A Multi-Objective Metaheuristic approach for the Transit Network Design Problem

Antonio Mauttone and María E. Urquhart

Operations Research Department, Computer Science Institute,
Engineering School, Universidad de la República
Address: J. Herrera y Reissig 565 Piso 5, Montevideo, Uruguay
Email: {mauttone|urquhart}@fing.edu.uy

March 21, 2007

Keywords: Transit Network Design Problem, multi-objective combinatorial optimization, GRASP

Abstract: We study the problem of the optimal design of routes and frequencies in urban public transit systems, the Transit Network Design Problem (TNDP). We model it as a multi-objective combinatorial optimization problem, which consists in optimizing simultaneously the conflicting objectives of users and operators. A new approximative algorithm based on the GRASP metaheuristic is proposed to solve the TNDP. This algorithm can be classified as a multi-objective metaheuristic since it produces a set of non-dominated solutions in a single run. It differs from most previous approaches, which have used the Weighted Sum Method to generate a set of non-dominated solutions by running a single-objective optimization algorithm for several weights representing different trade-off levels between the conflicting objectives. Numerical results are presented, showing that the multi-objective metaheuristic is more efficient in terms of execution time than the Weighted Sum Method.

Una metaheurística multi-objetivo para el problema de diseño de redes de transporte público

Palabras clave: diseño de redes de transporte público, optimización combinatoria multi-objetivo, GRASP

Resumen: Se estudia el problema del diseño óptimo de recorridos y frecuencias en sistemas de transporte público urbano, el Transit Network Design Problem (TNDP). El mismo se modela como un problema de optimización combinatoria multi-objetivo, consistente en optimizar simultaneamente los objetivos en conflicto de los usuarios y los operadores. Se propone un nuevo algoritmo aproximado basado en la metaheurística GRASP para resolver el TNDP. Este algoritmo puede ser clasificado como una metaheurística multi-objetivo, dado que produce un conjunto de soluciones no dominadas en una sola ejecución. La propuesta difiere de la mayoría de los enfoques anteriores, que han usado el Método de Suma Ponderada para generar un conjunto de soluciones dominadas, ejecutando un algoritmo de optimización de objetivo único para varios pesos que representan diferentes grados de compromiso entre los objetivos en conflicto. Se presentan resultados numéricos, mostrando que la metaheurística multi-objetivo es más eficiente en términos de tiempo de ejecución que el Método de Suma Ponderada.

1 Introduction

The planning of urban public transit systems involves making decisions about many infrastructure and operational aspects. Ceder and Wilson [8] present a decomposition of the entire planning process in a sequence of activities, namely, network design, frequency setting, timetable development, bus and driver scheduling. At the network design level a set of routes is defined, where each route is a sequence of links of the road network. This activity is performed in the context of strategic planning, taking decisions in a long time horizon [12], usually regulated by the authorities. In the most general case, route and frequency definition has to take into account the interests of users (people who employ or need to employ public transport) and operators (private companies who own the resources to give the service). The simultaneous optimization of the system from the viewpoints of both users and operators poses a design trade-off. Roughly speaking, a better service is offered to the users in the presence of more routes and high frequencies. Because invariably there exist upper bounds in the companies resources which make the operation profitable and in the monetary cost feasible for the users, a convenient trade-off level has to be established, maybe entailing the evaluation of various alternative system designs. Thus, the problem of the optimal design of routes and frequencies has an intrinsic multi-objective nature.

The Transit Network Design Problem (TNDP) aims to find a set of routes with their corresponding frequencies for an urban public transit system, optimizing the conflicting objectives of users and operators. Main problem data are the road network and the demand between different points of the city. Constraints are usually related to demand satisfaction, required level of service and resource availability (but there can be other additional constraints). Frequencies are included as decision variables in network optimization models, because they also have a direct influence in the cost structure of both users and operators. The exact resolution of the TNDP has the following difficulties, enumerated among others in [4, 9]:

- High combinatorial complexity: Israeli and Ceder [18] classified the problem as a complex variant of the generalized network design problem [21], which is NP-hard.
- TNDP requires an assignment submodel: the evaluation of a solution from the users viewpoint needs a behavior model of the passengers concerning the routes and frequencies of that solution. The assignment model of passengers to routes (or simply *assignment model*) specifies how the demand is distributed among a given set of routes, and its solution constitutes a hard problem by itself [12], in this case posed as a subproblem of the TNDP.
- Multi-objective nature: the existence of conflicting objectives adds complexity to the problem, either in the *a priori* estimation of the relative importance of the objectives, or in the calculation of several solutions with different trade-off levels between the conflicting objectives.

The combinatorial complexity of the TNDP has been tackled in the existing literature exclusively by means of approximative methods. Complete enumeration of feasible solutions is prohibitively expensive; mathematical programming formulations exists only for simplified versions of the problem [7, 30]. The first algorithms published for the TNDP were heuristics [2, 5, 22, 27]. Lately, several applications of metaheuristics have been proposed, most of them using Genetic Algorithms with different coding schemes [24, 25, 29], Tabu Search [15] and GRASP [23].

The assignment models generally used in the context of the TNDP aim to give a realistic representation of the interaction between passengers and buses; but their complexity must be kept bounded given the impact they have in the overall efficiency of the optimization algorithms. Most common approaches are all-or-nothing assignment [24], common lines and transfers [3], and detailed network treatment [15].

Most of previous works have considered the multi-objective nature of the TNDP by using an *a priori* estimation of a vector of weights to express a particular trade-off level between the conflicting objectives [4, 15, 24, 25, 29]. Only the work of Israeli and Ceder [18] implements a heuristic to explicitly obtain a set of solutions with different trade-off levels. All the existing metaheuristic based algorithms for the TNDP (with the sole exception of [23]), solve a single-objective optimization problem by summarizing the different objectives in one by using a vector of weights.

In the last years, a growing amount of work has been published about metaheuristics specially designed for multi-objective combinatorial optimization problems. This type of algorithms has been denominated as *multi-objective metaheuristics* and they are defined by different authors as: methods that aim at generating a good set of non-dominated solutions in a single run [19], and algorithms that deal with the multiple objectives directly [14]. The basic idea of multi-objective metaheuristics is to adapt the mechanisms of their original single-objective counterparts to handle effectively and efficiently multi-objective optimization problems [10, 13, 14].

In this work we present an approximative algorithm based in the GRASP metaheuristic [26] to solve the TNDP with a multi-objective approach. It allows to obtain in a single run, a set of non-dominated solutions representing different trade-off levels between the conflicting objectives. According to this, it can be classified as a multi-objective metaheuristic based algorithm, since it is specially developed to take into account the multi-objective nature of the problem. It includes a new route set construction procedure, and a neighborhood definition that is used to search for a near optimal set of frequencies for a particular trade-off level. The assignment model of Baaj and Mahmassani [3] is used to calculate some variables needed by the optimization procedure. The proposed methodology is tested with the case of Mandl [4]. Existing and proposed measures are calculated for different non-dominated solutions; these measures can be useful for the decision maker. Moreover, we show evidence that the multi-objective metaheuristic algorithm is more efficient than a single-objective version of it, which is used as a subroutine in the Weighted Sum Method [11].

The paper is organized as follows. We formally state the problem and used notation in Sect. 2 and present the adopted multi-objective approach in Sect. 3. Details of the GRASP implementation are given in Sect. 4. Numerical results are presented in Sect. 5, and finally some conclusions and future work are formulated in Sect. 6.

2 Problem definition and notation employed

We model the network that enables the definition of routes as an undirected graph $G = (N, E)$, where N is the set of vertices ($|N| = n$) and E is the set of edges representing connections between vertices. The cost c_e of an edge $e = (i, j) \in E$ models the in-vehicle travel time, i.e. the time spent by vehicles to travel between vertices i and j; this cost is the same for every route that uses edge e, and it is independent of the occupation level of the vehicles. An origin-destination matrix $D = \{d_{ij}, i, j \in [1..n]\}$ is given, which characterizes the demand; d_{ij} denotes the demand from vertex i to vertex j, expressed in trips per time unit in a given time horizon (to be accomplished by one passenger that will occupy one place in a vehicle). A route is a sequence of adjacent vertices in G. Since it is composed by undirected edges, it has forward and backward directions. A solution S for the TNDP is a pair (R, F) where $R = \{r_1, \ldots, r_r\}$ is the set of routes and $F = \{f_1, \ldots, f_r\}$ is the set of frequencies; each f_k is a real value that represents the inverse of the time between subsequent vehicles on route r_k. We denominate line k to the pair (r_k, f_k). Given a solution S, the assignment model produces a set of corresponding flows $\Phi(S) = \{\Phi_1(S), \ldots, \Phi_r(S)\}$, which express the distribution of the demand D among the lines of S. We define $\Phi_k(S) = \{\overrightarrow{\phi}_{e,k}, \overleftarrow{\phi}_{e,k}, e \in r_k\}$ as the set of flows in route r_k, where $\overrightarrow{\phi}_{e,k}$ and $\overleftarrow{\phi}_{e,k}$ are forward and backward flow respectively, in edge e in route r_k. Flows are expressed in the same units as the origin-destination matrix D.

The conflicting objectives of users and operators are modelled with functions Z_1 and Z_2 respectively, which have to be minimized simultaneously. The former is defined as

$$Z_1(S) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}(tv_{ij} + tw_{ij} + tt_{ij}) \ . \tag{1}$$

Using this function, we minimize the overall time needed to transport the users between their corresponding origin and destination vertices. It has three components: in-vehicle travel time tv, waiting time tw and transfer time tt. These values are determined by the assignment model; tv_{ij} is calculated using the costs of the edges of G that are used by lines in S connecting vertices i and j; tw_{ij} depends of the frequencies of these same lines; tt_{ij} is a penalty (expressed in time units) which represents the discomfort of transfers from the users viewpoint (we define σ_t as the penalty of each demand unit which has to perform transfers).

Operator costs are more difficult to express in a general sense, as they depend of each particular case. For this reason we use the number of simultaneously operating vehicles (fleet size) needed to operate the routes with given frequencies in a solution S, as a proxy for operator costs. It is expressed as

$$Z_2(S) = \sum_{r_k \in R} f_k t_k \ , \tag{2}$$

where $t_k = 2 \sum_{e \in r_k} c_e$ is the total duration (round-trip time) of route r_k.

For a given solution $S = (R, F)$, we define $D_0(S) \in [0, 1]$ as the proportion of the total demand $D_{tot} = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}$ satisfied for routes R directly (without transfers), independently of frequencies F. Analogous, $D_{01}(S)$ is the proportion of D_{tot} satisfied for routes R directly or indirectly (one transfer, at most), independently of F. D_0^{min} and D_{01}^{min} are constant values, which constrain $D_0(S)$ and $D_{01}(S)$ respectively as

$$D_0(S) \geq D_0^{min} \ , \tag{3}$$

$$D_{01}(S) \geq D_{01}^{min} \ . \tag{4}$$

We consider lower and upper values for frequencies, f_{min} and f_{max} respectively. While the former takes care of the level of service offered to the users, the latter represents a limit imposed by the operational possibilities of the transit mode. These constraints are expressed as

$$f_{min} \leq f_k \leq f_{max} \qquad \forall f_k \in F \ . \tag{5}$$

The maximum load factor constraint imposes an additional condition for the frequencies. It is expressed as

$$f_k \geq \frac{\phi_k^*}{\omega_{max} Q} \qquad \forall f_k \in F \ , \tag{6}$$

where $\phi_k^* = \max \ \Phi_k(S)$ is the critical flow in route r_k and Q is the seating capacity of vehicles. The given constant $\omega_{max} \geq 1$ is the maximum load factor in vehicles, expressing a tolerance in the number of standing passengers; according to this, $\omega_{max} Q$ is the maximum allowed capacity of vehicles.

We denominate as $\mathcal{P}$ the problem defined by the simultaneous optimization of objective functions (1) and (2), under constraints (3), (4), (5) and (6).

3 Multi-objective approach

Problem $\mathcal{P}$ can be classified according to [13] as a multi-objective combinatorial optimization (MOCO) problem. The discrete nature of the variables that represent the structure of routes gives the combinatorial characteristic. The conflicting objectives represented by functions (1) and (2) result in the existence of a set of *Pareto optimal solutions* instead of a single optimal solution.

The multi-objective nature of the TNDP as it is posed by objective functions (1) and (2) (or by similar formulations), has been tackled by the following approaches in the existing literature:

- Making an *a priori* estimation of the relative importance of the conflicting objectives in the form of a vector of weights, and then solving a single-objective optimization problem [24, 25, 29]. Some authors suggest that by varying the values of these weights, a set of non-dominated solutions can be obtained [4]. We denominate such approach as *weighted sum approach.*
- Calculating a set of non-dominated solutions and then selecting a single one [18], that we denominate as *multi-objective approach.*

In this work we adopt the latter approach, and we concentrate in finding a good set of non-dominated solutions which can be used in subsequent steps by the decision maker, either to select a single non-dominated solution, to compare the solutions with an existing solution or to evaluate alternative solutions.

3.1 Definitions and terminology

Given the multi-objective nature of problem $\mathcal{P}$, it does not have a single optimal solution S^*; instead it has a set of *Pareto optimal solutions* P^*, called *optimal Pareto front* [11]. The optimal Pareto front of a multi-objective optimization problem with feasible set of solutions C is the non-dominated set of C, i.e. the subset of solutions of C that are not dominated by other solutions of C. A solution S_1 dominates another solution S_2 if S_1 is no worse than S_2 in all objectives and S_1 is strictly better than S_2 in at least one objective. Moreover, two solutions S_3 and S_4 in C are not dominated if both S_3 does not dominate S_4 and S_4 does not dominate S_3.

When we refer to elements in the feasible set C, we are dealing with the so called *decision space*, i.e. the space where variables take values. On the other hand, domination is defined according to the values of the objective functions evaluated over solutions of C in the so called *objective space*. Figure 1 exemplifies these concepts for a bi-objective minimization problem; that is the case of the TNDP.

3.2 Multi-objective metaheuristics

To solve exactly problem $\mathcal{P}$ according to a multi-objective approach requires finding all of its Pareto optimal solutions. Most MOCO problems are proven to be NP-hard as well as #P-hard, and this is true even for problems which have efficient algorithms in the single-objective case [14]. This implies that there is no chance to have a polynomial time algorithm to exactly solve them,

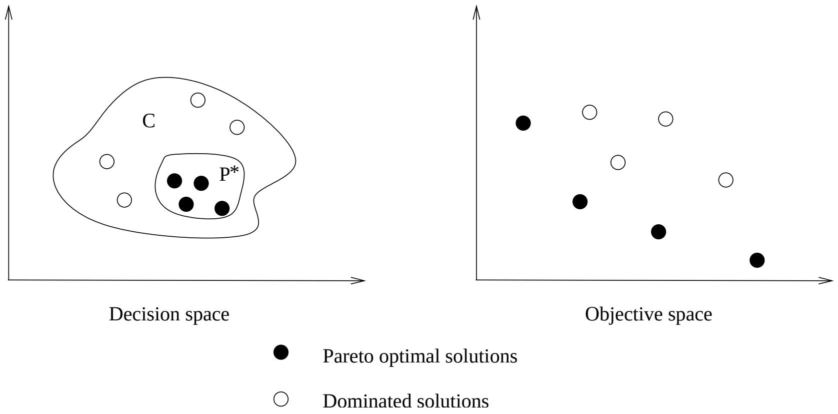


Fig. 1. Decision and objective spaces for a bi-objective minimization problem

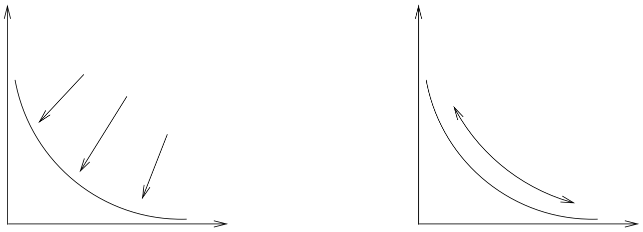
and even to count the elements of the optimal Pareto front. For these reasons we consider the use of an approximative algorithm to obtain an *approximated Pareto front* for $\mathcal{P}$.

Methods for multi-objective optimization that use algorithms for single-objective optimization as subroutines are considered in [11] as part of the *classical methods*. Most popular classical methods used for MOCO problems are the *Weighted Sum Method* and the *ϵ-Constraint Method* [14]. They can find Pareto optimal solutions in some cases, but they have the following disadvantages [11]:

- Only one Pareto optimal solution can be expected to be found in a single run of the single-objective optimization algorithm.
- Not all Pareto optimal solutions can be found by some algorithms in non-convex multi-objective optimization problems. This is true for MOCO problems given the discrete nature of its variables (see unsupported efficient solutions [14]).
- They requiere some problem knowledge, such as suitable weights or values to constraint objective functions.

Multi-objective metaheuristics have been proposed as candidates to overcome the difficulties of classical methods in solving MOCO problems. Main algorithmic difference with their single-objective counterparts, is the adopted search mechanism to obtain a good approximated Pareto front. In multi-objective optimization, the goals of an approximative algorithm are two, namely [11]:

- Closeness: one seeks to find solutions which are close to the optimal Pareto front (Fig. 2(a)).
- Diversity: one also wants to obtain a set of non-dominated solutions which covers different areas of the objective space, representing a diverse set of trade-off levels between the conflicting objectives (Fig. 2(b)).



(a) Closeness (b) Diversity

Fig. 2. Two goals of an approximative algorithm for a bi-objective minimization problem

On seeking closeness and diversity, multi-objective metaheuristics have to implement specially designed search mechanisms, which have to deal with both decision and objective spaces.

The field of multi-objective metaheuristics has been growing in the last years. Main metaheuristics adapted for multi-objective optimization are Evolutionary Algorithms, Simulated Annealing, Tabu Search and Ant Colony Optimization. Some work has also been published about adaptation of Artificial Neural Networks, GRASP and Scatter Search for MOCO problems [14]. Surveys about multi-objective metaheuristics can be found in [10] and [14].

3.3 Multi-objective GRASP for the TNDP

GRASP (Greedy Randomized Adaptive Search Procedures [26]) is a metaheuristic for combinatorial optimization problems. It consist in the repeated execution of a solution construction procedure followed by a local search. The construction is performed using a greedy criterion, by adding iteratively to a solution, elements which are randomly selected from a candidate list. The local search requires the definition of a neighborhood structure, through which to successively advance in the direction of improvement of the objective function. The sequence of construction and local search (GRASP iteration) is repeated a given number of times, obtaining different trajectories in the feasible space. Finally, the best found solution is returned.

In this work we adapt GRASP to solve the TNDP with a multi-objective approach. Existing adaptations of GRASP for MOCO problems can be found in [6, 23, 28]. The general structure of the proposed algorithm, that we call as GRASP TNDP, is the following (implementation details are given in Sect. 4):

- The construction procedure generates a set of routes R, which fulfills constraints (3) and (4) of the optimization problem $\mathcal{P}$. Routes are constructed by using shortest paths between vertices in G and then inserting additional pairs of vertices on them.
- The local search calculates a near optimal set of frequencies F, according to constraints (5) and (6), for a given trade-off level between the conflicting

objectives represented by functions (1) and (2). This procedure takes a random vector of weights and uses a neighborhood structure to advance in the direction of improvement of a single composite objective function. The neighborhood of a solution is defined by varying its frequencies in a predetermined set.

In this way, at each GRASP iteration, different points in both decision and objective spaces are sampled. Different trade-off levels are obtained by varying from one GRASP iteration to other, parameters of maximum route duration t_{max} at the construction procedure and random vector of weights λ at the local search. All solutions of the trajectory of the local search are added to the set of potentially non-dominated solutions under construction P. At the end of each GRASP iteration, all dominated solutions in P are deleted. The assignment model of Baaj and Mahmassani [3] is used to evaluate objective function $Z_1(S)$ for each solution S and to verify frequency feasibility according to constraint (6).

4 The algorithm

When we instantiate the GRASP metaheuristic for a particular application, we have to tailor all its problem dependent aspects. The constructive algorithm has to be specified, which entails to specify how to build the list of candidate elements to be added to the solution under construction, how these elements are ranked at each step of the construction according a required adaptive greedy function, how to construct the restricted candidate list, and how the elements are selected from that list. For the local search, a neighborhood structure and its exploration strategy have to be defined. Also a stopping rule is required.

4.1 Constructive algorithm

The constructive algorithm (Fig. 3) starts with an empty set of routes R, and iteratively seeks to satisfy the demand specified by origin-destination matrix D. At each iteration step, a restricted candidate list rcl is constructed by selecting the $\alpha|l|$ pairs of vertices (i, j) with highest demand d_{ij} in l, where $\alpha \in [0..1]$ is a parameter of GRASP and l is a list made of all pairs of vertices whose demand is not yet satisfied (directly) by routes in R. The pair of vertices (u, v) is randomly selected from rcl by using a distribution with probabilities which are proportional to the demand. This approach is described in [26] as a method for setting probabilities to the elements of rcl by introducing bias functions. In this case we define $bias(u, v) = d_{uv}$, and the corresponding probability is

$$\mathrm{Prob}(u, v) = \frac{bias(u, v)}{\sum_{(i,j)\in rcl} bias(i, j)} \ . \tag{7}$$

The corresponding demand d_{uv} is satisfied according to one of the two following possibilities:

1. Creating a new route, using the shortest path between u and v in G, measured in in-vehicle travel time.
2. Inserting vertices u and v in convenient positions of a convenient route of R. We evaluate the cost of insertion of both u and v between all pair of consecutive vertices in routes of R. When we insert a vertex v' in a route between two (consecutive) vertices i and j, we connect it with them using the shortest paths in G between v' and i, and v' and j respectively, ignoring resulting routes with loops. In a route formed by p vertices, there are $(p+1)(p+2)$ insertion possibilities if neither u nor v belongs to the route and $(p+1)$ possibilities if either u or v belongs to the route. The most convenient route and the most convenient positions for insertion of vertices u and v on it, are those which minimize the cost increase in the solution under construction, measured in terms of in-vehicle travel time.

The lowest cost increase due to insertion of vertices u and v in a route of R according to case 2 is compared with the cost of the shortest path between u and v according to case 1; the best (less costly) case is selected and the algorithm proceeds.

Two constraints are imposed in routes under construction: maximum duration t_{max} and maximum circuity factor ρ_{max}. The circuity factor of a route r with extreme vertices u and v, is defined in [5] as the ratio between the in-vehicle travel time between u and v using r, and the cost of the shortest path between u and v in G (independently of any route).

The construction algorithm ends when constraints (3) and (4) are satisfied. The general structure of the proposed constructive algorithm is inspired in the Route Generation Algorithm (RGA) of Baaj and Mahmassani [5]. The main difference is that where RGA inserts individual vertices on existing routes of the solution under construction, our algorithm inserts pairs of vertices, trying to satisfy directly the demand associated to them.

4.2 Local search

The local search operates with the set of frequencies $F = \{f_1, \ldots, f_r\} \in \Theta^r$ of a solution $S = (R, F)$; this means than only frequencies are decision variables in this phase (Fig. 4). The neighborhood N_S of S is defined as:

$$N_S = \{S' = (R, F') \text{ with } F' = \{f'_1, \ldots, f'_r\} \in \Theta^r \;/ \exists j \in [1..r] \text{ which fulfills } f_j \text{ and } f'_j \text{ are consecutive in } \Theta^r \text{ and } f_i = f'_i \; \forall i \in [1..r], i \neq j\},$$

where $\Theta = \{\theta_1, \ldots, \theta_f\} \in \mathbb{R}^f$ is a set of values of frequencies sorted in increasing order, which fulfills $\theta_1 \geq f_{min}$ and $\theta_f \leq f_{max}$.

According to this definition of neighborhood, the local search algorithm evaluates the costs of increasing or decreasing the frequencies in all routes of

```
procedure Construction(in $D_0^{min}$, $D_{01}^{min}$, in $\rho_{max}$, $t_{max}$, in $\alpha$, out $R$);
  $R \leftarrow \emptyset$;
  $D_0(S) \leftarrow 0$;
  $D_{01}(S) \leftarrow 0$;
  $l \leftarrow$ List of pairs of vertices $(i, j)$ of $G$ with $d_{ij} \neq 0$;
  while $D_0(S) < D_0^{min}$ or $D_{01}(S) < D_{01}^{min}$ do
    $rcl \leftarrow$ Construct according $\alpha$ and $l$;
    $(u, v) \leftarrow$ Select randomly from $rcl$ according (7);
    $r \leftarrow$ Create a route with the shortest path between $u$ and $v$ in $G$;
    $r' \leftarrow$ Create a route by inserting $u$ and $v$ in the most convenient
            positions in the more convenient route $r''$ in $R$;
    if $cost(r) < cost(r') - cost(r'')$ then
      $R \leftarrow R \cup \{r\}$;
      Delete from $l$ pairs of vertices whose demand is satisfied by $r$;
    else
      $R \leftarrow R \cup \{r'\} - \{r''\}$;
      Delete from $l$ pairs of vertices whose demand is satisfied by $r'$;
    end if;
    Update $D_0(S)$ and $D_{01}(S)$;
  end while;
  return $R$;
end Construction;
```

Fig. 3. Constructive algorithm

```
procedure LocalSearch(in $\lambda$, in $S$, in out $P$);
  $current \leftarrow S$;
  $P \leftarrow P \cup \{current\}$;
  $stop \leftarrow false$;
  repeat
    $S' \leftarrow$ FirstImprovement($current$, $\lambda$);
    if $S'$ better than $current$ then
      $current \leftarrow S'$;
      $P \leftarrow P \cup \{current\}$;
    else
      $stop \leftarrow true$;
    end if;
  until $stop$;
  return $P$;
end LocalSearch;
```

Fig. 4. Local search

solution S. At each step of the local search, the cardinality of N_S can be at most $2|R|$; however this number can be smaller when there is a constraint of minimum feasible frequencies in S (determined by the assignment model) or when there are routes with frequencies in an extreme of the set Θ. The local search receives a random vector of weights $\lambda = (\lambda_1, \lambda_2)$ and successively moves forward to the neighbor which minimizes the composite objective function $\lambda_1 Z_1 + \lambda_2 Z_2$, using a first improving strategy [26]. The evaluation of each neighbor solution implies an invocation to the algorithm that implements the assignment model.

4.3 GRASP TNDP

Figure 5 presents a pseudo code of the GRASP TNDP algorithm. It begins by calculating the shortest path between all pairs of vertices in G; this can be made only once, independently of the GRASP iterations, because the cost of the edges of G are considered as constant, not depending on the flows produced by different solutions.

```
procedure GRASP_TNDP(in $D_0^{min}$, $D_{01}^{min}$, in $\rho_{max}$, $t_{max}^{ini}$, $t_{max}^{end}$,
                     in $NumIterations$, $\alpha$, out $P$);
  Calculate shortest paths between all pairs of vertices in $G$;
  $P \leftarrow \emptyset$;
  for $i = 1$ to $NumIterations$ do
    $t_{max} \leftarrow$ Random uniform value in $[t_{max}^{ini}, t_{max}^{end}]$;
    Construction($D_0^{min}, D_{01}^{min}, \rho_{max}, t_{max}, \alpha, R$);
    $F \leftarrow$ Initial frequencies;
    $S \leftarrow (R, F)$;
    $\lambda \leftarrow$ Random vector of weights;
    LocalSearch($\lambda, S, P$);
    Delete dominated solutions of $P$;
  end for;
  return $P$;
end GRASP_TNDP;
```

Fig. 5. GRASP TNDP algorithm

The maximum duration of routes t_{max} is determined at each GRASP iteration by sampling a random uniform value in the real interval $[t_{max}^{ini}, t_{max}^{end}]$ (whose extremes are parameters to be given). This feature is included to obtain diverse solutions, each having internally homogeneous characteristics (all routes with approximately the same duration).

The initial frequencies for each route in R are determined as the maximum of f_{min} and the minimum feasible value of Θ according to constraint (6), calculated by the assignment model.

The random vector of weights $\lambda = (\lambda_1, \lambda_2)$ is determined by sampling a random uniform value in the real interval $[0, 1]$ for λ_1 and then setting $\lambda_2 = 1 - \lambda_1$. The values of Z_1 and Z_2 are normalized.

5 Numerical results

We test the GRASP TNDP algorithm with numerical data. The performed studies are designed to:

1. Investigate whether the proposed algorithm produces a set of diverse non-dominated solutions for the TNDP. Some descriptive measures are calculated, which can be useful for the decision maker.
2. Compare the relative efficiency of the multi-objective algorithm with respect to a single-objective version of it, used as subroutine in the Weighted Sum Method.

Parameter	Value	Units
D_0^{min}	0.9	-
D_{01}^{min}	1.0	-
ρ_{max}	1.5	-
t_{max}^{ini}	40	minutes
t_{max}^{end}	120	minutes
σ_t	5	minutes
Q	40	seats
ω_{max}	1.25	-
f_{min}	1/60	vehicles/minute
f_{max}	1/2	vehicles/minute
Θ	{1/60, 1/50, 1/40, 1/30, 1/20, 1/10, 1/5, 1/2}	vehicles/minute
α	0.2	-

Table 1. Parameter configuration

We use the test case of Mandl, which is taken from [4]. The network has 15 vertices and 21 edges and the origin-destination matrix has 76% of elements different from zero. The algorithm was coded in C++ and all tests were ran in a PC Pentium 4, with a 2.6 GHz processor and 512 MB of RAM memory. The used parameter configuration for both model and algorithm is shown in Table 1.

5.1 Results of GRASP TNDP

The execution of the GRASP TNDP algorithm with parameter configuration given in Table 1 and 1000 GRASP iterations took 392 seconds and produced an approximated Pareto front P composed by 106 non-dominated solutions. Table 2 shows for 10 solutions (selected as representative points from different regions of P), the corresponding values of Z_1 (along with its components tv, tw and tt), Z_2, number of routes $|R|$ and averaged values over each route of headway $1/f$ and duration t.

In Table 2 we can observe that the trade-off level between the conflicting objectives can be characterized by the values of the number of routes and the average of the route headway. In this way, although there is no monotonic tendency, we can say that solutions with low cost for the users (and therefore with high cost for the operators) are characterized by high values of $|R|$ and low values of $1/f$, and vice versa. Moreover, we can observe that variation along the Pareto front is higher in the waiting time component than in the in-vehicle travel time component of Z_1.
We also present several measures that could be useful for the decision maker. Two of them are relative to distances to lower bounds from the users viewpoint (equations (8) and (9)) and the other two are relative to the utilization of buses. All these measures can be used as a guide in the selection of one non-dominated solution or in the evaluation of an existing one.

Solution	Z_1	tv	tw	tt	Z_2	$\|R\|$	$1/f$	t
1	125.20	112.41	8.38	4.41	155.40	11	3	32
2	134.42	116.96	14.06	3.40	77.40	5	4	56
3	141.69	117.74	20.06	3.89	50.40	5	5	50
4	150.12	118.40	28.28	3.44	35.30	5	10	49
5	160.83	118.10	40.89	1.84	26.77	6	15	53
6	173.00	118.42	52.08	2.50	19.40	4	15	66
7	184.82	122.52	58.28	4.03	15.73	3	17	67
8	205.62	127.74	74.52	3.37	12.10	3	23	67
9	255.38	121.22	129.90	4.27	7.42	4	35	57
10	308.73	126.61	178.65	3.47	5.00	4	43	52

Table 2. Results of GRASP TNDP

From [4] we take the idea of an optimal route set from the users viewpoint, which allows every pair of vertices (i, j) to transport its demand d_{ij} along the shortest path in G, with cost t^*_{ij}. According to this, a lower bound for in-vehicle travel time is defined as

$$tv^* = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} t^*_{ij} \ . \qquad (8)$$

We propose an analog definition of lower bound for the waiting time. An optimal frequency set from the users viewpoint can be defined when every pair of vertices is served by a route with frequency equals to f_{max}. This is in reality a pseudo lower bound, since there can exist solutions where some pairs of vertices are served by more that one route with the maximum frequency.

$$tw^* = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{d_{ij}}{2f_{max}} \tag{9}$$

Table 3 shows for the same solutions of Table 2, values of distances from tv and tw to tv^* and tw^* respectively, where the distance from a value v to its lower bound v^* is defined as $dist(v, v^*) = (v - v^*)/v^*$. Table 3 also shows measures relative to the utilization of buses, averaged over each route, namely:

- Mean utilization $\bar{\Delta}$, defined for each route r_k as $\bar{\phi}_k/(f_k Q)$, where
$$\bar{\phi}_k = \frac{\sum_{e\in r_k}(\overrightarrow{\phi}_{e,k} + \overleftarrow{\phi}_{e,k})c_e}{2\sum_{e\in r_k} c_e} .$$
- Critical utilization Δ^*, defined for each route r_k as $\phi^*_k/(f_k Q)$.

In Table 3 we can observe that the trade-off level between the conflicting objectives also can be characterized by values of distances to lower bounds and utilization of buses. In this way, solutions with low cost for the users are characterized by low values of all these four measures, and vice versa.

Solution	$dist(tv, tv^*)$	$dist(tw, tw^*)$	$\bar{\Delta}$	Δ^*
1	0.04	-0.69	0.02	0.02
2	0.08	-0.48	0.04	0.05
3	0.09	-0.26	0.05	0.07
4	0.09	0.05	0.07	0.09
5	0.09	0.51	0.10	0.14
6	0.09	0.93	0.14	0.20
7	0.13	1.16	0.20	0.29
8	0.18	1.76	0.28	0.39
9	0.12	3.81	0.41	0.53
10	0.17	5.61	0.59	0.78

Table 3. Results of GRASP TNDP, additional measures

Distance to lower bound in the in-vehicle travel time component is no greater than 0.17. Nevertheless, distance to lower bound in the waiting time component is up to 5.61. We can observe that solutions with low cost for the users present negative values in $dist(tw, tw^*)$, since tw^* is in reality a pseudo lower bound, and solutions 1, 2 and 3 have a high average of frequencies (see column $1/f$ in Table 2). It is worth to mention that this is rather a theoretical result, since from a practical viewpoint it may not be possible to operate several lines with high frequencies on a same edge of the network.

Moreover, we can observe that values of both mean and critical utilization of buses are lower than 1.00 for all solutions. Maximum critical utilization is shown by solution 10; this value can not be greater than the level specified by the maximum load factor parameter ω_{max} (configured as 1.25 for this work).

5.2 Comparison with the Weighted Sum Method

We compare the efficiency of GRASP TNDP with respect to the Weighted Sum Method [11]. When it is applied to problem $\mathcal{P}$ (Sect. 2), the Weighted Sum Method consist in minimizing the objective function (10) under constraints (3)-(6), for a given set of different vectors of weights $\lambda = (\lambda_1, \lambda_2)$.

$$Z(S) = \lambda_1 Z_1(S) + \lambda_2 Z_2(S) \tag{10}$$

Now we call as MO GRASP TNDP to the multi-objective algorithm presented in Sect. 4, and SO GRASP TNDP to its single-objective variant, which is used as subroutine in the classical Weighted Sum Method. SO GRASP TNDP differs from MO GRASP TNDP in the following aspects:

- It produces a single solution at every run.
- It receives a vector of weights $\lambda = (\lambda_1, \lambda_2)$, where $\lambda_1 + \lambda_2 = 1$, representing the relative importance between the conflicting objectives Z_1 and Z_2, which is used for:
  - Setting $t_{max} = t_{max}^{ini} + \lambda_2(t_{max}^{end} - t_{max}^{ini})$ in the construction procedure, for all GRASP iterations.
  - Composing a single objective function $Z = \lambda_1 Z_1 + \lambda_2 Z_2$ in the local search.

We perform the execution of SO GRASP TNDP with 11 different vectors of weights of the form $\lambda^i = (0.1i, 1 - 0.1i)$, with $i \in [0..10]$. We use the same parameter configuration and number of GRASP iterations for each execution of SO GRASP TNDP, as for the corresponding single execution of MO GRASP TNDP.

We define the relative efficiency of MO GRASP TNDP with respect to the Weighted Sum Method as

$$EF = \frac{Q_{MG}/T_{MG}}{Q_{WS}/T_{WS}} ,$$

where Q_{MG} and Q_{WS} are quality measures of the Pareto fronts produced by MO GRASP TNDP and the Weighted Sum Method respectively, and T_{MG}, T_{WS} are their respective execution times. The used quality measure is the *hypervolume*, defined in [11] as the volume in the objective space covered (dominated) by solutions in a Pareto front. This measure is intended to reflect the closeness and diversity of a Pareto front for a multi-objective optimization problem. Figure 6 exemplifies the hypervolume for a bi-objective optimization problem; it is represented by the hatched region, where S^* is a fictitious solution, being its corresponding values in the objective space the highest obtained values in all objectives.

Table 4 shows measures relative to quality Q and cardinality of the obtained Pareto fronts $|P|$, execution time T in seconds of both algorithms (MO GRASP TNDP MG and Weighted Sum Method WS), and efficiency EF.

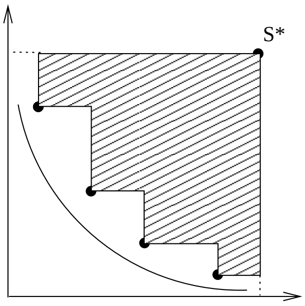


Fig. 6. Hypervolume for a bi-objective minimization problem

The comparison is made for number of GRASP iterations *NumIterations* equal to 10, 100, 500 and 1000. We can observe that the efficiency is always approximately equal to the number of different vectors of weights used in the Weighted Sum Method. Furthermore, the efficiency seems to grow according to the number of GRASP iterations; Figures 7(a) and 7(b) shows graphically the Pareto fronts produced by MO GRASP TNDP and the Weighted Sum Method, for 10 and 1000 GRASP iterations respectively.

6 Conclusions and future work

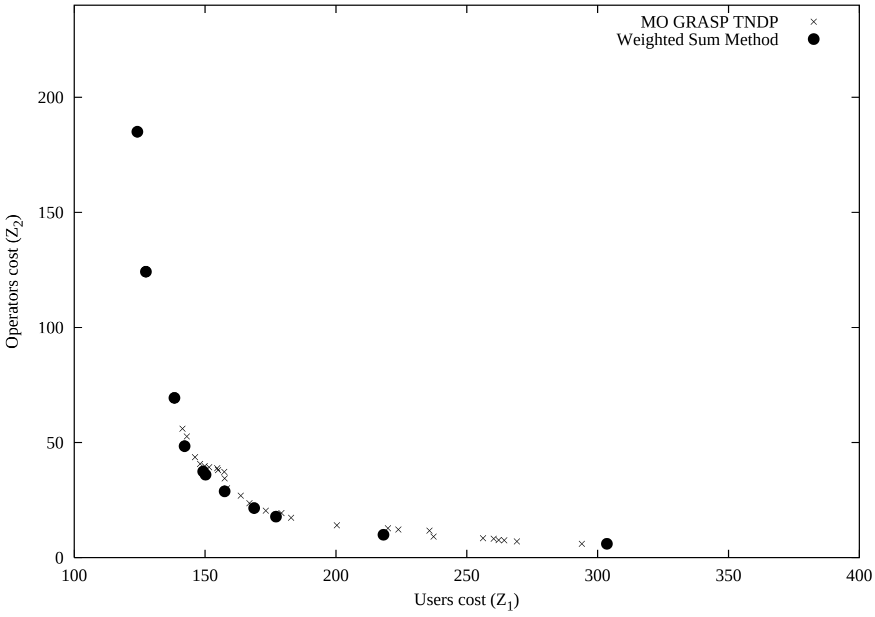
We show that the proposed multi-objective algorithm is capable to produce a diverse set of non-dominated solutions in a single run. Existing and proposed measures are presented, which can be useful for the decision maker, to characterize a given solution of the Pareto front, with respect to the interests of both users and operators.

We also observe that the multi-objective approach is more efficient than the Weighted Sum Method, since both algorithms produce results of similar quality (or slightly better for the multi-objective approach, which also obtains a larger number of points in the Pareto front), while the execution time is lower for the former (actually inversely proportional to the number of different vectors of weights used in the latter).

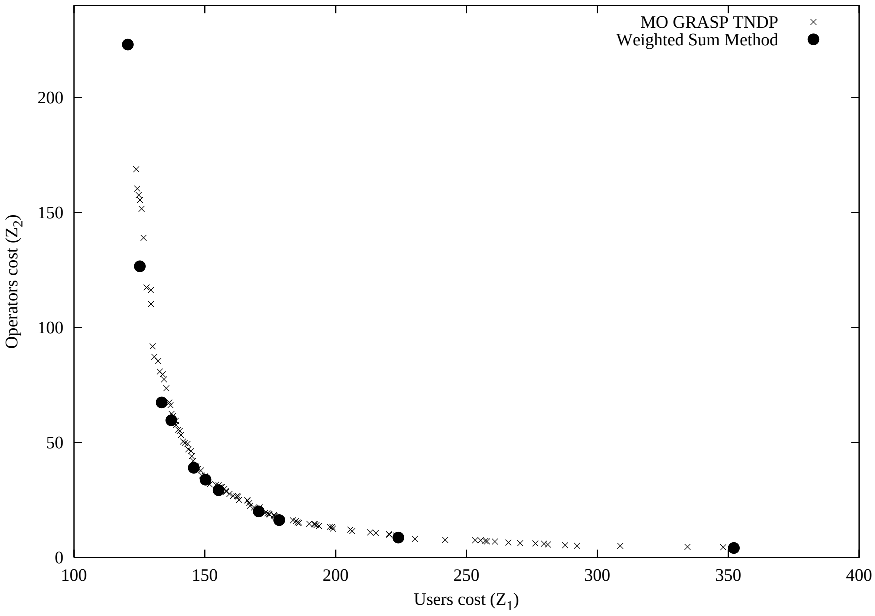
For future research, we identify several directions. One of them is the need of designing mechanisms to improve the diversity of the produced Pareto front for the MO GRASP TNDP algorithm, as it can be seen in Fig. 7(b); this could be obtained for example by applying a reactive GRASP strategy [26].

Parallel implementations have been shown to improve the efficiency of algorithms which are serial originally; this is the case of many GRASP based algorithms [26] as well as of one based in Genetic Algorithms for the TNDP [1].

Most of metaheuristic algorithms for the TNDP are implementations of Genetic Algorithms [24, 25, 29] for single-objective optimization. These ideas can be used to design multi-objective versions, since there are several applications of Genetic Algorithms to multi-objective optimization [10, 11].



(a) 10 GRASP iterations



(b) 1000 GRASP iterations

Fig. 7. Obtained Pareto fronts according to the number of GRASP iterations

$NumIterations$	10	100	500	1000
Q_{MG}	0.873	0.916	0.922	0.925
$\|P_{MG}\|$	27	76	94	106
T_{MG}	4	41	201	392
Q_{WS}	0.902	0.911	0.907	0.910
$\|P_{WS}\|$	11	11	11	11
T_{WS}	47	469	2310	4577
EF	11.37	11.50	11.69	11.87

Table 4. Relative efficiency of GRASP TNDP with respect to the Weighted Sum Method

The presented numerical results lack of an evaluation of the closeness to the optimal Pareto front (which was not available). A possible way to accomplish that evaluation consist in implementing a modified version of the GRASP based algorithm to solve a different optimization model (for example [7], for which a mathematical programming formulation is available).

The hypothesis of inelastic demand is used to simplify the model at this stage of the work. However, for certain cases, elastic demand must be considered in order to model the changes in the origin-destination matrix according to the supply of public transport. Elastic demand has been incorporated to the TNDP in [16, 17, 20] and a challenging work consist in trying to incorporate in a consistent way, this characteristic to the presented multi-objective approach.

Acknowledgements. To the Comisión Sectorial de Investigación Científica of the Universidad de la República and the Programa de Desarrollo Tecnológico for the financial support, to Héctor Cancela for his contributions, and to the Programa de Desarrollo de las Ciencias Básicas. This work is partially supported by Programme ALFA II-0457-FA-FCD-FI-FC.

References

1. Agrawal J, Tom M (2004) Transit route network design using parallel Genetic Algorithm. Journal of Computing in Civil Engineering 18(3):248-256
2. Axhausen KW, Smith R (1984) Evaluation of heuristic transit network optimization algorithms. Transportation Research Record 976:7–20
3. Baaj MH, Mahmassani HS (1990) TRUST: A LISP program for the analysis of transit route configurations. Transportation Research Record 1283:125-135
4. Baaj MH, Mahmassani HS (1991) An AI-based approach for transit route system planning and design. Journal of Advanced Transportation 25(2):187-210
5. Baaj MH, Mahmassani HS (1995) Hybrid route generation heuristic algorithm for the design of transit networks. Transportation Research C 3(1):31-50
6. Baldoquín G (2002) Approximate solution of an extended 0/1 knapsack problem using GRASP. In: XI Congreso Latino-Iberoamericano de Investigación de Operaciones, Concepción, Chile

7. Borndörfer R, Grötschel M, Pfetsch ME (2004) Models for line planning in public transport. In: 9th International Conference on Computer Aided Scheduling of Public Transport, San Diego, United States
8. Ceder A, Wilson NHM (1986) Bus network design. Transportation Research B, 20(4):331-344
9. Chakroborty P (2003) Genetic Algorithms for optimal urban transit network design. Computer-Aided Civil and Infrastructure Engineering 18(3):184-200
10. Coello CA (2000) An updated survey of GA-based multiobjective optimization techniques. ACM Computing Surveys 32(2):109–143
11. Deb K (2001) Multi-Objective Optimization using Evolutionary Algorithms. John Wiley and Sons
12. Desaulniers G, Hickman MD (2003) Public transit. Technical Report G-2003-77 GERAD
13. Ehrgott M, Gandibleux X (2000) A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum 22:425–460
14. Ehrgott M, Gandibleux X (2004) Approximative solution methods for multiobjective combinatorial optimization. Top: Revista de la Asociación Española de Estadística e Investigación Operativa 12(1):1-89
15. Fan W, Machemehl RB (2004) A Tabu Search based heuristic method for the transit route network design problem. In: 9th International Conference on Computer Aided Scheduling of Public Transport, San Diego, United States
16. Fan W, Machemehl RB (2004) Optimal transit route network design problem: algorithms, implementations, and numerical results. Technical Report 167244-1 University of Texas at Austin
17. Hasselström D (1981) Public transportation planning - A mathematical programming approach. Doctoral Dissertation, University of Göteborg
18. Israeli Y, Ceder A (1993) Transit route design using scheduling and multiobjective programming techniques. In: Daduna J R, Branco I, Pinto J M (eds). Proceedings of the Sixth International Workshop on Computer Aided Scheduling of Public Transport, Springer
19. Jaszkiewicz A (2004) Evaluation of multiple objective metaheuristics. In: Gandibleux X, Sevaux M, Sorensen K, T'kindt V (eds). Metaheuristics for Multiobjective Optimisation, Springer
20. Lee YJ, Vuchic VR (2005) Transit network design with variable demand. Journal of Transportation Engineering 131(1):1–10
21. Magnanti TL, Wong RT (1984) Network design and transportation planning: Models and algorithms. Transportation Science 18(1):1-55
22. Mandl CE (1980) Evaluation and optimization of urban public transportation networks. European Journal of Operational Research 5(6):396–404
23. Mauttone A, Urquhart ME (2004) GRASP para el diseño de recorridos en transporte público. In: XII Congreso Latino-Iberoamericano de Investigación de Operaciones, La Habana, Cuba
24. Ngamchai S, Lovell D (2003) Optimal time transfer in bus transit route network design using a Genetic Algorithm. Journal of Transportation Engineering 129(5):510-521
25. Krishna Rao KV, Muralidhar S, Dhingra S L (2000) Public transport routing and scheduling using Genetic Algorithms. In: 8th International Conference on Computer Aided Scheduling of Public Transport, Berlin, Germany

26. Resende M, Ribeiro C (2003) Greedy Randomized Adaptive Search Procedures. In: Glover F, Kochenberger G (eds). Handbook of Metaheuristics, Kluwer Academic Publishers
27. Silman LA, Barziliy Z, Passy U (1974) Planning the route system for urban buses. Computers and Operations Research 1(2):201-211
28. Soares D, Arroyo J (2004) A GRASP algorithm for the multi-objective knapsack problem. In: XXIV International Conference of the Chilean Computer Science Society, Arica, Chile
29. Tom VM, Mohan S (2003) Transit route network design using frequency coded Genetic Algorithm. Journal of Transportation Engineering 129(2):186-195
30. Wan QK, Lo HK (2003) A mixed integer formulation for multiple-route transit network design. Journal of Mathematical Modelling and Algorithms 2(4):299-308