ELSEVIER

Advances in Engineering Software 37 (2006) 351–359

ADVANCES IN ENGINEERING SOFTWARE

www.elsevier.com/locate/advengsoft

Solving the feeder bus network design problem by genetic algorithms and ant colony optimization

S.N. Kuan, H.L. Ong *, K.M. Ng

Department of Industrial and Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119620, Singapore

Received 8 February 2005; received in revised form 6 October 2005; accepted 19 October 2005
Available online 28 December 2005

Abstract

This paper proposes the design and analysis of two metaheuristics, genetic algorithms and ant colony optimization, for solving the feeder bus network design problem. A study of how these proposed heuristics perform is carried out on several randomly generated test problems to evaluate their computational efficiency and the quality of solutions obtained by them. The results are also compared to those published in the literature. Computational experiments have shown that both heuristics are comparable to the state-of-the-art algorithms such as simulated annealing and tabu search.

Keywords: Feeder bus; Metaheuristics; Genetic algorithm; Ant colony optimization

1. Introduction

In city regions where transit demand is high and widely spread, an intermodal transit system consisting of the integration of a rail line and a number of feeder bus routes connecting to the transfer stations is inevitable. The rail lines serve as a highly efficient and convenient mode of transport which is able to carry high volumes of passengers to and from the city, while the feeder bus routes serve to transport passengers from the bus stops to the rail network. The main challenge of this problem is to design an efficient feeder bus route network to integrate and coordinate the rail and bus services. A better integrated intermodal system would lead to a reduction of operating costs and an increase in revenues through maintaining shorter routes and eliminating duplicated routes by the trains and the buses, as well as a higher service quality and satisfaction level of passengers resulting from better coverage and shorter travel times with minimal delay.

Despite the lack of substantial work in the area of intermodal transit systems, there have been increased interests in the recent years on the design of a feeder bus route network serving a rail system. Generally, existing approaches can be classified into two main categories: analytic and network approaches. Analytic models require the shape of the street geometry to be prespecified, and a known demand function which represents the spatial distribution of demand in the service area. The analytic approaches include those proposed by Wirasinghe et al. [1–3], Kuah and Perl [4], Chien and Schonfeld [5], Chien and Yang [6] and Chien et al. [7]. Network models, on the other hand, do not require one to prespecify the geometry of the street network. In these network models, the service area is represented by nodes in which the passenger demand is assumed to be concentrated, and links representing the segments of the transit routes. The network approaches include those proposed by Kuah and Perl [8], Martins and Pato [9] and Shrivastav and Dhingra [10].

In this paper, we focus on solving the feeder bus network design problem (FBNDP) as presented by Kuah and Perl [8] and Martins and Pato [9]. Kuah and Perl [8] formally defined the FBNDP using a mathematical programming model. In this model, the route structure and the operating frequency are being optimized with the objective function of minimizing the sum of passenger and operator costs. Kuah and Perl [8] also proposed a heuristic method for solving the FBNDP which generalized the 'savings approach' to incorporate operating frequency. Martins and Pato [9] extended the work of Kuah and Perl [8] by designing better heuristics to improve the previously obtained solutions. They built the initial solution using the sequential savings approach proposed by Kuah and Perl [8], with the initial solution being improved using heuristic procedures.

A real-world FBNDP usually cannot be solved to optimality using existing exact optimization methods due to its size and

* Corresponding author.

0965-9978/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2005.10.003

complexity. The main objective of this paper is to design better algorithms to the FBNDP by exploring the use of metaheuristics to solve the FBNDP. In Kuan et al. [11], the results of applying simulated annealing and tabu search to the FBNDP are described. Here, we describe how genetic algorithms and ant colony colonization could also be applied to solve the FBNDP.

The rest of the paper is organized as follows: Section 2 presents a brief description of the FBNDP. In Section 3, we describe our approach in solving the FBNDP by explaining how an initial feasible solution was generated and how the two metaheuristics were being applied to solve the FBNDP. The computational experiments and results are shown in Section 4. Lastly, the conclusions of our findings are summarized in Section 5.

2. Problem description

In the FBNDP, the feeder bus network serves to carry the passengers from the bus stops to the various stations in an existing rail network. The focus of the FBNDP is on the design of a set of feeder bus routes and the determination of the operating frequency on each route, such that the objective function of the sum of operator and user costs is minimized. The operator cost is related to the total length travelled by the buses, while the user cost is a function of total passenger travel time including the waiting time and riding time on both the bus and the train.

An example of a network model representing the FBNDP is shown in Fig. 1. The network models of the FBNDP use two types of nodes, rail nodes and bus nodes, to represent the railway stations in a rail transport system and the bus stops, respectively, within a given service area. The rail transport network is assumed to be fixed, that is, defined in advance and not subject to changes, and is represented as links joining the rail nodes shown in the left diagram of Fig. 1. The location of bus stops and hourly demands of passengers at each bus stop are also prespecified. The travel cost in the rail system between each pair of railway stations, the distance between each pair of bus stops and between every bus stop and every railway station, the maximum bus route length, the number of bus routes, the capacity and operating speed of the fleet of buses over the planning period are also given. The FBNDP involves linking bus nodes to rail nodes, in which these bus links represent feeder bus route segments, as shown in the right diagram of Fig. 1.

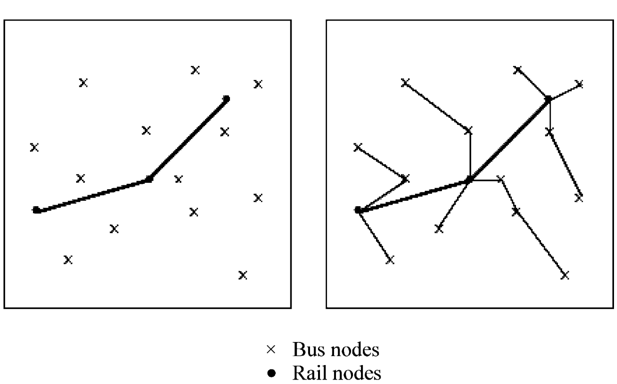

Fig. 1. Feeder bus network design problem.

The FBNDP can be formulated as a hierarchical transportation network design problem with side constraints (see e.g. [12,13]), where the primary path represents the rail line and the secondary paths represent the feeder bus routes. Kuah and Perl [8], on the other hand, formulated a nonlinear integer programming model for the many-to-one (multiple origins, single destination) FBNDP. In this paper, the many-to-one FBNDP is considered under the assumptions: (1) each bus stop is served by one feeder bus route only; (2) each feeder-bus route is linked to exactly one station; (3) all buses have standard operating speeds and capacities; (4) the feeder-bus is assumed to halt at all stops on its route. Peak-period work trips to the Central Business District in the morning exhibit such a pattern. In this case, all passengers share a common destination, identified as the central city station. Passengers gathered at bus stops in the service area wish to access this destination by first travelling by bus to any of the stations and then proceeding to the city centre by train.

3. Proposed methodology

In this section, we present our approach for solving the FBNDP. The approach involves building an initial feasible solution using the route construction heuristic and then using metaheuristics for improving the initial solution. Metaheuristics arise in the recent development for solving hard combinatorial optimization problems and they incorporate concepts derived from artificial intelligence, biological evolution, mathematical and physical sciences to improve their performance. Examples of metaheuristics include simulated annealing (SA), tabu search (TS), genetic algorithm (GA) and ant colony optimization (ACO) (see [7,11,14–21] for details). In this research, GA and ACO have been developed and applied to solve the FBNDP.

To improve the solutions of the FBNDP, some heuristic procedures for the travelling salesman problem (TSP), such as that from Lin and Kernighan [22] can be applied after an initial solution has been constructed and whenever a new best solution has been found. The purpose is to reorder the stop sequence in every single route to reduce the route distance, which in turn reduces the total cost, before the local search is continued.

3.1. Generating the initial solution

The initial solution is constructed on a stochastic basis. We give a brief description of the construction method with emphasis on route length feasibility as other constraints on the routes could be handled in a similar way.

Each route is built sequentially as follows: first, a station is selected at random; then, randomly chosen stops are added to the route linking to this station. Each time a stop is added, the length of the route is checked. When it exceeds the maximum route length, the current route is terminated and a new route will be built in the same way. The procedure continues until all the stops have been included in the routes. However, random choice of stops without any restriction may cause too many bad selections, resulting in a poor initial solution. To resolve this issue, the concept of *delimiter* proposed by Van Breedam [14] can be applied. A delimiter is a restriction in terms of the distance between the stations and the stops to be selected at random. The initial delimiter is calculated as follows:

For each stop i, determine the distance of its nearest station j:

$$\mathrm{Dist}_i = \min_j d_{ij} \tag{1}$$

The *initial delimiter* D_I is equal to the maximum of the set of minimum distances calculated:

$$D_\mathrm{I} = \max_i \mathrm{Dist}_i \tag{2}$$

Thus, the distance between the random stops and the stations selected must be less than or equal to D_I, or else a new stop will be generated. In this way, the delimiter will prevent linking a stop and a station that are too far apart such that they are unlikely to form a good solution.

For a given number of bus routes, K, it is possible that an initial solution could not be found in the situation that both the given bus routes and the maximum route length are too small. To deal with this situation, we can relax the maximum route length constraint by imposing a penalty cost in the objective function if it is being violated. Thus, if we find that after all the K routes have been generated and yet there are some stops that could not be included in these routes due to the maximum route length constraint, we will insert these stops to the routes with the least penalty cost possible.

3.2. Solution improvement using genetic algorithm

GA, developed by Holland [15], is an intelligent search heuristic inspired by Darwin's theory about evolution. According to Darwin's evolution theory, only the best-fit individuals should survive and create new offsprings, whereas the least-fit individuals will be eliminated. The GA simulates this behaviour by maintaining a population of solutions with a fitness value associated with each solution. Each solution corresponds to a chromosome. By means of some selection techniques and reproduction operators, a new population of offsprings with better overall fitness is obtained and it replaces the old population. This cycle is repeated until a terminating condition (e.g. number of generations) is satisfied, or a satisfactory solution is found.

3.2.1. Representation

The representation of a FBNDP solution used here is an integer string of variable length, depending on the number of routes formed. Each string consists of several substrings, with each substring being a sequence of stops in a route ended by the station that these stops are attached to. For example, assuming that 1–10 are stop indices and 11–13 are station indices, a solution of this form:

Route 1: 1 2 3 4 **11**
Route 2: 5 6 7 **12**
Route 3: 8 9 10 **13**

is represented as

1 2 3 4 **11** 5 6 7 **12** 8 9 10 **13**.

This representation is unique and one chromosome can only be decoded to one solution. Decoding the chromosome into route configurations is simple because every station in the string indicates the end of each route.

3.2.2. Initial population and fitness

Initial solutions in the population are generated at random. The fitness value of a solution is measured by its objective value, and no penalty function has been included. This is because even though the initial solutions may be infeasible, feasible solutions are always being ensured during the reproduction process.

3.2.3. Selection

A tournament selection mechanism, adapted from Tan et al. [16,17], is used to select parents for reproduction. In this tournament selection, the population of n solutions is duplicated to get two identical copies. These two populations are arbitrarily ranked. For population P_1, each pair of adjacent chromosomes (with indices $2i$ and $2i+1$) is compared. The one with smaller fitness value qualifies to be a potential parent. After comparing all the $n/2$ pairs in P_1, $n/2$ 'fathers' f_i, are obtained. This process is repeated for population P_2 to get $n/2$ 'mothers' m_i.

3.2.4. Reproduction

The reproduction process of GA consists of crossover and mutation. In our implementation, a new crossover technique is designed which is able to ensure the validity of the offsprings, as well as to preserve the edges in the parents. The crossover operation starts by choosing one parent string and then listing the edges of both parent strings that are next to the stop represented by the first gene of the chosen parent string. The shortest edge between the current stop and its edges is chosen as the next gene in the string. The process is continued until the feasibility of the current route with respect to route length is violated. When this occurs, the next stop in the chosen parent string that has yet to be included in the offspring chromosome will be selected as the first stop of the next route. Also, when there are no more unvisited edges to be chosen, the current route is terminated and the same procedure is continued for the next route. As an example to illustrate the crossover operation, we define the following parents.

Parent 1: 1 2 3 4 **11** 5 6 7 **12** 8 9 10 **13**
Parent 2: 10 7 4 **11** 1 3 **12** 5 9 **12** 2 6 8 **13**

Suppose parent 1 is being chosen to generate the first offspring. Starting from the first gene in parent 1, the first gene in offspring 1 is chosen to be stop 1. Now, the edges next to stop 1 are stops 2 and 3 from parent 1 and parent 2, respectively. Assuming that $d_{13} < d_{12}$, stop 3 is chosen as the next gene in the offspring. Similarly, the edges next to stop 3 are stops 2 and 4 from parent 1 (stop 1 has already been included). Assuming that $d_{34} < d_{32}$, stop 4 is then chosen as the next gene. The process is continued until the feasibility of the current route is violated. Then, station 11, which is attached to the first stop in this route (stop 1), is put next on the string. Scanning through the parent 1 string, the next unvisited stop is stop 2 and it becomes the first stop of the next route. This process continues, resulting in the following offspring.

Offspring 1: 1 3 4 **11** 2 … **11**…

The crossover then continues with choosing parent 2 and generating the second offspring in a similar manner.

The advantage of this crossover method is that besides being able to preserve most of the edges in the parent chromosomes, it can also inherit the best qualities from the parents as well, while ensuring the validity of the new offsprings produced.

The mutation operation is performed on an offspring by first scanning through its string representation and choosing the first two stops based on a prespecified mutation probability. These two stops are then swapped in the operation.

3.2.5. Elitism

Elitism is achieved by substituting the worst 4% chromosomes in the offspring with the best 4% in the parents as proposed by Tan et al. [16,17].

3.2.6. Parameters of GA

The crossover probability (P_c), the mutation probability (P_m), the population size (*pop_size*), and the number of generations (*n_gen*) are all determined experimentally.

3.2.7. GA procedure

Step 1 Generate random initial population of size, *pop_size*.
Step 2 Evaluate the fitness value for each chromosome in the population.
Step 3 Create a new population by repeating the following steps until the new population is completed:
  Select two parent chromosomes from a population by tournament selection.
  With a crossover probability P_c, perform crossover on the parents to form new offsprings. If no crossover was performed, the offspring is an exact copy of the parent.
  With a mutation probability P_m, mutate new offspring at random.
  Place the new offspring in a new population.
  Replace the worse 4% solutions in the new population with the best 4% in the parent's population.
Step 4 Update the old population with the newly generated population.
Step 5 If total number of generations, *n_gen*, is reached, improve the best solution in current population by some TSP heuristics; else, go to step 2.
Step 6 Return the best solution found.

3.3. Solution improvement using ant colony optimization

ACO is a recently proposed metaheuristic approach for solving combinatorial optimization problems. The first ACO algorithm proposed in the literature was ant system (AS) by Dorigo et al. [18] and was first applied to the TSP. ACO is inspired by the behaviour of real ants which lay pheromone trails on the paths they have travelled so as to direct the search of future ants foraging for food. Despite being a rather recent metaheuristic, ACO algorithms have already been applied to a large number of different combinatorial optimization problems, including the vehicle routing problem (VRP). In analogy to the biological example, ACO is based on indirect communication of a set of artificial ants mediated by artificial pheromone trails. Since FBNDP and VRP are closely related, the proposed method is highly influenced by the application of the ant system to VRP as described in Bullnheimer et al. [19,20].

3.3.1. Ant activity

Firstly, each individual ant constructs a feasible solution on its own using a probabilistic decision rule that makes use of pheromone trails τ and heuristic information η. The amount of pheromone accumulated on each link between the stops and the stations reflect the ants' acquired search experience. When the pheromone level on a particular link is high, it means that this link should exist in order to achieve a good solution. Pheromone trails change dynamically with run-time. Heuristic values represent a priori information about the problem and is fixed throughout the search using the reciprocal of the distances between the stops and the stations. The construction procedure of each ant k is as follows: an ant is assigned to a stop initially. Due to the multi-station nature of the problem, a decision has to be made to determine which station this stop has to link to. Here, a new probability function is defined to measure how desirable it is to link stop i to station st:

$$p_{i,\mathrm{st}}^{k} = \frac{[\tau_{i,\mathrm{st}}]^{\alpha}[\eta_{i,\mathrm{st}}]^{\beta}}{\sum_{H\in\Omega}[\tau_{i,H}]^{\alpha}[\eta_{i,H}]^{\beta}} \quad \text{if st} \in \Omega_{\mathrm{st}} \tag{3}$$

where

d_i^{st} Distance between stop i and station st
$\tau_{i,\mathrm{st}}$ Intensity of pheromone trail between stop i and station st
$\eta_{i,\mathrm{st}} = 1/d_i^{\mathrm{st}}$ Visibility of station st from stop i (available heuristic information)
α Parameter to regulate the influence of pheromone trail, $\tau_{i,\mathrm{st}}$
β Parameter to regulate the influence of heuristic information, $\eta_{i,\mathrm{st}}$

Ω_{st} The set of stations which stop i can be assigned to

After a station has been chosen according to the above probability function, a route will be constructed by successively choosing stops to be visited, until the feasibility of the route becomes invalid. When this happens, the current route will be ended and the next unvisited stop will be selected. The process continues until all stops are visited. The probability of visiting stop j from stop i is similar to that for the TSP:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h\in\Omega} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta} \quad \text{if } j \in \Omega \tag{4}$$

where

d_{ij} Distance between stops i and j
τ_{ij} Intensity of pheromone trail between stops i and j
$\eta_{ij}=1/d_{ij}$ Visibility of stop j from stop i (available heuristic information)
α Parameter to regulate the influence of pheromone trail, τ_{ij}
β Parameter to regulate the influence of heuristic information, η_{ij}
Ω The set of stops which ant k has not yet visited

However, in the VRP (and also FBNDP), not only is the relative location of two cities important but their relative location to the depot is also essential. This information can be captured by the *savings* proposed by Clarke and Wright [23], which measures the favourability of linking stop j after stop i with respect to the station which stop i is assigned to. Thus,

$$\eta_{ij} = s_{ij}^{st} = d_i^{st} + d_j^{st} - d_{ij} \tag{5}$$

where

d_i^{st} Distance from station st to stop i
s_{ij}^{st} Savings from linking stops i and j to a route assigned to station st

Tillman and Cain [24] discovered that this savings equation is only applicable to the case of a single station. With their concept of 'modified distance' for multiple stations, the modified distance for each stop i from each station st is computed as:

$$\tilde{d}_i^{st} = \min_m d_i^m - \left(d_i^{st} - \min_m d_i^m\right) \tag{6}$$

where $\min_m d_i^m$ is the distance between stop i and its nearest station m.

The modified savings from linking stops i and j to a route assigned to station st is then

$$\eta_{ij} = \tilde{s}_{ij}^{st} = \tilde{d}_i^{st} + \tilde{d}_j^{st} - d_{ij} \tag{7}$$

where

$\tilde{d}_i^{st}$ Modified distance from station st to stop i
$\tilde{s}_{ij}^{st}$ Modified savings from linking stops i and j to a route assigned to station st

3.3.2. Trail update

After all artificial ants have constructed a feasible solution, the pheromone trails are updated depending on the quality of their objective function value, the total cost TC^k. A combined elitist and ranking strategy is used and the trail intensities are updated as:

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \sum_{\mu=1}^{\sigma-1} \Delta\tau_{ij}^{\mu} + \Delta\tau_{ij}^{*} \tag{8}$$

where

$\rho \in [0,1]$ Pheromone trail evaporation rate
σ Number of elitist ants
μ Ranking index
$\Delta\tau_{ij}^{\mu}$ Increase of trail level on edge (i, j) caused by the μth best ants
$\Delta\tau_{ij}^{*}$ Increase of trail level on edge (i, j) caused by the elitist ants

By analogy to nature, part of the pheromone evaporates. The first term calculates how much the existing pheromone trails evaporates. The second term ranks the ants according to the quality of the solutions that they generated and the contribution of each ant to the pheromone trail level update is weighted according to the rank μ of the ant (Bullnheimer et al. [21]). Also, only elitist ants are allowed to deposit extra pheromone. The pheromone trail is increased by $\Delta\tau_{ij}^{\mu} = (\sigma-\mu)Q/TC^{\mu}$ if edge (i, j) is used by the μth best ant with total cost TC^{μ} and zero otherwise, where Q is the total quantity of pheromone laid by each ant per iteration. The third term represents the elitist strategy introduced by Dorigo et al. [18] of giving extra emphasis on the paths used by the ant that has built the best solution. For all σ elitist ants, if the edges (i, j) travelled by them also belong to the best solution found so far (with total cost TC^*), the trail intensity is increased by an amount $\Delta\tau_{ij}^{*} = \sigma Q/TC^*$ (zero otherwise).

When solving a problem with large neighbourhoods, solution construction by an ant without restriction is time consuming and the probability that many ants visit the same state is small. Hence, during solution construction when an ant tries to move from one stop to another, only its *n_NB* nearest neighbours are considered. If the nearest stops have all been visited, the current route is ended and a new route has to be constructed. Based on studies for VRP (Bullnheimer et al. [19]), ACO yields good results when the number of ants is set equal to the number of cities and each ant starts its tour from another city. Thus, for the FBNDP, the number of ants is set equal to the number of stops and each ant is placed at each stop during initialization. The number of elitist ants σ and total number of iterations *max_Iter* are determined experimentally.

3.3.3. ACO procedure

Step 1 Initialize.
Step 2 For *max_Iter* iterations:
For each ant $k=1$ to m, generate a new solution using Eqs. (3) and (4). Improve the solution by solving TSP for each route using 2-opt.
Calculate the total cost for all solutions and rank them accordingly.
Update the pheromone trails using Eq. (8).
If a new best solution is found, perform TSP for each route using 2-opt.
Step 3: Return the best solution found.

4. Computational experiments and results

All the proposed algorithms are coded in Visual C++ and run in a Pentium 900 Mhz PC under Windows XP.

4.1. Benchmark and test problems

The benchmark problem is taken from Kuah and Perl [8], which includes 55 stops and 4 stations covering a service area of 2×2.5 mile^2. The bus-stop density is 11 stops per square mile, with hourly demand density of approximately 2200 passengers per square mile. The values for the other parameters are shown in Table 1.

Besides the benchmark problem, 20 test problems are also randomly generated that vary in size and structure. The problem size is determined by (1) size of service area, (2) number of stations in service area, and (3) density of bus stops in service area. The two sizes of the service area are square areas of 3×3 miles^2 (small) and 5×5 miles^2 (large), with 10 problems generated for each size. The number of stations in the service area is set to 4 and 7 stations for small and large problems, respectively. The density of bus stops in the service area is set between 4 and 6 stops/miles^2. The bus stops are randomly and uniformly located in the service area.

For all these 20 randomly generated problems, the demand at each bus stop is fixed at 150 passengers per hour. The maximum allowable route lengths are 2.5 and 4 miles for the small and large problems, respectively. The values for the parameters are the same as the base problem, except for the maximum allowable route lengths.

Table 1
Values for parameters of benchmark problem

Descriptions	Units	Value
Bus operating-cost, λ_0	\$/vehicle-mile	3.0
Rail user-cost, C_{js}/mile	\$/passenger-mile	0.15
Riding-time cost, λ_r	\$/passenger-hour	4.0
Waiting-time cost, λ_w	\$/passenger-hour	8.0
Maximum allowable route length, D_k	Mile	2.5
Bus capacity, c	Seat	50
Number of bus routes, K	Bus route	16
Bus operating-speed, U	Mile/hour	20
Maximum available seat-hours, ASH	Seat-hours	5500

Table 2
Comparison between our metaheuristics and the best-known results

Heuristic approach	Total cost
Literature	
Savings heuristic[a]	6846
SA[b]	6519
TS with intensification[b]	6338
Our metaheuristics	
GA	6412
ACO	6535

[a] From Kuah and Perl [8].
[b] From Kuan et al. [11].

4.2. Computational results

The parameters are chosen experimentally to ensure a compromise between the running time and the solution quality. This is done by varying the parameter values over an estimated range and doing trial runs for all the problems generated. For GA, the parameters set are $n\_gen=10(I+J)$ and pop_size = 100 (I is the number of stops and J is the number of stations). Other parameters set are $P_c=0.8$ and $P_m=0.06$. For ACO, the parameters set are $max\_Iter=1.5(I+J)$ and $n\_NB=20\%$ of I. Other parameters set are: $\alpha=1$, $\beta=5$, $\rho=0.15$, $\sigma=6$ and $Q=100$.

4.2.1. Comparison with best-known results

A comparison between the metaheuristics that we have proposed and the best-known results in literature for the benchmark problem is shown in Table 2. It can be seen that GA is able to obtain a solution close to the best-known solution obtained by TS with intensification, while ACO obtains a solution comparable to SA.

4.2.2. Comparison of results of competing metaheuristics

For each of the problems, a total of 20 runs are performed and the average total costs, the best total cost and the average computational times are recorded. Table 3 shows the comparison of the best total costs obtained by each metaheuristic for the benchmark (base) and the 20 test problems. In Table 4, we have also computed the average percentage deviation based on the average total costs from the best-known solution achieved, with the corresponding graph being shown in Fig. 2. From these tables, we are able to summarize the comparison between these metaheuristics as follows:

(1) In terms of the best total cost, GA and ACO are comparable to TS with intensification, and outperform SA in many instances. For problem 14, ACO obtains the best result.

Table 3
Comparison of best total costs

Problem	SA	TS with intensification	GA	ACO
Base	6520	6338	6412	6535
1	4462	4432	4447	4478
2	4581	4467	4546	4690
3	4686	4521	4576	4644
4	4413	4259	4381	4639
5	4195	4085	4111	4234
6	6749	6660	6786	6839
7	7576	7459	7599	7566
8	6773	6633	6707	6727
9	6113	5857	5992	5913
10	5951	5854	5904	5980
11	17297	17202	17503	17327
12	16636	16671	16768	16815
13	14548	14406	14611	14461
14	16326	16159	16547	16030
15	13938	13304	13684	14225
16	21134	21092	21316	21166
17	20001	19843	20634	20680
18	17944	17628	18178	18010
19	20299	20280	20964	21048
20	17270	16681	17065	17363

(2) In terms of average total cost, GA has an average of 3.1% deviation from the best-known solution, which is only about 1.1% worse than TS with intensification. ACO has an average of 6.3% deviation from the best-known solution and is comparable to SA.

The results are encouraging because this is the first reported attempt to implement ACO for the FBNDP and they show ACO's competence when compared to the state-of-the-art heuristics such as SA.

4.2.3. Comparison of computational times

Table 5 summarizes the comparison of the average computational times obtained by each metaheuristic for the base and the 20 test problems. It can be seen that GA and ACO clock similar amount of time, and they generally require more time than SA but less time than TS with intensification. GA requires more time to perform the crossover because validity checks have to be done repeatedly to ensure that the maximum route length is not exceeded. ACO spends more time on updating the pheromone trails by checking for elitist ants and desirable edges.

4.3. Recommendations from computational comparisons

The results show that TS with intensification has the best performance in terms of the quality of the solution obtained, regardless of the nature of the problem. However, it is also computationally intensive as reflected in the large computation time needed. Thus, without any requirement on the run time needed for solving the FBNDP, TS with intensification is the ideal solution method. When it is essential that the FBNDP be solved in the smallest computation time possible, SA would be the best candidate. However, the quality of the solutions obtained by SA is not as good as that of the other heuristics, such as GA and ACO. Hence, GA and ACO would be preferred if solutions of good quality were to be obtained within reasonable computation times. The computational results also show that ACO is more effective for problems with large size

Table 4
Average percentage deviation from best solution obtained

Problem	Best known	SA (%)	TS with intensification (%)	GA (%)	ACO (%)
Base	6338	8.0	2.3	2.9	6.0
1	4432	5.7	1.3	2.0	4.9
2	4467	9.9	2.6	3.9	9.0
3	4521	8.2	2.6	4.0	6.0
4	4259	11.6	2.9	4.9	13.7
5	4085	7.6	1.0	1.8	7.8
6	6660	5.9	1.9	3.8	5.7
7	7459	4.9	1.5	2.2	5.4
8	6633	6.8	2.4	2.4	5.6
9	5857	8.5	2.4	3.2	6.3
10	5854	6.3	2.2	3.2	5.7
11	17202	3.0	1.8	2.5	3.6
12	16636	3.2	1.4	2.3	4.0
13	14406	2.8	1.8	2.2	3.3
14	16030	5.5	3.6	3.7	5.6
15	13304	8.9	1.8	3.1	8.6
16	21092	2.2	1.8	2.5	2.8
17	19843	4.0	2.0	4.9	8.3
18	17628	4.4	1.7	3.9	4.5
19	20280	5.2	1.8	4.0	6.5
20	16681	6.3	2.2	3.0	8.7
Average% deviation		6.1	2.0	3.2	6.3

Percentage deviation = (Average total cost − best known result)/best known result × 100%.

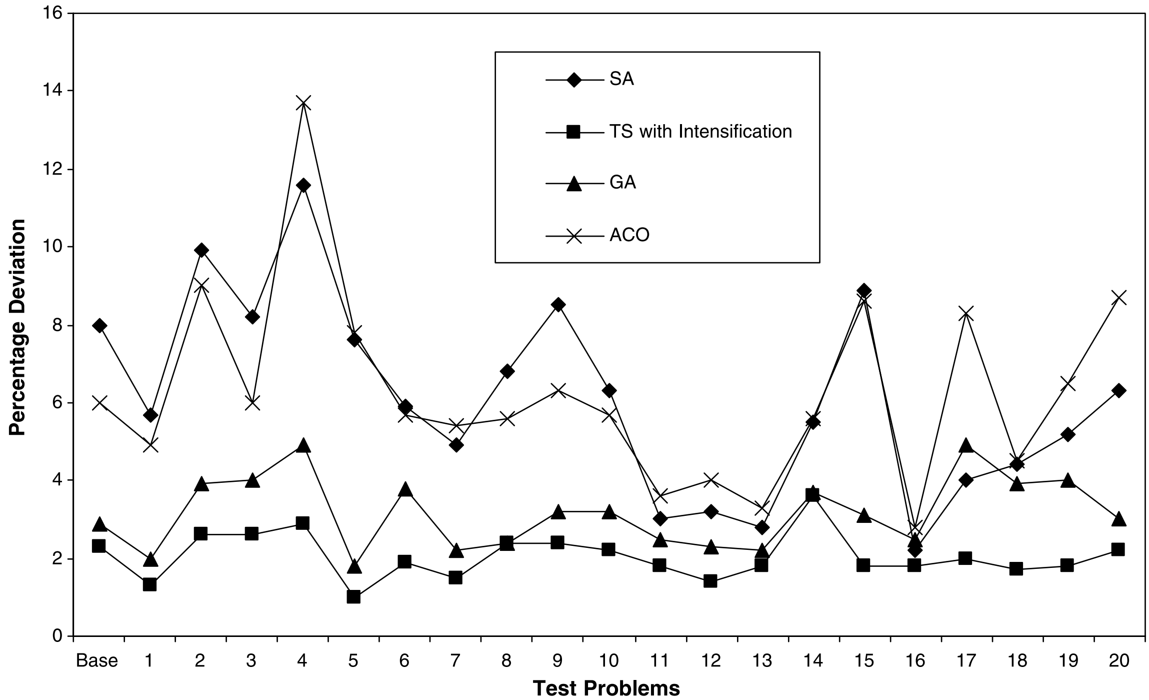

Fig. 2. Comparison of percentage deviation.

and it is even able to obtain the best solution when compared with the other metaheuristics for one of the test problems with large size, while GA appears to be favoured for problems with smaller size.

5. Summary and conclusion

In this paper, we have presented our methodology for solving the FBNDP. The FBNDP focuses on the design of a set of feeder bus routes and the determination of the operating frequency on each route, such that the objective function of the sum of operator and user costs is minimized. The main objective of our research is to develop better algorithms that are able to give a good solution for the FBNDP in a reasonable amount of computation time. The metaheuristics GA and ACO are developed and applied to solve the FBNDP. For GA, a new crossover operator is designed which is able to preserve most of the edges in the parent solutions, as well as inherit the best qualities from the parents while ensuring the validity of the new offsprings produced. ACO is designed according to the multi-station nature of the problem by calculating two probability functions. A combined elitist and ranking strategy is used for pheromone update. Computational results are compared to those published in the literature for the base problem. A comparative study is also carried out on 20 test problems generated at random to further compare the performance of the metaheuristics in terms of computational efficiency and solution quality. GA is comparable to TS with intensification, while ACO is able to give satisfactory results and surpasses some of the SA results for certain type of problems.

In conclusion, this research has two major contributions to the area of feeder bus route design problem. It is among the first to solve the complex FBNDP using ant colony optimization. In addition, it is also the first to present a comparison of the performance of diverse metaheuristics for the FBNDP with some recommendations on their usage.

Several directions for future research in solving the FBNDP are possible. For GA and ACO, the encouraging results indicate that there is potential for further improvement in their procedures, such as using the 3-opt procedure to optimize each route formed when applying GA and ACO, so that their

Table 5
Comparison of average computational times in seconds

Problem	SA	TS with intensification	GA	ACO
Base	4.6	13.8	10.1	9.0
1	5.1	4.1	5.5	5.0
2	6.9	4.3	4.7	7.4
3	5.8	4.0	5.8	6.0
4	5.1	3.5	5.7	6.1
5	4.3	4.0	4.8	4.7
6	6.5	11.5	9.5	9.6
7	7.7	10.3	8.7	10.9
8	6.6	8.8	7.7	10.0
9	6.2	12.1	10.3	8.5
10	6.2	9.8	9.3	9.2
11	89.9	147.5	127.5	113.6
12	86.0	137.4	105.6	92.8
13	87.1	84.9	76.8	103.8
14	88.2	120.0	127.3	109.8
15	78.9	93.0	87.4	97.3
16	108.7	250.0	167.4	131.8
17	109.5	196.5	148.9	120.9
18	91.3	150.5	127.8	107.2
19	111.9	223.3	163.2	139.8
20	88.6	152.6	144.2	110.4

performance could be as good as that of TS with intensification. The computational comparisons among the various metaheuristics also indicate that for TS with intensification, the reduction of computation time without sacrificing solution quality through more efficient diversification strategies would be a valuable area of research, while for SA, the choice of better and more sophisticated neighbourhood structures to improve the solution quality could be looked at. Besides exploring on how the design of metaheuristics can be further improved, it is also possible to consider modifying the objective function and constraints of the problem to consider more scenarios in future research work. Other extensions include considering the elastic nature of passenger demands and running computational experiments for larger problem sizes.

References

[1] Wirasinghe SC. Assignment of buses in a coordinated rail and bus transit system. Proceedings of the seventh international symposium on transportation and traffic flow theory; 1977. p. 673–96.

[2] Wirasinghe SC, Hurdle VF, Newell GF. Optimal parameters for a coordinated rail and bus transit system. Transport Sci 1977;11(4):359–74.

[3] Wirasinghe SC. Nearly optimal parameters for a rail/feeder-bus system on a rectangular grid. Transport Res Part A 1980;14(1):33–40.

[4] Kuah GK, Perl J. Optimization of feeder bus routes and bus-stop spacing. J Transport Eng 1988;114(3):341–54.

[5] Chien S, Schonfeld P. Joint optimization of a rail transit line and its feeder bus system. J Adv Transport 1998;32(3):253–84.

[6] Chien S, Yang Z. Optimal feeder bus routes on irregular street networks. J Adv Transport 2000;34(2):213–48.

[7] Chien S, Yang Z, Hou E. Genetic algorithm approach for transit route planning and design. J Transport Eng 2001;127(3):200–7.

[8] Kuah GK, Perl J. The feeder-bus network-design problem. J Oper Res Soc 1989;40(8):751–67.

[9] Martins CL, Pato MV. Search strategies for the feeder bus network design problem. Eur J Oper Res 1998;106:425–40.

[10] Shrivastav P, Dhingra SL. Development of feeder routes for suburban railway stations using heuristic approach. J Transport Eng 2001;127(4):334–41.

[11] Kuan SN, Ong HL, Ng KM. Applying metaheuristics to feeder bus network design problem. Asia Pac J Oper Res 2004;21(4):543–60.

[12] Lin C-C, Chen S-H. The hierarchical network design problem for time-definite express common carriers. Transport Res Part B 2004;38(3):271–83.

[13] Current JR. The design of a hierarchical transportation network with transshipment facilities. Transport Sci 1988;22(4):270–7.

[14] Van Breedam A. Comparing descent heuristics and metaheuristics for the vehicle routing problem. Comput Oper Res 2001;28:289–315.

[15] Holland JH. Adaptation in natural and artificial systems. Ann Arbor: The University of Michigan Press; 1975.

[16] Tan KC, Lee LH, Zhu QL, Ou K. Heuristic methods for vehicle routing problem with time windows. Artif Intell Eng 2001;15:281–95.

[17] Tan KC, Lee LH, Ou K. Artificial intelligence heuristics in solving vehicle routing problems with time window constraints. Eng Appl Artif Intell 2001;14:825–37.

[18] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B 1996;26(1):29–41.

[19] Bullnheimer B, Hartl RF, Strauss C. Applying the ant system to the vehicle routing problem. In: Voss S, Martello S, Osman IH, Roucairol C, editors. Metaheuristics: advances and trends in local search paradigms for optimization. Dordrecht: Kluwer; 1999. p. 285–96.

[20] Bullnheimer B, Hartl RF, Strauss C. An improved ant system algorithm for the vehicle routing problem. Ann Oper Res 1999;89:319–28.

[21] Bullnheimer B, Hartl RF, Strauss C. A new rank-based version of the ant system: a computational study. Cent Eur J Oper Res Econ 1999;7(1):25–38.

[22] Lin S, Kernighan BW. An effective heuristic algorithm for the traveling-salesman problem. Oper Res 1973;21(2):498–516.

[23] Clark G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Oper Res 1964;12:568–81.

[24] Tillman FA, Cain TM. An upper bound algorithm for the single and multiple terminal delivery problem. Manage Sci 1972;18:664–82.