A Markov Chain Monte Carlo Algorithm for the Quadratic Assignment Problem Based on Replicator Equations

Takehiro Nishiyama, Kazuo Tsuchiya, and Katsuyoshi Tsujita

Dept. of Aeronautics and Astronautics, Graduate School of Engineering,
Kyoto University, Kyoto, Japan
{nisiyama, tsuchiya, tsujita}@kuaero.kyoto-u.ac.jp

Abstract. This paper proposes an optimization algorithm for the Quadratic Assignment Problem (QAP) based on replicator equations. If the growth rate of a replicator equation is composed of the performance index and the constraints of the QAP suitably, by increasing the value of the control parameter in the growth rate, the equilibrium solutions which correspond to the feasible solutions of the QAP become stable in order from the one with smaller value of the performance index. Based on the characteristics of the system, the following optimization algorithm is constructed; the control parameter is set so that the equilibrium solutions corresponding to the feasible solutions with smaller values of the performance index become stable, and then in the solution space of the replicator equations, a Markov chain Monte Carlo algorithm is carried out. The proposed algorithm is applied to many problem instances in the QAPLIB. It is revealed that the algorithm can obtain the solutions equivalent to the best known solutions in short time. Especially, for some large scale instances, the new solutions with the same cost as the best known solutions are obtained.

1 Introduction

The Quadratic Assignment Problem (QAP) [1] is one of the hardest combinatorial optimization problems. The QAP is formulated as a problem to find a N dimensional permutation matrix which minimizes a performance index, where N is the size of the problem. Among approximation methods for the QAP, there are dynamical systems approaches; a dynamical system consisting of the elements of a $N \times N$ matrix is constructed. The mutual interactions between the elements are determined so that equilibrium solutions of the system become permutation matrices with smaller values of the performance index, i.e. approximate solutions for the QAP. In many studies, the dynamical system is constructed as a gradient system [2, 3]; a potential function is composed of the performance index and the constraints, and the dynamical system is constructed as a gradient vector field of the potential function. The system has equilibrium solutions as the minima of the potential function which correspond to the approximate solutions of the QAP. On the other hand, we have constructed the dynamical

system as a replicator equation [4]. The replicator equation is the equation in which derivatives of the variables are proportional to the state of the variables. The proportional coefficients are called growth rates. When the growth rates are determined based on the performance index and the constraints of the QAP suitably, the system has the following characteristics; all feasible solutions of the QAP are the equilibrium solutions of the system. By increasing the value of the parameter (control parameter) in the growth rate, the feasible solutions become stable in order, from the one having the smallest value of the performance index to the largest one. This means that when the control parameter is set suitably, the dynamical system has only solutions with smaller values of the performance index as the stable solutions. In this paper, we propose the following Markov chain Monte Carlo algorithm based on the characteristics of the replicator equations; the growth rates of the replicator equations are designed based on the performance index and the constraints of the QAP. The control parameter in the growth rate is set so that the equilibrium solutions corresponding to the feasible solutions with smaller values of the performance index become stable. The replicator equations are calculated with some initial values to obtain an equilibrium solution. Then, setting the initial values in some neighborhood of the solution, the replicator equations are calculated again to obtain the next equilibrium solution. The obtained solution is accepted according to some probability. This procedure is repeated to give a sequence of solutions. The proposed algorithm is applied to some problem instances in the QAPLIB [5], and in many cases, gives solutions comparable to the best known solutions. Especially, for some large scale instances, the proposed algorithm gives new solutions having the same values of the performance index as the best known solutions. Haken et al. have proposed an optimization algorithm based on the replicator equations [6]. But in their method, the above-mentioned characteristics of the equilibrium solutions of the replicator equations are not used explicitly. On the other hand, Ishii and Niitsuma have proposed a dynamical systems approach in which the search space is restricted [7]. But their method does not utilize the characteristics that the search space is composed of the good approximate solutions of the QAP which our method utilizes to improve the performance.

2 Quadratic Assignment Problem (QAP)

The Quadratic Assignment Problem (QAP) [1] is considered one of the hardest combinatorial optimization problems. Given a set $\mathcal{N} = \{1, 2, \cdots, N\}$ and $N \times N$ matrices $A = (a_{ij}), B = (b_{kl})$, the QAP is defined as follows:

$$\min_{p \in \Pi_{\mathcal{N}}} L(p), \quad L(p) = \sum_{i,j} a_{ij} b_{p(i)p(j)}, \tag{1}$$

where $\Pi_{\mathcal{N}}$ is the set of all permutations of $\mathcal{N}$, and p is an element of it. Letting $\Pi_{N \times N}$ the set of all $N \times N$ permutation matrices and $X = (x_{ij})$ an element of

it, the QAP is also represented as the following matrix form:

$$\min_{X \in \Pi_{N \times N}} L(X), \quad L(X) = \mathrm{trace}(A^T X^T B X) = \sum_{i,i',j,j'} a_{jj'} b_{ii'} x_{ij} x_{i'j'}. \tag{2}$$

A typical example of the QAP is the facility location problem; consider assigning N facilities to N locations, where a_{ij} represents the flow of materials from facility i to facility j and b_{kl} is the distance from location k to location l. The cost of assigning facility i to location k and facility j to location l is $a_{ij}b_{kl}$. The objective of the problem is to find an assignment of all facilities to all locations such that the total cost is minimized.

3 Proposed Dynamical System and Its Characteristics [4]

For the QAP, we have proposed the following replicator equation:

$$\dot{u}_{ij} = f_{ij}(u_{i'j'}, \alpha_0, \alpha_1) u_{ij}, \tag{3a}$$

$$f_{ij} = (1 - u_{ij}^2) - \frac{\alpha_0}{2}\Big(\sum_{i' \neq i} u_{i'j}^2 + \sum_{j' \neq j} u_{ij'}^2\Big) - \frac{\alpha_1}{2} \sum_{i',j'} (a_{jj'} b_{ii'} + a_{j'j} b_{i'i}) u_{i'j'}^2 \tag{3b}$$

$$(i, j = 1, \cdots, N),$$

where f_{ij} is called the growth rate, and the parameters are $\alpha_0 > 0$, $0 \leq \alpha_1 \ll 1$. The first term of the growth rate f_{ij} leads each u_{ij}^2 to unity. The second term represents the competition between elements having same subscripts i (j), and the parameter α_0 determines the strength of the competition. The third term is derived from the gradient of the performance index:

$$\frac{1}{2} \frac{\partial L(U)}{\partial u_{ij}} = \sum_{i',j'} (a_{jj'} b_{ii'} + a_{j'j} b_{i'i}) u_{i'j'}^2 u_{ij}, \tag{4}$$

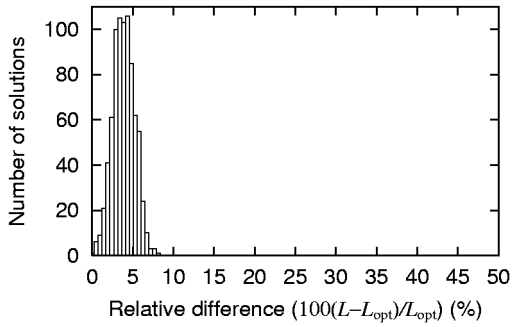
where $U = (u_{ij}^2)$, and suppresses the solutions having larger values of the performance index.

The dynamical system (3) has equilibrium solutions $u_{ij}^{(p)}$ $(p \in \Pi_{\mathcal{N}})$:

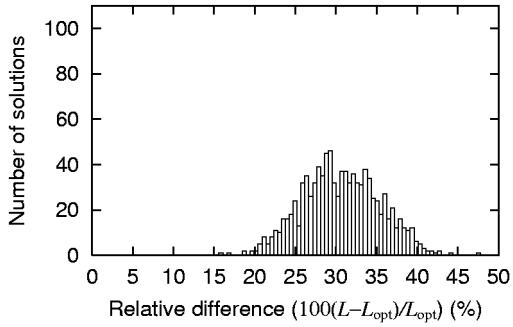
$$u_{ij}^{(p)2} \begin{cases} \simeq 1 \ (i = p(j)) \\ = 0 \ (i \neq p(j)) \end{cases} (\forall i). \tag{5}$$

The solutions $U^{(p)} = (u_{ij}^{(p)})$ are called the feasible solutions which corresponds to permutation matrices $X^{(p)} = (x_{ij}^{(p)})$:

$$x_{ij}^{(p)} \begin{cases} = 1 \ (i = p(j)) \\ = 0 \ (i \neq p(j)) \end{cases} (\forall i). \tag{6}$$



(a) $\alpha_0 = 1.01$



(b) $\alpha_0 = 3.0$

Fig. 1. Distribution of the feasible solutions ($L_{\rm opt}$ is the optimal value of L)

The stability condition for a feasible solution $U^{(p)}$ corresponding to a permutation matrix $X^{(p)}$ is approximately given as follows:

$$\alpha_0 > 1 + \frac{\alpha_1}{N-1}(L(X^{(p)}) - \bar{L}), \quad \bar{L}\text{: constant} \tag{7}$$

The condition (7) indicates that when α_0 is close to 1, only feasible solutions having smaller values of the performance index L, i.e. good approximate solutions of the QAP, are stable.

To verify the stability condition (7), the dynamical system (3) is computed with many sets of random initial values using a problem instance called "Nug20" ($N = 20$) in the QAPLIB [5]. Figures 1 (a), (b) are the results for $\alpha_0 = 1.01$ and $\alpha_0 = 3.0$ respectively. In the case of $\alpha_0 = 1.01$, only good solutions are obtained as compared with the case of $\alpha_0 = 3.0$.

1. $M :=$ size of the neighborhood
2. $T_0 :=$ initial temperature
3. $X^{(p_0)} :=$ initial permutation matrix
4. $n := 0$ (iteration)
5. $\alpha_0 := 1 + \epsilon$ ($\epsilon \ll 1$)
6. **while** $n < n_{\max}$ **do**
   - 6.1. Choose M rows and columns randomly in the matrix U
   - 6.2. Give random initial values to the corresponding $M \times M$ elements u_{ij} where the rest of the elements are fixed to the value of $U^{(p_n)}$
   - 6.3. Calculate (3) to obtain the equilibrium solution $U^{(p_{n+1})}$ and corresponding permutation matrix $X^{(p_{n+1})}$. If the solution is not feasible, go back to Step 6.1
   - 6.4. Apply 2-opt method to $X^{(p_{n+1})}$
   - 6.5. Accept $X^{(p_{n+1})}$ with probability $e^{-[L(X^{(p_{n+1})})-L(X^{(p_n)})]^+/T_n}$, where $[a]^+ = \max\{a, 0\}$. If rejected, $X^{(p_{n+1})} := X^{(p_n)}, U^{(p_{n+1})} := U^{(p_n)}$
   - 6.6. $T_{n+1} = b \cdot T_n$ ($b < 1$)
   - 6.7. $n := n + 1$

Fig. 2. Algorithm

4 Optimization Algorithm

Based on the above analysis, we propose an optimization algorithm for the QAP using the Markov chain Monte Carlo algorithm.

First, the search space is constructed as follows; the replicator equation (3) is derived according to the given QAP. The control parameter α_0 is set close to 1 so that (3) has stable equilibrium solutions $U^{(p)}$ corresponding to permutation matrices $X^{(p)}$ with smaller values of the performance index of the QAP. The search space is composed of these stable equilibrium solutions $U^{(p)}$. Next, given a solution $U^{(p_n)}$ corresponding to the permutation matrix $X^{(p_n)}$, new solution $U^{(p_{n+1})}$ is searched in the 'M-neighborhood' of the current solution as follows; M rows and columns are randomly chosen in the matrix U of (3). Random values are given to the corresponding elements u_{ij} as the initial values and then (3) is calculated, where the rest of the elements are fixed to the values of $U^{(p_n)}$. The new solution $U^{(p_{n+1})}$ and the corresponding permutation matrix $X^{(p_{n+1})}$ are obtained. A local search method called the 2-opt method is applied to the obtained solution. The 2-opt method is a simple heuristic method and adopted to slightly modify the solution to be the local minimum in the 2-neighborhood of the solution. The new solution $X^{(p_{n+1})}$ is accepted or rejected based on the Metropolis method [8]; if the performance index value is decreased by the change of the solution, the new solution is accepted, and if the value is increased, the new solution is accepted with the probability $\exp(-\Delta L/T)$ according to the amount of the increase ΔL. The parameter T called the temperature is decreased every step of the algorithm by multiplying a constant b (< 1). The whole procedure of the proposed algorithm is shown in Fig. 2.

Table 1. Solutions of the proposed algorithm (%)

M	Mean	Standard deviation	Minimum
5	0.014	0.022	0
10	0.0021	0.0019	0
15	0.015	0.027	0

Table 2. Solutions of the proposed algorithm with dynamics and without dynamics (random) (%)

	Mean	Standard deviation	Minimum
Dynamics	0.0021	0.0019	0
Random	0.015	0.017	0

This algorithm is not a true simulated annealing [9] in two respects. First, since the new solution is searched in the M-neighborhood of the current solution among the stable equilibrium solutions of (3), the ergodicity does not always hold. Second, since the obtained solution is modified by the 2-opt method in each step of the algorithm, the detailed balance is not satisfied. Therefore, the convergence properties of the proposed algorithm are not guaranteed. But, numerical experiments mentioned below reveal that the algorithm can obtain good solutions for many instances in the QAPLIB.

5 Numerical Experiments

Numerical experiments are carried out using some large scale problem instances in the QAPLIB. First, the effect of the size M of the neighborhood was checked using an instance "Wil100" ($N = 100$). The parameters are $\alpha_0 = 1.01, \alpha_1 = 0.003, b = 0.99995, n_{\max} = 50000$, and $T_0 = 300.0$. The resulted means, standard deviations and minima of ten separate trials for each of $M = 5, 10, 15$ are shown in Table 1. These values are the relative differences $100(L - L_{\text{opt}})/L_{\text{opt}}$ (%) from the best known solution L_{opt}. The best result was obtained when $M = 10$. The reason is considered that good parts of the solution may be changed when a new solution is obtained if M is too large. But it is necessary to consider further about the optimum value of M. Next, we compared the proposed method with the method where the new solution is randomly generated in the neighborhood of the current solution. The computation was carried out using the problem instance "Wil100" with $M = 10$. The results are shown in Table 2. The proposed algorithm gives better performance on average. This indicates that the proposed method searches effectively in the relatively large neighborhood using the dynamical system. Finally, the performance of the proposed algorithm was verified using some large scale problem instances in the QAPLIB, i.e. "Sko100a", "Sko100b",

Table 3. Performance

Name	N	$L_{\rm opt}$ (algorithm)	L	Difference(%)
Sko100a	100	152002 (GEN)	152002	0
Sko100b	100	153890 (GEN)	153890	0
Sko100f	100	149036 (GEN)	149036	0
Tai100a	100	21125314 (Re-TS)	21146176	0.099
wil100	100	273038 (GEN)	273038	0
Tho150	150	8133484 (SIMJ)	8135474	0.024

"Sko100f", "Wil100", "Tai100a" ($N = 100$) and "Tho150" ($N = 150$). The results are shown in Table 3, where $L_{\rm opt}$ is the best known solution given in the QAPLIB and in the parentheses the name of the algorithms which gave the solutions, i.e. Genetic Hybrids (GEN), Reactive Tabu Search (Re-TS), Simulated Jumping (SIMJ), are shown. L is the solution by the proposed algorithm and in the last column the relative difference from $L_{\rm opt}$ is shown. The solutions having the same values of the performance index as the best known solutions were obtained for the four of the six problem instances. [1]

In the proposed method, since the size of the neighborhood is set to $M(\sim 10)$, the dynamical system with only $M \times M$ elements is calculated even if the size of the problem is $N(\sim 100)$. Therefore the computation time for one step of the algorithm is very short. On a COMPAQ AlphaStation XP900 computer, the total computation time for 50000 steps of the algorithm was only about 1–2 hours.

6 Conclusion

In this paper, we proposed a Markov chain Monte Carlo algorithm based on the replicator equations. In the proposed dynamical system, only good approximate solutions are obtained by appropriately setting a control parameter in the system. Therefore, using the system, good solutions are efficiently searched in relatively large neighborhood in each step of the algorithm. The proposed algorithm is applied to some large scale benchmark problems of the QAP. It was shown that the algorithm can obtain the solutions equivalent to the best known solutions in short time. Especially, the new solutions with the same performance index values as the best known solutions were obtained for some problem instances.

References

1. Pardalos, P. M., Rendl, F. and Wolkowicz, H.: The Quadratic Assignment Problem: A Survey and Recent Developments. In: Pardalos, P. and Wolkowicz, H. (eds.):

[1] The obtained permutation matrices were different from the ones given in the QAPLIB.

Quadratic Assignment and Related Problems, Vol. 16 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society (1994) 1–42
2. Hopfield, J. J. and Tank, D. W.: "Neural" Computation of Decisions in Optimization Problems. Biological Cybernetics **52** (1985) 141–152
3. Ishii, S. and Sato, M.: Constrained Neural Approaches to Quadratic Assignment Problems. Neural Networks **11** (1998) 1073–1082
4. Tsuchiya, K., Nishiyama, T. and Tsujita, K.: A Deterministic Annealing Algorithm for a Combinatorial Optimization Problem Using Replicator Equations. Physica D **149** (2001) 161–173
5. Burkard, R. E., Karisch, S. and Rendl, F.: QAPLIB – A Quadratic Assignment Problem Library. Journal of Global Optimization **10** (1997) 391–403
6. Haken, H., Schanz, M. and Starke, J.: Treatment of Combinatorial Optimization Problems using Selection Equations with Cost Terms - Part I: Two-Dimensional Assignment Problems. Physica D **134** (1999) 227–241
7. Ishii, S. and Niitsuma, H.: λ-opt Neural Approaches to Quadratic Assignment Problems. Neural Computation **12** (2000) 2209–2225
8. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E.: Equation of state calculation by fast computing machines. Journal of the Chemical Physics **21** (1953) 1087–1092
9. Kirkpatrick, S., Gelatt, C. D., Jr. and Vecchi, M. P.: Optimization by Simulated Annealing. Science **220** (1983) 671–680