

TR-H-258

**λ - Opt Neural Approaches to Quadratic
Assignment Problem**

**Hiroataka NIITSUMA (NAIST) and
Shin ISHII (NAIST/ATR-HIP)**

1998.11.16

ATR人間情報通信研究所

〒619-0288 京都府相楽郡精華町光台2-2 TEL: 0774-95-1011

ATR Human Information Processing Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

Telephone: +81-774-95-1011

Fax : +81-774-95-1008

λ -opt neural approaches to quadratic assignment problem

Hiroataka Niitsuma†

Shin Ishii†‡

† Nara Institute of Science and Technology

8916-5 Takayama-cho, Ikoma-shi, Nara, 630-0101 Japan

‡ ATR Human Information Processing Research Laboratories

2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

(TEL) +81-743-72-5251 (FAX) +81-743-72-5259

(E-mail) ishii@is.aist-nara.ac.jp

Abstract

In this paper, we propose new analog neural methods to combinatorial optimization problems, in particular, quadratic assignment problem. Our proposed methods are based on an analog version of the λ -opt heuristics, which simultaneously changes assignments for λ elements in a permutation. Since we can take a relatively large λ value, our new methods can achieve a middle-range search over the possible solutions, and this helps the system neglect shallow local minima and escape from local minima. In experiments, we have applied our methods to relatively large-scale ($N = 80 \sim 150$) QAPs. Results have shown that our new methods are comparable to the present champion algorithms, and for two benchmark problems, they are able to obtain better solutions than the previous champion algorithms.

1 Introduction

In this paper, we propose new analog neural methods to combinatorial optimization problems. In particular, we deal with quadratic assignment problem (QAP) (Burkard, Karisch and Rendl, 1991), which is known to be a very difficult combinatorial optimization problem. However, our proposed approaches can be applied to various combinatorial optimization problems where each solution is represented as a permutation. An example is traveling salesman problem (TSP).

In the case of TSP, a solution is represented by aligning the city indices along the salesman's route, which is a permutation of the set of city indices. A solution of QAP can also be represented by a permutation. In the conventional neural approaches, on the other hand, an Ising (binary) spin system (Hopfield and Tank, 1985; Bilbro, Mann, Miller, Snyder, Van den Bout and White, 1989) or a Potts spin system (Peterson and Söderberg, 1989; Ishii and Sato, 1997) has been used to represent a solution. The constraints for the permutation are then implemented as "soft" constraints, namely, penalty terms for violations are added to the objective function. This often produces infeasible solutions, which is one of the reasons why these approaches are not very good when the problem scale becomes large.

We (Ishii and Sato, 1996) previously proposed a neural optimization method called the doubly constrained network (DCN), in which all of the permutation constraints are treated as hard. Namely, the space of possible spin configurations is almost equivalent to that of the permutations. The obtained solution is therefore feasible in general, and it is better than the solution that would be produced by the Ising or Potts spin approach. Even in DCN, however, several problems exist: (1) Since the algorithm is deterministic, the system can not escape from a local minimum by itself. In addition, there is no solution variety even if the initial conditions are variously prepared; and (2) In order to obtain a proper permutation, a careful deterministic annealing is actually needed. These features prevent our approach from achieving further improvements in the solution.

In our new approach, we apply a replacement to a permutation as a basic operation. In addition, we introduce non-equilibrium dynamics to DCN. These modifications overcome the above-mentioned problems. The non-equilibrium dynamics makes the system to escape from local minima and it results in a solution variety. The basic replacement operation ensures a solution to be a valid permutation.

When the basic operation is an exchange of two elements in a permutation, it is called "2-opt" search. When λ (> 2) elements in a permutation are changed at once, the operation is called " λ -opt" search (Lin and Kernighan, 1973; Martin, Otto and Feltman, 1992). The 2-opt search is a very simple heuristic algorithm, but it easily falls into a local minimum. A neural network approach based on the 2-opt heuristics was proposed by Hasegawa et al. (Hasegawa, Ikeguchi and Aihara, 1998). In order to escape from local minima, tabu list (Taillard, 1995) and chaos (Hasegawa, Ikeguchi and Aihara, 1998) have been proposed. On the other hand, λ -opt ($\lambda > 2$) heuristics get computationally heavy, as λ becomes large.

Our proposed methods are based on an analog version of the λ -opt heuristics. Since we employ analog neural approaches based on our DCN formulation, it is possible to set λ as large as $15 \sim 60$, and this enables the system to search for good permutations over a middle-range region one after another. In addition, the relatively large λ values prevent the system from falling into shallow local minima around the present system state.

In experiments, we apply our new methods to relatively large-scale ($N = 80 \sim 150$) QAPs

taken from QAPLIB (Burkard, Karisch and Rendl, 1991), which is a standard set of QAP benchmark problems. Our new approaches are comparable to the present champion algorithms, which are based on tabu search (Taillard, 1991; Battiti and Tecchiolli, 1994), genetic algorithms (Fleurent and Ferland, 1994), and simulated annealing (Amin, 1998). Moreover, for two benchmark problems, the new approaches are able to better the champion algorithms. Consequently, our methods can be considered as one of the stronger algorithms for QAP.

2 λ -DCN

2.1 Quadratic assignment problem

Quadratic assignment problem (QAP) is known as a very difficult combinatorial optimization problem (Burkard, Karisch & Rendl, 1991). Naturally, it belongs to the class of NP-hard problems. A typical instance of QAP is a facility location problem, in which a set of facilities are to be assigned to an equal number of locations at the minimal cost. For each pair of facilities, an amount of flow is given, and for each pair of locations, a distance is given. The cost is defined as the summation of the product of the flow between two facilities and the distance between the locations to which the facilities are assigned. Many combinatorial optimization problems, such as the traveling salesman problem, maximum clique problem, and graph isomorphism problem, are special and easy cases of QAP.

Let \mathbf{D} and \mathbf{F} denote an $(N \times N)$ -dimensional distance and flow matrices, respectively. QAP with a problem size N is defined by

$$\min_{p \in \Pi} \sum_{a=1}^N \sum_{b=1}^N D_{a,b} F_{p(a),p(b)}, \quad (1)$$

where Π is the set of all permutations of $\{1, 2, \dots, N\}$ and $p(a)$ gives the element assigned to the location a in a permutation $p \in \Pi$.

2.2 λ -DCN equations

In order to deal with problem (1), we define an $(N \times N)$ -dimensional assignment matrix: $S_{a,n} = 1$ if $p(a) = n$ and $S_{a,n} = 0$ otherwise. In order for the assignment matrix \mathbf{S} to represent a valid permutation $p(a)$, the constraints: $\sum_{n=1}^N S_{a,n} = 1$ ($a = 1, \dots, N$) and $\sum_{a=1}^N S_{a,n} = 1$ ($n = 1, \dots, N$) should be satisfied. By using the assignment matrix, problem (1) can be defined as a minimization of an objective energy function:

$$E_{obj}(\mathbf{S}) = \sum_{a,b,n,m=1}^N D_{a,b} F_{n,m} S_{a,n} S_{b,m}, \quad (2)$$

subject to the assignment matrix constraints. It should be noted that if the facility matrix is of the form: $F_{n,m} = (\delta_{n,(m+1)} + \delta_{n,(m-1)})/2$, objective function (2) is equivalent to that of an N -city TSP. $\delta_{i,j}$ is Kronecker's delta. Namely, TSP is a special case of QAP.

Next, we define a permutation matrix by

$$Y_{a,b} = \begin{cases} 1 & \text{if element } p(b) \text{ changes its location to } a \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Here, although an assignment matrix \mathbf{S} can be regarded as a permutation matrix applied to the identity assignment, $p(n) = n$ ($n = 1, \dots, N$), we distinguish assignment matrices and permutation matrices for description convenience. For the permutation matrix, there are also the following constraints:

$$\sum_{a=1}^N Y_{a,b} = 1 \quad (b = 1, \dots, N) \quad (4a)$$

$$\sum_{b=1}^N Y_{a,b} = 1 \quad (a = 1, \dots, N) \quad (4b)$$

We also consider the following constraint:

$$\sum_{a=1}^N Y_{a,a} = N - \lambda, \quad (5)$$

where $0 \leq \lambda \leq N$ is a constant integer. Constraint (5) restricts the number of elements whose location is changed by the permutation \mathbf{Y} to λ . In the following, we call a permutation matrix \mathbf{Y} subject to constraints (4) and (5) a λ -permutation matrix. When a permutation matrix \mathbf{Y} applies to an assignment matrix \mathbf{S} , the new assignment matrix \mathbf{S}' is given by

$$\mathbf{S}' = \mathbf{Y} \mathbf{S}. \quad (6)$$

Next, we consider a set of λ -permutation matrices: $\Lambda = \{(\mathbf{Y}^k, P^k) | k = 1, \dots\}$, where P^k is the probability for the permutation \mathbf{Y}^k . The ensemble average for the new assignment matrices applied by this set of permutation matrices is given by

$$\langle \mathbf{S}' \rangle_{\Lambda} = \langle \mathbf{Y} \rangle_{\Lambda} \mathbf{S}, \quad (7)$$

where $\langle \cdot \rangle_{\Lambda}$ is the mean with respect to the permutation set, namely, $\langle f(k) \rangle_{\Lambda} \equiv \sum_k f(k) P^k$. We use the notations: $\mathbf{V} \equiv \langle \mathbf{S}' \rangle_{\Lambda}$ and $\mathbf{X} \equiv \langle \mathbf{Y} \rangle_{\Lambda}$.

Let us now consider a minimization of the objective energy function, $E_{obj}(\mathbf{S}')$, with respect to the new assignments. In order to make this discrete problem a continuous one, we introduce a free energy function:

$$F = \langle E_{obj}(\mathbf{S}') \rangle_{\Lambda} - TH \quad (8a)$$

$$H \equiv - \sum_{\mathbf{S}'} P(\mathbf{S}') \log P(\mathbf{S}') \approx - \sum_{a,n=1}^N V_{a,n} \log V_{a,n}, \quad (8b)$$

where $P(\cdot)$ denotes the probability for that configuration. T is the temperature and H is the entropy of the possible configurations.

Here, we assume the objective energy function has a quadratic form:

$$E_{obj}(\mathbf{S}') = \frac{1}{2} \sum_{a,n,b,m=1}^N W_{a,n;b,m} S'_{a,n} S'_{b,m} + \sum_{a,n=1}^N I_{a,n} S'_{a,n}, \quad (9)$$

where the weight matrix is symmetric, i.e., $W_{a,n;b,m} = W_{b,m;a,n}$ ($a, n, b, m = 1, \dots, N$). In this case, we can conduct the following approximation:

$$\begin{aligned} \langle E_{obj}(\mathbf{S}') \rangle_{\Lambda} &\approx \frac{1}{2} \sum_{a,n,b,m=1}^N W_{a,n;b,m} V_{a,n} V_{b,m} + \sum_{a,n=1}^N I_{a,n} V_{a,n} \\ &= \frac{1}{2} \sum_{a,b,c,d=1}^N \left(\sum_{n,m=1}^N W_{a,n;b,m} S_{c,n} S_{d,m} \right) X_{a,c} X_{b,d} + \sum_{a,b=1}^N \left(\sum_{n=1}^N I_{a,n} S_{b,n} \right) X_{a,b} \\ &= \frac{1}{2} \sum_{a,b,c,d}^N W_{a,b;c,d=1}^*(\mathbf{S}) X_{a,b} X_{c,d} + \sum_{a,b=1}^N I_{a,b}^*(\mathbf{S}) X_{a,b} \equiv E^*(\mathbf{X}; \mathbf{S}) \end{aligned} \quad (10)$$

where $W_{a,b;c,d}^*(\mathbf{S}) \equiv \sum_{n,m} W_{a,n;c,m} S_{b,n} S_{d,m}$ and $I_{a,b}^*(\mathbf{S}) \equiv \sum_n I_{a,n} S_{b,n}$. The new weight matrix is also symmetric, i.e., $W_{a,b;c,d}^* = W_{c,d;a,b}^*$. In equations (8b) and (10), we use a similar approximation to the mean-field theory (Bilbro, Mann, Miller, Snyder, Van den Bout and White, 1989; Peterson and Söderberg, 1989), which is not rigorously applicable to the configuration space of the assignment matrices.

On the other hand, the entropy is given by

$$H \approx - \sum_{a,n=1}^N V_{a,n} \log V_{a,n} = - \sum_{a,n=1}^N X_{a,p^{-1}(n)} \log X_{a,p^{-1}(n)} = - \sum_{a,b=1}^N X_{a,b} \log X_{a,b} \equiv H^*(\mathbf{X}), \quad (11)$$

where p is the permutation represented by \mathbf{S} . Similarly, constraints (4) and (5) are also represented by the average permutation matrix \mathbf{X} .

Accordingly, the new continuous problem is defined by

$$\text{minimize } F^*(\mathbf{X}; \mathbf{S}) = E^*(\mathbf{X}; \mathbf{S}) - TH^*(\mathbf{X}) \quad (12a)$$

$$\text{subject to } \sum_{a=1}^N X_{a,b} = 1 \quad (b = 1, \dots, N) \quad (12b)$$

$$\sum_{b=1}^N X_{a,b} = 1 \quad (a = 1, \dots, N) \quad (12c)$$

$$\sum_{a=1}^N X_{a,a} = N - \lambda. \quad (12d)$$

The new problem is a minimization of the free energy function (12a), which is defined by the average permutation and not by the average assignment. In addition, the problem is not a global one. It is a local problem, depending on the previous assignment \mathbf{S} . In problem (12), the entropy (11), with constraints (12b) and (12c), prevents the average permutation \mathbf{X} from going out of its domain $[0, 1]^{N^2}$. In this sense, it functions as a barrier function (Luenberger, 1989). It should be noted that the constant λ is not necessarily an integer in this continuous problem.

Using the Lagrange method, a solution to problem (12) can be given by the following simultaneous equations.

$$X_{a,b} = \frac{U_{a,b}}{\alpha_a \beta_b \gamma^{\delta_{a,b}}} \quad (13a)$$

$$U_{a,b} \equiv \exp\left(-\frac{1}{T} \frac{\partial E^*(\mathbf{X}; \mathbf{S})}{\partial X_{a,b}}\right) \quad (13b)$$

$$\alpha_a = \sum_b \frac{U_{a,b}}{\beta_b \gamma^{\delta_{a,b}}} \quad (13c)$$

$$\beta_b = \sum_a \frac{U_{a,b}}{\alpha_a \gamma^{\delta_{a,b}}} \quad (13d)$$

$$\gamma = \frac{1}{N - \lambda} \sum_a \frac{U_{a,a}}{\alpha_a \beta_a}, \quad (13e)$$

where α_a ($a = 1, \dots, N$), β_b ($b = 1, \dots, N$) and γ are Lagrange multipliers. For the energy function (10), $\partial E^*/\partial X_{a,b} = \sum_{c,d} W_{a,b;c,d}^* X_{c,d} + I_{a,b}^*$. The equations defined by (13) are called λ -DCN equations.

2.3 Basic algorithm

In order to obtain solutions to λ -DCN equations, the following basic algorithm works well in a practical sense.

1. Set t at 0. Set $\mathbf{X}(0)$ to be

$$X_{a,b}(0) = \begin{cases} (1 - \lambda/N)(1 + \epsilon_{a,b}) & \text{if } a = b \\ \lambda(1 + \epsilon_{a,b})/(N(N - 1)) & \text{if } a \neq b \end{cases}, \quad (14)$$

where $\epsilon_{a,b}$ denotes small noise, e.g., a uniform random value in $[-0.1, 0.1]$.

2. For all a, b , calculate:

$$U_{a,b}(t+1) = \exp\left(-\frac{1}{T} \frac{\partial E^*(\mathbf{X}(t); \mathbf{S})}{\partial X_{a,b}(t)}\right). \quad (15)$$

3. Set α_a^{old} to be $\alpha_a(t)$ for all a , set β_b^{old} to be $\sum_a \frac{U_{a,b}(t+1)}{\alpha_a(t)\gamma(t)^{\delta_{a,b}}}$ for all b , and set γ^{old} to be $\gamma(t)$.

4. The following substeps are iterated.

- (a) For all a , calculate:

$$\alpha_a^{new} = \sum_{b=1}^N \frac{U_{a,b}(t+1)}{\beta_b^{old} (\gamma^{old})^{\delta_{a,b}}}. \quad (16)$$

- (b) For all b , calculate:

$$\beta_b^{new} = \sum_{a=1}^N \frac{U_{a,b}(t+1)}{\alpha_a^{new} (\gamma^{old})^{\delta_{b,a}}}. \quad (17)$$

- (c) Calculate:

$$\gamma^{new} = \frac{1}{N - \lambda} \sum_{a=1}^N \frac{U_{a,a}(t+1)}{\alpha_a^{new} \beta_a^{new}}. \quad (18)$$

If each of α_a, β_b , and γ converges, set $\alpha_a(t+1), \beta_b(t+1)$, and $\gamma(t+1)$ to those values. After that, rescale $\alpha(t+1)$ to $\sum_a \alpha_a(t+1) = 1$.

5. For all a, b , calculate:

$$X_{a,b}(t+1) = \frac{U_{a,b}(t+1)/(\beta_b(t+1)\gamma(t+1)^{\delta_{a,b}})}{\sum_c (U_{a,c}(t+1)/(\beta_c(t+1)\gamma(t+1)^{\delta_{a,c}}))}. \quad (19)$$

6. If $\mathbf{X}(t)$ converges, exit the algorithm. Otherwise, add 1 to t and go to step 2.

Like in DCN (Ishii and Sato, 1996), this algorithm does not necessarily converge. However, the free energy function (12a) is likely to decrease after a single process of the above basic algorithm.

2.4 λ -DCN algorithm

When the temperature T is small, the average permutation matrix \mathbf{X} obtained by the above basic algorithm is close to a vertex of the hypercube domain. Then, the matrix is regarded as signifying one of the λ -permutation matrices. Moreover, we can obtain several permutation matrices during a single process of the basic algorithm. Among these, we choose the best permutation matrix, i.e., the one that minimizes the objective function (9). This process retrieves a good new assignment matrix that is an application of one of the λ -permutation matrices to the previous assignment matrix \mathbf{S} .

Our proposed procedure, which is called λ -DCN, repeats the above-mentioned process. Figure 1 schematically shows this procedure. After a single process, a new assignment \mathbf{S}' which is transformed from the previous assignment \mathbf{S} by a λ -permutation \mathbf{Y} is retrieved. The new assignment \mathbf{S}' is expected to be improved so as to have a lower objective function value than the previous assignment. In the following process, another assignment \mathbf{S}'' is retrieved from \mathbf{S}' by calculating a good λ -permutation \mathbf{Y}' .

When the parameter λ is equal to 2, this procedure corresponds to an analog version of the 2-opt heuristics, which is a local search algorithm over the nearest neighbors. However, by taking a relatively large λ value, our λ -DCN algorithm can achieve a middle-range search over the possible assignments, and this helps the system neglect shallow local minima around the previous assignment, and search for good local minima of the objective energy function. It should be noted that this procedure never terminates, namely, non-equilibrium dynamics is introduced. In other words, the system is not trapped by any local minimum.

3 λ -interior DCN

Since the λ -DCN algorithm retrieves assignments using λ -permutations, it often ignores good local minima located in the nearby neighborhood around the current assignment.

In order to deal with this problem, we propose another method here, where the equality constraint (5) is replaced by the following inequality constraint:

$$\sum_{a=1}^N Y_{a,a} \geq N - \lambda, \quad (20)$$

which restricts the number of elements relocated by the permutation \mathbf{Y} to be smaller than λ . In order to implement this inequality constraint, the following barrier function is additionally introduced in the free energy function (12a):

$$K^*(\mathbf{X}) \equiv T \left(\sum_{a=1}^N X_{a,a} - M \right) \log \left(\sum_{a=1}^N X_{a,a} - M \right) + T(\theta - 1) \left(\sum_{a=1}^N X_{a,a} - M \right), \quad (21)$$

where $M \equiv N - \lambda$ and $\theta (= 0 \sim 1)$ is a constant parameter. In this case, constraint (12d) is removed.

Using barrier function (21), equations (13b) and (13e) in the DCN equations are replaced by

$$U_{a,b} = \exp \left(-\frac{1}{T} \frac{\partial E^*(\mathbf{X}; \mathbf{S})}{\partial X_{a,b}} - \theta \delta_{a,b} \right) \quad (22b)$$

$$\gamma = \frac{\sqrt{M^2 + 4 \sum_a \frac{U_{a,a}}{\alpha_a \beta_a}} - M}{2}. \quad (22e)$$

This derivation is described in the Appendix. Then, in the basic algorithm in Section 2.3, equation (15) is replaced by

$$U_{a,b}(t+1) = \exp \left(-\frac{1}{T} \frac{\partial E^*(\mathbf{X}(t); \mathbf{S})}{\partial X_{a,b}(t)} - \theta \delta_{a,b} \right), \quad (23)$$

and equation (18) is replaced by

$$\gamma^{new} = \frac{\sqrt{M^2 + 4 \sum_a \frac{U_{a,a}(t+1)}{\alpha_a^{new} \beta_a^{new}}} - M}{2}. \quad (24)$$

The other parts are the same as in the basic algorithm.

The modified basic algorithm searches for a good l -permutation where $0 \leq l \leq \lambda$ from the previous assignment \mathbf{S} . Therefore, it is considered that the new procedure achieves a local search and a middle-range search simultaneously, in the vicinity of the previous assignment. The new procedure is called the λ -interior DCN algorithm.

4 Experiments

The procedures above can be applied to any quadratic problem whose solutions are represented by permutations. In the case of QAP (1), we set $W_{a,n;b,m} = D_{a,b} F_{n,m}$ ($a, n, b, m = 1, \dots, N$) and $I_{a,n} = 0$ ($a, n = 1, \dots, N$). λ is set at $15 \sim 60$, depending on the problem.

The benchmark problems used in the experiments are ‘‘Tai80a’’ ($N = 80$), ‘‘Tai100a’’ ($N = 100$), ‘‘Wil100’’ ($N = 100$), and ‘‘Tho150’’ ($N = 150$), all of which are taken from QAPLIB (Burkard, Karisch and Rendl, 1991). The distance and facility matrices for these problems are symmetric and dense. ‘‘Tho150’’ is the largest problem with its dense matrices in QAPLIB. For ‘‘Tai80a’’ and ‘‘Tai100a’’, the champion algorithms (Taillard, 1991; Battiti and Tecchiolli, 1994) are based on tabu search. For ‘‘Wil100’’, the champion algorithm (Fleurent and Ferland, 1994)

is a combined algorithm of a genetic algorithm and tabu search. For “Tho150”, the champion algorithm (Amin, 1998) is called simulated jumping.

Table I shows the experimental results. The value in each column is a deviation in percentage from the champion data. For comparison, the table also contains solutions obtained by our previous approach (Ishii and Sato, 1998), i.e., DCN annealing, and the best solutions that the 2-opt heuristics could obtain from among many runs. With DCN annealing, λ -DCN, and λ -interior DCN, the obtained solutions were improved by the 2-opt heuristics. For λ -DCN and λ -interior DCN, the table also contains the application numbers of the basic algorithm until the best solutions are retrieved.

In Table I, we can see that the results of our new methods are very good. For “Wil100” and “Tho150”, the obtained solutions are comparable to the champion data. For “Tai80a” and “Tai100a”, both of the proposed methods achieved better solutions than the previous champion algorithms.

Table I : Obtained solutions

“Name” represents the problem’s name, “ N ” is the problem’s scale, “feas.sol” is the previous best solution, “algorithm” is the previous champion algorithm, “DCA” is the solution obtained by DCN annealing, “2-opt” is the best solution obtained by many 2-opt runs for various initial conditions, “ λ -DCN” is the solution obtained by our λ -DCN method, and “ λ -inter DCN” is the solution obtained by our λ -interior DCN method. Each solution is represented by the deviation in percentage from the previous best solution. A number in parentheses denotes the number of the basic algorithm applied to retrieve the best solution.

Name	N	feas.sol	algorithm	DCA	2-opt	λ -DCN	λ -inter DCN
Tai80a	80	13557864	robust tabu search	1.2%	3.1%	-0.060% (2568)	-0.040% (2481)
Tai100a	100	21125314	reactive tabu search	5.7%	3.2%	-0.015% (688)	-0.082% (5656)
Wil100	100	273038	genetic hybrids	0.27%	0.73%	0.46% (2109)	0.070% (2563)
Tho150	150	8133484	simulated jumping	0.33%	1.5%	0.24% (7139)	0.28% (6956)

5 Conclusion

When λ is relatively large, it is computationally difficult to search for a good assignment over all of the assignments reachable by λ -permutations from the current assignment. In this paper, we proposed a couple of neural methods that partially achieve this mechanism, by considering the free energy function dependent on the current assignment. They are λ -DCN and λ -interior DCN. The new methods are also improvements of our previously proposed DCN annealing in two points: (1) Since non-equilibrium dynamics is introduced, the system is never trapped by a local minimum, and the obtained solutions have variety. Accordingly, as the number of iterations becomes large, improvements in the solution can be expected; and (2) Since we take a permutation as a basic operation, the retrieved solutions always represent valid permutations. This is one of the reasons for why our basic algorithm practically works well.

In experiments, we applied our methods to relatively large-scale ($N = 80 \sim 150$) QAPs. Consequently, it was found that our new methods are comparable to the present champion

algorithms, and for two benchmark problems, they are better than the champion algorithms. We therefore conclude that our new methods are one of the stronger algorithms for QAP.

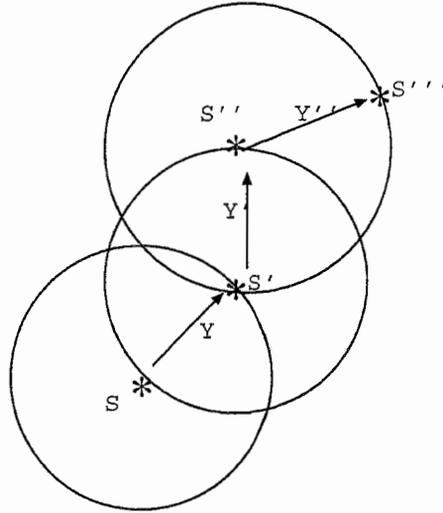


Figure 1

Schematic figure of the proposed λ -DCN procedure, which retrieves a new assignment S' from the previous assignment S . After that, the procedure retrieves another assignment S'' , and continues this process. S' is an assignment transformed from S by a λ -permutation Y .

References

- [1] Amin, S. (1998). Simulated jumping. *Annals of Operations Research*, to appear.
- [2] Battiti, R., & Tecchiolli, G. (1994). The reactive tabu search. *ORSA Journal on Computing*, **6**, 126-140.
- [3] Bilbro, G., Mann, R., Miller, T. K., Snyder, W. E., Van den Bout, D. E., & White, M. (1989). Optimization by mean field annealing. In *Advances in Neural Information Processing Systems I* (Denver 1988), D. S. Touretzky (ed.), 91-98. San Mateo: Morgan Kaufmann.
- [4] Burkard, R. E., Karisch, S. E., & Rendl, F. (1991). QAPLIB - A quadratic assignment problem library. *European Journal of Operational Research*, **55**, 115-119.
- [5] Fleurent, C., & Ferland, J. A. (1994). Genetic hybrids for the quadratic assignment problem. In *Quadratic assignment and related problems*, P. Pardalos, & H. Wolkowics (eds.), volume 16, 173-187. DIMACS Series in Discrete Mathematics and Theoretical Computer Science.

- [6] Hasegawa, T., Ikeguchi, M., & Aihara, K. (1998). Harnessing of chaotic dynamics for solving combinatorial optimization problems. In *Proceedings of the Fifth International Conference on Neural Information Processing*, (eds. S. Usui and T. Omori), **2**, pp. 749-752, Burke: IOS Press.
- [7] Hopfield, J. J., & Tank, D. W. (1985). "Neural" computations of decisions in optimization problems. *Biological Cybernetics*, **52**, 141-152.
- [8] Ishii, S., & Sato, M. (1996). Doubly constrained network for combinatorial optimization. *ATR Technical Report, TR-H-193*, Kyoto: ATR.
- [9] Ishii, S., & Sato, M. (1997). Chaotic Potts spin model for combinatorial optimization problems. *Neural Networks*, **10**, 941-963.
- [10] Ishii, S., & Sato, M. (1998). Constrained neural approaches to quadratic assignment problems. *Neural Networks*, **11**, 1073-1082.
- [11] Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell Systems Technical Journal*, **44**, 2245-2269.
- [12] Lin, S., & Kernighan, B. W. (1973). An efficient heuristic algorithm for the traveling salesman problem. *Operations Research*, **21**, 498-516.
- [13] Luenberger, D. G. (1989). *Linear and Nonlinear Programming* (2nd edition). Reading: Addison-Wesley.
- [14] Martin, S. W., Otto, O., & Feltman, E. W. (1992). Large-step markov chains for the traveling salesman problem. *Operations Research Letters*, **11**, 219-224.
- [15] Peterson, C., & Söderberg, B. (1989). A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, **1**, 3-22.
- [16] Taillard, E. D. (1991). Robust taboo search for the quadratic assignment problem. *Parallel Computing*, **17**, 443-455.
- [17] Taillard, E. D. (1995). Comparison of iterative searches for the quadratic assignment problem. *Location Science*, **3**, 87-105.

Appendix

Here, we derive the λ -interior DCN equations defined by (13a), (22b), (13c), (13d), and (22e). In order to minimize $F^*(\mathbf{X}; \mathbf{S}) = E^*(\mathbf{X}; \mathbf{S}) - TH^*(\mathbf{X}) + B^*(\mathbf{X})$ subject to constraints (12b) and (12c), we define a Lagrange function:

$$L = F^* + \sum_{a=1}^N A_a \left(\sum_{b=1}^N X_{a,b} - 1 \right) + \sum_{b=1}^N B_b \left(\sum_{a=1}^N X_{a,b} - 1 \right), \quad (\text{A.1})$$

where A_a ($a = 1, \dots, N$) and B_b ($b = 1, \dots, N$) are Lagrange multipliers. A stationary condition of the Lagrange function is given by

$$\frac{\partial L}{\partial X_{a,b}} = \frac{\partial E^*}{\partial X_{a,b}} + T \left[\log X_{a,b} + 1 + \delta_{a,b} \left(\log \left(\sum_a X_{a,a} - M \right) + \theta \right) \right] + A_a + B_b = 0, \quad (\text{A.2})$$

with constraints (12b) and (12c). Equation (A.2) is solved as

$$X_{a,b} = \exp \left(-\frac{1}{T} \frac{\partial E^*}{\partial X_{a,b}} \right) / (\alpha_a \beta_b) \quad \text{if } a \neq b \quad (\text{A.3a})$$

$$X_{a,a} \left(\sum_a X_{a,a} - M \right) = \exp \left(-\frac{1}{T} \frac{\partial E^*}{\partial X_{a,a}} - \theta \right) / (\alpha_a \beta_a), \quad (\text{A.3b})$$

where $\alpha_a \equiv \exp(A_a/T + 1)$ and $\beta_b \equiv \exp(B_b/T)$ are the converted Lagrange multipliers. Equation (A.3a) is equivalent to (13a) and (22b) for $a \neq b$. The summation of (A.3b) over a gives a quadratic equation for $\sum_a X_{a,a}$. A solution of the quadratic equation is given by

$$\sum_a X_{a,a} = \frac{M + \sqrt{M^2 + 4 \sum_a \frac{U_{a,a}}{\alpha_a \beta_a}}}{2}. \quad (\text{A.4})$$

By defining $\gamma \equiv \sum_a X_{a,a} - M$, the equation (22e) is obtained. Equation (22b) for $a = b$ is straightforward from (A.3b) and the definition of γ .