

$$\Delta T_R = h + T(s_{r_{|R|}}, s_n),$$

without involving the stations $s_{r_1}, s_{r_2}, ..., s_{r_{|R|-1}}$ selected at the previous steps.

On the contrary, the computation of the *dependent criteria* (e.g., demand satisfaction and route directness) relies on the previously selected stations. For example, the demand satisfaction can be inferred from the number of historical passenger flows $F(s_i, s_j)$ between stations s_i and s_j, thereby resulting in a simplified definition

$$D_R = \sum_{i=1}^{|R|-1} \sum_{j=i+1}^{|R|} F(s_{r_i}, s_{r_j}).$$

Hence, if s_n is selected as the next station, the demand satisfaction will increase by

$$\Delta D_R = \sum_{i=1}^{|R|} F(s_{r_i}, s_n),$$

which accumulates the passenger flows from all previously selected stations.

Without the loss of generality, we will demonstrate the computation of the average gain in the respective representative criteria of two criterion types (i.e., route service time T_R for the dependent criteria and demand satisfaction D_R for the independent criteria) if the route subspace $P_{od}(R^k)$ is chosen.

II(a). Estimation of the independent criteria. The dynamic programming approach is employed to estimate ΔT_R $(P_{od}(R^k))$, the average gain in route service time by choosing the route subspace $P_{od}(R^k)$, based on the station graph. Since the station graph is directed and acyclic, the station nodes in the graph can be sorted topologically to obtain a sequence of stations Q. Let $\eta(s_i)$ be the mapping function that yields the index of the station s_i in the sequence of stations Q. We maintain two variables for each station $s_i \in Q$, namely, the number of feasible routes that pass this station n_i and average route service time t_i measured from this station to the destination s_d. By setting $n_d = 1, t_d = 0$ for the destination s_d and iterating over all stations reversely from $Q_{\eta(s_d)-1}$ to $Q_{\eta(s_{r_k})}$, for each station s_i we have:

$$n_i = \sum_{e_j \in E_i} n_{e_j},$$
$$t_i = \frac{1}{n_i} \sum_{e_j \in E_i} n_{e_j} [t_{e_j} + T(s_i, s_{e_j}) + h].$$

Hence, we have obtained $\Delta T_R(P_{od}(R^k)) = t_{r_k}$, which is the average route service time of all feasible routes passing the chosen station s_{r_k}. The time complexity of this calculation is $\mathcal{O}(|Q|)$, linearly w.r.t. the number of stations.

II(b). Estimation of the dependent criteria: In order to compute $\Delta D_R(P_{od}(R^k))$, the average gain in demand satisfaction by choosing the route subspace $P_{od}(R^k)$, we first compute a counting matrix N where each entry N_{ij} indicates how many feasible routes exist between a pair of stations s_i and s_j in the station graph. This matrix can be easily generated with the dynamic programming approach similar to the estimation of the independent criteria. Thereafter, let s_v be the station that may be possibly selected in a subsequent search step, i.e., $\eta(s_{r_v}) > \eta(s_{r_k})$. To calculate the number of passenger flows $F(s_u, s_v)$ from a station s_u selected before s_v, we derive a weighted passenger flow function $F'(s_u, s_v)$, which is the multiplication of the flow F by the number of feasible routes in the route subspace $P_{od}(R^k)$ that pass s_u and s_v sequentially:

$$F'(s_u, s_v) = \begin{cases} N_{r_k v} N_{vd} F(s_u, s_v), & \text{if } \eta(s_u) \le \eta(s_{r_k}) \\ N_{r_k u} N_{uv} N_{vd} F(s_u, s_v). & \text{otherwise} \end{cases}$$

Therefore, by iterating over the possible intermediate station s_u and averaging the passenger flows, we have

$$\Delta D_R(P_{od}(R^k)) = \frac{1}{N_{r_k d}} \sum_{v=\eta(s_{r_k})}^{\eta(s_d)} \{ [\sum_{u \in R^k} F'(s_u, s_v)] + [\sum_{u=\eta(s_{r_k})+1}^{v-1} F'(s_u, s_v)]\}.$$

However, evaluating this equation directly on the fly is rather inefficient. Observing that only the enumeration of u is related to $R^k \setminus (r_k)$, we can precompute $\Delta D_{R;u,r_k}$ in advance for all $(s_u, s_{r_k}) \in S^2$. Hence, we have

$$\Delta D_R(P_{od}(R_k)) = \sum_{u \in R_k} \Delta D_{R;u,r_k}$$

In this way, the time complexity of computing the aforementioned equation is reduced to $\mathcal{O}(|R_k|)$ with precomputation.

III. Generalization of the estimation method: For the independent criteria, our approach naturally works well because the computation of the gain satisfies the optimal substructure and overlapping subproblem properties required by the dynamic programming algorithm [30] given that A) the estimated gain does not depend on the past selections of stations and B) the station graph is a directed acyclic graph.

For the dependent criteria, our approach is tailored for the criteria computed in pairwise. Nevertheless, for the criteria that require triple-wise or more complex computation procedures, the generalization of the proposed approach can be realized in exchange of efficiency penalty. To the best of our knowledge, however, none of such complex criteria are observed to be extensively used in the transit route planning [1].

B. Extracting Pareto-Optimal Transit Route Set

To efficiently extract the pareto-optimal transit routes scattered across the vast solution space, we integrate the proposed estimation heuristic with a novel random search method, Monte-Carlo route search (MCRS), derived from the MCTS framework. In the proposed method, we extend all four stages of MCTS to enable the proposed method to perform a highly-effective search process on the directed acyclic station graphs.

1) Monte-Carlo Tree Search: We introduce the basic idea of MCTS as follows. Generally, with a given state, a MCTS process is conducted to determine which action to take. The process comprises four consecutive stages, namely, selection, expansion, simulation, and backpropagation stages. The search maintains with a state tree, where the nodes represent states and the edges indicate actions, and repeats these for stages until states are exhausted or the time limit is reached. Initially, the state tree only has a root node representing the given state.

First, at the *selection* stage, we traverse through the state tree starting from the root node. A node is *fully-expanded* if all viable actions has been explored at least once at this state, and the number of child nodes will be equal to the number of viable actions. If the current node is fully-expanded, a child node will be determined to continue the traversal. The most popular method to select a child node is UCB formula [29], which were designed to balance between the number of times the node is visited and the estimated value of the node (i.e., exploration vs. exploitation). Otherwise, we proceed to the next stage with the current node and a randomly-selected viable action which is yet to be explored.

Next, at the *expansion* stage, we randomly select a viable action which is yet to be explored. A new node representing the state after the selected action is taken is then inserted as the child of the previously selected node in the selection stage.

Thereafter, at the *simulation* stage, we follow a default policy, random in the most settings, to move from one state to another, starting from the inserted one. The simulation ends at the final state, and a reward will be obtained.

Finally, at the *backpropagation* stage, we trace back the traversal path taken in the selection stage up to the root of the state tree and update the estimated value for every node on the path with the reward of the simulation.

By repeating these four stages for a number of iterations, we are able to estimate the values of all viable actions at the root node, thereby enabling an informed decision-making process at the current state. For the state-of-the-art research on MCTS, we refer readers to Browne et al.'s survey [10].

TABLE I: The hyperparameters used in the proposed Monte-Carlo route search method.

Notation	Description
λ	The bias between the value of the route subspace and the number of visited times in the ζ-UCB function
α	The bias towards the route subspaces with higher average gains
ρ	The number of search steps
ψ_1	The initial number of search iterations
ψ_{min}	The minimum number of search iterations
μ	The decay of the number of search iterations after each search step
τ	The number of child nodes promoted after each search step

2) Monte-Carlo Route Search: Based on the MCTS framework, we develop our Monte-Carlo route search method as follows. The hyperparameters used in the proposed method are summarized in Table I.

I. Initialization: In MCRS, each node denoted by $m(R^k)$ at depth k in the state tree represents a route subspace $P_{od}(R^k)$. For each state node $m(R_u^k = \{r_1, ..., r_k\})$, we define the viable actions that can be taken at the node as $A_{m(R_u^k)} = E_{r_k}$, where $E_{r_k} = \{e_1, e_2, ...\}$ comprises the indexes of all adjacent stations s_{e_i} of s_{r_k}. Hence, by taking an action $e_u \in A_{m(R_u^k)}$ at the state node $m(R_u^k)$, we can obtain a new child state node $m(R_v^{k+1})$ where $R_v^{k+1} = R_u^k \cup (e_u)$. Initially, a state tree is built with a single root node $m(R^1)$, which represents the root route subspace $P_{od}(R^1)$ derived from the route prefix $R^1 = (o)$ containing only the origin station. In addition, we maintain the number of discovered pareto-optimal transit routes ζ_m and the number of visited times w_m for each node m in the state tree, and a pareto-optimal transit route set P_s that comprises the pareto-optimal transit routes discovered in the search process.

II. Selection: Let m_p be the root node $m(R^1)$ of the state tree. If m_p is fully-expanded, we attempt to select a child node m_q with the maximum ζ-UCB value, which is adapted from the upper confidence bound (UCB) formula [29] by using the number of discovered pareto-optimal transit routes ζ_{m_q} as the value estimation of nodes:

$$\zeta\text{-UCB}(m_q) = \frac{\zeta_{m_q}}{w_{m_q}} + \lambda\sqrt{\frac{\ln w_{m_p}}{w_{m_q}}},$$

where λ is a hyperparameter that balances between the nodes with higher values or less visited times. ζ-UCB formula provides higher flexibility than the traditional one because the removal of pareto-optimal transit routes also affects the value estimations of prior selection paths. The selection then continues recursively with $m_p \leftarrow m_q$. Otherwise if m_p is not fully-expanded, we proceed to the expansion stage with m_p.

III. Expansion: Let the unexplored viable actions at the node $m_p(R_u^k)$ be $A'_{m_p} = \{e_1, e_2, ...\}$. For each action e_i, we estimate the total average gain of the derived route subspace $P_{od}(R_u^k \cup (e_i))$ in all criteria $c_i \in C$ as follows:

$$G(e_i) = \frac{1}{|C|} \sum_{c_i \in C} \mathcal{N}_{c_i}(\Delta c_i(P_{od}(R_u^k \cup (e_i)))).$$

$\mathcal{N}_{c_i}$ normalizes Δc_i linearly to $[\epsilon, 1]$, where $\epsilon = 0.05$ indicates the smallest normalized gain corresponding to the least Δc_i. Then, we choose an action $e' \in A'_{m_p}$ based on the probability of each action defined as follows:

$$\text{Prob}(e') = G(e')^\alpha / (\sum_{e_j \in A'_{m_p}} G(e_j)^\alpha),$$

where α is a hyperparameter that controls the bias towards the route subspaces with higher average gains. After e' is chosen, we insert a new node $m_p(R_v^{k+1} = R_u^k \cup (e'))$ as the child node of $m_p(R_u^k)$ and proceed to the simulation stage.

IV. Simulation: Starting from the station $s_{e'}$, we generate a candidate route R' by selecting the subsequent stations $s_{r_{k+2}}, s_{r_{k+3}}, ...$ recursively with the strategy identical to the one used in the expansion stage until the destination station s_d is reached. Thereafter, we test whether R' is a pareto-optimal transit route by determining if $R' \in P'_s$, where the pareto-optimal transit route set P'_s is obtained by removing all routes that conflict with Definition 1 from $P_s \cup \{R'\}$.

V. Backpropagation: If R' is a pareto-optimal transit route, we increment ζ_{m_i} for every node m_i on the path from m_p to the root node $m(R^1)$. Similarly, for the routes in $P_s \setminus P'_s$ that are dominated by R', we decrease ζ_{m_i} for every node m_i involved in these routes. Thereafter, we repeat the search process from the selection stage with $P_s \leftarrow P'_s$.

VI. Progressive pruning: Initiating the selection process constantly with the root node can be inefficient because pareto-optimal transit routes may only exist in the route subspaces that are associated with some of the descendant nodes. Hence, after a number of iterations ψ_i at the i-th search step, we choose the top-τ child nodes of the current root nodes based on the number of discovered pareto-optimal transit routes, and the subsequent selection processes will be commenced directly from these child nodes instead of the root node. Furthermore, the number of iterations will be shrinked gradually after each search step to reduce the time costs for smaller route subspaces

$$\psi_{i+1} = \min(\mu\psi_i, \psi_{\min}),$$

since we will have more confidence in these route subspaces. The search is terminated after a given number of search steps ρ or if the time limit is exceeded.

V. Evaluation

This section presents the evaluation of our approach conducted empirically on a real-world dataset and compared against the state-of-the-art baseline approach. In addition, we extensively analyzed the hyperparameter sensitivity in our method w.r.t. those defined in Table I.

A. Baseline Approach

To the best of our knowledge, PBS algorithm [9] is the state-of-the-art method that attempts to find pareto-optimal transit routes based on two performance criteria. Thus, we compared our approach with the PBS algorithm regarding the performance of extracting pareto-optimal transit routes.

The PBS algorithm maintains a set of discovered pareto-optimal transit routes and repeats the following procedure on a prebuilt station graph to generate candidate routes. Initially, the algorithm starts from the origin station and greedily selects stations recursively. Suppose k stations have been selected for the candidate route $R' = (r_1, ..., r_k)$, the algorithm randomly selects the next station s_{e_i} from the adjacent stations E_{r_k} based on the probability $\text{Prob}(e_i)$ computed from the accumulated passenger flow F^*:

$$F^*(e_i) = \sum_{r_i \in R'} F(s_{r_i}, s_{e_i}),$$

$$\text{Prob}(e_i) = \frac{F^*(e_i)}{\sum_{e_j \in E_{r_k}} F^*(e_j)}.$$

The algorithm then proceeds with $R' \leftarrow R' \cup (e_i)$ and repeats such selection until the destination station is reached. Finally, the pareto-optimal transit route set is updated to include R'. The algorithm can be terminated after any number of iterations or some time flexibly.

B. Experiment Setup

In this section, we briefly describe the route performance criteria, datasets, testing ground, and evaluation metrics used in the comparative evaluation.

Route performance criteria. The route service time T_R (an independent criterion) and demand satisfaction D_R (a dependent criterion) are selected to measure the quality of the generated pareto-optimal transit routes. By choosing these two criteria, we attempted to ensure a fair comparison because the PBS algorithm was designed purely based on these two criteria, and the algorithm deeply integrated the demand satisfaction criterion into its greedy heuristic.

Data description. We evaluated our approach with the data collected from the existing bus transit network in Beijing, China. Two types of data were extracted as follows.

- *Station data* comprises the coordinates of 1,799 bus stops as the potential stations in S. The distances among the stations is computed with OSRM [31].
- *Passenger flow data* describes the number of trips between each pair of stations extracted from the sampled check-in and check-out records of bus transit cards occurred between Jan., 2012 and May, 2013. In total, 1,422,649 trips are obtained by combining the consecutive records with the same card ID.

Testing ground. We implemented both algorithms in Go programming language. The evaluation was conducted on a Linux workstation with 2x 10-core Intel® Xeon® Silver 4114 CPU @ 2.20GHz and 128G memory. Tests were parallelized with 16 concurrent threads.

Evaluation metrics. To ensure a relatively fair comparison on the algorithm performance, we limited our approach and the baseline to perform the same number of tests to determine whether a candidate route is a pareto-optimal transit route,

TABLE II: Our approach is compared against the baseline method with two pairs of origin and destination stations, one for the small solution space, and the other for the large space. The space size indicates the number of feasible routes in the generated station graph.

	OD Pair #1 (Small)	**OD Pair #2 (Large)**
Origin	Wangjingqiao West	Bajiao Amusement Park
Destination	Zhoujiazhuang	Baliqiao
Distance	13.8km	35km
Space Size	4.863×10^{16}	3.426×10^{30}

instead of the execution time. We compared our approach with the baseline w.r.t. the following metrics:

- The *pareto-optimal transit route set difference* between the pareto-optimal transit route sets generated by our approach and by the baseline. The difference falls between -1 and 1, where 1 indicates our approach produces the pareto-optimal transit routes better than any routes found by the baseline, while -1 indicates the otherwise.
- The *execution time* in seconds indicating the time spent on finding a pareto-optimal transit route set.

C. Performance Comparison

First, we randomly selected 150 pairs of origin and destination stations and run both algorithms for 50 times to obtain stable results. The pareto-optimal route set differences between the results produced by our algorithm and the baseline and the execution time of both algorithms were recorded and presented in Fig. 2. From the results, we observed that in most cases our algorithm significantly outperformed the baseline w.r.t. the pareto-optimal route set difference (Fig. 2a) with much less execution time (Fig. 2b).

To evaluate how the size of solution space affects the algorithm performance, we randomly chose two pairs of origin and destination stations as listed in the Table II, one corresponding to the small solution space and the other corresponding the large solution space. Then, we repeated our approach ($\lambda = 0.005, \alpha = 5, \rho = 10, \psi_1 = 2^15, \psi_{\text{min}} = 2^8, \mu = 0.5, \tau = 8$) and the baseline method for 500 times on these two pairs to obtain a stable result. The result presented in Figure 3 indicates that our approach is highly promising: for both large and small solution spaces, not only our approach has found significantly better pareto-optimal transit routes compared with the baseline (the pareto-optimal transit route set differences are 0.93711 ± 0.00435 for OD pair #1 and 0.96483 ± 0.00905 for OD pair #2, as in Figure 3a), but also with much less time (our approach converges in $9706.81\text{ms} \pm 166.07$ for #1 and $17986.05\text{ms} \pm 528.42$ for #2, while the baseline performs the same number of pareto-optimal transit route tests for $17651.21\text{ms} \pm 233.51$ and $24202.94\text{ms} \pm 594.99$, respectively, as in Figure 3b). The result also shows that our method generates better routes for the large solution space, which is close to the real-world settings, than the baseline. In addition, the criterion distributions of the pareto-optimal transit routes discovered by our approach and by the baseline on OD pair #2 in 20 runs were visualized in Figure 4, showing the dominating advantages of our results.

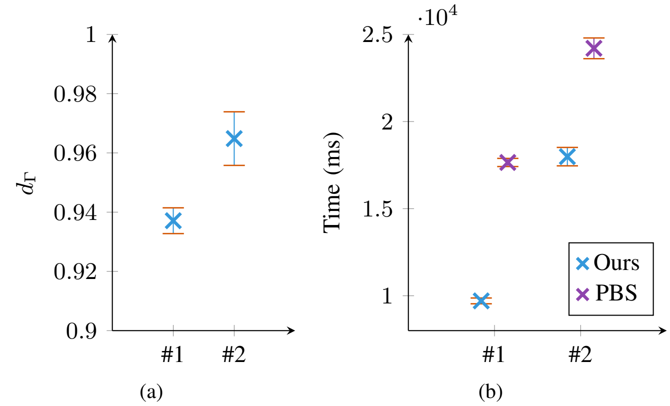
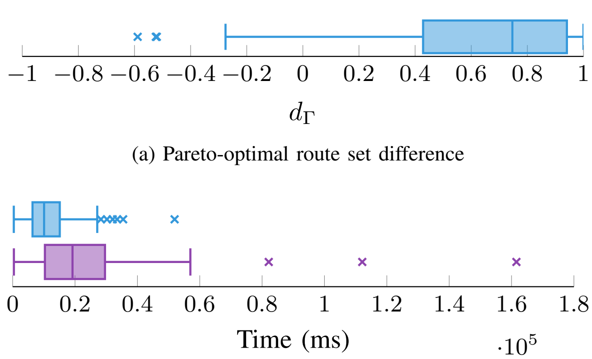


Fig. 3: The performance comparison between our approach and the state-of-the-art baseline. (a) The pareto-optimal transit route difference between our results and the baseline's results on OD #1 and #2. (b) The execution time costs of our method and the baseline on OD #1 and #2.

D. Sensitivity Analysis

The sensitivity of the hyperparameters listed in Table I was evaluated by studying their effects on the pareto-optimal transit route set difference d_Γ compared against the proposed



(a) Pareto-optimal route set difference

(b) Execution time of our algorithm and the baseline

Fig. 2: The comparative evaluation results were obtained from 150 randomly-selected pairs of origin and destination stations.

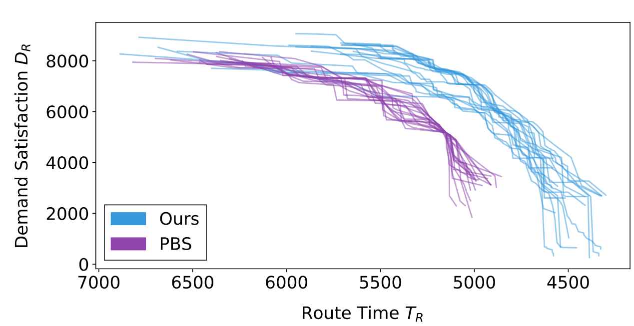
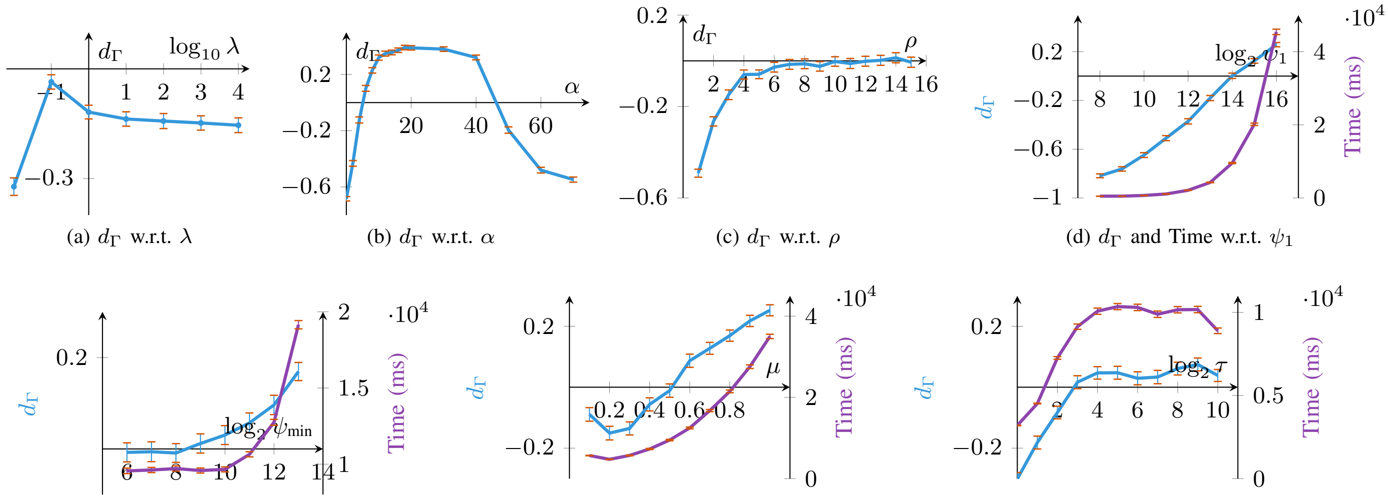


Fig. 4: The visualization of the pareto-optimal transit routes discovered by our method and by the state-of-the-art baseline on OD pair #2 w.r.t. the route service time and demand satisfaction criteria.



(a) d_Γ w.r.t. λ (b) d_Γ w.r.t. α (c) d_Γ w.r.t. ρ (d) d_Γ and Time w.r.t. ψ_1

(e) d_Γ and Time w.r.t. $\psi_{\min}$ (f) d_Γ and Time w.r.t. μ (g) d_Γ and Time w.r.t. τ

Fig. 5: The sensitivity analysis of the hyperparameters. By comparing against the proposed algorithm with the default hyperparameters, the effects of each hyperparameter on the resulting pareto-optimal transit route set difference are extensively studied.

approach with the following default values: $\lambda = 0.005, \alpha = 5, \rho = 10, \psi_1 = 2^{14}, \psi_{\min} = 2^8, \mu = 0.5, \tau = 8$. The evaluation was conducted on OD Pair #1, and each sample was repeated for 500 times in an attempt to obtain stable results. Based on the results (Figure 5), we note a few observations as follows.

Small λ in the proposed ζ-UCT function may result in bad results (Figure 5a) because the algorithm will be trapped in local optima if only the number of discovered pareto-optimal transit routes is considered during the selection stage. In contrast, larger λ leads to the exhaustive search behavior, thereby deteriorating the quality of the results as well. This observation well demonstrates the effectiveness of the ζ-UCT function in guiding the algorithm to select the most productive nodes in the state tree.

As shown in Figure 5b, the proposed estimation heuristic ($10 \leq \alpha \leq 40$) generates better results than simulating randomly ($\alpha = 0$). However, for greater α, the performance of the algorithm decreases due to the aggressive greedy strategy.

The progressive pruning technique is proven to be an effective approach trading off between the search cost and the algorithm performance. With larger ψ_1, $\psi_{\min}$, and μ, the results become incrementally better at the cost of the fast growing search time (Figures 5d, 5e, and 5f). Moreover, increasing the number of search steps ρ and the number of child nodes promoted after each step τ also boosts the performance of the algorithm to a certain extent (Figures 5c and 5g).

VI. Discussion

This section discusses the implications, limitations, and future directions of our study.

A. Implications

Our study is the first step towards the application of sophisticated random search methods in the pareto-optimal planning of multi-criteria transit routes. This problem distinguishes itself from other multi-objective optimization problems that are solvable with MCTS because the complex criteria measuring the performance of transit routes would require tailored heuristics to be better incorporated into the random search framework. Nevertheless, the MCTS framework remains an ideal candidate to solve this problem not only because of its excellent capabilities in search performance and efficiency despite of the huge solution space, but also the progressive nature of the framework where the optimal solution could always be achievable if sufficient computation resources were given. Therefore, based on the MCTS framework, we developed MCRS to tackle the route planning problem by integrating novel heuristics into a tailored efficient search framework. The MCRS method enables transportation experts to determine the most suitable route by conveniently conducting comparative analysis among multiple alternatives instead of purely relying on the traditional optimization blackbox, which tends to produce results that are generally difficult to customized or explained. Furthermore, the efficiency of the MCRS method also facilitates the iterative design procedure of transit routes, in which users may add or remove criteria to rapidly generate different pareto-optimal transit route sets for better decision-making. With MCRS, informed transit route planning procedures can be established to refine the existing transit route topologies and subsequently improve the performance of public transit systems.

B. Limitations

We observed two limitations in our study. The first limitation lies in the efficiency of MCRS. Although MCRS has demonstrated its superior efficiency compared with the state-of-the-art approach, tens of seconds is still required to produce a pareto-optimal transit route set for large solution spaces. Accelerating the proposed method to real-time will greatly benefit many interactive analysis applications, including route planning and public transportation system diagnosis. This can be achieved by parallelizing the MCRS algorithm to utilize multiple cores or even computational clusters. We will leave this as a part of the future work. The second limitation arises from the performance criteria used in the evaluation. Our method naturally supports finding pareto-optimal transit routes based on multiple criteria since both of the hueristics and search framework are criterion-agnostic. However, due to the lack of baselines, we use only two criteria in the comparative evaluation. For three or more criteria, we plan to develop an interactive visualization system based on MCRS for analyzing pareto-optimal transit routes similar to [32] and evaluate the system in the field with transportation experts to unleash the full potential of the proposed method.

C. Future Work

In the future, we plan to reimplement our method with the parallelization and distributed computation techniques to further improve the efficiency in generating pareto-optimal transit routes. Moreover, deep learning models, which have been reported to successfully aid MCTS [21], can be incorporated into MCRS, which may help generating the optimal solution much more effectively. Based on MCRS, we will further develop visual analytics systems to assist transportation experts in better interpreting pareto-optimal transit routes.

VII. Conclusion

In this study, we propose a new estimation method for calculating the average gains of route subspaces based on the given performance criteria. Furthermore, we derive a novel random search approach based on MCTS to approximate the pareto-optimal transit route set on the station graph efficiently with the proposed estimation method. The extensive comparative and sensitivity evaluation has demonstrated the effectiveness of our approach in extracting pareto-optimal transit routes with the considerable improvements in the efficiency and performance over the state-of-the-art method. In the future, we would like to extend our approach by combining visual analytic approaches to assist transportation experts in analyzing pareto-optimal transit routes effectively and evaluate our approach in the field for multi-criteria scenarios.

References

[1] V. Guihaire and J.-K. Hao, "Transit network design and scheduling: A global review," *Transportation Research Part A*, vol. 42, no. 10, pp. 1251–1273, 2008.

[2] C. MacKechnie, "How do bus routes and schedules get planned?" https://www.thoughtco.com/bus-routes-and-schedules-planning-2798726, 2018, [Online; accessed 09-Feb-2019].

[3] F. Zhao and A. Gan, "Optimization of transit network to minimize transfers," Lehman Center for Transportation Research, Florida International University, Tech. Rep. BD015-02, December 2003.

[4] C. E. Mandl, "Evaluation and optimization of urban public transportation networks," *European Journal of Operational Research*, vol. 5, no. 6, pp. 396–404, 1980.

[5] J. Guan, H. Yang, and S. C. Wirasinghe, "Simultaneous optimization of transit line configuration and passenger line assignment," *Transportation Research Part B*, vol. 40, no. 10, pp. 885–902, 2006.

[6] F. Zhao and I. Ubaka, "Transit Network Optimization - Minimizing Transfers and Optimizing Route Directness," *Journal of Public Transportation*, vol. 7, no. 1, pp. 63–82, Mar. 2004. [Online]. Available: http://scholarcommons.usf.edu/jpt/vol7/iss1/4/

[7] W. Fan and R. B. Machemehl, "Optimal transit route network design problem with variable transit demand: Genetic algorithm approach," *Journal of Transportation Engineering*, vol. 132, no. 1, pp. 40–51, Jan. 2006.

[8] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in *Proc. of ICDE*, 2001, pp. 421–430.

[9] C. Chen, D. Zhang, N. Li, and Z.-H. Zhou, "B-Planner: Planning bidirectional night bus routes using large-scale taxi gps traces," *IEEE TITS*, vol. 15, no. 4, pp. 1451–1465, 2014.

[10] C. Browne, E. J. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. P. Liebana, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE TCIAIG*, vol. 4, no. 1, pp. 1–43, 2012. [Online]. Available: https://doi.org/10.1109/TCIAIG.2012.2186810

[11] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE TITS*, vol. 12, no. 4, pp. 1624–1639, 2011.

[12] B. Yu, Z. Yang, C. Cheng, and C. Liu, "Optimizing bus transit network with parallel ant colony algorithm," in *Proc. of EASTS*, vol. 5, 2005, pp. 374–389.

[13] F. Pinelli, R. Nair, F. Calabrese, M. Berlingerio, G. D. Lorenzo, and M. L. Sbodio, "Data-driven transit network design from mobile phone trajectories," *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1724–1733, 2016. [Online]. Available: https://doi.org/10.1109/TITS.2015.2496783

[14] L. Xiao, X. Fan, H. Mao, C. Xu, P. Lu, and S. Luo, "When taxi meets bus: Night bus stop planning over large-scale traffic data," in *Proc. of CCBD*, 2016, pp. 19–24.

[15] Z. Li, J. Wang, Z. Shi, and Y. Zou, "Relieve the congestion by shuttle bus in rush hours using aggregation clustering algorithm on group travel pattern," *Concurrency and Computation: Practice and Experience*, p. e4847, 2018.

[16] F. Zhao and X. Zeng, "Optimization of transit route network, vehicle headways and timetables for large-scale transit networks," *EJOR*, vol. 186, no. 2, pp. 841–855, 2008.

[17] S. Pattnaik, S. Mohan, and V. Tom, "Urban bus transit route network design using genetic algorithm," *Journal of transportation engineering*, vol. 124, no. 4, pp. 368–375, 1998.

[18] W. Y. Szeto and Y. Wu, "A simultaneous bus route design and frequency setting problem for tin shui wai, hong kong," *EJOR*, vol. 209, no. 2, pp. 141–155, 2011.

[19] W. Fan and R. B. Machemehl, "Using a simulated annealing algorithm to solve the transit route network design problem," *Journal of transportation engineering*, vol. 132, no. 2, pp. 122–132, 2006.

[20] D. Delling, T. Pajor, and R. F. Werneck, "Round-Based Public Transit Routing," *Transportation Science*, vol. 49, no. 3, pp. 591–604, Aug. 2015. [Online]. Available: http://pubsonline.informs.org/doi/10.1287/trsc.2014.0534

[21] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[22] X. Guo, S. P. Singh, H. Lee, R. L. Lewis, and X. Wang, "Deep learning for real-time atari game play using offline monte-carlo tree search planning," in *Proc. of NIPS*, 2014, pp. 3338–3346.

[23] B. Kartal, E. Nunes, J. Godoy, and M. L. Gini, "Monte carlo tree search for multi-robot task allocation," in *Proc. of AAAI*, 2016, pp. 4222–4223.

[24] M. H. Segler, M. Preuss, and M. P. Waller, "Planning chemical syntheses with deep neural networks and symbolic ai," *Nature*, vol. 555, no. 7698, p. 604, 2018.

[25] E. J. Powley, D. Whitehouse, and P. I. Cowling, "Monte carlo tree search with macro-actions and heuristic route planning for the multiobjective physical travelling salesman problem," in *2013 IEEE Conference on Computational Inteligence in Games (CIG), Niagara Falls, ON, Canada, August 11-13, 2013*, 2013, pp. 1–8. [Online]. Available: https://doi.org/10.1109/CIG.2013.6633658

[26] D. P. Liebana, E. J. Powley, D. Whitehouse, P. Rohlfshagen, S. Samothrakis, P. I. Cowling, and S. M. Lucas, "Solving the physical traveling salesman problem: Tree search and macro actions," *IEEE TCIAIG*, vol. 6, no. 1, pp. 31–45, 2014.

[27] W. Wang and M. Sebag, "Multi-objective Monte-Carlo Tree Search," in *Proceedings of the Asian Conference on Machine Learning*, 2012, pp. 507–522.

[28] D. Perez, S. Mostaghim, S. Samothrakis, and S. M. Lucas, "Multiobjective Monte Carlo Tree Search for Real-Time Games," *IEEE Transactions on Computational Intelligence and AI in Games*,

vol. 7, no. 4, pp. 347–360, Dec. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/6872573/

[29] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *Proc. of ECML*, 2006, pp. 282–293.

[30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009. [Online]. Available: http://mitpress.mit.edu/books/introduction-algorithms

[31] D. Luxen and C. Vetter, "Real-time routing with openstreetmap data," in *Proc. of SIGSPATIAL*. ACM, 2011, pp. 513–516. [Online]. Available: http://doi.acm.org/10.1145/2093973.2094062

[32] D. Weng, H. Zhu, J. Bao, Y. Zheng, and Y. Wu, "Homefinder revisited: Finding ideal homes with reachability-centric multi-criteria decision making," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*, 2018, p. 247. [Online]. Available: https://doi.org/10.1145/3173574.3173821