Towards the User Equilibrium in Traffic Assignment Using GRASP with Path Relinking

Gabriel de O. Ramos
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
goramos@inf.ufrgs.br

Ana L. C. Bazzan
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
bazzan@inf.ufrgs.br

ABSTRACT

Solving the traffic assignment problem (TAP) is an important step towards an efficient usage of the traffic infrastructure. A fundamental assignment model is the so-called User Equilibrium (UE), which may turn into a complex optimisation problem. In this paper, we present the use of the GRASP metaheuristic to approximate the UE of the TAP. A path relinking mechanism is also employed to promote a higher coverage of the search space. Moreover, we propose a novel performance evaluation function, which measures the number of vehicles that have an incentive to deviate from the routes to which they were assigned. Through experiments, we show that our approach outperforms classical algorithms, providing solutions that are, on average, significantly closer to the UE. Furthermore, when compared to classical methods, the fairness achieved by our assignments is considerably better. These results indicate that our approach is efficient and robust, producing reasonably stable assignments.

CCS Concepts

•**Theory of computation** → **Optimization with randomized search heuristics;** *Exact and approximate computation of equilibria;* •**Computing methodologies** → *Artificial intelligence; Search methodologies; Modeling and simulation;*

Keywords

traffic assignment, user equilibrium, metaheuristics, GRASP

1. INTRODUCTION

Efficient urban mobility has shown to be a major challenge towards smart cities. Traditional approaches for dealing with arising traffic issues include increasing the physical capacity of existing traffic infrastructure. Such approaches, however, have proven unsustainable from many perspectives. Moreover, as stated by the Braess' paradox [4], expanding the infrastructure's capacity may lead to a deterioration in the traffic performance. Against this background, ways of making a more efficient use of the existing infrastructure have been increasingly studied. Among such ways, traffic assignment is a particularly relevant one.

Traffic assignment represents an important step towards modelling and simulating transportation systems. Specifically, the traffic assignment problem (TAP) addresses how to efficiently connect the physical infrastructure (supply) and the vehicles[1] that are going to use it. In other words, by solving the TAP one can promote a more efficient use of the currently available traffic infrastructure.

Solving the TAP, however, is not a trivial task. As a first attempt, one may consider assigning vehicles to their corresponding shortest routes. However, such an approach may lead to the concentration of vehicles in a small number of roads (a.k.a. links), which might incur in huge congestions. The problem is also unlikely to be overcome by simply creating alternative routes. Due to the limited availability of resources (links), different routes may share the same links. In other words, the problem is more related to how to efficiently *spread* the traffic than with the number of routes to be used itself. Precisely, solving the TAP means distributing the vehicles into the road network regarding their origins and destinations (a.k.a. OD pairs) in order to minimise some associated cost (e.g., travel time, money expenses, polluting emissions). Therefore, the TAP is essentially an optimization problem.

Two fundamental traffic assignment models have been developed to solve the TAP: system optimal (SO) and user equilibrium (UE). In an SO approach, one seeks to find an assignment that minimises the travel costs on a global level, i.e., it describes the system at its best operation. However, minimising the global cost may lead to increasing the cost of some users. Consequently, some users may have an incentive to deviate from the routes to which they were assigned. Hence, an SO solution may be infeasible in practice. The UE approach, on the other hand, describes solutions from the users' behaviour point of view. In essence, in a UE configuration, every user is assigned to one of the best available routes for its OD pair. Along these lines, the advantage of a UE solution (over an SO one) is that users have no incentive to deviate from the routes they were assigned to. Thus, the UE configuration is more likely to be pursued in practice by the drivers than an SO one.

Many methods have been proposed for solving the TAP using one or both models described before, each with its strengths and weaknesses. What must be clear, however,

GECCO '15, July 11 - 15, 2015, Madrid, Spain
© 2015 ACM. ISBN 978-1-4503-3472-3/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2739480.2754755

[1] Henceforth, we use the terms vehicle and driver alternately.

is that in both approaches the search space can be huge and solving the problem optimally may be infeasible. To this respect, an alternative that has drawn attention is the development of methods capable of finding approximate solutions to the TAP. Specifically, the use of heuristics in the TAP has shown a promising direction.

In this paper, we approach the TAP in terms of UE with the GRASP+PR algorithm. GRASP+PR stands for the Greedy Randomised Adaptive Search Procedure algorithm embodied with the Path Relinking mechanism. GRASP is a heuristic search algorithm that greedily builds independent solutions and improve them with local search. The PR mechanism is used to increase the diversity of solutions. We propose a novel representation of solutions, which allows the modelling of TAP in terms of the UE. Through such a representation, GRASP+PR can be used to minimise the number of vehicles that are assigned to non-least cost routes, thus enhancing the assignments' fairness. To the best of our knowledge, this is the first approach to address the UE by minimising the amount of vehicles on non-least cost routes. Our proposal is empirically evaluated, and shown to provide solutions that are, on average, considerably fairer and closer to the UE than those from classical methods. Such results show that our approach is efficient, providing reasonably stable assignments.

The present work is organised as follows. Basic concepts are introduced in Section 2. A brief overview of related work and the GRASP+PR algorithm are presented in Sections 3 and 4, respectively. The problem modelling and the use of GRASP+PR in our approach are detailed in Section 5. The evaluation of our proposal is shown in Section 6. Concluding remarks and future directions are presented Section 7.

2. TRAFFIC ASSIGNMENT

In this section, the basic concepts surrounding traffic assignment are presented. For a more comprehensive overview the reader is referred to [16, chapter 10] or [3, chapter 4].

A road network can be represented as a directed graph $G = (N, L)$, where the set N of nodes represents the intersections, and the set L of directed links represents the roads between intersections. Each link $l_k \in L$ has an associated cost c_k, which is given as a function of some of its attributes, like length, speed limit, capacity, load/flow (of vehicles), etc. The cost of a link represents some sort of measure, like travel time, money expenses, polluting emissions, etc.

The flow of vehicles in a network (the demand) is based on the amount of trips made between different origins and destinations. Let T_{ij} be the demand for trips between origin i and destination j, i.e., the number of vehicles that need to reach j from i. The set of all such demands is represented by an origin-destination (OD) matrix $T = \{T_{ij} \mid \forall i, j \in N,\ i \neq j,\ T_{ij} \geq 0\}$. A trip is made by means of a route $R \subseteq L$, which is a sequence of links that connects an origin to a destination. The cost C_R of a given route R can be obtained by summing up the costs of the links that comprise it, as in Equation (1).

$$C_R = \sum_{l_k \in R} c_k \qquad (1)$$

In order to compute the cost of the links, the concept of volume-delay functions (VDF) needs to be introduced. A VDF is a well-known abstraction used to express the delay (cost) of a link depending on the volume (flow) of vehicles on it. In other words, VDFs provide a means to study how the flow affects the cost on a link. Such a concept is important because the TAP is typically studied in a macroscopic level, i.e., the vehicles movement is considered in an abstract way. A VDF takes as input the flow of vehicles within a link and, based on the characteristics of such link (e.g., length, capacity, etc.), returns a cost on it. A simple VDF is presented in Equation (2), with t_k and f_k representing, respectively, the free flow travel time (minimum travel time when the link is not congested), and the number of vehicles on link $l_k \in L$. This particular VDF represents the travel time on link k, which is increased by 0.02 for each vehicle/hour of flow.

$$c_k(f_k) = t_k + 0.02 \times f_k \qquad (2)$$

As previously stated, two fundamental models to solve the TAP are System Optimal (SO) and User Equilibrium (UE). Recall that an SO assignment describes the system as its best operation. On the other hand, UE stands for the traffic condition stated by the Wardrop's first principle: "under equilibrium conditions traffic arranges itself in congested networks such that all used routes between an OD pair have equal and minimum costs while all unused routes have greater or equal costs" [22]. We refer the reader to [11] for the mathematical formulation of these models. That said, although the SO model may seem the most efficient one, it has the disadvantage that some users may have no incentive to actually follow the routes to which they were assigned. In fact, some users would change their routes to those with minimum cost. Thus, the UE model has shown a more realistic way of assigning routes to vehicles.

The UE model can be analytically stated as an optimization problem. However, this model has the drawback of not being solvable algebraically, except for very simple scenarios. Hence, providing approximate solutions to the problem has shown a promising direction. Before advancing to this topic, some traditional methods need to be introduced.

In general, methods for the TAP involve two steps [16]: (i) using a set of rules to identify desirable routes (e.g., the fastest) in each OD pair (this is usually performed by shortest path algorithms), and (ii) assigning suitable portions of the OD trips to these routes so that a given objective is achieved. It should be noted, however, that the desirability of a given route may vary due to congestion effects. Specifically, desirable routes may become less attractive than others as the flow on their links increase. On this basis, the two steps above are generally performed iteratively, until a given (non-iterative) convergence criterion is achieved.

The simplest traffic assignment method is known as *all-or-nothing*, which perform the task without accounting for congestion effects. Specifically, the all-or-nothing method selects the shortest route for each OD pair, and then assigns the entire corresponding flow to that route. This approach, however, is quite naïve, since assigning all vehicles to the same routes (one per OD pair) can lead to congestions. Although this method is of low interest in practice, it has been used by more sophisticated methods. Furthermore, it may be useful as an upper bound.

More sophisticated methods for traffic assignment are iterative ones, which account for congestion. One of these methods is the so-called *incremental assignment*. The incremental assignment method iteratively loads proportional fractions p_n (with $\sum_n p_n = 1$) of the demand into the lowest cost routes (one for each OD pair). A new set of lowest cost

Algorithm 1: Pseudocode of MSA (adapted from [16])

```
1 select an initial set of current link costs (usually free-flow
  travel times); initialise the flow of every link $k$: $f_k = 0$; set
  $n = 0$;
2 create an auxiliary all-or-nothing assignment based on
  current link costs, obtaining a set of auxiliary flows $F$;
3 calculate the current flow of every link $k$ as:
  $f_k^n = (1 - \psi) f_k^{n-1} + \psi F_k$, with $\psi \in [0, 1]$;
4 for every link k, update the cost $c_k$ based on the flow $f_k^n$; if
  the predefined stop criterion was achieved, then stop;
  otherwise proceed to step 2;
```

routes is created (using the all-or-nothing assignment) on every iteration based on the costs resulting from the flows accumulated on previous iterations. A typical set of values for p_n is $\{0.4, 0.3, 0.2, 0.1\}$. This method, however, does not have convergence guarantees, because, once a flow is assigned, it cannot be changed from one iteration to another.

Another iterative method that accounts for congestion effects is the *method of successive averages* (MSA). MSA iteratively generates an all-or-nothing assignment, and changes a fraction ψ of the previous iteration's assignment according to the all-or-nothing one. This method is repeated until a given stop criterion is achieved. To be precise, the flow at the n-th iteration is a linear combination of the flow at the $(n-1)$-th iteration and an auxiliary flow resulting from an all-or-nothing assignment in the n-th iteration. The MSA can be formalised as in Algorithm 1. It has been shown [20] that MSA produces solutions convergent to the UE when $\psi = 1/n$ (with $n \to \infty$). However, this method may take many iterations, especially in more complex scenarios, thus being very inefficient.

3. RELATED WORK

Many approaches have been proposed to solve the TAP from many perspectives. Broadly speaking, we can classify these approaches into two fronts: global, when the assignment is performed by a central authority in charge of this process; and individual, when each vehicle is responsible for choosing its own route. In this section we present some relevant works on each of these fronts, starting with the former.

A genetic algorithm for the TAP is presented in [2] and [6], aiming at computing the SO. In these approaches, each vehicle has a corresponding position in the chromosome, and the value within this position corresponds to one among k predefined routes (regarding the vehicle's OD pair) to which the vehicle can be assigned. The main difference between these two approaches lies in the way the traffic is simulated: in [2] the simulation is macroscopic, whereas in [6] it is microscopic. As opposed to the macroscopic settings that we have seen so far, in a microscopic simulation the vehicles' movement is considered in detail (speed, gap between vehicles, etc.). Although microscopic simulation is more realistic, it is much more complex to handle. Despite of that difference, both approaches present good results regarding the purpose to which they were proposed. The UE, however, is not addressed by these approaches. Moreover, the size of the neighbourhood resulting from the proposed modelling is excessively large.

The TAP is approached from a game theoretic point of view in [1]. In this approach, the problem is modelled as a population game, where each population (i.e., vehicles from the same OD pair) has a probability vector (a.k.a. strategy) representing its possible routes. A genetic algorithm is used to evolve the strategy profiles, aiming at the UE. Results show that the UE may be achieved under certain mutation rates. However, this approach considers just three populations.

The TAP is indirectly addressed by means of the toll booth problem (TBP) in [5]. The TBP consists in selecting links to charge tolls so that drivers have a monetary incentive to spread over the network. In this sense, by putting tolls in some links, one can induce drivers to act on behalf of the social welfare, bridging the gap between the UE and SO. In this work, a biased random-key genetic algorithm is used to accomplish this task. However, this approach depends on a high number of tolls to achieve good results, what may be impractical to deploy. Moreover, charging tolls is not a popular alternative from the drivers' viewpoint.

In [21], a genetic algorithm is presented for optimising traffic signal settings. Basically, on each iteration of the algorithm, the evolved traffic signal settings are used as input for a modified version of MSA. The aim here is to jointly optimize the traffic signal settings and the TAP. This approach, however, is more focused on the former problem.

A fuzzy-based route choice model was proposed in [14] to tackle the uncertainties regarding the drivers' decision making. The basic idea is to maintain routes' cost predictions in a fuzzy set, thus representing knowledge imprecisions and traffic uncertainties. On the basis of such representation, the model allows the association of a degree of attractiveness with each route. This work, however, is more concerned with tackling the uncertainties of the process than with the assignment optimality.

In [18], a percentile-based route choice model is used to account for the travel time budgets of drivers. The employed model assumes that drivers make decisions with respect to the route travel time distributions collected from past experiences. In this approach, the focus is on identifying the amount of time that each vehicle must allocate to its trip in order to reach its destination on time. However, this approach does not compute the SO nor the UE.

Regarding the aforementioned second front, agent-based approaches have shown to be a natural way of modelling traffic problems. The work of [8] presents an Inverted Ant Colony Optimization (IACO) algorithm for minimising the SO in a decentralised, microscopic way. IACO is a variant of the ACO algorithm in which pheromone repels, instead of attracting, ants. To this respect, vehicles deposit pheromone into the roads they travel, and avoid travelling through congested ones. The objective is to keep routes as diverse as possible. However, the pheromone needs to be kept by a central authority, which is a quite strong requirement for such a distributed system. Moreover, this work does not provide any statement on the optimality of the solutions.

A game-theoretic approach is also used in [10], where an evolutionary version of the minority game is employed. In this work, each driver has a set of strategies used to predict the links occupancy based on historic data. Each driver scores its strategies based on occupancy and travel times previously experimented by themselves. However, this approach assumes that historic network information is available to all drivers. Furthermore, the approach was validated in a quite simple scenario.

A number of recent agent-based approaches make use of reinforcement learning algorithms to promote route choice in a distributed fashion. Some examples include: [19] using learning automata; and [13] using Q-learning and considering how a single agent impact the global performance. In these approaches, the drivers know their OD pairs but do not know their routes a priori. In this sense, intersections are decision points, where drivers can decide what link to take next. This kind of *en-route* decision allows the drivers to account for the variability in their travel times due to their peers' decisions. The objective here is to compute the SO. However, as previously discussed, the SO is not realistic when the behaviour of selfish drivers is being considered.

The impact of different levels of information in the route choice process has increasingly attracted attention. The idea here is to provide an understanding on how the demand reacts to traffic information. Some works that study the impact of traffic information in the drivers' route choice behaviour are: [7], in which neural networks were used to predict the compliance level of drivers who receive traffic suggestions; and [17], where a fuzzy-based system is used to predict the route behaviour of drivers who are provided with traffic information. However, the focus of these approaches is on the impact of the traffic information itself, and none of them is assessed in terms of UE or SO.

Therefore, as can be seen, many approaches discussed here do not fully address the UE. Moreover, most approaches presented here for the UE are suitable for small scenarios only. An exception is the work of [5], which on the other hand has a strong dependence on a high number of tools.

4. THE GRASP+PR ALGORITHM

4.1 Greedy Randomised Adaptive Search Procedure

GRASP is a multistart greedy metaheuristic introduced in [9]. Basically, GRASP works as follows: at each iteration a greedy randomised solution is built from scratch and a local search procedure is used to improve it. GRASP is multistart in a sense that iterations are independent among themselves, i.e., the knowledge acquired in previous iterations is not used in the subsequent ones. With regard to the final result, since the algorithm is multistart, the best solution found so far is stored and returned in the end of the execution. We now present some basic concepts, slightly adapted to the context of the current work.

The problem to be addressed by GRASP is defined by a ground set $E = \{e_1, \dots, e_{|E|}\}$, a set of feasible solutions $F \subseteq 2^E$, and an objective function $\phi : 2^E \to \mathbb{R}$. A solution $S \in F$ is typically represented by a vector over the ground set E. Considering a minimisation problem, the aim is to find an optimal solution $S^* \in F$ such that $\phi(S^*) \leq \phi(S), \forall S \in F$. The definitions of ground set, objective function, and feasible solutions are specific to each problem. In this regard, one possible representation of the TAP would be: E as the set of drivers, ϕ as the travel cost and F as all possible assignments.

In a greedy heuristic, solutions are built progressively: at each step, a single element of the ground set is selected and included in the solution under construction, until a feasible[2] solution is obtained. The selection process evaluates how much the inclusion of each element increases the solution cost. Based on such evaluation, a common selection strategy is the α-greedy, in which one selects among the $\alpha\%$ best elements uniformly at random.

It must be remarked that the greedily generated solutions may not be optimal, thus a local search procedure is employed by GRASP to improve them. A local search works by successively applying small changes in a solution until a (local) optimum is reached. Each such change is called a *movement*, and each solution produced by a single movement is called a *neighbour*. The *neighbourhood* of a solution S is the set $N(S) \subseteq F$. A solution S is said to be a local optimum if $\phi(S) \leq \phi(S'), \forall S' \in N(S)$. Different strategies can be employed to guide the local search. A common strategy is known as *best improvement*, in which the search is restricted to the best neighbourhood $B(S) = \{S' \in N(S) \mid \phi(S') < \phi(S)\}$, and a best neighbour is selected uniformly at random.

4.2 Path Relinking

The path relinking (PR) mechanism was initially proposed in [12] within the context of tabu search. Broadly speaking, PR explores the trajectory between an initial and a target solution. The result of PR is the best solution found in the traversed trajectory.

Some definitions are necessary. Let S_s and S_t be the initial and target solutions. The distance $\Delta(S_s, S_t)$ between two given solutions measures how many movements are necessary to transform one into the other. The PR mechanism works by exploring a directed neighbourhood $D(S_s) = \{S' \in N(S_s) \mid \Delta(S', S_t) < \Delta(S_s, S_t)\}$. A common strategy here is also the best improvement. In this sense, the search is performed from S_s to S_t and takes at most $\Delta(S_s, S_t)$ iterations.

In the context of GRASP, the PR mechanism is applied at every iteration, after the local search procedure is applied to the incumbent solution S. A set P of reference solutions is maintained. This set can be seen as a kind of memory, and it is defined to store the best and more diverse solutions found so far. From this set, a solution S_t can be selected as target. A common selection strategy here is to select a solution S' from P with probability proportional to $\Delta(S, S')$. This strategy gives priority to farthest targets, thus promoting a higher coverage of the search space. GRASP+PR was introduced in [15].

5. TRAFFIC ASSIGNMENT WITH GRASP+PR

5.1 Modelling

5.1.1 Solution Encoding and Evaluation

A solution S for the TAP represents an assignment of vehicles into routes. Recall that the demand is represented by an OD matrix $T = \{T_{ij} \mid \forall i, j \in N, i \neq j, T_{ij} > 0\}$, where T_{ij} is the demand for trips between origin i and destination j. A trip is made through a route $R \subseteq L$. In this work, we limit the available routes for each OD pair to the k shortest ones, among which the demand must be partitioned. Let $R_{ij} = \{R^1_{ij}, \dots, R^k_{ij}\}$ be the set of k shortest routes between

[2] Infeasible solutions may also be produced. Our approach, however, prevent them to occur (as shown in Section 5). Thus, we omit such concept to simplify the notation.

origin i and destination j. These sets are computed considering free flow travel times, and this process can be performed by the KSP algorithm [23].

Given the above definitions, a natural approach is to represent the problem as a vector of integers (as in [2, 6]), where the value in the i-th position identifies the route to which the i-th vehicle was assigned. In such a representation, however, the size of a solution is equal to the total demand, which may be huge. On this regard, in this paper we employ a more compact solution representation, where vehicles are grouped by OD pair and route. Specifically, a solution S is a matrix of size $|T| \times k$, where rows represent OD pairs, and columns represent the available routes for these OD pairs. A value on position $S[ij][r]$ represents the amount of vehicles from T_{ij} assigned to route R^r_{ij}. Observe that the r-th route from two distinct OD pairs are obviously not the same. Based upon such representation, the size of a solution is $\Theta(|T| \times k)$, which is lower than of the representation in [2, 6].

The solution evaluation is defined in terms of the UE. Remark that, under UE, drivers are assigned to the least cost routes, thus having no incentive to deviate from such assignment. In this sense, we measure the performance of an assignment by counting how many vehicles were *not* assigned to their respective least cost routes, i.e., we measure those that would have an incentive to deviate from the assignment. The value of a solution S can be formalised by $\phi : S \to \mathbb{N}$ as in Equation (3), where $R^*_{ij} = \min_{R \in R_{ij}} C(R)$ represents any[3] least cost route of the OD pair ij under assignment S. In the function, the farther the solution is from the UE, the higher its value, and a value 0 means the UE itself. Thus, the objective here is to minimise ϕ. It is worth noting that minimising ϕ promotes fairer assignments, since fewer drivers are assigned to routes worse than those of their peers.

$$\phi(S) = \sum_{T_{ij} \in T} \sum_{R^r_{ij} \in R_{ij}} \begin{cases} S[ij][r] & \text{if } C(R^r_{ij}) > C(R^*_{ij}) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

5.1.2 Neighbourhood

We now define the neighbourhood of a given solution S. Recall that a neighbour is obtained by performing a single movement in the solution. A movement here is characterised by moving a single vehicle from any route $R^r_{ij} \in \{R^r_{ij} \in R_{ij} \mid C(R^r_{ij}) > C(R^*_{ij})\}$ to R^*_{ij}. This means that at most $k-1$ movements can be done for each of the $|T|$ OD pairs. Thus, the number of neighbours is $O(|T| \times k)$ (we omit the proof due to the space limitation). Observe that, as the value of solutions approaches zero (the UE), the number of least cost routes increases, and, consequently, the number of neighbours decreases.

Two interesting properties of the present modelling must be highlighted. The first is that infeasible solutions are automatically avoided. An infeasible solution would have either a non-natural number (in any position) or a total demand different from that of the OD matrix. However, a movement consists in changing the route of a single vehicle, thus the demand does not change, and empty routes can only receive vehicles. The second property is that our neighbourhood definition avoids redundant neighbours. The definition would otherwise allow redundant neighbours if the movement of *each* vehicle from route R^r_{ij} to route R^*_{ij} were considered in particular. In this case, such neighbours would be redundant because, from the solution definition perspective, they would represent exactly the same solutions.

```
Algorithm 2: Overview of implemented GRASP+PR
Input: MaxIterations          // number of iterations
       $\alpha$               // due to the $\alpha$-greedy strategy
       $\beta$                // size of the reference set
       $\gamma$               // probability of uniform solutions
$S \leftarrow$ initialise best solution so far;
$P \leftarrow$ initialise with $\beta$ uniform solutions;
for $i \in \{1, \ldots, \text{MaxIterations}\}$ do
    $S' \leftarrow \begin{cases} \mathit{GreedySolution}(\alpha) & \text{with prob. } 1-\gamma \\ \mathit{UniformSolution}() & \text{otherwise} \end{cases}$ ;
    $S' \leftarrow$ LocalSearch($S'$);
    $S'' \leftarrow$sample a solution from $P$;
    $S' \leftarrow \mathit{PathRelinking}(S', S'')$;
    $S' \leftarrow$ LocalSearch($S'$);
    if $\phi(S') < \phi(S)$ then
        $S \leftarrow S'$;
    end
    $P \leftarrow$ update reference set with $S'$;
end
return $S$;
```

5.2 Methods

A general overview of our GRASP+PR approach is presented in Algorithm 2.

5.2.1 Solution Generation

In GRASP, solutions are generated in a randomised greedy way. However, to prevent any bias in the generation procedure, we also allow solutions to be generated uniformly. These two procedures are used alternately, according to parameter γ: a solution is greedily generated with probability $(1-\gamma)$, or uniformly generated with probability γ. Both procedures are described next.

In the greedy procedure, the ground set E has one element for each OD pair and route (considering its k routes). Let $E^r_{ij} \in E$ be the element corresponding to route r of OD pair ij, and let its cost be the amount that $\phi(S)$ would increase if $S[ij][r]$ were incremented by 1. Given this, the process starts with an empty solution S. On each iteration of the procedure, the cost of every element in E is updated and the α-greedy strategy is used to select (uniformly at random) one among the α% best such elements. After the element of the ground set is selected, the corresponding position in the solution is incremented by 1. This process is repeated until all vehicles are assigned to a route, thus preventing the predefined demand from being exceeded.

The uniform generation procedure is much simpler. The process also starts with an empty solution S. Consider an OD pair ij, its demand T_{ij}, and the corresponding set of routes R_{ij}. Let $U_{ij} = \{0, u_2, \ldots, u_k, T_{ij}\}$ be a sorted set, where $u_2, \ldots, u_k$ correspond to $k-1$ natural numbers in the interval $[0, T_{ij})$, generated uniformly at random. The solution is defined as $S[ij][r] = U_{ij}[r+1] - U_{ij}[r]$ for every route r of OD pair ij. This process is performed for every OD pair. The result is a feasible assignment, with demands distributed uniformly at random among the available routes.

5.2.2 Local Search and Path Relinking

Once a solution is generated, a local search procedure is applied to reach its corresponding local optimum. Specif-

[3] As many routes may have the same cost, we refer to R^*_{ij} as any least cost route. Ties are broken at random.

ically, given a solution S, the procedure explores its best neighbourhood $B(S)$ using the best improvement strategy. The process is performed until the local optimum is reached.

The PR mechanism is performed at each iteration, after the local search procedure. As previously mentioned, PR explores the trajectory between two given solutions. In this sense, let S be the incumbent solution, $\Delta(S, S')$ the distance between two given solutions, and P be the set of reference solutions. A target solution S_t is selected from P with probability proportional to $\Delta(S, S_t)$, i.e., the greater the distance to the incumbent solution, the higher the probability of being selected. Once a solution is selected, PR employs a best improvement search strategy to traverse from S to S_t. The best solution found in such traverse is returned, and a local search is also applied on it.

Concerning the reference set, its size is limited to keep at most β solutions. In this sense, P is maintained as follows. Initially, P is populated with β uniformly generated solutions. Next, on each iteration of GRASP, an incumbent solution S can replace another in P if its cost is lower than the worst in P, i.e., if $\phi(S) < \max_{S' \in P} \phi(S')$. In this case, S replaces the solution $S' = \arg\min_{S' \in P} \Delta(S, S')$, i.e., the one which is closest to him. This update mechanism on the reference set is performed at the end of each iteration.

6. EMPIRICAL EVALUATION

6.1 Experiments

Our approach is evaluated in a non-trivial road network introduced in the exercise 10.1 of [16] (henceforth OW). The OW network is shown in Figure 1, where the numbers on links represent their free flow travel time (in minutes). This scenario has four OD pairs: AL, AM, BL, and BM, with a demand of 600, 400, 300, and 400 vehicles, respectively. The cost of the links is obtained by means of the simple VDF in Equation (2), though more sophisticated ones could be used as well.

The performance of solutions is measured by means of Equation (3), introduced in present work. An additional, classical, performance metric, namely that of Equation (4), is also employed here. This latter metric accounts for the relative cost of travel times of non-least cost routes (further details in [16]). Hereafter, we refer to equations (3) and (4) as ϕ and δ, respectively. Remark that both ϕ and δ measure how far a given solution is from the UE. Finally, the average travel time (*avg-tt* hereafter, measured in minutes) is employed as a third performance metric. For the three metrics, the lower the resulting value, the better.

$$\delta(S) = \frac{\sum_{T_{ij} \in T} \sum_{R^r_{ij} \in R_{ij}} S[ij][r] \times (C(R^r_{ij}) - C(R^*_{ij}))}{\sum_{T_{ij} \in T} T_{ij} \times C(R^*_{ij})} \qquad (4)$$

Experiments were set up as follows. The set of shortest routes is generated only once, using the KSP algorithm [23]. These sets of routes are presented in Table 1, for $k = 4$. Regarding the parameters configurations for GRASP+PR, different combinations of values for these were tested. For each such combination, 30 repetitions were performed, where a repetition corresponds to a single execution of the algorithm with 1000 iterations. This calibration process is presented in Appendix A and the best configuration found was $k = 4$, $\alpha = 0.2$, $\beta = 20$, and $\gamma = 0.5$. The results of such configuration were selected for further analysis in the next subsection.

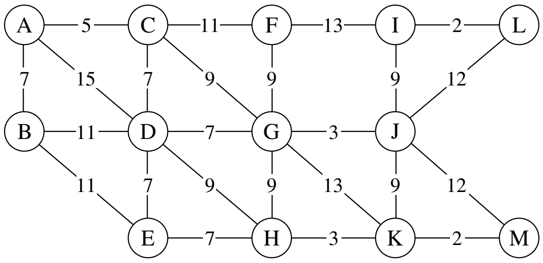

Figure 1: OW road network (adapted from [16]).

Table 1: Set of $k = 4$ shortest routes per OD pair

OD	Route 1	Route 2	Route 3	Route 4
AL	ACGJIL	ACGJL	ACFIL	ACDGJIL
AM	ACDHKM	ACGJKM	ACGHKM	ADHKM
BL	BDGJIL	BDGJL	BACGJIL	BACGJL
BM	BEHKM	BDHKM	BDEHKM	BEDHKM

Our approach is compared against the MSA, introduced in Section 2, with $\psi = 1/n$. In the case of MSA, no repetitions were needed since it is deterministic. We note that MSA has converged in around 450 iterations. Student's t-tests were applied to compare our approach against MSA. In this sense, henceforth, any claim that one approach is better than the other is supported by such tests at the 5% significance levels, except when otherwise stated. We omit the p-values due to the lack of space.

6.2 Results and Discussion

The performance of GRASP+PR along iterations is presented in Figure 2. In the plot, results represent the average and standard deviation over 30 repetitions. Recall that the incumbent solution is the one generated in the current iteration of the algorithm, whereas the best solution so far is the one that shall be returned at the end of the algorithm. As can be seen, the average value of incumbent solutions varies considerably. This happens because of the independent, randomised nature of the solutions generation process. On the other hand, it can also be seen that both the average value of incumbent solutions and the best so far decrease with time. This is due to the path relinking mechanism, which keeps

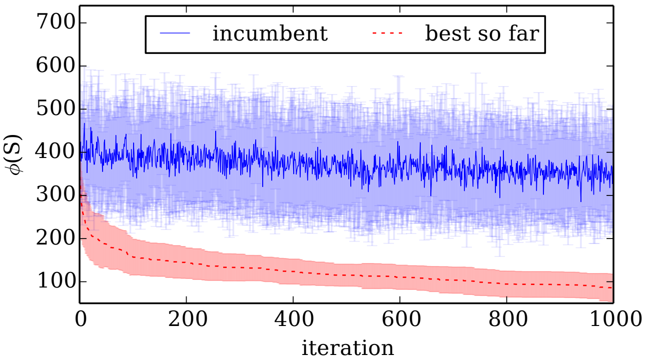

Figure 2: Average performance of GRASP+PR along iterations. Shaded lines represent the standard deviation.

Table 2: Performance comparison per OD pair and overall

	MSA			GRASP+PR		
OD	ϕ	δ	*avg-tt*	ϕ	δ	*avg-tt*
AL	215	**0.0001**	71.21	**23.37 (31.53)**	0.0011 (0.002)	**68.39 (0.52)**
AM	194	0.0032	**65.74**	**28.60 (30.71)**	**0.0007 (0.001)**	74.56 (0.56)
BL	62	0.0020	69.35	**13.87 (24.08)**	**0.0010 (0.002)**	**66.19 (0.64)**
BM	67	**0.0001**	**62.13**	**20.17 (16.42)**	0.0012 (0.001)	69.67 (0.53)
Overall	538	0.0053	**67.46**	**86.00 (32.56)**	**0.0039 (0.002)**	69.75 (0.20)

a kind of memory by means of the reference set. In this regard, the knowledge from previous iterations is exploited to improve the incumbent solutions.

The comparison of our approach against MSA is shown in Table 2, in terms of ϕ, δ, and *avg-tt* . Results are presented by OD pair as well as overall. In the case of GRASP, each result represents the average over 30 repetitions, thus standard deviations are also shown (in parenthesis). The best results for each combination of OD pair and metric are shown in bold.

Initially, consider the overall results. It can be observed that, on average, the solutions provided by GRASP+PR are considerably closer to the UE than those by MSA. In other words, our assignments are fairer than those of MSA. This is due to our problem formulation, which minimises the number of vehicles assigned to non-least cost routes. On the other hand, the solution yielded by MSA has a lower average travel time. However, the point here is that our approach is concerned with approximating the UE, not minimising the average travel costs. In fact, this latter is the focus of SO, not UE. The advantage of MSA in the *avg-tt* metric can be explained by its structure: it works by iteratively changing the route of a dwindling portion of the demand in order to narrow the gap among the vehicles' travel times. However, from the UE definition, minimising travel time per se is not the main objective.

Now, consider the results per OD pair. Considering the metric ϕ, GRASP+PR clearly outperforms MSA. The reason here is obvious: our approach was designed to minimise ϕ. In this same sense, it would be natural to expect MSA outperforming GRASP+PR in the *avg-tt* metric. However, it can be seen that each algorithm outperforms the other in half of the cases, and the same holds for the δ metric. For the *avg-tt* metric, one can see that our approach performed better in the pairs going to L, whereas MSA performed better in the pairs going to M. These differences are due to the structure of the approaches, and to the specific road network under consideration here. In the case of δ, however, it can be observed that our approach fared better in the OD pairs AM and BL. Assigning routes to these two OD pairs is more difficult since all of their routes overlap in at least one link. On the other hand, MSA presented better results for the OD pairs AL and BM, whose sets of available routes have no overlapping links between them. Thus, such results show that our approach provides fairer assignments, and, at least for the OW network, is more robust than MSA.

Regarding the *avg-tt* metric, some additional aspects (not shown here in their totality) can be analysed. The average difference between the best and worst routes of each OD pair is useful to measure the level of unfairness of an assignment. In our approach, this difference accounts for 4.38 (1.9) minutes, whereas in MSA it accounts for 7.39 minutes. When considering the best and worst routes over all OD pairs, in our approach this difference is of 12 (1.5) minutes, whilst in MSA it is of 22 minutes. As seen, the level of unfairness achieved by our approach is lower than of MSA. This means that, compared to MSA, the level of unfairness obtained by our approach has less impact on the vehicles. Furthermore, the number of vehicles that actually suffer from such unfairness is considerably lower in our assignments, as shown through the ϕ metric.

Summing up, the results achieved by our approach demonstrate its efficiency, fairness, and robustness to approximate the UE of the TAP. Despite the heuristic nature of our approach, it became evident that our solutions are much fairer and closer to the UE than those provided by the MSA. Indeed, the results obtained by our approach are, on average, 26% (δ metric) and 84% (ϕ metric) better than those of MSA. Furthermore, regarding the best solution found by our approach over all iterations, its values were only $\phi = 17$ and $\delta = 0.000916$. Naturally, the results presented here correspond to the OW network and may not hold for specific road networks. Such a fact, however, does not invalidate the generality of our results, since the OW network presents the same problems of other bigger road networks, such as congestions, routes overlapping, multiple OD pairs, etc.

7. CONCLUSIONS

Solving the traffic assignment problem (TAP) is an important step towards an efficient usage of the traffic infrastructure. In this paper, we presented the use of the GRASP+PR metaheuristic to approximate the user equilibrium (UE) of the TAP. A novel performance metric was introduced, which measures the amount of vehicles assigned to non-least cost routes. Through experiments, we show that our approach outperforms the well-known method of successive averages (MSA), providing solutions that are, on average, significantly closer to the UE. Furthermore, our approach has shown to provide assignments that are considerably fairer than those of MSA.

In spite of the satisfactory results achieved, there is still space for improvements. We have introduced a performance metric and showed that other metrics can also be employed. An interesting approach here would be to employ a multi-objective method in order to optimise several functions at the same time. Another interesting direction would be to model the UE in a disaggregate, agent-based nature. Such direction would lead to more realistic assignment methods from the drivers' point of view. Improving the process for generating the set of routes would also be useful. Some ideas in this direction include: allowing a variable number of routes per OD pair (or even per vehicle), and updating the set of routes periodically based on costs from previous iterations. Minor improvements include investigating other strategies for neighbourhood exploration, solutions construction, and path relinking.

8. ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments. The authors are partially supported by CAPES and CNPq.

9. REFERENCES

[1] A. L. C. Bazzan. Co-evolution of the dynamics in population games: the case of traffic flow assignment. In *Proc. of the 14th An. Conf. on Genetic and Evol. Comp. (companion)*, pages 43–50, 2012. ACM.

[2] A. L. C. Bazzan, D. Cagara, and B. Scheuermann. An evolutionary approach to traffic assignment. In *2014 IEEE Symp. on Comp. Intel. in Vehicles and Transp. Systems (CIVTS)*, SSCI, pages 43–50, 2014.

[3] A. L. C. Bazzan and F. Klügl. *Introduction to Intelligent Systems in Traffic and Transportation.* Morgan and Claypool, 2013.

[4] D. Braess, A. Nagurney, and T. Wakolbinger. On a paradox of traffic planning. *Transportation Science*, 39(4):446–450, 2005.

[5] L. S. Buriol, M. J. Hirsh, P. M. Pardalos, T. Querido, M. G. Resende, and M. Ritt. A biased random-key genetic algorithm for road congestion minimization. *Optimization Letters*, 4:619–633, 2010.

[6] D. Cagara, A. L. C. Bazzan, and B. Scheuermann. Getting you faster to work: A genetic algorithm approach to the traffic assignment problem. In *Proc. of the 16th GECCO (comp.)*, pages 105–106, 2014. ACM.

[7] H. Dia and S. Panwai. Modelling drivers' compliance and route choice behaviour in response to travel information. *Journal of Nonlinear Dynamics*, 49(4):493–509, 2007.

[8] J. C. Dias, P. Machado, D. C. Silva, and P. H. Abreu. An inverted ant colony optimization approach to traffic. *Engineering Applications of Artificial Intelligence*, 36(0):122–133, 2014.

[9] T. A. Feo and M. G. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2):67–71, 1989.

[10] S. M. Galib and I. Moser. Road traffic optimisation using an evolutionary game. In *Proc. of the 13th GECCO (companion)*, pages 519–526, 2011. ACM.

[11] C. Gawron. *Simulation-based traffic assignment.* PhD thesis, University of Cologne, Cologne, Germany, 1998.

[12] F. Glover. *Interfaces in Computer Science and Operations Research*, chap. Tabu Search and Adaptive Memory Programming – Advances, Applications and Challenges, pages 1–75. Springer, 1997.

[13] R. Grunitzki, G. de O. Ramos, and A. L. C. Bazzan. Individual versus difference rewards on reinforcement learning for route choice. In *2014 Brazilian Conf. on Intelligent Systems (BRACIS)*, pages 253–258, 2014.

[14] V. Henn. Fuzzy route choice model for traffic assignment. *Fuzzy Sets and Sys.*, 116(1):77–101, 2000.

[15] M. Laguna and R. Martí. Grasp and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11(1):44–52, 1999.

[16] J. Ortúzar and L. G. Willumsen. *Modelling Transport.* John Wiley & Sons, 3rd edition, 2001.

[17] S. Peeta and J. Yu. A hybrid model for driver route choice incorporating en-route attributes and real-time information effects. *Net. and Sp. Econ.*, 5:21–40, 2005.

[18] A. J. Pel and A. J. Nicholson. Network effects of percentile-based route choice behaviour for stochastic travel times under exogenous capacity variations. In *Proc. of the 16th IEEE ITSC*, pages 1864–1869, 2013.

[19] G. de O. Ramos and R. Grunitzki. An improved learning automata approach for the route choice problem. In Koch, Meneguzzi, and Lakkaraju, eds., *Agent Technology for Intelligent Mobile Services and Smart Societies*, volume 498 of *CCIS*, pages 56–67. Springer, 2015.

[20] Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis With Mathematical Programming Methods.* Prentice-Hall, Englewood Cliffs, USA, 1984.

[21] H. Varia, P. Gundaliya, and S. Dhingra. Application of genetic algorithms for joint optimization of signal setting parameters and dynamic traffic assignment for the real network data. *Research in Transportation Economics*, 38(1):35–44, 2013.

[22] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers*, volume 2, pages 325–378, 1952.

[23] J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

APPENDIX

A. PARAMETERS TUNING

Here we present a brief analysis on the values defined for the parameters involved in our approach. The test scenario is the same as in Section 6, namely the OW network. Regarding the parameters, the following values were defined for each of them: $k = \{4, 10\}$, $\alpha = \{0.2, 0.5, 0.8\}$, $\beta = 20$, $\gamma = \{0.2, 0.5, 0.8\}$. The number of iterations per execution of the algorithm was set to 1000 in all cases. To improve the accuracy of results, each combination of parameters was repeated 30 times.

The present scenario characterises a factorial experiment, where factors are represented by the parameters, levels by the parameter's values, and the response variable corresponds to the value of solutions. In this sense, the factorial analysis of variation (ANOVA) test was employed to check for statistic equivalence of the parameters' values. In ANOVA, a null hypothesis is formulated for each factor, stating that its different levels produce the same mean response, i.e., all values of the given parameter produce the same results on average. The alternative hypothesis is that not all levels produce the same mean response. All tests here were applied at the 5% significance levels.

The resulting p-values for factors k, α, and γ, were 2×10^{-16}, 0.6323, and 4.6×10^{-16}, respectively. This means that no significant differences among the values of parameter α were detected. In this sense, we adopt $\alpha = 0.2$. On the other hand, the null hypothesis was rejected for parameters k and γ. We go forward on these factors, and apply a post-hoc test, namely the Student's t-test. Starting with parameter k, the null hypothesis is the same as before, but with the alternative hypothesis stating that $k = 4$ produces better solutions than $k = 10$. The p-value is 2.2×10^{-16}, meaning that $k = 4$ is better. Indeed, the average value of solutions when $k = 4$ is 93.90 ± 33.09, which is much lower than the 503.78 ± 59.76 of $k = 10$. Now, for parameter γ, we fix $\alpha = 0.2$ and $k = 4$. One test is then applied for each pair of values for γ. The resulting p-values 3.3×10^{-5}, 5×10^{-8}, and 0.1593, indicate that, respectively, $\gamma = 0.5$ is better than $\gamma = 0.2$, $\gamma = 0.8$ is better than $\gamma = 0.2$, and that $\gamma = 0.5$ and $\gamma = 0.8$ are equivalent. Thus, we adopt $\gamma = 0.5$ to provide a good balance between greedy and uniform solutions.