Applied Soft Computing 21 (2014) 654–676

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Solving the Urban Transit Routing Problem using a particle swarm optimization based algorithm

Panagiotis N. Kechagiopoulos[a], Grigorios N. Beligiannis[a,b,*]

[a] Hellenic Open University, Parodos Aristotelous 18, GR-26335 Patras, Greece
[b] Department of Business Administration of Food and Agricultural Enterprises, University of Patras, G. Seferi 2, GR-30100 Agrinio, Greece

ARTICLE INFO

Article history:
Received 6 February 2013
Received in revised form 16 February 2014
Accepted 4 April 2014
Available online 18 April 2014

Keywords:
Particle swarm optimization
Vehicle routing
Urban Transit Routing Problem
Mandl's benchmark problem
Real world application

ABSTRACT

The Urban Transit Routing Problem (UTRP) comprises an NP-hard problem that deals with the construction of route networks for public transit networks. It is a highly complex and multiply constrained problem, in which the assessment of candidate route networks can be both time consuming and challenging. Except for that, a multitude of potential solutions are usually rejected due to infeasibility. Because of this difficulty, soft computing algorithms can be very effective for its efficient solution. The success of these methods, however, depends mainly on the quality of the representation of candidate solutions, on the efficiency of the initialization procedure and on the suitability of the modification operators used.

An optimization algorithm, based on particle swarm optimization, is designed and presented in the current contribution, aiming at the efficient solution of UTRP. Apart from the development of the optimization algorithm, emphasis is also given on appropriate representation of candidate solutions, the route networks in other words, and the respective evaluation procedure. The latter procedure considers not only the quality of service offered to each passenger, but also the costs of the operator. Results are compared on the basis of Mandl's benchmark problem of a Swiss bus network, which is probably the only widely investigated and accepted benchmark problem in the relevant literature. Comparison of the obtained results with other results published in the literature shows that the performance of the proposed soft computing algorithm is quite competitive compared to existing techniques.

1. Introduction

Nowadays, the urban transportation system comprises a major characteristic of the social, economic and physical framework of each urban area. The design of effective urban transportation systems has become a vital problem due to the rational development of modern cities and concerns about environmental issues and pollution. In respective bibliography, urban transportation systems are divided into two basic categories [1]: (i) public transportation networks and (ii) private transportation networks. Private transportation networks have many advantages compared to the public ones [2]:

- Passengers of private transportation networks do not usually have to wait for a vehicle at the beginning of their journey. Also, they do not have to change vehicles through their trip. On the other hand, passengers of public transportation networks usually have to both wait for a vehicle at the beginning of their journey and change vehicles through their trip, which, of course, increases their waiting time.
- Private transportation road networks are most of the time better developed than public transport networks. This means that private transportation road networks have usually more nodes and more links between them. As a result, people using private transportation networks have to walk less than people using public ones.
- A trip using a private transportation network is much more convenient compared to traveling using a public transportation one.

Although most of the people believe that using private cars is the faster, most comfortable and convenient option, there are many negative issues that have to be thought of. The excessive growth in the number of private cars, traveling on urban road networks, has caused extreme traffic jams and as a result increased travel time for people that travel within urban areas using their own vehicles. Moreover, if more cars are traveling on urban road networks more air pollution and noise as far as more energy consumption and

[*] Corresponding author at: G. Seferi 2, GR-30100 Agrinio, Greece. Tel.: +30 2641074194; fax: +30 2641074179.
E-mail addresses: gbeligia@upatras.gr, gbeligia@uwg.gr (G.N. Beligiannis).

http://dx.doi.org/10.1016/j.asoc.2014.04.005
1568-4946/© 2014 Elsevier B.V. All rights reserved.

accidents will take place. Today, due to all these issues more and more people throughout the world recognize the importance of having efficient urban public transport systems [52]. In general, the public transportation system includes many different transportation means such as buses, trams, trolleys, trains and underground or metro services. Negative effects of private transportation networks can be substantially reduced by having efficient public transportation systems [53]. This will definitely happen because if more people travel with fewer transportation means, it is for certain that traffic congestion, fuel consumption, accidents and pollution will be reduced [54]. Additionally, effective public transportation systems can also invigorate the shopping center of a city by (i) eliminating traffic jams in pedestrian shopping areas, (ii) reducing the need for parking spaces inside city center, (iii) promoting retail sales and (iv) creating new job opportunities.

However, public transportation systems in most cities have suffered underfunding during the previous decades [55,56]. Consequently, most people are not willing to give up the privacy and comfort of using their own vehicles and prefer to tolerate traffic jams and parking difficulties instead of using the public transportation system which they thought as unreliable and inconvenient. As a result, public transportation systems should be definitely improved, by providing better travel services and inform the public about them, in order to convince more people to travel with it instead of using their own car [57].

It is for certain that bus services comprise the major component of a public transportation system. The most important features of bus services, that characterize their quality, are minimizing waiting and in-vehicle travel times and avoiding the need to change vehicles en route where possible. In order bus services to serve adequately these two major characteristics the operation of frequent and reliable services is needed. However, an ideal case, where a truly efficient bus service, satisfying all customers needs, will keep all operator costs in check, is really very difficult to achieve [3]. Operator costs mainly refer to the total number of busses, the total bus running distances and the total operation hours. In most countries, it is usually the bus companies that are responsible for designing the exact bus routes and schedules for an urban area. However, there also cases, for example China, where the local governments are responsible for the determination of all bus routes and schedules [4].

It is obvious that the main concern of a bus company is maximizing its profits while on the other hand the main concern of a local government is to fulfill all needs of the traveling public [58]. In both cases, a satisfactory transportation service should be provided; otherwise the public will not make any use of it. Moreover, all local transport regulations and policies have to be observed. Due to the fact the local authorities are usually accountable to the local community who provide the funding, it is difficult for them to maintain bus routes that are underutilized [59]. Also, since all bus operators are obliged to operate within their budgets, certain constraints, such as limiting the number of available buses and/or the number and lengths of operating bus routes are imposed [60]. The local authorities, on the other hand, are responsible for ensuring an acceptable level of transportation service while at the same time limiting the negative effects of operating buses, such as emissions [61].

In many countries, public transportation networks which are directed by private bus companies are subsidized by local authorities so as to ensure that in all cases a minimum level of service is maintained [4]. Every so often, local authorities and/or private bus companies review bus routes and schedules. Occasionally, the public transportation system of a whole area may be reviewed in order to facilitate some changes in the road network, for example, new one way systems, new ways for trams or trolleys and/or new pedestrian precincts. However, in most cases, relatively minor adjustments are applied, so as to comply with new government policies or minor changes in levels of demand.

A thorough review of bus transit route and schedule design, throughout the world, reveals that in practice no common methodology is used. In the early years, all bus route networks and schedules were designed manually. As a result, the efficiency of the resulting networks was highly dependent on the designer's experience and his/her knowledge of existing resource constraints and transportation demands [5]. Of course, these manual attempts to provide an acceptable solution to the UTRP problem were not able to solve large network problems efficiently. More recently, some commercial software packages have been used, mostly as interactive tools for decision support and visualization. These software packages, that have been widely used in the transporting planning industry include, among others, Cube Voyager [6], Emme [7], SATURN [8] and VISUM [9].

In the last decades, many soft computing approaches such as genetic/evolutionary algorithms, simulating annealing and tabu search have been applied in order to solve urban routing and scheduling problems [10–19]. Also, in recent years, there has been an attempt to create a well established scientific framework for solving the Urban Transit Routing Problem (UTRP) [52]. Many researchers have developed effective methods in order to solve variations of the UTRP and made their input data sets available to other researchers through the internet [41,50,62,63]. As a result, one can make comparisons among them and evaluate their effectiveness.

In this contribution, a new adaptive algorithm based on particle swarm optimization (PSO) [20,21] has been designed, developed and applied to the UTRP. The reason why we decided to use a PSO based algorithm to solve this specific problem is that PSO based algorithms have been widely applied in many constrained optimization problems having very satisfactory results [22–33] and that although there are plenty of PSO algorithms applied to timetabling and scheduling problems in the literature, there is no specific PSO based approach, to the best of our knowledge, applied to UTRP. This was one of our main motivations in order to design and apply a PSO based algorithm in order to solve effectively and efficiently the UTRP. Moreover, the proposed algorithm can be also applied to the suburban transportation problem.

In order to the demonstrate the effectiveness and efficiency of the proposed PSO based algorithm, its performance is compared with seven other very effective approaches published in the literature that have been applied to the same instance of the UTRP [2,15,34,35,62,63]. This instance is widely known in the literature as Mandl's Swiss transit network [2]. It refers to a Swiss urban bus transportation network and it is the only widely used UTRP instance in the respective literature.

In [34], Kidwai made an attempt to use an optimization tool for solving Mandl's UTRP instance. However, a closer look at the procedure proposed in [34] shows that it still remains primarily a heuristic algorithm with use of the optimization tool only marginal. Chakroborty and Wivedi proposed in [15] a technique for the development of "optimal" transit route given the information on link travel times and transit demand. In the proposed method, genetic algorithms are used to develop the "optimal' set of routes. Experimental results demonstrate that the proposed procedure performs better than existing techniques. In [35] Fan and Mumford presented a framework which can be used as a starting point for solving the UTRP. They devised a simple model of the UTRP to evaluate candidate route sets. Their approach is validated using simple hill-climbing and simulated annealing algorithms. According to experimental results presented, the proposed method improves upon published results for Mandl's benchmark problem. In [50], Fan et al. proposed a simple evolutionary multi-objective optimization technique to solve the UTRP. First they presented a representation of the UTRP and introduced their two key objectives, which are to minimize both passenger costs and operator costs. Their experimental results concern the Mandl's benchmark data and a larger transport network. Zhang et al. used in [62] a multi-objective optimization model which consisted of the perspectives of both passengers and operators. They applied a simple multi-objective optimization algorithm to obtain the optimal results of the promoted model applied to the Swiss urban transit network of Mandl. Finally, in [63] a Genetic Algorithm is proposed to solve the UTRP. The main objective is to minimize the passenger costs where the quality of the route sets is evaluated by a set of parameters.

All seven approaches mentioned above, as well as the PSO based algorithm presented in the current contribution use the same formalism for modeling the UTRP, try to minimize an equivalent fitness function (the one presented in Section 3.3) and use the same performance criteria in order to evaluate the quality of the resulted solution. As a result, a straightforward comparison of their experimental results is fair. The comparison of the bus transportation networks created by the proposed PSO based algorithm and the bus transportation networks created by the other seven former approaches is based on five criteria. The first criterion is the average in-vehicle travel time experienced by each passenger of the transit network. The second criterion is the percentage of passengers traveling directly from their origin to their destination. The third criterion is the percentage of passengers traveling from their origin to their destination with one transfer. The fourth criterion is the percentage of passengers traveling from their origin to their destination with two transfers. Finally, the fifth criterion is the percentage of passengers who have to make more than two transfers in order to go from their origin to their destination or cannot reach their destination using public network at all.

Computational results showed that the proposed PSO based algorithm achieves better results compared to all seven approaches mentioned above to solve Mandl's Swiss transit network problem. Except for comparing the performance of the proposed algorithm with other approaches published in the literature we also conducted exhaustive experiments in order to investigate the effect of parameter setting to algorithm's performance. Experimental results showed that the proposed algorithm is robust and manages to design effective bus transportation networks in all cases.

Moreover, we experimented by enriching the fitness function with factors that represent not only the needs of the public but also the profit of bus companies in order to demonstrate the proposed algorithm's ability to cope with contradictory factors. The cost of the operator has been introduced in [50] and [62]. In [50] a simple evolutionary multi-objective optimization algorithm is proposed that mainly operates conducting neighborhood moves. The objective function implemented is the sum of two conflicting parameters. The first is the *passenger cost* which is represented by the total journey time over all passengers (equivalent to cost F_1 of our work). The second is the *operator cost* which is represented by the total length of the route set (equivalent to cost F_4 of our work). Two main cases have been evaluated by the authors. The first one refers to a route set design representing completely the passengers' benefit while the second refers to a route set design representing completely the operator's benefit. Non-dominated tradeoff solutions are also presented in a graph without further commenting though.

In [62] the optimization algorithm used is the same as in [50]. Improvements are made in the objective function, where for *passenger cost* a weighted sum of time accumulated over all passengers (equivalent to cost F_1 of our work) and the total number of transfers for the entire demand (a simpler combination of F_2 and F_3 of our work) is used. Total route length is again used as the *operator cost* (equivalent to cost F_4 of our work). Three cases are evaluated in this work, namely, passenger optimal, operator optimal and passenger–operator balanced solutions.

The innovation of our approach lies on the fact that the various conflicting (or not) elements of route set design are all considered in separate, independently parameterized and weighted, costs of the objective function (F_1 to F_4). This way, and as also shown by the results presented in Section 5, we are capable to efficiently design routes that specifically target desired characteristics. For example cost F_2 is a weighted sum of the individual amounts of transfer demands covered by 0, 1 and 2 transfers. As seen in Section 5.4 our proposed algorithm can distinguish between these elements and promote for example zero transfers at the expense of the total journey time and total length of the route set. This feature of our approach is innovative compared to the other cited multi-objective approaches. Moreover, as seen in Section 5.2 a similar approach can be applied for the effect of the operational cost versus the other costs, in line with the manuscripts by Fan et al. [50] and Zhang et al. [62].

The rest of the paper is organized as follows. Section 2 defines and presents the UTRP. Section 3 describes the proposed PSO based algorithm relative to particle encoding, initialization procedure, evaluation function and PSO operations. Section 4 describes the input data used and compares the performance of the proposed algorithm to that of existing approaches while Section 5 investigates the effect of parameter setting to algorithm's performance. Finally, Section 6 provides summary and future work.

2. Problem description

2.1. Urban Transit Network Design Problem (UTNDP)

In the literature, the problem of designing urban transit routes and schedules which deal with practical constraints is referred to as the Urban Transit Network Design Problem (UTNDP).

The UTNDP belongs to a broader class of optimization problems that is commonly known as Vehicle Routing Problems (VRPs). Moreover, its two main components are the Urban Transit Routing Problem (UTRP) and the Urban Transit Scheduling Problem (UTSP) [13]. Generally speaking, the UTRP deals with the development of efficient and effective transit routes taking into account existing road networks and predefined pick-up/drop off points. The UTSP, on the other hand, involves the assignment of schedules to all vehicles that are used to carry passengers on a road network. Naturally, the UTRP and the UTSP are solved sequentially because the development of routes should be completed before the development of schedules starts.

The UTNDP is a very difficult optimization problem due to the fact the there are many different criteria that should be optimized in order to meet the needs of all passengers but also to minimize the costs of the service operators. Transportation companies try, of course, to minimize their costs, a fact that may lead to providing poor transportation services to the traveling public. Factors that affect a transportation company's costs are transit vehicle size and miles, vehicle operation hours required for a specific route configuration and, of course, the fleet size [36]. On the other hand, from the passenger's point of view, an efficient public transportation system is one which provides rapid travel times and frequent services between all source and destination points, having a minimum of transfers between vehicles on the way. Moreover, there are other stake-holders involved in the development of efficient urban transit routes and schedules such as national and local government, local business and taxpayers. Although all the above mentioned parties will benefit from an efficient public transportation service, each one of them evaluates the transportation service based on its own profit and perspective.

2.2. Urban Transit Routing Problem (UTRP)

As stated in the previous sections, the UTRP aims to develop a set of vehicle routes for an existing urban transit network while satisfying specific constraints. In fact, it comprises the design of physical transportation routes needed to solve the UTNDP [37]. In a transportation network, adjacent nodes, for example bus stops, are linked by an edge. A specific route consists of several nodes connected by edges which all together form a transportation path. In Fig. 1, an example of a simple transit network is presented that consists of five nodes and seven connections (edges). A possible bus route consists of several nodes connected by edges in order to form a path. For example, it is possible to form a valid route between nodes 1, 0 and 3 since there are edges in the transit network that actually connect these nodes. On the contrary, a bus route between nodes 2, 3 and 0 cannot be formed since there are no edges connecting these nodes in the transit network. One or more transportation paths (valid bus routes) can be combined to form a route set. A route network consists of all the routes of a route set that have been superimposed. In Fig. 1, two possible route networks are presented that consist of different route sets. Obviously, a route network should contain all the nodes of the original transit network, but may not contain all its edges. Each route network is a sub graph of the original transit network. In the ideal case, the nodes of the route network (sub graph) should be connected in such way that there is a path connecting every node of the graph with every other node, so as to satisfy all public traveling demands.

In order to form an efficient route network, it is necessary to obtain accurate estimations of all travel demands. Travel demands can be estimated in several ways [38]:

- undertaking a public and private vehicle analysis.
- carrying out a survey on the local population.
- examining current ticket sales, etc.

However, in practice, such estimations are very difficult mainly because travel demands change dynamically and are extremely sensitive to factors like quality of service, pricing policy, etc. Ideally, a good route network will ensure that routes having the most traveling demands are satisfied with short travel paths and few vehicle transfers. This, however, will probably affect the service level of less popular routes. Except for satisfying public's transfer demands, the design of an affective route network is also affected by several other factors such as local government's transport management policies, local area's street environment, etc. [39].

2.2.1. Input data and optimization criteria

The UTRP input data used by the PSO based algorithm presented in the current contribution is described as follows:

- data concerning the structure of the road network, that is, how the nodes of the road network are connected.
- data concerning travel times between the nodes of the road network, that is, how much time is needed in order a vehicle to travel between any two nodes of the road network.
- data concerning travel demands between any two nodes of the road network.

The optimization criteria used in order to evaluate the performance of the proposed PSO based algorithm, which have been adopted by many researchers in the literature [10,13,15,34,40,41] are the following:

- the percentage of demand unsatisfied, which should be equal to zero.
- the percentage of demand satisfied without any transfers, which should be as high as possible.
- the average travel time in minutes per transit user, which should be as low as possible.
- the efficiency of the route network, which should be as high as possible (this means that travel requirements with a heavy demand should be satisfied with higher priority compared to routes having lower transfer demands).

Moreover, in practice, some real world constraints should be satisfied [36]:

- each route should have a minimum and a maximum length (defining a minimum length is necessary to ensure connectivity of the route network while defining a maximum length is necessary to aid bus schedule adherence).
- the road network should comprise a connected graph, that is, there should be a path connecting any two nodes of the road network, so that any passenger using the route network can travel between any two nodes (Fig. 2).
- no cycles or backtracks should be allowed in individual routes.
- the number of routes in the route set will be predefined (usually by the service provider due to cost limitations).

The above four constraints were also adopted by the proposed PSO based algorithm.

2.2.2. Solution approaches and drawbacks

As stated in the previous section the basic aim of designing a route network is to optimize specific criteria that define its efficiency and quality. As a result, the mathematical model of the problem should be formulated in such a way that its solution should lead to an optimal set of routes for specific input data and conditions. In the literature, there have been many attempts to formulate and solve the UTRP problem using a mathematical approach [10,42]. However, as noticed in [13], the majority of these attempts describe the UTRP by using an optimization function and defining some specific constraints which refer to some desired characteristics of the route network. Nevertheless, none of these mathematical approaches can define the specific routes of the network through the mathematical optimization problem. According to [10] the difficulty to represent the UTRP using a mathematical approach is due to the fact that the problem is discrete. For example, the basic variables of the problem are the nodes of the network and the routes connecting them. Newell noticed in [43] that the UTRP is in its general form a non-convex optimization problem which is very difficult to solve. As a result, the main characteristics of the UTRP, which make it difficult to solve, are its nonlinearity and the necessity to use logical variables, for example, in order to define whether two nodes are connected with each other or not. According to Chakroborty [13], the difficulty to solve the UTRP using conventional methods does not lie in its complexity but mainly because mathematical approaches fail to represent it in a proper and effective way. Mathematical approaches either fail to represent correctly the UTRP problem or manage to do so by adding extreme computational burden. As a result, most of the approaches applied to the UTRP problem are based on heuristic methods [10,44–47].

From above, one can come to the conclusion that the UTRP problem is a very difficult problem. Its main characteristics, which make it very difficult but also very challenging to solve, are the following:

i. it is NP-complete in its general form, as far as its computational complexity is concerned, meaning that the difficulty to find a solution rises exponentially to its size and a deterministic

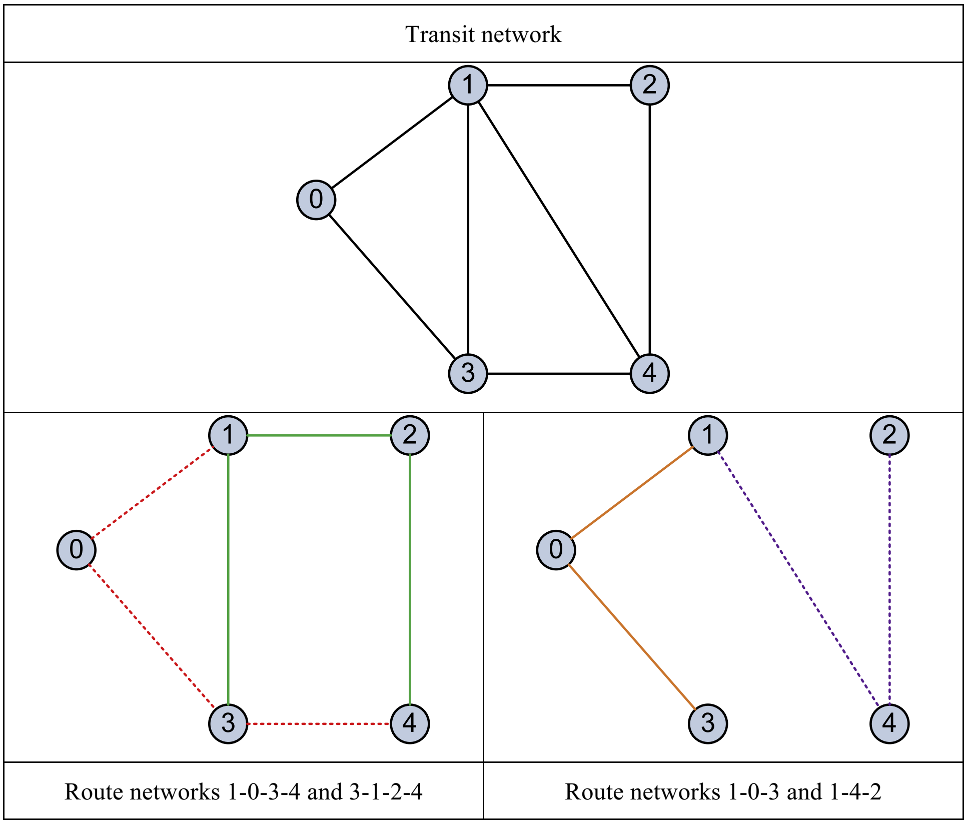

Transit network
0
1
2
3
4
Route networks 1-0-3-4 and 3-1-2-4
Route networks 1-0-3 and 1-4-2

Fig. 1. An example of a transit network and two possible valid route networks.

algorithm, giving an acceptable solution in polynomial time, cannot be found.

ii. in the literature, although many different versions of the problem have been presented, no commonly accepted models have been adopted.

iii. UTRP's constraints are very difficult to model, check and meet. For example, to check whether a route set is valid, that is whether the nodes of the route set are fully connected, adds significant computational burden.

iv. different parts of the final solution strongly depend on each other. For example, each route cannot be evaluated separately. The quality of a route depends strongly on the other routes belonging to the final route set. As a result all routes belonging to the final route set have to be evaluated together as a whole.

v. it involves many contradictory targets that have to be met and as a result it is an inherently multi parametric problem [48]. For example, reducing service costs, maximizing area service coverage and minimizing average travel time are three contradictory targets which compete with each other.

vi. collecting all necessary input data in order to design an affective route set can be really difficult, especially regarding travel demands. As a consequence, route set design can be totally wrong if input data are poor, regardless the solution approach applied. In practice, travel demands are different during a day (Fig. 3), a fact that makes the UTRP very difficult. Ideally, the UTRP problem should be solved only once in order to create the final route network. However, since travel demands are dynamically changed, many alterations have to be applied so as to fulfill new public transportation needs. Finally, travel services are highly affected by the weather and the road network condition. These factors should be also taken into account in order to apply changes to existing route networks.

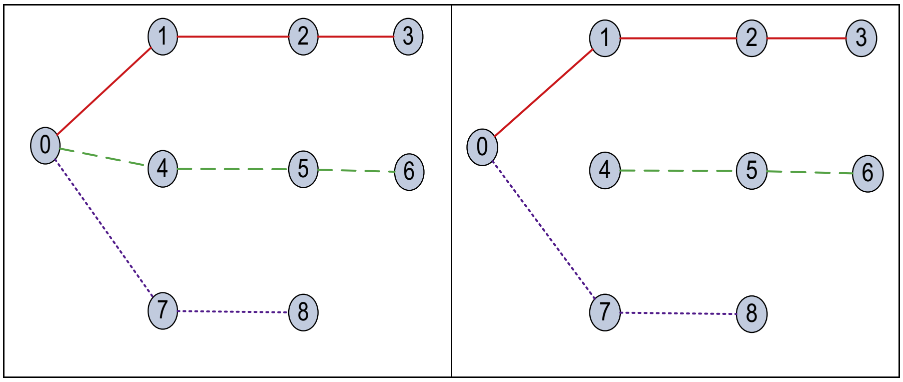


Fig. 2. An example of a connected (left) and a non-connected (right) road network.

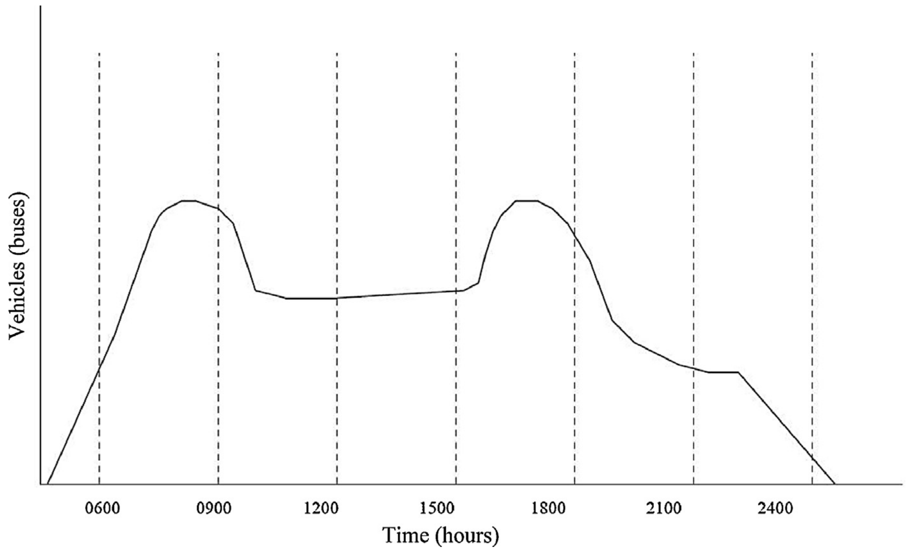


Fig. 3. Typical demand changes in the number of vehicles (buses) needed during a day [49].

Table 1
An example of the route set representation used by the proposed algorithm.

Route 1:	10	11	12	4	2	1	−1	−1	−1	−1
Route 2:	1	2	5	4	6	8	10	14	−1	−1
Route 3:	10	11	13	14	10	8	6	3	2	−1
Route 4:	9	15	7	10	8	6	3	2	5	4

3. The proposed PSO based algorithm

3.1. Route set representation

For the route set representation used by the proposed PSO based algorithm a two dimensional array is selected. A similar route representation approach was also selected in [35]. For example, assuming that the route set comprises of four different routes each of them having 10 nodes at the most, an array of $4 \times 10 = 40$ cells is needed in order to represent the solution (Table 1).

In the route set representation example presented in Table 1, route 1 starts at node 10, continues to node 11, then goes to node 12, etc. The same holds for the other three routes. It is important to notice that a value of −1 in a cell means that the cell is empty, that is the route has ended in a previous node. For example, route 1 comprises of 6 nodes. As stated before, the total number of routes is specific, set by the transportation (bus) company due to cost limitations, while the maximum length is usually defined by local authorities' regulations. Both parameters' values are predefined by the user before the execution of the algorithm.

3.2. Initialization procedure

The proposed PSO based algorithm, as other PSO or evolutionary approaches do, operates on a population of possible solutions for a predefined number of generations. This population has of course to be initialized in such a way that it does not obstruct the evolution of the applied algorithm. Ideally, the optimization algorithm should be able to reach optimal route sets independently of the initial population of route sets. In practice, however, since the UTRP is a very difficult problem to solve (see Section 2.2.2), it is preferable the initial population of route sets not to be generated at random but to be created following some specific logical principles. Each route set is created by defining the number of routes it consists of. Each route is created by initially selecting the first node and then selecting the rest of the nodes till some specific termination criteria are satisfied. In order not to add computational burden on the basic algorithm, the initialization procedure is quite simple, aiming only at the generation of an initial route set having some specific good characteristics. The initialization procedure, which is described in detail in the following sections, has been mainly based on the approach presented in [15] by Chakroborty and Wivedi, partly changed using the *Make-Small-Change* procedure, as described by Fan and Mumford [35] and Fan et al. [50]. The initial population consists of N particles, where N is a number set by the user. As a result, the steps described in the following sections are repeated N times, one time for each particle.

3.2.1. Selecting the first node

The specific steps executed to select the first node of each route are the following:

Step 1: Estimate the activity level $a_i = A(i) + P(i)$ for each route node i, where $A(i)$ and $P(i)$ are the number of routes ending at and beginning from node i, respectively. The activity level refers to the amount of travel demands (routes per day between nodes), which is part of the input data of the problem. For the Swiss road network by Mandl, studied in the current contribution, these travel demands are presented in Table 4 (Section 4.1). The activity level of each node ($A(i) + P(i)$) is essentially the sum of each line of this table. This number represents the amount of routes needed per day to satisfy the passengers' travel demands at each specific node. For route 0, for example, the activity level would be 1320. The routes, however, are not initialized at this step yet.

Step 2: Sort the nodes in descending order based on the activity level of each node.

Step 3: Select the first K nodes from the sorted list created in Step 2. The first K nodes comprise the Initial Node Set (INS), where parameter K is defined by the user. The value of parameter K was chosen equal to 14, in line with the work of Chakroborty and Wivedi, on which the initialization procedure was mainly based. Given that the number of nodes in the Swiss network studied is 15, the selected value of K means that all nodes with some activity are considered as possible first nodes, besides node 14 which, as seen in Table 4, does not attract or generate any transfers.

Step 4: Based on the activity level of nodes belonging to INS, a probability $p_j^{(I)}$ is assigned to each node j, which reflects the probability of each node to be selected as the first node of the route. This probability is calculated as follows:

$$p_j^{(I)} = \frac{a_j}{\sum_{\forall j \in INS} a_j} \quad (1)$$

Step 5: Select at random a node from INS based on the values of probabilities $p_j{}^{(I)}$ in order to be the first node of the route.

Since, a node is selected, it is extracted from INS and the probabilities $p_j{}^{(I)}$ are calculated from the beginning, as described above. This procedure is executed for R times, where R is the number of routes of each route set.

3.2.2. *Selecting the rest nodes*

In the following paragraphs, as Previous Node (PN) we assume the last node which has been inserted in a route. For example, when the third node is selected then PN node is the second one.

The specific steps executed to select the rest nodes of each route are the following:

Step 1: Find for a specific PN all its adjacent nodes, that is, all nodes that are directly connected to PN. This data are available as input data of the algorithm in form of an adjacency matrix, where a positive value in a cell [i,j] means that nodes i and j are directly connected, while the specific value of the cell reflects the time distance between nodes i and j. Adjacent nodes form a set called Vicinity Node Set (VNS). In order one node to be inserted in VNS, except for being adjacent to PN, it should not have been already inserted in the current route. This check avoids cycles in routes.

Step 2: Based on the activity level of nodes belonging to VNS, a probability $p_k{}^{(N)}$ is calculated, which reflects the probability of node k to be selected as the next code of the current route. This probability is calculated as follows:

$$p_k^{(N)} = \frac{d_k a_k}{\sum_{\forall k \in VNS} d_k a_k} \quad (2)$$

where a_k is the activity level of node k, which is calculated as described above, and d_k is a user defined term. The determination of d_k is not straightforward and depends entirely on the respective connectivity of the node network under investigation. Parameter d_k is a biasing term that intends to tune the initial route construction, by avoiding, e.g. the generation of circles, etc. In such cases, the value of d_k is sustained low, so as the selection probability of node k to be small. Moreover, this parameter is not statically defined. On the contrary, its value changes dynamically during the initialization procedure for each node separately. In current work and based on exhaustive experiments conducted during the development and tuning of the algorithm, we implemented the following: we observed that nodes connected with only one other node of the network, for example nodes 0 and 8 in Fig. 7, provoke premature termination of a route if selected as intermediate nodes at the initialization stage, given that no direction reversals are allowed. As a result, the value of d_k for these nodes is set initially equal to 0.5, while for the rest of the nodes it is set equal to 1. As soon as a node is selected, as part of a route, its term d_k is divided by 10, in order to bias toward the selection of other nodes which have not been selected already. This choice aims to have, if possible, all nodes participating to each route set, since as described in Section 2.2.1, a major characteristic of a good route set is to fulfill all transfer demands. Implementing the above procedure, we managed to obtain a balanced initialization procedure with almost all particles of the swarm containing all nodes already from this stage. In conclusion, it should be clarified that the above procedure is not based on some theoretical reasoning, but relies, as mentioned, on the exhaustive experimental trials we conducted.

Step 3: Define the next node of a route by selecting at random a node belonging to VNS based on the values of probabilities $p_k{}^{(N)}$. As mentioned above, as soon as a node is selected as part of a route, its term d_k is divided by 10

3.2.3. *Creating the route set*

The procedure described in Sections 3.2.1 and 3.2.2 is repeated until one of the following criteria is satisfied:

i. the number of a route's nodes is equal to maximum number of nodes (M) a route can consist of.
ii. the length of a route in minutes is equal to maximum route length (L).

The values of both parameters M and L are user defined.

However, in many cases, a route is completed before one of the above criteria is satisfied, since the adding nodes procedure may lead to a point where no more nodes can be added in VNS. To cope with these cases some features of the *Make-Small-Change* procedure, as described by Fan and Mumford [35] and Fan et al. [50], have been incorporated. More precisely, the order of the route nodes is flipped and new nodes are added starting from the first node of the route which is now the PN. In order to ensure that each route has as many nodes as possible, 10 attempts are executed. The final route created is the one having the biggest number of nodes. Creating different members in each route set is repeated till all R members of each route set are completed based on the criteria described above.

3.3. *Route set evaluation*

Evaluating a route alone has no sense, since its path depends strongly on the rest of the members of the same route set. As a result, all members of a route set should be evaluated as a whole. In order to evaluate the route sets a number of criteria have been adopted, which provide a measure of "quality" of each route set. These criteria are the following:

(1) The average (or total) in-vehicle travel time (including transfer time) experienced by each passenger (or all passengers) of the transit network.
(2) The percentage of passengers traveling directly (i.e. without a single transfer) from their origin to their destination.
(3) The percentage of passengers traveling from their origin to their destination by making a single transfer.
(4) The percentage of passengers traveling from their origin to their destination by transferring twice.
(5) The percentage of passengers who cannot use the transit network to go from their origin to their destination by making a maximum of two transfers.

These criteria were used in the proposed PSO based algorithm since they have been already introduced in the literature by other researchers [2,10,15,35,44] in order to evaluate the quality of their route sets. More precisely, in the current contribution we adopted the approach presented in [15,35], where all the above presented criteria are used in order to estimate one single number, i.e. *TOTFIT*(r). The value of *TOTFIT*(r), which suffices in order to evaluate the fitness of each route set r, is calculated as follows:

$$TOTFIT(r) = \omega_1 F_1(r) + \omega_2 F_2(r) + \omega_3 F_3(r) \quad (3)$$

where $F_1(r)$ is the score obtained by evaluating the route set r using the first criterion only, $F_2(r)$ is the score obtained by evaluating the route set r using the second, third, and fourth criteria and $F_3(r)$ is the score obtained by evaluating the route set r using the fifth criterion only. Parameters ω_1, ω_2, and ω_3 are user specified weights for scores $F_1(r)$, $F_2(r)$ and $F_3(r)$, respectively. In the present contribution, the values of all these three weights were taken equal to one, so as to be in total accordance with the results presented in [15,35]. Since the fitness function used by these two approaches and the proposed soft computing approach is the same, a straight comparison between their results is quite fair. Next, an analytical description of the fitness function estimation procedure is presented, which is based on the methodology introduced by Chakroborty and Wivedi [15].

3.3.1. Calculating $F_1(r)$

$F_1(r)$ is a score which reflects the average time spent by each passenger when he/she travels using a specific route set. Its value is small if the respective average traveling time is big and big if the respective average traveling time is small. In order to estimate $F_1(r)$, not only the average traveling time has to be calculated but one has also to determine whether this value should be considered as "big" or "small". The estimation of $F_1(r)$ is done using the following steps:

Step 1: For every node pair $[i,j]$ and a given route set r the In-Vehicle Travel time $IVT_{i,j}(r)$ is computed by determining the smallest time in which a passenger can travel from node i to node j using the routes of the r^{th} transit route set. Traveling time $T_p(r)$, on a path p, is calculated by the following equation:

$$T_p(r) = \sum\nolimits_{\forall a \in p} t_a + nU \tag{4}$$

where t_a is the traveling time to node a starting from the previous node of the path, n is the number of transfers involved in path p and U is a time penalty paid for each transfer. In the present contribution the value of parameter U is set equal to 5 min, in accordance with other approaches [2,10,15,35]. $IVT_{i,j}(r)$ is the smallest $T_p(r)$ of all possible paths connecting nodes i and j.

Step 2: The absolute minimum traveling time ${T_{i,j}}^{min}$ between nodes i and j is calculated. Its value depends only on the road network and not on the route set and the respective transfer delays. In order to calculate ${T_{i,j}}^{min}$ the algorithm of Floyd [51] is used.

Step 3: Traveling time ${T_{i,j}}^{min}$ is compared with the respective $IVT_{i,j}(r)$ so as to determine index $f_{i,j}(r)$, which indicates whether $IVT_{i,j}(r)$ is "big" or "small". Index $f_{i,j}(r)$ is calculated using the following equation:

$$f_{i,j}(r) = \begin{cases} -\left(\dfrac{\beta_1}{x_m} + \dfrac{K_1}{x_m^2}\right)x^2 + \beta_1 x + K_1 & 0 \le x \le x_m \\ 0 & x > x_m \end{cases} \tag{5}$$

where $x = IVT_{i,j}(r) - T_{i,j}^{\min}$, x_m is the upper limit value of x, K_1 is a positive user defined parameter which defines the maximum value of $f_{i,j}(r)$ and $K_1/x_m \le \beta_1 \le 0$. Index $f_{i,j}(r)$ is then used in order to calculate the value of $F_1(r)$ as follows:

$$F_1(r) = \frac{\sum\nolimits_{\forall (i,j) \in S(r)} d_{i,j} f_{i,j}(r)}{\sum\nolimits_{\forall (i,j) \in S(r)} d_{i,j}} \tag{6}$$

where $S(r)$ is the set of node pairs $[i,j]$ for which transfer demands, $d_{i,j}$, are satisfied by route set r.

Table 2
Results of both route set evaluation methods for best route sets published in [2] and [35].

Evaluation parameters	Mandl [2]		Fan and Mumford [35]	
	Method A	Method B (published)	Method A	Method B (published)
d_0 (%)	67.69	69.94	86.06	93.26
d_1 (%)	26.53	29.93	13.81	6.74
d_2 (%)	5.78	0.13	0.13	0.00
d_{unsat} (%)	0.00	0.00	0.00	0.00
ATT (min)	13.18	12.90	11.72	11.37

Table 3
Results of both route set evaluation methods for best route sets published in [15].

	Chakroborty and Wivedi [15]		
Evaluation parameters	Method A	Method B	Published results
d_0 (%)	72.70	79.38	86.86
d_1 (%)	19.78	17.60	12.0
d_2 (%)	7.51	3.02	1.14
d_{unsat} (%)	0.00	0.00	0.00
ATT (min)	11.89	11.52	11.90

3.3.1.1. Calculating minimum travel time $IVT_{i,j}(r)$. At this point it is important to notice that the calculation of minimum time using Eq. (4) is not so obvious, as it may seem at first, because it is possible to have different results depending on whether transfer times are being considered or not. In practice, a passenger may either select a route without taking into account possible transfer delays or select a different route having in mind all delays provoked by possible transfers. In the current contribution, both route evaluation methods, *Method A* and *Method B*, respectively, were examined. As noticed after exhaustive experiments, *Method B* is much more computational demanding. As a result, we decided to use *Method A* in order to evaluate routes. However, after comparing the results of these methods with the respective results published in the literature, it was obvious that at least in two cases [2,35] *Method B* was used. The best solution found by the proposed PSO based algorithm is evaluated, in the end, using both *Methods A* and *B* so as to have a fair comparison of the results of the proposed algorithm with the results of other contributions applied to the same instance of the UTRP.

Table 2 presents the results of both route set evaluation *Methods*, *A* and *B*, for the best solutions, as published in [2] and [35] More precisely, for the Swiss Network which is described in detail in Section 4, the best solution presented in [2], in the case of a set of four routes, is the following:

Route 1:	0	1	2	5	7	9	10	12
Route 2:	4	3	5	7	14	6		
Route 3:	11	3	5	14	8			
Route 4:	12	13	9					

Respectively, the best solution proposed in [35] in the case of a set of four routes is the following:

Route 1:	9	13	12	10	11	3	1	0
Route 2:	11	10	9	7	5	2	1	0
Route 3:	10	9	7	5	3	4	1	2
Route 4:	1	2	5	7	9	6	14	8

As presented in Table 2, using *Method B* we were able to reproduce all published results successfully. It is also interesting to mention that *Method B* produces results at least as good or better to results produced by Method A. Average travel time (ATT) is less in both cases (see Table 2), while the percentage of passengers traveling directly (i.e. without a single transfer) from their origin to their destination is bigger. Logically, passengers taking into account transfer delays, due to intermediate transfers, are able to select shorter paths.

On the contrary, as presented in Table 3, we were not able to reproduce the published values of Chackroborty and Wivedi [15].

The best solution published in [15], in the case of a set of four routes, is presented below:

Route 1:	5	7	9	6	14						
Route 2:	13	12	10	9	7	5	2	1	4	3	11
Route 3:	8	14	5	3	1						
Route 4:	0	1	2	5	7	9	12	10	11		

The evaluation parameter values presented in [15], for the above solution, are different from the ones estimated by us using *Methods A* and *B*. The authors, though, do not describe in detail which method they used. However, when comparing the results of the proposed PSO algorithm with the results presented in [15] the evaluation parameter values used are the same with the ones used in [15]. However, we were not able to reproduce the published results as presented in Table 3.

3.3.2. Calculating $F_2(r)$

$F_2(r)$ is a score which reflects the percentage of passengers traveling from their origin to their destination either directly, or by making a single transfer or transferring twice. Calculation of $F_2(r)$ is done using the following equation:

$$F_2(r) = \frac{K_2 - \beta_2 \alpha}{\alpha^2} d_T^2(r) + \beta_2 d_T(r) \tag{7}$$

where K_2 is a positive user defined parameter which defines the upper limit of $F_2(r)$, $K_2/\alpha \leq \beta_2 \leq 2\,K_2/\alpha$ and a is the upper limit of $d_T(r)$; $d_T(r)$ is an averaged sum calculated as follows:

$$d_T(r) = ad_0(r) + bd_1(r) + cd_3(r) \tag{8}$$

where $d_0(r)$ is the percentage of passengers traveling from their origin to their destination directly, $d_1(r)$ is the percentage of passengers traveling from their origin to their destination making a single transfer and $d_2(r)$ is the percentage of passengers traveling from their origin to their destination transferring twice. Parameters a, b and c are user defined weights satisfying inequality $a \geq b \geq c$. They represent the respective importance of the ways in which the demand should be satisfied in a qualitative route set, for example, trying to have a low percentage of intermediate transfers may lead to a larger average travel time.

3.3.3. Calculating $F_3(r)$

$F_3(r)$ is a score which reflects the percentage of passengers who cannot travel from their origin to their destination using route set r. It is calculated using the following formula:

$$F_3(r) = -(\beta_3 + K_3) \cdot d_{unsat}^2(r) + \beta_3 \cdot d_{unsat}(r) + K_3 \tag{9}$$

where K_3 is a positive user defined parameter which defines the upper limit of $F_3(r)$, $-K_3 \leq \beta_3 \leq 0$ and $d_{unsat}(r)$ is the percentage of total transit demand which is unsatisfied using route set r.

3.3.4. Service provider's operational cost – calculating $F_4(r)$

The approach of the UTRP used in the present contribution and in [2,15,35,50] cannot cope with operational costs relating to frequency of services needed for a route in order to provide a qualitative transfer service to passengers. For studying this kind of operational costs vehicle scheduling issues should be considered. However, other kind of operational costs, such as the average fuel consumption per vehicle depend on the total route length [34] and can be examined using the present approach.

The fitness function proposed by Chakroborty and Wivedi in [15] does not include a cost factor for this kind of operational costs. In the present contribution, we included a cost score $F_4(r)$ in the evaluation function used, which depends on the total length of each route set, in order to study the possibility to design qualitative route sets for the service provider's point of view. More precisely, the total length L_{tot} of each route set is calculated as the sum of all time distances between nodes which comprise the routes of each route set. These time distances are supposed to be proportional to fuel consumption. Score $F_4(r)$ is calculated as follows:

$$F_4(r) = \begin{cases} -\left(\dfrac{\beta_4}{x_{4m}} + \dfrac{K_4}{x_{4m}^2}\right) \cdot L_{tot}^2 + \beta_4 \cdot L_{tot} + K_4 & F_4(r) \leq K_4 \\ K_4 & F_4(r) > K_4 \end{cases} \tag{10}$$

where K_4 is a positive user defined parameter which defines the upper limit of $F_4(r)$, $-K_4/x_{4m} \leq \beta_4 \leq 0$ and x_{4m} is a limit value for L_{tot} according to which route sets having L_{tot} values smaller than the value of x_{4m} are supposed to be optimal. In these cases, $F_4(r)$ reaches its maximum value and L_{tot} equals $L_{tot} - x_{4m}$.

3.4. Modifying route set procedure

In this section the procedure used in order to modify route sets is presented in detail. This modification procedure is a PSO based one and it is, according to our knowledge, the first time in the literature that a PSO based procedure is applied to UTRP in order to modify and evolve route sets. As described in [32], due to the discrete nature of this kind of optimization problems applying the classic PSO's velocity vector in order to evolve particles is not advisable. In the proposed procedure, particles are modified at each iteration of the PSO based algorithm, by inserting elements either from the personal best of each particle or the global best of the whole swarm [32]. The elements of particles modified are either parts of routes or whole routes.

The modification procedure is evolutionary, that is, at each iteration the route sets (particles) of the current population are modified, using specific alteration methods, in order to create better route sets for the population of the next iteration. These alteration methods are applied, at each iteration, to the whole particle population (swarm). As described in Section 3.1 each particle is represented using a two dimensional array, which in fact constitutes a route set. In the proposed algorithm we apply two different modification procedures, namely *Method 1* and *Method 2*.

Method 1 is twofold and concerns: (i) interchanging route parts between each current particle and its personal best and (ii) interchanging route parts between each current particle and the global best of the swarm. In both cases, at first, two routes are selected, one from the current particle and one from the personal or the global best, respectively. These selections are made at random, that is, all routes are equally possible to be selected. In order to select the routes, the rand function of MATLAB is used, which returns a random value following the standard uniform distribution from the open interval (0,1). This is in turn multiplied with the number of routes in the set (e.g. 4) and then rounded to the nearest integer in the ascending direction. As such, a random route has been chosen. The same approach is applied for both the particle under modification and the global/personal best particle. Following this step, a common node between the two routes is searched for iteratively. As such, starting from the first node of one route and in an increasing direction the nodes of the other route are checked one by one also in an increasing direction. Hence, the first nodes of each routes that are common will be selected.

If a common node is found, the method checks whether interchanging route parts between the current particle and its personal or global best leads to a cycle. More precisely, the method checks whether the route part of the personal or global best which will be inserted in the current particle has common nodes with the rest part of the current particle. In cases where no cycle is created, interchanging of nodes is performed; otherwise another common node between the selected routes is searched. If no common node is found, the method performs nine other attempts to find a common node using other pairs of routes. The mutation operation proposed in [63] is a bit similar to *Method 1*: parts of routes are interchanged

Table 4
Transfer demands (routes per day between nodes) for the benchmark Swiss road network [4].

j	*i*														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	400	200	60	80	150	75	75	30	160	30	25	35	0	0
1	400	0	50	120	20	180	90	90	15	130	20	10	10	5	0
2	200	50	0	40	60	180	90	90	15	45	20	10	10	5	0
3	60	120	40	0	50	100	50	50	15	240	40	25	10	5	0
4	80	20	60	50	0	50	25	25	10	120	20	15	5	0	0
5	150	180	180	100	50	0	100	100	30	880	60	15	15	10	0
6	75	90	90	50	25	100	0	50	15	440	35	10	10	5	0
7	75	90	90	50	25	100	50	0	15	440	35	10	10	5	0
8	30	15	15	15	10	30	15	15	0	140	20	5	0	0	0
9	160	130	45	240	120	880	440	440	140	0	600	250	500	200	0
10	30	20	20	40	20	60	35	35	20	600	0	75	95	15	0
11	25	10	10	25	15	15	10	10	5	250	75	0	70	0	0
12	35	10	10	10	5	15	10	10	0	500	95	70	0	45	0
13	0	5	5	5	0	10	5	5	0	200	15	0	45	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

at appropriate nodes, but in [63] these changes take place within the same route set, while in *Method 1* they take place among personal/global best and current particles.

To make a long story short, *Method 1* can be described in steps as follows:

1. Two routes are selected randomly; one from the particle under modification and one from the personal or global best.
2. A common node is attempted to be found between this pair of routes.
3. If a common node is found, it is verified that no circle occurs by interchanging route parts between selected nodes.
4. If a circle is identified, another common node is searched for, in the same pair of routes.
5. This procedure terminates if a suitable node is found (with no circle generation) or if no suitable node is identified.

In case that no suitable node is found, the algorithm selects a new pair of routes, again randomly, trying to find a suitable node for interchanging route parts following the entire procedure described above. This happens at a maximum of nine additional times. However, if a node is indeed found, then the entire procedure terminates and a new particle is chosen from the swarm to be modified.

For example, supposing that from the route network presented in Fig. 4 routes with continuous lines are selected for interchanging parts. One feasible interchange is at node 4, where part 4-5-6 of the global best may replace part 4-6 of the current particle. In fact, in Fig. 4 there are multiple common nodes between the selected routes. These are nodes 2, 3, 4 and 6. Fig. 4 is used as a simple graphical example of the application of the modification procedure. In this specific case, the route with continuous line has been selected from the current particle (top row, left panel) to be modified by interchanging a part of it with a route part from the route with continuous line of the personal or global best particle (top row, right panel). At first, a common node is selected by the algorithm, specifically node 4. As a result of the modification procedure the 4-6 part of the route of the particle is interchanged with the 4-5-6 part of the best particle. As such, the resulting particle after modification is the one at the bottom of Fig. 4 (particle after modification). The new route with continuous line is now 7-8-2-3-4-5-6 instead of 7-8-2-3-4-6. Of course, this is a simplified example for illustration purposes mainly. The actual algorithm would start looking for common nodes in the selected routes from "left to right" and in this specific case would have chosen to change parts based on node 2, which is the first common one. Then, it would have searched for a circle and, since in this case there is no circle, it would have made that change instead of reaching node 4.

Method 2 is much simpler since it interchanges whole routes between each current particle and its personal or global best. For example, supposing that from the route network presented in Fig. 5 routes with dashed lines are selected for interchanging parts. Route selection is again made at random, following the procedure described in previous paragraphs. The crossover operation presented in [63] is similar to *Method 2*: entire routes are interchanged between chosen chromosomes.

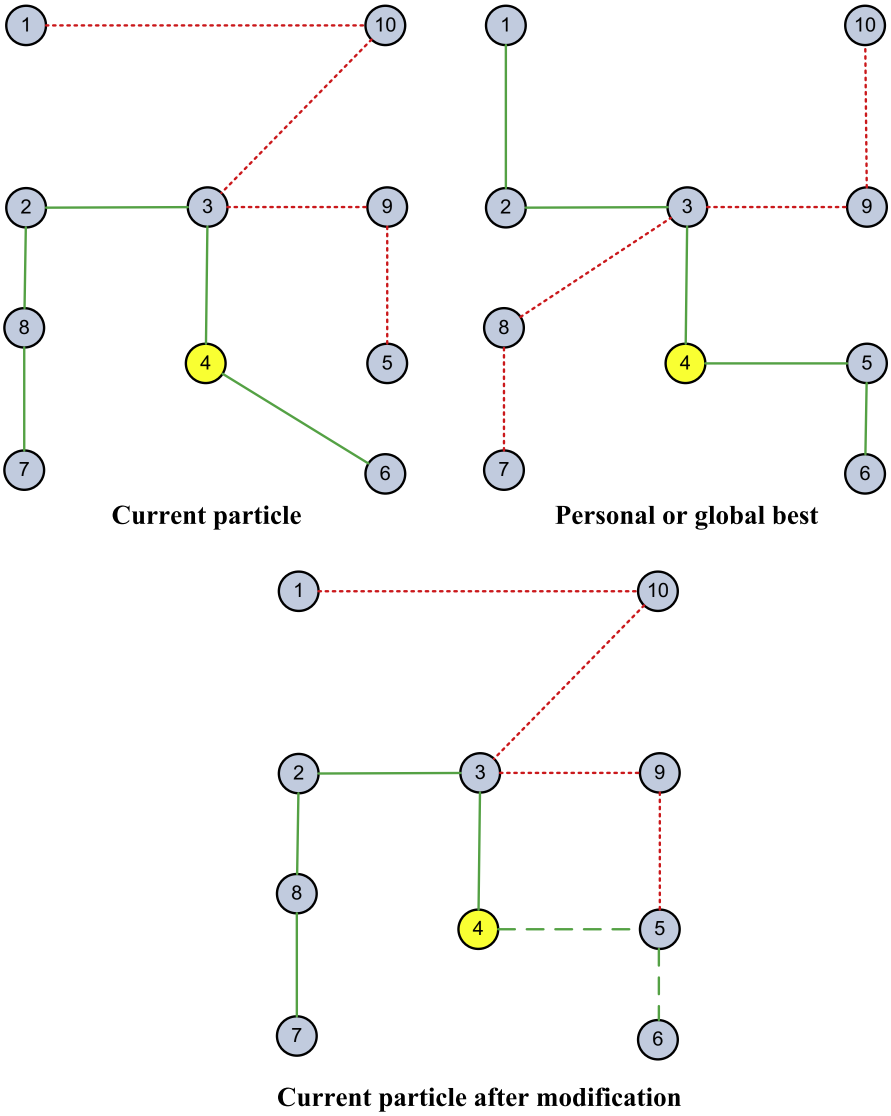
At this point we have to mention that in case the current particle's fitness becomes better that the fitness of its personal best, after the application of each modification method, its personal best fitness value is updated. The same holds also in case the current particle's fitness becomes better that the fitness of the global best. The flowchart of the proposed PSO based algorithm is presented in Fig. 6.

4. Computational results

In this section the computational results of the application of the proposed PSO based algorithm to the UTRP are presented. In order to demonstrate the efficiency and the effectiveness of the proposed algorithm the Swiss road network introduced by Mandl [2] is used. This road network has been already used as a benchmark problem by many researchers in the literature [10,15,35,50]. Four different instances of the problem have been studied, each having different number of routes, in order our experimental results to be straight comparable with the results published by other researchers. More precisely, we examined four different cases with four, six, seven and eight routes in the route set, respectively. For each case, the best route set is presented and the respective experimental results are compared with the results published by Mandl [2], Baaj and Mahmassani [10], Kidwai [50], Chakroborty and Wivedi [15] and Fan and Mumford [35]. In all these approaches, a transfer penalty of 5 min is applied to each route for each required transfer. The same transfer penalty value is also used by the proposed PSO based approach. The evaluation criteria used in order to compare the performance of these algorithms with the algorithm presented in the current contribution are the ones presented in Section 3.3.

4.1. Input data

The road network used as input by the proposed PSO based algorithm was firstly presented by Mandl [2] and is a real Swiss road network which comprises of 15 nodes and 21 connections between them. This road network has been widely examined by many optimization approaches such as [10,15,35,50]. All data concerning node connections, transfer times between nodes (a number



Current particle

Personal or global best

Current particle after modification

Fig. 4. Example of *Method 1*.

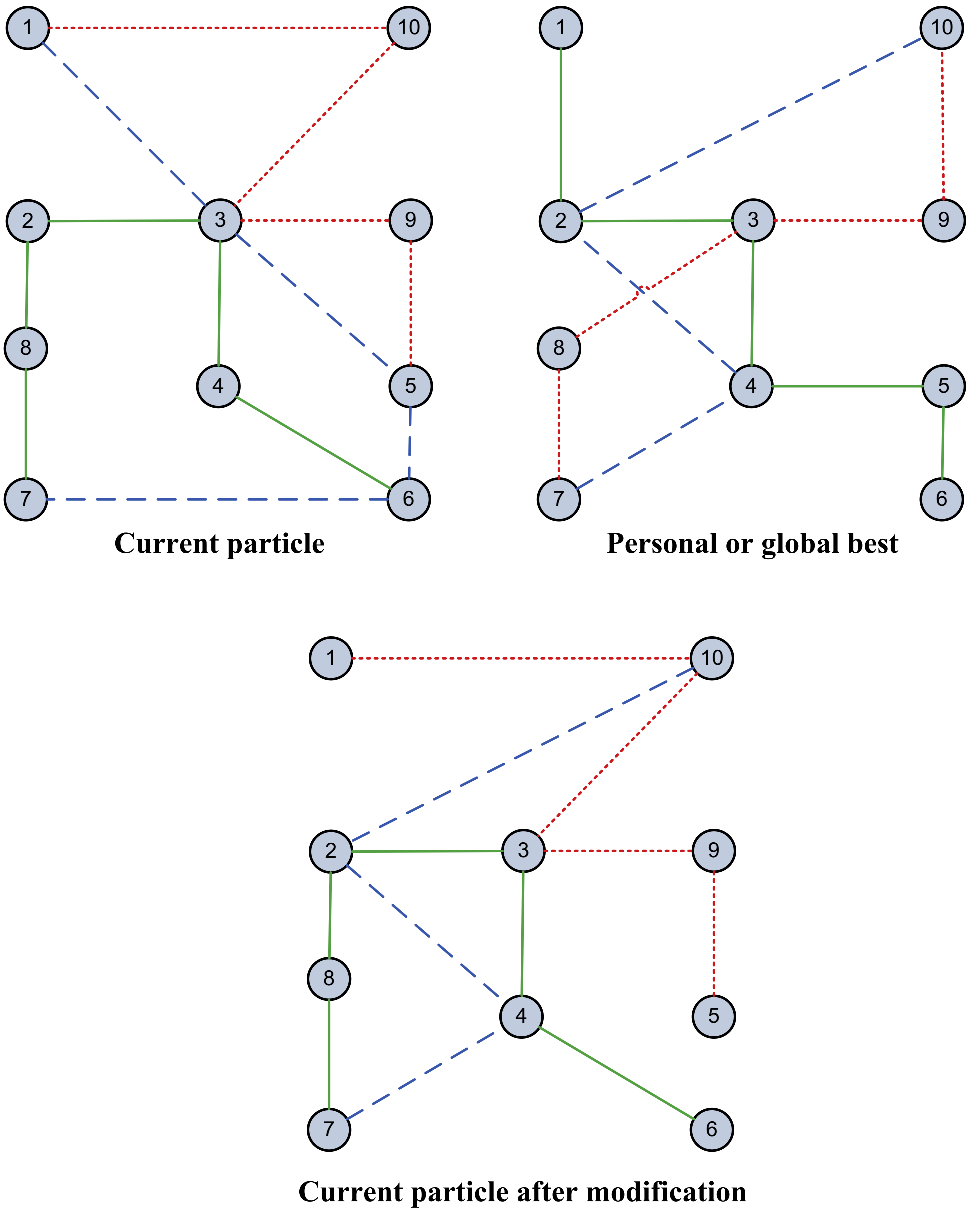
in an edge connecting two nodes is the transfer time between these two nodes) and transfer demands are presented in Fig. 7 and Table 4. Total transfer demands include 15,570 routes per day, while node 14 is the only node having zero activity level, that is, it neither generates nor attracts transfer demands. However, this node is essential for the network since it comprises an intermediate node of many routes and the only way to reach node 8. As stated in Section 3.2.2, all information concerning node connections and time distances between nodes is provided in form of an adjacency matrix.

4.2. Algorithm's parameters and evaluation criteria

The values used for algorithm's parameters are the ones proposed by Chakroborty and Wivedi [15] in order to have a fair comparison between all algorithms' results. More precisely, the parameter values used are the following:

- transfer penalty: 5 min
- weight values concerning scores $F_1(r)$, $F_2(r)$ and $F_3(r)$, that is, values of ω_1, ω_2 and ω_3: 1
- parameter values concerning scores $F_1(r)$, $F_2(r)$ and $F_3(r)$, that is, values of K_1, K_2, K_3 and K_4: $K_1 = K_2 = K_3 = 10$, $K_4 = 0$
- parameter values concerning $d_T(r)$, that is, values of a, b and c: $a = b = c = 1$
- particle swarm size: 50
- maximum number of nodes in each route: 8
- minimum number of nodes in each route: 3
- maximum route length at the initialization stage: 50 min
- total number of iterations: 200

The selection of the swarm size and the number of iterations is based on exhaustive experimental trials we conducted during the development of the algorithm. These two parameters clearly affect the computational cost of the proposed algorithm. We observed that decreasing excessively the size of the swarm, e.g. to values lower that 30, the search would stagnate some times to non-optimal solutions, where a high percentage of transfer demands would be satisfied with more than two transfers. Similarly, increasing the swarm size to values higher than 100, would affect excessively the computational cost, without increasing significantly the quality of results. As such, for the chosen Swiss network, a swarm size equal to 50 was chosen as a reasonable compromise. Of course, for a different network other values would have to be chosen.



Current particle

Personal or global best

Current particle after modification

Fig. 5. Example of *Method 2*.

Experimental results generated by the proposed PSO based algorithm are compared with the respective results published in the literature based on the evaluation criteria introduced in Section 3.3, which are the following:

- percentage of total transfer demands satisfied directly (d_0), with one transfer (d_1) or with two transfers (d_2) by the route set, which are used in order to calculate score $F_2(r)$ of the evaluation function.
- percentage of total transfer demands satisfied with more than two transfers or not satisfied at all by the route set (d_{unsat}), which is used in order to calculate score $F_3(r)$ of the evaluation function.
- average in-vehicle travel time (*ATT*), including time penalties for possible transfers, experienced by each passenger of the transit network, which is used in order to calculate score $F_1(r)$ of the evaluation function.

At this point, we have to mention that for each different test case 100 Monte Carlo runs were carried out and the respective results include the values of all evaluation criteria concerning the best solution (route set), the worst solution (route set), the average solution (route set), the median solution (route set),the respective standard deviation and the relative standard deviation.

4.3. *Computational complexity and experimental protocol*

The computational complexity of the proposed algorithm can be estimated as follows, assuming that the following parameters are defined:

- *gen_num*: the maximum number of generations
- *pop_size*: the size of the population of particles
- *max_routes*: the maximum number of routes allowed
- *max_nodes*: the maximum number of nodes in each route allowed
- *net_nodes*: the number of nodes of the whole transit network

As seen in Fig. 6 the proposed algorithm consists of five different procedures, namely, Initialization procedure, *Method 1*, *Method 2*,

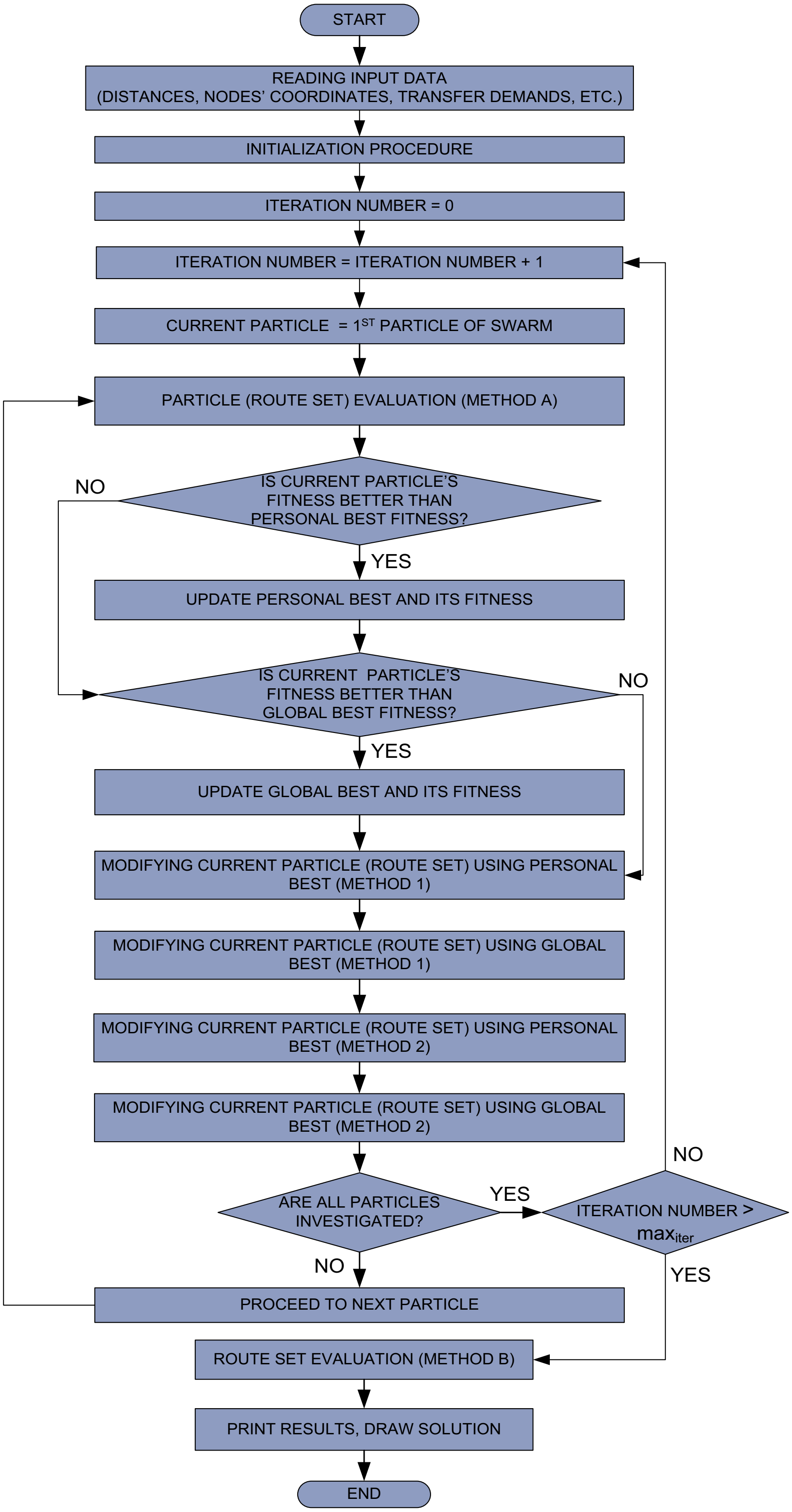

START
READING INPUT DATA
(DISTANCES, NODES' COORDINATES, TRANSFER DEMANDS, ETC.)
INITIALIZATION PROCEDURE
ITERATION NUMBER = 0
ITERATION NUMBER = ITERATION NUMBER + 1
CURRENT PARTICLE = 1ST PARTICLE OF SWARM
PARTICLE (ROUTE SET) EVALUATION (METHOD A)
IS CURRENT PARTICLE'S FITNESS BETTER THAN PERSONAL BEST FITNESS?
NO
YES
UPDATE PERSONAL BEST AND ITS FITNESS
IS CURRENT PARTICLE'S FITNESS BETTER THAN GLOBAL BEST FITNESS?
NO
YES
UPDATE GLOBAL BEST AND ITS FITNESS
MODIFYING CURRENT PARTICLE (ROUTE SET) USING PERSONAL BEST (METHOD 1)
MODIFYING CURRENT PARTICLE (ROUTE SET) USING GLOBAL BEST (METHOD 1)
MODIFYING CURRENT PARTICLE (ROUTE SET) USING PERSONAL BEST (METHOD 2)
MODIFYING CURRENT PARTICLE (ROUTE SET) USING GLOBAL BEST (METHOD 2)
ARE ALL PARTICLES INVESTIGATED?
YES
ITERATION NUMBER > maxiter
NO
NO
YES
PROCEED TO NEXT PARTICLE
ROUTE SET EVALUATION (METHOD B)
PRINT RESULTS, DRAW SOLUTION
END

Fig. 6. Flowchart of the proposed PSO based algorithm.

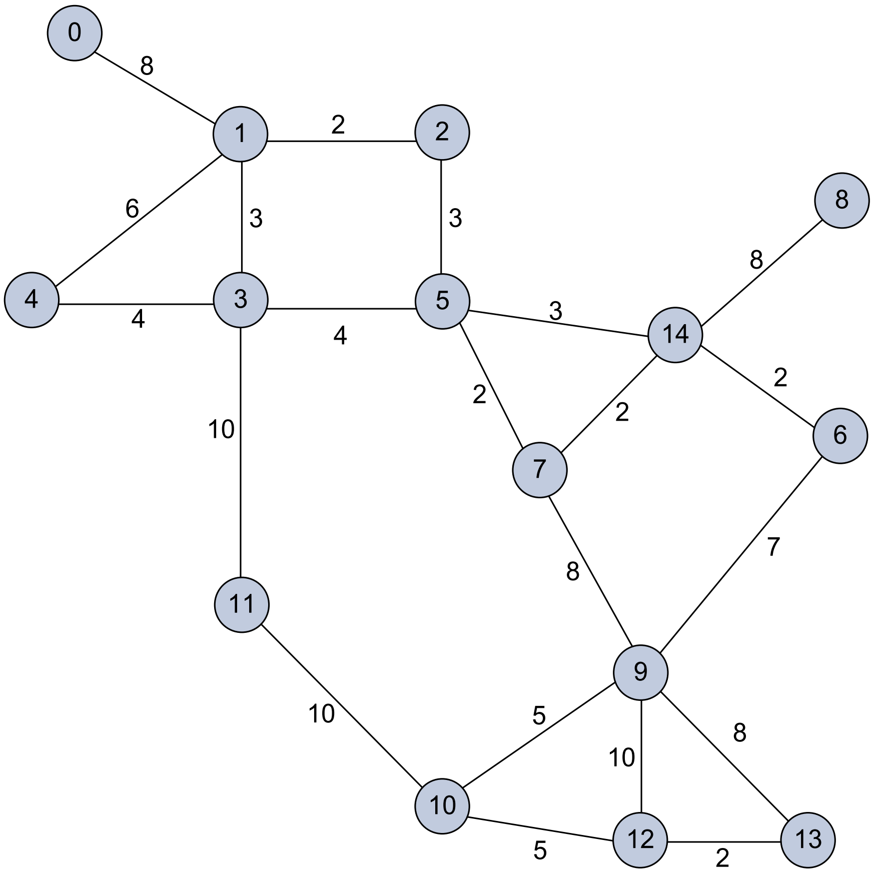


Fig. 7. The benchmark Swiss road network first presented by Mandl [4].

Table 5
Execution time (in seconds) of the proposed algorithm for route set design with 4, 6, 7 and 8 routes per set (STD: standard deviation, RSD: relative standard deviation).

Number of routes	Min	Avg	Median	Max	STD	RSD
4	230.82	280.76	284.35	314.54	24.65	8.78%
6	242.66	300.23	304.39	335.48	23.30	7.76%
7	287.80	321.62	323.21	347.52	16.13	5.01%
8	316.25	344.38	342.40	362.22	11.93	3.46%

Evaluation (*Method A*), Evaluation (*Method B*). The computational complexity of each one of these procedures is as follows:

- Initialization procedure: O(pop_size*net_nodes*max_routes*max_nodes)
- *Method 1*: O(gen_num*pop_size*max_nodes^4)
- *Method 2*: O(gen_num*pop_size)
- Evaluation (*Method A*): O(*gen_num*pop_size*^{2}*net_nodes*^{2}*max_routes*max_nodes*)
- Evaluation (*Method B*): O(*net_nodes*^{3}*max_routes*max_nodes*)

The procedure with the biggest asymptotical computational complexity which determines the computational complexity of the whole algorithm is *Method 1*. However, in practical applications, since *max_nodes* << *pop_size* and *max_nodes* ≤ *net_nodes* the procedure which determines the execution time of the proposed algorithm is *Method A*.

In Table 5 we present the execution time needed in seconds to run the experiments for cases of route set design with 4, 6, 7 and 8 routes per set. These data correspond to the runs conducted to generate the results presented in Tables 7, 9, 11 and 13. All runs were executed on a 1.83 GHz Intel® Core^{TM} Duo Processor T2400 with 1024 MB DDR2 533 MHz memory. Moreover, for all runs a swarm of 50 particles was used and 200 iterations were executed.

Table 6
The best route set, having four routes, constructed by the proposed algorithm.

Route 1:	13	9	6	14	5	3	1	0
Route 2:	8	14	6	9	10	11	3	1
Route 3:	0	1	4	3	5	7	9	10
Route 4:	0	1	2	5	7	9	10	12

4.4. Route set design with four, six, seven and eight routes

The best route set, having four routes, constructed by the proposed algorithm using the parameter values presented in Section 4.2 is presented in Fig. 8 and Table 6.

In Table 7 the results of the proposed algorithm are compared with the respective experimental results published in the literature for route sets consisting of four routes. As presented in Table 7 the route set constructed by the proposed algorithm has lower *ATT* compared to route sets constructed in [2,15,34,35,50,62,63]. Moreover, we have to mention that the values achieved for the rest of the evaluation criteria, that is, d_0, d_1, d_2, and d_{unsat}, by the proposed algorithm, are more or less the same as the ones achieved by other approaches.

The best route set, having six routes, constructed by the proposed algorithm using the parameter values presented in Section 4.2 is presented in Fig. 9 and Table 8.

In Table 9 the results of the proposed algorithm are compared with the respective experimental results published in the literature for route sets consisting of six routes. As presented in Table 9, the route set constructed by the proposed algorithm has lower

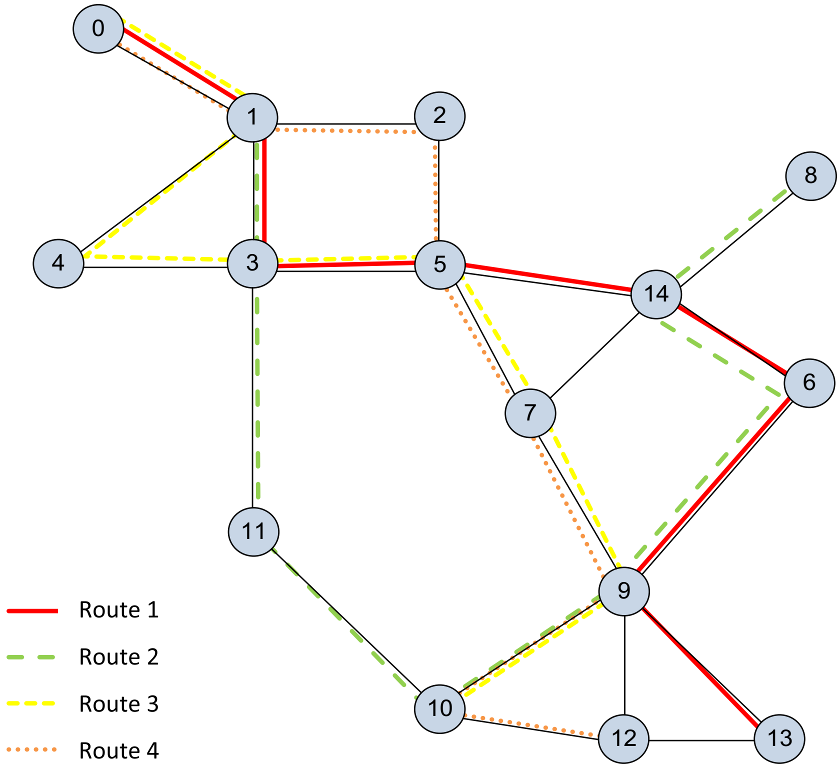


Fig. 8. The best route set, having four routes, constructed by the proposed algorithm.

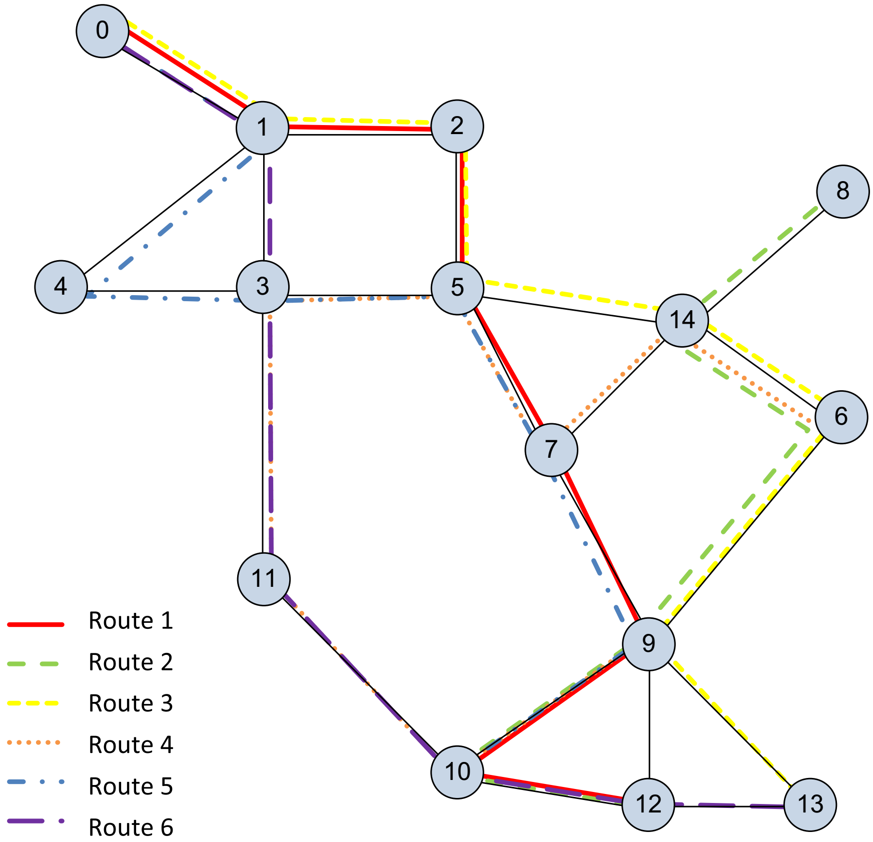


Fig. 9. The best route set, having six routes, constructed by the proposed algorithm.

Table 7
Comparing the best route set, having four routes, constructed by the proposed algorithm with route sets constructed by other approaches.

Algorithm	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	*ATT* (mpu[a])
Mandl [2]	69.94	29.93	0.13	0.00	12.90
Kidwai [34]	72.95	26.91	0.13	0.00	12.72
Chakroborty and Wivedi [15]	86.86	12.00	1.14	0.00	11.90
Fan and Mumford, best [35]	93.26	6.74	0.00	0.00	11.37
Fan and Mumford, HC[b], average [35]	91.83	8.17	0.00	0.00	11.69
Fan and Mumford, SA[c], average [35]	92.48	7.52	0.00	0.00	11.55
Fan, Mumford and Evans [50]	90.88	8.35	0.77	0.00	10.65
Zhang, Lu and Fan [62]	91.46	8.54	0.00	0.00	10.65
Chew and Lee, best [63]	93.71	6.29	0.00	0.00	10.82
Chew and Lee, average [63]	92.88	6.91	0.20	0.00	11.16
PSO best	91.84	7.64	0.51	0.00	10.64
PSO average	90.52	8.75	0.73	0.00	10.71
PSO median	90.56	8.67	0.83	0.00	10.67
PSO worst	90.56	8.61	0.83	0.00	10.85
PSO st. deviation	1.18	1.26	0.33	0.00	0.09
PSO RSD	1.30%	14.35%	45.70%	–	0.85%

[a] mpu: minutes per user.
[b] HC: hill climbing based algorithm.
[c] SA: simulated annealing based algorithm.

Table 8
The best route set, having six routes, constructed by the proposed algorithm.

Route 1:	0	1	2	5	7	9	10	12
Route 2:	8	14	6	9	10	12		
Route 3:	13	9	6	14	5	2	1	0
Route 4:	9	10	11	3	5	7	14	6
Route 5:	0	1	4	3	5	7	9	10
Route 6:	13	12	10	11	3	1	0	

Table 9
Comparing the best route set, having six routes, constructed by the proposed algorithm with route sets constructed by other approaches.

Algorithm	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	*ATT* (mpu)
Baaj and Mahmassani [10]	78.61	21.39	0.00	0.00	11.86
Kidwai [34]	77.92	19.62	2.40	0.00	11.87
Chakroborty and Wivedi [15]	86.04	13.96	0.00	0.00	10.30
Fan and Mumford, best [35]	91.52	8.48	0.00	0.00	10.48
Fan and Mumford, HC, average [35]	90.23	9.26	0.51	0.00	10.78
Fan and Mumford, SA, average [35]	90.87	8.74	0.39	0.00	10.65
Fan, Mumford and Evans [50]	93.19	6.23	0.58	0.00	10.46
Zhang, Lu and Fan [62]	91.12	8.88	0.00	0.00	10.50
Chew and Lee, best [63]	95.57	4.43	0.00	0.00	10.28
Chew and Lee, average [63]	93.85	5.88	0.24	0.03	10.51
PSO best	96.21	3.47	0.32	0.00	10.23
PSO average	95.62	4.28	0.10	0.00	10.28
PSO median	95.79	4.11	0.00	0.00	10.27
PSO worst	94.09	5.27	0.64	0.00	10.37
PSO st. deviation	0.60	0.58	0.16	0.00	0.05
PSO RSD	0.63%	13.59%	152.74%	–	0.45%

Table 10
The best route set, having seven routes, constructed by the proposed algorithm.

Route 1:	8	14	5	2	1	3	11	
Route 2:	2	1	4	3	5	14	6	9
Route 3:	0	1	2	5	7	9	10	12
Route 4:	0	1	4	3	5	7	9	10
Route 5:	8	14	6	9	13	12	10	11
Route 6:	10	9	6	14	5	2	1	0
Route 7:	9	10	11	3	5	7	14	6

Table 11
Comparing the best route set, having seven routes, constructed by the proposed algorithm with route sets constructed by other approaches.

Algorithm	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	*ATT* (mpu)
Baaj and Mahmassani [10]	80.99	19.01	0.00	0.00	12.50
Kidwai [34]	93.91	6.09	0.00	0.00	10.70
Chakroborty and Wivedi [15]	89.15	10.85	0.00	0.00	10.15
Fan and Mumford, best [35]	93.32	6.36	0.32	0.00	10.42
Fan and Mumford, HC, average [35]	92.21	7.13	0.66	0.00	10.74
Fan and Mumford, SA, average [35]	92.47	6.95	0.58	0.00	10.62
Fan, Mumford and Evans [50]	92.55	6.68	0.77	0.00	10.44
Zhang, Lu and Fan [62]	92.89	7.11	0.00	0.00	10.46
Chew and Lee, best [63]	95.57	4.43	0.00	0.00	10.27
Chew and Lee, average [63]	96.47	3.53	0.00	0.00	10.31
PSO best	97.17	2.83	0.00	0.00	10.16
PSO average	96.55	3.45	0.01	0.00	10.23
PSO median	96.60	3.40	0.00	0.00	10.20
PSO worst	95.63	4.37	0.00	0.00	10.40
PSO st. deviation	0.48	0.48	0.02	0.00	0.06
PSO RSD	0.50%	13.92%	346.11%	–	0.63%

ATT compared to route sets constructed in [10,15,34,35,50,62,63]. Moreover, we have to mention that the percentage of total transfer demands satisfied directly (d_0), achieved by the proposed algorithm, is significant higher compared to the ones achieved by all other approaches and reaches 96.21%.

The best route set, having seven routes, constructed by the proposed algorithm using the parameter values presented in Section 4.2 is presented in Fig. 10 and Table 10.

In Table 11 the results of the proposed algorithm are compared with the respective experimental results published in the literature for route sets consisting of seven routes. As presented in Table 11, the route set constructed by the proposed algorithm has lower *ATT* compared to route sets constructed in [10,34,35,50,62,63] and slightly bigger compared to the route set constructed in [15]. Moreover, we have to mention that the percentage of total transfer demands satisfied directly (d_0), achieved by the proposed algorithm is far higher compared to the ones achieved by all other approaches and reaches 97.17%.

The best route set, having eight routes, constructed by the proposed algorithm using the parameter values presented in Section 4.2 is presented in Fig. 11 and Table 12.

In Table 13 the results of the proposed algorithm are compared with the respective experimental results published in the literature for route sets consisting of eight routes. As presented in Table 13, the route set constructed by the proposed algorithm has lower *ATT* compared to all route sets constructed in [10,15,34,35,50,62] and [63]. Moreover, we have to mention that the percentage of total transfer demands satisfied directly (d_0), achieved by the proposed algorithm is far higher compared to the ones achieved by all other approaches, except for [63], and reaches 97.75%.

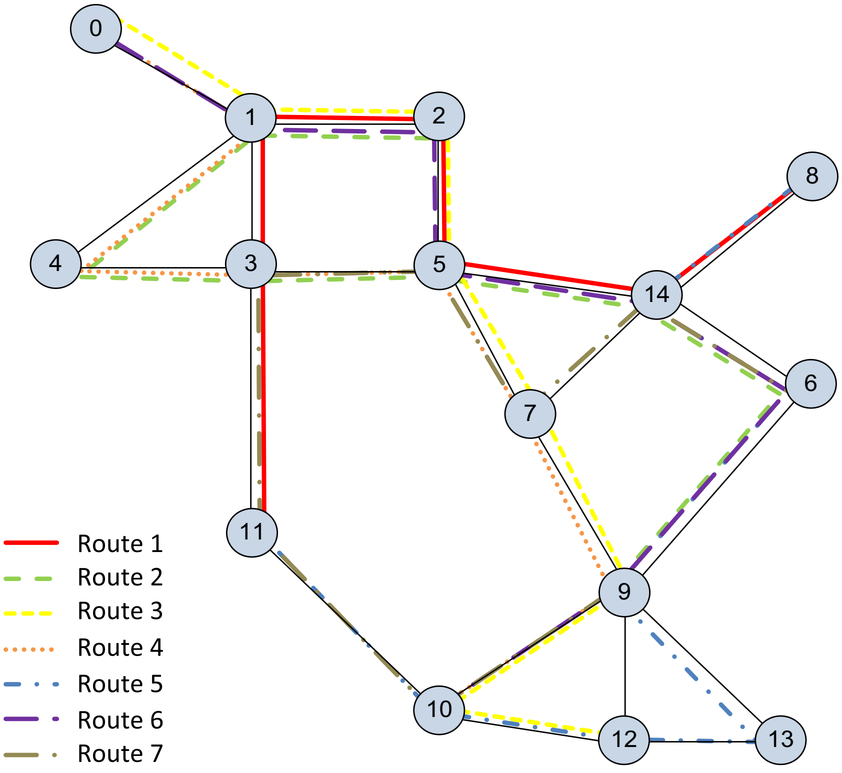


Fig. 10. The best route set, having seven routes, constructed by the proposed algorithm.

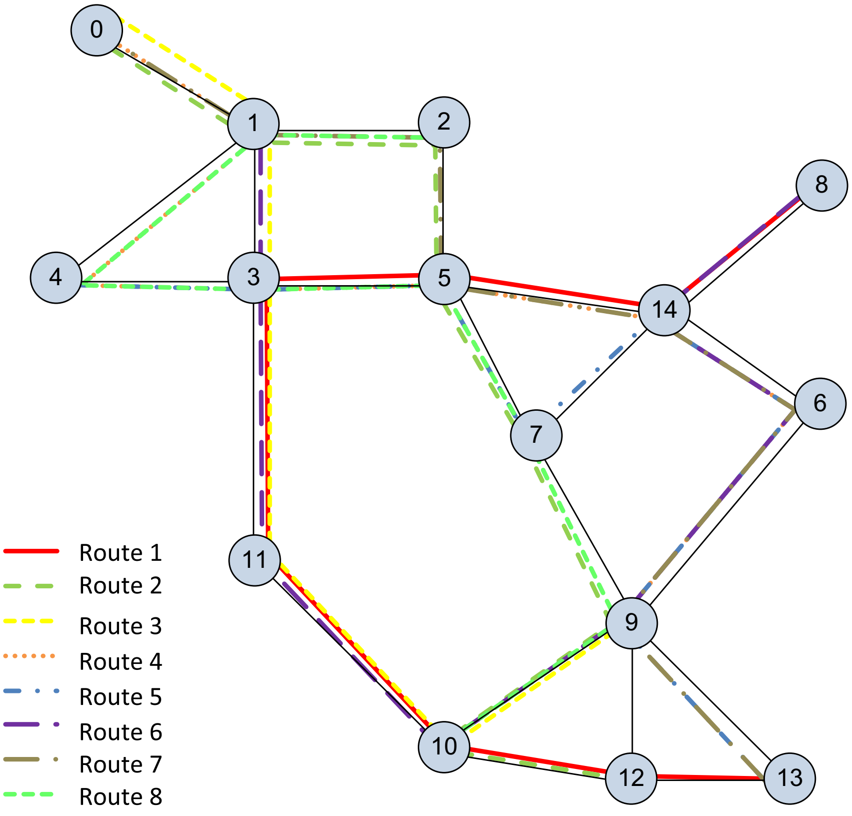


Fig. 11. The best route set, having eight routes, constructed by the proposed algorithm.

Table 12
The best route set, having eight routes, constructed by the proposed algorithm.

Route 1:	8	14	5	3	11	10	12	13
Route 2:	0	1	2	5	7	9	10	12
Route 3:	0	1	3	11	10	9		
Route 4:	0	1	4	3	5	14	6	9
Route 5:	13	9	6	14	7	5	3	4
Route 6:	8	14	6	9	10	11	3	1
Route 7:	13	9	6	14	5	2	1	0
Route 8:	2	1	4	3	5	7	9	10

Table 13
Comparing the best route set, having eight routes, constructed by the proposed algorithm with route sets constructed by other approaches.

Algorithm	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	*ATT* (mpu)
Baaj and Mahmassani [10]	79.96	20.04	0.00	0.00	11.86
Kidwai [34]	84.73	15.27	0.00	0.00	11.22
Chakroborty and Wivedi [15]	90.38	9.58	0.00	0.00	10.46
Fan and Mumford, best [35]	94.54	5.46	0.00	0.00	10.36
Fan and Mumford, HC, average [35]	93.23	6.18	0.59	0.00	10.69
Fan and Mumford, SA, average [35]	93.65	5.88	0.47	0.00	10.58
Fan, Mumford and Evans [50]	91.33	8.67	0.00	0.00	10.45
Zhang, Lu and Fan [62]	93.14	6.86	0.00	0.00	10.42
Chew and Lee, best [63]	97.82	2.18	0.00	0.00	10.19
Chew and Lee, average [63]	96.16	3.84	0.00	0.00	10.31
PSO best	97.75	2.25	0.00	0.00	10.13
PSO average	97.47	2.53	0.00	0.00	10.17
PSO median	97.40	2.60	0.00	0.00	10.15
PSO worst	96.79	3.21	0.00	0.00	10.20
PSO st. deviation	0.45	0.45	0.00	0.00	0.04
PSO RSD	0.47%	17.92%	–	–	0.37%

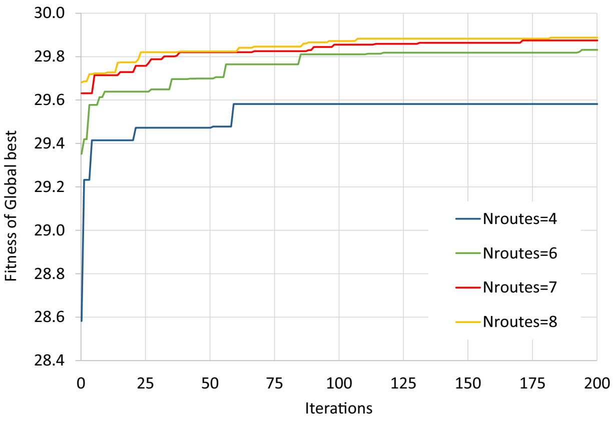


Fig. 12. Evolution of evaluation function value of global best solution against number of iterations.

4.5. Global optimization aspects

In Fig. 12 the evolution of the evaluation function value of the global best solution against the number of iterations is presented for the cases of 4, 6, 7 and 8 routes in the route set. For all cases a swarm population of 50 particles is considered, while all of the remaining initialization parameters are the same for all cases and as described in Section 4.2. The first point displayed in this plot is the global best of the population generated right after the initialization procedure. One can observe that the global best fitness of the initial population is getting bigger as the number of routes in the network increases. This is logical, since a higher number of routes of the same length (max 8 nodes) can cover the demands of a given network more satisfactorily. This indicates also the efficiency of the proposed initialization procedure. Except for that, it should be noticed that the final fitness of the global best particle after 200 iterations depends also on the number of routes in the network, for similar reasons. A sharp increase in the fitness of the global best particle is observed during the initial iterations, while afterwards a more gradual increase occurs. Of course, it should be mentioned that during each iteration, especially for the initial ones, the fitness of the global best particle increases multiple times and not just once, since multiple modified particles might improve on its value. This is the main reason that the initial steps seem so abrupt.

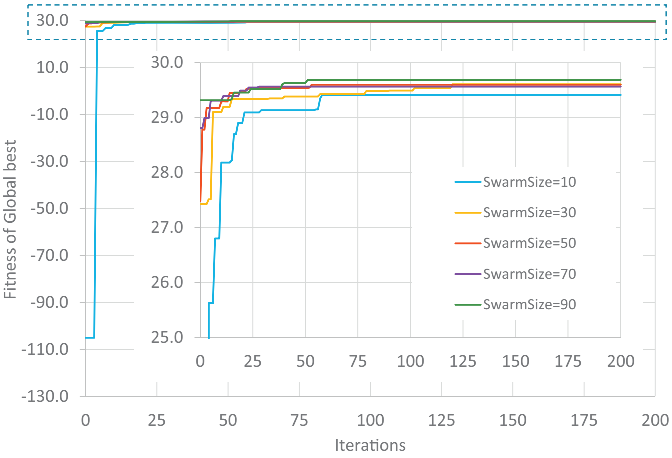


Fig. 13. Evolution of evaluation function value of global best solution against number of particles in the swarm (case of four routes).

Looking at Fig. 12 it may appear that the absolute value of the fitness function is not evolving significantly. However, the actual evolution of the quality of the route set is clearly significant. For example, in Fig. 12 for the case of 4 routes the fitness function progresses from 28.6 to 29.6. This change represents actually an evolution of d_0, d_1 and d_2 (the percentages of passengers traveling from their origin to their destination with zero, one and two transfers respectively) from 76.75%, 16.12% and 7.13% to 87.35%, 11.50% and 1.16%, while at the same time the average travel time *ATT* drops from 12.89 min to 10.84 min. This evolution is clearly much more important for the quality of the route set than what the change in the absolute value of the fitness would suggest.

A similar graph is presented in Fig. 13, where the evolution of the evaluation function value of the global best solution against the number of particles in the swarm is presented for the case of four routes. Again, one can observe that increasing the size of the swarm leads to a higher fitness function value both at the initial iterations and the last one. Of course, as seen in Fig. 13, after increasing the swarm population sufficiently the global fitness appears to converge to similar values for all cases. On the contrary, if we use a relative small swarm size, it is possible that the search will be trapped in local optima, given the small diversity of the initial population. It is also very important to note again that the algorithm successfully manages to quickly improve the fitness of the population even for the “difficult” case of swarm size equal to 10. This proves that, even though the initialization procedure is clearly important, generating most of the times already good quality solutions, as also seen in Fig. 12, the PSO procedure is also very well capable of improving low quality populations to high quality solutions.

At this point, we have to make clear that convergence of the proposed algorithm cannot be proven theoretically. Like with most metaheuristic methods, a sufficiently good solution can most probably be obtained using the proposed PSO algorithm across many data sets. However, there is no guarantee that a globally optimal solution has been found. The algorithm does not explicitly "converge" to a solution, in the sense that it does not terminate on account of e.g. reaching certain tolerance criteria. As mentioned also in Section 4.2, the method terminates after conducting a certain amount of user defined iterations. This number has been selected empirically after extensive testing of the algorithm at varying combinations of parameters. In summary, as seen in Fig. 13 a close enough to the global optimum solution can probably be found provided an adequate amount of iterations are allowed.

4.6. *Computational cost comparison*

Unfortunately, a very limited number of researchers reports detailed performance data, making a direct comparison of computational costs difficult. Fan and Mumford do report data for their studied methods, which were two, namely simulated annealing (SA) and hill-climbing (HC). Specifically, for HC they used 100,000 iterations, while for SA they performed 1000 cooling steps, with 100 iterations within the inner loop. Given the different nature of the algorithms, a direct comparison of iterations is probably not indicative, but in any case in our work we conducted 200 iterations using a swarm of 50 particles.

Fan and Mumford report also average run times, which for HC were around 250 s, while for SA these were lower and about 90 s. In our case, for the four cases studied (networks of 4, 6, 7 and 8 routes) these run times fluctuated from about 280 s to 340 s. Thus, the order of magnitude of our results is similar to the results of Fan and Mumford. Moreover, as seen by the performance graphs in Figs. 12 and 13, near optimal results are achieved in most of the times already from iteration 100. Hence, we believe that the comparisons are indeed fair in computational costs.

For the work of Chakroborty and Wivedi [15] the only information available is that they used 20 chromosomes in their GA based approach. In the work of Chew and Lee [63], which is also GA based, 100 chromosomes were used. The authors reported the following performance data: run times were from 20 to 40 s, while generations spanned from 60 to 90. Hence, the method of Chew and Lee is in general faster to ours, however our results are better. Zhang, Lu and Fan [62] only report that they used a population size of 500, while iterations spanned from 2000 to 8000. In their previous work, Fan, Mumford and Evans [50] used a population size of 200 and iterations from 1000 to 5000.

5. Analysis and discussion

5.1. *Trade-off among different objectives*

One can observe that the problem faced in this paper is a multi-objective one. The weights, ω_i, of each score, F_i, in equation 3 can be set independently to emphasize the importance of the route set characteristic they evaluate. Moreover, the parameters K_i of each F_i score provide an additional implicit way to allocate higher or lower importance in a specific score, since these parameters represent the maximum values the F_i scores can get. In the current contribution, as mentioned also in Section 4.2, the values of the weights are all taken equal to 1, while the maximum score values are all taken equal to 10 to be in line with relevant literature. As such, the objective function has a maximum possible value equal to 30.

It should be highlighted though that scores F_1, F_2 and F_3 are not exactly competing with each other. F_1 is a score which reflects the average time spent by each passenger when he/she travels using a specific route set. F_2 is a score which reflects the percentage of passengers traveling from their origin to their destination either directly, or by making one or two transfers. Finally, F_3 is a score which reflects the percentage of passengers who travel from their origin to their destination with more than two transfers.

All these scores approach their maximum value (10) when the quality of the route set improves. For example, the higher the number or passengers who travel with 0, 1 and 2 transfers the lower the percentage of passengers who transfer more than 2 times. Hence, both scores F_2 and F_3 would increase in parallel. The effect of F_1 is not so straightforward though. It is logical to assume that by minimizing the number of transfers the average travel time would also decrease, since there would be fewer penalties for waiting for transfers. However, it is possible that a route set can satisfy travel demands with no transfers by generating very long routes. In that case, the travel time would increase since connections between nearby nodes would be achieved via long "peripheral" routes. This is a very likely design possibility at bigger networks. Score F_1 by trying to minimize the average travel time avoids this route set design possibility. Hence, we believe it is necessary to include all scores in the fitness function. In any case, in current work, the restriction of maximum length of each route to 8 nodes effectively avoids this possibility. Hence, scores F_1, F_2 and F_3 are indeed not competing.

In conclusion, we should mention that similar criteria and equivalent scores are utilized by the majority of relevant literature, as also seen in our comparison tables. Even if the fitness function of other works does not include explicitly scores F_1, F_2 and F_3, it always includes at least one element that corresponds to the number of transfers (scores F_2 and F_3 in our case) and one element that accounts for the average travel time (score F_1 in our case).

The multi-objective nature of the developed algorithm is more clearly visible by the consideration of score F_4. This score depends on the total length of the entire route set, which in turn can be correlated to operational costs, such as the average fuel consumption. The lower the total route set length, the higher the score of F_4. As such, F_4 introduces in the objective function the point of view of the service provider and is competing with the other F_i scores. The influence of this score on the quality of the results obtained by the proposed algorithm is extensively studied in Section 5.2. Experimental results show that the algorithm can indeed be effectively used to construct route sets that balance the conflicting goals by choosing its parameters accordingly. For all experimental results in Section 4 the weight of F_4 was set equal to zero, so as to obtain optimal results from the passengers' point of view and be able to compare with other approaches published in the literature.

5.2. *Investigating the effect of operational cost to route set design*

As mentioned in Section 3.3.4, the evaluation of a route set can be performed either from the passengers' or from the companies' point of view. In fact, score $F_4(r)$ represents the operational profit of transport companies and is contradictory to scores $F_1(r)$, $F_2(r)$ and $F_3(r)$ which represent the needs of the traveling public. In all experimental results presented in Section 4, score $F_4(r)$ was deactivated so as not to affect the value of total fitness Eq. (3). As a result, all routes are designed in order to serve the traveling public as good as possible. In this section, however, we investigate the effect of the total length of all routes since this represents a more direct profit for transport companies.

Score F_4 is summed in a similar way to the rest of the F_i scores ($TOTFIT = \omega_1 {}^*F_1 + \omega_2 {}^*F_2 + \omega_3 {}^*F_3 + \omega_4 {}^*F_4$.). The ω_4 weight, in line with the other weights, is set equal to 1. As explained in Section 5.1, the implementation of these weights can be considered a bit redundant, given that a similar prioritization can be achieved by properly defining the respective K_i values. As seen in Sections 3.3.1–3.3.4, the K_i values represent the maximum values the respective F_i scores can

Table 14
Investigating the effect of operational cost to route set design when maximum number of nodes per route is restricted to 8 (number of routes = 4).

x_{4m} value	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	ATT (mpu)	L_{tot} (min)
x_{4m} = 20	68.45	24.73	6.51	0.30	13.69	75.53
x_{4m} = 40	68.68	25.35	5.83	0.13	13.79	76.03
x_{4m} = 60	71.71	24.06	4.12	0.11	13.18	77.00
x_{4m} = 80	75.37	21.97	2.47	0.18	12.40	80.20
x_{4m} = 100	84.28	14.92	0.33	0.46	11.41	98.50
x_{4m} = 120	89.66	9.60	0.65	0.10	10.94	116.77
x_{4m} = 140	90.21	9.13	0.66	0.00	10.77	133.04
x_{4m} = 160	89.96	9.11	0.93	0.00	10.71	141.70
x_{4m} = 180	90.66	8.61	0.73	0.00	10.72	143.88
x_{4m} = 200	90.47	8.71	0.82	0.00	10.71	139.96
F_4 = 0 (only passengers' needs are taken into consideration)	90.52	8.75	0.73	0.00	10.71	142.31

obtain. Hence, by modifying the K_i values an analogous emphasis can be put on a specific score. The same result can be achieved by varying the respective ω_i weights. As such, all of these weights were kept equal to 1 in the current implementation.

5.2.1. *Setting maximum number of nodes per route equal to 8*

In this section, we set the value of parameter K_4 equal to 10, while the value of parameter x_{4m} varies from 20 to 200 min. In Table 14, the average values of all evaluation criteria are presented, after conducting 100 Monte Carlo runs, for number of routes equal to four. It is obvious that in order to minimize companies' operational costs, the needs of traveling public are poorly served. More precisely, as the value of x_{4m} varies from 200 to 20 min the average traveling time (ATT) increases while at the same time the percentage of total transfer demands satisfied directly (d_0) decreases. On the contrary, a significant decrease in the total length of all routes (L_{tot}) is achieved which reaches the value of 75.53 min in case x_{4m} equals 20 min. It is worth noticing that the algorithm manages to satisfy all traveling demands while x_{4m} is bigger than 120 min. Setting x_{4m} to values equal or lower than 120 min leads the algorithm to solutions where some traveling demands are satisfied with more than two transfers or they are not satisfied at all, that is, to solutions where the value of criterion d_{unsat} is bigger than zero. Finally, we can also notice that setting x_{4m} to values bigger than 140 min does not improve the algorithm's results. Having bigger total length of all routes values (L_{tot}) does not lead to solutions of higher quality, since the maximum number of nodes per route, for this set of experiments, is restricted to 8.

In Table 15 and Fig. 14 we present the best route set, having four routes, constructed by the proposed algorithm if the value of parameter x_{4m} is set to 20 min. The significant decrease on the operational costs achieved is obvious, since the number of nodes per route is significant smaller compare to the route set presented in Table 6 and Fig. 8.

5.2.2. *Setting maximum number of nodes per route equal to 10*

In this section, we set the value of parameter K_4 equal to 10, while the value of parameter x_{4m} varies from 80 to 160 min. In Table 16, the average values of all evaluation criteria are presented, after conducting 100 Monte Carlo runs. It is worth noticing that the algorithm constructs route sets where some traveling demands are satisfied with more than two transfers or they are not satisfied at all, if the value of parameter x_{4m} is equal or smaller than 80 min, while in the previous case (Section 5.2.1), where the maximum number of nodes per route was restricted to 8, this problem occurred when the value of parameter x_{4m} was equal or smaller than 120 min. In general, all results presented in Table 16, for different values of x_{4m}, are equivalent or better to the respective ones presented in Table 14.

Table 15
The best route set, having four routes, constructed by the proposed algorithm for x_{4m} = 20 min.

Route 1:	8	14	5	3	4	
Route 2:	13	12	10			
Route 3:	5	7	9	10	11	
Route 4:	0	1	2	5	14	6

Table 16
Investigating the effect of operational cost to route set design when maximum number of nodes per route is restricted to 10.

x_{4m} value	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	ATT (mpu)	L_{tot} (min)
x_{4m} = 80	74.07	23.50	2.38	0.05	13.40	82.45
x_{4m} = 100	82.30	16.34	1.36	0.00	13.06	95.20
x_{4m} = 120	87.63	11.53	0.84	0.00	11.48	114.05
x_{4m} = 140	90.56	8.80	0.65	0.00	10.90	131.60
x_{4m} = 160	90.78	8.60	0.62	0.00	10.83	142.40
F_4 = 0 (only passengers' needs are taken into consideration)	91.17	8.22	0.62	0.00	10.65	168.83

It is obvious that setting proper values to algorithm's parameters will enable the proposed algorithm to construct quality route sets that could fulfill all passengers' demands while at the same time decrease operational costs. However, designing route sets with very small route lengths should be avoided, since it can lead to other problems such as higher frequency of bus schedules.

5.3. *Investigating the effect of maximum route length (number of nodes) to route set design*

As described in Section 5.2 the total length of all routes (L_{tot}) affects directly the quality of the transportation service offered to traveling public. However, as mentioned in Section 1, the length of each route is also affected by many regulations and policies, for example, concerning drivers' working hours. In this section, we investigate the affect of the maximum number of nodes per route, using the road network of Mandl with four routes as an example. In Table 17, the average values of all evaluation criteria are presented, after conducting 100 Monte Carlo runs, with maximum number of nodes per route varying from 3 to 14 nodes.

As presented in Table 17 the maximum number of nodes affects significantly the quality of results. If the number of nodes is set to values smaller than 8, the algorithm constructs solutions where some traveling demands are satisfied with more than two transfers or they are not satisfied at all, that is, solutions where the value of criterion d_{unsat} is bigger than zero. At the same time, the

Table 17
Investigating the effect of maximum route length (number of nodes per route) to route set design.

Maximum number of nodes	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	ATT (mpu)	L_{tot} (min)
3 nodes	43.19	11.11	0.00	45.70	–	47.50
4 nodes	62.21	15.03	0.60	22.17	–	68.17
5 nodes	68.91	19.96	3.38	7.74	14.14	91.73
6 nodes	78.66	17.70	1.38	2.27	12.66	108.57
7 nodes	85.50	13.12	0.86	0.52	11.00	131.73
8 nodes	90.12	9.20	0.68	0.00	10.76	142.36
9 nodes	92.52	6.88	0.60	0.00	10.56	160.20
10 nodes	93.55	6.00	0.45	0.00	10.48	177.83
11 nodes	94.06	5.60	0.35	0.00	10.42	189.30
12 nodes	94.53	5.30	0.16	0.00	10.40	197.50
13 nodes	95.21	4.64	0.15	0.00	10.38	197.23
14 nodes	95.16	4.63	0.20	0.00	10.40	200.27

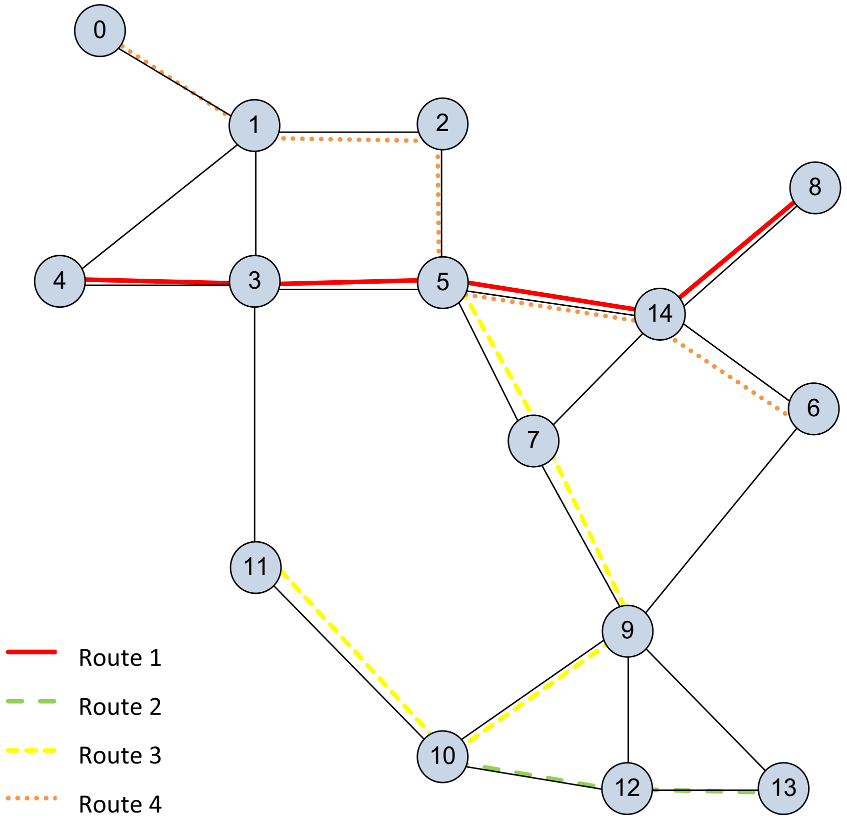


Fig. 14. The best route set, having four routes, constructed by the proposed algorithm for $x_{4m} = 20$ min.

percentages of transfer demands satisfied with one transfer (d_1) or with two transfers (d_2) are increased. This, however, leads to a significant increase in the average in-vehicle travel time (*ATT*) due to the waiting time added for each transfer (as stated in Section 3.3.1 a penalty of 5 min is added for each transfer). On the other hand, the total length of all routes (L_{tot}) is decreased if the maximum number of nodes per route is restricted. Especially, if the maximum number of nodes per route equals 3 or 4 the value of criterion *ATT* cannot be estimated, since there are unsatisfied travel demands (some nodes of the road network are not included in the resulting route set).

If the number of nodes is set to values bigger than 8, the improvement in the quality of the resulting route sets is obvious. The percentage of total transfer demands satisfied directly (d_0) is increased as the maximum number of nodes per route is increased. Also, the respective percentages of transfer demands satisfied with one transfer (d_1) or with two transfers (d_2) are decreased. However, as seen in Table 17, setting the number of maximum nodes per route to a value equal or higher than 11 does not affect the quality of the resulting route sets significantly (the values of criteria d_0 and *ATT* are more or less stable). This demonstrates experimentally that increasing the length of routes alone is not enough in order to improve the quality of transport services.

5.4. Investigating the effect of parameters a, b and c to route set design

In this section we investigate whether the service of the traveling public can be improved using proper values for parameters concerning $d_T(r)$, that is, parameters *a*, *b* and *c* (Section 3.3.2). These parameters represent weights that relate to the percentage of passengers traveling from their origin to their destination making zero, one or two transfers, respectively. As stated in Section 3.3.2, parameters a, b and c are user defined satisfying inequality $a \geq b \geq c$.

Table 18
Investigating the effect of weight *a* to route set design when maximum number of nodes per route is restricted to 8.

Weight *a*	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	*ATT* (mpu)	L_{tot} (min)
1	90.31	8.82	0.87	0.00	10.74	141.35
2	91.27	8.04	0.69	0.00	10.74	140.13
3	91.31	8.13	0.56	0.00	10.77	142.35
4	91.14	8.20	0.65	0.00	10.73	139.85
5	91.36	7.98	0.66	0.00	10.77	140.30
6	91.34	7.98	0.68	0.00	10.73	142.40
7	91.69	7.59	0.72	0.00	10.73	141.33
8	91.67	7.64	0.70	0.00	10.81	137.23
9	91.44	7.81	0.75	0.00	10.76	139.44
10	91.59	7.81	0.60	0.00	10.81	141.12

Experimental results presented in the following sections concern only weight *a*. Similar results hold also for the rest two weights, namely, *b* and *c*.

5.4.1. Setting maximum number of nodes per route equal to 8

In Table 18, the average values of all evaluation criteria are presented, after conducting 100 Monte Carlo runs, with parameter *a* varying from 1 to 10 while the rest of the parameters where set equal to 1. Experimental results demonstrate that the value of parameter *a* does not affect the quality of the resulting route sets. Increasing parameter *a* from 1 to 10 just causes a slight increase in the value of d_0 with a respective slight decrease in values of d_1 and d_2.

5.4.2. Setting maximum number of nodes per route equal to 10

In Table 19, the average values of all evaluation criteria are presented, after conducting 100 Monte Carlo runs, with parameter *a* varying from 1 to 10 while the rest of the parameters where set equal to 1. However, this time the maximum number of nodes per

Table 19
Investigating the effect of weight *a* to route set design when maximum number of nodes per route is restricted to 10.

Weight *a*	d_0 (%)	d_1 (%)	d_2 (%)	d_{unsat} (%)	*ATT* (mpu)	L_{tot} (min)
1	93.42	6.00	0.58	0.00	10.46	175.07
2	94.67	4.97	0.36	0.00	10.93	176.33
3	95.03	4.66	0.31	0.00	11.39	173.50
4	95.66	4.02	0.31	0.00	11.71	176.53
5	96.92	2.95	0.13	0.00	12.67	184.80
6	96.49	3.32	0.19	0.00	12.34	179.23
7	96.98	2.87	0.14	0.00	12.55	181.80
8	96.82	3.03	0.15	0.00	12.58	181.53
9	97.52	2.38	0.09	0.00	13.21	185.30
10	96.97	2.87	0.16	0.00	12.66	182.70

route equals 10. As presented in Table 19, the value of parameter *a* affects the quality of the resulting route sets. Increasing parameter *a* from 1 to 10 causes a significant increase in the value of d_0 with a respective decrease in values of d_1 and d_2. At the same time, a respective increase in the values of parameters *ATT* and L_{tot} occurs. If the maximum number of nodes per route is set to a larger value (for example 10 instead of 8) setting the value of weight *a* to larger values leads to significant improvement of the resulting route sets.

6. Conclusions and future work

In this contribution a novel PSO based algorithm has been designed, implemented and applied to the Urban Transit Routing Problem (UTRP) in order to construct feasible and efficient route networks for public transit networks. Results are compared on the basis of Mandl's benchmark problem of a Swiss bus network, which is probably the widely investigated and accepted benchmark problem in the relevant literature. Computational results showed that the proposed PSO based algorithm achieves competitive results compared to seven other very effective methods published in the literature. Finally, the application of the proposed PSO based algorithm to other UTRP available data sets, as well as to other scheduling problems, such as the Urban Transit Scheduling Problem (UTSP), will be one of the main issues of our future work.

References

[1] P. White, Public Transport: Its Planning, Management and Operation, 4th ed., Spon Press, London, 2002.

[2] C.E. Mandl, Applied Network Optimization, Academic Press, London, 1979.

[3] Q.F. Zeng, K.C. Mouskos, Heuristic Search Strategies to Solve Transportation Network Design Problems, Final Report, Department of Transportation, New Jersey, 1997.

[4] S. Stephen, Z. Liu, China's urban transport development strategy, in: Proceedings of a Symposium in Beijing, The World Bank, Washington, DC, 1996, pp. 352–855.

[5] W. Fan, R.B. Machemehl, Optimal Transit Route Network Design Problem: Algorithms, Implementations and Numerical Results, Research Report SWUTC/04/167244-1, Centre for Transportation Research, University of Texas at Austin, 2004.

[6] Cube Voyager | Citilabs. http://www.citilabs.com/products/cube/cube-voyager (accessed 04.02.13).

[7] INRO>Emme transportation forecasting software. http://www.inrosoftware.com/en/products/emme/index.php (accessed 04.02.13).

[8] ATKINS–SATURN. http://www.saturnsoftware.co.uk/index.html (accessed 04.02.13).

[9] Vision Traffic–PTV Group. http://www.ptv-vision.com/en-uk/products/vision-traffic-suite/ptv-visum/overview/ (accessed 04.02.13).

[10] M.H. Baaj, H.S. Mahmassani, An AI-based approach for transit route system planning and design, J. Adv. Transp. 25 (2) (1991) 187–209.

[11] M.H. Baaj, H.S. Mahmassani, Hybrid route generation heuristic algorithm for the design of transit networks, Transp. Res Part C 3 (1) (1995) 31–50.

[12] S.B. Pattnaik, S. Mohan, V.M. Tom, Urban bus transit route network design using genetic algorithm, J. Transp. Eng. 124 (4) (1998) 368–375.

[13] P. Chakroborty, Genetic algorithms for optimal urban transit network design, Comput.-Aided Civil Infrastruct. Eng. 18 (3) (2003) 184–200.

[14] P. Chakroborty, K. Deb, P.S. Subrahmanyam, Optimal scheduling of urban transit systems using genetic algorithms, J. Transp. Eng. 121 (6) (1995) 544–553.

[15] P. Chakroborty, T. Wivedi, Optimal Route Network Design for transit systems using genetic algorithms, Eng. Optimiz. 34 (1) (2002) 83–100.

[16] W. Fan, R.B. Machemehl, Optimal transit route network design problem with variable transit demand: genetic algorithm approach, J. Transp. Eng. 132 (1) (2006) 40–51.

[17] W. Fan, R.B. Machemehl, Using a simulated annealing algorithm to solve the transit route network design problem, J. Transp. Eng. 132 (2) (2006) 122–132.

[18] W. Fan, R.B. Machemehl, A tabu search based heuristic method for the transit route network design problem–computer-aided systems in public transport Lecture Notes in Economics and Mathematical Systems, vol. 600, Springer, Berlin, 2008, pp. 387–408.

[19] W. Fan, R.B. Machemehl, Tabu search strategies for the public transportation network optimizations with variable transit demand, Comput.-Aided Civil Infrastruct. Eng. 23 (8) (2008) 502–520.

[20] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: IEEE Congress on Evolutionary Computation (CEC 1999), 1999, pp. 1945–1950.

[21] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization. An overview, Swarm Intell. 1 (1) (2007) 33–57.

[22] S.-C. Chu, Y.-T. Chen, J.-H. Ho, Timetable scheduling using particle swarm optimization, in: 1st International Conference on Innovative Computing, Information and Control, 2006, pp. 324–327.

[23] D. Qarouni-Fard, A. Najafi-Ardabifi, M.-H. Moeinzadeh, S.-R. Sharifian, E. Asgarian, J. Mohammadzadeh, Finding feasible timetables with particle swarm optimization, in: 4th International Conference on Innovations in Information Technology, 2008, pp. 387–391.

[24] I. Sheau Fen Ho, D. Safaai Mohd Hashim, S. Zaiton, A study on PSO-based university course timetabling problem, in: International Conference on Advanced Computer Control, 2008, pp. 648–651.

[25] I. Sheau Fen Ho, D. Safaai Mohd Hashim, S. Zaiton, A combination of PSO and local search in university course timetabling problem, in: International Conference on Computer Engineering and Technology, 2009, pp. 492–495.

[26] I. Sheau Fen Ho, D. Safaai Mohd Hashim, S. Zaiton, University course timetable planning using hybrid particle swarm optimization, in: 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009, pp. 239–245.

[27] Y.-M. Wang, J.I. Jun-Zhong, Particle swarm optimization algorithm for solving examination timetabling problem, J. Comput. Appl. 29 (Z1) (2009) 137–140.

[28] E. Montero, M.-C. Riff, L. Altamirano, A PSO algorithm to solve a Real Course + Exam Timetabling Problem, in: International Conference on Swarm Intelligence, 2011, pp. 24-1–24-8.

[29] M.A.A. Aziz, M.N. Taib, N.M. Hussin, An improved event selection technique in a modified PSO algorithm to solve class scheduling problems, in: IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), 2009, pp. 203–208.

[30] M.A.A. Aziz, M.N. Taib, N.M. Hussin, Assignments acceptance strategy in a modified PSO algorithm to elevate local optima in solving class scheduling problems, in: 6th International Colloquium on Signal Processing and its Applications (CSPA), 2010, pp. 359–363.

[31] M.A.A. Aziz, M.N. Taib, N.M. Hussin, The effects of Event Selection Based on Soft Constraint Violation (ESSCV) in a modified PSO algorithm to solve class scheduling problems, in: International Conference on Computer Applications and Industrial Electronics (ICCAIE 2010), 2010, pp. 584–587.

[32] X. Ioannis, Tassopoulos, N. Grigorios, Beligiannis, Using particle swarm optimization to solve effectively the school timetabling problem, Soft Comput. 16 (7) (2012) 1229–1252.

[33] X. Ioannis, Tassopoulos, N. Grigorios, Beligiannis:, A hybrid particle swarm optimization based algorithm for high school timetabling problems, Appl. Soft Comput. 12 (11) (2012) 3472–3489.

[34] F.A. Kidwai, Optimal Design of Bus Transit Network: A Genetic Algorithm Based Approach, Indian Institute of Technology, Kanpur, India, 1998 (PhD dissertation).

[35] L. Fan, C.L. Mumford, A metaheuristic approach to the urban transit routing problem, J. Heuristics 16 (3) (2010) 353–372.

[36] F. Zhao, A. Gan, Optimization of Transit Network to Minimize Transfers, Final Report, Contact No. BD015-02, Research Office, Florida Department of Transportation, Tallahassee, FL, 2003.

[37] L.E. Fernandez, C.J. Cea, R.H. Malbran, Demand responsive urban public transport system design: methodology and application, Transp. Res. A: Policy Pract. 42 (7) (2005) 951–972.

[38] R. Balcombe, The Demand for Public Transport: A Practical Guide, Transportation Research Laboratory Report (TRL593), Transportation Research Laboratory, London, UK, 2004.

[39] B. Emerson, Design and Planning Guidelines for Public Transport Infrastructure – Bus Route Planning and Transit Streets (8803-500-001), Public Transport Authority, USA, 2003.

[40] C.E. Mandl, Evaluation and optimization of urban public transport networks, in: 3rd Congress on Operations Research, Amsterdam, 1979, pp. 396–404.

[41] Z. Yang, B. Yu, C. Cheng, A parallel ant colony algorithm for bus network optimization, Comput.-Aided Civil Infrastruct. Eng. 22 (1) (2007) 44–55.

[42] Y. Israeli, A. Ceder, Designing transit routes at the network level, in: IEEE Vehicle Navigation and Information Systems Conference, 1989, pp. 310–316.

[43] G.F. Newell, Some issues relating to the optimal design of bus routes, Transp. Sci. 13 (1) (1979) 20–35.

[44] C.E. Mandl, Evaluation and optimization of urban public transportation networks, Eur. J. Oper. Res. 5 (6) (1980) 396–404.

[45] A. Ceder, N.H.M. Wilson, Bus network design, Transp. Res. B: Methodol. 20 (4) (1986) 331–344.

[46] D. Dubois, G. Bel, M. Llibre, Set of methods in transportation network synthesis and analysis, J. Oper. Res. Soc. 30 (9) (1979) 797–808.
[47] W. Lampkin, P.D. Saalmans, Design of routes service frequencies and schedules for a municipal bus undertaking – a case study, Oper. Res. Q. 18 (4) (1967) 375–397.
[48] A. Amberg, W. Domschke, S. Voss, Multiple center capacitated arc routing problems: a tabu search algorithm using capacitated trees, Eur. J. Oper. Res. 124 (2) (2000) 360–376.
[49] S. Fores, L. Proll, A. Wren, A column generation approach to bus driver scheduling, in: M.H.G. Bell (Ed.), Transportation Networks: Recent Methodological Advances, Pergamon, Amsterdam, 1998, pp. 195–208.
[50] L. Fan, C.L. Mumford, D. Evans, A simple multi-objective optimization algorithm for the urban transit routing problem, in: IEEE Congress on Evolutionary Computation, (CEC 2009), 2009, pp. 1–7.
[51] R.W. Floyd, Algorithm 97: Shortest path, Commun. ACM 5 (6) (1962) 345.
[52] R.Z. Farahani, E. Miandoabchi, W.Y. Szeto, H. Rashidi, A review of urban transportation network design problems, Eur. J. Oper. Res. 229 (2) (2013) 281–302.
[53] R.L. Mackett, M. Edwards, The impact of new urban public transport systems: will the expectations be met? Transp. Res. A: Policy Pract. 32 (4) (1998) 231–245.
[54] M.V. Chester, A. Horvath, Environmental assessment of passenger transportation should include infrastructure and supply chains, Environ. Res. Lett. 4 (2) (2009) 024008.
[55] S. Mitric, Urban transport lending by the World Bank: the last decade, Res. Transp. Econ. 40 (1) (2013) 19–33.
[56] APTA Says Transportation Authorization Proposal Underfunds Critical Needs of Nation's Public Transportation Systems, Statement by American Public Transportation Association (APTA) President William Millar, http://www.apta.com/mediacenter/pressreleases/2011/Pages/110707_.aspx (accessed 7.7.11).
[57] D.A. Hensher, The imbalance between car and public transport use in Australia: why does it exist? Trans. Policy 5 (4) (1998) 193–204.
[58] P.N. Balchin, G.H. Bull, J.L. Kieve, Urban Land Economics and Public Policy, 5th ed., Palgrave Macmillan, London, 1995.
[59] M. Criden, The Stranded Poor: Recognizing the Importance of Public Transport for Low-income Households, Issue Brief, National Association for State Community Services Program, 2008.
[60] M. Edwards, R.L. Mackett, Developing new urban public transport systems: an irrational decision-making process, Transp. Policy 3 (4) (1996) 225–239.
[61] M. Chester, S. Pincetl, Z. Elizabeth, W. Eisenstein, J. Matute, Infrastructure and automobile shifts: positioning transit to reduce life-cycle environmental impacts for urban sustainability goals, Environ. Res. Lett. 8 (1) (2013) 015041.
[62] J. Zhang, H. Lu, L. Fan, The multi-objective optimization algorithm to a simple model of urban transit routing problem, in: Sixth International Conference on Natural Computation (ICNC 2010), 2010, pp. 2812–2815.
[63] O.S.C. Chew, L.S. Lee, A genetic algorithm for urban transit routing problem, Int. J. Mod. Phys.: Conf. Ser. 09 (2012) 411–421.