IEICE Proceeding Series

Efficiency of Chaotic Search and Realization of Ideal Search for Combinatorial Optimization

Mikio Hasegawa

Vol. 2 pp. 354-357
Publication Date: 2014/03/18
Online ISSN: 2188-5079

2013 International Symposium on Nonlinear Theory and its Applications
NOLTA2013, Santa Fe, USA, September 8-11, 2013

Efficiency of Chaotic Search and Realization of Ideal Search for Combinatorial Optimization

Mikio Hasegawa

†Department of Electrical Engineering, Tokyo University of Science
6-3-1 Niijyuku, Katsushika, Tokyo, 125-8585, Japan
Email: hasegawa@ee.kagu.tus.ac.jp

Abstract—This paper shows the causes of effectiveness of the chaotic noise for combinatorial optimization problems, and realizes ideal searches based on the theory in its background. Our previous works showed that effective chaotic noise for combinatorial optimization has negative autocorrelation. It has been also shown in a previous research that such negative autocorrelation in asynchronous sequences minimizes the cross-correlation among them. By applying such dynamics to asynchronously updated combinatorial optimization algorithms, cross-correlation among the heuristic operations becomes lowest and ideally complex search can be realized. In this paper, such ideal searches are applied to the Hopfield-Tank neural networks and the 2-opt methods. The results clearly show that the negative autocorrelation dynamics minimizes the cross-correlation among the heuristic operations and such ideal complex searches have highest performance. The proposed scheme does not require careful parameter settings to maximize its performance even for the large-scale problems.

1. Introduction

To the heuristic solution search algorithms for combinatorial optimization problems, stochastic or deterministic fluctuations have been applied in order to avoid traps at their local optima. As one of the stochastic methods, the simulated annealing [1] makes stochastic fluctuations gradually decrease and obtains better solution when it converges. As a deterministic approach, tabu searches avoid repeated search of same areas by forbidding the previous moves [2]. As another deterministic fluctuation to avoid sticking at undesirable local optimum, effectiveness of the chaotic dynamics has been shown by many results [3]–[7].

The first proposal of the chaotic optimization was on the Hopfield-Tank neural networks and their performances could be improved better than those with the stochastic fluctuations [3]. Chaotic dynamics has been also applied to the heuristic methods, such as the 2-opt methods for the Traveling Salesman Problems (TSPs) [4, 5] and the 2-exchanges for the Quadratic Assignment Problems (QAPs) [6]. Those results clearly show that the chaotic dynamics is more effective than the conventional stochastic or deterministic approaches even for large-scale problems.

Effectiveness of the chaotic fluctuation has been carefully analyzed from several points of views. An analysis using the method of surrogate data in Ref. [7] showed that an important characteristic of the chaotic fluctuations effective for performance improvement of heuristic algorithms is its specific autocorrelation. The results clearly showed that the stochastic surrogate data preserving the autocorrelation of the chaotic dynamics improves the performance of heuristic algorithm to the same level as the chaotic dynamics can do. Such effective autocorrelation of the chaotic dynamics has negative value in lag 1 and decreases with damped oscillation.

Importance of such autocorrelation with negative autocorrelation could be also seen in the chaotic CDMA [8, 9]. Asynchronous cross-correlation between the sequences can be minimized by the sequences, whose autocorrelation is $R(\tau) = C \times \lambda^{\tau}$, $\lambda = -2 + \sqrt{3}$ [9].

According to the theory, the chaotic fluctuations with negative autocorrelation make asynchronously updated heuristic algorithms having lowest cross-correlation among the heuristic operations (moves) in a searching space. By applying such a lowest-cross correlation moves among the heuristic operations, highly distributed searching dynamics can be generated, and it enables ideal avoids of local optimum solutions.

This paper shows the performance of the novel method using negative autocorrelation dynamics by applying it to asynchronously updated heuristic methods, the Hopfield-Tank neural networks and the 2-opt methods. The Lebesgue Spectrum Filter (LSF) is introduced to make autocorrelation of searching dynamics $R(\tau) = C \times \lambda^{\tau}$. First, the performances of a heuristic method with additive noise are analyzed to clarify effectiveness of the negative autocorrelation. Second, the proposed ideal search is applied to the neural networks and the 2-opt methods. Effectiveness of the proposed approach is shown by analyzing relations between solvable performances, autocorrelation of heuristic operations, and the cross-correlation among the operations.

2. Effects of Chaotic Noise for Asynchronously Updated Searches

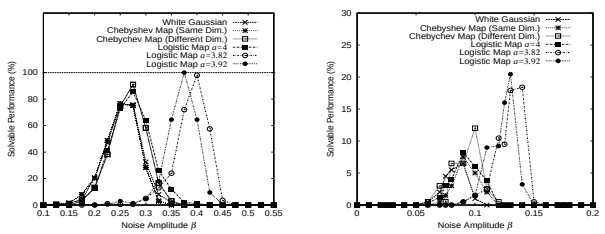
In this section, various additive noise sequences are applied to the Hopfield-Tank neural networks, and the depen-

dency of the performances on the noises are analyzed. The additive noise is introduced to the neural networks by using the following simple neuron update equation,

$$x_{ij} = (t+1) f[\sum_{k=1}^{N} \sum_{l=1}^{N} w_{ijkl} x_{kl}(t) - \theta_{ij} + \beta z_{ij}(t)], \quad (1)$$

where $x_{ij}(t)$ is the state of the (i, j)th neuron at time t, w_{ijkl} is the connection weights between the (i, j)th and the (k, l)th neurons, θ_{ij} is the threshold of the (i, j)th neuron, $z_{ij}(t)$ is the noise sequence applied to the (i, j)th neuron at time t, β is the amplitude of the noise, the output function f is the sigmoidal function, $f(y) = 1/(1 + \exp -y/\epsilon)$, respectively. For the additive noise sequences $z_{ij}(t)$, stochastic or chaotic sequences are introduced. They are normalized to have zero mean and unit variance.

Here we compare the performances of the neural networks with several different stochastic and chaotic additive noise sequences, which have different autocorrelations. In Fig. 1, we compare the results on the 20-city TSP used in Ref. [7] and the QAP, Nug12 in QAPLIB [12], by showing the rate of the optimum solutions obtained in 1000 different initial conditions, with changing the noise amplitude β. Cutoff time for each run is set at 2000 iterations. As the stochastic sequence, we introduce the white Gaussian noise. As the chaotic sequences, we introduce two types of the Chebychev maps chaos and the logistic map chaos, with different parameters.



(a)TSP (b)QAP

Figure 1: Solvable performances of the mutually connected neural network with chaotic and stochastic noises, whose autocorrelations are different.

From Figs. 1 (a) and (b), we can see that the chaotic sequences, generated by the logistic map with $a = 3.82$ and 3.92, improve the performance of the neural network much better than other noise sequences. As clearly shown in Fig. 2, these two chaotic noises, the logistic map with $a = 3.82$ and 3.92, have the negative value in lag 1 and converge to zero with damped oscillations. Other noises corresponding to the lower performance have white autocorrelation. The analysis in Ref. [7] using the method of surrogate data showed that specific autocorrelation of the chaotic noise is effective for performance improvement. By the numerical results in Figs. 1 and 2, importance of the negative autocorrelation noise can be clearly understand. In the optimization algorithms, the negative autocorrelation noise improves the performance of the asynchronously updated heuristic methods much better than the white noise.

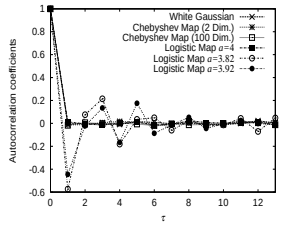


Figure 2: Autocorrelation coefficients of the chaotic and the stochastic noises, used as the additive noise for the mutually connected neural networks.

3. Theory of Negative Autocorrelation for Asynchronously Updated Heuristic Operations

The previous section showed that the chaotic dynamics with negative autocorrelation improves the performance of the asynchronously updated heuristic methods. Such negative autocorrelation chaos has been also applied in other field, wireless communications using the CDMA. In DS-CDMA, cross-correlation among the sequences should be lower to minimize interferences among the user signals. In Ref. 14, autocorrelation of each sequence to minimize the cross-correlation among the sequences has been calculated mathematically.

Here, cross-correlation between two sequences, X and Y, which takes two states, 1 or -1, is calculated, as an example. For the asynchronously updated methods, asynchronous mutual cross-correlation between X and Y can be minimized by the following terms, which is brought by the difference of the timing between the sequences caused by asynchronousness,

$$I = (1-\phi) R_N^{E/O}(l; X, Y) + \phi R_N^{E/O}(l+1; X, Y), \quad (2)$$

where, ϕ is the difference of the timing between X and Y caused by the asynchronous updating, $R_N^{E/O}$ is the average mutual interference of the even and the odd cross-correlation between the sequences X and Y of the lth symbol, respectively. When we define each sequence, taking -1 and 1 by the Markov chain with the following state transition probability matrix,

$$P(\lambda) = \begin{bmatrix} \frac{1+\lambda}{2} & \frac{1-\lambda}{2} \\ \frac{1-\lambda}{2} & \frac{1+\lambda}{2} \end{bmatrix} \quad (3)$$

the average asynchronous mutual interference among the sequences X and Y can be obtained as the following when $\phi \neq 0$,

$$E_\phi[E_{XY}[I^2]] = \frac{2}{3} \frac{1+\lambda+\lambda^2}{1-\lambda} \quad (4)$$

From the Eq. (4), it becomes clear that the asynchronous cross-correlation between the sequences can be minimized

by setting λ at value corresponding to the minimum of $E_\phi[E_{XY}[I^2]]$, which can be obtained as $\lambda = -2 + \sqrt{3}$.

From the above calculation result, we can understand that the autocorrelation function of the ideal sequence becomes, $R(\tau) = C \times \lambda^\tau$ with $\lambda = -2 + \sqrt{3}$. In the chaotic CDMA researches [8, 9, 13], effectiveness of such sequences, whose autocorrelation is around $\lambda = -2 + \sqrt{3}$, has also been shown by computer simulations. In this research, we analyze the effectiveness of such minimization of cross-correlation for the asynchronously updated heuristic methods for combinatorial optimization problems. In the asynchronously updated solution search algorithms, the lowest cross-correlation makes the dynamics the most distributive and theoretically ideal search will be realized.

4. Realization of Ideal Search on Asynchronously Updated Methods

In this section, we generate such ideal dynamics using the LSF [13], and analyze the effectiveness in a heuristic algorithm as an example, the mutually connected neural network used in Sec. 2. The LSF has been proposed by Umeno et al. [13] for generating the ideal sequences for the asynchronous CDMA, which can be expressed as a Finite Impulse Response (FIR) Filter,

$$\hat{f}(t) = \sum_{\tau=0}^{M} r^\tau f(t-\tau). \tag{5}$$

By applying this filter to the white sequences, which has zero autocorrelations, their autocorrelation can be modified to $C \times r^\tau$. Therefore, by setting $r = -2 + \sqrt{3}$, ideal autocorrelation dynamics to minimize the asynchronous cross-correlation can be achieved by such filter. In Ref. [13], the bit error rate performance could be improved 15% compared to the conventional white sequence cases. By setting $M \to \infty$, Eq. (5) can be modified to the suitable form for numerical computation, $\hat{f}(t) = r\hat{f}(t-1) + f(t)$.

Here, we apply the LSF to the improvement of the solution search performance of the asynchronously updated heuristic algorithms. First, we apply the LSF to the mutually connected neural network, and analyze its spatiotemporal searching dynamics. To introduce the LSF, the output function of the neuronal update equation in Eq. (1) is modified to the following form,

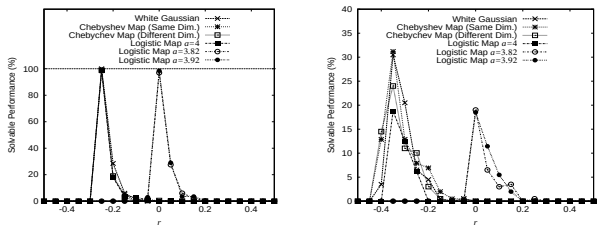
$$y_{ij}(t+1) = \sum_{k=1}^{N}\sum_{l=1}^{N} w_{ijkl}x_{kl}(t) - \theta_{ij} + \beta z_{ij}(t), \tag{6}$$

$$\hat{y}_{ij}(t+1) = r\hat{y}_{ij}(t) + y_{ij}(t+1), \tag{7}$$

$$x_{ij}(t+1) = 1/(1+\exp(-\hat{y}_{ij}(t+1)/\epsilon)). \tag{8}$$

In the above equations, Eq. (7) is corresponding to the LSF, and modifies the autocorrelation of the internal state of each neuron.

The solvable performances of this modified neural networks is shown in Fig. 3, with changing the parameter r of the LSF in Eq. (7). By comparing with the results in Fig. 1, we can see that the results using the noise, which has white autocorrelation, could be much improved, and their performances become the same level as the negative autocorrelation chaotic noise for the TSP, and are improved more than 30% better than the original chaotic method for the QAP. For such improvements in white autocorrelation noise cases, the negative autocorrelation parameter r around $-2 + \sqrt{3} \approx -0.268$ has the best performances. These results confirm the theories of the solution search ability described in previous section that the negative autocorrelation for each axis makes the lowest cross-correlation among the axes of the searching dynamics and that makes ideal distributive searching dynamics.



(a)TSP (b)QAP

Figure 3: Relation between the solvable performances of the mutually connected neural network with the LSF and autocorrelation parameter.

For more clearly verifying the theory, the relation between the autocorrelation coefficient of each axis (neuron) and the asynchronous cross-correlation among the axes (neurons) is shown in Fig. 4. The autocorrelation and the cross-correlation are calculated using $\hat{y}_{ij}(t)$, which is the output of the LSF. The autocorrelation is the value at the lag 1, which corresponds to λ. The cross-correlation is the average of all combinations of the axes. From the Fig. 4, it becomes clearly that the negative autocorrelation around $-2 + \sqrt{3}$ makes the asynchronous cross-correlation minimum. This minimum cross-correlation makes the algorithm the most distributive and improves the performance of the solution search algorithm.

5. Application of Ideal Search Generation Method to 2-opt for Large TSPs

In this section, the proposed approach is applied to the 2-opt method for the TSPs, which is applicable to much larger combinatorial optimization problems than the Hopfield-Tank neural networks. Although the solution space does not corresponds to the state of the algorithm as in the mutually connected neural network, the 2-opt have much better performance and more useful. Since the 2-opt is also asynchronously updated heuristic algorithm, the negative autocorrelation dynamics will improve the performance as well.

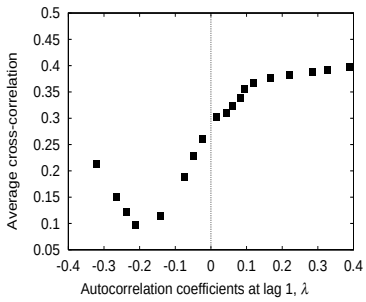


Figure 4: Relation between the autocorrelation coefficients at lag 1 and the average cross-correlation in the searching dynamics of the mutually connected neural networks with the LSF.

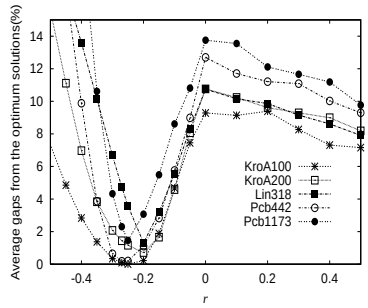


Figure 5: Relation between autocorrelation parameter of the LSF applied to the 2-opt method and its performances on five TSPs.

In order to introduce the LSF to the 2-opt, first the update equation of the 2-opt method is defined as the following equations,

$$s_{ij}(t+1) = \Delta_{ij}(t), \tag{9}$$

where $\Delta_{ij}(t)$ are the difference of the tour length decreased by the 2-opt exchange, which connects the city i and the city j, at time t, and $s_{ij}(t)$ is the corresponding internal state, respectively. When $s_{ij}(t) > 0$, the corresponding 2-opt exchange is really applied and the current solution is updated.

To apply the LSF to the 2-opt method, the LSF is introduced to the Eq. (9) as follows, with setting $M \to \infty$,

$$s_{ij}(t+1) = \sum \tau = 0^{M} r^{\tau} \Delta_{ij}(t-\tau) = r s_{ij}(T) + \Delta i j(t), \tag{10}$$

By making the decisions of the 2-opt exchange using this $s_{ij}(t)$, each asynchronous decision can be modified to have the negative autocorrelation, and the cross-correlation among the operations can be minimized and the ideal distributive search is realized also for the 2-opt method.

The relation between the autocorrelation tuning parameter r and the solvable performance are shown in Fig. 5. It shows the results on five problems whose sizes are from 100 to 1173 from TSPLIB [14]. The performances are evaluated by the gaps between the average obtained solutions and the known optimum solutions. To obtain average solution, 100 different initial conditions are used. Each run is cut at 10000 iterations. From Fig. 5, the results of the 2-opt with LSF around $-2 + \sqrt{3}$ are the best for all problems. The gaps are almost 0 % even for large-scale problems and the performance was better than the 2-opt with stochastic additive noise. From those results, we confirm that the proposed approach using the LSF to generate negative autocorrelation much improves the performances of the asynchronously updated heuristic methods.

6. Conclusion

This paper showed the performance of the proposed scheme using ideally distributive searching dynamics for asynchronously updated heuristic algorithms. By using negative autocorrelation dynamics for each asynchronous heuristic operation of the solution update algorithm, cross-correlation among the operations can be minimized, and ideally distributive solution search dynamics can be realized. The results clearly showed improvement of the proposed scheme also for the larger problems.

Our future work is to verify the effectiveness of the proposed approach on various asynchronously updated heuristic algorithms. We will make clear proof of the effectiveness of the proposed scheme and show the strong advantage of the proposed approach.

References

[1] N. Metropolis et al., *J. Chem. Physics*, 21, 1087, 1953.

[2] F. Glober et al., *Annals. of Oper. Res.*, 41, 3–28, 1993.

[3] H. Nozawa, *Chaos*, 2, 377–386, 1992.

[4] M. Hasegawa et al., *Physical Review Letters*, 79, 2344–2347, 1997..

[5] M. Hasegawa et al., *Neural Networks*, 15, 111–123, 2002.

[6] M. Hasegawa et al., *Euro. J. Oper. Res.*, 139, 543–556, 2002.

[7] M. Hasegawa et al., *IEICE Trans.* E80-A, 206–213, 1997.

[8] G. Mazzini et al., *Electronics Letters*, 35, 1054, 1999.

[9] T. Kohda et al., Proc. IEEE ISCAS, V, 225–228, 2000.

[10] J. J. Hopfield and D. W. Tank, *Biological Cybernetics*, 52, 141–152, 1985.

[11] M. Hasegawa et al., *IEICE Trans. Commun.*, E91-B, 110–118, 2008.

[12] R. Burkard et al., *J. Global Optimization*, 10, 391–403, 1997.

[13] K. Umeno and A. Yamaguchi, *IEICE Trans.*, E85-A, 849–852, 2002.

[14] G. Reinelt, *ORSA J. on Computing*, 3, 376–384, 1991.