IEICE
TRANSACTIONS

on Fundamentals of Electronics,
Communications
and Computer Sciences

VOL. E100-A NO. 9
SEPTEMBER 2017

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

LETTER

Neighbor-Interactive Bee Colony for Problems with Local Structures

Phuc Nguyen HONG[†], *Nonmember*, Chang Wook AHN[††], *Member, and* Jaehoon (Paul) JEONG[†††a)], *Nonmember*

SUMMARY In this letter, we integrate domain information into the original artificial bee colony algorithm to create a novel, neighbor-interactive bee colony algorithm. We use the Hamming distance measure to compute variable dependency between two binary variables and employ the Gini correlation coefficient to compute variable relation between integer variables. The proposed optimization method was evaluated by minimizing binary Ising models, integer Potts models, and trapped functions. Experimental results show that the proposed method outperformed the traditional artificial bee colony and other meta-heuristics in all the testing cases.

key words: *artificial bee colony, Hamming distance, Gini correlation coefficient*

1. Introduction

Meta-heuristic optimization methods have been popularly used to solve complex problems that are difficult to solve using derivative-based methods. Some popular meta-heuristic methods are differential evolution (DE) [1], genetic algorithm (GA) [2], clonal selection algorithm (CSA) [3], and artificial bee colony (ABC) [4].

Since ABC was introduced [4], it has been used to address several problems in various fields. Recently, the linkage artificial bee colony (LABC) method [5] has been introduced to address the linkage problem where variables are dependent and interactive. However, LABC does not employ important available information from the problem models, such as the Poss models [6] in image processing and computer vision that a pixel (a variable) interacts locally with its neighboring pixels.

In this letter, we present a novel, neighbor-interactive bee colony (NIBC) method that efficiently employs the variable-interactive information from the problem model and can operate more quickly and accurately than LABC as well as other meta-heuristic testing methods. In addition, we introduce the Hamming distance measure and Gini correlation coefficient to compute the interaction between two variables instead of using mutual information. Mutual information requires large numbers of instances for a variable in integer-value problems. Therefore, mutual information operates inadequate for small instances. For example, when the population size of ABC is about 100 (suggested population size for ABC) and feasible space [0,9] for an integer variable, the variable instances is too small to compute joint distribution between two variables robustly. On the other hand, mutual information becomes too complex for binary variable because in these cases the Hamming distance can compute the similarity between two binary data series accurately and quickly. The contributions of this letter are threefold:

- We propose the novel NIBC method that can extract the information of specific models and resolve the linkage information problem efficiently.
- We introduce dependency measures to compute the interaction between two variables.
- We evaluated NIBC using Ising and Potts models, and trapped functions. The experimental results show that NIBC outperforms other testing methods.

2. Variable Dependency Measures

2.1 Hamming Distance

The correlation between two series of binary data points is computed by using Hamming distance measure, which count the number of value differences between the two data series. Let x and y be two bit series, then the Hamming distance between x and y is computed as $H(x, y) = \sum_i \delta(x_i, y_i)$ where δ is an indicator function that returns 0 when the two compared values are the same, and vice versa.

2.2 Gini Correlation Coefficient

The Gini correlation coefficient (GCO) can compute the dependency of the monotonically non-linear association of two data series using the rearrangement inequality. Consider two order sets $\mathbf{L} = \{l_1, l_2, ..., l_m\}$ and $\mathbf{R} = \{r_1, r_2, ..., r_m\}$ of length m. The order set pairwise results are the set $\mathbf{G} = \{(l_1, r_1), ..., (l_m, r_m)\}$. By rearranging $\mathbf{G}$ according to the values of l_i, we obtain $\mathbf{G}_1 = \{(l_{(1)}, r_{[1]}), ..., (l_{(m)}, r_{[m]})\}$, where $(l_{(1)} \leq ... \leq l_{(m)})$ are the order statistics of $\mathbf{L}$ and $r_{[1]}, ..., r_{[m]}$ are the associated concomitants. Rearranging $\mathbf{G}$ according to the values of r_i, we obtain $\mathbf{G}_2 = \{(l_{[1]}, r_{(1)}), ..., (l_{[m]}, r_{(m)})\}$, where $r_{(1)} \leq ... \leq r_{(m)}$. Thus, GCO becomes

Manuscript received March 8, 2017.
Manuscript revised April 22, 2017.

[†] The author is with Department of Computer Engineering, Sungkyunkwan University, Republic of Korea.

[††] The author is with School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Republic of Korea.

[†††] The author is with Department of Interaction Science, Sungkyunkwan University, Republic of Korea.

a) E-mail: pauljeong@skku.edu (Corresponding author)

DOI: 10.1587/transfun.E100.A.2034

$$GCO\,(\mathbf{L},\mathbf{R}) \triangleq \frac{\sum_{i=1}^{m}(2i-1-m)r_{[i]}}{\sum_{i=1}^{m}(2i-1-m)r_{(i)}}, \tag{1}$$

where $GCO\,(\mathbf{L},\mathbf{R})$ is in the interval $[-1,1]$, and a larger value of it indicates higher correlation between $\mathbf{L}$ and $\mathbf{R}$.

3. ABC and NIBC

3.1 ABC

The ABC algorithm [4] mimics the foraging behavior of real honey bees and includes three groups: employed, onlooker, and scout bees. The employed bees search for solutions neighboring the candidate solution, $\mathbf{P}_i$, in their memory to find those that have better fitness. A candidate neighboring solution, $\mathbf{s}$, is created by stochastically changing randomly selected variable j of $\mathbf{P}_i$:

$$\mathbf{s}_j = \mathbf{P}_{i,j} + rand(-1,1) \times (\mathbf{P}_{i,j} - \mathbf{P}_{k,j}) \tag{2}$$

where k indicates the k^{th} candidate solution index, randomly selected from the candidate solutions and $rand(-1,1)$ generates a random number between -1 and 1 with a uniform distribution. The onlooker bees select promising candidate solutions probabilistically based on fitness information from the employed bees. After that a candidate solution is selected, the neighborhood of the solution is produced using Eq. (2), and a greedy selection is applied.

3.2 Proposed NIBC

In computer vision and image processing problems, such as stereo matching, optical flow, and segmentation, problems can be modeled as an energy function in a Markov random field using input images: these models are often based on Ising or Potts models. In complex problems, the variables are pairwise or multivariate dependent. Hence, changing a variable directly or indirectly affects the values of the other variables. When the value of one variable changes, this directly affects its neighborhood variables and indirectly affects others. Therefore, the optimal value val^* for one variable directly depends on the values of the neighborhood variables and indirectly depends on the values of the others.

The original ABC [4] fails to solve these kinds of problems because it manipulates one variable at a time, which leads to being trapped in weak suboptimal solutions. Recently, LABC has been introduced to solve these kinds of interactive problems. However, even thought LABC aimed to solve the Markov random field models, it assumes that a variable can associate with any other variables and does not employ information on variables interacting with their locally neighborhood variables (For example, one variable directly depends on eight neighbor variables). This leads to LABC inefficient in both its computation and accuracy.

Alg. 1 Procedure of Employed Bees Phase in NIBC

```
for each $\mathbf{P}_i \in \mathbf{P}$
    $\mathbf{s} \leftarrow \mathbf{P}_i$
    $\mathbf{s}_j$ changed by Eq. (2)
    for each $v \in \Omega_j$
        if $L(j,v) > \theta$
            $\mathbf{s}_v$ changed by Eq. (2) with $l = rand(1, FN)$
            $\mathbf{P}_i \leftarrow$ greedy selection between $\mathbf{P}_i$ and $\mathbf{s}$
        end if
    end for
end for
```

Alg. 2 Procedure of Onlooker Bees Phase in NIBC

```
for each $\mathbf{P}_i \in \mathbf{P}$
    $prob(i) \leftarrow$ compute probability for $\mathbf{P}_i$
end for
for each $\mathbf{P}_i \in \mathbf{P}$
    if $rand() < prob(i)$
        $\mathbf{s}_j$ changed by Eq. (2)
        for each $v \in \Omega_j$
            if $L(j,v) > \theta$
                $\mathbf{s}_v$ changed by Eq. (2) with $l = rand(1, FN)$
                $\mathbf{P}_i \leftarrow$ greedy selection between $\mathbf{P}_i$ and $\mathbf{s}$
            end if
        end for
    end if
end for
```

However, the eight variable associations can have different levels. For instance, one of two interactive variables (pixels) that belong to the same object in an image may have a higher association with two interactive variables on the object boundary. Therefore, we use a dependency measure to evaluate the level of interaction between the eight interactive variable pairs. We use the Hamming distance measure for binary-variable problems, while we employ the Gini correlation coefficient for integer-variable problems.

We can estimate the interactions among the eight interactive variable pairs indirectly by using information extracted from the population of candidate solutions. Once the dependencies of a set of variables have been determined, an algorithm can be used to manipulate all of the variables simultaneously. Hence, ABC behavior in the employed and onlooker bee phases should be changed to adapt to the new purpose. In our proposed algorithm, we use the Hamming distance and Gini correlation coefficient to measure the dependency $\mathbf{L}(i,j)$ between the two sets of data points of variables i and j, respectively.

In the employed and onlooker bee phases, a variable is selected in order to change its value. We can find the set of associated dependent variables based on the computed dependence values of neighboring variables, after which the values of these are also changed. Two variables are defined to be dependent when $\mathbf{L}(i,j)$ is greater than the threshold Θ. For $j = rand(1,n)$ and $k = rand(1,FN)$ where n is a problem size and FN is a population size, and let $\mathbf{P}$ and $\mathbf{P}_i$ be population and a solution, respectively, the employed bees and onlooker bees steps of the proposed NIBC algorithm is shown in Algs. 1 and 2. The employed bee and onlooker bee phases, after variable $\mathbf{s}_j$ for each selected solution $\mathbf{s}$ is changed, the interactive variable $\mathbf{s}_v$ is obtained using the information on the interaction level between j and v.

4. Experimental Results

NIBC and the previously discussed population-based metaheuristics (GA, DE, CSA, ABC and LABC) were tested on Ising and Potts models and trapped functions. Each experiment was run 50 times with the same seeds. For CSA and DE/rand/1/bin (DE), the parameter values were carefully selected. It has been suggested that N_{ABC} is optimal when it is within the interval $[50, 100]$ [4], and so $N_{ABC} = 100$ was chosen for ABC. For LABC, the parameter values were set as has been previously suggested in [5]. For the NIBC algorithm, we chose $\Theta = 0.05$ and $N_{NIBC} = 100$. For all the experiments, the limitation of the number of function evaluation is set to $1,000 \times n$.

We used a sign test technique to evaluate NIBC and the other test algorithms. The sign test technique evaluates the overall performances of the tested methods by counting the number of cases where the algorithm is the overall winner (Win(+)) or the overall loser (Lose(-)).

4.1 Ising Model

The energy function for the binary Potts model is $E(D) = E_{data}(D) + E_{smooth}(D)$, where $E_{data}(D)$ is the measurement of the cost value from each pixel (variable); and $E_{smooth}(D)$ is the measurement of the smoothness. $E_{smooth}(D)$ is defined as $E_{smooth}(D) = \sum_{\langle i,j\rangle\in\Omega} s\left(D_i, D_j\right)$, where

$$s\left(D_i, D_j\right) = \begin{cases} 0 & if\ D_i = D_j \\ \Delta & othewise, \end{cases} \tag{3}$$

Δ is a predefined penalty value that balances the smoothness and data terms, Ω is the set of neighboring variables in the grid, and $s(.)$ is a smoothness function that gives a penalty if the values of two variables are different. In these experiments, we used the Hamming distance for the computation of variable dependency.

Table 1 shows the performance of the optimization methods to minimize Ising models with various Δ values. The Ising model contains the pairwise interaction between two variables, as shown in (3). As ABC, CSA, DE, and GA are not aware of the variable-interactive problem in Ising model, they finalized with weak sub-optimal solutions, while NIBC manipulates a group of interactive variable at a time. As a result, NIBC significantly outperformed the algorithms for all the testing cases. In addition, LABC took computation cost to compute useless relations. Hence, LABC required large computation to achieve strong optimize solutions, and for problems with large number of variable and feasible space, the inefficiency prevents LABC to apply to real applications. Therefore, NIBC can reach to strong optimized solutions faster than LABC and outperformed LABC for most of the testing cases.

Table 1 Isings model with grid size n=100.

Delta	NIBC	ABC	CSA	DE	GA	LABC
Δ=0.2	Win (+)	48	48	47	48	43
	Lose (-)	2	2	3	2	7
	p-value	0.05	0.05	0.05	0.05	0.05
Δ=0.4	Win (+)	47	48	47	48	46
	Lose (-)	3	2	3	2	4
	p-value	0.05	0.05	0.05	0.05	0.05
Δ=0.6	Win (+)	47	47	48	49	43
	Lose (-)	3	3	2	1	7
	p-value	0.05	0.05	0.05	0.05	0.05
Δ=0.8	Win (+)	48	48	47	49	42
	Lose (-)	2	2	3	1	8
	p-value	0.05	0.05	0.05	0.05	0.05

Table 2 Integer-based Potts model with grid size n=100.

Delta	NIBC	ABC	CSA	DE	GA	LABC
Δ=0.2	Win (+)	47	49	49	49	42
	Lose (-)	3	1	1	1	8
	p-value	0.05	0.05	0.05	0.05	0.05
Δ=0.4	Win (+)	46	48	49	48	44
	Lose (-)	4	2	1	2	6
	p-value	0.05	0.05	0.05	0.05	0.05
Δ=0.6	Win (+)	47	49	48	49	43
	Lose (-)	3	1	2	1	7
	p-value	0.05	0.05	0.05	0.05	0.05
Δ=0.8	Win (+)	47	48	49	48	41
	Lose (-)	3	2	1	2	9
	p-value	0.05	0.05	0.05	0.05	0.05

4.2 Integer-Based Potts Model

In this subsection, we evaluated the performance of the testing methods to optimize the Potts models with various Δ values. We used GCO to compute variable dependency. The feasible space for the variables is $[0, 9]$. Table 2 shows the results for optimizing the Potts models. As with the binary models, NIBC outperformed the other methods because it effectively exploits the local variable interactions between variables.

4.3 Binary Trapped Function

The k-bit trapped function is a challenging problem, in which the problem can be divided into blocks of k-bit subproblems. For each k-bit sub-problem, v_i, the number of ones is $v_i = \sum_{j=1}^{k} x_{j+i\times k}$, the fitness of the block v_i is

$$f(v_i) = \begin{cases} k & if\ v_i = k \\ k - 1 - \mathrm{v}_i & otherwise, \end{cases} \tag{4}$$

and the fitness of a solution is $f(\mathbf{x}) = \sum_i f(v_i)$, where the optimal value for a solution is obtained when each of its blocks has the optimal value. Let $k = 4$, for example, then the optimum value for the block v_1 is $[x_1, x_2, x_3, x_4] = [1, 1, 1, 1]$. We evaluated the testing methods on four trap functions, and used the Hamming distance for the dependency computation. The dimension of these functions were set with $n = k \times 10$ where k is a the number of bits in a block of the trap function. For trapped function, two variables are considered to be neighbors if the position distance is smaller

Table 3 Pairwise comparison of NIBC with the other testing method using sign test technique for the trapped functions.

Trapped Function	NIBC	LABC	ABC	GA	CSA	DE
k=2	Win (+)	41	50	50	50	50
	Lose (-)	9	0	0	0	0
k=3	Win (+)	39	50	50	50	50
	Lose (-)	11	0	0	0	0
k=4	Win (+)	43	50	50	50	50
	Lose (-)	7	0	0	0	0
k=5	Win (+)	41	50	50	50	50
	Lose (-)	9	0	0	0	0

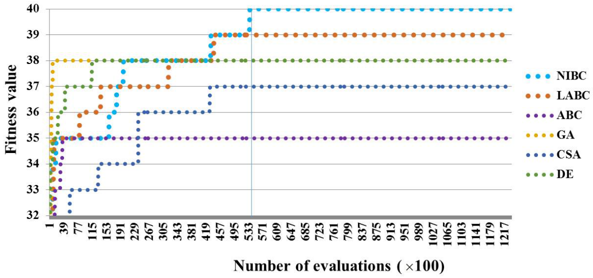

Fig. 1 Fitness values of the testing method using the trapped function with $k = 4$. The optimal fitness value is 40. The vertical line presents the position that NIBC obtains the optimal solution.

than or equal to k.

Tables 3 shows the performance comparison of the testing methods using the sign test technique for the different trap functions with different k values. Among the testing methods, only NIBC and LABC are aware of the interaction of variables. However, only NIBC can extract the locally interactive information and apply for optimization process. Therefore, it performed better LABC for most the testing cases. Other testing methods assume variables are independent and are trapped to sub-optimal solutions. In the trapped functions, optimization methods can be blocked if they process variables one by one. However, NIBC efficiently employs the interactive information and optimize the trapped functions effectively. For these functions, NIBC totally outperformed the conventional GA, ABC, DE, and CSA methods.

In addition, to have a different view of the performance of the testing optimization methods, we used the 4-bit trapped function to compute the fitness values of each method. The optimal fitness value is 40 using Eq. (4) with $n = k \times 10$. Figure 1 shows the quantitative results of the testing optimization methods. ABC, CSA, DE, and GA are not aware of the interactive problem, so they reached sub-optimal solutions. Even though keeping their running processes, they cannot reach the optimal solution because they assume that the variables of a problem are independent. Note that the optimized fitness values of these four methods depend on the population initiation and size. Therefore, they can be trapped in different sub-optimal solutions.

On the other hand, since LABC and NIBC are aware of the problem of the variable interaction, they perform better than the other four testing methods. However, LABC performed inefficiently because it does not employ the fact that variables are locally interactive. On the other hand, since NIBC operated efficiently, its fitness values were improved effectively in order to obtain the optimal solution much faster than LABC.

4.4 Computation Time

The Hamming distance and Gini methods have complexity of order $O(N)$ and $O(N \log N)$, respectively. Because NIBC required a smaller number of function evaluations to reach to optimize solutions than LABC, the proposed method can operate faster than LABC, especially in binary-variable problems.

We used the integer Potts model with problem size $n = 100$ to evaluate computation time. The experimental PC platform had an Intel Core2 Duo 3.40 GHz CPU and 4.00 GB of memory. DE took approximately 0.014 seconds, whereas CSA required approximately 0.015 seconds, ABC 0.031 second, and GA 0.012 seconds. LABC spent approximately 0.98 seconds, while NIBC took about 0.82.

5. Conclusion

We proposed an efficient optimization method that operates robustly and accurately for 2-dimensional interactive problems such as those posed by Ising and Potts models and trapped function, commonly used in computer vision and image processing problems. In addition, we proposed a new dependency measure that can compute the association between two variables. Experimental results showed that the proposed optimization method outperformed other legacy methods. In future work, we would apply the proposed method to specific functions in computer vision and image processing, such as segmentation and optical flow.

Acknowledgments

This research was supported by Basic Science Research Program through the NRF (2017R1D1A1B03035885) and IITP grant funded by the Korea government (MSIP) (B0717-17-0070).

References

[1] R. Storn and K. Price, "Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces," J. Global Optim., vol.11, no.4, pp.341–359, 1997.

[2] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.

[3] N.C. Leandro and F.J. Von Zuben, "Learning and optimization using the clonal selection principle," IEEE Trans. Evol. Comput., vol.6, no.3, pp.239–251, 2002.

[4] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," J. Global Optim., vol.39, no.3, pp.459–471, 2007.

[5] N.H. Phuc and W.A. Chang, "Linkage artificial bee colony for solving linkage problems," Expert. Syst. Appl., vol.61, pp.378–385, 2016.

[6] Q.D. Vinh, D.N. Vinh, and W.J. Jae, "Robust matching cost function for stereo correspondence using matching by tone mapping and adaptive orthogonal integral image," IEEE Trans. Image Process., vol.24, no.12, pp.5416–5431, 2015.