CASPT 2015

Transit Network Design by Hybrid Guided Genetic Algorithm With Elitism

Md. Khaledur Rahman · Muhammad Ali Nayeem · M. Sohel Rahman

Abstract Transit network design problem (TNDP) deals with devising an efficient set of routes for public transport. In this paper we propose Guided Genetic Algorithm With Elitism (GGAWE), a hybridization of two metaheuristic techniques to solve TNDP. To validate our proposed approach, we have conducted extensive experiments on both benchmark data and real network data. Our experimental results reveal the superiority of our approach over the state of the art.

Keywords transit network design · metaheuristics · genetic algorithm · hybridization

1 Introduction

Efficient public transport systems are in high demand in every corner of the world. As a result significant research effort has been given to design optimal transport systems that involve maximizing the number of satisfied passengers, minimizing the total number of transfers and minimizing the total travel time of all served passengers. Several studies have suggested that computer based tools should be employed more for designing and evaluating public transit networks (Nielsen (2005); Zhao and Gan (2003)).

In this paper, we study the transit routing problem and present a hybrid metaheuristics framework for solving it. We introduce Guided Genetic Algorithm With Elitism (GGAWE) for the transit network design problem, that allows us to concentrate on the key issues of minimizing the travel time and the number of transfers simultaneously. For the information, authors have recently introduced Genetic Algorithm With Elitism (GAWE) and its variant

Md. Khaledur Rahman, Muhammad Ali Nayeem, M. Sohel Rahman
AℓEDA Group, Dept. of CSE, BUET, Dhaka-1000, Bangladesh
Tel.: +880 29665612
Fax: +880 29665612
E-mail: {khaled.cse.07, nayeem007, sohel.kcl}@gmail.com

(Nayeem et al (2014)), but GGAWE is different from it. GGAWE is totally a novel hybrid technique where the problem of getting stuck in local optima has been overcome. Very briefly GGAWE focuses on certain edges of a route set which tend to cause local optima and tries to avoid them. We show the effectiveness of our schemes, by comparing our results with previously published results (Nayeem et al (2014); Nikoli and Teodorovi (2013)) on some benchmark instances.

The rest of this paper is organized as follows. A brief literature review is given in section 2. Section 3 formally defines the problem. Proposed solutions to the problem are given in section 4. Experimental results and analyses are provided in section 5. Finally, we briefly conclude in section 6 with some future research directions.

2 Literature Review

Since TNDP is a hard optimization problem, a number of heuristic and meta-heuristic based approaches have been proposed in the literature. We start by reviewing a heuristic algorithm proposed by Mandl to find a set of the best transit routes (Mandl (1979)). He developed a solution in two stages: first a feasible set of routes was generated, and then heuristics were applied to improve the quality of the initial route set. His proposed network is used as one of the benchmark networks in the literature.

A few review and survey papers have been published in the literature that documents and discusses the results in the literature from different perspectives. Kepaptsoglou and Karlaftis (2009) presented and reviewed research results in the area of transit route network design problem. Design objectives, operating environment parameters and solution approaches are especially analyzed in the paper. The review of Derrible and Kenneday (2011) is devoted to the applications of the graph theory in transit network design. Schoebel (2012) made a review of the various bus, railway, tram, and underground line planning models.

Baaj and Mahmassani (1995) described and implemented a heuristic route generation algorithm for the route network design. Generally it determines an initial set of skeletons and expands them to form transit routes, which heavily depends on the travel demand matrix. On the other hand, Charkroborty and Dwivedi (2002) took an approach for encoding by listing the nodes explicitly in their genetic algorithm. This work was enhanced further by Chakroborty (2003) to cover scheduling as well as routing.

Fan and Machemehl (2008) considered the design of public transportation networks in the case of variable demand. The authors developed a multi-objective model. Their solution methodology was based on the tabu search method. Later, Fan and Mumford (2010) proposed a model of the urban transit routing problem that evaluated candidate route sets. Their proposed approach used hill-climbing and simulated annealing techniques.

Recently, Nikoli and Teodorovi (2013) have introduced swarm optimization namely bee colony optimization algorithm in TNDP. Following up the work of Nikoli and Teodorovi, Nayeem et al (2014) have proposed two variations of genetic algorithm to solve the TNDP which to the best of our knowledge is the best result in the literature to date.

3 Problem Definition

We define the Transit Network Design Problem (TNDP) following the terminologies of Nikoli and Teodorovi (2013). Here, a road network is described by the graph $G = (N, E)$, where N is the set of nodes representing the bus stops and E is the set of edges representing the street segments. A route used by the transit passengers is described by a path in the graph.

We have a demand matrix denoted by d_{ij}, which represents the number of trips per time unit between node i and node j. We also denote by D the origin-destination matrix (O-D matrix) as follows:

$$D = \{d_{ij} | i, j \in [1, 2, ..., |N|]\} \quad (1)$$

We assume that the given road network is connected and undirected and there are sufficient vehicles on each route to ensure that the demand between every pair of nodes is satisfied. We also know the travel time matrix for the road network denoted by tr_{ij}, which represents the in-vehicle travel time between the node i and the node j. By TR, we denote the travel time matrix:

$$TR = \{tr_{ij} | i, j \in [1, 2, ..., |N|]\} \quad (2)$$

The main indicator that we use to describe the level of transit service is the total travel time spent by the users of the transit service. We express the quality of the solution generated in minutes. We calculate the total travel time T of all passengers in the network in the following way:

$$T = TT + w_1 TTR + w_2 TU, \quad (3)$$

where,

TT is the total in-vehicle time of all served passengers,

TTR is the total number of transfers in the network,

TU is the total number of unsatisfied passengers (we assume that the passenger is unsatisfied when she/he has to make more than two transfers during the trip),

w_1 is the time penalty for one transfer (5 mins),

w_2 is the time penalty for one unsatisfied passenger (Optimal Average Travel Time (ATT) + 50 mins).

Therefore, the fitness of a route set is $\frac{1}{T}$. So our objective is to find a set of routes R such that T is minimized. In short, we can define the transit network design problem in the following way: For a given set of n nodes, known origin-destination matrix D that describes demand among these nodes, and known travel time matrix TR, generate a set of transit routes which we call route set on a network such that the total travel time T of all passengers is minimized.

4 The proposed solution

In this work, we have used hybridization of two metaheuristic techniques Guided Local Search (GLS) and Genetic Algorithm With Elitism (GAWE) which we call the Guided Genetic Algorithm With Elitism (GGAWE). We have customized the traditional implementations to match with our designed approach for solving TNDP. As will be reported later, the smooth exploitative nature of GAWE combined with the exploration force of GLS, successfully results in quite high quality route sets.

GLS, introduced by Voudouris and Tsang (1995), is basically a variation of hill-climbing that tries to identify solution components which appear too often in local optima, and penalizes later solutions which use those components so as to force exploration elsewhere. GGAWE is a population based method which takes the idea of maintaining diversity of route sets in the population from GLS but instead of using hill-climbing (as is the case in GLS) it uses GAWE for producing new route sets. GAWE is an exploitative version of genetic algorithm where the initial route set is built using a greedy algorithm and gradually improved with specially designed mutation and crossover operators (Nayeem et al (2014)).

For transit network designing we have treated the edges of the road network as components. We have maintained a matrix of edge penalties keeping records of how often each edge has appeared in high-quality routes. We denote by p_{ij} the penalty of edge between i and j, e_{ij}. Instead of using fitness we have used adjusted fitness which takes both fitness and the edge penalties into account. Given a route set R, adjusted fitness of R is defined as follows:

$$AdjustedFitness(R) = Fitness(R) - \beta \sum_{\forall\ e_{ij} \in E\ found\ in\ R} p_{ij} \tag{4}$$

Thus GGAWE is looking for route sets both of high quality but also ones which are relatively novel in the sense that it tries to use edges which havent́ been used much in high quality routes before. The parameter β determines the degree to which novelty figures in the final fitness computation.

After several generations of modified GAWE in this adjusted fitness space, GGAWE then takes its current best route set S from the current population, which is presumably at or near a local optimum, and increases the penalties of certain edges which can be found in this route set. To be likely to have its penalty increased, an edge must have three qualities. First, it must appear in Sthat is, it is partly responsible for the local optimum and should be avoided. Second, it will tend to have lower passenger demand and higher travel time: we wish to move away from the least important edges in the route set first. Third, it will tend to have lower penalty. This is because GGAWE does not want to penalize the same components forever.

To determine the edges whose penalties should be increased, GGAWE first computes the penalizability of each edge e_{ij} in S with current penalty p_{ij} as follows:

$$Penalizability(e_{ij}) = \frac{tr_{ij}}{(1 + p_{ij})(1 + d_{ij})} \tag{5}$$

We normalize the *Penalizability* of each edge e_{ij} using the maximum and minimum penalizability of all edges found in S as follows:

$$\frac{Penalizability(e_{ij}) - Minimum\ Penalizability}{Maximum\ Penalizability - Minimum\ Penalizability} \tag{6}$$

GGAWE then picks all those edges whose normalized penalizability is greater than 0.8 and increments their penalties by 1.

If we let the edge penalties keeping on piling up, it might happen that an edge once responsible for causing local optima is always banned from participating in the routes even if it is no more responsible for local optima at some later part. So GGAWE also decreases all the edge penalties a bit at each iteration.

To sum up, GGAWE starts with a global population of initial route sets. At each iteration it applies modified GAWE on the global population for some generations using adjusted fitness. Then GGAWE adjusts the current edge penalties by increasing penalties for edges commonly found in local optima and decreases all edge penalties at a fixed rate. The full algorithm is presented in Algorithm 1.

Note that in GGAWE we have employed a modified version of GAWE (Nayeem et al (2014)). Now we summarize the modifications we have made in previous implementation of GAWE (Nayeem et al (2014)) to suit with the current frame. Here we keep on counting the number of successive generations where the fitness of the solution that is best so far is not improved. If this count reaches a predefined parameter we halt GAWE and replace m number of randomly selected solutions from the current population with our extracted initial solution. Here m is a random number between $[0-2]$. Now the selection procedure considers the adjusted fitness of the route set. GAWE uses two different mutations namely small modification and big modification. In small modification, it expands or shortens a route by one edge. While expanding a route, it selects one of the terminals and adds a new bus stop randomly chosen from the nodes adjacent to the selected terminal. But in modified GAWE (Algorithm 2), while expanding a route, we now select the new bus stop to add from the adjacent nodes using roulette wheel selection with the probability defined as inverse of edge penalty. Thus we encourage adding edges with lower penalties.

5 Experimental Results

To conduct our experiments, we have used a gcc-g++3.4.5 compiler with ParadisEO2.0 framework in Code::Blocks13.12 IDE and Netbeans 7.4 IDE. We

Fig. 1 Algorithm 1: Guided Genetic Algorithm With Elitism (GGAWE)

```
popSize ← desired population size
eliteSize ← desired number of eilte individuals
routeSetSize ← number of routes in the individual
t ← tournament size for fitness
P_ms ← probability of doing small modification in Mutation          ▷ high
P_delete ← probability of deleting the selected terminal in small modification
maxGen ← maximum number of generations for each run of GAWE
stableCount ← maximum number of successive generations in each run of GAWE where
the best fitness is same
maxIter ← maximum number of iterations of GGAWE
p ← matrix of edge penalties, initially zero
r_d ← decreasing rate of edge penalty
β ← significance of edge penalties in adjusted fitness
P ← {}
Best ← ∅                  ▷ keeps the best so far route set according to actual fitness
S ← ∅                     ▷ keeps the best so far route set according to adjusted fitness
for popsize times do
    P ← P ∪ {InitialRouteSet(routeSetSize)}
end for
for iter ← 1 to maxIter do
    Call Modified GAWE (Algorithm 2)
    for each edge e_ij appearing in S do
        Calculate Penalizability(e_ij)
        Keep maximum penalizability in maxPen and minimum penalizability in minPen
    end for
    for each edge e_ij appearing in S do
                  ▷ Penalize the top penalizable edges by increasing their penalties
        if (Penalizability(e_ij) − minPen) / (maxPen − minPen) > 0.8 then
            p_ij ← p_ij + 1
        end if
    end for
    for each edge e_ij do
        p_ij ← (1 − r_d)p_ij
    end for
    iter ← iter + 1
end for
return Best
```

have used two Desktop PCs and one Labtop PC of 3.3 GHz Intel Core i3 processor with 8 GB RAM, 3.5 GHz Intel Core i7 processor with 8 GB RAM and 2.5 GHz Intel Core i5 processor with 6 GB RAM respectively.

To validate the expected improvement by the proposed hybrid technique, we have compared the performance of our algorithm with other methods found in the literature in terms of the performance metrics shown in Table 1.

The values of the performance metrics d_0, d_1, d_2 and d_{un} collectively show the quality of a route set. For any route set, the summation of values of these four metrics equals to 1. At first, a route set tries to meet the passenger demand as much as possible without any transfer; then it tries to meet the remaining demand as much as possible with only one transfer; then it tries to do the same for the remaining demand with two transfers; and finally, the rest of the demand are left as unsatisfied. A good route set always tries to meet the most

Fig. 2 Algorithm 2: Modified GAWE

$stableCount \leftarrow 0$
$prevBest \leftarrow Best$
for $gen \leftarrow 1$ to $maxGen$ **do**
  for each individual $P_i \in P$ **do**
    AssessAdjustedFitness(P_i)
    if $Best = \emptyset$ or Fitness(P_i) $> Fitness(Best)$ **then**
      $Best \leftarrow P_i$
    end if
    if S $= \emptyset$ or AdjustedFitness(P_i) $>$ AdjustedFitness(S) **then**
      $S \leftarrow P_i$
    end if
  end for
  $Q \leftarrow$ the $eliteSize$ individuals in P with highest AdjustedFitness, breaking ties at random
  for $(popSize - eliteSize)/2$ times **do**
    Parent $P_a \leftarrow$ TournamentSelection(P)   ▷ Using AdjustedFitness
    Parent $P_b \leftarrow$ TournamentSelection(P)   ▷ Using AdjustedFitness
    Children $(C_a, C_b) \leftarrow UniformCrossover(Copy(P_a), Copy(P_b))$
    $Q \leftarrow Q \cup \{$Mutation(C_a), Mutation(C_b)$\}$
  end for
  $P \leftarrow Q$
  if AdjustedFitness($prevBest$) $=$ AdjustedFitness($Best$) **then**
    $stableCounter \leftarrow stableCounter + 1$
    if stableCounter $=$ stableCount **then**
      Randomly Shuffle P
      $randValue \leftarrow$ pick a random number uniformly between 0 to 2
      for $i \leftarrow 0$ to $randValue - 1$ **do**
        $P \leftarrow P - P_i$
        $P \leftarrow P \cup \{InitialRouteSet(routeSetSize)\}$
      end for
      $Break$
    end if
  else
    $prevBest \leftarrow Best$
    $stableCounter \leftarrow 0$
  end if
  $gen \leftarrow gen + 1$
end for

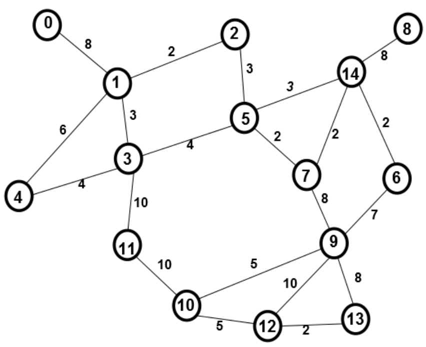

Fig. 3 Mandl's Swiss Road Network.

Table 1 Performance Metrics

d_0	the percentage of demand satisfied without any transfers
d_1	the percentage of demand satisfied with one transfer
d_2	the percentage of demand satisfied with two transfers
d_{un}	the percentage of demand unsatisfied
ATT	average travel time in minutes per transit user (mpu). This travel time includes transfer waiting times, and transfer time that is equal to 5 min per passenger

of the passenger demand with the least number of transfers. The effect of the distribution of passenger demand among d_0, d_1, d_2 and d_{un} is summarized by the value of ATT as it considers the time wasted by the transfers. Therefore, it shows the overall quality of a route set. So if we have two route sets having very close values of d_0, d_1, d_2 and d_{un}, we can compare them using ATT, which should be lower for a better solution.

Nayeem et al (2014) had another version of genetic algorithm named GAWIP where at each generation the population size is increased by the number of *elite* individuals. Although GAWIP can produce high quality solutions in case of small bus networks such as Mandl's 15 node Swiss network, its running time is huge. As a result, GAWIP is not suitable for real bus networks where there are numerous nodes and edges. So we have decided to omit GAWIP from our experimental study here.

At first we show our experimental results in case of Mandl's road network (15 nodes and 20 edges) which is currently the only benchmark available (see Figure 3). Here we have compared the performance of our best final solution to other previous works considering four situations: 4 routes, 6 routes, 7 routes and 8 routes in each route set. The parameter values of our Algorithm 1, used for all these experiments are shown in Table 5. The comparison is shown in Table 2. We can see that the results obtained by GGAWE is very competitive with GAWE and clearly better than other previous works.

Parameter	**Value**
$popSize$	18
$eliteSize$	2
t	10
P_{swap}	$\frac{1}{routeSetSize}$
P_{ms}	0.7
P_{delete}	0.4
$maxGen$	20
$maxIter$	20
$stableCount$	15
β	0.8

Table 2 The comparison among the final solutions generated by our approach and the previous approaches for Mandl's route network.

No. of routes	Metrics	Mandl (1979)	Baaj and Mahmassani (1995)	Kidwai (1998)	Charkroborty and Dwivedi (2002)	Fan and Machemehl (2008)	Nikolic and Teodorovic (2013)	GAWE	GGAWE
4	d_0	69.94	N	72.95	86.86	93.26	92.1	96.14	96.40
	d_1	29.93	N	26.92	12	6.74	7.19	3.47	3.15
	d_2	0.13	N	0.13	1.14	0	0.71	0.39	0.45
	d_{un}	0	N	0	0	0	0	0	0
	ATT	12.9	N	12.72	11.9	11.37	10.51	10.49	10.50
6	d_0	N	78.61	77.92	86.04	91.52	95.63	98.39	98.39
	d_1	N	21.39	19.68	13.96	8.48	4.37	1.61	1.61
	d_2	N	0	2.4	0	0	0	0	0
	d_{un}	N	0	0	0	0	0	0	0
	ATT	N	11.86	11.87	10.3	10.48	10.23	10.14	10.13
7	d_0	N	80.99	93.91	89.15	93.32	98.52	99.17	99.49
	d_1	N	19.01	6.09	10.85	6.36	1.48	0.83	0.51
	d_2	N	0	0	0	0.32	0	0	0
	d_{un}	N	0	0	0	0	0	0	0
	ATT	N	12.5	10.69	10.15	10.42	10.15	10.07	10.07
8	d_0	N	79.96	84.73	90.38	94.54	98.97	99.86	99.87
	d_1	N	20.04	15.27	9.62	5.46	1.03	0.14	0.13
	d_2	N	0	0	0	0	0	0	0
	d_{un}	N	0	0	0	0	0	0	0
	ATT	N	11.86	11.22	10.46	10.36	10.09	10.03	10.03

To show the applicability of GGAWE in real world, we have also performed several experiments with three real bus networks namely Mumford1, Mumford2 and Mumford3 (Mumford (2013)) developed based on information manually extracted from bus route network maps for three real cities. The properties of those real networks are shown in Table 5.

For the sake of comparison with previous works we have run GGAWE for a total of 200 generations ($maxGen$ 20, $maxIter$ 10) and reported the average results over 20 independent runs for each network. All the other metrics of Table 5 have been kept same. In case of these large networks we find that GGAWE is clearly ahead of GAWE and the work of Mumford (2013) (see Table 6). We have found that GGAWE works well for bus networks with a large number of nodes and edges. This might be due to the fact that GGAWE en-

Table 3 Comparison among GLS, GAWE (Nayeem et al (2014)) and GGAWE for Mandl's route network.

Network (no. of route)	Metrics	GLS			GAWE			GGAWE			
		Best	Avg	Worst	Best	Avg	Worst	Best	Avg	Worst	Variance
Mandl (4)	d_0	94.61	91.59	88.44	96.14	93.39	86.96	96.40	93.76	91.20	1.78E-04
	d_1	5.07	7.29	7.06	3.47	5.55	8.8	3.15	5.48	7.71	1.63E-04
	d_2	0.32	1.12	4.50	0.39	1.06	4.24	0.45	0.76	1.09	1.10E-05
	d_{un}	0	0	0	0	0	0	0	0	0	0
	ATT	10.39	10.72	11.03	10.49	10.5	11.12	10.50	10.50	10.75	1.04E-02
Mandl (6)	d_0	95.95	93.62	90.82	98.39	97.5	96.08	98.39	97.48	96.47	2.68E-05
	d_1	3.98	5.82	5.20	1.61	2.49	3.92	1.61	2.48	3.15	2.29E-05
	d_2	0.07	0.36	0	0	0.01	0	0	0.04	0.38	1.21E-06
	d_{un}	0	0.20	3.98	0	0	0	0	0	0	0
	ATT	10.27	10.42	10.21	10.14	10.17	10.22	10.13	10.17	10.25	8.76E-04
Mandl (7)	d_0	96.08	95	93.96	99.17	98.35	97.24	99.49	98.66	97.37	2.91E-05
	d_1	3.92	4.74	5.52	0.83	1.65	2.76	0.51	1.34	2.63	2.91E-05
	d_2	0	0.26	0.52	0	0	0	0	0	0	0
	d_{un}	0	0	0	0	0	0	0	0	0	0
	ATT	10.26	10.31	10.38	10.07	10.11	10.16	10.07	10.10	10.17	7.55E-04
Mandl (8)	d_0	97.11	95.53	93.39	99.87	99.28	98.65	99.87	99.31	98.46	1.53E-05
	d_1	2.76	4.35	6.61	0.13	0.72	1.35	0.13	0.69	1.54	1.53E-05
	d_2	0.13	0.12	0	0	0	0	0	0	0	0
	d_{un}	0	0	0	0	0	0	0	0	0	0
	ATT	10.19	10.27	10.38	10.03	10.07	10.10	10.03	10.06	10.09	3.05E-04

forces exploration implicitly by adjusting the edge penalties which ultimately affects the operation of our selection and mutation operators. This in turn encourages choosing novel edges in the route set. But to reveal their strength we need to provide them enough options to choose from. So when the network contains a large number of edges, these operators get the opportunity to come into action as a result GGAWE seems to make the difference. Finally we have summarized the improvement of GGAWE over GLS and GAWE in Table 3

Table 4 Comparison among GLS, GAWE (Nayeem et al (2014)) and GGAWE for real bus networks.

Network (no. of route)	Metrics	GLS			GAWE			GGAWE		
		Best	Avg	Worst	Best	Avg	Worst	Best	Avg	Worst
Mumford1 (15)	d_0	29.31	27.59	24.96	41.39	37.71	33.11	44.34	40.33	38.26
	d_1	54.82	54.18	58.9	53.73	56.37	58.63	52.39	55.55	56.40
	d_2	12.67	12.01	12.85	4.9	5.9	8.3	3.27	4.12	5.34
	d_{un}	3.20	6.22	3.30	0.02	0.03	0	0	0	0
	ATT	26.45	26.65	26.84	23.64	23.96	24.35	23.37	23.60	23.47
Mumford2 (56)	d_0	28.82	27.56	26.63	34.09	32.53	31.4	35.63	33.92	33.03
	d_1	58.96	58.46	60.73	61.94	93.53	63.64	61.51	63.01	63.85
	d_2	6.65	7.43	7.12	3.9	3.9	5.0	2.86	3.07	3.12
	d_{un}	5.57	6.55	5.52	0	0	0	0	0	0
	ATT	27.74	27.87	28.10	26.58	26.63	26.64	26.45	26.45	26.43
Mumford3 (60)	d_0	25.79	24.79	23.58	30.20	29.15	28.07	31.47	30.29	29.07
	d_1	55.40	53.40	55.17	63.83	64.31	64.03	63.17	64.29	64.97
	d_2	9.40	9.22	11.19	6.0	6.5	7.9	5.36	5.42	5.96
	d_{un}	9.41	12.59	10.06	0	0	0.02	0	0	0
	ATT	30.86	30.92	31.28	29.47	29.65	29.75	29.43	29.41	29.50

Table 5 Properties of Real Data Sets.

Data set	Location	No. of Nodes	links	No. of Routes
Mumford1	Yubei	70	210	15
Mumford2	Brighton	110	385	56
Mumford3	Cardiff	127	425	60

Table 6 Comparison of our results with Mumford (2013) and GAWE (Nayeem et al. 2014).

	Mumford1			Mumford2			Mumford3		
Metrics	Mumford (2013)	Nayeem et al (2014)	GGAWE	Mumford (2013)	Nayeem et al (2014)	GGAWE	Mumford (2013)	Nayeem et al (2014)	GGAWE
d_0	36.60	37.71	40.33	30.92	32.53	33.92	27.46	29.15	30.27
d_1	52.42	56.37	55.56	51.29	63.53	63.01	50.97	64.31	64.29
d_2	10.71	5.88	4.12	16.36	3.93	3.08	18.76	6.5	5.42
d_{un}	0.26	0	0.01	1.44	0	0.1	2.81	0	0.38
ATT (mins)	24.79	23.96	23.6	28.65	26.63	26.45	31.44	29.65	29.41

and Table 4. Here we can see that GGAWE produce better effect than the effects of GLS and GAWE. We have also reported $Variance$ in Table 3 for GGAWE. As the $Variance$ is very small for each of the case, it seems that there are very little differences among the metrics' value of GGAWE.

6 Conclusion

In this paper, we have proposed GGAWE which is a hybridized but novel approach. It uses the explorative feature of GLS and combines with the *elitism* characteristics of GAWE as hybridization. We have shown extended experimental results on our proposed algorithms for both benchmark data and real public bus networks. We have also shown that our experimental results are competitive with previous work of Nayeem et al (2014) and better than all other previous methods. The proposed algorithm can generate high quality solutions. We found that our GGAWE is best suited for large networks with many edges.

We are also considering other objectives for future research to this multi-objective transit network design problem such as traffic jam, performance of vehicles, alternative route, operator cost, etc.

Acknowledgements The authors would like to thank anonymous persons who are professionally involved at CSE Department in BUET. It would be really impossible without their kind support and help. The authors also would like to declare that this is a self-funded research work which has been carried out by some members of AℓEDA group in BUET.

References

Baaj MH, Mahmassani HS (1995) Hybrid route generation heuristic algorithm for the design of transit networks. Transportation Research C 3(1):31 – 50

Chakroborty P (2003) Genetic algorithms for optimal urban transit network design. Computer-Aided Civil and Infrastructure Engineering 18:184 – 200

Charkroborty P, Dwivedi T (2002) Optimal route network design for transit systems using genetic algorithms. Engineering Optimization 34(1):83 – 100

Derrible S, Kenneday C (2011) Applications of graph theory and network science to transit network design. Transport reviews 31:495 – 519

Fan L, Mumford CL (2010) A metaheuristic approach to the urban transit routing problem. Journal of Heuristics 16:353 – 372

Fan W, Machemehl RB (2008) Tabu search strategies for the public transportation network optimizations with variable transit demand. Computer Aided Civil and Infrastructure Engineering 23:502 – 520

Kepaptsoglou K, Karlaftis M (2009) Transit route network design problem. ReviewJournal of Transportation Engineering - ASCE 135(8):491 – 505

Kidwai FA (1998) Optimal design of bus transit network: A genetic algorithm based approach. Tech. rep., Indian Institute of Technology, Kanpur, India, phd. dissertation

Mandl CE (1979) Evaluation and optimization of urban public transportation network. European Journal of Operational Research 5:396 – 404

Mumford CL (2013) New heuristic and evolutionary operators for the multi-objective urban transit routing problem. IEEE Congress on Evolutionary Computation (CEC) 12:939 – 946

Nayeem MA, Rahman MK, Rahman MS (2014) Transit network design by genetic algorithm with elitism. Transportation Research C 46:30–45

Nielsen G (2005) Hitrans best practice guide 2: Public transport planning the networks. Tech. rep., European Union Interreg IIIB (North Sea Region), tech. Rep.

Nikoli M, Teodorovi D (2013) Transit network design by bee colony optimization. Expert Systems with Applications 40(15):5945 – 5955

Schoebel A (2012) Line planning in public transportation: models and methods. OR Spectrum 34:491 – 510

Voudouris C, Tsang E (1995) Guided local search. European Journal of Operational Research 113:1–18

Zhao F, Gan A (2003) Optimization of transit network to minimize transfers. Tech. rep., Lehman Center for Transportation Research, December, Florida International University, 605 Suwannee Street, MS 30, Tallahassee FL 32399-0450, tech. Rep. BD-015-02