NOTE **Communicated by Eric Mjolsness**

A Lagrange Multiplier and Hopfield-Type Barrier Function Method for the Traveling Salesman Problem

Chuangyin Dang
mecdang@cityu.edu.hk
Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong

Lei Xu
lxu@cse.cuhk.edu.hk
Department of Computer Science and Engineering, Chinese University of Hong Kong, New Territories, Hong Kong

A Lagrange multiplier and Hopfield-type barrier function method is proposed for approximating a solution of the traveling salesman problem. The method is derived from applications of Lagrange multipliers and a Hopfield-type barrier function and attempts to produce a solution of high quality by generating a minimum point of a barrier problem for a sequence of descending values of the barrier parameter. For any given value of the barrier parameter, the method searches for a minimum point of the barrier problem in a feasible descent direction, which has a desired property that lower and upper bounds on variables are always satisfied automatically if the step length is a number between zero and one. At each iteration, the feasible descent direction is found by updating Lagrange multipliers with a globally convergent iterative procedure. For any given value of the barrier parameter, the method converges to a stationary point of the barrier problem without any condition on the objective function. Theoretical and numerical results show that the method seems more effective and efficient than the softassign algorithm.

1 Introduction

The traveling salesman problem (TSP) is an NP-hard combinatorial optimization problem and has many important applications. In order to solve it, a number of classic algorithms and heuristics have been proposed. We refer to Lawler, Lenstra, Rinnooy, and Shmoys (1985) for an excellent survey of techniques for solving the problem.

Since Hopfield and Tank (1985), combinatorial optimization has become a popular topic in the literature of neural computation. Many neural computational models for combinatorial optimization have been developed. They include Aiyer, Niranjan, and Fallside (1990); van den Bout and Miller (1990);

Durbin and Willshaw (1987); Gee, Aiyer, and Prager (1993); Gee and Prager (1994); Gold, Mjolsness, and Rangarajan (1994); Gold and Rangarajan (1996); Peterson and Soderberg (1989); Rangarajan, Gold, and Mjolsness (1996); Simic (1990); Urahama (1996); Wacholder, Han, and Mann (1989); Waugh and Westervelt (1993); Wolfe, Parry, and MacMillan (1994); Xu (1994); and Yuille and Kosowsky (1994). A systematic investigation of such neural computational models for combinatorial optimization can be found in van den Berg (1996) and Cichocki and Unbehaunen (1993). Most of these algorithms are of the deterministic annealing type, which is a heuristic continuation method that attempts to find the global minimum of the effective energy at high temperature and track it as the temperature decreases. There is no guarantee that the minimum at high temperature can always be tracked to the minimum at low temperature, but the experimental results are encouraging (Yuille & Kosowsky, 1994).

We propose a Lagrange multiplier and a Hopfield-type barrier function method for approximating a solution of the TSP. The method is derived from applications of Lagrange multipliers to handle equality constraints and a Hopfield-type barrier function to deal with lower and upper bounds on variables. The method is a deterministic annealing algorithm that attempts to produce a high-quality solution by generating a minimum point of a barrier problem for a sequence of descending values of the barrier parameter. For any given value of the barrier parameter, the method searches for a minimum point of the barrier problem in a feasible descent direction, which has the desired property that the lower and upper bounds on variables are always satisfied automatically if the step length is a number between zero and one. At each iteration, the feasible descent direction is found by updating Lagrange multipliers with a globally convergent iterative procedure. For any given value of the barrier parameter, the method converges to a stationary point of the barrier problem without any condition on the objective function. Theoretical and numerical results show that the method seems more effective and efficient than the softassign algorithm.

The rest of this paper is organized as follows. We introduce the Hopfield-type barrier function and derive some properties in section 2. We present the method in section 3. We report some numerical results in section 4. We conclude in section 5.

2 Hopfield-Type Barrier Function

The problem we consider is as follows. Given n cities, find a tour such that each city is visited exactly once and that the total distance traveled is minimized. Let

$$v_{ik} = \begin{cases} 1 & \text{if City } i \text{ is the } k\text{th city to be visited in a tour,} \\ 0 & \text{otherwise,} \end{cases}$$

where $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$, and $v = (v_{11}, v_{12}, \ldots, v_{1n}, \ldots, v_{n1}, v_{n2}, \ldots, v_{nn})^\top$. In Hopfield and Tank (1985), the problem was formulated as

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n d_{ij} v_{ik} v_{j,k+1} \\ \text{subject to} \quad & \textstyle\sum_{j=1}^n v_{ij} = 1, \quad i = 1, 2, \ldots, n, \\ & \textstyle\sum_{i=1}^n v_{ij} = 1, \quad j = 1, 2, \ldots, n, \\ & v_{ij} \in \{0, 1\}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n, \end{aligned} \tag{2.1}$$

where d_{ij} denotes the distance from city i to city j and $v_{j,k+1} = v_{j1}$ for $k = n$. Clearly, for any given $\rho \geq 0$, equation 2.1 is equivalent to

$$\begin{aligned} \min \quad & e_0(v) = \textstyle\sum_{i=1}^n \sum_{j=1}^n \left(\sum_{k=1}^n d_{ij} v_{ik} v_{j,k+1} - \frac{1}{2}\rho v_{ij}^2 \right) \\ \text{subject to} \quad & \textstyle\sum_{j=1}^n v_{ij} = 1, \quad i = 1, 2, \ldots, n, \\ & \textstyle\sum_{i=1}^n v_{ij} = 1, \quad j = 1, 2, \ldots, n, \\ & v_{ij} \in \{0, 1\}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{2.2}$$

The continuous relaxation of equation 2.2 yields

$$\begin{aligned} \min \quad & e_0(v) = \textstyle\sum_{i=1}^n \sum_{j=1}^n \left(\sum_{k=1}^n d_{ij} v_{ik} v_{j,k+1} - \frac{1}{2}\rho v_{ij}^2 \right) \\ \text{subject to} \quad & \textstyle\sum_{j=1}^n v_{ij} = 1, \quad i = 1, 2, \ldots, n, \\ & \textstyle\sum_{i=1}^n v_{ij} = 1, \quad j = 1, 2, \ldots, n, \\ & 0 \leq v_{ij} \leq 1, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{2.3}$$

When ρ is sufficiently large, one can see that an optimal solution of equation 2.3 is an integer solution. Thus, when ρ is sufficiently large, equation 2.3 is equivalent to equation 2.1. The term $-\frac{1}{2}\rho \sum_{i=1}^n \sum_{j=1}^n v_{ij}^2$ was introduced in Rangarajan et al. (1996) to obtain a strictly concave function $e_0(v)$ on the null space of the constraint matrix for convergence of their softassign algorithm to a stationary point of a barrier problem. We note that the size of ρ affects the quality of the solution produced by a deterministic annealing algorithm, and it should be as small as possible. However, when ρ is a small, positive number but still satisfies that equation 2.3 is equivalent to equation 2.1, the softassign algorithm may not converge to a stationary point of the barrier problem since $e_0(v)$ may not be strictly concave on the null space of the constraint matrix. Numerical tests demonstrate that it indeed occurs to the softassign algorithm.

Following Xu (1995), we introduce a Hopfield-type barrier term,

$$d(v_{ij}) = v_{ij} \ln v_{ij} + (1 - v_{ij}) \ln(1 - v_{ij}), \tag{2.4}$$

to incorporate $0 \leq x_{ij} \leq 1$ into the objective function of equation 2.3 and obtain

$$\begin{aligned} \min \quad & e(v;\beta) = e_0(v) + \beta \textstyle\sum_{i=1}^n \sum_{j=1}^n d(v_{ij}) \\ \text{subject to} \quad & \textstyle\sum_{j=1}^n v_{ij} = 1, \quad i = 1, 2, \ldots, n, \\ & \textstyle\sum_{i=1}^n v_{ij} = 1, \quad j = 1, 2, \ldots, n, \end{aligned} \tag{2.5}$$

where β is a positive barrier parameter. The barrier term, equation 2.4, appeared first in an energy function given by Hopfield (1984) and has been extensively used in the literature. Instead of solving equation 2.3 directly, we consider a scheme that obtains a solution of it from the solution of equation 2.5 at the limit of $\beta \downarrow 0$.

Let $b(v) = \sum_{i=1}^n \sum_{j=1}^n d(v_{ij})$. Then $e(v;\beta) = e_0(v) + \beta b(v)$. Let

$$P = \left\{ v \;\middle|\; \begin{array}{l} \sum_{j=1}^n v_{ij} = 1, \quad i = 1, 2, \ldots, n, \\ \sum_{i=1}^n v_{ij} = 1, \quad j = 1, 2, \ldots, n, \\ 0 \leq v_{ij} \leq 1, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n \end{array} \right\}$$

and

$$B = \{v \mid 0 \leq v_{ij} \leq 1, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n\}.$$

Then P is the feasible region of equation 2.3. Let us define $d(0) = d(1) = 0$. Since $\lim_{v_{ij} \to 0^+} d(v_{ij}) = \lim_{v_{ij} \to 1^-} d(v_{ij}) = 0$; hence, $b(v)$ is continuous on B. From $b(v)$, we obtain

$$\frac{\partial b(v)}{\partial v_{ij}} = \ln v_{ij} - \ln(1 - v_{ij}) = \ln \frac{v_{ij}}{1 - v_{ij}}.$$

Then

$$\lim_{v_{ij} \to 0^+} \frac{\partial b(v)}{\partial v_{ij}} = -\infty \quad \text{and} \quad \lim_{v_{ij} \to 1^-} \frac{\partial b(v)}{\partial v_{ij}} = \infty.$$

Observe

$$\frac{\partial e_0(v)}{\partial v_{ij}} = \sum_{k=1}^n (d_{ki} v_{k,j-1} + d_{ik} v_{k,j+1}) - \rho v_{ij},$$

where $v_{k,j-1} = v_{kn}$ for $j = 1$, and $v_{k,j+1} = v_{k1}$ for $j = n$. Thus, $\frac{\partial e_0(v)}{\partial v_{ij}}$ is bounded on B. From

$$\frac{\partial e(v;\beta)}{\partial v_{ij}} = \frac{\partial e_0(v)}{\partial v_{ij}} + \beta \frac{\partial b(v)}{\partial v_{ij}},$$

we obtain

$$\lim_{v_{ij} \to 0^+} \frac{\partial e(v;\beta)}{\partial v_{ij}} = -\infty \quad \text{and} \quad \lim_{v_{ij} \to 1^-} \frac{\partial e(v;\beta)}{\partial v_{ij}} = \infty.$$

Lemma 1. *For any given $\beta > 0$, if v^* is a local minimum point of equation 2.5, v^* is an interior point of P, that is, $0 < v^*_{ij} < 1$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$.*[1]

Let

$$L(v, \lambda^r, \lambda^c) = e(v; \beta) + \sum_{i=1}^{n} \lambda_i^r \left(\sum_{j=1}^{n} v_{ij} - 1 \right) + \sum_{j=1}^{n} \lambda_j^c \left(\sum_{i=1}^{n} v_{ij} - 1 \right).$$

Lemma 1 indicates that if v^* is a local minimum point of equation 2.5, then there exist λ^{r*} and λ^{c*} satisfying

$$\begin{aligned} &\nabla_v L(v^*, \lambda^{r*}, \lambda^{c*}) = 0, \\ &\textstyle\sum_{j=1}^{n} v^*_{ij} = 1, \quad i = 1, 2, \ldots, n, \\ &\textstyle\sum_{i=1}^{n} v^*_{ij} = 1, \quad j = 1, 2, \ldots, n, \end{aligned}$$

where

$$\nabla_v L(v, \lambda^r, \lambda^c) = \left(\frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{11}}, \frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{12}}, \ldots, \frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{1n}}, \ldots, \frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{n1}}, \frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{n2}}, \ldots, \frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{nn}} \right)^\top$$

with

$$\frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{ij}} = \frac{\partial e_0(v)}{\partial v_{ij}} + \lambda_i^r + \lambda_j^c + \beta \ln \frac{v_{ij}}{1 - v_{ij}},$$

$i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$.

Let β_k, $k = 1, 2, \ldots$, be a sequence of positive numbers satisfying

$$\beta_1 > \beta_2 > \cdots$$

and $\lim_{k\to\infty} \beta_k = 0$. For $k = 1, 2, \ldots$, let $v(\beta_k)$ denote a global minimum point of equation 2.5 with $\beta = \beta_k$.

Theorem 1. *For $k = 1, 2, \ldots$,*

$$e_0(v(\beta_k)) \geq e_0(v(\beta_{k+1})),$$

and any limit point of $v(\beta_k)$, $k = 1, 2, \ldots$, is a global minimum point of equation 2.3.

---

[1] All the proofs of lemmas and theorems in this article can be found on-line at www.cityu.edu.hk/meem/mecdang.

This theorem indicates that a global minimum point of equation 2.3 can be obtained if we are able to generate a global minimum point of equation 2.5 for a sequence of descending values of the barrier parameter with zero limit.

Theorem 2. *For $k = 1, 2, \ldots,$ let v^k be a local minimum point of equation 2.5 with $\beta = \beta_k$. For any limit point v^* of v^k, $k = 1, 2, \ldots,$ if there are no $\lambda^r = (\lambda_1^r, \lambda_2^r, \ldots, \lambda_n^r)^\top$ and $\lambda^c = (\lambda_1^c, \lambda_2^c, \ldots, \lambda_n^c)^\top$ satisfying*

$$\frac{\partial e_0(v^*)}{\partial v_{ij}} + \lambda_i^r + \lambda_j^c = 0,$$

$i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$, then v^* is a local minimum point of equation 2.3.

This theorem indicates that at least a local minimum point of equation 2.3 can be obtained if we are able to generate a local minimum point of equation 2.5 for a sequence of descending values of the barrier parameter with zero limit.

3 The Method

Stimulated from the results in the previous section, we propose in this section a method for approximating a solution of equation 2.3. The idea of the method is as follows: Choose β_0 to be a sufficiently large, positive number satisfying that $e(v; \beta_0)$ is strictly convex. Let β_q, $q = 0, 1, \ldots,$ be a sequence of positive numbers satisfying

$$\beta_0 > \beta_1 > \cdots$$

and $\lim_{q\to\infty} \beta_q = 0$. Choose $v^{*,0}$ to be the unique minimum point of equation 2.5 with $\beta = \beta_0$. For $q = 1, 2, \ldots,$ starting at $v^{*,q-1}$, we search for a minimum point $v^{*,q}$ of equation 2.5 with $\beta = \beta_q$.

Given any $\beta > 0$, consider the first-order necessary optimality condition for equation 2.5:

$$\begin{aligned} &\nabla_v L(v, \lambda^r, \lambda^c) = 0, \\ &\textstyle\sum_{j=1}^n v_{ij} = 1, \quad i = 1, 2, \ldots, n, \\ &\textstyle\sum_{i=1}^n v_{ij} = 1, \quad j = 1, 2, \ldots, n. \end{aligned}$$

From

$$\frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{ij}} = \frac{\partial e_0(v)}{\partial v_{ij}} + \lambda_i^r + \lambda_j^c + \beta \ln \frac{v_{ij}}{1 - v_{ij}} = 0,$$

we obtain

$$v_{ij} = \frac{1}{1 + \exp((\frac{\partial e_0(v)}{\partial v_{ij}} + \lambda_i^r + \lambda_j^c)/\beta)}.$$

Let $r_i = \exp\left(\frac{\lambda_i^r}{\beta}\right)$ and $c_j = \exp\left(\frac{\lambda_j^c}{\beta}\right)$. Then,

$$v_{ij} = \frac{1}{1 + r_i c_j \exp(\frac{\partial e_0(v)}{\partial v_{ij}}/\beta)}.$$

For convenience of the following discussions, let $\alpha_{ij}(v) = \exp\left(\frac{\partial e_0(v)}{\partial v_{ij}}/\beta\right)$. Then,

$$v_{ij} = \frac{1}{1 + r_i c_j \alpha_{ij}(v)}. \tag{3.1}$$

Substituting equation 3.1 into $\sum_{j=1}^{n} v_{ij} = 1$, $i = 1, 2, \ldots, n$, and $\sum_{i=1}^{n} v_{ij} = 1$, $j = 1, 2, \ldots, n$, we obtain

$$\begin{aligned} \textstyle\sum_{j=1}^{n} \frac{1}{1+r_i c_j \alpha_{ij}(v)} &= 1, \quad i = 1, 2, \ldots, n, \\ \textstyle\sum_{i=1}^{n} \frac{1}{1+r_i c_j \alpha_{ij}(v)} &= 1, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{3.2}$$

Based on the above notations, a conceptual algorithm was proposed in Xu (1995) for approximating a solution of equation 2.3, which is as follows:

- Fix r and c. Use equation 3.1 to obtain v.
- Fix v. Solve equation 3.2 for r and c.

Let

$$h_{ij}(v, r, c) = \frac{1}{1 + r_i c_j \alpha_{ij}(v)}$$

and

$$\begin{aligned} h(v, r, c) = {} & (h_{11}(v, r, c), h_{12}(v, r, c), \ldots, h_{1n}(v, r, c), \\ & \ldots, h_{n1}(v, r, c), h_{n2}(v, r, c), \ldots, h_{nn}(v, r, c))^\top. \end{aligned}$$

If v is an interior point of B, the following lemma shows that $h(v, r, c) - v$ is a descent direction of $L(v, \lambda^r, \lambda^v)$.

Lemma 2. *Assume* $0 < v_{ij} < 1$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$.

1. $\frac{\partial L(v,\lambda^r,\lambda^v)}{\partial v_{ij}} > 0$ *if* $h_{ij}(v, r, c) - v_{ij} < 0$.
2. $\frac{\partial L(v,\lambda^r,\lambda^v)}{\partial v_{ij}} < 0$ *if* $h_{ij}(v, r, c) - v_{ij} > 0$.
3. $\frac{\partial L(v,\lambda^r,\lambda^v)}{\partial v_{ij}} = 0$ *if* $h_{ij}(v, r, c) - v_{ij} = 0$.
4. $(h(v, r, c) - v)^\top \nabla_v L(v, \lambda^r, \lambda^c) < 0$ *if* $h(v, r, c) - v \neq 0$.
5. $(h(v, r, c) - v)^\top \nabla_v e(v; \beta) < 0$ *if* $h(v, r, c) - v \neq 0$ *and* $\sum_{k=1}^{n}(h_{ik}(v, r, c) - v_{ik}) = \sum_{k=1}^{n}(h_{kj}(v, r, c) - v_{kj}) = 0$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$.

Proof. We only need to show that $\frac{\partial L(v,\lambda^r,\lambda^v)}{\partial v_{ij}} > 0$ if $h_{ij}(v, r, c) - v_{ij} < 0$. The rest can be obtained similarly or in a straightforward manner. From

$$h_{ij}(v, r, c) - v_{ij} = \frac{1}{1 + r_i c_j \alpha_{ij}(v)} - v_{ij} < 0,$$

we obtain

$$1 < r_i c_j \alpha_{ij}(v) \frac{v_{ij}}{1 - v_{ij}}. \tag{3.3}$$

Applying the natural logarithm, ln, to both sides of equation 3.3, we get

$$\begin{aligned} 0 &< \ln\left(r_i c_j \alpha_{ij}(v) \frac{v_{ij}}{1 - v_{ij}}\right) \\ &= \ln \alpha_{ij}(v) + \ln r_i + \ln c_j + \ln \frac{v_{ij}}{1 - v_{ij}} \\ &= \frac{1}{\beta}\frac{\partial e_0(v)}{\partial v_{ij}} + \frac{1}{\beta}\lambda_i^r + \frac{1}{\beta}\lambda_j^c + \ln \frac{v_{ij}}{1 - v_{ij}} = \frac{1}{\beta}\frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{ij}}. \end{aligned}$$

Thus,

$$\frac{\partial L(v, \lambda^r, \lambda^c)}{\partial v_{ij}} > 0.$$

The lemma follows.

Since $0 < h_{ij}(v, r, c) < 1$, we note that the descent direction $h(v, r, c) - v$ has a desired property that any point generated along $h(v, r, c) - v$ satisfies automatically the lower and upper bounds if $v \in B$ and the step length is a number between zero and one.

For any given point v, we use $(r(v), c(v))$ to denote a positive solution of equation 3.2. Let v be an interior point of P. In order for $h(v, r, c) - v$ to become a feasible descent direction of equation 2.5, we need to compute a positive solution $(r(v), c(v))$ of equation 3.2. Let

$$\begin{aligned} f(r, c) = \frac{1}{2}\Biggl(&\sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{1}{1 + r_i c_j \alpha_{ij}(v)} - 1\right)^2 \\ &+ \sum_{j=1}^{n}\left(\sum_{i=1}^{n}\frac{1}{1 + r_i c_j \alpha_{ij}(v)} - 1\right)^2\Biggr). \end{aligned}$$

Observe that the value of $f(r, c)$ equals zero only at a solution of equation 3.2. For $i = 1, 2, \ldots, n$, let

$$x_i(r, c) = r_i \left(\sum_{j=1}^{n} \frac{1}{1 + r_i c_j \alpha_{ij}(v)} - 1 \right),$$

and for $j = 1, 2, \ldots, n$, let

$$y_j(r, c) = c_j \left(\sum_{i=1}^{n} \frac{1}{1 + r_i c_j \alpha_{ij}(v)} - 1 \right).$$

Let

$$\begin{aligned} x(r, c) &= (x_1(r, c), x_2(r, c), \ldots, x_n(r, c))^\top \\ y(r, c) &= (y_1(r, c), y_2(r, c), \ldots, y_n(r, c))^\top. \end{aligned}$$

One can easily prove that

$$\begin{pmatrix} x(r, c) \\ y(r, c) \end{pmatrix}$$

is a descent direction of $f(r, c)$. For any given v, based on this descent direction, the following iterative procedure is proposed for computing a positive solution $(r(v), c(v))$ of equation 3.2.

Take (r^0, c^0) to be an arbitrary positive vector, and for $k = 0, 1, \ldots,$ let

$$\begin{aligned} r^{k+1} &= r^k + \mu_k x(r^k, c^k), \\ c^{k+1} &= c^k + \mu_k y(r^k, c^k), \end{aligned} \tag{3.4}$$

where μ_k is a number in $[0, 1]$ satisfying

$$f(r^{k+1}, c^{k+1}) = \min_{\mu \in [0,1]} f(r^k + \mu_k x(r^k, c^k), c^k + \mu_k y(r^k, c^k)).$$

Observe that $(r^k, c^k) > 0$, $k = 0, 1, \ldots$. There are many ways to determine μ_k (Minoux, 1986). For example, one can simply choose μ_k to be any number in $(0, 1]$ satisfying $\sum_{l=0}^{k} \mu_l \to \infty$ and $\mu_k \to 0$ as $k \to \infty$. We have found in our numerical tests that when μ_k is any fixed number in $(0, 1]$, the iterative procedure, equation 3.4, converges to a positive solution of equation 3.2.

Theorem 3. *For any given v, every limit point of (r^k, c^k), $k = 0, 1, \ldots,$ generated by the iterative procedure, equation 3.4, is a positive solution of equation 3.2.*

Based on the feasible descent direction, $h(v, r(v), c(v))-v$, and the iterative procedure, equation 3.4, we have developed a method for approximating a solution of equation 2.3, which can be stated as follows:

Step 0: Let $\epsilon > 0$ be a given tolerance. Let β_0 be a sufficiently large, positive number satisfying that $e(v; \beta_0)$ is convex. Choose an arbitrary interior point $\bar{v} \in B$, and two arbitrary positive vectors, r^0 and c^0. Take an arbitrary positive number $\eta \in (0, 1)$ (in general, η should be close to one). Given $v = \bar{v}$, use equation 3.4 to obtain a positive solution $(r(\bar{v}), c(\bar{v}))$ of equation 3.2. Let $r^0 = r(\bar{v})$ and $c^0 = c(\bar{v})$. Let

$$v^0 = (v^0_{11}, v^0_{12}, \ldots, v^0_{1n}, \ldots, v^0_{n1}, v^0_{n2}, \ldots, v^0_{nn})^\top$$

with

$$v^0_{ij} = \frac{1}{1 + r_i(\bar{v})c_j(\bar{v})\alpha_{ij}(\bar{v})},$$

where $i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, n$. Let $q = 0$ and $k = 0$, and go to step 1.

Step 1: Given $v = v^k$, use equation 3.4 to obtain a positive solution $(r(v^k), c(v^k))$ of equation 3.2. Let $r^0 = r(v^k)$ and $c^0 = c(v^k)$. Go to step 2.

Step 2: Let

$$\begin{aligned} &h(v^k, r(v^k), c(v^k)) \\ &\quad = (h_{11}(v^k, r(v^k), c(v^k)), h_{12}(v^k, r(v^k), c(v^k)), \ldots, \\ &\quad h_{1n}(v^k, r(v^k), c(v^k)), \ldots, h_{n1}(v^k, r(v^k), c(v^k)), h_{n2}(v^k, r(v^k), c(v^k)), \\ &\quad \ldots, h_{nn}(v^k, r(v^k), c(v^k)))^\top \end{aligned}$$

with

$$h_{ij}(v^k, r(v^k), c(v^k)) = \frac{1}{1 + r_i(v^k)c_j(v^k)\alpha_{ij}(v^k)},$$

where $i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, n$. If $\|h(v^k, r(v^k), c(v^k)) - v^k\| < \epsilon$, do as follows:

- If β_q is sufficiently small, the method terminates.
- Otherwise, let $v^{*,q} = v^k$, $v^0 = v^k$, $\beta_{q+1} = \eta\beta_q$, $q = q + 1$, and $k = 0$, and go to step 1.

If $\|h(v^k, r(v^k), c(v^k)) - v^k\| \geq \epsilon$, do as follows: Compute

$$v^{k+1} = v^k + \theta_k(h(v^k, r(v^k), c(v^k)) - v^k), \tag{3.5}$$

where θ_k is a number in $[0, 1]$ satisfying

$$e(v^{k+1}; \beta_q) = \min_{\theta\in[0,1]} e(v^k + \theta(h(v^k, r(v^k), c(v^k)) - v^k); \beta_q).$$

Let $k = k + 1$, and go to step 1.

Note that an exact positive solution $(r(v^k), c(v^k))$ of equation 3.2 for $v = v^k$ and an exact solution of $\min_{\theta \in [0,1]} e(v^k + \theta(h(v^k, r(v^k), c(v^k)) - v^k); \beta_q)$ are not required in the implementation of the method, and their approximate solutions will do. There are many ways to determine θ_k (Minoux, 1986). For example, one can simply choose θ_k to be any number in (0, 1] satisfying $\sum_{l=0}^{k} \theta_l \to \infty$ and $\theta_k \to 0$ as $k \to \infty$. The method is insensitive to the starting point since $e(v, \beta_0)$ is convex over B.

Theorem 4. *For $\beta = \beta_q$, every limit point of v^k, $k = 0, 1, \ldots,$ generated by equation 2.5 is a stationary point of equation 3.5.*

Although it is difficult to prove that for any given $\beta > 0$, a limit point of v^k, $k = 0, 1, \ldots,$ generated by equation 3.5 is at least a local minimum point of equation 2.5, in general, it is indeed at least a local minimum point of equation 2.5. Theorem 2 implies that every limit point of $v^{*,q}$, $q = 0, 1, \ldots,$ is at least a local minimum point of equation 2.3 if $v^{*,q}$ is a minimum point of equation 2.5 with $\beta = \beta_q$.

For $\beta = \beta_q$, our method can be proved to converge to a stationary point of equation 2.5 for any given ρ; however, the softassign algorithm can be proved to converge to a stationary point of equation 2.5 only if ρ is sufficiently large so that $e_0(v)$ is strictly concave on the null space of the constraint matrix (Rangarajan, Yuille, & Mjolsness, 1999). Numerical tests also show that the softassign algorithm does not converge to a stationary point of equation 2.5 if the condition is not satisfied. Thus, for the softassign algorithm to converge, one has to determine the size of ρ through estimating the maximum eigenvalue of the matrix of the objective function of equation 2.1, which requires some extra computational work. As we pointed out, the size of ρ affects the quality of a solution generated by a deterministic annealing algorithm, and it should be as small as possible. Since our method converges for any ρ, one can start with a smaller positive ρ and then increase ρ if the solution generated by the method is not a near integer solution. In this respect, our method is better than the softassign algorithm. Numerical results support this argument.

4 Numerical Results

The method has been used to approximate solutions of a number of TSP instances. The method succeeds in finding an optimal or near-optimal tour for each of the TSP instances. In our implementation of the method,

1. $\epsilon = 0.01$ and $\beta_0 = 200$.

2. We take $r^0 = (r_1^0, r_2^0, \ldots, r_n^0)^\top$ and $c^0 = (c_1^0, c_2^0, \ldots, c_n^0)^\top$ to be two random vectors satisfying $0 < r_i^0 < 1$ and $0 < c_i^0 < 1$, $i = 1, 2, \ldots, n$.

3. $\mu_k = 0.95$, and for any given v, the iterative procedure, equation 3.4, terminates as soon as $\sqrt{f(r^k, c^k)} < 0.001$.
4. We replace $e(x; \beta)$ with $L(v, \lambda^r, \lambda^c)$ in the method since $(r(v^k), c(v^k))$ is an approximate solution of equation 3.2.
5. θ_k is determined with the following Armijo-type line search: $\theta_k = \xi^{m_k}$, with m_k being the smallest nonnegative integer satisfying

$$\begin{aligned} &L(v^k + \xi^{m_k}(h(v^k, r(v^k), c(v^k)) - v^k), \lambda^{r,k}, \lambda^{c,k}) \\ &\quad \le L(v^k, \lambda^{r,k}, \lambda^{c,k}) \\ &\qquad + \xi^{m_k}\gamma\,(h(v^k, r(v^k), c(v^k)) - v^k)^\top \nabla_v L(v^k, \lambda^{r,k}, \lambda^{c,k}), \end{aligned}$$

where ξ and γ are any two numbers in $(0, 1)$ (we set $\xi = 0.6$ and $\gamma = 0.8$),

$$\lambda^{r,k} = \beta_q(\ln r_1(v^k), \ln r_2(v^k), \ldots, \ln r_n(v^k))^\top,$$

and

$$\lambda^{c,k} = \beta_q(\ln c_1(v^k), \ln c_2(v^k), \ldots, \ln c_n(v^k))^\top.$$

The method terminates as soon as $\beta_q < 1$. To produce a solution of higher quality, the size of ρ should be as small as possible. However, a small ρ may lead to a fractional solution $v^{*,q}$. To make sure that an integer solution will be generated, we continue the following procedure:

Step 0: Let $\beta = 1$, $v^0 = v^{*,q}$, and $k = 0$. Go to step 1.

Step 1: Let $v^* = (v^*_{11}, v^*_{12}, \ldots, v^*_{1n}, \ldots, v^*_{n1}, v^*_{n2}, \ldots, v^*_{nn})^\top$ with

$$v^*_{ij} = \begin{cases} 1 & \text{if } v^k_{ij} \ge 0.9, \\ 0 & \text{if } v^k_{ij} < 0.9, \end{cases}$$

$i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$. If $v^* \in P$, the procedure terminates. Otherwise, let $\rho = \rho + 2$, and go to step 2.

Step 2: Given $v = v^k$, use equation 3.4 to obtain a positive solution $(r(v^k), c(v^k))$ of equation 3.2. Let $r^0 = r(v^k)$, $c^0 = c(v^k)$,

$$\lambda^{r,k} = (\ln r_1(v^k), \ln r_2(v^k), \ldots, \ln r_n(v^k))^\top,$$

and

$$\lambda^{c,k} = (\ln c_1(v^k), \ln c_2(v^k), \ldots, \ln c_n(v^k))^\top.$$

Go to step 3.

Step 3: Let

$$\begin{aligned} &h(v^k, r(v^k), c(v^k)) \\ &\quad = (h_{11}(v^k, r(v^k), c(v^k)), h_{12}(v^k, r(v^k), c(v^k)), \ldots, \\ &\qquad h_{1n}(v^k, r(v^k), c(v^k)), \ldots, h_{n1}(v^k, r(v^k), c(v^k)), \\ &\qquad h_{n2}(v^k, r(v^k), c(v^k)), \ldots, h_{nn}(v^k, r(v^k), c(v^k)))^\top \end{aligned}$$

with

$$h_{ij}(v^k, r(v^k), c(v^k)) = \frac{1}{1 + r_i(v^k)c_j(v^k)\alpha_{ij}(v^k)},$$

where $i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, n$. If $\|h(v^k, r(v^k), c(v^k)) - v^k\| < \epsilon$, let $v^0 = v^k$ and $k = 0$, and go to step 1. Otherwise, compute

$$v^{k+1} = v^k + \theta_k(h(v^k, r(v^k), c(v^k)) - v^k),$$

where θ_k is a number in [0, 1] satisfying

$$L(v^{k+1}, \lambda^{r,k}, \lambda^{c,k}) = \min_{\theta \in [0,1]} L(v^k + \theta(h(v^k, r(v^k), c(v^k)) - v^k), \lambda^{r,k}, \lambda^{c,k}).$$

Let $k = k + 1$, and go to step 2.

The method is programmed in MATLAB. To compare the method with the softassign algorithm proposed in Gold et al. (1994) and Rangarajan et al. (1996, 1999) and the softassign algorithm modified by introducing line search, the softassign algorithm and its modified version are also programmed in MATLAB. All our numerical tests are done on a PC computer. In the presentations of numerical results, DM stands for our method, SA the softassign algorithm, MSA the modified version of the softassign algorithm, CT the computation time in seconds, OPT the length of an optimal tour, OBJ the length of a tour generated by an algorithm, OBJD the length of the tour generated by our method, OBJSA the length of the tour generated by the softassign algorithm or its modified version, and $RE = \frac{OBJ-OPT}{OPT}$. Numerical results are as follows.

Example 1. These ten TSP instances are from a well-known web site, TSPLIB. We have used the method, the softassign algorithm, and the modified softassign algorithm to approximate solutions of these TSP instances. Numerical results are presented in Figures 1, 2, 3, and 4 and Table 1, where the softassign algorithm fails to converge when $\rho = 30$.

Example 2. These TSP instances have 100 cities and are generated randomly. Every city is a point in a square with integer coordinates (x, y)

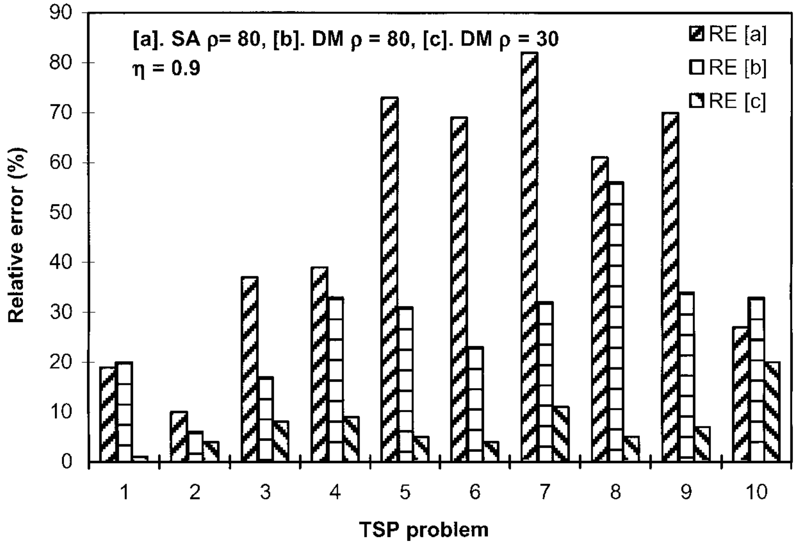

Figure 1: Relative error to optimal tour. 1. bays29, 2. att48, 3. eil51, 4. berlin52, 5. st70, 6. eil76, 7. pr76, 8. rd100, 9. eil101, 10. lin105.

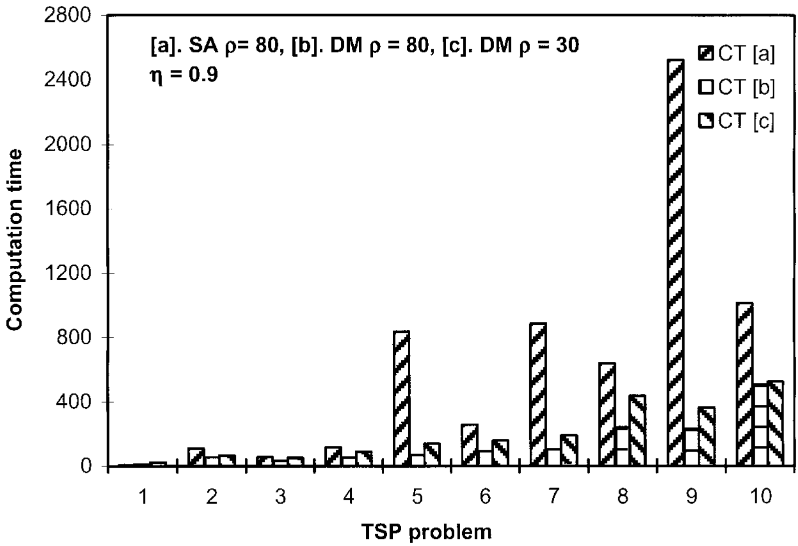

Figure 2: Computation time for different algorithms. 1. bays29, 2. att48, 3. eil51, 4. berlin52, 5. st70, 6. eil76, 7. pr76, 8. rd100, 9. eil101, 10. lin105.

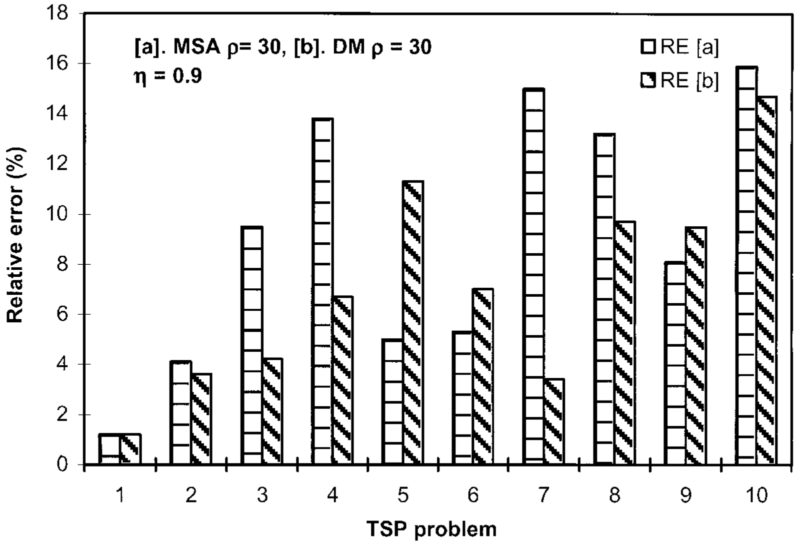

Figure 3: Relative error to optimal tour. 1. bays29, 2. att48, 3. eil51, 4. berlin52, 5. st70, 6. eil76, 7. pr76, 8. rd100, 9. eil101, 10. lin105.

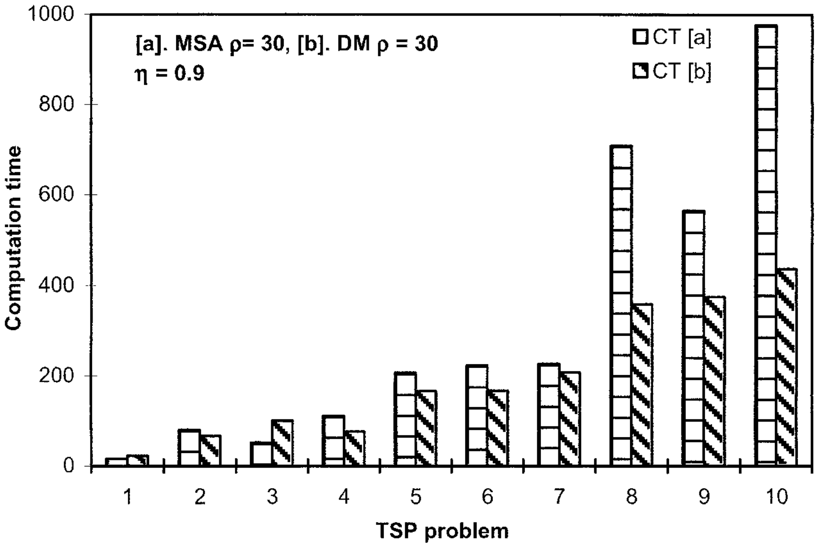

Figure 4: Computation time for different algorithms. 1. bays29, 2. att48, 3. eil51, 4. berlin52, 5. st70, 6. eil76, 7. pr76, 8. rd100, 9. eil101, 10. lin105.

Table 1: Numerical Results for TSP Instances from TSPLIB.

η		SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$	SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$	SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$	SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$	SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$
0.9		bays29, $n = 29$			att48, $n = 48$			eil51, $n = 51$			berlin52, $n = 52$			st70, $n = 70$		
	CT	7	9	22	109	53	64	55	32	50	115	51	87	835	66	138
	OBJ	2404	2430	2045	36,943	35,434	34,736	591	501	464	10,494	10017	8205	1174	887	710
	RE(%)	19	20	1	10	6	4	37	17	8	39	33	9	73	31	5
		eil76, $n = 76$			pr76, $n = 76$			rd100, $n = 100$			eil101, $n = 101$			lin105, $n = 105$		
	CT	256	89	157	885	101	187	636	240	436	2525	229	362	1011	505	522
	OBJ	923	671	566	196,520	142,331	120,417	13,223	12,788	8626	1091	862	687	18,272	19,080	17,278
	RE(%)	69	23	4	82	32	11	61	56	5	70	34	7	27	33	20
0.95		bays29, $n = 29$			att48, $n = 48$			eil51, $n = 51$			berlin52, $n = 52$			st70, $n = 70$		
	CT	22	13	30	104	67	79	59	53	76	121	55	89	1052	109	189
	OBJ	3093	2472	2045	36,293	36,106	34,736	572	588	449	11,793	8857	7837	1198	867	690
	RE(%)	53	22	1	8	8	4	33	37	4	56	17	4	76	28	2
		eil76, $n = 76$			pr76, $n = 76$			rd100, $n = 100$			eil101, $n = 101$			lin105, $n = 105$		
	CT	355	131	237	476	133	223	2382	367	412	1193	416	473	1020	641	682
	OBJ	852	638	567	189,949	169,769	118,459	13,506	10,483	9172	988	926	680	17,310	19,532	16,788
	RE(%)	56	17	4	76	57	10	65	28	12	54	44	6	20	36	17

(continued)

Table 1: (*continued*).

		MSA $\rho = 30$	DM $\rho = 30$	MSA $\rho = 30$	DM $\rho = 30$	MSA $\rho = 30$	DM $\rho = 30$	MSA $\rho = 30$	DM $\rho = 30$	MSA $\rho = 30$	DM $\rho = 30$
0.9		bays29, $n = 29$		att48, $n = 48$		eil51, $n = 51$		berlin52, $n = 52$		st70, $n = 70$	
	CT	16	22	80	66	52	100	111	76	206	166
	OBJ	2045	2045	34,903	34,736	471	448	8588	8047	713	756
	RE(%)	1.2	1.2	4.1	3.6	9.5	4.2	13.8	6.7	5.0	11.3
		eil76, $n = 76$		pr76, $n = 76$		rd100, $n = 100$		eil101, $n = 101$		lin105, $n = 105$	
	CT	223	167	226	207	708	358	564	374	978	436
	OBJ	574	583	124,408	111,862	9287	8999	694	703	16,676	16,498
	RE(%)	5.3	7.0	15.0	3.4	13.2	9.7	8.1	9.5	15.9	14.7
0.95		bays29, $n = 29$		att48, $n = 48$		eil51, $n = 51$		berlin52, $n = 52$		st70, $n = 70$	
	CT	21	22	93	75	68	85	110	99	253	188
	OBJ	2124	2148	35,068	34,735	461	468	8214	8373	780	704
	RE(%)	5.2	6.3	4.6	3.6	7.2	8.8	8.9	11.0	14.9	3.7
		eil76, $n = 76$		pr76, $n = 76$		rd100, $n = 100$		eil101, $n = 101$		lin105, $n = 105$	
	CT	252	217	291	212	910	449	750	522	877	673
	OBJ	581	567	119,332	118,453	8930	8761	694	678	16,694	18,581
	RE(%)	6.6	4.0	10.3	9.5	8.8	6.8	8.1	5.6	16.1	29.2

satisfying $0 \leq x \leq 100$ and $0 \leq y \leq 100$. We have used the method, the softassign algorithm, and the modified softassign algorithm to approximate solutions of a number of TSP instances. Numerical results are presented in Table 2, where the softassign algorithm fails to converge when $\rho = 30$.

From these numerical results, one can see that our method seems more effective and efficient than the softassign algorithm. Comparing our method with the softassign algorithm modified by introducing line search, one can find that our method is significantly superior to the modified softassign algorithm in computational time, although the quality of solutions generated by our method on average is only slightly better than those generated by the modified softassign algorithm. The reason that our method is faster than the softassign algorithm and its modified version lies in the procedures for updating Lagrange multipliers. Our procedure for updating Lagrange multipliers is much more efficient than Sinkhorn's approach adopted in the softassign algorithm for updating Lagrange multipliers. Although our method has advantages over the softassign algorithm and its modified version, it still may not compete with the elastic net and nonneural algorithms for TSP. The idea presented here for constructing a procedure to update Lagrange multipliers can also be applied to solving more complicated problems.

5 Conclusion

We have developed a Lagrange multiplier and Hopfield-type barrier function method for approximating a solution of the TSP. Some theoretical results have been derived. For any given barrier parameter, we have proved that the method converges to a stationary point of equation 2.5 without any condition on the objective function, which is stronger than the convergence result for the softassign algorithm. The numerical results show that the method seems more effective and efficient than the softassign algorithm. The method would be improved with a faster iterative procedure to update Lagrange multipliers for obtaining a feasible descent direction.

Acknowledgments

We thank the anonymous referees for their constructive comments and remarks, which have significantly improved the quality of this article. The preliminary version of this article was completed when C.D. was on leave at the Chinese University of Hong Kong from 1997 to 1998. The work was supported by SRG 7001061 of CityU, CUHK Direct Grant 220500680, Ho Sin-Hang Education Endowment Fund HSH 95/02, and Research Fellow and Research Associate Scheme of the CUHK Research Committee.

Table 2: Numerical Results for TSP Instances Generated Randomly.

η		SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$	SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$	SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$	SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$	SA $\rho = 80$	DM $\rho = 80$	DM $\rho = 30$
0.9			1			2			3			4			5	
	CT	1019	209	343	895	254	409	3053	336	313	1371	249	418	798	250	311
	OBJ	1327	1172	814	1240	1084	896	1430	987	867	1396	1040	808	1295	1129	869
	$\frac{OBJD}{OBJSA}$		0.88	0.61		0.87	0.72		0.69	0.61		0.75	0.58		0.87	0.67
			6			7			8			9			10	
	CT	6553	365	397	633	275	357	641	297	332	1399	190	374	843	197	464
	OBJ	1412	976	821	1345	1256	870	1304	1200	889	1196	1059	810	1366	1115	856
	$\frac{OBJD}{OBJSA}$		0.69	0.58		0.93	0.65		0.92	0.68		0.89	0.68		0.82	0.63
0.95			1			2			3			4			5	
	CT	905	486	421	834	285	430	4960	320	437	1928	321	488	630	400	400
	OBJ	1359	1262	814	1262	1088	896	1586	989	867	1347	995	808	1254	1053	869
	$\frac{OBJD}{OBJSA}$		0.93	0.60		0.86	0.71		0.62	0.55		0.74	0.60		0.84	0.69
			6			7			8			9			10	
	CT	3496	282	455	565	279	437	628	281	413	1473	274	438	974	266	570
	OBJ	1361	927	817	1397	1157	870	1299	1252	889	1237	1015	810	1371	1057	856
	$\frac{OBJD}{OBJSA}$		0.68	0.60		0.83	0.62		0.96	0.68		0.82	0.66		0.77	0.62

(*continued*)

Table 2: (*continued*).

		MSA $\rho = 30$	DM $\rho = 30$	$\frac{OBJD}{OBJSA}$	MSA $\rho = 30$	DM $\rho = 30$	$\frac{OBJD}{OBJSA}$	MSA $\rho = 30$	DM $\rho = 30$	$\frac{OBJD}{OBJSA}$	MSA $\rho = 30$	DM $\rho = 30$	$\frac{OBJD}{OBJSA}$	MSA $\rho = 30$	DM $\rho = 30$	$\frac{OBJD}{OBJSA}$
0.9			1			2			3			4			5	
	CT	527	364		617	409		381	500		513	341		846	415	
	OBJ	887	814	0.92	950	912	0.96	942	832	0.88	931	874	0.94	795	789	0.99
			6			7			8			9			10	
	CT	417	405		537	297		471	370		603	363		719	336	
	OBJ	902	885	0.98	904	879	0.97	827	901	1.09	815	827	1.02	846	889	1.05
0.95			1			2			3			4			5	
	CT	603	420		636	418		639	511		791	476		742	461	
	OBJ	890	874	0.98	908	846	0.93	869	861	0.99	894	898	1.00	883	869	0.98
			6			7			8			9			10	
	CT	642	495		700	471		666	425		654	584		625	417	
	OBJ	865	806	0.93	824	782	0.95	872	850	0.97	849	852	1.00	1012	822	0.81

References

Aiyer, S., Niranjan, M., & Fallside, F. (1990). A theoretical investigation into the performance of the Hopfield model. *IEEE Transactions on Neural Networks*, *1*, 204–215.

Cichocki, A., & Unbehaunen, R. (1993). *Neural networks for optimization and signal processing*. New York: Wiley.

Durbin, R., & Willshaw, D. (1987). An analogue approach to the traveling salesman problem using an elastic network method. *Nature*, *326*, 689–691.

Gee, A., Aiyer, S., & Prager, R. (1993). An analytical framework for optimizing neural networks. *Neural Networks*, *6*, 79–97.

Gee, A., & Prager, R. (1994). Polyhedral combinatorics and neural networks. *Neural Computation*, *6*, 161–180.

Gold, S., Mjolsness, E., & Rangarajan, A. (1994). Clustering with a domain-specific distance measure. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems*, *6* (pp. 96–103). San Mateo, CA: Morgan Kaufmann.

Gold, S., & Rangarajan, A. (1996). Softassign versus softmax: Benchmarking in combinatorial optimization. In D. Touretzky, M. Mozer, & M. Hasselmo (Eds.), *Advances in neural information processing systems*, *8* (pp. 626–632). Cambridge, MA: MIT Press.

Hopfield, J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the USA*, *81*, 3088–3092.

Hopfield, J., & Tank, D. (1985). Neural computation of decisions in optimization problems. *Biological Cybernetics*, *52*, 141–152.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1985). *The traveling salesman problem*. New York: Wiley.

Minoux, M. (1986). *Mathematical programming: Theory and algorithms*. New York: Wiley.

Peterson, C., & Soderberg, B. (1989). A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, *1*, 3–22.

Rangarajan, A., Gold, S., & Mjolsness, E. (1996). A novel optimizing network architecture with applications. *Neural Computation*, *8*, 1041–1060.

Rangarajan, A., Yuille, A., & Mjolsness, E. (1999). Convergence properties of the softassign quadratic assignment algorithm. *Neural Computation*, *11*, 1455–1474.

Simic, P. (1990). Statistical mechanics as the underlying theory of "elastic" and "neural" optimizations. *Networks*, *1*, 89–103.

Urahama, K. (1996). Gradient projection network: Analog solver for linearly constrained nonlinear programming. *Neural Computation*, *6*, 1061–1073.

van den Berg, J. (1996). *Neural relaxation dynamics*. Unpublished doctoral dissertation, Erasmus University of Rotterdam, Rotterdam, Netherlands.

van den Bout, D., & Miller, T., III (1990). Graph partitioning using annealed networks. *IEEE Transactions on Neural Networks*, *1*, 192–203.

Wacholder, E., Han, J., & Mann, R. (1989). A neural network algorithm for the multiple traveling salesman problem. *Biological Cybernetics*, *61*, 11–19.

Waugh, F., & Westervelt, R. (1993). Analog neural networks with local competition: I. Dynamics and stability. *Physical Review E*, *47*, 4524–4536.

Wolfe, W., Parry, M., & MacMillan, J. (1994). Hopfield-style neural networks and the TSP. In *Proceedings of the IEEE International Conference on Neural Networks*, *7* (pp. 4577–4582). Piscataway, NJ: IEEE Press.

Xu, L. (1994). Combinatorial optimization neural nets based on a hybrid of Lagrange and transformation approaches. In *Proceedings of the World Congress on Neural Networks* (pp. 399–404). San Diego, CA.

Xu, L. (1995). On the hybrid LT combinatorial optimization: New U-shape barrier, sigmoid activation, least leaking energy and maximum entropy. In *Proceedings of the International Conference on Neural Information Processing (ICONIP'95)* (pp. 309–312). Beijing: Publishing House of Electronics Industry.

Yuille, A., & Kosowsky, J. (1994). Statistical physics algorithms that converge. *Neural Computation*, *6*, 341–356.

Received March 29, 1999; accepted April 27, 2001.