# Selection of Auxiliary Objectives in the Travelling Salesman Problem using Reinforcement Learning

Irina Petrova
ITMO University
49 Kronverkskiy ave.
Saint-Petersburg, Russia
petrova@rain.ifmo.ru

Arina Buzdalova
ITMO University
49 Kronverkskiy ave.
Saint-Petersburg, Russia
abuzdalova@gmail.com

## ABSTRACT

Auxiliary objectives can be used to increase efficiency of a single-objective evolutionary algorithm (EA). The corresponding approach is called multi-objectivization.

We consider two multi-objectivization methods: EA+RL and MOEA+RL, where MOEA is multi-objective EA, RL is reinforcement learning. In these methods, RL algorithm is used to select an objective during optimization process. In EA+RL only the selected objective is optimized, so a single-objective EA is used. In MOEA+RL the selected objective is optimized together with the target objective. Previously in these methods, RL for stationary environments were used. Recently, a new non-stationary RL algorithm was proposed. This algorithm was specially developed for the case when behaviour of auxiliary objectives changes during optimization process. However, this RL algorithm was tested only with EA+RL on some simple benchmark problems.

In the present paper we apply EA+RL and MOEA+RL for non-stationary environments to the travelling salesman problem (TSP) and compare them with the previously used multi-objectivization methods, as well as with EA+RL and MOEA+RL for stationary environments. We also analyze different types of auxiliary objectives for the TSP problem. For the most of the considered problem instances, EA+RL and MOEA+RL for non-stationary environments perform better than the other considered algorithms.

## Keywords

multi-objectivization; helper-objectives; non-stationarity

## 1. INTRODUCTION

Efficiency of a single-objective evolutionary algorithm (EA) can be increased by using auxiliary objectives [2,14,15]. The corresponding approach is called multi-objectivization. It may help to increase genetic diversity and to minimize the effect of getting stuck in local optima [9, 10]. The general idea of multi-objectivization is described below.

### 1.1 Multi-objectivization

Let us consider approaches of multi-objectivization. One of them, proposed by Knowles et al. [10], is based on decomposition of the target objective into several auxiliary objectives. These auxiliary objectives are optimized simultaneously instead of the target objective. The same principle is used in the work by Jähne et al. [8]. In this approach auxiliary objectives should be independent, but it is not always easy or possible [10].

Another approach, proposed by Jensen [9], is to use some additional objectives and optimize some of them together with the target objective. This approach was successfully applied to some NP-hard problems, for example, the Job Shop Scheduling Problem and the Travelling Salesman Problem [9, 11]. At each step of the algorithm, one or several auxiliary objectives are selected to be optimized [11].

There are several ways to select auxiliary objectives at each step of the optimization process. One of them is to select auxiliary objectives in random order [9]. Another one is to use an ad-hoc heuristic [11]. The first approach is general, but does not use information about an optimization problem being solved. The second one was proposed specially for the Job Shop Scheduling Problem and may not be applicable to some other problems. The EA+RL method was designed to deal with these issues [4].

#### *1.1.1 EA+RL method*

In the EA+RL method, reinforcement learning (RL) [7,21] is used to select an auxiliary objective, which is optimized in the current iteration of EA. In RL, an agent applies an action to an environment, then the environment returns some representation of its state and a numerical reward to the agent, and the process repeats.

In the EA+RL method, EA is treated as an environment. Each action of the agent corresponds to selection of an objective to be optimized at the current generation. The agent selects an objective from a set of auxiliary objectives and the target objective. Properties of auxiliary objectives usually are not known in advance, so the agent should learn which objective is the most efficient at the current optimization stage. Note that it is not aimed to optimize the auxiliary objectives, they are just used to increase the efficiency of optimization of the target objective.

Generally, the goal of RL is to maximize the total reward [21], calculated by formula $E[\sum_{t=0}^{\infty} \gamma^t r_t]$, where $\gamma$ is a discount factor and $r_t$ is a reward obtained at the $t^{\text{th}}$ iteration. In the EA+RL method, the reward $r_t$ is based on the difference of the target objective values in $t^{\text{th}}$ and $(t+1)^{\text{th}}$ iterations [13]. Therefore, the total reward which is roughly equivalent to the difference between the final and the initial values of the target objective, is maximized.

The EA+RL method is illustrated in Fig. 1, where $t$ is the number of the current iteration of EA. This method was shown to be efficient for a number of problems [4].

#### *1.1.2 MOEA+RL method*

In Jensen's approach the target objective and the selected

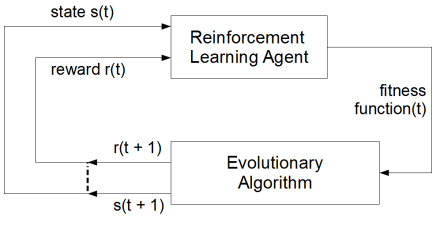


**Figure 1: EA+RL objective selection method**

auxiliary objective are optimized simultaneously by multi-objective EA (MOEA), which is designed to optimize several objectives simultaneously. One of the most widely used MOEAs [19] is NSGA-II [5]. This algorithm selects individuals according to the definition of Pareto-optimality.

A method of adaptive selection of auxiliary objectives in MOEAs was proposed in works [16]. The method is called MOEA + RL, as it is based on reinforcement learning. The only difference of MOEA+RL from EA+RL is that MOEA instead of EA is used, so the selected auxiliary objective is optimized simultaneously with the target objective.

## 1.2 RL in non-stationary environment

In early studies, it was implied that the environment was stationary. The environment is stationary if the obtained reward depends only on the applied action and the state of the environment [21]. Consequently, RL algorithms for stationary environments, such as Q-learning, were used. However, in the case when properties of auxiliary objectives change during the process of optimization, the reward for the same action can be different in the same RL state.

There are two ways of dealing with potential non-stationarity. The one way is to prevent non-stationarity by the properly defined state. However, it is not always easy or possible to define the appropriate state. What is more, for the most considered problems such state is still unknown. Another way is to use algorithms of RL for non-stationary environments.

Existing methods of RL for non-stationary environments were analyzed in the paper [17]. The Reinforcement Learning Context Detection (RLCD) [20] algorithm was applied in the EA+RL method for solving a benchmark problem but the experiments showed that RLCD was not efficient. Therefore another RL approach which can be used in the EA+RL method was proposed [17]. It was designed for dealing with non-stationarity, which arises when properties of auxiliary objectives may change independently of a state.

This RL approach is based on the conventional Q-learning algorithm [21]. The pseudocode of this approach is presented in Algorithm 1. The core idea of the approach is to reset $Q$ values when properties of auxiliary objectives have changed. There are two conditions for resetting $Q$ values. The first condition (line 20) indicates that behaviour of auxiliary objectives have changed. The second condition (line 22) shows that EA got stuck in local optima.

This approach was used to solve a benchmark problem [17]. The achieved results were better than the results obtained with the previously used methods. In this work, we apply this RL approach together with EA and MOEA for solving the Travelling Salesman Problem (TSP).

The rest of the paper is organized as follows. First, a TSP problem is described. Second, we analyze existing multi-objectivization approaches previously used to solve

**Algorithm 1** The proposed method

1: Form initial generation $G_0$
2: Initialize $Q(s,a) \leftarrow 0$ for each state $s$ and action $a$
3: Initialize iteration counter: $k \leftarrow 0$
4: Initialize first reset counter: $reset_1 \leftarrow 0$
5: Initialize second reset counter: $reset_2 \leftarrow 0$
6: **while** (specified number of generations or maximum value of target objective not reached) **do**
7: Evaluate current state $s_k$ and pass it to agent
8: Select action $a : Q(s,a) = \max_{a'} Q(s,a')$
9: Generate next generation $G_{k+1}$
10: Calculate reward $r$ and the next state $s'$
11: $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$
12: Initialize reset identifier: $reset \leftarrow false$
13: **if** reward is less than zero: $r < 0$ **then**
14: increment first reset counter: $reset_1 \leftarrow reset_1 + 1$
15: **if** reward is equal to zero: $r = 0$ **then**
16: increment second reset counter: $reset_2 \leftarrow reset_2 + 1$
17: **if** reward is greater than zero: $r > 0$ **then**
18: set first reset counter to zero: $reset_1 \leftarrow 0$
19: set second reset counter to zero: $reset_2 \leftarrow 0$
20: **if** first condition is fulfilled: $reset_1 = p_1$ **then**
21: $reset \leftarrow true$
22: **if** second condition is fulfilled: $reset_2 = p_2$ **then**
23: set second reset counter to zero: $reset_2 \leftarrow 0$
24: generate random number $n$ from 0 to 1
25: **if** $n \leq \frac{optimal - current}{optimal}$ **then**
26: $reset \leftarrow true$
27: **if** reset of learning is needed: $reset = true$ **then**
28: set $Q(s,\ a)$ for each $s$ and $a$ to zero : $Q(s,a) \leftarrow 0$
29: set first reset counter to zero: $reset_1 \leftarrow 0$
30: set second reset counter to zero: $reset_2 \leftarrow 0$
31: Update iteration counter: $k \leftarrow k + 1$

TSP, and explain why it is reasonable to use RL for non-stationary environments for solving TSP. Then the experiments are described. Finally, the EA+RL and MOEA+RL methods for non-stationary environment are compared with the other multi-objectivization approaches, which were studied in solving of TSP.

# 2. SOLVING TRAVELLING SALESMAN PROBLEM

TSP [1] is a classical combinatorial optimization problem [9]. It consists of a set of $n$ cities $c_1, \ldots, c_n$ and an associated $n \times n$ distance matrix $M$. The entries in $M$ represent the distances between the cities, so $M(c_1, c_2)$ is the distance from $c_1$ to $c_2$. The goal is to find a Hamiltonian path (a circular path visiting each city exactly once) with the smallest possible total distance. If $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a permutation of $(1, 2, \ldots, n)$ representing the tour of the cities, then the distance associated with the tour can be calculated as

$$D(\pi) = \sum_{i=1}^{n} M(c_{\pi[i]}, c_{\pi[i \oplus 1]}), \text{where}$$

$$i \oplus 1 = \begin{cases} i+1, i < n \\ 1, i = n \end{cases} \quad (1)$$

## 2.1 Related work

Several multi-objectivization approaches were used to solve

TSP. In all considered approaches, the target objective is the total distance of the tour. The auxiliary objectives are generated according to a specific approach. Three different ways to generate them are described below.

#### 2.1.1 Constructing objectives from decomposition

This approach was proposed by Knowles et al. [10]. The target objective is decomposed into distances of two subtours. The subtours are defined by two cities, $a$ and $b$. So the auxiliary objectives are:

$$f_1(\pi, a, b) = \sum_{i=\pi^{-1}[a]}^{\pi^{-1}[b]-1} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}), \quad (2)$$

$$f_2(\pi, a, b) = \sum_{i=\pi^{-1}[b]}^{n} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}) + \sum_{i=1}^{\pi^{-1}[a]-1} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}), \quad (3)$$

where $\pi^{-1}[x]$ is the position of $x$ in $\pi$. Note that $f_1(\pi, a, b) + f_2(\pi, a, b) = D(\pi)$. So the optimum of the target objective problem is one of the Pareto optimal points as required by the definition of multi-objectivization.

#### 2.1.2 Adding new objectives

In contrast to Knowles et al., Jensen proposed to add new auxiliary objectives, so called helper-objectives [9]. The auxiliary objectives proposed by Knowles et al. have a weakness for symmetric problems. So Jensen proposed the following auxiliary objectives without this drawback:

$$h(\pi, p) = \sum_{i=1}^{|p|} M(c_{\pi[\pi^{-1}[p[i]]\ominus 1]}, c_{\pi[i]}) + M(c_{\pi[i]}, c_{\pi[\pi^{-1}[p[i]]\oplus 1]}) \quad (4)$$

where $p$ is a subset of $1, 2, \ldots, n$ and $\ominus$ is the reverse of $\oplus$. The set $p$ is generated randomly, each city has a 50% probability of being in $p$. It is shown in [9] that it is the most efficient to simultaneously optimize one of the auxiliary objectives and the target objective. Each auxiliary objective is used for the same amount of consequent iterations. The order of auxiliary objectives is generated randomly.

#### 2.1.3 Constructing objectives from segmentation

The auxiliary objectives in Jensen's approach are selected randomly, so an inefficient objective may be used. Usage of an inappropriate auxiliary objective can shift search away from the global optimum. Jähne et al. proposed decomposition method that deals with this issue [8].

This method is based on auxiliary objectives proposed by Jensen. Two subsets of auxiliary objectives are generated: the subset $p$ is generated as in Jensen's approach, and $p^C$ is the complementary set of $p$. Then the two corresponding auxiliary objectives $h_1(\pi, p)$ and $h_2(\pi, p^C)$ are generated. The sum of these two objectives is equal to twice the target objective $D(\pi)$. So the requirement that the optimum of the target objective is one of the Pareto optimal points is fulfilled.

Jähne et al. showed that their approach outperforms methods of Jensen and Knowles et al. They also analysed three heuristic approaches of splitting the set of cities into subsets $p$ and $p^C$. Usage of these approaches increased efficiency of their method. In 2014, Lochtefeld et al. [12] proposed two new approaches of generating subsets for this decomposition method.

As we can see, the recent research on solving TSP by multi-objectivization is focused on improving the method proposed by Jähne et al. In the present work we analyse another direction of improvement and do not optimize all the objectives simultaneously but rather select them dynamically, which is closer to Jensen's approach.

### 2.2 Why we use algorithms for non-stationary environment

For all the considered definitions of the auxiliary objectives, their values depend on the current tour. An individual represents the tour. So the properties of the auxiliary objectives depend on the considered individual, thus potential non-stationarity exists. There is no state specially developed to prevent this non-stationarity. So we use a common RL state, proposed in [18]. This state was also used in non-stationary RL approach [17].

The RL state is a vector of three integer numbers which depend on the number of the current generation, the value of the target objective and entropy of fitness. The number of a generation is binned into four intervals on a logarithmic scale. The value of the target objective is averaged over the current generation, normalized and binned into four intervals. Entropy is binned into three equally sized discrete intervals. Therefore, a state is a vector of three numbers of the corresponding intervals. A lot of individuals correspond to each state. So the same auxiliary objective can have different properties at the same state.

## 3. EXPERIMENTS

We considered EA+RL and MOEA+RL methods and compared them with the approaches proposed by Knowles et al., Jensen and Jähne et al. RL algorithms for stationary and non-stationary environments were analysed. We compare the EA+RL method with a conventional single-objective genetic algorithm (GA), the simulated annealing (SA) method and approach, proposed by Knowles et al. MOEA+RL method is compared with the approaches of Jensen and Jähne et al. We considered two RL algoritms. One of them was stationary RL algorithm was Q-learning with $\varepsilon$-greedy strategy [21]. Another one was the non-stationary RL algorithm [17].

We compared all the algorithms on the fixed number of fitness calculations. The approach by Knowles et al. and the EA+RL method needed much more fitness calculations than MOEA + RL, Jensen and Jähne et al. approaches. So we did not compare EA+RL method with the approaches of Jensen and Jähne et al. for this reason. We also did not compare MOEA+RL with the approach of Knowles et al. for the same reason.

The implementation of NSGA-II may influence the results of the considered algorithms [3]. So we implemented approaches of Knowles et al. and Jähne et al. ourselves and ran all the algorithms using the same NSGA-II implementation.

All the considered algorithms were run 30 times on each problem instance, then the results were averaged. As in the works of Knowles et al. and Jähne et al., for all algorithms the same auxiliary objectives were generated once and used

**Table 1: Average (the top of a cell) and best (the bottom of a cell) target objective values. The dark (light) colored background corresponds to the first (second) best result.**

| Instance | Optimum | GA | SA | Knowles | S EA+RL K | NS EA+RL K | S EA+RL J | NS EA+RL J |
|---|---|---|---|---|---|---|---|---|
| ran20 | 1.91 | 2.03 | 2.55 | 2.66 | 1.93 | 1.92 | 1.96 | 1.92 |
| | | 1.91 | 2.54 | 2.54 | 1.91 | 1.91 | 1.91 | 1.91 |
| ran50 | 2.04 | 2.63 | 2.30 | 2.32 | 2.49 | 2.42 | 2.55 | 2.29 |
| | | 2.34 | 2.13 | 2.18 | 2.19 | 2.14 | 2.29 | 2.06 |
| euc50 | 5.03 | 5.62 | 5.72 | 5.78 | 5.51 | 5.49 | 5.51 | 5.48 |
| | | 5.37 | 5.69 | 5.69 | 5.37 | 5.37 | 5.37 | 5.37 |
| euc100 | 7.12 | 8.27 | 7.98 | 7.97 | 8.14 | 8.09 | 8.28 | 8.09 |
| | | 7.96 | 7.85 | 7.79 | 7.91 | 7.95 | 8.01 | 7.90 |
| kroB100 | 22141 | 23296.24 | 22529.20 | 22546.10 | 22952.11 | 22776.89 | 23161.56 | 22391.01 |
| | | 22509 | 22217 | 22141 | 22432 | 22243 | 22611 | 22139 |

for all runs. Parameters of the considered algorithms are described below.

## 3.1 EA+RL experiment description

We considered all five problem instances from the paper by Knowles et al. [10] and compared the EA+RL method and a traditional single-objective GA, SA and the approach proposed by Knowles et al.

We used the same evolutionary operator as Knowles et al. [10] did, namely the 2-change mutation operator. This operator selects two non-identical cities and reverses the order of all the cities between (and including) them.

There were 100 individuals in a generation of EA. The number of fitness evaluations was taken from [10].

For the $\varepsilon$-greedy Q-learning, learning rate was set to $\alpha = 0.6$, discount factor was $\gamma = 0.01$, exploration probability was $\varepsilon = 0.3$. For the non-stationary RL approach, parameters was set to $\alpha = 0.6$, $\gamma = 0.01$, $p_1 = 10$, $p_2 = 500$. We examined different values from 0 to 1000 for $p_2$ parameter and the best performance was achieved with $p_2 = 500$. This setting was found to be significantly better than $p_2$ in the ranges 0 –400 and 600–1000.

## 3.2 EA+RL experiment results

Experiment results of applying EA+RL method are shown in Table 1. For each problem, the average and the best target objective values are presented in the first line of a cell and in the second line of a cell correspondingly. The first column contains names of the instances. The best known solution is shown in the second column. The third column contains results obtained using traditional GA. The next two columns contain the results of simulated annealing (SA) and the approach proposed by Knowles et al. (Knowles) taken from [10]. The next two colums contain the results of EA+RL with $\varepsilon$-greedy Q-learning (S EA+RL K) and non-stationary RL (NS EA+RL K). In both S EA+RL K and NS EA+RL K the auxiliary objectives proposed by Knowles et al. are used. The next colum contains the results of EA+RL with $\varepsilon$-greedy Q-learning (S EA+RL J) with two auxiliary objectives which were generated as proposed by Jähne et al. The last column contains the results obtained with EA+RL with non-stationary RL (NS EA+RL J) with the same two auxiliary objectives.

The dark green background corresponds to the first best average result for each problem instance, while the light green background corresponds to the second best average result. The orange background corresponds to the first best result for each problem instance, while the light orange background corresponds to the second best result. The deviation of the average fitness in all considered algorithms is about 0.8%.

Let us compare the non-stationary EA+RL method which was run on the auxiliary objectives proposed by Jähne et al. with the other considered methods. It outperforms the non-stationary EA+RL method with auxiliary objectives proposed by Knowles et al. for 4 problem instances out of 5. For the euc100 problem instance, these approaches perform on par. It also outperforms the method of Knowles et al. for 4 problem instances out of 5. For the euc100 problem instance, the method proposed by Knowles et al. shows the best result. Finally, the considered method outperforms the remained approaches for all 5 problem instances.

Therefore, usage of auxiliary objectives proposed by Jähne et al. improves non-stationary EA+RL, so it outperforms other methods for the most of the considered problem instances. Note that at the same time the results obtained using these auxiliary objectives with stationary EA+RL are worse than the results obtained using auxiliary objectives proposed by Knowles et al.

## 3.3 MOEA+RL experiment description

We considered some of the problem instances from the works of Jensen [9] and Jähne et al. [8]. These instances were taken from the TSPLIB website[1]. They consist of 100 to 1002 cities. The number in the name of an instance corresponds to the number of cities.

We used the same mutation and crossover operators as in [8, 9]. We also applied 2-Opt heuristic, which was used in [8, 9]. There were 100 individuals in a generation of MOEA. The number of fitness evaluations were calculated by the formula from [8]: $E(N) = \sqrt{N^3} \times 15$, where $N$ is the number of the cities.

In the MOEA+RL method and the approach of Jensen the target objective and one of auxiliary objectives were simultaneously optimized.

For the $\varepsilon$-greedy Q-learning, learning rate was set to $\alpha = 0.6$, discount factor was $\gamma = 0.01$, exploration probability was $\varepsilon = 0.3$. For the non-stationary RL approach, parameters were set to $\alpha = 0.6$, $\gamma = 0.01$, $p_1 = 10$, $p_2 = 10$. We examined different values from 0 to 100 for the $p_2$ parameter. The best performance was achieved with $p_2 = 10$. The dif-

[1]http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

**Table 2: Average (the top of a cell) and best (the bottom of a cell) target objective values. The dark (light) colored background corresponds to the first (second) best result.**

| Instance | Optimum | NS MOEA+RL | S MOEA+RL | Jähne | Jensen-Jähne | Jensen |
|---|---|---|---|---|---|---|
| kroB100 | 22141 | 22144 | 22145 | 22150 | 22158 | 22155 |
| | | 22139 | 22139 | 22139 | 22139 | 22139 |
| kroD100 | 21294 | 21342 | 21353 | 21344 | 21349 | 21347 |
| | | 21294 | 21294 | 21294 | 21294 | 21294 |
| kroE100 | 21294 | 22093 | 22095 | 22169 | 22095 | 22100 |
| | | 22068 | 22068 | 22068 | 22068 | 22068 |
| eil101 | 629 | 641.39 | 641.84 | 641.50 | 641.59 | 641.95 |
| | | 640 | 640 | 640 | 640 | 640 |
| pr124 | 59030 | 59030 | 59030 | 59030 | 59032 | 59052 |
| | | 59030 | 59030 | 59030 | 59030 | 59030 |
| bier127 | 118282 | 118324 | 118394 | 118387 | 118408 | 118394 |
| | | 118293 | 118293 | 118293 | 118293 | 118293 |
| pr136 | 96772 | 96975 | 97000 | 96980 | 97193 | 97063 |
| | | 96785 | 96835 | 96795 | 96785 | 96835 |
| kroA150 | 26524 | 26540 | 26558 | 26533 | 26557 | 26558 |
| | | 26524 | 26524 | 26524 | 26524 | 26554 |
| kroB150 | 26130 | 26153 | 26166 | 26170 | 26166 | 26174 |
| | | 26127 | 26127 | 26127 | 26127 | 26127 |
| pr152 | 73682 | 73693 | 73702 | 73904 | 73820 | 73821 |
| | | 73683 | 73683 | 73687 | 73683 | 73820 |
| pr439 | 107217 | 107675 | 107677 | 107748 | 108035 | 107743 |
| | | 107241 | 107248 | 107301 | 107258 | 107248 |
| rat575 | 6773 | 6869 | 6872 | 6874 | 6863 | 6877 |
| | | 6833 | 6824 | 6847 | 6835 | 6826 |
| pr1002 | 259045 | 263158 | 263318 | 263425 | 263184 | 263189 |
| | | 261444 | 261970 | 261231 | 262023 | 260971 |

ference between $p_2$ values for EA+RL and for MOEA+RL can be explained by the different number of fitness evaluations and, as a consequence, the different number of iterations.

## 3.4 MOEA+RL experiment results

Experiment results of the MOEA+RL method are shown in Table 2. For each problem, the average and the best target objective values are presented in the first and in the second lines of a cell correspondingly. The first column contains names of the instances. The best known solutions are shown in the second column. The next four colums contain results of MOEA+RL with non-stationary RL approach (NS MOEA+RL), MOEA+RL with $\varepsilon$-greedy Q-learning (S MOEA+RL), Jähne et al. approach [8] (Jähne) and Jensen approach [9] (Jensen-Jähne). In all these approaches, two auxiliary objectives which were generated as proposed by Jähne et al. were used. The last column contains the results of the Jensen approach which was run on ten auxiliary objectives generated as proposed by Jensen.

The dark green background corresponds to the first best average result for each problem instance, while the light green background corresponds to the second best average result. The orange background corresponds to the first best result for each problem instance, while the light orange background corresponds to the second best result. The deviation of the average fitness in all considered algorithms is about 0.05%.

Let us compare non-stationary MOEA+RL with the other methods. The results of the comparison are presented in

**Table 3: Comparison of non-stationary MOEA+RL with the other methods**

| | S MOEA+RL | Jähne | Jensen-Jähne | Jensen |
|---|---|---|---|---|
| Better | 12 | 11 | 12 | 13 |
| Equal | 1 | 1 | 0 | 0 |
| Worse | 0 | 1 | 1 | 0 |

Table 3. This table contains numbers of the instances that correspond to the cases when the results obtained with the considered algorithm were better, worse or equal to the results obtained with the other algorithms. For the most problem instances, the MOEA+RL method with non-stationary RL outperforms the other considered methods.

According to the multiple signed test [6], the MOEA+RL method with non-stationary RL is distinguishable from the other ones at the level of statistical significance $\alpha = 0.05$. To sum up, non-stationary MOEA+RL with the auxiliary objectives proposed by Jähne et al. turns to be the most efficient approach for the considered set of the TSP problem instances.

# 4. CONCLUSION

We applied the recently proposed non-stationary RL approach together with the EA+RL and MOEA+RL methods for solving the Travelling Salesman Problem.

We compared the results of non-stationary EA+RL with the results of stationary EA+RL, traditional single-objective genetic algorithm, simulated annealing and the approach

proposed by Knowles et al. We also studied usage of the auxiliary objectives proposed by Knowles et al. and the auxiliary objectives proposed by Jähne et al. Non-stationary EA+RL with the auxiliary objectives proposed by Jähne et al. turned to be more efficient than the other algorithms. However, these considered algorithms, including non-stationary EA+RL, need much more fitness calculations than the MOEA + RL and the methods of Jensen and Jähne et al.

We compared the results of non-stationary MOEA+RL with the results of stationary MOEA+RL, the method of Jähne et al. and the method of Jensen which was run both on Jensen's auxiliary objectives and the auxiliary objectives proposed by Jähne et al. The non-stationary MOEA+RL method with the auxiliary objectives proposed by Jähne et al. turned to be the most efficient for the considered set of TSP instances.

To sum up, the following conclusions may be drawn:

- The obtained results confirm that the auxiliary objectives proposed by Jähne et al. [8] are efficient for solving the Travelling Salesman Problem.
- Using non-stationary reinforcement learning to dynamically select auxiliary objectives is a promising approach, since it outperformed other considered methods. More precisely, the combined use of the auxiliary objectives from [8] and selection with non-stationary reinforcement learning gave the best results.
- Let us highlight the two major ways of using auxiliary objectives. The first way is to simultaneously optimize the auxiliary objectives, while the target objective is not optimized explicitly [10]. Most of the recent research is focused on this approach [8, 12]. The second way is to optimize the target objective together with a one dynamically selected auxiliary objective [9]. The results of the present work suggest that the second approach may be more efficient than the first one when a proper selection method is used. Particularly, for the considered instances of the travelling salesman problem, the second approach with the reinforcement learning based selection outperformed the other methods.

# 5. REFERENCES


[1] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, NJ, USA, 2007.

[2] T. Brys, A. Harutyunyan, P. Vrancx, M. E. Taylor, D. Kudenko, and A. Nowé. Multi-objectivization of reinforcement learning problems by reward shaping. In *2014 International Joint Conference on Neural Networks*, pages 2315–2322, 2014.

[3] M. Buzdalov, I. Petrova, and A. Buzdalova. NSGA-II implementation details may influence quality of solutions for the job-shop scheduling problem. In *Proceedings of Genetic and Evolutionary Computation Conference (Companion)*, pages 1445–1446, 2014.

[4] A. Buzdalova and M. Buzdalov. Increasing Efficiency of Evolutionary Algorithms by Choosing between Auxiliary Fitness Functions with Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning and Applications*, volume 1, pages 150–155, 2012.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. *Transactions on Evolutionary Computation*, 6:182–197, 2000.

[6] J. Derrac, S. Garcia, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

[7] A. Gosavi. Reinforcement Learning: A Tutorial Survey and Recent Advances. *INFORMS Journal on Computing*, 21(2):178–192, 2009.

[8] M. Jähne, X. Li, and J. Branke. Evolutionary algorithms and multi-objectivization for the travelling salesman problem. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 595–602, New York, NY, USA, 2009. ACM.

[9] M. T. Jensen. Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation: Evolutionary Computation Combinatorial Optimization. *Journal of Mathematical Modelling and Algorithms*, 3(4):323–347, 2004.

[10] J. D. Knowles, R. A. Watson, and D. Corne. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pages 269–283. Springer-Verlag, 2001.

[11] D. F. Lochtefeld and F. W. Ciarallo. Deterministic Helper-Objective Sequences Applied to Job-Shop Scheduling. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 431–438. ACM, 2010.

[12] D. F. Lochtefeld and F. W. Ciarallo. An analysis of decomposition approaches in multi-objectivization via segmentation. *Appl. Soft Comput.*, 18:209–222, 2014.

[13] S. D. Müller, N. N. Schraudolph, and P. D. Koumoutsakos. Step size adaptation in evolution strategies using reinforcement learning. In *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 151–156. IEEE Press, 2002.

[14] F. Neumann and I. Wegener. Minimum Spanning Trees Made Easier via Multi-objective Optimization. *Natural Computing*, 5(3):305–319, 2006.

[15] F. Neumann and I. Wegener. Can Single-Objective Optimization Profit from Multiobjective Optimization? In *Multiobjective Problem Solving from Nature*, Natural Computing Series, pages 115–130. Springer Berlin Heidelberg, 2008.

[16] I. Petrova, A. Buzdalova, and M. Buzdalov. Improved Helper-Objective Optimization Strategy for Job-Shop Scheduling Problem. In *Proceedings of the International Conference on Machine Learning and Applications*, volume 2, pages 374–377. IEEE Computer Society, 2013.

[17] I. Petrova, A. Buzdalova, and M. Buzdalov. Improved selection of auxiliary objectives using reinforcement

learning in non-stationary environment. In *Proceedings of the International Conference on Machine Learning and Applications*, pages 580–583, 2014.

[18] Y. Sakurai, K. Takada, T. Kawabe, and S. Tsuruta. A method to control parameters of evolutionary algorithms by using reinforcement learning. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2010 Sixth International Conference on*, pages 74–79, 2010.

[19] C. Segura, C. A. C. Coello, G. Miranda, and C. Léon. Using multi-objective evolutionary algorithms for single-objective optimization. *4OR*, 3(11):201–228, 2013.

[20] B. C. D. Silva, E. W. Basso, A. L. C. Bazzan, and P. M. Engel. Dealing with Non-stationary Environments Using Context Detection. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 217–224. ACM Press, 2006.

[21] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.