Complex & Intelligent Systems (2022) 8:2001–2015
https://doi.org/10.1007/s40747-021-00444-4

ORIGINAL ARTICLE

Reinforcement learning for the traveling salesman problem with refueling

André L. C. Ottoni[1] · Erivelton G. Nepomuceno[2] · Marcos S. de Oliveira[3] · Daniela C. R. de Oliveira[3]

Received: 1 December 2020 / Accepted: 10 June 2021 / Published online: 16 June 2021

Abstract

The traveling salesman problem (TSP) is one of the best-known combinatorial optimization problems. Many methods derived from TSP have been applied to study autonomous vehicle route planning with fuel constraints. Nevertheless, less attention has been paid to reinforcement learning (RL) as a potential method to solve refueling problems. This paper employs RL to solve the traveling salesman problem With refueling (TSPWR). The technique proposes a model (actions, states, reinforcements) and RL-TSPWR algorithm. Focus is given on the analysis of RL parameters and on the refueling influence in route learning optimization of fuel cost. Two RL algorithms: Q-learning and SARSA are compared. In addition, RL parameter estimation is performed by Response Surface Methodology, Analysis of Variance and Tukey Test. The proposed method achieves the best solution in 15 out of 16 case studies.

Keywords Reinforcement learning · Traveling salesman with refueling problem · Tuning of parameters

Introduction

The traveling salesman problem (TSP) is one of the best-known combinatorial optimization problems and is often considered in autonomous vehicle route planning [11,19,31,48,50,65,80]. In a TSP, the sequence of autonomous agent movements should optimize a route between a set of nodes [3,16,32,33,55]. Moreover, the agent must visit each node (city) only once, considering equivalent the initial final position (goal) of route. In this aspect, the TSP generalizations encompass various aspects of mobile robotics, such as restrictions of the vehicle [48], dynamic environments [65] and multiple vehicles [38,80].

An important research area for autonomous vehicle route planning considers fuel constraints [35,78]. In such cases, the challenge is to define a route to ensure that the vehicle carries out all the way without finishing the fuel. Following this same line, refueling problems seek to optimize the expenditure on the fuel purchase for road routes [27,60,71].

Vehicle refueling problems have been extensively investigated [25,27,36,42,43,56,57,62,69–71]. One of the lines of study is the fixed route vehicle refueling problem (FRVRP), where the goal is to select the refueling points on a fixed route. [27,43,73]. For example, [43] have presented a linear time greedy algorithm for the FRVRP. There are also applications of FRVRP to real problems. [60] have developed other example, where a fixed route refueling model for a case study of a Brazilian carrier; [73] have analyzed the influence of fuel weight, congestion, and acceleration on refueling policy optimization. Other works seek to analyze the refueling policy on variables routes [27,71]. In this sense, it is worth highlighting the applications based on TSP [67,71,82]. Suzuki [71] has presented a model that addresses the Traveling Salesman Problem With Time Windows and refueling. The goal is to define a route to minimize fuel consumption, respecting the time window for each customer [71]. Other applications of

✉ André L. C. Ottoni
andre.ottoni@ufrb.edu.br

Erivelton G. Nepomuceno
nepomuceno@ufsj.edu.br

Marcos S. de Oliveira
mso@ufsj.edu.br

Daniela C. R. de Oliveira
daniela@ufsj.edu.br

[1] Technologic and Exact Center, Federal University of Recôncavo da Bahia (UFRB), Cruz das Almas, Brazil

[2] Control and Modelling Group (GCOM), Department of Electrical Engineering, Federal University of São João del-Rei (UFSJ), São João del-Rei, Brazil

[3] Department of Mathematics and Statistics, Federal University of São João del-Rei (UFSJ), São João del-Rei, Brazil

TSP with refueling are unmanned aerial vehicles [67] and geosynchronous satellites [82]. It is important to point out that refueling problems are usually classified into four groups [27]: refueling with fixed route, refueling with variable route, TSP with uniform cost at each point and TSP with the fuel cost varying in the localities. In this sense, the last class can be applied to treat refueling in road networks in Brazil, where fuel price variations are found in each city according to data from the Brazilian National Petroleum Agency (ANP)[1].

In the literature, several methods have already been applied to solve refueling problems [35,56,67,72,77,82]. Levy et al. [35] have adopted heuristics Variable Neighborhood Descent and Variable Neighborhood Search (VNS) to vehicle routing problem with fuel constraints. The work [77] also adopts the VNS to optimize a fleet with alternative-fuel (gasoline or diesel vehicles). Zhang et al. [82] have used the Ant Colony Optimization to solve refueling multiple geosynchronous problems. The author of [72] discusses Simulated Annealing and Tabu Search methods for the pollution routing problem (minimize the fuel consumption or pollutants emission). Other papers presented news algorithms to optimize refueling problems [56,67]. Although Reinforcement Learning has shown to be a great tool to combinatorial optimization problems there is less attention to solve refueling problems.

Reinforcement learning (RL) is an artificial intelligence technique with relevant applications in robotics [8,15,28–30,37], path planning [20,39,47,59,75,76] and combinatorial optimization problems [4,7,13,14,21,44,53,54,64,79], such as the TSP [1,2,18,22,41,45,52,66,81]. In RL, an agent learns from rewards and penalties in interacting with an environment [68]. One of the main topics of investigation in RL is the estimation of learning parameters, like learning rate (α) and discount factor (γ), ϵ-$greedy$ and reinforcement function [6,17,23,24,40,54,63]. In fact, parameter definition can directly influence a good route learning [5,12,52,54]. Bal and Mahalik [5] have shown how to estimate the parameters α and γ by trial and error for a simulated navigation environment. In Ottoni et al. [52], the authors have presented a systematic approach for the RL parameter estimation using Response Surfaces Methodology (RSM). In [54], a complete factorial experiment and the Scott-Knott have been used to find the best combination of factors (ϵ-$greedy$ and reinforcement function) for the Sequential Ordering Problem. The paper [12], in turn, has proposed a method based on evolutionary computation to seek the best reinforcement function and Deep Learning network architecture for an autonomous navigation problem. Yet, no rigorous method for estimating the parameters for refueling problems has been found.

To overcome the lack of a parameter estimation framework for refueling problems, this work introduces a statistical methodology for tuning RL parameters employed on traveling salesman problem With refueling. More specifically, we have analyzed how the RL parameters and the refueling problems characteristics influence the learning of routes to optimize the fuel cost. We have proposed an RL structure to solve the traveling salesman problem with refueling (TSPWR), through a model (actions, states, reinforcements) and RL-TSPWR algorithm. Instances to solve uniform and non-uniform cost routes were worked out based on ANP data. The experiments involve simulations with two traditional RL algorithms: Q-learning [74] and SARSA [68]. In addition, RL parameter estimation is performed using statistical methods: RSM [51], Analysis of Variance (ANOVA) [49] and Tukey Test [49]. Best solutions have been found in 15 out of 16 analyzed numerical experiments.

The remainder of this paper is organized as follows. The second and the third sections present basic theoretical concepts of the RL and TSPWR, respectively. Then, the fourth section describes the proposed technique. The results are given in the fifth section and concluding remarks are delivered in the sixth section.

Reinforcement learning

Reinforcement learning (RL) is a machine-learning technique based on Markov decision processes (MDPs) [26,61,68,74]. MDPs are structured from finite sets of actions, states, reinforcement and a state transition model. The learner agent interacts with the environment in a sequence of steps in time (t): (i) the agent receives a representation of the environment (state); (ii) select and execute an action; (iii) receive the reinforcement signal; (iv) update the learning matrix; (v) observe the new state of the environment [68].

In RL, the goal is to learn a policy (π) that maximizes numerical reinforcement [68]. A policy defines the agent behavior, mapping states into actions. The ϵ-$greedy$ method is an example of the action selection policy adopted in RL [68]. In this method, the parameter ϵ ($0 < \epsilon < 1$) is defined and the policy $\pi(s)$ is applied according to the following equation [68]:

$$\pi(s) = \begin{cases} a^*, & \text{with probability} \quad 1 - \epsilon \\ a_a, & \text{with probability} \quad \epsilon, \end{cases} \tag{1}$$

where $\pi(s)$ is the decision policy for the current state s, a^* is the best estimated action for the state s at the current time and a_a is a random action selected with probability ϵ.

SARSA [68] and Q-learning [74] are common RL algorithms. These methods are based on temporal difference learning (TD), that is, updates do not need to refer to real-time intervals, but to successive decision-making steps. The SARSA (see Algorithm 1) is an RL on-policy TD Control

[1] http://anp.gov.br/preco/.

algorithm, which depends on the next action (a_{t+1}) defined by the policy $\pi(s)$ to update the learning matrix, according to the following equation:

$$Q_{t+1} = Q_t(s,a) + \alpha[r(s,a) + \gamma Q_t(s',a') - Q_t(s,a)], \tag{2}$$

where s is a state and a is an action at the current instant (t), respectively; s' is state and a' is action at the next instant ($t+1$); $Q_t(s,a)$ is the value at time t in the Q matrix for the pair state × action (s,a). Q_{t+1} is the updating of the learning matrix in $t+1$ by executing the action a in state s; $r(s,a)$ is the reinforcement by the execution of the pair (s,a); α is the learning rate; γ is the discount factor.

The parameters learning rate (α) and discount factor (γ) are adopted in several algorithms [68]. These parameters can be set between 0 and 1. The learning rate controls overlap speed of new information and the discount factor describes an agent preference between current and future rewards. If $\gamma \approx 1$, then the future rewards are highly significant. Otherwise, if $\gamma \ll 1$, the current rewards are more relevant at the instant t than the subsequent rewards (discounted) [61,68].

```
Set the parameters: $\alpha$, $\gamma$ and $\epsilon$
For each pair s,a to initialise the matrix Q(s,a)=0
Observe the state s
Select the action a using $\epsilon$-greedy method
repeat
    Take the action a
    Receive immediate reward  r(s, a)
    Observe the new state s'
    Select the new action a' using $\epsilon$-greedy method
    Update Q (s, a) with Eq. (2)
    s = s'
    a = a'
until the stopping criterion is satisfied;
```

Algorithm 1: SARSA.

On the other hand, Q-learning (see Algorithm 2) is an off-policy TD Control algorithm [61,68,74]. In that sense, it does not depend on the next action (a_{t+1}) to perform the update at the instant t, according to the following equation:

$$Q_{t+1} = Q_t(s,a) + \alpha\left[r(s,a) + \gamma \max_{a'} Q(s',a') - Q_t(s,a)\right], \tag{3}$$

where $\max_{a'} Q(s',a')$ is the utility of s', that is, the maximum value in the line of Q referring to the new state.

```
Set the parameters: $\alpha$, $\gamma$ and $\epsilon$
For each pair s,a the matrix Q(s,a)=0 should be
  initialised
Observe the state s
repeat
    Select the action a using $\epsilon$-greedy method
    Take the action a
    Receive immediate reward  r(s, a)
    Observe the new state s'
    Update Q (s, a) with Eq. (3)
    s = s'
until the stopping criterion is satisfied;
```

Algorithm 2: Q-learning.

Traveling salesman problem with refueling

The problem considered in this work is the path planning in a road network for autonomous vehicles. A mobile agent must travel through a set of cities and decide where to refuel to minimize the final route cost. For this, the Traveling Salesman Problem With refueling (TSPWR) is adopted in two forms: uniform and non-uniform cost [27]. In the first case, a vehicle must visit a set of locations and return to the starting city at the route end and fuel price does not vary between route stations. Second, in the problem with non-uniform cost, there are different selling cost for the fuel in the cities.

The following restrictions are considered: vehicle fuel tank capacity, minimum amount of fuel for refueling and guarantee of completing the entire route [60]. In addition, this work considers the possibility of using a tow truck in case fuel runs out between two locations, which requires an additional cost for that.

Problem formulation

A mathematical formulation for the proposed problem is based in [10,60,70] and contains two decision variables: $u_{i,j}$ and z_{ij}. The $u_{i,j}$ assumes 1 if the arc (i,j) makes up the solution and 0 otherwise. Also, z_{ij} is a decision variable that gets 1 only if the tow truck is used between the locations i and j. This formulation is presented in the following equations:

$$Min \sum_{i=1}^{N} \sum_{j=1}^{N} c_j l_j u_{ij} + g_{ij} z_{ij}, \tag{4}$$

subject to:

$$\sum_{i=1}^{N} u_{ij} = 1 \qquad j = 1, \ldots, N, \tag{5}$$

$$\sum_{j=1}^{N} u_{ij} = 1 \qquad i = 1, \ldots, N, \tag{6}$$

$$f_j + l_j \leq L_{\max} u_{ij} \qquad i, j = 1, \ldots, N, \tag{7}$$

$$l_j = (L_{\max} - f_j) w_{ij} \qquad i, j = 1, \ldots, N, \tag{8}$$

$$l_j \geq L_{\min} w_{ij} \qquad i, j = 1, \ldots, N, \tag{9}$$

$$u_{ij}, w_{ij}, z_{ij} \in \{0, 1\} \qquad i, j = 1, \ldots, N, \tag{10}$$

$$\{c_j, f_j, g_{ij}, l_j, L_{\max}, L_{\min}\} \geq 0 \quad i, j = 1, \ldots, N, \tag{11}$$

$$U = u_{ij} \in V \qquad i, j = 1, \ldots, N, \tag{12}$$

where N is a set of nodes. In addition, the refueling cost in the city j is c_j and l_j is the amount of fuel replenished in j. The tow truck cost on an arc (i, j) is represented by g_{ij}. Thus, Eq. (4) is the objective function, wherein the total route cost given by the sum of refueling and tow truck costs should be minimized. Equations (5) and (6) ensure that each location is visited only once. Furthermore, Eq. (7) ensures that the amount of fuel in the tank $(f_j + l_j)$ does not exceed maximum capacity $(L_{\max})$, where f_j is the reservoir level at the time of arrival in the city j. Equation (8) ensures that the vehicle completes the maximum tank level when refueling, where $w_{ij} = 1$ if refueling occurs at location j. Besides that, Eq. (9) restricts the minimum quantity for refueling. In addition, Eqs. (10) and (11) ensure that the variables u_{ij}, w_j, z_{ij} are binary and the other variables are non-negative, respectively. Finally, in Eq. (12), the set V represents any set of constraints that eliminate the formation of sub-routes.

Instances

In this paper, four instances are proposed: Bahia30D, Minas24D, Minas30D and Minas57D. Each instance involves a set of cities from two Brazilian states (Minas Gerais and Bahia). The data is composed by the Euclidean distances between localities, calculated from the coordinates (latitude and longitude). In addition, also the diesel average cost (D) in each city was defined from the ANP website data obtained in December 2018. Then, the cities are described in the following format "city (diesel average price in Reais (R$)—Brazilian currency)":

Bahia30D: Alagoinhas (3.307), Barreiras (3.654), Brumado (3.646), Caetite (3.688), Camaçari (3.358), Eunápolis (3.526), Feira de Santana (3.346), Guanambi (3.620), Ilhéus (3.761), Ipirá (3.304), Irecê (3.650), Itabuna (3.599), Itamaraju (3.520), Jacobina (3.592), Jaguaquara (3.336), Jequié (3.597), Juazeiro (3.668), Lauro de Freitas (3.270), Livramento de Nossa Senhora (3.721), Paulo Afonso (3.683), Poções (3.380), Porto Seguro (4.067), Salvador (3.399), Santo Antônio de Jesus (3.340), Senhor do Bonfim (3.481), Serrinha (3.443), Simões Filho (3.367), Teixeira de Freitas (3.545), Valen-ça (3.532) and Vitória da Conquista (3.291).

Minas24D: Araguari (3.321), B. Horizonte (3.471), Betim (3.408), Campo Belo (3.433), Contagem (3.393), Formiga (3.418), Governador Valadares (3.366), Guaxupé (3.446), Itabira (3.476), Ituiutaba (3.437), Juiz de Fora (3.307), Monte Carmelo (3.428), Montes Claros (3.458), Oliveira (3.361), Patos de Minas (3.526), Poços de Caldas (3.613), Pouso Alegre (3.453), Sete Lagoas (3.238), Teófilo Otoni (3.443), Três Corações (3.735), Uberaba (3.51), Uberlândia (3.476), Unaí (3.486) and Varginha (3.511).

Minas30D: Cities in Minas24D more Araxá (3.399), Barbacena (3.475), Divinópolis (3.507), Ipatinga (3.483), Lavras (3.774) and Passos (3.657).

Minas57D: Cities in Minas30D more Alfenas (3.624), Bom Despacho (3.249), Caratinga (3.429), Congonhas (3.557), C. Lafaiete (3.629), Coronel Fabriciano (3.668), Curvelo (3.288), Frutal (3.583), Itajubá (3.456), Itaúna (3.444), Janaúba (3.586), Januária (3.726), João Monlevade (3.421), João Pinheiro (3.533), Leopoldina (3.287), Manhuaçu (3.422), Muriaé (3.458), Nova Lima (3.724), Ouro Preto (3.72), Pará de Minas (3.526), Paracatu (3.656), Patrocínio (3.608), Sabará (3.532), São João del-Rei (3.712), São Sebatião do Paraíso (3.529), Timóteo (3.459) and Ubá (3.545).

Methodology

The methodology proposed in this paper consists of four steps. First, the RL model is structured in states, actions and reinforcement functions. After that, the algorithm for solving the TSPWR with Reinforcement Learning (RL-TSPWR) is proposed. The following steps present the experiments and methods for tuning RL parameters. Response Surface Models were used to optimize α and γ, wherein the best combinations of the reinforcement function and ϵ are obtained by means of ANOVA and Tukey test.

Reinforcement learning model

The model aims to enable the agent to learn how to path planning that minimizes refueling cost and distance. For this, the RL model defined for the TSPWR resolution consists of a set of states, actions and reinforcements. The wording adopted is based on previous studies that applied RL in TSP solution: [9,41,52]. The proposed structure is as follows:

- *States*: locations (nodes) that the agent (traveling salesman) must visit to perform the route. In this sense, the number of states varies according to the instance nodes.
- *Action*: intention to move to another location (state) of the problem. In addition, the refueling action is performed whenever the vehicle arrives at a location with less than 25% of tank level maximum capacity (0.25 × $L_{\max}$).
- *Reinforcements*: functions were defined to associate the cost with the movement between two localities, the refueling cost in each city and tow truck cost. Five different types of reinforcements have been proposed, according to the following equations:

$$R_1 = -(d_{ij} + c_j), \tag{13}$$

$$R_2 = -c_j, \tag{14}$$

$$R_3 = -d_{ij}, \tag{15}$$

$$R_4 = -(c_j + g_{ij}z_{ij}), \tag{16}$$

$$R_5 = -(d_{ij} + c_j + g_{ij}z_{ij}), \tag{17}$$

where d_{ij} is the distance between cities i and j; c_j is the refueling cost in the node j; tow truck cost on an arc (i, j) is represented by g_{ij} and z_{ij} is a decision variable that gets 1 only if the tow truck is used between the locations i and j. Thus, the higher the total cost of moving and refueling, the more negative the penalty for route formation is.

RL-TSPWR algorithm

This section presents the RL-TSPWR algorithm, which applies RL (Q-learning version) in the TSPWR solution (see Algorithm 3). The variables of the proposed algorithm are associated with the mathematical formulation of Eqs. (4)–(12).

In this paper, the simulated vehicle has small truck features for all experiments and TSPWR constants are maximum fuel tank capacity of 150 l ($L_{\max} = 150$); average diesel consumption of 7 km/l; the tow truck cost of using was fixed at R$ 200.00 ($g_{ij} = 200$), and the reference level (level-ref) is 25% of maximum capacity ($0.25 \times L_{\max} = 37.5$).

The RL-TSPWR starts by initializing RL parameters, the learning matrix, TSPWR variables/constants and the initial state (s_0) (lines 1–4). Then, the execution loops start (lines 5 and 6). Subsequently, the destination city is selected, and the action is performed (lines 7 and 8). After that, the new fuel tank level is calculated from the distance between cities (i and j) and the average consumption (km/l). In line 10, the calculation of the truck cost is started. If the tank level is less than zero, then the vehicle reached the destination city (j) without fuel. Then, it is necessary to perform the reset at the tank level, assign the truck cost value (g_{ij}) and the decision variable z_{ij} receives 1. Otherwise, the truck cost for the arc (i, j) is zero ($z_{ij} = 0$). In line 17, computation of the refueling cost is initialized. If fuel level at the destination node (level) is less than the reference level (level-ref) and city is not the initial, then the vehicle must be refueled. In this way, the litres amount (l_j) and the cost of refueling ($c_j l_j$) are calculated (lines 21 and 22). In addition, the tank level is updated with the maximum vehicle level ($L_{\max}$). If the vehicle does not need to refuel, this cost is zero ($c_j l_j = 0$). Then, the total cost on the route is updated, based on the sum of the truck cost and refueling cost (line 27). The distance traveled on the route is also updated (line 28). Subsequently, the reinforcement is calculated, such as Eq. (13). Finally, the RL operations are carried out: new state notice, update Q matrix and current state (lines 30 and 31).

```
1  Set the parameters: $\alpha$, $\gamma$ and $\epsilon$ ;
2  Initialize the matrix **Q(s, a)** $= 0$ ;
3  Initialize TSPWR variables and constants ;
4  Observe the state $s_0$: initial city;
5  repeat
6      repeat
7          Select the action $a$ (destination city) using $\epsilon$-greedy method ;
8          Take the action $a$;
9          Calculation of the fuel level in the tank: level;
10         % Calculation of the truck cost:;
11         if level $< 0$ then
12             Reset the tank level: level $= 0$;
13             Calculation of the truck cost: $g_{ij}$;
14             $z_{ij} = 1$ ;
15         else
16             The truck cost is zero;
17             $z_{ij} = 0$;
18         end
19         % Calculation of the refueling cost:;
20         if (level < level-ref) and ( $a$ =! $s_0$) then
21             Calculation litres amount for refueling: $l_j$;
22             Calculation of the refueling cost: $c_j l_j$;
23             Maximum tank level: level $= L_{max}$ ;
24         else
25             The refueling cost is zero: $c_j l_j = 0$;
26         end
27         Updates the total cost (route): Eq. (4) ;
28         Updates the distance travelled on the route;
29         Receive immediate reward: Eqs. (13) to (17);
30         Observe the new state $s'$ (new city);
31         Update Q(s,a): Eq. (3);
32         $s = s'$;
33     until complete the route;
34 until the stopping criterion is satisfied;
```

Algorithm 3: RL-TSPWR Algorithm. This is an application of the Q-learning algorithm for solving the TSPWR.

Algorithm 3 executes its instructions from two repeat loops. The first repetition structure is controlled by the number of episodes (stopping criterion). On the other hand, the second loop is dependent on the number of locations in the instance (iterations for the formation of one route). Thus, the complexity of the RL-TSPWR algorithm can be represented by the number of learning iterations (nr) to provide a solution (Eq. 18):

$$nr = E \times N, \tag{18}$$

where E is the number of episodes and N is the number of locations for the instance. Table 1 exemplifies the RL-TSPWR complexity (using 10,000 episodes):

Table 1 shows the efficiency of the proposed structure. For example, the Minas57D ($N = 57$) instance has 7.110×10^{74} possible solutions. In contrast, the algorithm presents a solution after a sequence of 570,000 learning iterations and only 10,000 routes explored. It is worth mentioning that the number of episodes is also a parameter that can be investigated.

Table 1 RL-TSPWR algorithm complexity. Difference (diff.) between the number of total possible TSPWR routes ($(N-1)!$) and the number of iterations explored by the RL-TSPWR in 10,000 episodes ($10000N$)

N	$(N-1)!$	$10000N$	$Diff.$
5	120	50,000	49,880
10	362,880	100,000	– 262,880
24	2.585×10^{22}	240,000	-2.585×10^{22}
30	8.842×10^{30}	300,000	-8.842×10^{30}
57	7.110×10^{74}	570,000	-7.110×10^{74}
100	9.333×10^{157}	1,000,000	-9.333×10^{157}

In this work, E assumed three values (1000; 10,000; 20,000) according to the experiment stage.

A version of the RL-TSPWR adopting SARSA algorithm is also proposed. For this, some small changes were made to the RL-TSPWR from Algorithm 1, such as in line 31, which Eq. (2) is used. In the experiments and results, the RL-TSPWR is discussed according to the version used: Q-learning or SARSA.

Tuning of RL parameters: α and γ

The purpose of this section is to present the methodology for tuning the RL parameters (α and γ) for the TSPWR. For this, experiments with different combinations of these parameters are proposed. In addition, mathematical modeling is adopted via response surface methodology to estimate α and γ. In this stage, the experimental methodology was based on recent works: [52,54].

RL parameters experiments: α and γ

Simulations were performed using the Matlab and were comprised by 16 groups of experiments (2 algorithms × 4 instances × 2 types of problems):

- *Instances*: Bahia30D, Minas24D, Minas30D and Minas57D.
- *Algorithms*: Q-learning and SARSA.
- *Types of problems*: non-uniform and uniform.

In addition, simulations were carried out for each group of experiments involving 64 combinations of the learning rate (α) and discount factor (γ). The values of these parameters being defined as:

- α: [0.01; 0.15; 0.30; 0.45; 0.60; 0.75; 0.90; 0.99].
- γ: [0.01; 0.15; 0.30; 0.45; 0.60; 0.75; 0.90; 0.99].

Each combination of parameters was simulated in 3 runs (repetitions) with 1000 episodes. A run is an independent repetition, that is, the learning is accumulated over the thousand episodes and always reset when starting a run. The episode performance measures are the total refueling cost and distance in the route. In addition, the ϵ-$greedy$ parameter was set to $\epsilon = 0.01$ and the reinforcement function adopted was R_1 (Eq. 13).

RSM

The response surface methodology (RSM) involves a set of statistical techniques for analyzing optimization problems. The structure and RSM model of second order is presented [51] as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + e, \quad (19)$$

where y is the response variable, x_1 and x_2 are the independent variables, β_n are the coefficients and the effect of the error (residual) is represented by e.

Ottoni et al. [52] have presented the mathematical modeling using RSM for the estimation of α and γ parameters. The structure proposed by [52] is given in the following equation:

$$\hat{y} = \beta_0 + \beta_1 \alpha + \beta_2 \gamma + \beta_3 \alpha^2 + \beta_4 \gamma^2 + \beta_5 \alpha\gamma. \quad (20)$$

where α and γ are the independent variables of the model and $\hat{y}$ is the predicted response.

In this work, 16 RSM models were adjusted using the *software* R [34,58], according to Table 2. These models aim to estimate α and γ to minimize the total cost on a route. Data referring to the lowest cost on the route (refueling + tow truck) have been used with a combination of α and γ.

Tuning of RL parameters: reinforcement function and ϵ

The second stage of experiments aims to analyze the influence of the reinforcement functions and ϵ parameter in TSPWR learning. For that, simulations with different combinations of these parameters are proposed. ANOVA and Tukey test were adopted to identify the best combinations of factors for the refueling problem. Besides that, the parameters (α and γ) estimated via RSM were used in the experiments in this section. The experimental and analysis methodology have been based in [54].

RL parameters experiments: reinforcement function and ϵ

In this step, the objective was to conduct experiments with two learning specifications: reinforcement function and ϵ parameter (ϵ-$greedy$ policy) as follows:

Table 2 Adjusted RSM models

Model	Instance	Algorithm	Problem
1		Q-learning	Non-uniform
2	Bahia30D	Q-learning	Uniform
3		SARSA	Non-uniform
4		SARSA	Uniform
5		Q-learning	Non-uniform
6	Minas24D	Q-learning	Uniform
7		SARSA	Non-uniform
8		SARSA	Uniform
9		Q-learning	Non-uniform
10	Minas30D	Q-learning	Uniform
11		SARSA	Non-uniform
12		SARSA	Uniform
13		Q-learning	Non-uniform
14	Minas57D	Q-learning	Uniform
15		SARSA	Non-uniform
16		SARSA	Uniform

- Reinforcement functions: R_1 [Eq. (13)], R_2 [Eq. (14)], R_3 [Eq. (15)], R_4 [Eq. (16)], and R_5 [Eq. (17)].
- Parameter ϵ: [0.01; 0.05; 0.10].

Simulations were comprised by 240 groups of experiments: 2 (algorithms) × 4 (instances) × 2 (problem types) × 5 (reinforcement functions) × 3 (ϵ values). In this respect, a total of 15 parameters combinations (R and ϵ) have been conducted for each model (Table 2). Each experiment was simulated in 10 runs (repetitions) with 10,000 episodes. The episode performance measures are the total refueling cost in the route.

The results of these experiments were used as data for the modeling presented in the next section.

Factorial design

In this step, a factorial design was developed to estimate the factor effects ($R \times \epsilon$) in the TSPWR simulations. The factors analyzed are the reinforcement function (five levels) and the parameter ϵ (three levels) [49,54]:

$$y_{jkl} = \mu + \eta_j + \theta_k + (\eta\theta)_{jk} + \xi_{jkl}, \tag{21}$$

where μ is the overall mean effect, η_j is the effect of the j_{th} level of the reinforcement functions ($j = 1, 2, 3, 4, 5$), θ_k is the effect of the k_{th} of ϵ-greedy politics ($k = 1, 2, 3$), $(\eta\theta)_{jk}$ is the effect of interaction between η_j and θ_k, and ξ_{jkl} is a random error component ($l = 1$ to 10).

Analysis of variance test was conducted to check if there is a difference between the treatment means. The level of significance adopted was 5%. When ANOVA indicates that there is a difference between the levels of the model, Tukey test of multiple comparisons [49] has been applied.

Comparison with other literature parameters

After developing the parameter tuning for the TSPWR, a new stage of experiments was performed with the estimated values. In addition, simulations were also carried out with parameters (α and γ) defined in other works that addressed of combinatorial optimization problems with RL resolution: $\alpha = 0.1$ and $\gamma = 0.3$ [9,18], $\alpha = 0.8$ and $\gamma = 0.9$ [66], $\alpha = 0.1$ and $\gamma = 0.9$ [45] and $\alpha = 0.9$ and $\gamma = 1$ [41].

The objective was to evaluate the performance of parameter adjustment for the TSPWR, in comparison with the use of values adopted in the literature in RL simulations for the classic TSP (or similar). These combinations of parameters were simulated in three repetitions with 20,000 episodes for each group of experiments.

Results

Tuning of RL parameters results: α and γ

The results adjusted for setting the RSM models are described below. The analysis is based on the work of [52].

Adjusted models

Measures of the adjusted models analysis should present normality of the residues, coefficient of multiple determination (R^2), adjusted coefficient of multiple determination (R_a^2) and significance of the coefficients.

The first test determines if the model residues follow a normal distribution. Adopting the Kolmogorov–Smirnov (KS) [46] test, it was observed that for the 16 models, the hypothesis of residual normality ($p_{KS} > 0.05$) was accepted, according to Table 3. Then, the values of R^2 and R_a^2 were analyzed. The more these coefficients are approaching 1, it evidenced a good fit of the model to the sample. Table 3 also shows the calculated values for R^2 and R_a^2.

Table 4 shows the adjusted coefficients for each model. In this sense, the test of significance of the individual coefficients, it points out that the coefficients are highly significant in all models ($p < 0.001$).

Stationary points

The analysis of stationary points allows us to verify the values that optimize the predicted response in the adjusted RSM models. In this respect, the estimation of the parameters α and γ refers to a second optimization problem to minimize

Table 3 Adjustment measures: p values of the KS test (p_{KS}), R^2 and R_a^2

Model	p_{KS}	R^2	R_a^2
1	0.7992	0.7636	0.7573
2	0.7966	0.7806	0.7747
3	0.6472	0.7929	0.7873
4	0.8622	0.7867	0.7809
5	0.3864	0.7817	0.7817
6	0.7555	0.8145	0.8095
7	0.8889	0.8373	0.8329
8	0.7294	0.8062	0.8010
9	0.5763	0.8321	0.8276
10	0.3175	0.8506	0.8466
11	0.5569	0.8584	0.8546
12	0.2780	0.8352	0.8308
13	0.9460	0.8553	0.8515
14	0.3722	0.8618	0.8581
15	0.8394	0.8557	0.8518
16	0.7492	0.8738	0.8704

Table 5 Stationary points

Model	α	γ
1	0.6980	0.0429
2	0.7131	0.0244
3	0.7321	0.0000
4	0.7069	0.0000
5	0.6361	0.2396
6	0.6361	0.2425
7	0.6265	0.2409
8	0.6197	0.2296
9	0.6627	0.1415
10	0.6574	0.1613
11	0.6474	0.1545
12	0.6605	0.1242
13	0.6951	0.1160
14	0.6870	0.1552
15	0.6973	0.1141
16	0.6967	0.1210

the predicted response $\hat{y}$ (route cost) in each model adjusted. The formulation of this problem is given by Eq. (22) [52]:

$$\begin{aligned} &\min_{\alpha,\gamma} && \hat{y} \\ &\text{subject to} && 0 \leq \alpha \leq 1, \quad 0 \leq \gamma \leq 1. \end{aligned} \tag{22}$$

Table 5 shows the stationary points obtained using the R software [34,58].

Tuning of RL parameters results: reinforcement function and ϵ

In this section, we present the experiments results for tuning the reinforcement function and the parameter ϵ. Initially, some graphics are shown for interaction between the factors. The interaction plots demonstrate the influence of these parameters (R and ϵ) on the TSPWR optimization process. After that, the results of ANOVA and Tukey test for full factorial experiment are presented.

Table 4 RSM adjusted coefficients

Model	β_0	β_1	β_2	β_3	β_4	β_5
1	2710.10	– 2319.30	– 438.10	1646.70	1195.20	480.70
2	2798.20	– 2726.90	– 516.90	1901.00	1173.70	644.40
3	2821.98	– 2839.97	– 483.02	1997.31	1058.81	924.06
4	2709.63	– 2701.70	– 264.93	1951.85	1024.47	637.31
5	2189.96	– 1785.36	– 978.54	1321.80	1466.82	433.15
6	2143.82	– 1596.51	– 937.44	1193.88	1512.57	320.33
7	2229.19	– 1918.10	– 1123.14	1430.11	1650.32	523.74
8	2175.42	– 1796.58	– 896.94	1377.56	1428.38	389.12
9	2545.87	– 2795.74	– 1011.82	2029.03	1812.24	752.80
10	2559.03	– 2834.34	– 1100.92	2068.00	1955.39	715.21
11	2555.97	-2912.14	– 1093.35	2157.97	1939.76	762.75
12	2516.03	– 2674.95	– 963.81	1950.91	1790.69	785.88
13	4869.70	– 6280.20	– 2276.40	4355.80	3995.60	1941.60
14	4903.00	– 6148.30	– 2756.00	4251.90	4501.30	1978.10
15	4900.40	– 6262.40	– 2208.90	4336.20	3916.80	1886.30
16	4888.40	– 6231.90	– 2455.10	4293.60	4219.60	2058.50

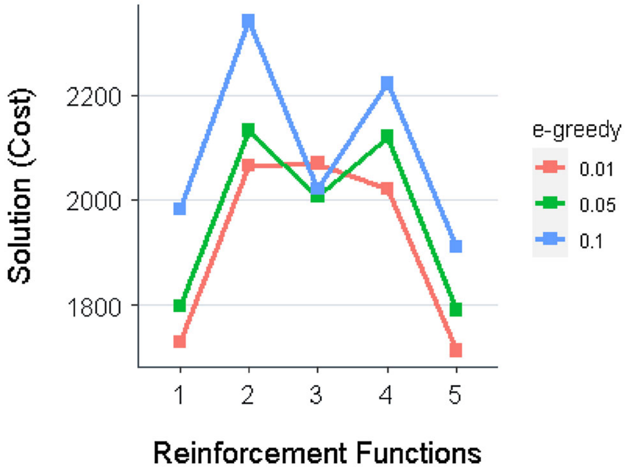

Fig. 1 Interaction plots between factors (reinforcement function $\times\epsilon$-greedy) for model 1 (Bahia30D/Non-Uniform/Q-learning)

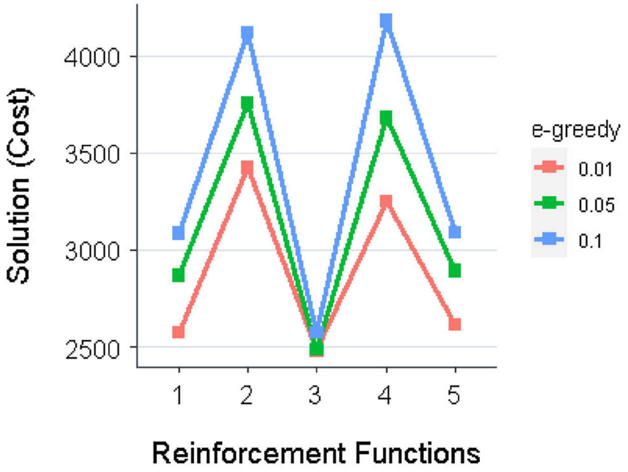

Fig. 2 Interaction plots between factors (reinforcement function $\times\epsilon$-greedy) for model 13 (Minas57D/Non-Uniform/Q-learning)

Interaction plots analysis

Interaction plots are important tools for analyzing the factors influence on the response variable. In this work, these graphs were approached in a preliminary analysis of the factorial design results to visualize effects of the ϵ-$greedy$ policy and the reinforcement function in the TSPWR solution.

To illustrate the graphical analysis, Figs. 1 and 2 present interaction plots for models 1 and 13, respectively. It is possible to observe that the combinations of $R_1 \times 0.01$ and $R_5 \times 0.01$ tend to minimize the response to the situation of Bahia30D/Non-Uniform/Q-learning (Model 1). On the other hand, in Fig. 2, referring to Minas57D/Non-Uniform/Q-learning, the best results are to adopt the reward function $R_3 \times 0.01$ or $R_3 \times \epsilon = 0.01$. In this respect, the simple change of instances (Bahia30D to Minas57D) directly influenced the combination performance ($R \times \epsilon$) for the TSPWR. Thus, the present analysis reinforces the need to adjust the ϵ-$greedy$ policy and reinforcement function according to the simulated data.

Factorial design results

Analysis of adjusted models of full factorial experiments was carried out in three phases: (i) residue normality analysis, (ii) analysis of variance and (iii) multiple comparison test. Adopting the KS test [46], the assumption of residue normality was confirmed for all models ($p_{\text{KS}} > 0.05$). ANOVA test was applied to check if there is a difference between the configuration performance ($R \times \epsilon$) in the TSPWR optimization. The results of analysis of variance showed that for the 16 models, the factors interaction is highly significant ($p < 0.001$). That is, there is a statistical difference in RL performance for TSPWR resolution, according to the reinforcement function and the parameter ϵ selected.

In this sense, Tukey multiple comparison test was then performed to identify the best combinations ($R \times \epsilon$) by factorial design model. Table 6 presents the results of the Tukey test and the residues normality tests (p_{KS}).

In Table 6, one can identify settings for each model ($R \times \epsilon$). which achieved the best results (“Tukey Test” column). Moreover, as in all situations Tukey test indicated more than one combination, a tiebreaker criterion was used: the lowest mean solution (cost) by combination. Thus, Table 6 also presents the best configuration for each column (“Best” column) and the respective mean solution.

For example, take the model 1 (Bahia30D, Q-learning, and Non-Uniform). In this case, the Tukey test showed that there are four combinations that showed good performances: $R_1 \times 0.01$, $R_1 \times 0.05$, $R_5 \times 0.01$ and $R_5 \times 0.05$. Furthermore, the configuration $R_5 \times 0.01$ times showed the lowest mean of the solution between those indicated by the multiple comparison test. On the other hand, observing the Model 16 (Minas57D, SARSA and Uniform), another 3 combinations ($R_3 \times 0.01$; $R_3 \times 0.05$; $R_3 \times 0.10$) were indicated by Tukey test. Thus, Table 6 reveals that for a simulated situation, it may be interesting to adopt different combinations of the reinforcement function and parameter ϵ to TSPWR optimization.

Further exploring the “Tukey Test” column in Table 6, it is important to highlight that $R_5 \times 0.01$ is the combination that most appeared among the settings indicated (on 15 of the 16 models). In this sense, it shows the relevance of the reinforcement functions that have the distance between the nodes (d_{ij}) as a term of the Equation: R_1 [Eq. (13)], R_3 [Eq. (15)], and R_5 [Eq. (17)].

Table 6 presents $R_5 \times 0.01$ as the most suitable combination (4 times). In this case, the second configurations were: $R_3 \times 0.01$, $R_3 \times 0.05$ and $R_5 \times 0.05$ (with three indications for each). That is, for none of the models, the reinforcement

Table 6 Tuning RL parameters results (reinforcement function and ϵ parameter): KS test (normality of residues), Tukey test (multiple comparison), the best configuration ($R \times \epsilon$) and solution for model

Instance	Alg	Pr	p_{ks}	Tukey test	Best	Solution
Bahia30D	Q	N	0.05	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_5\epsilon_1$; $R_5\epsilon_2$	$R_5 \times 0.01$	1712.50
	Q	U	0.63	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_5\epsilon_1$; $R_5\epsilon_2$	$R_1 \times 0.01$	1713.17
	S	N	0.47	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$; $R_5\epsilon_2$	$R_5 \times 0.01$	1670.43
	S	U	0.11	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$; $R_5\epsilon_2$	$R_1 \times 0.01$	1694.78
Minas24D	Q	N	0.18	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_1\epsilon_3$; $R_3\epsilon_2$; $R_5\epsilon_1$; $R_5\epsilon_2$; $R_5\epsilon_3$	$R_5 \times 0.05$	1596.99
	Q	U	0.06	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_1\epsilon_3$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_4\epsilon_1$; $R_5\epsilon_1$; $R_5\epsilon_3$	$R_5 \times 0.05$	1612.21
	S	N	0.08	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_1\epsilon_3$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_4\epsilon_1$; $R_5\epsilon_1$; $R_5\epsilon_2$	$R_5 \times 0.01$	1604.43
	S	U	0.08	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_1\epsilon_3$; $R_2\epsilon_1$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_4\epsilon_1$; $R_5\epsilon_1$; $R_5\epsilon_2$; $R_5\epsilon_3$	$R_5 \times 0.05$	1608.35
Minas30D	Q	N	0.05	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$; $R_5\epsilon_2$	$R_5 \times 0.01$	1607.59
	Q	U	0.05	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_1\epsilon_3$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$; $R_5\epsilon_2$; $R_5\epsilon_3$	$R_1 \times 0.01$	1624.80
	S	N	0.06	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$; $R_5\epsilon_2$	$R_3 \times 0.05$	1596.60
	S	U	0.06	$R_1\epsilon_1$; $R_1\epsilon_2$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$; $R_5\epsilon_2$	$R_3 \times 0.05$	1614.01
Minas57D	Q	N	0.05	$R_1\epsilon_1$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$	$R_3 \times 0.01$	2481.11
	Q	U	0.05	$R_1\epsilon_1$; $R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$	$R_3 \times 0.05$	2553.16
	S	N	0.12	$R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$; $R_5\epsilon_1$	$R_3 \times 0.01$	2407.53
	S	U	0.05	$R_3\epsilon_1$; $R_3\epsilon_2$; $R_3\epsilon_3$	$R_3 \times 0.01$	2427.77

Alg algorithm, *Pr* problem, *Q* Q-learning, *S* SARSA, *U* uniform, *N* non-uniform, $\epsilon_1 = 0.01$, $\epsilon_2 = 0.05$, $\epsilon_3 = 0.10$

functions R_2 or R_4 or the parameter $\epsilon = 0.10$ are presented as parameters as shown best for the experiments.

It is also important to highlight the differences in reinforcement functions performance according to the instance adopted. For example, for the Minas24D instance, in all models the best configuration ("Best" column in the Table 6) contains the term R_5. However, this is not repeated for the Minas57D instance, where the indicated reinforcement function was R_3. Thus, one hypothesis is that the difference between the number of instance nodes directly influenced the reinforcement function performance.

RL-TSPWR parameters

In this section, the final estimated parameters for the TSPWR instances are presented. Table 7 shows the best parameters (lowest cost in Reais—Brazilian currency) per each of the 16 situations (4 instances × 2 problems × 2 algorithms).

From Table 7, when analyzing the reinforcement functions, it is noticed that R_5 was indicated 7 times. Moreover, it appears that: (i) for all instances, the reinforcement function (R_1—Eq. (13), R_3—Eq. (15) or R_5— Eq. (17)) has the distance between the nodes term (d_{ij}); (ii) for 10 cases in three instances (Bahia30D, Minas24D and Minas30D), the reinforcement function (R_1—Eq. (13) or R_5—Eq. (17)) has the refueling cost term (c_j).

When observing the ϵ-*greedy* policy, the value of $\epsilon = 0.01$ achieved the best results in most cases (10 times). On the other hand, for the learning rate and discount factor parameters, it is possible to define tuning ranges in Table 7: α = [0.6197, 0.7321] and γ = [0.000, 0.2425].

Comparison with other works

Comparison with literature parameters

In this section, results of the parameters adjusted by this paper (see Table 7) for the TSPWR are presented in comparison with the adoption of fixed parameters (α and γ) in the literature [18,41,45,66], which are referred to studies that applied the RL in simulations of the classic TSP (or similar). Table 8 shows the best solutions found (cost in Reais—Brazilian currency) in this phase.

The proposed technique achieved best results in 15 out of 16 groups of experiments according to Table 8. This shows the capacity of the proposed methodology to tuning of parameters suitable for the TSPWR. In addition, it reveals the importance of performing parameter adjustment according to the conditions of the simulation (instance, algorithm and problem).

Comparison with literature approaches

In this section, five features of the proposed technique were compared with other works in the literature: problem, refueling problem characteristics, optimization approach, tuning RL parameters and methods. Table 8 presents this comparative study with the following works in the literature: I [27],

Table 7 Reinforcement Learning parameters estimated for TSPWR instances

Instance	Algorithm	Problem	R	ϵ	α	γ
Bahia30D	Q-learning	Non-uniform	R_5	0.01	0.6980	0.0429
	Q-learning	Uniform	R_1	0.01	0.7131	0.0244
	SARSA	Non-uniform	R_5	0.01	0.7321	0.0000
	SARSA	Uniform	R_1	0.01	0.7069	0.0000
Minas24D	Q-learning	Non-uniform	R_5	0.05	0.6361	0.2396
	Q-learning	Uniform	R_5	0.05	0.6361	0.2425
	SARSA	Non-uniform	R_5	0.01	0.6265	0.2409
	SARSA	Uniform	R_5	0.05	0.6197	0.2296
Minas30D	Q-learning	Non-uniform	R_5	0.01	0.6627	0.1415
	Q-learning	Uniform	R_1	0.01	0.6574	0.1613
	SARSA	Non-uniform	R_3	0.05	0.6474	0.1545
	SARSA	Uniform	R_3	0.05	0.6605	0.1242
Minas57D	Q-learning	Non-uniform	R_3	0.01	0.6951	0.1160
	Q-learning	Uniform	R_3	0.05	0.6870	0.1552
	SARSA	Non-uniform	R_3	0.01	0.6973	0.1141
	SARSA	Uniform	R_3	0.01	0.6967	0.1210

Table 8 Best solutions found (cost in Reais—Brazilian currency) adopting the values of the estimated values and parameters defined (α and γ) in other works by groups of experiments

Instance	Algorithm	Problem	Proposed	D95	S01	Z09	L10
Bahia30D	Q-learning	Non-uniform	**1667.00**	1834.28	2169.86	1911.74	3400.01
	Q-learning	Uniform	**1678.78**	1771.72	2218.13	1804.92	3224.46
	SARSA	Non-uniform	**1652.22**	1808.75	2510.49	1798.65	3352.84
	SARSA	Uniform	**1635.27**	1796.43	2433.02	1783.59	3355.97
Minas24D	Q-learning	Non-uniform	**1380.63**	1601.03	1753.01	1592.60	2508.32
	Q-learning	Uniform	1616.62	**1402.84**	1729.79	1643.03	2144.97
	SARSA	Non-uniform	**1571.65**	1597.52	1820.28	1594.55	2427.38
	SARSA	Uniform	**1544.64**	1613.09	2088.56	1624.59	2619.41
Minas30D	Q-learning	Non-uniform	**1583.77**	1631.55	2109.50	1786.26	2973.52
	Q-learning	Uniform	**1614.10**	1615.19	2163.29	1779.96	3182.78
	SARSA	Non-uniform	**1603.50**	**1603.50**	2567.22	1816.16	3137.02
	SARSA	Uniform	**1605.54**	1701.47	2496.14	1828.68	3123.13
Minas57D	Q-learning	Non-uniform	**2444.22**	2892.93	3938.13	2911.35	5402.81
	Q-learning	Uniform	**2458.35**	2836.67	4168.51	2963.26	5495.39
	SARSA	Non-uniform	**2382.55**	2851.24	4271.34	2918.99	5778.20
	SARSA	Uniform	**2396.86**	2732.95	4178.62	2991.03	5967.12

Solutions with parameters (α and γ) described in D95: [18], S01: [66], Z09: [45], L10: [41]. Proposed: Solutions with parameters defined in the Table 7. Values in boldface indicate the best result found for each experiment

II [43], III [71], IV [60], V [18], VI [2], VII [52] and VIII [54] Table 9.

The first important aspect of this work is the TSP approach in conjunction with the refueling problem. Generally, the TSP is applied to minimize the distance on the route, as in [2,18,52]. However, there is less attention in the literature for TSP with refueling [27,71].

Another relevant point of this proposal is application in variable routes. In the literature, when specifically observed the refueling problems, in many works only a fixed route is adopted, as in [43,60]. In fact, applying the refueling problem on variable routes is much more complex than on fixed routes [27]. It is also worth noting that, only the work of [60] also considered data from Brazilian road networks in the simulations. In this regard, it is worth mentioning that the developed instances (Bahia30D, Minas24D, Minas30D and Minas57) will be made available in the public database format: TSPWR-Library. In addition, the TSPWR proposed modeling (Sect. 3) innovates when considering the possibil-

Table 9 Comparison of this proposal with different works in the literature: I [27], II [43], III [71], IV [60], V [18], VI [2], VII [52] and VIII [54]

		Proposed	I [27]	II [43]	III [71]	IV [60]	V [18]	VI [2]	VII [52]	VIII [54]
Problem	TSP or variations	✓	✓	–	✓	–	✓	✓	✓	✓
	refueling	✓	✓	✓	✓	✓	–	–	–	–
Refueling problem	Fixed route	–	✓	✓	–	✓	–	–	–	–
	Variable route	✓	✓	–	✓	–	–	–	–	–
	Brazilian data	✓	–	–	–	✓	–	–	–	–
	TSPWR-library	✓	–	–	–	–	–	–	–	–
	Truck cost	✓	–	–	–	–	–	–	–	–
Optimization approach	Reinforcement learning	✓	–	–	–	–	✓	✓	✓	✓
	Other methods	–	✓	✓	✓	✓	–	–	–	–
Tuning RL parameters	Reinforcement function	✓	–	–	–	–	✓	✓	–	✓
	ϵ	✓	–	–	–	–	–	✓	–	✓
	α	✓	–	–	–	–	–	✓	✓	–
	γ	✓	–	–	–	–	✓	–	✓	–
Tuning methods	RSM	✓	–	–	–	–	–	–	✓	–
	ANOVA	✓	–	–	–	–	✓	–	–	✓
	Scott-Knott	–	–	–	–	–	–	–	–	✓
	Tukey	✓	–	–	–	–	–	–	–	–

ity of using a tow truck if the fuel runs out between two locations.

The proposed application of the RL for TSPWR is another important aspect of this paper. For this, the RL model was structured in states, actions and reward functions, considering the TSPWR characteristics. In addition, the algorithm (RL-TSPWR) for the application of RL in TSPWR was proposed. In the literature, studies that addressed the refueling problem used other methods, such as: VNS [77], Ant Colony Optimization [82] and Tabu Search [72].

We have avoided to compare RL techniques with other meta-heuristics in TSPWR resolution since RL methods have been carefully adjusted for application in the proposed refueling instances. Other meta-heuristics from literature have not been so far made the same adjustments. For example, to simulate a local search algorithm, such as VNS [77], it would be necessary to carry out a best initial solution study and which neighborhood structures would be adequate to generate good results for the problem in question. Also, implementation of Genetic Algorithms would require definition of the evolutionary parameters (selection, reproduction and mutation) suitable for application in the proposed TSPWR instances.

To exemplify, simulations were carried out using the VNS meta-heuristic to solve TSPWR instances. The initial solution was defined as an ordered sequence of cities. Already the neighborhood structure was based on random changes in the visit order of the nodes. In this regard, the VNS meta-heuristic achieved worse performances in the four instances: Bahia30D (4424.2), Minas24D (2972.4), Minas30D (3388.8) and Minas57D (8470.0). However, it is emphasized that the VNS is a local search algorithm that would probably perform better with tuning of initial solution. In this respect, this is an important advantage of RL methods, as it is not necessary to provide an initial solution.

Finally, we highlight the use of statistical methods (RSM, ANOVA and Tukey Test) in the tuning RL parameters process. In comparison with other works [2,18,52,54], only this proposal made the 4 parameters adjustment: reinforcement function, ϵ, α and γ.

Contributions of this paper

Based on the comparison with other literature works, the main contributions of this paper are highlighted:

1. Reinforcement Learning Approach to refueling problems solution.
2. Proposal of the RL-TSPWR Algorithm.
3. Statistical methodology for tuning of four RL parameters (reinforcement function, ϵ, α and γ) uniting concepts presented in [52] and [54].
4. New mathematical formulation for refueling problems using tow truck cost, variable routes and non-uniform cost.
5. Development of instances (TSPWR-Library) with fuel cost data for Brazilian cities.

Conclusion

This paper has applied Reinforcement Learning to the Traveling Salesman Problem with refueling. The outline of the contributions of this paper relative to the recent literature in the field can be summarized as: (i) proposal for TSPWR formulation problem; (ii) algorithm for applying the RL to the TSPWR resolution; (iii) development of instances based on real data from the ANP; (iv) experiments realization under uniform and non-uniform cost conditions; (v) tuning of RL parameters applied to TSPWR using the statistical methods.

Estimated parameters with statistical methods achieved the best solution in 15 out of 16 experimental groups. These results are valid for the two algorithms (Q-learning and SARSA) and for simulations with uniform and non-uniform fuel prices in each location. In addition, using ANOVA and Tukey test it was possible to find the best combination of reinforcement function and ϵ-$greedy$ policy for each instance. It is worth mentioning that the reinforcement functions obtained different performance according to the data analyzed. Nevertheless, in all cases adjusted reinforcement function has the distance between nodes (d_{ij}) term. By analyzing the $\epsilon - greedy$ policy, it is clear that the value of $\epsilon = 0.01$ reached the best solutions in most cases.

In future works, experiments with more instances and vehicle types are expected. New instances based on the TSPLIB library should be investigated. In addition, it is expected to analyze other factors, such as fuel type and vehicle model. Moreover, simulations with other meta-heuristics in the TSPWR instances should be investigated. In this aspect, computational complexity of the methods should be analyzed, and the convergence issue should also be discussed.

Funding The authors are grateful to CAPES, CNPq/INERGE, FAPEMIG, UFSJ and UFRB

Declarations

Conflict of interest The authors listed in this article declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Alipour MM, Razavi SN (2015) A new multiagent reinforcement learning algorithm to solve the symmetric traveling salesman problem. Multiagent Grid Syst 11(2):107–119
2. Alipour MM, Razavi SN, Derakhshi MRF, Balafar MA (2018) A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem. Neural Comput Appl 30(9):2935–2951
3. Applegate D, Bixby R, Chvátal V, Cook W (2011) The traveling salesman problem: a computational study. Princeton University Press, Princeton
4. Arin A, Rabadi G (2017) Integrating estimation of distribution algorithms versus q-learning into meta-raps for solving the 0–1 multidimensional knapsack problem. Comp Ind Eng 112:706–720
5. Bal SJ, Mahalik NP (2014) A simulation study on reinforcement learning for navigation application. Artif Intell Appl 1(2):43–53
6. Barsce JC, Palombarini JA, Martínez EC (2017) Towards autonomous reinforcement learning: automatic setting of hyper-parameters using bayesian optimization. In: 2017 XLIII Latin American Computer Conference (CLEI), pp 1–9
7. Bello I, Pham H, Le Q, Norouzi M, Bengio S (2019) Neural combinatorial optimization with reinforcement learning. In: 5th International Conference on Learning Representations, ICLR 2017—Workshop Track Proceedings (cited By 5)
8. Bianchi RA, Santos PE, Da Silva IJ, Celiberto LA, de Mantaras RL (2018) Heuristically accelerated reinforcement learning by means of case-based reasoning and transfer learning. J Intell Robot Syst 91(2):301–312
9. Bianchi RAC, Ribeiro CHC, Costa AHR (2009) On the relation between ant colony optimization and heuristically accelerated reinforcement learning. In: 1st International Workshop on Hybrid Control of Autonomous System, pp 49–55
10. Bodin L, Golden B, Assad A, Ball M (1983) Routing and scheduling of vehicles and crews—the state of the art. Comp Oper Res 10(2):63–211
11. Budak G, Chen X (2020) Evaluation of the size of time windows for the travelling salesman problem in delivery operations. Complex Intell Syst 6(3):681–695
12. Chiang H-TL, Faust A, Fiser M, Francis A (2019) Learning navigation behaviors end-to-end with autorl. IEEE Robot Autom Lett 4(2):2007–2014
13. Costa ML, Padilha CAA, Melo JD, Neto ADD (2016) Hierarchical reinforcement learning and parallel computing applied to the k-server problem. IEEE Latin Am Trans 14(10):4351–4357
14. Cunha B, Madureira AM, Fonseca B, Coelho D (2020) Deep reinforcement learning as a job shop scheduling solver: a literature review. In: Madureira AM, Abraham A, Gandhi N, Varela ML (eds) Hybrid intelligent systems. Springer International Publishing, Cham, pp 350–359
15. Cunha J, Serra R, Lau N, Lopes L, Neves A (2015) Batch reinforcement learning for robotic soccer using the q-batch update-rule. J Intell Robot Syst Theory Appl 80(3–4):385–399 cited by 4
16. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66
17. Even-Dar E, Mansour Y (2003) Learning rates for Q-learning. J Mach Learn Res 5:1–25
18. Gambardella LM, Dorigo M (1995) Ant-Q: a reinforcement learning approach to the traveling salesman problem. In: Proceedings of

the 12th International Conference on Machine Learning, pp 252–260

19. Giardini G, Kalmár-Nagy T (2011). Genetic algorithm for combinatorial path planning: the subtour problem. Math Probl Eng 2011
20. Haghzad Klidbary S, Bagheri Shouraki S, Sheikhpour Kourabbaslou S (2017) Path planning of modular robots on various terrains using q-learning versus optimization algorithms. Intell Serv Robot 10(2):121–136
21. Hamzehi S, Bogenberger K, Franeck P, Kaltenhäuser B (2019) Combinatorial reinforcement learning of linear assignment problems. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp 3314–3321
22. Hu Y, Yao Y, Lee W (2020) A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs. Knowl-Based Syst 204:106244
23. Hutter F, Hoos H, Leyton-Brown K (2014) An efficient approach for assessing hyperparameter importance. In: Proceedings of International Conference on Machine Learning 2014 (ICML 2014), pp 754–762
24. Hutter F, Kotthoff L, Vanschoren J, editors (2019) Automated machine learning: methods, systems, challenges. Springer. In press, http://automl.org/book
25. Jeong I-J, Illades Boy C (2018) Routing and refueling plans to minimize travel time in alternative-fuel vehicles. Int J Sustain Transp 12(8):583–591
26. Kaelbling L, Littman M, Moore A (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285
27. Khuller S, Malekian A, Mestre J (2007) To fill or not to fill: the gas station problem. In: European Symposium on Algorithms. Springer, pp 534–545
28. Kober J, Bagnell JA, Peters J (2013) Reinforcement learning in robotics: a survey. Int J Robot Res 32(11):1238–1274
29. Konar A, Chakraborty IG, Singh SJ, Jain LC, Nagar AK (2013) A deterministic improved q-learning for path planning of a mobile robot. IEEE Trans Syst Man Cybern Syst 43(5):1141–1153
30. Kormushev P, Calinon S, Caldwell D (2013) Reinforcement learning in robotics: applications and real-world challenges. Robotics 2(3):122–148 cited By 50
31. Kyaw PT, Paing A, Thu TT, Mohan RE, Le AV, Veerajagadheswar P (2020) Coverage path planning for decomposition reconfigurable grid-maps using deep reinforcement learning based travelling salesman problem. IEEE Access 8:225945–225956
32. Laporte G (1992) The traveling salesman problem: an overview of exact and approximate algorithms. Eur J Oper Res 59(2):231–247 cited By 484
33. Larrañaga P, Kuijpers C, Murga R, Inza I, Dizdarevic S (1999) Genetic algorithms for the travelling salesman problem: a review of representations and operators. Artif Intell Rev 13(2):129–170
34. Lenth RV (2009) Response-surface methods in R, using RSM. J Stat Softw 32(7):1–17
35. Levy D, Sundar K, Rathinam S (2014) Heuristics for routing heterogeneous unmanned vehicles with fuel constraints. Math Probl Eng 2014
36. Li C, Xu B (2020) Optimal scheduling of multiple sun-synchronous orbit satellites refueling. Adv Space Res 66(2):345–358
37. Li D, Zhao D, Zhang Q, Chen Y (2019) Reinforcement learning and deep learning based lateral control for autonomous driving [application notes]. IEEE Comput Intell Mag 14(2):83–98
38. Li J, Zhou M, Sun Q, Dai X, Yu X (2015) Colored traveling salesman problem. IEEE Trans Cybern 45(11):2390–2401
39. Li S, Xu X, Zuo L (2015) Dynamic path planning of a mobile robot with improved q-learning algorithm. In: Information and Automation, 2015 IEEE International Conference on, pp 409–414. IEEE
40. Liessner R, Schmitt J, Dietermann A, Bäker B (2019) Hyperparameter optimization for deep reinforcement learning in vehicle energy management. In: 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)
41. Lima-Júnior FC, Neto ADD, Melo JD (2010) Traveling salesman problem, theory and applications, chapter hybrid metaheuristics using reinforcement learning applied to salesman traveling problem. InTech, London, pp 213–236
42. Lin SH (2008) Finding optimal refueling policies in transportation networks. Algorithmic Aspects in Information and Management, Finding Optimal Refueling Policies in Transportation Networks 5034:280–291
43. Lin SH, Gertsch N, Russell J (2007) A linear-time algorithm for finding optimal vehicle refueling policies. Oper Res Lett 35(3):290–296
44. Lins RAS, Dória ADN, de Melo JD (2019) Deep reinforcement learning applied to the k-server problem. Expert Syst Appl 135:212–218
45. Liu F, Zeng G (2009) Study of genetic algorithm with reinforcement learning to solve the TSP. Expert Syst Appl 36(3):6995–7001
46. Lopes RHC (2011) Kolmogorov–Smirnov test. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 718–720
47. Low ES, Ong P, Cheah KC (2019) Solving the optimal path planning of a mobile robot using improved q-learning. Robot Auton Syst 115:143–161
48. Macharet DG, Campos MFM (2018) A survey on routing problems and robotic systems. Robotica 36(12):1781–1803
49. Montgomery DC (2017) Design and analysis of experiments, 9th edn. Wiley, New York
50. Murray C, Chu A (2015) The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transp Res Part C: Emerg Technol 54:86–109
51. Myers R H, Montgomery D C, Anderson-Cook C M (2009) Response surface methodology: process and product optimization using designed experiments, 3rd edn. Wiley, London
52. Ottoni ALC, Nepomuceno EG, de Oliveira MS (2018) A response surface model approach to parameter estimation of reinforcement learning for the travelling salesman problem. J Control Autom Electr Syst 29(3):350–359
53. Ottoni ALC, Nepomuceno EG, de Oliveira MS (2020) Development of a pedagogical graphical interface for the reinforcement learning. IEEE Latin Am Trans 18(01):92–101
54. Ottoni ALC, Nepomuceno EG, de Oliveira MS, de Oliveira DCR (2020) Tuning of reinforcement learning parameters applied to sop using the Scott-Knott method. Soft Comp 24(6):4441–4453
55. Ouaarab A, Ahiod B, Yang X-S (2014) Discrete cuckoo search algorithm for the travelling salesman problem. Neural Comp Appl 24(7–8):1659–1669
56. Papadopoulos K, Christofides D (2018) A fast algorithm for the gas station problem. Inform Process Lett 131:55–59 cited By 3
57. Polychronis G, Lalis S (2019) Dynamic vehicle routing under uncertain travel costs and refueling opportunities. In: Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019), pp 52–63
58. R Core Team (2018) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna
59. Rakshit P, Konar A, Bhowmik P, Goswami I, Das S, Jain LC, Nagar AK (2013) Realization of an adaptive memetic algorithm using differential evolution and q-learning: a case study in multirobot path planning. IEEE Trans Syst Man Cybern Syst 43(4):814–831
60. Rodrigues Junior AD, Cruz MMC (2013) A generic decision model of refueling policies: a case study of a Brazilian motor carrier. J Transp Lit 7(4):8–22
61. Russell SJ, Norvig P (2013) Artificial intelligence. Campus, 3rd ed
62. Schiffer M, Schneider M, Walther G, Laporte G (2019) Vehicle routing and location routing with intermediate stops: a review. Transp Sci 53(2):319–343 cited By 3

63. Schweighofer N, Doya K (2003) Meta-learning in reinforcement learning. Neural Netw 16(1):5–9
64. Silva MAL, de Souza SR, Souza MJF, Bazzan ALC (2019) A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. Expert Syst Appl 131:148–171
65. Sipahioglu A, Yazici A, Parlaktuna O, Gurel U (2008) Real-time tour construction for a mobile robot in a dynamic environment. Robot Auton Syst 56(4):289–295
66. Sun R, Tatsumi S, Zhao G (2001) Multiagent reinforcement learning method with an improved ant colony system. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 3:1612–1617
67. Sundar K, Rathinam S (2014) Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. IEEE Trans Autom Sci Eng 11(1):287–294 cited By 54
68. Sutton R, Barto A (2018) Reinforcement learning: an introduction, 2nd edn. MIT Press, Cambridge
69. Suzuki Y (2008) A generic model of motor-carrier fuel optimization. Naval Res Logist 55(8):737–746
70. Suzuki Y (2009) A decision support system of dynamic vehicle refueling. Decis Support Syst 46(2):522–531
71. Suzuki Y (2012) A decision support system of vehicle routing and refueling for motor carriers with time-sensitive demands. Decis Support Syst 54(1):758–767
72. Suzuki Y (2016) A dual-objective metaheuristic approach to solve practical pollution routing problem. Int J Prod Econ 176:143–153
73. Suzuki Y, Lan B (2018) Cutting fuel consumption of truckload carriers by using new enhanced refueling policies. Int J Prod Econ 202:69–80
74. Watkins CJ, Dayan P (1992) Technical note Q-learning. Mach Learn 8(3):279–292
75. Woo MH, Lee S-H, Cha HM (2018) A study on the optimal route design considering time of mobile robot using recurrent neural network and reinforcement learning. J Mech Sci Technol 32(10):4933–4939
76. Yan C, Xiang X (2018) A path planning algorithm for UAV based on improved q-learning. In: 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS), pp 1–5
77. Yavuz M, Çapar I (2017) Alternative-fuel vehicle adoption in service fleets: Impact evaluation through optimization modeling. Transp Sci 51(2):480–493 cited By 5
78. Yoo C, Fitch R, Sukkarieh S (2016) Online task planning and control for fuel-constrained aerial robots in wind fields. Int J Robot Res 35(5):438–453
79. Yu JJQ, Yu W, Gu J (2019) Online vehicle routing with neural combinatorial optimization and deep reinforcement learning. IEEE Trans Intell Transp Syst 20(10):3806–3817
80. Yu Z, Jinhai L, Guochang G, Rubo Z, Haiyan Y (2002) An implementation of evolutionary computation for path planning of cooperative mobile robots. In: Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on, vol 3, pages 1798–1802. IEEE
81. Zhang R, Prokhorchuk A, Dauwels J (2020) Deep reinforcement learning for traveling salesman problem with time windows and rejections. In: Proceedings of the International Joint Conference on Neural Networks, pp 1–8
82. Zhang T-J, Yang Y-K, Wang B-H, Li Z, Shen H-X, Li H-N (2019) Optimal scheduling for location geosynchronous satellites refueling problem. Acta Astronautica

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.