Experimental Study of Variation Local Search Mechanism for Bee Algorithm Feature Selection

M. Mahmuddin[1] and Aras Ghazi Mohammed Al-dawoodi[1,2]
[1]*School of Computing, Universiti Utara Malaysia, 06010, Kedah Malaysia*.
[2]*College of Computer and Mathematical Sciences, Tikrit University, Tikrit, Iraq.*
ady@uum.edu.my

***Abstract*—The Bees Algorithm (BA) has been applied for finding the best possible subset features of a dataset. However, the main issue of the BA for feature selection is that it requires long computational time. This is due to the nature of BA combination search approach that exploits neighborhoods with random explorative. This situation creates unwanted sub-optimum solution(s) leading to the lack of accuracy and longer processing time. A set of different local neighborhood search extension and their combination approaches have been proposed, including Simple-swap, 2-Opt, 3-Opt, and 4-Opt. The performance of the proposed mechanism was compared and analyzed using benchmark dataset. The results from experimental work confirmed that the proposed approach provides better accuracy with suitable time.**

***Index Terms*—Wrapper Feature Selection; Optimization; Bees Algorithm**

I. INTRODUCTION

Over the recent years, many nature-inspired optimization techniques have been applied on feature selection problem, and one of these techniques is Bees Algorithm (BA) [1]. BA works based on the scout's bees foraging food activities, where worker bees search for promising flower patches which contain large amounts of food (nectar or pollen). During this quest, a percentage of the population is kept back. When they return to the hive, a certain amount of their food is deposited. Later the scout bees perform the "waggle dance" on the "dance floor" to disseminate crucial food's source information [2]. After the waggle dance, the scout bee goes back to the flower patch followed by the recruiter bees. In case of more promising patches, more recruiter bees are sent, thus resulting in quicker and efficient collection of food for the colony. This idea has successfully been applied in many areas, including parameters setting optimization, data clustering, and other combinatorial optimization problems.

However, due to repetitive iteration, the BA approach takes longer execution time to get the optimal result, especially in local search neighborhood procedures [2,3]. Bees spend a lot of time identifying the global optimal solution or choosing a good location for producing the best fitness. Moreover, BA involves a huge number of computational processes to obtain a good solution, especially in complex problem. The approach does not guarantee any optimum solutions for the problem, mainly due to the lack of accuracy. Therefore, a proper mechanism is needed to overcome these challenges, and new operators to a) reduce computational processes, b) increase accuracy, and c) improve speed are proposed.

This paper is organized as follows. In Section II, the algorithm elated BA for feature selection methods are given. In Section III, the proposed improved mechanism and the combination neighborhood search-extension are presented. The experimental results of comparing the algorithm proposed in this paper with other algorithms are also presented in Section IV. Finally, this work is summarized in the last section.

II. THE BEES ALGORITHM FEATURE SELECTION

A. The Bees Algorithm Mechanism

The foraging activities of bees prompted researchers to attempt an imitation of their activities to find the best possible solution. This requires a few adjustments in a computer and programming environment. Firstly, a predefined number of bees will be dispersed to the food source. Unlike the factors of natural bees (weather conditions, humidity and the total number of bees in the hive), the total number of bees in computer programming depends on the nature of the problem. This initialization process of the bio-bees searching algorithm also occurs in many optimization algorithms including genetic algorithm [4,5], ant colony optimization [6] and particle swarm intelligence [7].

Secondly, in computer programming, bees are known as a swarm agent and the food is known as a problem function. The swarm agent goes to every potential site and returns with the solution that best meets the criteria of the problem function. The initialization solution is randomly generated to make it more robust and cover most of the possible solutions.

The BA starts with initialization of parameters in defining the total number of bees, range of searching space, normal and 'elite' sites. A stopping criterion is set to ensure the algorithm does not exceeded the total number of iteration or the fitness value. More information of the algorithm is in [1].

B. Bit Feature Representation

The BA requires a new way of representing the feature to make it easy for distributing and selecting the feature subset. The BA also was developed for continuous domains considering the necessity to modify the neighborhood part by replacing the patch with a local search operator. Since it is not possible to directly use BA for feature selection, an extension of BA is needed. This is done by introducing bit feature subset representation, where '1' indicates that a particular feature is selected and '0' means otherwise. In this approach, forward selection or backward elimination [8] techniques would be inappropriate since it takes longer to find suitable subsets. For example, if the total number of the original features is 8 and only 7 of them are selected, a few representations of the new subset features are generated: a) '01111111' means that the first feature from the total of the original features will not be selected, and b) '1111110' means that only the first seven

features are selected.

C. Wrapper BA Feature Selection

One of the primary objectives of feature selection is to generate a lesser m number of features of the original data N where $m < N$, $m \neq 0$. As feature dimensionality is reduced, fewer features will be evaluated causing less computational time and thus faster learning. Besides, it produces simple rules (general model) which are more meaningful to the user due to the preference of an easy description instead of a complicated one. Features selection normally can be categorized into three main approaches, namely the filter, wrapper and hybrid. Popular approach such as the wrapper has some advantages due to its simplicity to interact with classifier to generate better result, and the minimal computational cost as compared to filter approach [9,10].
Guided random wrapper feature generated by the BA is adopted for this purpose. Data will be generated randomly and the goodness of the generated subset feature will be determined using multilayer perceptron (MLP). The BA evaluation also will determine the fitness of this subset features and only the best fittest will be stored. Overall, the wrapper BA feature selection as depicted in Figure 1. The BA process will be repeated until the termination criterion is met either by the total number of replication or when the fitness is reached.

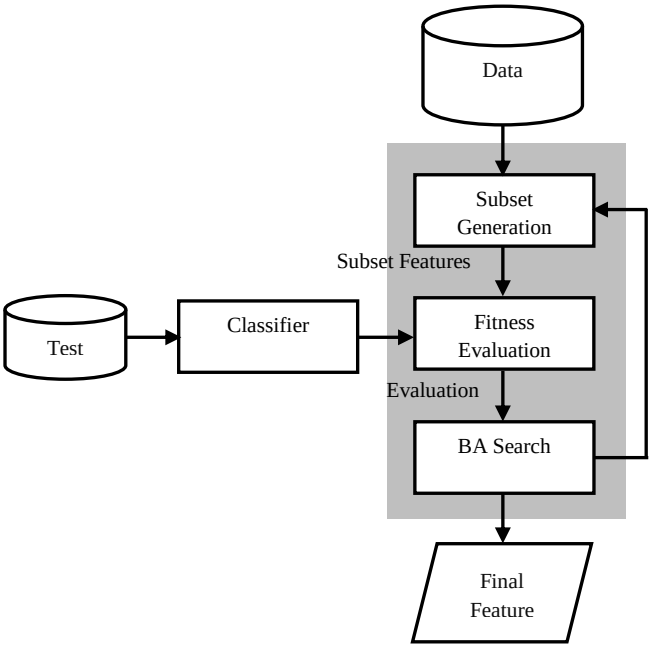

Figure 1: Flow diagram of the processes of the system BA wrapper feature selection

The MLP network has an outstanding capability to extract patterns and interpret the meaning of data, which is too complicated to recognize. The network relies on iterative learning from the initial experience of a given data pattern. The MLP is also capable of predicting and detecting the trends of a complex dataset, which can be seen in many pattern recognition applications, including the gesture recognition system and the material science.

III. EXPERIMENT

A. Principle of Swap Subset Mechanism

This work has been proposed in [2] as part of the original idea for BA feature selection. In his work, several mechanisms and extensions have been proposed, including the simple swap, 2-Opt, 3-Opt and 4-Opt. Each of the possible combination bit mutations is proposed. An extension of this work is the multiplication of its variation and the introduction of a new extension of these combinations.

Table 1
Variation of the proposed experiment

Type	Method's Name	Operation
Original Method	O1	Simple swap + mutation
	O2	2-Opt + mutation
	O3	3-Opt + mutation
New Extension	E1	4-Opt + Mutation
	E2	Simple Swap + 2-Opt + Mutation
	E3	Simple Swap + 3-Opt + Mutation
	E4	Simple Swap + 4-Opt + Mutation
	E5	2-Opt + 3-Opt + Mutation
	E6	2-Opt + 4-Opt + Mutation
	E7	3-Opt + 4-Opt + Mutation
	E8	Simple Swap + 2-Opt + 3-Opt + Mutation
	E9	2-Opt + 3-Opt + 4-Opt + Mutation
	E10	Simple Swap + 2-Opt + 4-Opt + Mutation
	E11	Simple Swap + 3-Opt + 4-Opt + Mutation
	E12	Simple Swap + 2-Opt + 3-Opt + 4-Opt + Mutation

This approach is implemented by employing a simple swap of features to be evaluated with swap values at two random points, L_1 and L_2 where $L_2 > L_1$. The value of the feature at the index of L1 (F_{L1}) will be swapped with the value of the feature at the index of L_2 (F_{L2}).

FS starts with a random generation of a population of binary strings (or bees). For each string, a new dataset is constructed using the selected features specified in the string. The training data of the dataset are used in training the MLP, whereas the remaining data or the test data are employed to evaluate the classification accuracy of the trained MLP. The proposed algorithm will start with the parameter initialisation phase that includes the following parameters: the total number of Bees (*n*), the total number of "elite" Bees (*e*), the number of sites selected for neighbourhood search (*m*), the number of Bees around selected location (*nsp*), and the number of Bees around each "elite" locations (*nep*).

The next step in this algorithm is the organisation of the bees based on the fitness value acquired from the initialisation process. Some of the new bees will be assigned to an elected number of features (also known as 'sites' in the original BA terminology). The top quality features will be identified by evaluating the fitness value of each bee. This best feature subset is known as "elite" (*e*) which becomes the target feature for other bees as they attempt to find possible better features from the existing ones. A total number of bees (*nep*) will be assigned to these potential features. Each of these new bees will be assigned to new possible better subset features.

The assignment of new features is based on the existing fitness features. The neighbourhood search process will start with the generation and identification of the best features. A newly generated feature F_{new} will be evaluated using the same principle equation as in the initialisation phase. A local search

method is applied to generate this new feature, f_{new} from the existing feature F_i. The local search method will implement a combination method for new feature generation process.

The remaining *n-m* features of other bees are filled by some other new candidates. These newly recruited bees have entirely new features which are randomly generated. The fitness of the newly generated features of each bee is calculated here.

In the next phase, features of each bee (old and newly generated) are organized once more to find the best features in the list of new bees. The process continues until it surpasses the total number of iterations.

B. 2-Opt Subset

It is a simple local search algorithm first proposed by [11] for solving the travelling problem of salesman. The 2-Opt operator is the simplest and easiest of all operators in the k-opt family to solve a problem [12]. The main objective of its implementation is to make small changes on the tour and check if the solution quality improves [11]. Although the basic move had already been suggested by [13], this move deletes two edges, thus breaking the tour into two paths, and then reconnects those paths in the other possible way [14]. The 2-Opt-based local search approach is also used in the old proposed algorithm [2]. In this study, the search starts by initializing two points of swapping, L_1 and L_2 , where $L_2 > L_1$. The process is implemented by swapping the feature values at indices L1 (F_{L1}) and L2 (F_{L2}). The swapping process continues with (F_{L1+1}= F_{L2-1}) until L_1 and L_2 have the same value.

C. 3-Opt Subset

It is an algorithm that creates three tour segments by removing three edges from the tour. This allows the addition of a new element to the method, thus reconnecting the tour segments in different ways [12] to locate the best possible way. This makes the 3-Opt-based search relatively slower than the 2-Opt based search, but it creates tours with higher quality than 2-Opt [15]. Therefore, in this proposed algorithm, 3-Opt-based search approach was also applied as an old idea used in [2]. This search approach involves random generation of reference points at L_1, L_2 and L_3 where $L_1 > L_2 > L_3$. The feature values at index L_1 and L_2 (F_{L1} and F_{L2} respectively) use a 2-Opt-based operation. Features between F_{L2+1} and F_{L3} will move into features between F_{L1} and $F_{L1+ (L3 - L2)}$.

The proposed algorithm also uses a modified 2-Opt-liked moving operation. 2-Opt involves two index points that are generated randomly. The movement starts with the first feature and continues with the next feature until the first index point, L_1 is found at feature F_{L1}. The next move starts at second index, L_2 at feature F_{L2} where the next move feature is read. The feature movement process of 2-Opt continues until the last feature, F total. The next feature to be read is next to index L_1, at F_{L1+1}. These moves stop at feature, F_{L2-1} by which time all features have been covered.

D. 4-Opt Subset

4-Opt move was first mentioned by Lin and Kernighan in 1973 [176] as an example of a simple move which cannot be normally generated by 3-Opt. This move is used by different modern algorithms for its ability to escape from the local optima [16].

The 4-Opt-based search approach is an extension to simple swap, 2-Opt, and 3-Opt operators. In the cases of the simple swap and 2-Opt operator, there is only one way to reconnect the tour fragments after deleting the two selected edges [2,12]. The 3-Opt operator chooses the best triple edges that are not yet connected to the current tour [15]. In contrast, the 4-Opt is used as the perturbation technique, and a stochastic 2-Opt is used as the embedded local search heuristic. The double-bridge move involves partitioning a permutation into 4 pieces (a, b, c, d) and putting them back together in a specific and jumbled ordering (a, d, c, b) in the TSP problem [17].

For this, reference points are generated randomly at L_1, L_2, L_3 and L_4 where $L_1 > L_2 > L_3 > L_4$. Feature values at index L_1, L_2, L_3 and L_4 (F_{L1}, F_{L2}, F_{L3} and F_{L4}, respectively) use a 2-Opt-based operation as the two sequential parts. The process is implemented by swapping the feature values at indices L_1 (F_{L1}) and L_2 (F_{L2}) as the first part and L_3 (F_{L3}) and L_4 (F_{L4}) as the second part. The swapping process continues with (F_{L1+1} = F_{L2-1}) and (F_{L3+1} = F_{L4-1}) until (L_1 and L_2) and (L_3 and L_4) have the same value.

E. Fitness Function

This approach generates random subset features and the total mean squared error (MSE) will be calculated for the fitness function. MSE calculation is based on the gained error from a MLP of each generated subset features.

$$F(x) = \frac{1}{(k_1 \times MSE) + (k_2 \times \frac{N_s}{N_t})} \tag{1}$$

where MSE are computed based on Eq. 2. Both functions are based from proposal in [2].

$$MSE = \frac{1}{N} \sum_{i=1}^{N+1} x_i \tag{2}$$

F. Datasets

Five different numeral datasets have been chosen from the UCI Database [18] to evaluate the proposed mechanism. These datasets have been chosen rationally due to a few reasons including availability, popularity and numerical datasets. A summary of the datasets are described in Table 2. Table 3 shows the standard parameters setting for MLP that have been integrated with the BA's parameters respectively. The standard parameters have been used in this work to ensure the similarity between the condition of the original work and the earlier version of this work.

Table 2
Summary of selected benchmarked datasets

Dataset Name	Number of Feature	Number of Instance	Data Type Refer
Wine	13	178	Real
Soybean	35	47	Real
SPECT Heart	22	267	Continuous
Ionosphere	34	351	Continuous
Lung-cancer	56	32	Integer
Hepatitis	19	155	Categorical, integer, real
Pima Indian Diabetes	8	768	Integer, real
Vehicle Silhouettes	18	946	Integer

Table 3
Summary of MLP and BA's parameters used in the experiment

Type	Parameters	Values
MLP	Number of Hidden Layer	1
	Desired Error	0.001
	Learning Momentum	0.1
	Learning Rate	0.3
	Number of Epoch	500
	Cross-Validation	10
BA	Number of Bees	25
	Number of Neighbourhood search	5
	Number of "elite" sites	2
	Number of bees recruited for elite sites	15
	Number of bees recruited for other sites	20

IV. RESULTS AND DISCUSSION

The overall measurement results are summarized in Table 4. All the experiments have been conducted on a computer with Intel Core I5, 2.53 GHz and 4-GB RAM. The Bees Algorithm feature selection algorithm was implemented using C++ programming language and tested using Weka version 3.7.12. From the obtained results, it can be noted that classification accuracies comparable with those for the full-feature cases were achieved despite large reductions in the number of features. This confirms the ability of the proposed method to choose the informative features. As shown in Figure 2, the effect of applying proposed extension and their combination on numerical dataset has significantly improved the time and increased the accuracy of the BA feature selection.

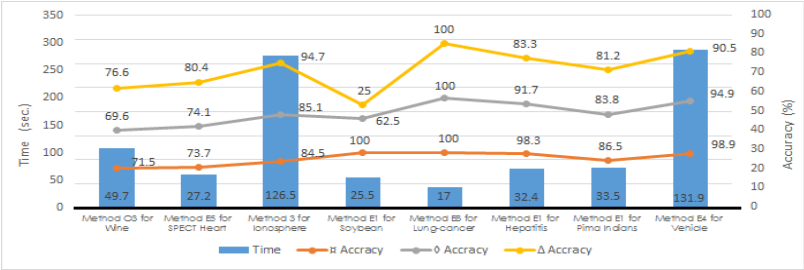

Figure 2: Maximum accuracy and minimum time for variation local search mechanism on different numerical dataset

Table 4
Results of the best proposed mechanism on benchmark datasets

Dataset	Method	#Selected Feature	MSE	∞Fit.	Time (sec.)	¤Acc (%)	◊Acc (%)	ΔAcc (%)
Wine Data set	O1	4/13	0.1950	0.6654	52.7	95.5	89.9	81.0
	O2	4/13	0.1733	0.6752	50.2	94.9	92.1	95.2
	O3	4/13	0.1645	0.6793	52.9	96.0	93.8	100.0
	E1	4/13	0.1469	0.6874	52.6	97.8	91.6	95.2
	E9	4/13	0.1641	0.6794	49.7	98.9	94.9	90.5
SPECT Heart Data set	O1	6/22	0.1835	0.6866	30.4	78.3	71.3	80.0
	O2	7/22	0.1942	0.6611	29.3	91.3	75.0	80.0
	O3	7/22	0.1941	0.6612	29.1	79.4	79.4	84.37
	E1	7/22	0.1882	0.6638	27.2	86.5	83.8	81.2
	E5	6/22	0.2069	0.6758	30.5	97.5	93.8	90.0
Ionosphere Data set	O1	12/34	0.0001	0.7390	126.5	98.3	91.7	83.3
	O2	12/34	0.0001	0.7390	127.7	98.6	90.9	78.6
	O3	12/34	0.0000	0.7390	146.1	98.9	92.3	90.5
	E1	13/34	0.0000	0.7233	130.6	98.6	91.5	79.8
	E9	12/34	0.0000	0.7390	136.3	98.9	92.3	89.9
Soybean Data set	O1	10/35	0.0078	0.7734	31.3	100.0	100.0	100.0
	O2	13/35	0.0045	0.7267	28.6	100.0	100.0	100.0
	O3	11/35	0.0034	0.7590	28.4	100.0	100.0	100.0
	E1	10/35	0.0044	0.7745	25.5	100.0	100.0	100.0
	E9	11/35	0.0056	0.7578	30.1	100.0	100.0	100.0
Lung-cancer Data set	O1	20/56	0.0948	0.7061	18.5	100.0	46.9	50.0
	O2	24/56	0.0574	0.6729	17.4	100.0	46.9	50.0
	O3	24/56	0.0407	0.7063	17.4	100.0	31.3	50.0
	E1	24/56	0.0003	0.7347	17.9	100.0	50.0	25.0
	E8	19/56	0.0116	0.7603	17.0	100.0	62.5	25.0
Hepatitis Data set	O1	5/19	0.0000	0.7916	38.4	87.1	77.4	94.7
	O2	6/19	0.0000	0.7599	35.5	94.7	79.4	95.5
	O3	7/19	0.3748	0.5736	34.9	92.3	82.6	78.9
	E1	6/19	0.0000	0.5979	32.4	84.5	85.1	94.7
	E9	4/19	0.0000	0.8260	35.0	90.3	85.1	78.1
Pima Indians Diabetes Data set	O1	2/8	0.0000	0.7999	50.3	74.7	67.0	80.4
	O2	2/8	0.0000	0.8000	52.4	63.8	65.0	61.
	O3	2/8	0.0000	0.7999	46.7	66.3	66.1	65.2
	E1	3/8	0.0000	0.8888	44.4	74.7	75.5	81.4
	E9	2/8	0.0000	0.7999	33.5	73.7	74.1	80.4
Vehicle Data set	O1	5/18	0.0001	0.7826	145.5	67.8	67.4	68.6
	O2	6/18	0.0000	0.7500	133.0	70.8	67.8	63.7
	O3	6/18	0.0000	0.7500	138.0	74.8	68.4	70.6
	E1	4/18	0.0000	0.8571	131.9	71.5	69.6	76.6
	E4	7/18	0.0000	0.7200	135.0	77.3	72.2	84.3

Total number of selected feature. For example 4/12 means 4 out of 12 total features have been selected.
∞ Fitness values obtained based on Eq. 2.
¤ Accuracy obtained number of accurately classify using MLP in WEKA use training set.
◊ Accuracy obtained number of accurately classify using MLP in WEKA use cross-validation (10).
Δ Accuracy obtained number of accurately classify using MLP in WEKA use percentage split 88%.

V. CONCLUSION

Enhancements to neighborhood search and parameter numbers are represented in this work. An extension operator 4-Opt and combination methods are introduced to reduce time consuming and increase the accuracy. From the overall obtained results, it can be concluded that the classification of accuracies and time execution comparable with those for the full-feature cases were achieved despite large reductions in the number of features. This confirms the ability of the proposed method to choose informative features. This study empirically showed the effect of applying proposed extension and their combination of numerical dataset. This was determined by comparing the results between the original methods and the original BA feature selection and the extension proposed of swapping mechanism of it. Finally, the experimental results confirmed that the proposed extension of the search neighborhood and their combination approaches provide an alternative approaches that offer good accuracy with suitable time in comparison to the original approach.

ACKNOWLEDGMENT

We are grateful for the UUM's support on this work.

REFERENCES

[1] Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., and Zaidi, M. 2006. The Bees Algorithm – A Novel Tool for Complex Optimisation Problems. 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006) Cardiff, UK: Elsevier. 454-459.

[2] Mahmuddin, M. 2008. Bees Algorithm in Machine Learning Problems. Cardiff University, Cardiff.

[3] Mahmuddin, M., and Yusof, Y. 2009. A Near-Optimal Centroids Initialization in K-Means Algorithm Using Bees Algorithm, Kuala Lumpur.

[4] Huang, C.-L., and Wang, C.-J. 2006. A GA-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications. 31(2): 231-240.

[5] Huang, J., Cai, Y., and Xu, X. 2006. A Wrapper for Feature Selection Based on Mutual Information. In C. Yunze (Ed.), 18th International Conference on Pattern Recognition, 2006(ICPR 2006) Hong Kong: IEEE Computer Society. 2: 618-621.

[6] Karnant, M., Thangavel, K., Sivakuar, R., and Geetha, K. 2006. Ant Colony Optimization for Feature Selection and Classification of Microcalcifications. `Digital mammograms 14th International Conference on Advanced Computing and Communications, 2006 (ADCOM 2006) Surathkal, India: IEEE Computer Society. 298-303.

[7] Xue, B., Zhang, M., and Browne, W. N. 2013. Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. Cybernetics, IEEE Transaction. 43(6): 1656-1671. doi: 10.1109/tsmcb.2012.2227469

[8] Kohavi, R., and John, G. H. 1997. Wrappers for Feature Subset Selection. Artificial Intelligence. 97(1-2): 273-324. doi: 10.1016/S0004-3702(97)00043-X

[9] Saeys, Y., Inza, I., and Larrañaga, P. 2007. A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics. 23(19): 2507-2517. doi: 10.1093/bioinformatics/btm344

[10] Garnes, Ø. L. 2009. Feature Selection for Text Categorisation. Master of Science in Computer Science, Norwegian University of Science and Technology, Trondheim.

[11] Croes, G. A. 1958. A Method for Solving Traveling-Salesman Problems. Operations Research. 6(6): 791-812.

[12] Fosin, J., Davidović, D., and Carić, T. 2013. A GPU Implementation of Local Search Operators for Symmetric Travelling Salesman Problem. Promet – Traffic & Transportation. 25(3): 225-234.

[13] Flood, M. M. 1956. The Traveling-Salesman Problem. Operations Research. 4(1): 61-75. doi: 10.2307/167517

[14] Johnson, D. S., and McGeoch, L. A. 1997. The Traveling Salesman Problem: A Case Study in Local Optimization. In E. H. L. Aartsand J. K. Lenstra (Eds.), Local Search in Combinatorial Optimization John Wiley and Sons, Ltd. 215-310.

[15] Walshaw, C. 2002. A Multilevel Approach to the Travelling Salesman Problem. Operations Research. 50(5): 862-877. doi: 10.1287/opre.50.5.862.373

[16] Glover, F. 1996. Finding A Best Traveling Salesman 4-Opt Move in The Same Time as a Best 2-Opt Move. Journal of Heuristics. 2(2): 169-179. doi: 10.1007/bf00247211

[17] Brownlee, J. 2012. Clever Algorithms: Nature-Inspired Programming Recipes: lulu.com.

[18] Lichman, M. 2013. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml