A deep neural network as a TABU support in solving LABS problem

Dominik Żurek, Marcin Pietroń, Kamil Piętak, Marek Kisiel-Dorohinicki

AGH University of Science and Technology
al. Adama Mickiewicza 30, 30-059 Krakow, Poland
{dzurek, pietron, kpietak, doroh}@agh.edu.pl

Abstract. One of the leading approaches for solving various hard discrete problems is designing advanced solvers based on local search heuristics. This observation is also relevant to the low autocorrelation binary sequence (LABS) – an open hard optimisation problem that has many applications. There are a lot of dedicated heuristics such as the steepest-descent local search algorithm (SDLS), Tabu search or xLostovka algorithms. This paper introduce a new concept of combining well-known solvers with neural networks that improve the solvers' parameters based on the local context. The contribution proposes the extension of Tabu search (one of the well-known optimisation heuristics) with the LSTM neural network to optimise the number of iterations for which particular bits are blocked. Regarding the presented results, it should be concluded that the proposed approach is a very promising direction for developing highly efficient heuristics for LABS problem.

Keywords: LABS, TABU, Neural network, LSTM

1 Introduction

This paper concentrates on solving hard discrete problems using a combination of local optimisation heuristics and neural networks that improves the chosen parameters of heuristics based on the current computation context. The concept is verified on the low autocorrelation binary sequence problem (LABS) and the Tabu search algorithm.

LABS [10] consists of finding a binary sequence $S = \{s_0, s_1, ..., s_{L-1}\}$ with length L, where $s_i \in \{-1, 1\}$, which minimises energy function $E(S)$:

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

There may varied techniques that try to solve the problem. The simplest method of solving LABS is the application of exhaustive enumeration; this provides the best results, but can be applied only to small values of L. There are also a lot of various heuristic algorithms that use some plausible rules to locate good

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_32

sequences more quickly. A well-known method for such techniques is *steepest descend local search* (SDLS) [1] or Tabu search [8]. In recent years, a few modern solvers based on the 'self-avoiding walk' concept have been proposed. The most promising solvers are *lssOrel* [3] and *xLostavka* [4], which are successfully used for finding skew-symmetric sequences of lengths between 301 and 401 [5]. These techniques can also be parallelised utilising GPGPU architectures as has been shown in the literature [12, 15].

A different direction of research is the application of agent-based biologically-inspired computational systems. One of such meta-heuristics approach is the concept of an evolutionary multi-agent system successfully applied for solving complex continues and discrete optimisation problems such as LABS [12, 15].

In the presented work, a deep learning network is incorporated for estimating optimal Tabu search parameters. The LSTM architecture was chosen as a base model in different optimisation domains such as sequence-to-sequence learning [13] or as a foundation of the pointer networks which are used to resolve the TSP problem. In order to resolve this NP-hard problem, *Vinyals et al.* [14] proposed training the recurrent neural network in a supervised manner to anticipate the order of visited cities. This approach was extended in research [2] in which the optimisation of the recurrent neural network which is resolving the TSP problem was gained by using a policy gradient method. Those researches have become the motivation for us to introduce the recurrent neural network as a support for Tabu search algorithm in resolving LABS problem.

The paper is organised as follows. In the next section, the Tabu search is presented as a starting point for further extensions. In the third section, a new concept of Tabu search with LSTM extensions (for predicting the M parameter) are described in detail. Then, the experimental results are presented and conclusions are drawn in the following sections. The paper is summarised in the last section where future work is also suggested.

2 Tabu search for LABS problem

The Tabu search is a well-known heuristic [9], adjusted to the LABS problem [6, 7]. The Tabu method can be seen as an extension of steepest descent local search with so-called banned states. In a similar manner to SDLS, Tabu explores the neighbourhood of a given sequence (i.e. all sequences with one bit changed) and also introduces a Tabu array, with the same length as a LABS sequence, which contains blocked indices. When a new better solution is found in an iteration, the index which led to this sequence is placed in the Tabu array and is therefore banned from further changes (in M following iterations). The details of the Tabu search are presented in Algorithm 1.

In contrast to the SDLS algorithm, applying the Tabu mechanism with blocked states helps to escape the attraction basins of a local minimum. The consecutive iterations don't have to produce sequences with lower energy if the Tabu array contains the best sequences from the current neighbourhood. This fact leads to proper directions in LABS solutions space exploration [11].

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_32

Algorithm 1 The Tabu search algorithm for the LABS problem (based on [7]). Symbols: S – input sequence; L – length of the sequence; $maxIters$ – number of iterations; E – sequence evaluation; $tabu$ – integer vector that describes how long sequence elements are blocked; $minTabu$, $extraTabu$ – auxiliary numbers used to set values in vector $tabu$; M – number of iterations for which a chosen bit should be blocked; $changedBit$ – an index of element (a bit) that has been changed.

function TABUSEARCH($S, maxIters$)
  $int[\,]\ tabu$ ▷ integer vector initialised with zeros
  $minTabu = maxIters/10$
  $extraTabu = maxIters/50$
  $S_{start} = S_{best} = S$
  $E_{best} = \text{EVALUATE}(S_{best})$
  for $i = 0$ to $maxIters - 1$ **do**
    $E_i = \infty$
    for $j = 0$ to $L - 1$ **do**
      $S_{tmp} = S_{start}$
      $S_{tmp}[j] = -1 * S_{tmp}[j]$
      $E_{tmp} = \text{EVALUATE}(S_{tmp})$
      if $i \geq tabu[j]$ **or** $E_{tmp} < E_{best}$ **then**
        if $E_{tmp} < E_i$ **then**
          $S_i = S_{tmp}$, $E_i = E_{tmp}$
          $changedBit = j$
    if $S_i \neq null$ **then** ▷ if better, non-blocked sequence is found
      $S_{start} = S_i$ ▷ set the current sequence as the starting point
      $M = minTabu + \text{RAND}(0, extraTabu)$
      $tabu[changedBit] = i + M$
    if $E_i > E_{best}$ **then**
      $S_{best} = S_i$, $E_{best} = E_i$
  return S_{best}

3 Improving Tabu search with LSTM

LSTM is an example of artificial recurrent neural network architecture. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points, but also entire sequences of data.

The main goal of this paper is to device a more effective technique to determine the value of the M parameter, the use of which, enables finding a more optimal value of the energy. For this purpose we introduce the LSTM neural network which is able to predict the most effective value of this parameter for a particular input sequence S with the length L and corresponding to this sequence energy E (see next subsection). For each different value of the parameter L there is a separate trained model.

Prepare training data. In order for it to be possible to train the neural network, for each value of the L parameter used in this paper, 1048576 (2^{20})

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_32

sequences were randomly generated and evaluated. The number of generated sequences represents X%, Y%, Z% percent of the total solution space, respectively for lengths 128, 192, and 256 (based on the formula $\frac{2^{20}}{2^L}$). Each generated sequence has received its own random value of the M parameter from the range [2;18] (adopted based on max iteration, see Section 4). Each tuple of sequence S and parameter M are put as the input to the module which is searching optimal sequence S' which means with a minimum value of energy E. This searching is realised through the use of very effective GPGPU implementation of Tabu search which was proposed in previous work [12]. Consequently, there is 2^{20} S, M, S' and energy E which was calculated based on sequence S' and its value of parameter M. Finally, this value is the corresponding energy for the sequence S. In order to teach the neural network to predict the optimal value of parameter M for any sequence, as an input to neural network there is placed a pair sequence S and energy E (the S' sequence is not used in the next phases).

Finding optimal parameters. The optimal value of parameters which were used in the training phase and to build neural network architecture, were determined empirically. Through the use of a very small part of the generated data, we trained a separate model for each L with a different number of hidden units (#HU), LSTM layers (#LL) and the value of learning-rate parameter (lr). The best results were gained for lr equal to 0.0001, #HU equals 50 and #LL equals 2. The model was trained over seventy epochs with use of the MSE as loss function and the Adam optimiser. For the best gained results, we built the optimal architecture of LSTM which then was used in real training and during the test phase.

Training process. During the training phase (Fig. 1), as an input to the neural network, the sequence with the length L and normalised value of energy calculated on the basis of this sequence is given (energies are normalised to the range : [-1;1], through the usage the *MinMaxScaler* from *scikit-learn* library[1]). In order to enhance the meaning of the energy, this value is copied to the input vector L times (without this repetition, the energy value would be imperceptible by LSTM model). During the training, the neural network is taught to extract dependencies between the arranging binary data in the input sequences S, corresponding to this sequence energy E and the value of the M parameter which is returned as an output of the neural network. The model is trained through 70 epochs to reduce the difference between the output value of the LSTM M' and the original value of this parameter M, which was used to calculate the value of the corresponding energy in preparing the training data phase. The training process was performed on the Nvidia Tesla V100-SXM2-32GB[2].This process was run once for each value of the L parameter and this took around ten hours.

[1] https://scikit-learn.org/
[2] https://www.nvidia.com/en-us/data-center/v100/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_32

Test process. The optimal solution of the LABS problem should have the minimum possible value of energy. Thereby, during the test phase, for each randomly generated test sequence, in place of real value of the energy (as occurred in the training phase), we put the smallest possible value of the energy that can be achieved, which after normalisation is equal to -1. As was the case in the training phase, this value of energy is copied L times to the input vector. As a result, the neural network returns value of the *M'*, which allows obtaining the smallest possible value of the energy for an given input sequence. The input sequence with predicted *M'* value is given as a input for the Tabu calculation module and the final energy is returned (Fig. 1).

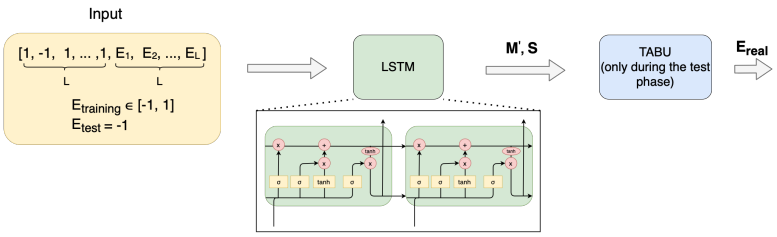


Fig. 1: Training and test phase of TABU with LSTM

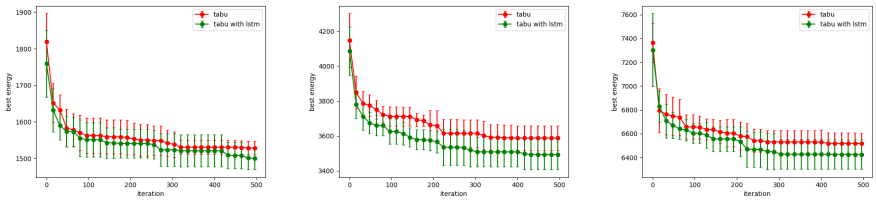
4 Effectiveness of the proposed algorithms

In order to measure the effectiveness of the proposed solution, energies from the version of Tabu with a trained LSTM neural network, are compared with the basic version of the Tabu search which was implemented in the Python language. In the original Tabu, the number of banned steps M, is determined as a sum of two factors: *minTabu* and *extraTabu* (Alg. 1, line 19). In our experiments, the value of the *maxIter* parameter was set based on promising results which was achieved in our previous work [12] and it is set to 128. Consequently, the value of the M parameters in the reference version of Tabu search is between $[13, 16]$.

Each algorithm was seeking the optimal solution for three different input lengths (128, 192, 256). In a single test, 512 random sequences are generated for which the Tabu calculations are run in both scenarios: i) base Tabu search, ii) tabu search with support of LSTM, where M parameter is prompted though the use of the neural network. The entire process was run 10 times, so consequently, 512x10 random sequences were tested for each value of parameter L.

Figure 2 demonstrates the mean value of energies from 10 independent runs after each iteration. As could be observed for each size of the problem, the best solution was obtained by the Tabu which was supported by the neural network. Moreover, with this solution for all test data, it was possible to find the lowest

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_32



(a) $L = 128$ (b) $L = 192$ (c) $L = 256$

Fig. 2: Energies achieved by basic Tabu search and Tabu search with the LSTM neural network for the different value of L parameter

value of the energy for all used sequence lengths, which is presented in Table 1. The test time is almost the same for both solutions and it equals 19' when L=128, 45' or 46' when L=192 and 77' or 79' for L=256, where longer time is always needed for the solution with LSTM but this difference is marginal.

Table 1: Min and average value during 10 runs through 512 iterations

Sequence Lenght	**Method**	**Min Value**	**Average Value**	**Sequence Lenght**	**Method**	**Min Value**	**Average Value**
128	**TABU**	1484	1555.48	**128**	**TABU with LSTM**	1444	1539.80
192	**TABU**	3432	3567.54	**192**	**TABU with LSTM**	3284	3567.54
256	**TABU**	6372	6595.80	**256**	**TABU with LSTM**	6180	6518.98

5 Conclusions and further work

The presented paper is the latest of our research related to finding an efficient approach to resolving the LABS problem. The presented work, as our first step of combining a neural network with local optimisation heuristics for hard discrete problems, shows that incorporating deep learning models for predicting the internal parameters of a Tabu search significantly improves the final results from our attempt to solve the LABS problem. It is also worth mentioning that the computational overhead caused by the additional step of the prediction of the M parameter is almost negligible. The presented method seems to be very promising and allows us to identify a few further lines of research such as improvement of the LSTM architecture and its topological impact on parameters relating to prediction efficiency, applying the concept to other local search techniques and the optimisation of the developed algorithms with GPGPU.

Acknowledgments The research presented in this paper was realised with funds of Polish Ministry of Science and Higher Education assigned to AGH University of Science and Technology and it was supported in part by PLGrid Infrastructure.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_32

Bibliography

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

[2] Bello, I., Pham, H.: Workshop track -iclr 2017 neural combinatorial optimization with reinforcement learning (2017)

[3] Bošković, B., Brglez, F., Brest, J.: Low-Autocorrelation Binary Sequences: On Improved Merit Factors and Runtime Predictions to Achieve Them. arXiv e-prints arXiv:1406.5301 (Jun 2014)

[4] Brest, J., Bošković, B.: A heuristic algorithm for a low autocorrelation binary sequence problem with odd length and high merit factor. IEEE Access **6**, 4127–4134 (2018). https://doi.org/10.1109/ACCESS.2018.2789916

[5] Brest, J., Bošković, B.: In searching of long skew-symmetric binary sequences with high merit factors (2020)

[6] Dotú, I., Van Hentenryck, P.: A Note on Low Autocorrelation Binary Sequences, pp. 685–689. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

[7] Gallardo, J.E., Cotta, C., Fernandez, A.J.: A memetic algorithm for the low autocorrelation binary sequence problem. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. pp. 1226–1233. GECCO '07, ACM, New York, NY, USA (2007)

[8] 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)

[9] Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Norwell, MA, USA (1997)

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

[11] Naick, B.S., Kumar, P.R.: Detection of low auto correlation binary sequences using meta heuristic approach. International Journal of Computer Applications **106**(10) (2014)

[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] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. p. 3104–3112. NIPS'14, MIT Press, Cambridge, MA, USA (2014)

[14] Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015)

[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)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_32