A Bottom-Up Design Model for Improving Efficiency of Transit System

Jiayang Li[1], Ruzhang Zhao[1], Meng Li[2,*], Yanfeng Ouyang[3]

***Abstract*— This paper presents a model to design high-performance public transit system, where the optimal solution is directly selected from routes built up on real road structure. The objective function is the travel time for all travelers and the infrastructure cost can also be under consideration. The optimization procedure consists of four components, including a method to simulate downtown travel demand using open-source data on the Internet, a method to narrow down the decision variable space with the help of reasonable restriction and map api, an efficient algorithm to compute average travel time using block representation of the urban area and an improved evolutionary algorithm to search the optimal solution. Eventually, this method is applied to design the transit system for Changzhi, China. The optimal solution is compared to the original transit system of Changzhi to test both the effect and reasonability of the algorithm.**

Key words: Transit system design, Transit system evaluation, Travel demand simulation, Evolutionary algorithm

I. Background

Public transit system is an essential component of urban system because a high-performance transit system can provide significant convenience for all citizens. Meanwhile, the convenience can encourage travelers to reduce the use of private cars, which will ease traffic congestion and pollution. With the increasing severity of urban problems nowadays, more contributions have been devoted to transit system planning. Different kinds of methods have been proposed to solve the transit system planning problem.

To obtain the optimal design, we should analyze the layout of bus line routes over the two-dimensional space. Without elaborate modification, the decision variable is a set of one-dimension lines defined on a two-dimension plane, leading to extremely time-consuming combinatorial optimization. Therefore, these models can only be solved through heuristics [1]–[4]. Although the optimal solution is derived from graph-structure approximation of road network, these approaches will provide relatively feasible network configuration [5]. Another approach is to derive the conceptual plans for geometric idealization of the city [5]: different structures are analyzed, e.g. grids structure [6], hub-and-spoke structure [7], and hybrid structure that combines the former two [8]. This approach is efficient, however, there may exist a gap when adapting the idealized optimal solution to real road conditions, if the road system is complicated. In summary, public transit system planners have to make a tradeoff between the practical feasibility and the computational time. Traditionally, the latter approach receives more recognization.

In this paper, the optimal design is directly obtained from real road conditions. We adopt reasonable method to narrow down the decision variable space with the help of Internet tools and propose an improved evolutionary algorithm to derive the optimal solution. To better present our findings, the rest of this paper is arranged as follows. The mathematical model is proposed in Section 2 and the algorithm is then applied to Changzhi in Section 3. Eventually, Section 4 concludes the paper.

II. Mathematical Model

Initially, we need to set the objective function of the model. Typically, both agency cost and quality of service are required to be under consideration. Quality of service is the measurement of the overall performance of the transit system. In this problem, we only consider characters relevant to the network design, including access time, in-vehicle time, waiting time and transfer time for travelers. The service quality can be evaluated for every traveler with certain travel demand, then we can aggregate the travel demand in the city and derive the average service quality. On the contrary, the agency cost is not easy to be evaluated due to the scale effect and the difficulty for estimating the investment. Even though the cost can be evaluated, the dimension of agency cost is completely different from the the dimension of quality of service. It is difficult to balance agency cost and quality of service in the same formulae. Therefore, only the user level of service is included in the objective function and the agency cost is included in the restriction by evaluating the difference of the new network and the original network, i.e. considering the marginal cost. Denoted AT as the average travel time for aggregate demand, the objective of the model is:

$$\min_{\mathcal{G}} AT(\mathcal{D}, \mathcal{G}),$$

where $\mathcal{D}$ is the aggregate demand and $\mathcal{G}$ is the transit network. This formulae is composed of four parts: $\mathcal{D}$, $\mathcal{G}$, AT, and $\min_{\mathcal{G}}$, leading to four subproblems solved in 4 subsections:

- Subsection A: how to simulate aggregate demand $\mathcal{D}$.
- Subsection B: how to determine transit network $\mathcal{G}$.
- Subsection C: how to compute average travel time AT.
- Subsection D: how to derive optimal solution $arg\min_{\mathcal{G}}$.

[1]Jiayang Li and Ruzhang Zhao are with Department of Mathematics, Tsinghua University, Beijing, China jy-li15, zrz15@mails.tsinghua.edu.cn

[2]Meng Li is with Department of Civil Engineering, Tsinghua University, Beijing, China mengli@tsinghua.edu.cn

[3]Yanfeng Ouyang is with Department of Civil and Environmental Engineering and Department of Industrial and Enterprise Systems Engineering (by courtesy), University of Illinois at Urbana-Champaign, Illinois, U.S.A. yfouyang@illinois.edu

A. *How to simulate the aggregate demand $\mathcal{D}$*

A method using open-source data is proposed in this paper to estimate the OD matrix. Travel demands in the city can be regarded as PoI-PoI (Point of Interest) pairs. Most travelers have at least two travel demands per day, including one from community to certain PoI and another from other place to community. Therefore, we can only consider C - PoI (Community - Point of Interest) pairs while neglect other PoI - PoI demands. Furthermore, among all C-PoI demands, C - S (Community - School) and C - O (Community - Office) demands are stable while others are unstable, like going to the shopping center or the park. If $\mathcal{D}$ is a random variable, then most demands are the random disturbance term while C - S and C - O demands are the constant term. Thus, it is reasonable to use C - S and C - O demands to represent $\mathcal{D}$. Since bus lines in real life always have bi-directions, we can only consider one-way demand. The procedure is:

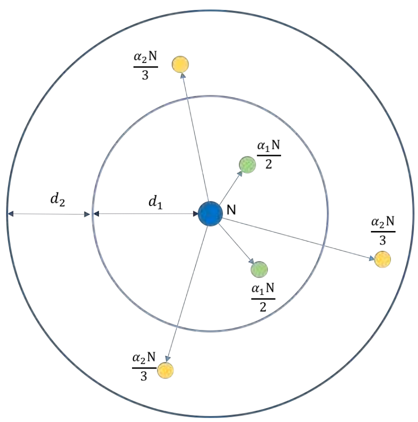

Fig. 1. An example of flow allocation

- Acquire a list of communities and their locations using map api, such as Google api or Gaode api.
- Acquire a list of schools and offices and their locations. Estimate the inflow (or outflow) of these PoIs during the rush hour.
- Assign passenger flow of PoIs to neighboring communities under appropriate rules: α_i percentage inflow (or outflow) of a PoI is from (or to) to communities within $(d_k, d_k + d_{k+1})$ kilometers, distributed uniformly.

The list of PoIs (communities, schools, offices) can be obtained from government and life information website or directly from map api with this function. For example, the list of schools can be downloaded from the local bureau of education, companies can be obtained from recruitment website and the communities can be acquired from website of house trading or renting website. Most map apis also provide function to inquire neighboring PoIs and the PoIs in the whole city is able to be gained by leting the neighborhood of inquired locations cover the urban area.

B. *How to determine the transit network $\mathcal{G}$*

Then we need to find out a method to determine the transit network $\mathcal{G}$. To improve the efficiency of the algorithm, we need to narrow down the variable space.

First, we can split network $\mathcal{G}$ into combination of bus line routes $\mathcal{L}_i$. Denote the urban area as $\mathscr{U}$, e.g. $[x_1, x_2] \times [y_1, y_2]$, then the variable space becomes $C(\mathscr{U})^{n_b}$, where $C(\mathscr{U})$ is the continuous function space defined on $\mathscr{U}$ and n_b is the number of bus lines. Without further modification, the natural variable space is an infinite dimensional space.

Then, we can split each $\mathcal{L}_i$ into points which can determine the route, i.e., split $C(\mathscr{U})$ into $\mathscr{U}^m$, where m is the minimal number of points that can determine a route $\mathcal{L}_i$. The variable space will become $\mathscr{U}^{mn_b}$, which is a finite dimensional space. Heuristically, a route can be determined by turning points. However, the route of bus line in real life always has a lot of turning points, thus the dimension of variable space using turning points is still extremely high. Therefore, we need to use less points to determine the route. A feasible method is: set several critical waypoints and then determine the route between terminal stations and critical points using the route recommended by map api. It can be seen that 1 - 3 critical waypoints can determine a route with much more turning points and the route recommended by map api can go through all kinds of roads with special directions, such as inclining street or crooked street. Meanwhile, some roads are not spacious enough to build up bus lines and the route recommendation from map app will also avoid these streets because the intelligent recommendation algorithm have a preference for main streets.

To narrow finite dimensional space into finite space, we can split $\mathscr{U}$ into representative points. Intuitively, significant PoIs, such as large schools, hospitals, shopping centers can be regarded as critical waypoints. However, then we need to judge which PoI is significant artificially. A more convenient method is to use original bus stations as critical waypoints. The significance of a bus station can be evaluated by the number of bus lines passing through the bus station. Denote the set of critical waypoints as $\mathbb{W}$. Furthermore, we make the assumption that building new terminal station is not allowable, since terminal station requires sufficient space in urban area. Denote $\mathbb{T}$ as the set of original terminal stations in the city. Then, the variable space becomes $(\mathbb{T} \times \mathbb{W}^m \times \mathbb{T})^{n_b}$, which is a finite space.

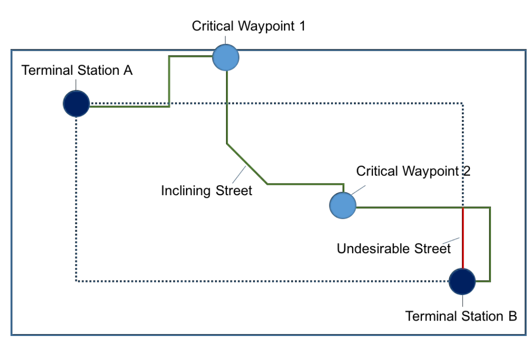

Fig. 2. An example of route determination

The variable space of the problem can be further narrowed down by restricting the feasible region of critical waypoints and neglecting unreasonable terminal station pairs. Given the two terminal stations of a normal bus line and denote the locations as $t_a = (x_a, y_a)$ and $t_b = (x_b, y_b)$, the waypoints of this line are always located in $\mathscr{R}(t_a, t_b) = ((1-\alpha)x_a, (1+\alpha)y_a) \times ((1-\alpha)x_b, (1+\alpha)y_b)$, where α is a small number. This empirical observation can be treated as a reasonable restriction of feasible region. Based on this restriction, given practical terminal stations, there will be no bus line with extremely long routes. In other words, the agency cost is under control. Meanwhile, it is unreasonable to set bus line route between two neighboring terminal stations, thus we only consider terminal station pairs with sufficient distance. Denote this set as $\mathbb{P}$.

Eventually, the variable space turns to be

$$(\{(t_a, w_1, \cdots, w_m, t_b) \in \mathbb{T} \times \mathbb{W} \times \cdots \times \mathbb{W} \times \mathbb{T} : \\ w_1, \cdots, w_m \in \mathscr{R}(t_a, t_b),\ (t_a, t_b) \in \mathbb{P}\})^n$$

C. How to compute the average travel time AT

In this project, we are required to compute the travel time of demands in the whole city and thousands of feasible network design need to be tested. Therefore, an efficient algorithm to compute average travel time is necessary. To improve the efficiency, the urban area need to be discretized into block representation. Denote the index set of blocks as Λ_p and the center location of Block k as p_k, where $k \in \Lambda_p$. The block partition can be based on geographical features, or just cut the urban area into grids with spacing D. Denote the index set of bus lines as Λ_l, then the route of each bus line can be represented by $\mathcal{R}_j : (p_1, \cdots, p_{s_j})$, where $j \in \Lambda_l$ and $p_1, \cdots, p_{s_j}$ is the center location of block k, $k = 1, \cdots, s_j$. Then we can aggregate $\{\mathcal{R}_j : j \in \Lambda_l\}$ into a graph using Algorithm 1. Denote v_{bus} as the average speed of bus and d as the Euclidean distance between points

Algorithm 1 Aggregate $\{\mathcal{L}_j : j \in \Lambda_l\}$ into a graph

for $i = 1 \to |\Lambda_l|$ **do**
  for $j = 1 \to s_j - 1$ **do**
    Add edge (p_j, p_{j+1}) with weight $\frac{d(p_j, p_{j+1})}{v_{bus}}$
  end for
end for

The graph defined by Algorithm 1 captures all routes in transit system, however, the shift from origins or destinations to bus lines is not recorded. To simplify the problem, we make the assumption that travelers will not get on or get off from other blocks if there exists one in the origin and destination blocks. Based on this assumption, we only need to add edges between no station blocks and neighboring blocks with bus lines. Denote v_{walk} as the average walking speed and Λ_a as the index set of blocks with bus lines passing through.

Algorithm 2 Add edges between no station blocks and neighboring stations

for $j \notin \Lambda_p$ **do**
  if $j \in \Lambda_a$ **then**
    $p_n = arg\min_k\{d(p_j, p_k) : k \in \Lambda_a\}$
    Add $Edge$ (p_j, p_n) with weight $\frac{d(p_j, p_n)}{v_{walk}}$
  end if
end for

Then the graph captures all transfers from block to block in the city. Given a travel demand and assume the origin is in Block o and destination is in Block d, then we can find the shortest path between p_o and p_d using different kinds of graph-based shortest path algorithms, like Dijkstra's algorithm. Denote the shortest path as $Path(p_0, p_d) = (p_{j_1}, \cdots, p_{j_t})$, where $p_{j_1} = p_o$ and $p_{j_t} = p_d$. This shortest path ignores the transfer time between bus lines, however, it is still a near-optimal solution for travelers and will be adopted by a lot of travelers in our real life. Although the transfer time is ignored when selecting the best route from origin to destination, it is not negligible when computing the average travel time, because a good transit system should prevent travelers from transferring too many times. Accordingly, we need to compute the transfer time of shortest path using Algorithm 3. This algorithm only adopts one-dimensional search, thus it is efficient. The drawback is that it will regard all paths with more than 2 transfers as 2 transfers.

Algorithm 3 Compute transfer time of $Path(p_o, p_d) = (p_{j_1}, \cdots, p_{j_t})$

$b = arg\min_k\{p_{j_k} : k \in \Lambda_a\}$
$\Lambda_b = \cup_{j:p_{j_b} \in \mathcal{L}_j}\{k : p_k \in \mathcal{L}_j\}$
$e = arg\max_k\{p_{j_k} : k \in \Lambda_a\}$
$\Lambda_e = \cup_{j:p_{j_e} \in \mathcal{L}_j}\{k : p_k \in \mathcal{L}_j\}$
$T_{trans} = 0$
for $k = b \to e$ **do**
  if $p_{j_k} \notin \Lambda_s$ **then**
    $T_{trans} = T_{trans} + 1$
    if $p_{j_k} \notin \Lambda_e$ **then**
      $T_{trans} = T_{trans} + 1$
    end if
    $break$
  end if
end for

Based on the partition of urban area, the travel demand of the city $\mathcal{D}$ can also be discretized into block representation $\mathcal{D} : \{(p_0, p_d) : f_{o,d}\}$, where $f_{0,d}$ is the aggregate flow from p_0 to p_d. Eventually, the algorithm for computing average travel time is illustrated in Algorithm 4, denoted H as the average headway of buses.

D. How to derive the optimal solution $arg\min_{\mathcal{G}}$

In this optimization problem, derivative is difficult to be defined. Therefore, we need to design a heuristic algorithm without the use of derivative. Here an improved evolutionary algorithm is proposed, where bus line $\mathcal{L}$ is gene, multiple genes constitute individual (transit system) $\mathcal{G}$ and several individuals constitute population $\mathcal{P}$. Algorithm 5 illustrates how to constitute population, denoted $Sample(A,\ k)$ as random sample with size k from set A.

Algorithm 4 Compute $AT(\mathcal{D}, \mathcal{G})$

$T_{sum} = 0,\ F_{sum} = 0$
for $(p_0, p_d) \in \mathcal{D}$ **do**
  Derive $Path(p_0, p_d)$ using Dijkstra's algorithm
  $T_{access} + T_{vehicle}$ = length of $Path(p_0, p_d))$
  Derive T_{trans} using Algorithm 3
  $T_{trans} = \frac{1}{2} H T_{trans}$
  $T_{wait} = \frac{1}{2} H$
  $T_{total} = T_{wait} + T_{access} + T_{vehicle} + T_{trans}$
  $T_{sum} = T_{sum} + T_{total} \cdot f_{0,d}$
  $F_{sum} = F_{sum} + f_{o,d}$
end for
$AT(\mathcal{D}, \mathcal{G}) = T_{sum}/F_{sum}$

Algorithm 5 Constitute Population with size n_g

Initialize Population $\mathcal{P}$, $k = 0$
while $k < n_g$ **do**
  Initialize Individual $\mathcal{G}$
  $\mathbb{P}_r = Sample(\mathbb{P},\ n)$
  for $(t_a, t_b) \in \mathbb{P}_r$ **do**
    $w_1, \cdots, w_m = Sample(\mathbb{W} \cap \mathscr{R}(t_a, t_b), m)$
    Add Gene $\mathcal{L} : (t_a, w_1, \cdots, w_m, t_b)$ into $\mathcal{G}$
  end for
  Add Individual $\mathcal{G}$ into $\mathcal{P}$
  $k = k + 1$
end while

During development, individuals are selected through a fitness-based process, where fitter individuals are more likely to survive. In this optimization, the fitness of individual ($\mathcal{G}$) is the average travel time for demand $\mathcal{D}$. The survival process is denoted as $Survival(\mathcal{G},\ k)$, where only the top k individuals survival after competition.

The first biological evolution we need to simulate is the reproduction among individuals in the population. It is time-consuming to consider reproduction among all individuals, thus we assume that superior individuals are more likely to copulate with others. Denote $\mathcal{P}[k]$ as the kth competitive individual in population $\mathcal{P}$ and $rand()$ as $U(0, 1)$ random number, then the reproduction process is illustrated in Algorithm 6. Here we assume that the possibility that $\mathcal{P}[k_1]$ and $\mathcal{P}[k_2]$ copulate is $\frac{3}{k_1+k_2}$.

The reproduction process will not change the gene pool of the population, which is malignant for survival. Therefore, we need to introduce variants in the population. In natural world, variants can be either competitive or vulnerable. While in our evolutionary algorithm, only benign mutation is under consideration. We use greedy algorithm to find benign variants: adjust one gene (bus line) at a time, remaining the terminal stations of the bus line. We assume that superior individuals are more likely to mutate, since the goal of the algorithm is to find optimal solution. The procedure is illustrated in Algorithm 7.

Algorithm 6 Reproduction process of $\mathcal{P}$ with size n_g

Initialize $\mathcal{P}_a$
for $k_1 = 1 \to n_g$ **do**
  for $k_2 = k_1 + 1 \to n_g$ **do**
    if $rand() > \frac{3}{k_1+k_2}$ **then**
      $continue$
    end if
    $\mathcal{G} = Sample(\mathcal{P}[k_1] \cup \mathcal{P}[k_2],\ n_b)$
    Add $\mathcal{G}$ into $\mathcal{P}_a$
  end for
end for
$\mathcal{G} = \mathcal{G} \cup \mathcal{G}_a$
$Survival(\mathcal{G},\ n_g)$

Algorithm 7 Produce Variants in $\mathcal{P}$ with size n_g

for $k = 1 \to n_g$ **do**
  if $rand() < \frac{k}{m_g}$ **then**
    $continue$
  end if
  for $\mathcal{L} : (t_a, w_1, \cdots, w_m, t_b) \in \mathcal{P}[k]$ **do**
    for $v_1, \cdots, v_m = Sample(\mathbb{W} \cap \mathscr{R}(t_a, t_b), m)$ **do**
      $\mathcal{L}' : (t_a, v_1, \cdots, v_m, t_b)$
      $\mathcal{G}$ = change $\mathcal{L}$ to $\mathcal{L}'$ in $\mathcal{P}[k]$
      if $AT(\mathcal{G}, \mathcal{D}) < AT(\mathcal{P}[k], \mathcal{D})$ **then**
        $\mathcal{P}[k] = \mathcal{G}$
        $break$
      end if
    end for
  end for
end for

To prevent local convergence, we introduce migration in the algorithm: eliminate inferior individuals in the population and let superior individuals from another population enter the population. The procedure is illustrated in Algorithm 8.

Algorithm 8 Migration from $\mathcal{P}_2$ to $\mathcal{P}_1$ with same size n_g

for $k = [\frac{n_g}{2}] \to n_g$ **do**
  if $rand() < \frac{2(n_g-k)}{n_g}$ **then**
    $continue$
  end if
  $\mathcal{P}_1[k] = \mathcal{P}_2[n_g - k]$
end for

Eventually, the improved evolutionary algorithm is illustrated in Algorithm 9, where reproduction, mutation and migration occur alternately to produce dominant individuals.

III. The Case of Changzhi

Changzhi is a transportation center in Shanxi, China, facilitated by two railways, three national highways and an airport. Internal transportation also includes a bus and taxi

Algorithm 9 Improved Evolutionary Algorithm

```
Constitue Population P_1 and P_2
while True do
    while k < T_c do
        Simulate Reproduction in P_1 and P_2
        k = k + 1
    end while
    Simulate Mutation in P_1 and P_2
    Simulate Migration P_1 → P_2
    Simulate Migration P_2 → P_1
    Terminate when meeting convergence condition
end while
```

network. In this project, the method proposed in the last section is implemented to design the internal bus network in the urban area of Changzhi. There are two kinds of bus lines in Changzhi: downtown line and suburb line. Downtown line provides downtown-to-downtown service, while suburb line provides suburb-to-downtown service. In this project, we ignore all the suburb lines and all the demands between downtown and suburb. In other words, the scenario is an isolated downtown area without any coupling with the outside world. The reason we omit the suburb part is that most suburb lines are necessary for the city even if the demand is low. Meanwhile, due to the simplex road network structure between suburb and downtown, it is difficult to find alternative main road between downtown and certain suburb area.

A. Data Preparation

We first need to simulate the demand $\mathcal{G}$ of the urban area. Since Changzhi is a small city without Central Business District, we believe that we can only use C - S demands to represent the aggregate demand of all citizens, because large companies are always within the area with large schools. The flow from or to School is estimated from the population of the school, which is obtained from Baidu Encyclopedia.

The locations of middle schools, primary schools and communities are depicted in Fig.3, where the color of middle school is blue, the color of primary school is red and the color of community is green while the size the point is determined by the population. Since the density of communities is relatively high, the uniform allocation of school population in Section 2.1 is also reasonable. The downtown area is divided into grids with spacing $D = 1km$ and we assume that $\alpha_1 = 0.7$, $\alpha_2 = 0.3$, $d_1 = 3km$ and $d_2 = 6km$. Fig.4 lines all the origin and destination of demands in the area, where line-width is related to the level of demand.

The critical waypoints are derived from the original bus stations, which are displayed in Fig.5 and the original terminal stations of downtown lines are depicted in Fig. 6.

B. Numerical Analysis

In the numerical analysis, we set $v_{bus} = 25km/h$, $v_{walk} = 6km/h$ and $H = 10min$ to compute average travel time. With regard to the improved evolutionary algorithm, we set

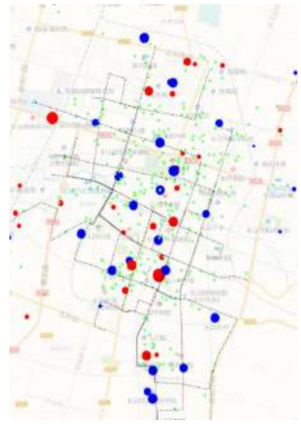

Fig. 3. Schools and Communities

Fig. 4. Demand Simulation

Fig. 5. Critical Points

Fig. 6. Terminal Stations

$m = 2$, $T_c = 5$, $n_b = 12$ and $n_g = 20$. Here $n_b = 12$ is the number of original downtowns lines in Changzhi. The agency cost of transit system is mainly determined by the number of bus lines. Thus, we can assume that the marginal cost is relatively low when $n_b = 12$. The population consists of 20 individuals and the iteration cycle is 5 Reproduction $\rightarrow$ 1 Mutation $\rightarrow$ Migration. The average travel time of each individual in $\mathcal{P}_1$ and $\mathcal{P}_2$ after each biological behavior is displayed in Fig. 7. It can be seen that the convergence rate is relatively high and each kind of biological behavior is helpful for the convergence.

The top six solutions selected from the last generation of $\mathcal{P}_1$ and $\mathcal{P}_2$ are illustrated in Fig. 8. The optimal travel time is about 19 min while the travel time based on the original network is about 24.5 min.

IV. Conclusion

In transit system planning, top-down design has received more people's support, where the optimal solution is derived from geometric idealization and then adapted to real condition. On the contrary, the model presented in this paper is a bottom-up design, beginning with routes in real road network. Internet data and efficient algorithm provide the feasibility of this method. This paper demonstrates a convenient and efficient method to simulate the aggregate

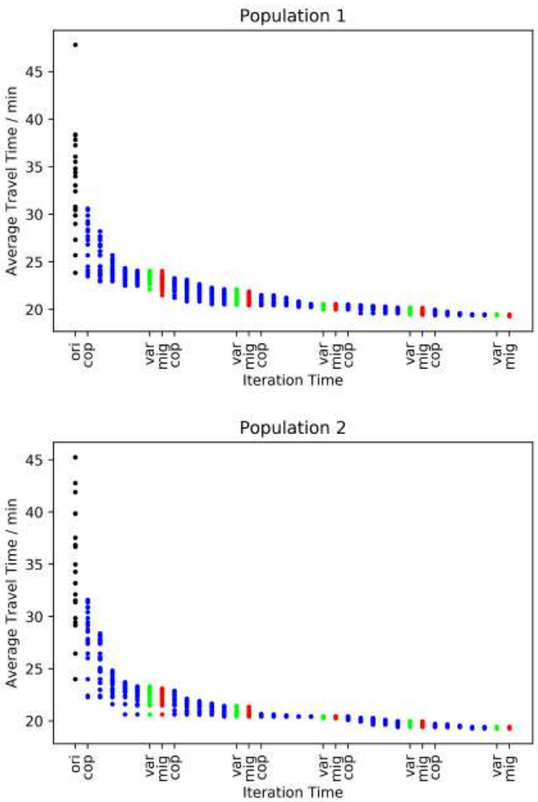

Fig. 7. Convergence Process

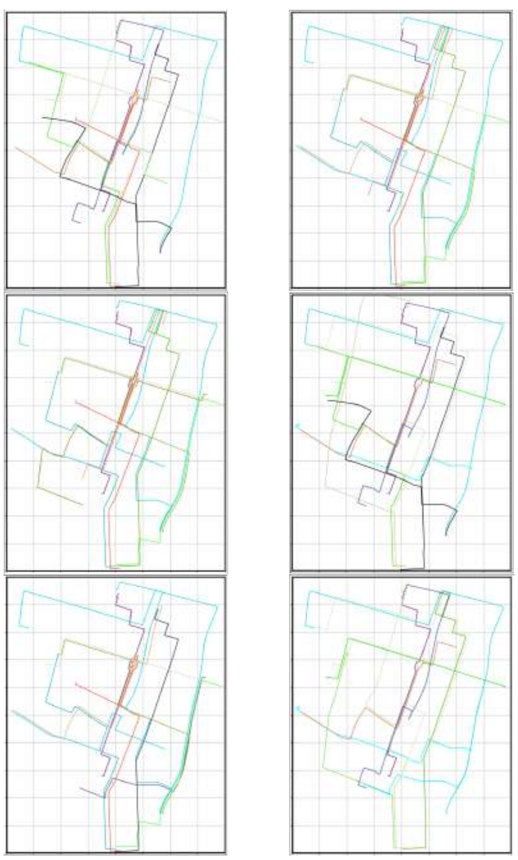

Fig. 8. Top six designs

demand of the city, an approximate method to estimate the average travel time and an improved evolutionary algorithm to derive optimal solution. Meanwhile, it can be believed that the original transit network is designed by city manager elaborately and has been adjusted for several times, thus it is already a well-performed system. It can be noticed that the difference of average travel time between optimal design and original system is notable but not exaggerated, which also supports the reliability of the method. This paper can also be extended in the following way:

- In the demand simulation process, we use C - PoI demands to approximate the aggregate demands in the entire urban area. Other demand certainly can be under consideration to improve the accuracy. While nowadays, the penetration rate of cell phone is almost 100%. Thus, the best way is to use cell phone signaling data to estimate travel demand.
- The suburb line design is ignored in this project for simplicity. Some of the suburb lines have fairly long routes in the urban area and this part can also be redesigned. By detecting the crossing point of the suburb line and the boundary of downtown area and regarding the crossing points as terminal stations, the scenario will be degenerated to the one in this paper.
- In this paper, we search the optimal solution without changing the number of bus lines in the original network. We can also minimize the agent cost (number of bus lines) without reducing the average travel time for travelers.

ACKNOWLEDGEMENT

This work was supported by the collaboration project of Department of Civil Engineering, Tsinghua University and Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign.

REFERENCES

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

[2] M. H. Baaj and H. S. Mahmassani, "Hybrid route generation heuristic algorithm for the design of transit networks," *Transportation Research Part C Emerging Technologies*, vol. 3, no. 1, pp. 31–50, 1995.

[3] S. B. Pattnaik, "Urban bus transit route network design using genetic algorithm," *Journal of Transportation Engineering*, vol. 124, no. 4, pp. 368–375, 1998.

[4] J. Ding, S. Feng, L. Li, and Y. Zhang, "Campus bus network design and evaluation based on the route property," *Tsinghua Science and Technology*, vol. 22, no. 5, pp. 539–550, 2017.

[5] M. Estrada, M. Roca-Riu, H. Badia, F. Robust, and C. F. Daganzo, "Design and implementation of efficient transit networks: Procedure, case study and validity test," *Transportation Research Part A*, vol. 17, no. 9, pp. 113–135, 2011.

[6] E. M. Holroyd, "The optimum bus service: A theoretical model for a large uniform urban area," in *International Symposium on the Theory of Traffic Flow*, 1967.

[7] G. F. Newell, "Some issues relating to the optimal design of bus routes," *Transportation Science*, vol. 13, no. 1, pp. 20–35, 1979.

[8] C. F. Daganzo, "Structure of competitive transit networks," *Transportation Research Part B Methodological*, vol. 44, no. 4, pp. 434–446, 2010.