ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

A Deep Double Q-Learning as a SDLS support in solving LABS problem

Dominik Żurek and Marcin Pietroń and Kamil Pietak and Kamil Faber *

AGH University of Krakow
al. Adama Mickiewicza 30, 30-059 Krakow, Poland

Abstract. Low Autocorrelation Binary Sequence (LABS) remains an open complex optimization problem with multiple applications. Existing studies rely primarily on advanced solvers based on local search heuristics, such as the steepest-descent local search algorithm (SDLS), Tabu search, or xLastovka algorithms. These approaches require searching through a large solution space, which is a computationally heavy and time-consuming process, leading to slower convergence. To improve convergence speed and allow for finding better solutions within a limited time, we propose the Deep Double Q-learning reinforcement learning algorithm for the LABS problem to support heuristic methods. The model aims to narrow down the search space without causing a drop in the final efficiency. Our experimental study showcases that the proposed approach is a promising direction for developing a highly efficient method for the LABS problem.

1 Introduction

LABS is one of the hard discrete problems which, despite wide research, remains an open optimization problem for long sequences. It also has a wide range of applications including communication engineering [13, 14], statistical mechanics [2, 11] and mathematics [7, 10].

The objective of LABS problem [5] is to find a binary sequence S with length L, where $s_i \in \{-1, 1\}$, which minimizes the energy function $E(S)$. The problem can be also defined as maximizing the *merit factor* $F(S)$ [6], which binds LABS energy level to the length of a given sequence:

$$C_k(S) = \sum_{i=0}^{L-k-1} s_i s_{i+k} \quad \text{and} \quad E(S) = \sum_{k=1}^{L-1} C_k^2(S) \quad \text{and} \quad F(S) = \frac{L^2}{2E(S)} \qquad (1)$$

Prevalent methods of solving LABS are heuristic algorithms that utilize plausible rules to locate satisfactory sequences more quickly. A well-known method is *Steepest Descend Local Search* (SDLS) [1], which is also very effective on GPGPU architectures [15, 12]. The authors of [16] proposed two new variants of the SDLS algorithm that extend the neighborhood of the sequence to a 2-bit and recurrent exploration of sequences at the 1 and 2-bit distance. Another popular approach is the Tabu search [4], which also achieves good results using the

*{dzurek, pietron, kpietak, kfaber}@agh.edu.pl. This paper was realized with funds of Polish Ministry of Science and Higher Education assigned to AGH University and it was supported by PLGrid Infrastructure.

GPGPU architecture [12]. Moreover, it is possible to leverage the neural network to improve the TABU parameters [17]. The machine learning techniques used are standard models or deep learning models that support standard optimization algorithms at a selected stage. There are also end-to-end solutions that completely replace standard optimization algorithms (graph neural networks and reinforcement learning) [3].

However, we can identify the following gaps: i) very scarce use of modern reinforcement learning (RL) techniques in hard discrete problems; ii) few methods that apply hybrid approach with deep reinforcement learning and local search algorithms. In this paper, we attempt to fill these gaps by proposing a method for solving challenging discrete problems by combining local optimization heuristics and Deep Reinforcement Learning (DRL) that minimizes the searching space based on the current computing context. We experimentally verify this concept on the low autocorrelation binary sequence problem (LABS) with SDLS search algorithm. In summary, we aim to answer the following research questions:

- **RQ1:** Can DRL improve the heuristic algorithm to achieve better results in the LABS problem?
- **RQ2:** Can DRL deal with the LABS's problem of any size?

2 Methodology

2.1 Environment, State, action and reward definition

We create an artificial environment simulating the SDLS algorithm resolving the LABS's problem. The main elements of the proposed environment are the input sequence S with length L, energy E and the best current sequence S_{best} for which the best current energy E_{best} was calculated. The environment was created in the *gym* library. In our proposed environment, the state s_t represents the input sequence S. Based on s_t the agent takes an action a_t that determines the index of value in S that should be changed to achieve the best possible energy E_i in a long-term.

RL is a framework for sequential decision-making. Therefore, immediate reward and long-term rewards are provided to the model. The model promotes maximizing the *merit factor* value (eq. 1) which, in our case, represents immediate reward. At the same time, the model is severely penalized by a -1 value when taking an action increases the energy value. We leverage the Bellman equation to calculate the reward after taking action t, as follows:

$$r_t = \begin{cases} -1 & \text{if } E_i > E_{i-1} \\ \sum_{i=N}^{i=t+1} \frac{1}{\gamma^t \cdot E_i} & \text{otherwise} \end{cases} \tag{2}$$

2.2 Double Deep Q-Learning Algorithm

In order to solve LABS using RL, we decided to use deep double Q-learning (DDQN) [9] because it leverages an off-policy approach capable of learning a

Algorithm 1 Deep Double Q-Learning for resolving LABS problem

Input: λ – learning rate, γ – discount factor, ϵ – small number > 0, N – Iterations number, J – Environment steps per iteration

```
1: Initialize primary network $Q_\Theta$ and target network $Q'_\Theta$.
2: Initialize replay buffer $D$
3: for $i \in \{0, 1, \ldots, N\}$ do
4:     Initialize state $s_0$
5:     for $j \in \{0, 1, \ldots, J\}$ do
6:         Following $\epsilon$-greedy policy select action $a_t$ (index of bit to change) with
           probability $\epsilon$. Otherwise observe current state $s_t$ (input sequence) and
           select action $a_t = max_a Q^A(s_t, a_t)$
7:         Execute $a_t$, observe next state $s_{t+1}$ and reward $r_t$ (calculated as F(S)
           from eq. 1 or set as -1 (see sec. 2.1))
8:         Store $(s_t, a_t, r_t, s_{t+1})$ in replay buffer $D$
9:     end for
10:    for update step do
11:        Sample $e = (s_t, a_t, r_t, s_{t+1})$ from $D$
12:        Compute target Q-value:
             $Q^t = r_t + \gamma Q_\Theta(s_{t+1}, argmax_{a'} Q_{\Theta'}(s_{t+1}, a'))$
13:        Perform optimizer step on $Q^t(s_t, a_t) - Q_\Theta(s_t, a))^2$
14:        Update target network parameters:
             $\Theta' \leftarrow \lambda * \Theta + (1 - \lambda) * \Theta'$
15:    end for
16: end for
```

general policy that can adapt to different situations. DDQN is a combination of a value estimation method Double Q-Learning [8] and a deep neural network (DNN). The algorithm consists of the exploration and exploitation phase. In each episode, the algorithm takes an action by sampling from a set of possible actions (in the exploration step) or choosing the best one from the current policy (in the exploitation step).

Algorithm 1 shows the DDQN process. It starts with the initialization of the primary network, target network, and replay buffer D (lines 1-2). Then, the algorithm executes N episodes. Each episode starts with state initialization as a randomly selected sequence s_0 with length L (line 4). Then, in every J step, the algorithm chooses the action following the ϵ-greedy policy (line 6). Each action changes a single bit in q sequence s_t, and is determined either as a random action (exploration) or the best action observed so far (exploitation). Then, the algorithm executes the action, observing the next state s_{t+1} and receiving a reward r_t (line 7). The action, rewards, and states are stored in the replay buffer (line 8) and used in the model training process (lines 10-15).

In inference mode, the trained model returns the best possible action for the given state until no future improvement can be observed (after some K step (see section 2.3)) the DDQN model is stuck in the local minimum).

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

2.3 The concept of hybrid platform

We also propose a Hybrid method utilizing both deep Q-learning and the SDLS method. The process starts with the RL model, looking for the best solution. After RL converges and does not yield any more improvements with additional iterations, its output sequence is passed to the traditional SDLS algorithm, which continues until no further improvements can be attained. This way, the RL narrows down the search space for the SDLS algorithm. The advantage of the hybrid approach is that DDQN can identify better sequences without analyzing all possible L solutions in every iteration. Consequently, the process converges faster through the initial steps and yields better results.

3 Experiments and conclusions

Before carrying out the final experiments, we performed preliminary experiments to set up the input parameters and scenarios. As a result, the experimental setup is as follows: i) Neural Network consists of three convolutional and three fully connected layers; ii) λ = 1e-4; iii) batch size is set to 128; iv) ϵ decays exponentially with start value equal to 0,9; v) number of environment steps per iteration equals 1024; vi) experiments are carried out for four sequence lengths: {64, 96, 128, 256} and for these values, the training time has been set appropriately {48h, 72h, 100h, 150h}. Moreover, the methods have limited time in which they need to find the most optimal solution (inference time) – 10 minutes.

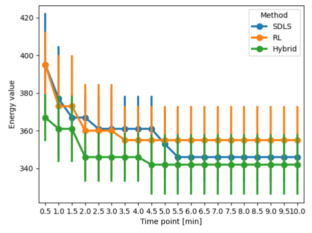
To answer our research questions from Section 1, we compare the three algorithms: i) traditional SDLS; ii) RL model; iii) hybrid of the RL model and traditional SDLS. We conduct experiments for four lengths of randomly generated input sequences (64, 96, 128, 256). Table 1 showcases the average and the highest results from 5 independent executions for every algorithm, where each algorithm had 10 minutes to find a solution. As we can observe, the RL model can handle the LABS problem with reasonable results close to the traditional SDLS algorithm for all considered lengths of input sequences (**RQ2**).

Another point worth analyzing is the hybrid of the RL and SDLS. The hybrid model achieves the best results (the lowest energy values) for all sequence lengths in the best and hybrid modes. The results showcase that RL can support heuristic algorithms such as SDLS to improve the quality of the solution (**RQ1**). Moreover, Figure 1 demonstrates how the results change over testing time. We observe that the RL model is more effective than the SDLS algorithm through the first 5.5 min ($L = \{64, 96\}$), 4.5 min ($L = 256$), and 8 min ($L = 128$).

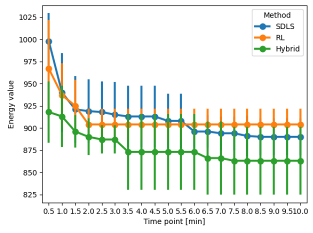
Table 1: Min and average of best value (5 independent runs; 10 minutes)

Sequence length	**SDLS**		**RL**		**HYBRID**	
	Minimum	**Average**	**Minimum**	**Average**	**Minimum**	**Average**
64	324	390	348	398	320	375
96	868	970	868	970	748	930
128	1600	1804	1668	1810	1564	1744
256	7228	7807	7108	7887	6812	7541

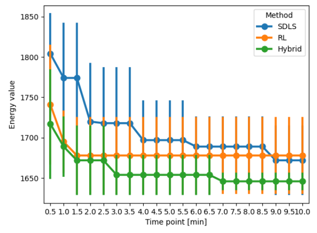
ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2.
Available from http://www.i6doc.com/en/.



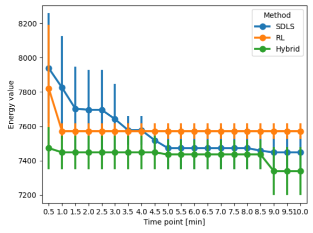
(a) Energy of LABS $L = 64$



(b) Energy of LABS $L = 96$



(c) Energy of LABS $L = 128$



(d) Energy of LABS $L = 256$

Fig. 1: Average best energies achieved by SDLS, RL and Hybrid during 5 independent runs through ten minutes

4 Summary and Future works

This paper introduces deep double Q-learning as a method that can successfully be applied to the LABS problem. Moreover, we showcase that combining DRL and local optimization heuristics yields better results than heuristics. These results create a foundation for a new promising direction in finding solutions to NP-hard problems by leveraging deep RL and hybrid methods. In future works, we will extend the presented approach with more algorithms dedicated to the LABS problems, such as Tabu search or self-avoiding walking.

References

[1] Bartholomew-Biggs, M.: The Steepest Descent Method, pp. 1–8. Springer US, Boston, MA (2008)

[2] Bernasconi, J.: Low autocorrelation binary sequences : statistical mechanics and configuration space analysis. Journal De Physique **48**, 559–567 (1987)

[3] Cappart, Q., Chételat, D., Khalil, E.B., Lodi, A., Morris, C., Velickovic, P.: Combinatorial optimization and reasoning with graph neural networks. J. Mach. Learn. Res. **24**, 1–61 (2023)

[4] Gallardo, J.E., Cotta, C., Fernández, A.J.: Finding low autocorrelation binary sequences with memetic algorithms. Appl. Soft Comput. **9**(4), 1252–1262 (Sep 2009)

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

[5] Golay, M.: Sieves for low autocorrelation binary sequences. IEEE Transactions on Information Theory **23**(1), 43–51 (1 1977)

[6] Golay, M.: The merit factor of long low autocorrelation binary sequences. IEEE Transactions on Information Theory **28**(3), 543–549 (5 1982)

[7] Günther, C., Schmidt, K.U.: Merit factors of polynomials derived from difference sets (2016)

[8] Hasselt, H.: Double q-learning. In: Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A. (eds.) Advances in Neural Information Processing Systems. vol. 23. Curran Associates, Inc. (2010)

[9] Hasselt, H.v., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of AAAI Conference. p. 2094–2100. AAAI Press (2016)

[10] Jedwab, J., Katz, D.J., Schmidt, K.U.: Advances in the merit factor problem for binary sequences. Journal of Combinatorial Theory **120**(4), 882–906 (2013)

[11] Leukhin, A.N., Potekhin, E.N.: A bernasconi model for constructing ground-state spin systems and optimal binary sequences. Journal of Physics: Conference Series **613**, 012006 (may 2015)

[12] Piętak, K., Żurek, D., Pietroń, M., Dymara, A., Kisiel-Dorohinicki, M.: Striving for performance of discrete optimisation via memetic agent-based systems in a hybrid cpu/gpu environment. Journal of Computational Science **31**, 151 – 162 (2019)

[13] Zeng, F., He, X., Zhang, Z., Xuan, G., Peng, Y., Yan, L.: Optimal and z-optimal type-ii odd-length binary z-complementary pairs. IEEE Communications Letters **24**(6), 1163–1167 (June 2020)

[14] Zhao, L., Song, J., Babu, P., Palomar, D.P.: A unified framework for low autocorrelation sequence design via majorization–minimization. IEEE Transactions on Signal Processing **65**(2), 438–453 (Jan 2017)

[15] Żurek, D., Piętak, K., Pietroń, M., Kisiel-Dorohinicki, M.: Toward hybrid platform for evolutionary computations of hard discrete problems. Procedia Computer Science **108**, 877–886 (2017)

[16] Żurek, D., Piętak, K., Pietroń, M., Kisiel-Dorohinicki, M.: New variants of sdls algorithm for labs problem dedicated to gpgpu architectures. In: International Conference on Computational Science. pp. 206–212. Springer (2021)

[17] Żurek, D., Pietroń, M., Piętak, K., Kisiel-Dorohinicki, M.: A deep neural network as a tabu support in solving labs problem. In: International Conference on Computational Science. pp. 237–243. Springer (2022)