A Novel Approach Based on Reinforcement Learning for Finding Global Optimum

Cenk Ozan¹, Ozgur Baskan², Soner Haldenbilen²

¹Department of Civil Engineering, Faculty of Engineering, Adnan Menderes University, Aydin, Turkey
²Department of Civil Engineering, Faculty of Engineering, Pamukkale University, Denizli, Turkey

Email: *cenk.ozan@adu.edu.tr

Abstract

A novel approach to optimizing any given mathematical function, called the MODified REinforcement Learning Algorithm (MORELA), is proposed. Although Reinforcement Learning (RL) is primarily developed for solving Markov decision problems, it can be used with some improvements to optimize mathematical functions. At the core of MORELA, a sub-environment is generated around the best solution found in the feasible solution space and compared with the original environment. Thus, MORELA makes it possible to discover global optimum for a mathematical function because it is sought around the best solution achieved in the previous learning episode using the sub-environment. The performance of MORELA has been tested with the results obtained from other optimization methods described in the literature. Results exposed that MORELA improved the performance of RL and performed better than many of the optimization methods to which it was compared in terms of the robustness measures adopted.

Keywords

Reinforcement Learning, Mathematical Function, Global Optimum, Sub-Environment, Robustness Measures

1. Introduction

If \( f(x) \) is a function of decision variables, where \( x \in S \), \( S \) is the feasible search space and \( S \subseteq \mathbb{R}^n \), an optimization problem can be defined as finding the value of \( x_{\text{best}} \) in \( S \) that makes \( f(x) \) optimal for all \( x \) values. Despite the fact that different meta-heuristic algorithms have been improved especially in last two decades, the contributions of Reinforcement Learning (RL) to this area are still limited comparing to others. Numerous studies such as genetic algorithm based methods [1] [2], ant colony based algorithms [3] [4], harmony
search approach [5], modified firefly algorithm [6] and hybrid heuristic methods [7] [8] [9] have been proposed to optimize any given mathematical function. Apart from these applications, there are some attempts for solving engineering problems in different fields using RL based algorithms in the relevant literature. Hsieh and Su [10] developed Q-learning based optimization algorithm for solving economic dispatch problem after they tested their proposed algorithm for optimizing standard mathematical test functions. Their results showed that the proposed algorithm outperformed many existing optimization algorithms. Similarly, Samma et al. [11] tested newly developed RL based Memetic Particle Swarm Optimization algorithm on mathematical functions and real-world benchmark problems. Numerical examples applied indicated that the proposed algorithm is able to produce better results than those did by other optimization algorithms given in the literature. Walraven et al. [12] proposed a new algorithm to minimize traffic flow using RL algorithm. They formulated traffic flow optimization problem as Markov Decision Process and used Q-learning method to reduce traffic congestion. Another application of the RL algorithm has been proposed by Tozer et al. [13]. They developed a new RL algorithm that is able to find optimal solution with several conflicting objectives. The proposed algorithm has been tested on multi-objectives path finding problems with deterministic and stochastic environments.

Although most of the methods described in the literature are able to discover global optimum of any given optimization problem, the performance of newly developed algorithms should be investigated. Therefore, we present a MOdified REinforcement Learning Algorithm (MORELA) approach which differs from RL based approaches by means of generating a sub-environment based on the best solution obtained so far which is saved to prevent the search being trapped at local optimums. And then, all of the function values with corresponding decision variables in MORELA are ranked from best to worst. In this way, the sub-environment is compared with the original environment. If one of the members of the sub-environment produces better functional value, it is added to the original environment and the worst solution is omitted. This makes the searching process more effective because the global optimum is sought around the best solution achieved so far, with the assistance of the sub-environment and the original environment.

RL has been attracted a lot of attention from scientific community for solving different class of problems especially in last decades [14]. In RL, there is a remarkable interaction between agent and environment which contains everything apart from the agent. The agent receives information from the environment through cooperating with each other. Depending on the information obtained, the agent changes the environment by means of implementing an action. This modification is transferred to the agent by means of a signal. The environment generates numerical values called rewards and the agent efforts to maximize them. The agent and environment affect each other at each time \( t \) in which the agent gets information about the environment’s state \( s_t \in S \), where \( S \) consists
of states. To this respect, the agent performs an action \( a_t \in A(s_t) \), where \( A(s_t) \) includes actions in state \( s_t \) and gains a reward \( r_{t+1} \in R \), and it is located in a new state \( s_{t+1} \). **Figure 1** represents this cooperation [15].

There are three primary types of RL based methods, each with its advantages and disadvantages. The Monte Carlo and Temporal difference learning methods are able to learn only from experience whereas the other one, called Dynamic programming, requires a model of the environment. Therefore, because of not need to have a model, they are superior to Dynamic programming. Indeed, temporal difference learning methods are at the core of RL [16]. On the other hand, Q-learning, one of the temporal difference methods, evaluates \( Q \) values, which represent quality of a given state-action pair [17]. It benefits experience to update members of \( Q \) table [14]. \( Q \) table has elements as \( Q(s, a) \) for each state-action pair. The \( Q \)-learning determines the \( Q \) value, which reflects an action \( a \) performed in a state \( s \), and selects the best actions [18]. The \( Q \) table is created as shown in **Table 1** [19].

\( Q \) learning algorithm includes in a sequence of learning episodes (i.e. iteration). At each learning episode, the agent chooses an action in accordance with information provided from a state \( s \). The agent deserves to receive a reward considering its \( Q \) value and observes the next state, \( s' \). The agent replaces its \( Q \) value with that provided according to Equation (1):

\[
Q(s, a) \leftarrow (1-\alpha) \times Q(s, a) + \alpha \times \left[ r_t(s, a) + \gamma \times \max_{a'} Q_{best}(s', a') \right]
\]

**Figure 1.** The cooperation between agent and environment.

**Table 1.** \( Q \) learning process.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initialize ( Q ) values</td>
</tr>
<tr>
<td>2</td>
<td>Repeat ( t ) times (( t ) = number of learning episodes)</td>
</tr>
<tr>
<td>3</td>
<td>Select a random state ( s )</td>
</tr>
<tr>
<td>4</td>
<td>Repeat until the end of the learning episode</td>
</tr>
<tr>
<td>5</td>
<td>Select an action ( a )</td>
</tr>
<tr>
<td>6</td>
<td>Receive an immediate reward ( r )</td>
</tr>
<tr>
<td>7</td>
<td>Observe the next state ( s' )</td>
</tr>
<tr>
<td>8</td>
<td>Update the ( Q ) table according to the update rule</td>
</tr>
<tr>
<td>9</td>
<td>Set ( s = s' )</td>
</tr>
</tbody>
</table>
where \( Q_t(s,a) \) is the updated Q value, \( Q_{t-1}(s,a) \) is the Q-value saved in the Q table, \( r_t(s,a) \) is the reward for state-action pair, \( \alpha \) is the learning rate, and \( \gamma \) is the discounting parameter \[18\].

This paper proposes a new and robust approach, called MORELA, to optimize any given mathematical function. MORELA approach varies from other RL based approaches through generating a sub-environment. In this way, developed MORELA approach has ability to find global optimum for any given mathematical optimization because it is sought both around the best solution achieved so far with the assistance of the sub-environment and the original environment. The rest of this paper is organized as follows. The definition of fundamental principles of MORELA is provided in Section 2. Section 3, which also contains comparison of MORELA and RL, various contrastive analyses of MORELA such as robustness analysis, comparisons with other related methods, explanation of evolving strategy of MORELA and investigation of the effect of high dimensionality, presents numerical experiments and the last section is conclusions.

2. The MORELA Approach

There are several studies related to RL combined with different heuristic methods for solving different types of optimization problems. Liu and Zeng \[20\] developed genetic algorithm based method with assistance of reinforcement mutation to tackle the problem of travelling salesman. Integrating RL with different algorithms has been used to address the problem of robot control with unrecognized obstacles \[21\]. The experimental results revealed that the hybrid approach is superior to RL for planning robot motion whereas RL faced some difficulties. Chen et al. \[22\] proposed genetic network programming with RL for stock trading. The comparison of the results with those obtained from other methods shows the noticeable performance of their method. Similarly, Wu et al. \[23\] improved a RL based method considering multi-agent for tackling job scheduling problems. Their results showed that the method is comparable to that of some centralized scheduling algorithms.

From a different viewpoint, Derhami et al. \[24\] applied RL based algorithms for ranking most relevant web pages to user’s search. The algorithms are tested by using well-known benchmark datasets. Their results showed that the use of RL makes noticeable improvements in ranking of web pages. Khamis and Gomaa \[25\] used an adaptive RL approach to tackle traffic signal control at junctions by considering multi-objective. Recently, Ozan et al. \[26\] developed RL based algorithm in determining optimal signal settings for area traffic control. Results revealed that the proposed algorithm outperforms genetic algorithm and hill climbing methods even if there is a heavy demand condition.

Hybridizing RL algorithms with other optimization methods is a powerful technique to tackle different types of optimization problems arising in different fields. Thus, in the context of this paper, we focus on applicability of RL based algorithms to discover global optimum for any mathematical function. The proposed algorithm called MORELA is on the basis of Q-learning, which is a
model-free RL approach. In addition, a sub-environment is generated in MORELA so that the environment consists of original and sub-environment differently from other RL based approaches as shown in Equation (2) [26].

\[
\begin{bmatrix}
Q_i^{11}(s,a) & Q_i^{12}(s,a) & \cdots & Q_i^{1n}(s,a) \\
Q_i^{21}(s,a) & Q_i^{22}(s,a) & \cdots & Q_i^{2n}(s,a) \\
\vdots & \vdots & \ddots & \vdots \\
Q_i^{m1}(s,a) & Q_i^{m2}(s,a) & \cdots & Q_i^{mn}(s,a) \\
Q_i^{(m+1)1}(s,a) & Q_i^{(m+1)2}(s,a) & \cdots & Q_i^{(m+1)n}(s,a) \\
\vdots & \vdots & \ddots & \vdots \\
Q_i^{(2m+1)1}(s,a) & Q_i^{(2m+1)2}(s,a) & \cdots & Q_i^{(2m+1)n}(s,a)
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
f_1 \\
f_2 \\
\vdots
\end{bmatrix}
\]  

where \( m \) is the size of the original environment, \( n \) is the number of decision variables, and \( f \) is fitness value at the \( t \)th learning episode. As shown in Equation (2), \( Q(s,a) \) value achieved in the previous learning episode is kept in the \( (m+1) \)th row. At the \( t \)th learning episode, a sub-environment is generated as given in Equation (3) and located between rows \( (m+2) \) and \( (2m+1) \). Thus, a global optimum is explored around the best solution with assistance of sub-environment with vector of \( \beta \) which must be decreased during algorithm process in order to make searching more effectively in a reduced environment. The limits of \( \beta \) can be selected by considering upper and lower constraints of a given problem [3].

\[
\text{rnd} \left( Q_{i-1}^{\text{best}}(s,a) - \beta; Q_i^{\text{best}}(s,a) + \beta \right)
\]  

After generating the sub-environment, the solution vectors located in both environments are ranked from best to worst according to their fitness values. With assistance of this sorting, the worst solution vector is excluded from environment whereas the solution vector provided a better functional value is included. Thus, MORELA may gain the ability to solve any given optimization problem without prematurely converging. Figure 2 shows the process of MORELA.

**Figure 2.** The process of MORELA.
In MORELA, each action \( a \) in state \( s \) is rewarded as shown in Equation (4) [26].

\[
r_i(s,a) = \frac{Q^{best}(s,a) - Q(s,a)}{Q_i(s,a)}
\]  

(4)

where \( r_i(s,a) \) is the reward function, \( Q(s,a) \) is the \( Q \) value and \( Q^{best}(s,a) \) is the best \( Q \) value obtained in the \( i^{th} \) learning episode. In MORELA, the reward value is determined for each member of the solution vector by considering its \( Q \) value and the best \( Q \) value provided so far. The reward values come to close to the value “0” at the end of the solution process because of the structure of the reward function. In fact, a solution receives less reward when it is located closer to global optimum than the others. On the other hand, the probability of global optimum finding for further located solutions may be increased by means of providing them bigger rewards. Thus, the reward function developed may be referred to as penalty contrary to reward.

3. Numerical Experiments

An application of MORELA was carried out by solving several mathematical functions taken from the literature. However, before solving these functions, it may be essential to demonstrate the effectiveness of MORELA over RL. For this purpose, a performance comparison was conducted by solving a mathematical function. MORELA was encoded in the MATLAB for all test functions, using a computer with Intel Core i7 2.70 GHz and 8 GB of RAM. The related solution parameters for MORELA were set as follows: the environment size is taken as 20, the discounting parameter \( \gamma \) and the learning rate \( \alpha \) were taken as 0.2 and 0.8, respectively. The search space parameter \( \beta \) was chosen according to the search domain for all test functions. The solution process was terminated when a pre-determined stopping criterion was met. The stopping criterion was properly selected for each function, and it theoretically guarantees that global optimum will be found eventually.

3.1. Comparison of MORELA and RL

The test function used to compare MORELA and RL is given in Equation (5). It has a global optimum solution of \( f(x_1, x_2) = -10 \) when \( (x_1, x_2) = (-10, 0) \). A graph of the objective function is given in Figure 3. The convergence behaviors of MORELA and RL are illustrated in Figure 4.

\[
f(x_1, x_2) = \frac{x_1}{1 + |x_2|}
\]

(5)

As seen in Figure 4, same initial solutions were used for both algorithms to compare them realistically. Simulation results reveal that MORELA requires many fewer learning episodes than RL although both algorithms are capable of finding globally optimal solution for this function. MORELA needs only 1176 learning episodes to find optimal solution whereas RL requires 7337.
3.2. Robustness Analysis

A robustness analysis for MORELA was carried out by using succeed ratio (SR) given in Equation (6).
\[ SR = \frac{100 \times N_s}{N_T} \]  

(6)

where \( N_s \) is the number of successful runs which indicates that the algorithm produces the best solution at the required accuracy and \( N_T \) is the total number of runs which is set to 50 to make a fair comparison. For this experiment, a run is accepted as successful when its objective function value is around 3.5% of global optimum. The robustness analysis for MORELA, PSACO [7] and other methods [27] are given in Table 2. As shown, MORELA is able to find global optimum with very high success in comparison with other algorithms, except PSACO. Although PSACO produces higher success ratios than MORELA for functions F12 and F13, for given total numbers of runs, MORELA and PSACO yield same results for functions F2, F7, F9 and F16. CPSO, PSO and GA in particular yield worse results than MORELA and PSACO.

3.3. Further Comparisons of MORELA with Other Methods

To gauge the performance of MORELA against the performance of some other methods described in the literature, sixteen well-known benchmark problems were used which are given in Appendix A. Functions 6, 7, 9, 13 and 16 are taken from Shelokar et al. [7]. Functions 8 and 14 are adopted from Sun and Dong [28] and Chen et al. [29], respectively. The rest of the functions are from Baskan et al. [3]. Table 3 lists algorithms compared with MORELA.

To assess the ability of MORELA, its performance was compared with 12 algorithms listed in Table 3. For this purpose, sixteen test functions were used based on 100 runs. The results are shown in Table 4 in terms of the best function value, the number of learning episodes, the best solution time, the success ratio, the average number of learning episodes and the average error. The best function value is the value obtained for all runs at the required accuracy that indicates that the algorithm reached to the global optimum. The required accuracy is determined as the absolute difference between the best function value and the theoretical global optimum. For this experiment, a value of “0” was chosen as required accuracy for all test functions. The number of learning episodes and the best solution time are the number of runs and the time required to obtain the best function value, respectively. The average number of learning episodes is de-

Table 2. Results of robustness analysis.

<table>
<thead>
<tr>
<th>Function</th>
<th>The values of SR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MORELA</td>
</tr>
<tr>
<td>F2</td>
<td>100</td>
</tr>
<tr>
<td>F7</td>
<td>100</td>
</tr>
<tr>
<td>F9</td>
<td>100</td>
</tr>
<tr>
<td>F12</td>
<td>98</td>
</tr>
<tr>
<td>F13</td>
<td>96</td>
</tr>
<tr>
<td>F16</td>
<td>100</td>
</tr>
</tbody>
</table>
**Table 3.** The algorithms compared with MORELA.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Algorithm</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>F4-F7</td>
<td>SZGA</td>
<td>Successive zooming genetic algorithm [1]</td>
</tr>
<tr>
<td>F1-F2-F3-F4</td>
<td>IGARSET</td>
<td>Improving GA [2]</td>
</tr>
<tr>
<td>F7-F12-F13</td>
<td>ACO</td>
<td>Ant colony optimization [30]</td>
</tr>
<tr>
<td>F4-F5-F11-F12-F13-F15-F16</td>
<td>PSACO</td>
<td>Particle swarm and ant colony algorithm [7]</td>
</tr>
<tr>
<td>F5</td>
<td>ECTS</td>
<td>Enhanced continuous tabu search [31]</td>
</tr>
<tr>
<td>F10</td>
<td>ACORSES</td>
<td>Ant colony optimization [3]</td>
</tr>
<tr>
<td>F8</td>
<td>SA</td>
<td>Simulated annealing [28]</td>
</tr>
<tr>
<td>F14</td>
<td>RW-PSO+BOF</td>
<td>Random walking particle swarm optimization [29]</td>
</tr>
<tr>
<td>F5-F6-F7-F9-F11-F12-F13</td>
<td>GA-PSO</td>
<td>Genetic algorithm particle swarm optimization [8]</td>
</tr>
<tr>
<td>F4-F7-F9</td>
<td>GAWLS</td>
<td>Genetic algorithm [32]</td>
</tr>
<tr>
<td>F1-F3</td>
<td>HAP</td>
<td>Hybrid ant particle optimization algorithm [4]</td>
</tr>
<tr>
<td>F1-F4-F10</td>
<td>ACO-NPU</td>
<td>Ant colony optimization [9]</td>
</tr>
<tr>
<td>All problems</td>
<td>MORELA</td>
<td>Modified reinforcement learning algorithm (This study)</td>
</tr>
</tbody>
</table>

**Table 4.** The results of MORELA and compared algorithms.

<table>
<thead>
<tr>
<th>Function</th>
<th>Method</th>
<th>Best function value</th>
<th>Number of learning episodes</th>
<th>Best solution time (sec)</th>
<th>Success ratio</th>
<th>Average number of learning episodes</th>
<th>Average error</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>IGARSET</td>
<td>0</td>
<td>2174</td>
<td>0.0568</td>
<td>NA</td>
<td>2375</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>ACO-NPU</td>
<td>0</td>
<td>20,000</td>
<td>0.0590</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>HAP</td>
<td>2.4893e−8</td>
<td>100</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>MORELA</td>
<td>0</td>
<td>68,760</td>
<td>0.8688</td>
<td>100</td>
<td>71,000</td>
<td>NA</td>
</tr>
<tr>
<td>F2</td>
<td>IGARSET</td>
<td>−2</td>
<td>2400</td>
<td>0.0614</td>
<td>NA</td>
<td>3111</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>MORELA</td>
<td>−2</td>
<td>34,920</td>
<td>1.1790</td>
<td>97</td>
<td>36,200</td>
<td>0</td>
</tr>
<tr>
<td>F3</td>
<td>IGARSET</td>
<td>2.08e−27</td>
<td>1821</td>
<td>0.0666</td>
<td>NA</td>
<td>2156</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>HAP</td>
<td>2.56e−39</td>
<td>100</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>MORELA</td>
<td>9.71e−40</td>
<td>31,620</td>
<td>0.9584</td>
<td>98</td>
<td>31,740</td>
<td>9.33e−34</td>
</tr>
<tr>
<td>F4</td>
<td>SZGA</td>
<td>2.9e−8</td>
<td>4000</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>IGARSET</td>
<td>0</td>
<td>1004</td>
<td>0.0485</td>
<td>NA</td>
<td>1065</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>GAWLS</td>
<td>0</td>
<td>2572</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>PSACO</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>370</td>
<td>5.55e−17</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>ACO-NPU</td>
<td>0</td>
<td>1000</td>
<td>0.0556</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>MORELA</td>
<td>0</td>
<td>34,340</td>
<td>0.5660</td>
<td>99</td>
<td>35,700</td>
<td>1.11e−18</td>
</tr>
<tr>
<td>F5</td>
<td>ECTS</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>338</td>
<td>3e−08</td>
</tr>
<tr>
<td></td>
<td>PSACO</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>190</td>
<td>7.69e−29</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>GA-PSO</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>206</td>
<td>0.00004</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>MORELA</td>
<td>0</td>
<td>36,955</td>
<td>21.1367</td>
<td>100</td>
<td>36,980</td>
<td>0</td>
</tr>
<tr>
<td>F6</td>
<td>GA-PSO</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>8254</td>
<td>0.00009</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>MORELA</td>
<td>−1</td>
<td>40,680</td>
<td>0.6722</td>
<td>100</td>
<td>40,960</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 1: Results of Different Algorithms for Various Test Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Algorithm</th>
<th>Type</th>
<th>Average Error</th>
<th>Best Function Value</th>
<th>Standard Time</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>F7</td>
<td>SZGA</td>
<td>3</td>
<td>NA</td>
<td>9000</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>ACO</td>
<td>NA</td>
<td>NA</td>
<td>0.11^a</td>
<td>NA</td>
<td>264^a</td>
<td>NA</td>
</tr>
<tr>
<td>GA-WLS</td>
<td>3</td>
<td>2573</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>GA-PSO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>25,706</td>
<td>0.00012</td>
</tr>
<tr>
<td>MORELA</td>
<td>3</td>
<td>33,920</td>
<td>0.5344</td>
<td>100</td>
<td>37,120</td>
<td>0</td>
</tr>
<tr>
<td>F8</td>
<td>SA</td>
<td>−9.9999994e−01</td>
<td>16,801</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>MORELA</td>
<td>−1</td>
<td>35,640</td>
<td>0.5365</td>
<td>100</td>
<td>38,380</td>
<td>0</td>
</tr>
<tr>
<td>F9</td>
<td>GAWLS</td>
<td>−186.7309</td>
<td>2568</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>GA-PSO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>96,211</td>
<td>0.00007</td>
</tr>
<tr>
<td>MORELA</td>
<td>−186.7309</td>
<td>16,740</td>
<td>0.3516</td>
<td>100</td>
<td>17,460</td>
<td>0</td>
</tr>
<tr>
<td>F10</td>
<td>ACORSES</td>
<td>−837.9658</td>
<td>1176</td>
<td>0.0690</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>ACO-NPU</td>
<td>−837.9658</td>
<td>750</td>
<td>0.0289</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>MORELA</td>
<td>−837.9658</td>
<td>14,880</td>
<td>0.2722</td>
<td>100</td>
<td>17,500</td>
<td>0</td>
</tr>
<tr>
<td>F11</td>
<td>PSACO</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>167</td>
<td>5.7061e−27</td>
</tr>
<tr>
<td>GA-PSO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>95</td>
<td>0.00005</td>
</tr>
<tr>
<td>MORELA</td>
<td>0</td>
<td>37,016</td>
<td>23.3041</td>
<td>100</td>
<td>37,856</td>
<td>0</td>
</tr>
<tr>
<td>F12</td>
<td>PSACO</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>592</td>
<td>2.0755e−11</td>
</tr>
<tr>
<td>ACO</td>
<td>NA</td>
<td>NA</td>
<td>0.74^a</td>
<td>NA</td>
<td>528^a</td>
<td>NA</td>
</tr>
<tr>
<td>GA-PSO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>2117</td>
<td>0.00020</td>
</tr>
<tr>
<td>MORELA</td>
<td>−3.8628^b</td>
<td>13,840</td>
<td>0.4053</td>
<td>96</td>
<td>15,700</td>
<td>1.015e−13</td>
</tr>
<tr>
<td>F13</td>
<td>PSACO</td>
<td>NA</td>
<td>NA</td>
<td>96</td>
<td>529</td>
<td>4.4789e−11</td>
</tr>
<tr>
<td>GA-PSO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>12,568</td>
<td>0.00024</td>
</tr>
<tr>
<td>ACO</td>
<td>NA</td>
<td>NA</td>
<td>4.10^a</td>
<td>NA</td>
<td>1344^b</td>
<td>NA</td>
</tr>
<tr>
<td>MORELA</td>
<td>−3.32^c</td>
<td>30,400</td>
<td>0.7804</td>
<td>96</td>
<td>32,200</td>
<td>2.3413e−16</td>
</tr>
<tr>
<td>F14</td>
<td>RW-PSO+BOF</td>
<td>NA</td>
<td>NA</td>
<td>96</td>
<td>529</td>
<td>4.4789e−11</td>
</tr>
<tr>
<td>MORELA</td>
<td>0</td>
<td>33,500</td>
<td>0.5254</td>
<td>100</td>
<td>34,440</td>
<td>0.00024</td>
</tr>
<tr>
<td>F15</td>
<td>PSACO</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>1081</td>
<td>6.23e−22</td>
</tr>
<tr>
<td>MORELA</td>
<td>0</td>
<td>37,010</td>
<td>19.8856</td>
<td>100</td>
<td>38,896</td>
<td>0</td>
</tr>
<tr>
<td>F16</td>
<td>PSACO</td>
<td>NA</td>
<td>NA</td>
<td>100</td>
<td>209</td>
<td>2.6185e−13</td>
</tr>
<tr>
<td>MORELA</td>
<td>0.3979^b</td>
<td>7680</td>
<td>0.1413</td>
<td>100</td>
<td>7900</td>
<td>0</td>
</tr>
</tbody>
</table>

NA: Not available; *The average number of function evaluations of four runs and running time in units of standard time.  
^aThe theoretical minimum value was considered to be four digits.  
^bThe theoretical minimum value was considered to be two digits.  
^cMean results of more than 30 independent trials.

The findings indicate that the MORELA showed remarkable performance for all of the test functions except F3, for which theoretical global optimum could not be found with the required accuracy, namely, 0. Although MORELA was not
able to solve this function, it produces better functional value than those provided by other compared algorithms, as shown in Table 4. MORELA also produces less average error for all test functions than the other methods considered. At the same time, most of these errors are equal to 0. This means that MORELA was able to find the global optimum for each run. Therefore, MORELA may be considered to be a robust algorithm for finding global optimum for any given mathematical function. As Table 4 shows, MORELA requires a greater number of learning episodes than the other algorithms for many of the test functions, due to accuracy required of it. Although the required number of learning episodes was found to be higher than the other algorithms, it can be ignored because of the required accuracy chosen (a value of 0) for all test functions considered in this comparison. The best solution times for the functions achieved by the algorithms considered are also given in Table 4. In the meantime, investigation of the effect of environment size and corresponding algorithm parameters is beyond the scope of this study.

3.4. Explanation of Evolving Strategy Provided by Sub-Environment

In addition, we have used Bohachevsky function (F4) in order to better explain how the evolving population is diversified by the sub-environment in MORELA. The function of F4 has a global optimum solution of \( f(x_1, x_2) = 0 \) when \((x_1, x_2) = 0\) for the case of 2 variables and the required accuracy was chosen to be 1e−15 units for this experiment. As it can be realized from Figure 5(a) and Figure 5(b), the sub-environment is firstly generated at 2nd learning episode using Equation (3), depending on the best solution found in the previous learning episode and \( \beta \) value.

When the main difference between Figure 5(a) and Figure 5(b) is observed, it can be clearly seen that the solution points are stably distributed in the original environment whereas the points in the sub-environment explore global optimum near the boundaries of the solution space given as \(-100 \leq x_1, x_2 \leq 100\) for this function. Although the solution points located in the original environment have a tendency to reach global optimum at the 10th learning episode as shown in Figure 5(c) and Figure 5(d), the others located in the sub-environment still continue to search global optimum near the boundaries of the solution space. This property provides to diversify the population of MORELA at each learning episode, and thus the probability of being trapped in local optimum is decreased. At 25th learning episode, the solution points in the original environment are almost close to global optimum, but the other points in the sub-environment are still dispersed in the solution space as can be seen in Figure 5(e) and Figure 5(f). Similarly, this tendency continues until MORELA reached to about 200th learning episode. After 250th learning episode, the solution points in the original environment are too close to global optimum whereas the others in the sub-environment still continues to explore new solution points around global optimum although they have a tendency to reach global optimum. Finally, at
1000th learning episode, all solution points populated in the original and sub-environment reached to global optimum as seen in Figure 5(k) and Figure 5(l).

### 3.5. Effect of High Dimensions

The Ackley function given in Equation (7) was chosen to explore the effect of high dimensions on the search capability of MORELA. The global minimum of this function is given as \( f(x) = 0 \) at \( x = 0 \). The algorithm was repeated 10 times to decrease the effect of randomness. The average number of learning episodes and average objective function values were recorded with different dimensions. For this experiment, the required accuracy was chosen to be \( 10^{-10} \) units.

\[
AK_{(\cdot)}(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{j=1}^{n} x_j^2} \right) - \exp \left( \frac{1}{n} \sum_{j=1}^{n} \cos (2\pi x_j) \right) + 20 + e
\]  

- \(-32.768 \leq x_i \leq 32.768; \ i = 1, 2, \ldots, n; \ n = \) the number of the variables
- \( x = (0, \ldots, 0), \ AK_{(\cdot)}(x) = 0 \)

**Figure 6** represents the variation of the average objective function value according to different dimensions.

As **Figure 6** shows, the MORELA shows acceptable performance even for high dimensions of the Ackley function. Average objective function values of
9.74e−11 and 9.89e−11 were obtained when the dimensions were equal to 10 and 10,000, respectively. These values are very close to each other, considering differences in dimensions. Results show that MORELA may be considered as an efficient way for finding global optimum based on required accuracy of a given mathematical function even if dimension of the problem became increased.

Figure 7 illustrates the variation in the average number of learning episodes as a function of dimension for the Ackley function.

Although the average number of learning episodes increased notably with increasing dimension up to 1000, the average number of learning episodes increased very little within the dimension range from 1000 to 10,000. This experiment clearly demonstrates that the number of learning episodes required by MORELA is not apparently affected by high dimensionality.

4. Conclusions

A powerful and robust algorithm called MORELA is proposed to find global opt-
timum for any given mathematical function. MORELA differs from RL based approaches by means of generating a sub-environment. Thus, this approach makes it possible to find global optimum, because it is sought both around the best solution achieved so far with the assistance of the sub-environment and the original environment.

The performance of MORELA was examined in several experiments, namely, a comparison of MORELA and RL, a robustness analysis, comparisons with other methods, explanation of evolving strategy of MORELA, and an investigation of the effect of high dimensionality. The comparison of MORELA and RL showed that MORELA requires many fewer learning episodes than RL to find global optimum for a given function. The robustness analysis revealed that MORELA is able to find global optimum with high success. MORELA was also tested on sixteen different test functions that are difficult to optimize, and its performance was compared with that of other available methods. MORELA has found global optimum for all of the test functions except F3, based on the required accuracy. Besides, the last experiment clearly shows that MORELA is not significantly affected by high dimensionality.

Finally, all numerical experiments indicate that MORELA performed well in finding global optimum of mathematical functions considered, compared to other methods. Based on the results of this study, it is expected that in future research, optimization methods based on RL will be found to possess great potential for solving various optimization problems.

References


ment Learning Algorithm for Solving Coordinated Signalized Networks. Transportation Research Part C, 54, 40-55.


Appendix A

**F1:**  
*Rosenbrock (2 variables)*

\[
f(x_1, x_2) = \left(100 * (x_1 - x_2^2)^2 \right) + (1 - x_1)^2
\]

- global minimum: \((x_1, x_2) = 1, f(x_1, x_2) = 0\)

**F2:**  
(2 variables)

\[
f(x_1, x_2) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)
\]

- \(-1 \leq x_1, x_2 \leq 1\)
- global minimum: \((x_1, x_2) = (0, 0), f(x_1, x_2) = -2\)

**F3:**  
(2 variables)

\[
f(x_1, x_2) = \frac{(x_1 - 3)^4}{1 + (x_1 - 3)^4} + \frac{(x_2 - 3)^4}{1 + (x_2 - 3)^4}
\]

- global minimum: \((x_1, x_2) = 3, f(x_1, x_2) = 0\)

**F4:**  
*Bohachevsky (2 variables)*

\[
f(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7
\]

- \(-100 \leq x_1, x_2 \leq 100\)
- global minimum: \((x_1, x_2) = 0, f(x_1, x_2) = 0\)

**F5:**  
*De Jong (3 variables)*

\[
f(x) = \sum_{i=1}^{n} x_i^2
\]

- \(-5.12 \leq x_i \leq 5.12; i = 1, 2, \ldots, n\)
- global minimum: \(x = (0, 0, \ldots, 0), f(x) = 0\)

**F6:**  
*Easom (2 variables)*

\[
f(x_1, x_2) = -\cos(x_1) \cos(x_2) \exp\left(- (x_1 - \pi)^2 - (x_2 - \pi)^2\right)
\]

- \(-100 \leq x_1, x_2 \leq 100\)
- global minimum: \((x_1, x_2) = (\pi, \pi), f(x_1, x_2) = -1\)

**F7:**  
*Goldstein-Price (2 variables)*

\[
f(x, y) = \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]
\]

\[
\times \left[1 + (2x_1 - 3x_2)^2 \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]
\]

- \(-2 \leq x_1, x_2 \leq 2\)
- global minimum: \((x_1, x_2) = (0, -1), f(x_1, x_2) = 3\)
F8: Drop wave (2 variables)

\[ f(x_1, x_2) = -\cos\left(12\sqrt{x_1^2 + x_2^2}\right) \]

- \(-5.12 \leq x_1, x_2 \leq 5.12\)
- global minimum: \((x_1, x_2) = (0, 0), f(x_1, x_2) = -1\)

F9: Shubert (2 variables)

\[ f(x_1, x_2) = \sum_{i=1}^{n} x_i \cdot \cos((i+1)x_i + 1) \sum_{i=1}^{n} i \cdot \cos((i+1)x_i + 1) \]

- \(-10 \leq x_1, x_2 \leq 10\)
- 18 global minima \(f(x_1, x_2) = -186.7309\)

F10: Schwefel (2 variables)

\[ f(x) = \sum_{i=1}^{n} -x_i \sin\sqrt{|x_i|} \]

- \(-500 \leq x_1, x_2 \leq 500\)
- global minimum: \((x_1, x_2) = (420.9687, 420.9687), f(x_1, x_2) = -n \cdot 418.9829\)

F11: Zakharov (2 variables)

\[ f(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i^2\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i^2\right)^4 \]

- \(-5 \leq x_i \leq 10, i = 1, 2, \ldots, n\)
- global minimum: \(x = (0, 0, \ldots, 0), f(x) = 0\)

F12: Hartman (3 variables)

\[ f(x) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{3} \alpha_{ij} (x_j - p_{ij})^2\right] \]

- \(0 \leq x_j \leq 1, j = 1, 2, \ldots, 3\)
- global minimum: \(x = (0.11, 0.555, 0.855), f(x) = -3.8628\)

\[
\begin{array}{cccccccc}
  l & \alpha_{11} & \alpha_{12} & \alpha_{21} & \alpha_{22} & \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} & \alpha_{41} \\
 1 & 3 & 10 & 30 & 1 & 0.3689 & 0.1170 & 0.2673 & \\
 2 & 0.1 & 10 & 35 & 1.2 & 0.4699 & 0.4387 & 0.7470 & \\
 3 & 3 & 10 & 30 & 3 & 0.1091 & 0.8742 & 0.5547 & \\
 4 & 0.1 & 10 & 35 & 3.2 & 0.03815 & 0.5743 & 0.8828 & \\
\end{array}
\]

F13: Hartman (6 variables)
\[ f(x) = -\sum_{i=1}^{4} c_i \exp \left[ -\sum_{j=1}^{6} \alpha_{ij} (x_j - p_{ij})^2 \right] \]

- \( 0 \leq x_j \leq 1, j = 1, 2, \ldots, 6 \)
- global minimum: \( x = (0.201, 0.150, 0.477, 0.275, 0.311, 0.657), f(x) = -3.32 \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
<th>( \alpha_4 )</th>
<th>( \alpha_5 )</th>
<th>( \alpha_6 )</th>
<th>( c_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>3</td>
<td>17</td>
<td>3.5</td>
<td>1.7</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.05</td>
<td>10</td>
<td>17</td>
<td>0.1</td>
<td>8</td>
<td>14</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3.5</td>
<td>1.7</td>
<td>10</td>
<td>17</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>8</td>
<td>0.05</td>
<td>10</td>
<td>0.1</td>
<td>14</td>
<td>3.2</td>
</tr>
</tbody>
</table>

**F14:**

Rastrigin (2 variables)

\[ f(x) = 10n + \sum_{i=1}^{n} x_i^2 - 10 \cos(2\pi x_i) \]

- \(-5.12 \leq x_i \leq 5.12, i = 1, 2, \ldots, n\)
- global minimum: \((x_1, x_2) = (0, 0), f(x) = 0\)

**F15:**

Griewank (8 variables)

\[ f(x) = \sum_{i=1}^{n} x_i^2 / 4000 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 \]

- \(-300 \leq x_i \leq 600, i = 1, 2, \ldots, n\)
- global minimum: \( x = (0, \ldots, 0), f(x) = 0\)

**F16:**

Branin (2 variables)

\[ f(x) = \left( x_2 - \frac{5.1}{4\pi} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos(x_1) + 10 \]

- \(-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15\)
- three global minimum: \((-\pi, 12.275), (\pi, 2.275), (3\pi, 2.475), f(x) = \frac{5}{4\pi}\)
Submit or recommend next manuscript to SCIRP and we will provide best service for you:

- Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.
- A wide selection of journals (inclusive of 9 subjects, more than 200 journals)
- Providing 24-hour high-quality service
- User-friendly online submission system
- Fair and swift peer-review system
- Efficient typesetting and proofreading procedure
- Display of the result of downloads and visits, as well as the number of cited articles
- Maximum dissemination of your research work

Submit your manuscript at: http://papersubmission.scirp.org/
Or contact ojop@scirp.org