11th International Conference on Information Science and Technology (ICIST)
Chengdu, China, May 21-23, 2021

Solving the Travelling Salesman Problem Based on Collaborative Neurodynamic Optimization with Discrete Hopfield Networks

Hongzong Li
Department of Computer Science
City University of Hong Kong
Hong Kong, China
hongzli2-c@my.cityu.edu.hk

Jiasen Wang
Future Network Research Center
Purple Mountain Laboratories
Nanjing, China
jiasenwangcs@yeah.net

Jun Wang
Department of Computer Science
City University of Hong Kong
Hong Kong, China
jwang.cs@cityu.edu.hk

2021 11th International Conference on Information Science and Technology (ICIST) | 978-1-6654-1266-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICIST52614.2021.9440588

***Abstract*—This paper addresses the travelling salesman problem (TSP) based on collaborative neurodynamic optimization (CNO). In the CNO approach to TSP, a population of discrete Hopfield networks are employed for searching local optimal solutions and repeatedly reinitialized by using the particle swarm optimization rule towards a global optimal solution. Experimental results for solving four TSP benchmarks are reported to substantiate the efficacy of the CNO approach.**

***Index Terms*—Travelling salesman problem, discrete Hopfield network, collaborative neurodynamic optimization.**

I. Introduction

The travelling salesman problem (TSP) is a classic combinatorial optimization problem. It is to find optimal tour that visits every city exactly once for a given list of cities with the minimum total distance and back to the starting city. There are numerous areas of applications. Specific applications include, but not limited to, the quadratic assignment problem, scheduling problems, and many other intractable combinatorial optimization problems [1].

The TSP is a well known NP-hard optimization problem that has been studied extensively: e.g., [2] [3]. Due to its computational complexity, it is used as a benchmark for many combinatorial optimization methods, such as Nearest Neighborhood Search (NNS), Simulated Annealing (SA), Tabu Search (TS), Neural Networks (NN), Ant Colony Optimization (ACO) and Genetic Algorithm (GA) [4].

With the advent of neural network research, a growing number of neural network has been used to solve optimization problems. John Hopfield points out that the networks of simple and similar neurons collectively can serve as powerful computation models in his seminal papers [5], [6]. In particular, the discrete and continuous Hopfield networks in [5], [6] are applied for linear programming and combinatorial optimization including TSP [7]–[11]. Since 1990's, numerous neurodynamic models are developed for solving various optimization problems such as linear and nonlinear programming (e.g., [7], [12]–[21]), nonsmooth optimization problems (e.g., [22], [23]), generalized convex optimization problems (e.g., [24], [25]), minimax optimization problems (e.g., [26], [27], complex-valued optimization problems (e.g., [28]), distributed optimization problems (e.g., [29], [30]), bi-level optimization problems (e.g., [31]), and combinatorial optimization (e.g., [32], [33]). In recent years, collaborative neurodynamic optimization (CNO) approaches becomes viable for global and combinatorial optimization [34]–[37]. In the CNO framework, multiple neurodynamic models work in parallel in search for local optimal solutions to a given optimization problem until convergence and the search processes repeat with updated initial states by using a meta-heuristic rule (e.g., PSO) in search for a globally optimal solution. It is proven theoretically and demonstrated experimentally that CNO approaches are almost surely convergent to global optima of global and combinatorial optimization problems [34]–[36], [38], [39].

This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region of China under Grant 11208517, Grant 11202318, and Grant 11202019.

In this paper, we propose a CNO approach to TSP based on discrete Hopfield networks (DHNs). We first formulate the TSP as a quadratic constrained binary optimization and reformulate it into a quadratic unconstrained binary optimization (QUBO) problem [40] via a penalty function converted from equality constraints. Then we propose a collaborative neurodynamic approach with a population of discrete Hopfield networks re-initialized repeatedly by using a particle swarm optimization update rule for solving the formulated QUBO problem.

The remainder of this paper is organized as follows. Section II provides necessary preliminary information. Section III states the problem formulation. Section IV presents the proposed neurodynamics-based approach to TSP. Section V reports experimental results in four data sets. Section VI concludes the paper.

II. Preliminaries

A. Discrete Hopfield Network

The discrete Hopfield network is a archetypical recurrent neural network characterized by its binary or bipolar states and activation function operating in discrete time [5]:

$$u(t) = Wx(t) - \theta, \tag{1}$$
$$x(t+1) = \sigma(u(t)), \tag{2}$$

where $u \in \mathbb{R}^n$ is the net-input vector, $x \in \mathbb{R}^n$ is the state vector, $W \in \mathbb{R}^{n\times n}$ is the connection weight matrix, $\theta \in \mathbb{R}^n$ is the threshold vector, and $\sigma(\cdot)$ is a vector-valued discontinuous activation function defined element-wisely as follows:

$$x_i(t+1) = \sigma(u_i) = \begin{cases} 0 & \text{if } \ u_i(t) \le 0, \\ 1 & \text{if } \ u_i(t) > 0. \end{cases}$$

It is shown that the discrete Hopfield network (2) is globally stable at an equilibrium $\bar{x}$ (i.e., $\lim_{t\to\infty} x(t) = \bar{x}$), if the connection weight matrix is symmetric (i.e., $W = W^T$), the main diagonal elements of W is zero (i.e., $w_{ii} = 0$, $\forall i$), and the activation is carried out asynchronously [5].

It is also shown [5] that the discrete Hopfield network is globally convergent to a local minima of the following combinatorial optimization problem with binary or bipolar decision variables:

$$\begin{aligned} \min \ & -\frac{1}{2}x^T W x + \theta^T x \\ \text{s.t.} \ & x \in \{0,1\}^n. \end{aligned} \tag{3}$$

That is, an equilibrium $\bar{x}$ is a local optimum of the optimization problem above. Note that the right-hand side of eqn. (1) is equal to the positive gradient of the objective function to be maximized or negative gradient of he objective function to be minimized. In other words, the DHN neurodynamics form a discrete gradient flow moving among vertices of the unit hypercube coordinate-wisely.

In view of the binary nature of state variable $x_i \in \{0,1\}$, $x_i^2 = x_i$ for $i = 1,2,\ldots,n$. As a result, the diagonal elements of the weight matrix in the quadratic term of (1) can always set to be zeros and add an equivalently linear term $\mathrm{diag}(w_{11},\ldots,w_{nn})x$.

The asynchronous activation of the discrete Hopfield network entails a long convergence time. To expedite the convergence of DHNs, it is better to activate the DHN neurons synchronously in batches. In the literature, there are several methods to activate neuronal states synchronously in batches [41]–[44]. In particular, it has been proved in [44] that the DHN is globally stable if the neurons are activated synchronously in batches, where the neurons in a batch are not directly connected.

B. Particle Swarm Optimization

Particle swarm optimization (PSO) is a class of meta-heuristic optimization algorithms. Like scattered search, a population of candidate solutions (so called particles) search better solutions by exchanging information among them according to their momentum and best known solutions from their individual and grouped particles.

Let N denote the number of the particles in the swarm, $p_i \in \mathbb{R}^n$ denote the position vector of the i^{th} particle (candidate solution) and p_i^* denote the best position found by the i^{th} particle individually ($i = 1,2,\ldots,N$), p^* denote the best position known to the swarm (solution set). For $i = 1,2,\ldots,N$, the velocity v and the position x are updated towards its global best p^* locations by the following updating rules:

$$\begin{cases} v_i(t+1) = & c_0 v_i(t) + c_1 r_1 (p_i^* - p_i(t)) \\ & + c_2 r_2 (p^* - p_i(t)), \\ p_i(t+1) = & p_i(t) + v_i(t+1), \end{cases} \tag{4}$$

where $c_0 \in [0,1]$ is an inertia parameter, $c_1, c_2 \in [0,1]$ are two acceleration constants, and $r_1, r_2 \in [0,1]$ are two random numbers.

C. Collaborative Neurodynamic Optimization

It is challenging for an individual neurodynamic model to solve some complex optimization problems such as global optimization problems with nonconvex objective functions or constraints and combinatorial optimization problems with binary variables, as a single neurodynamic model may stuck in a local minimum [45]. In recent years, several collaborative neurodynamic optimization approaches with multiple neurodynamic models are developed for solving distributed optimization problems (e.g., [29], [30]), bi-level optimization problems (e.g., [31]), nonconvex and global optimization problems (e.g., [34]–[36], [46]), multi-objective optimization (e.g., [30], [47]), and combinatorial optimization problems (e.g., [36], [37]).

Collaborative neurodynamic optimization is a framework of hybrid intelligence that designed for solving global optimization problems. It integrates the local search ability of individual neurodynamic models together with the global search ability of CNO approach. For global, combinatorial, and multi-objective optimization, initial states of a group of neurodynamic models are repetitively updated by using meta-heuristics such as particle swarm optimization. It is proven that a collaborative neurodynamic approaches are almost surely convergent to the global optimal solutions of the optimization problems [34]–[37], [46].

III. Problem Formulations

A. Problem Formulation

Consider the following travelling salesman problem in a vertex representation form [1]:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1, l\neq j}^{n} d_{jl} x_{ij} x_{i+1,l}, \tag{5a}$$
$$\text{s.t.} \ \sum_{i=1}^{n} x_{ij} = 1, \ j = 1,...,n, \tag{5b}$$
$$\sum_{j=1}^{n} x_{ij} = 1, \ i = 1,...,n, \tag{5c}$$
$$x_{ij} \in \{0,1\}, \ i,j = 1,...,n. \tag{5d}$$

where n is the number of cities, d_{jl} is the distance between cities j and l, the decision variable $x_{ij} = 1$ means that the ith stop is city j, and $x_{n+1,k} := x_{1,k}$.

The objective function (5a) quantifies the total distance of tour. Constraints in (5b) ensure that the salesman enter a city only once. Constraints in (5c) ensure that the salesman leaves each city only once. Constraints in (5b) and (5c) together imply that $X = [x_{ij}]$ is a permutation matrix. As a result, there are $n!$ feasible solutions in the QAP. If the origin is assumed to be given, the number of feasible solutions is re are $(n-1)!$.

The above formulation is known as the Euclidean TSP where the distance matrix $D = [d_{jl}]$ is symmetric (i.e., $d_{lj} = d_{jl}$ for all l, j).

B. Problem Reformulation

To facilitate subsequent reformulation, the objective function is rewritten as

$$f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\gamma_{ik}d_{jl}x_{ij}x_{kl},$$

where $\gamma_{ik} = 1$ if $k = i+1, j \neq l$, else $\gamma_{ik} = 0$.

Let $\hat{D} \in \mathbb{R}^{n^2\times n^2}$ and its elements be defined as

$$\hat{d}_{(i-1)n+j,(k-1)n+l} = \gamma_{ik}d_{jl},$$

for $i, j, k, l = 1, ..., n$.

Let $x = [x_{11}, x_{12}, ..., x_{1n}, x_{21}, x_{22}, ..., x_{2n}, ..., x_{nn}]^T \in \{0,1\}^{n^2}$, The objective function $f(x)$ is rewritten in a vector form:

$$f(x) = x^T\hat{D}x.$$

To handle the constraints in (5b) and (5c), a quadratic penalty function with two terms is defined as follows:

$$p(x) \quad = \quad \frac{1}{2}[\sum_{j=1}^{N}(\sum_{i=1}^{N}x_{ij}-1)^2 + \sum_{i=1}^{N}(\sum_{j=1}^{N}x_{ij}-1)^2].$$

Let

$$A = \begin{bmatrix} I_1 & I_2 & \cdots & I_n \\ I & I & \cdots & I \end{bmatrix} \in \{0,1\}^{2n\times n^2},$$

$$e = \underbrace{[1, 1, ..., 1]^T}_{2n},$$

where $I_i \in \{0,1\}^n$ is a matrix with ones in the ith row and zeros in other rows.

The penalty function $p(x)$ then can be rewritten as follow:

$$p(x) = \frac{1}{2}\|Ax - e\|^2.$$

Let a penalized objective function be defined as follows:

$$f_\rho(x) \quad = \quad f(x) + \rho p(x),$$

where ρ is a sufficiently large positive penalty parameter. The penalized objective function is rewritten as follows:

$$f_\rho(x) = x^T\hat{D}x + \frac{\rho}{2}\|Ax - e\|^2.$$

Based on the penalty function, the original problem in (5) is reformulated into a QUBO problem as follows:

$$\begin{aligned} &\text{minimize} \quad f_\rho(x), \\ &\text{subject to} \quad x \in \{0,1\}^{N^2}. \end{aligned} \tag{6}$$

It is known that problems (5) and (6) are equivalent in terms of their optimal solutions if the penalty parameter is sufficiently large [48].

Let

$$\hat{W} = -[\hat{D}^T + \hat{D} + \rho(A^TA)]$$

$$W = \hat{W} - \text{diag}(\hat{w}_{11}, \hat{w}_{22}, ..., \hat{w}_{n^2n^2}) \tag{7}$$

$$\theta = -\frac{1}{2}[\hat{w}_{11}, \hat{w}_{22}, ..., \hat{w}_{n^2n^2}]^T - \rho A^T e \tag{8}$$

where $w_{(i-1)N+j,(k-1)N+l} = -d_{jl}$ if $i = k+1$ and $k \neq l$, $w_{(i-1)N+j,(k-1)N+l} = -d_{jl}$ if $i = k-1$ and $j \neq l$, $w_{(i-1)N+j,(k-1)N+l} = -d_{jl}$ if $i = 1$, $k = N$ and $j \neq l$, $w_{(i-1)N+j,(k-1)N+l} = -d_{jl}$ if $i = N$, $k = 1$ and $j \neq l$, $w_{(i-1)N+j,(k-1)N+l} = -\rho$ if $i = k$ and $j \neq l$, $w_{(i-1)N+j,(k-1)N+l} = -\rho$ if $i \neq k$ and $j = l$, $w_{(i-1)N+j,(k-1)N+l} = 0$ if $i = k$ and $j = l$, $w_{(i-1)N+j,(k-1)N+l} = 0$ if $i = k-2, k-3, ..., k-(N-2)$ and $j \neq l$, $w_{(i-1)N+j,(k-1)N+l} = 0$ if $i = k+2, k+3, ..., k+(N-2)$ and $j \neq l$ and $\theta_{(i-1)N+j} = -\rho$.

The TSP is reformulated as problem (3) with W in (7) and θ in (8).

IV. CNO-based Algorithms

In this paper, the CNO approach to TSP employs a population of DHNs. It is shown in [44] that the neurons can be activated synchronously in batches in $2n$ batches if n is even and in $3n$ batches if n is odd. As a matter of fact, finding the optimal partition of neurons without inter-neuron connections is a combinatorial optimization problem. In this study, by examining the distribution of zero elements in W, we found that the neurons can be partitioned into $2n+1$ batches where the neurons in the same batch are not connected to each other. As a result, neurons in the same batch can also be activated synchronously. Algorithm 1 is designed to compute the batches.

Algorithm 2 is designed to realize the DHN activated in batches.

$U(0,1)$ denotes a random variable with image $[0,1]$, $P_{[0,1]}(x)$ denotes a projection function with image $[0,1]$, N denotes the number of Hopfield networks, $c_0 \in [0,1]$ is an inertia parameter, $c_1, c_2 \in [0,1]$ are two acceleration constants, Similar to the approach in [36], a CNO approach by using multiple Hopfield networks is presented in Algorithm 3. Step 2 - step 7 employ a population of DHNs for scatter search. Step 9 - step 12 update the global optimal solution. Step 15 - step 20 reinitialize the state of DHNs.

V. Experimental results

A. Setups

There are large numbers of TSP test problem instances. TSPLIB is such a collection of TSP instances [49]. In this

Algorithm 1: Determining the batches of neurons

Input: The number of cities n.
Output: Batches of the neurons.

for $i = 1 : 2n+1$ **do**
  ξ=i;
  while $\xi \leq n^2$ **do**
    Add ξ to the $\mathscr{b}_i$, where $\mathscr{b}_i$ is the index set of the ith batch;
    $\xi \leftarrow i + 2n + 1$;
  end
end
return $\mathscr{B}$.

Algorithm 2: The DHN activated in batches

Input: The number of cities n, Initial solution: $x_1, x_2, ..., x_{n^2}$, Graph G = (V,E) where V is the set of vertices or nodes and E is the set of arcs or edges between vertices, Batches of the neurons $\mathscr{B}$.
Output: Equilibrium $\bar{x}$ of the DHN.

while $\Delta = 1$ **do**
  $\Delta \leftarrow 0$;
  for *i=1:2n+1* **do**
    for *j=1:*n^2 **do**
      if $j \in \mathscr{b}_i$ **then**
        $u_j \leftarrow \sum_{k=1}^{n^2} w_{jk} x_k - \theta_j$;
        if $x_j \neq \sigma(u_j)$ **then**
          $x_j \leftarrow \sigma(u_j)$;
          $\Delta \leftarrow 1$
        end
      end
    end
  end
end
return $\bar{x}$.

Algorithm 3: TSP based on CNO approach

Input: Population size N, initial states $[y^{(1)}(0), ..., y^{(N)}(0)] \in \{0,1\}^{n^2 \times N}$, velocity vector $v^{(i)} \in [-1,1]^{n^2}$, termination criteria M, termination counter m, penalty parameter ρ, particle/group best solutions $y^{(i)}$/ y^*, $f(y^{(i)}) = f(y^*) = \infty$, PSO-based state initialization parameters c_0, c_1 and c_2.
Output: y^*.

while $m \leq M$ **do**
  for $i = 1$ **to** N **do**
    Obtain the equilibrium state $\bar{y}^{(i)}$ of the ith Hopfield network (Algorithm 2) with initial state $y^{(i)}(0)$ and penalty parameter ρ by using asynchronous iteration;
    if $f(\bar{y}^{(i)}) < f(y^{(i)})$ **then**
      $y^{(i)} \leftarrow \bar{y}^{(i)}$;
    end
  end
  $i^* = \arg\min_i \{f(y^{(1)}), ..., f(y^{(i)}), ..., f(y^{(N)}\}$;
  if $f(y^{(i^*)}) < f(y^*)$ **then**
    $y^* \leftarrow y^{(i^*)}$;
    $m \leftarrow 0$;
  else
    $m \leftarrow m + 1$;
  end
  for $i = 1$ **to** N **do**
    Update velocity $v^{(i)} = c_0 v^{(i)} + c_1 U(0,1)(y^{(i)} - \bar{y}^{(i)}) + c_2 U(0,1)(y^* - \bar{y}^{(i)})$;
    Update initial state $y^{(i)}(0) = y^{(i)}(0) + v^{(i)}$;
    $y^{(i)}(0) = P_{[0,1]}(y^{(i)}(0))$;
    $y^{(i)}(0) = \text{round}(y^{(i)}(0))$;
  end
end
return y^*.

paper, we evaluate the proposed CNO approach on datasets ulysses16, burma14, ulysses22 and bays29 [50], where the number in name indicates the number of cities.

In this paper, the hyper-parameters setting is based on Monte Carlo test. The selection of two hyper-parameters N (population size) and M (termination criteria) in Algorithm 3 is based on a Monte Carlo test with 10-run from random initial states on the data set burma14 and data set ulysses16. Fig. 1 and Fig. 6 depict the Monte Carlo test results in terms of mean values and standard deviations obtained by using the CNO approach on burma14 and ulysses16 with several M and N. As shown in Fig. 1, the objective function values reach its minimum for most runs with $M = 20$ and $N = 200$ on burma14. As shown in Fig. 6, the objective function values reach its minimum for most runs with $M = 20$ and $N = 300$ on ulysses16. As a result, the two hyper-parameters are set as $M = 20$ and $N = 200$ on burma14 and set as $M = 20$ and $N = 300$ on ulysses16 in the experiments.

In Algorithm 3, the maximum number of iterations is set as 1500. In the PSO rule (4), $c_0 = 1$ and $c_1 = c_2 = 0.1$.

B. Experimental Results

Fig. 2 and Fig. 7 depicts the batches of neurons on burma14 and ulysses16 respectively . The neurons in the same batches can also be activated synchronously since they are not connected to each other. Fig. 3, Fig. 8, Fig. 11 and Fig. 14 plots four snapshots of the dynamic behaviors of a single DHN (Algorithm 2) and its associated function values, including transient states x, objective function $f(x)$, penalty function $p(x)$ with $\rho = 10^6$ on burma14, ulysses16, ulysses22 and bays29, respectively. Moreover, when the penalty function reaches zero, the objective function and the penalized objective function are equal. Fig. 4, Fig. 9, Fig. 12 and Fig. 15 illustrate the convergent behavior of the CNO approach on burma14, ulysses16, ulysses22 and bays29, respectively. Fig. 5, Fig. 10

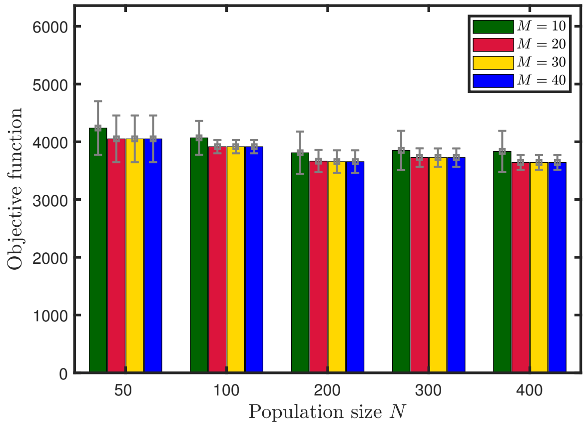

Fig. 1. Monte Carlo test results in terms of Mean values and standard deviations obtained by using the CNO approach on burma14 with several M and N.

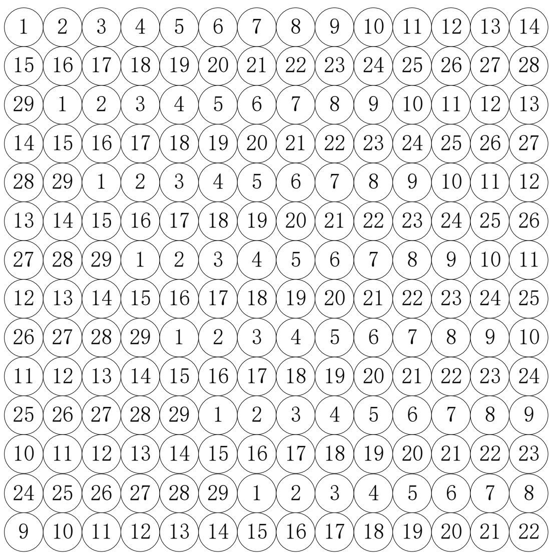

Fig. 2. The batches of neurons generated by Algorithm 1 on the data set burma14.

and Fig. 13 depict the tours on burma14, ulysses16, ulysses22 and bays29, respectively. Table I depicts the results in terms of best/worst values, mean values, and standard deviations with different data sets.

VI. Concluding Remarks

In this paper, a collaborative neurodynamic approach by using a population of DHN activated in batches is used for solving TSP. The TSP is formulated as a constrained binary quadratic programming problem and is reformulated as a QUBO problem with a penalty function. In the framework of collaborative neurodynamic optimization framework, the initial states of a few hundreds of Hopfield networks are updated based on particle swarm optimization. Experimental results are elaborated to substantiate the efficacy of the approach to TSP.

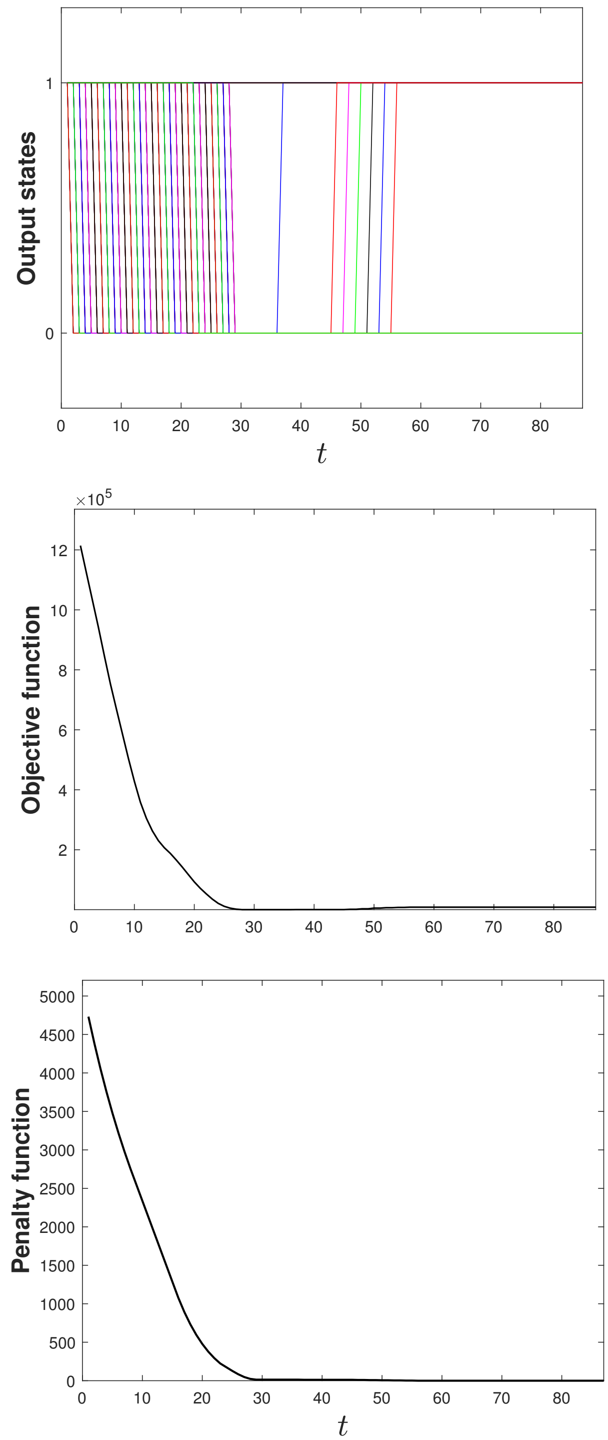

Fig. 3. Snapshots of neuronal states, objective function values and penalty function values of DHN (Algorithm 2) on the data set burma14 with $M = 20$ and $N = 200$.

To improve the scalability of the collaborative neurodynamic approach to the TSP, unsupervised, supervised, and semi-supervised learning algorithms should be integrated. For example, clustering, deep learning, and reinforcement learning could be used to generate prior knowledge as TSP constraints

TABLE I

THE RESULTS IN TERMS OF BEST/WORST VALUES, MEAN VALUES, AND STANDARD DEVIATIONS.

Data Set	n	# of solutions	# of neurons	# of batches	M	N	optimum	best / worst	mean±std
burma14	14	6.2×10^9	196	29	20	200	3323	<u>3323</u> / 4033	3674±190
ulysses16	16	1.3×10^{12}	256	33	20	300	6859	<u>6859</u> / 7828	7365±295
ulysses22	22	5.1×10^{19}	484	45	30	3000	7013	<u>7013</u> / 8413	7695±354
bays29	29	3.0×10^{29}	841	59	30	3000	2020	2254 / 2839	2555±144

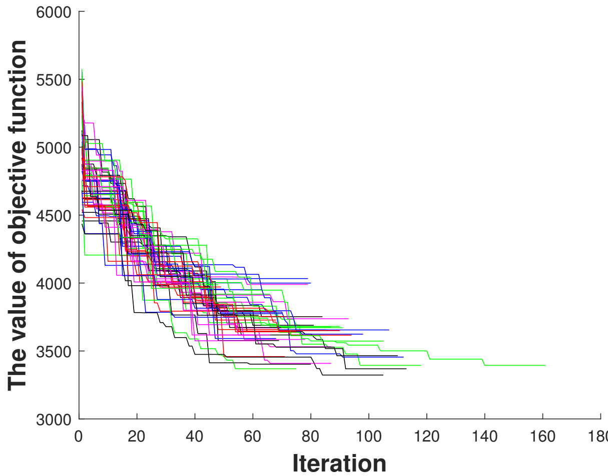

Fig. 4. The convergent behavior of the CNO approach on the data set burma14 with $M = 20$ and $N = 200$.

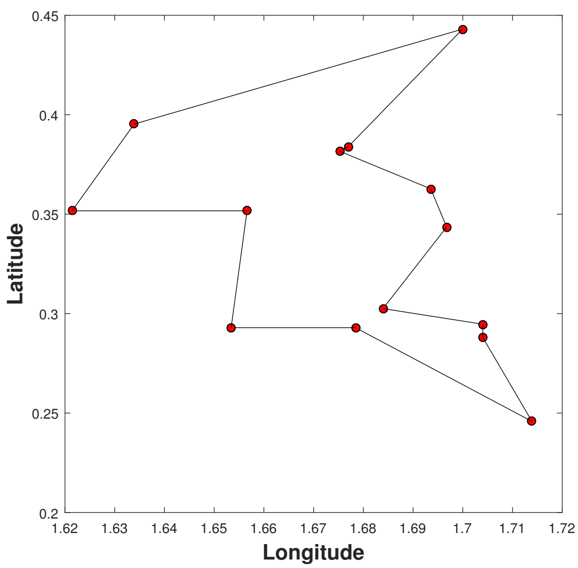

Fig. 5. Optimal tour with total distance 3323 obtained by using the CNO approach on the data set burma14 with $M = 20$ and $N = 200$.

for to reduce temporal and spatial complexity of the proposed neurodynamic approach.

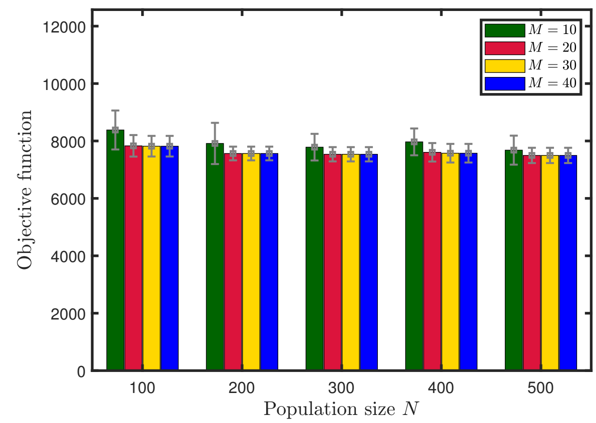

Fig. 6. Monte Carlo test results in terms of Mean values and standard deviations obtained by using the CNO approach on ulysses16 with several M and N.

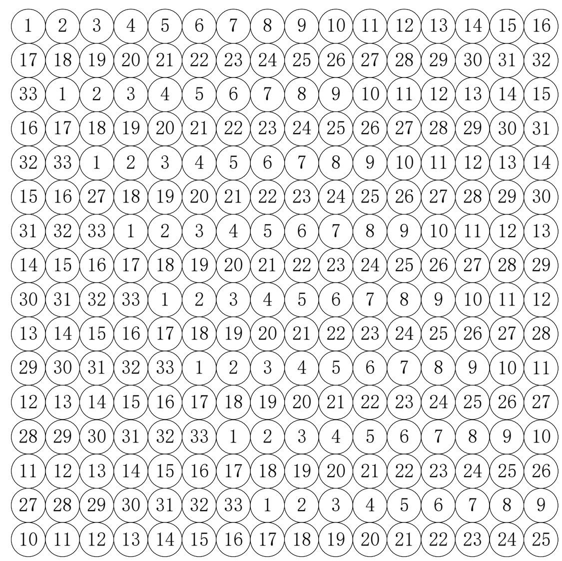

Fig. 7. The batches of neurons generated by Algorithm 1 on the data set ulysses16.

REFERENCES

[1] J. Wang, "A deterministic connectionist machine for the traveling salesman problem," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 1990, pp. 374–375.

[2] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, "Erratum: The traveling salesman problem: A guided tour of combinatorial optimization," *Journal of the Operational Research Society*, vol. 37, no. 6, pp. 655–655, 1986.

[3] S. A. Mulder and D. C. Wunsch II, "Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks," *Neural Networks*, vol. 16, no. 5-6, pp. 827–832, 2003.

[4] Y. F. Liao, D. H. Yau, and C. L. Chen, "Evolutionary algorithm to traveling salesman problems," *Computers & Mathematics with Applications:*

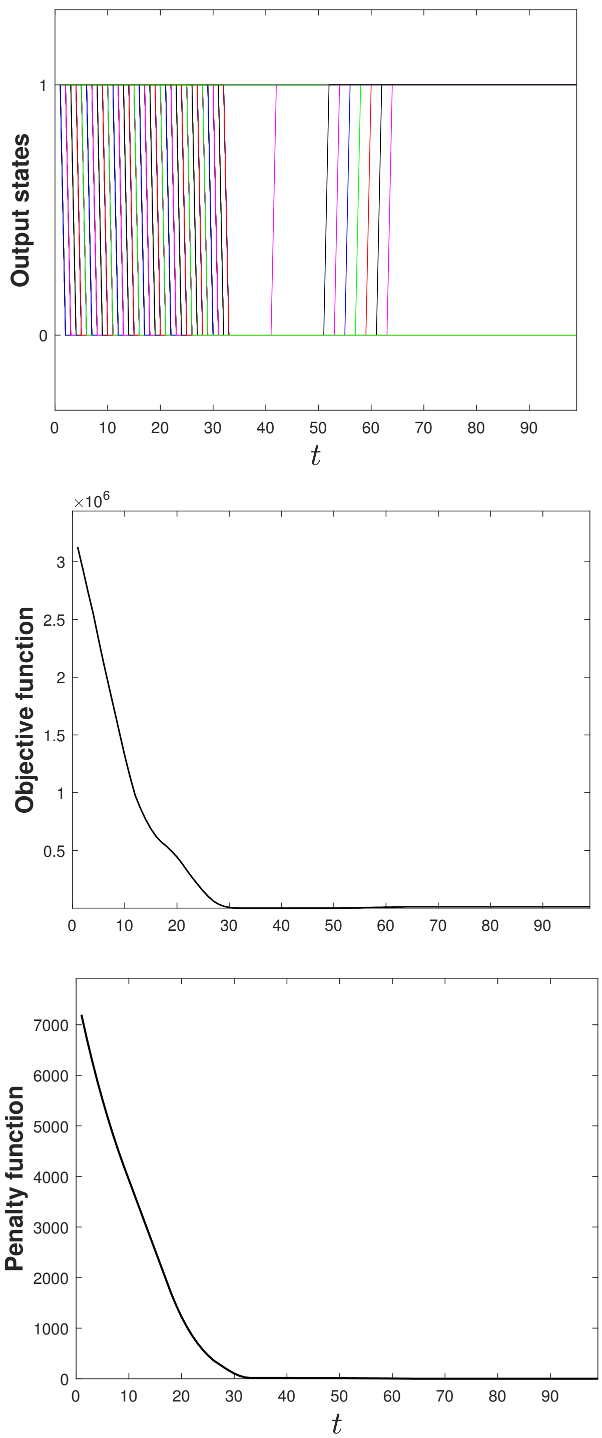

Fig. 8. Snapshots of neuronal states, objective function values and penalty function values of DHN (Algorithm 2) on the data set ulysses16 with $M = 20$ and $N = 300$.

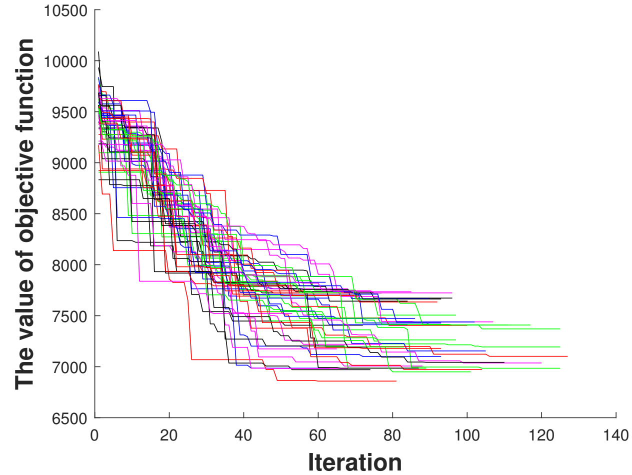

Fig. 9. The convergent behavior of the CNO approach on the data set ulysses16 with $M = 20$ and $N = 300$.

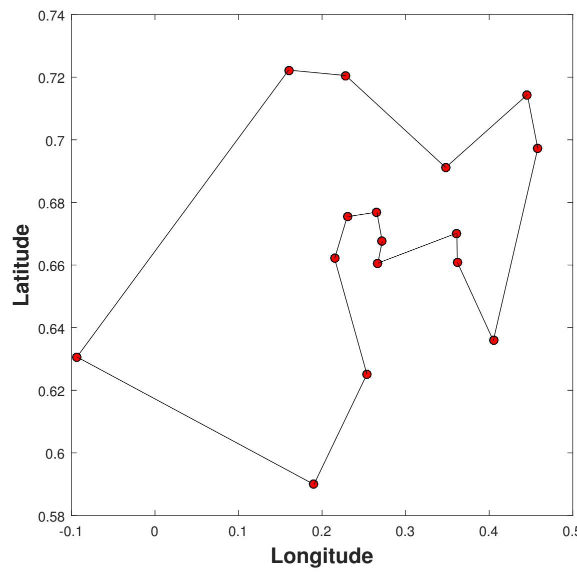

Fig. 10. Optimal tour with total distance 6859 obtained by using the CNO approach on the data set ulysses16 with $M = 20$ and $N = 300$.

An International Journal, 2012.

[5] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[6] ——, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences*, vol. 81, no. 10, pp. 3088–3092, 1984.

[7] D. W. Tank and J. J. Hopfield, "Simple 'neural'optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits and Systems*, vol. 33, no. 5, pp. 533–541, 1986.

[8] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, no. 3, pp. 141–152, 1985.

[9] ——, "Computing with neural circuits: a model," *Science*, vol. 233, no. 4764, pp. 625–633, 1986.

[10] P. M. Talaván and J. Yáñez, "Parameter setting of the hopfield network applied to tsp," *Neural Networks*, vol. 15, no. 3, pp. 363–373, 2002.

[11] K. C. Tan, H. Tang, and S. S. Ge, "On parameter settings of hopfield networks applied to traveling salesman problems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 5, pp. 994–1002, 2005.

[12] Y. Xia and J. Wang, "A general methodology for designing globally convergent optimization neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1331–1343, 1998.

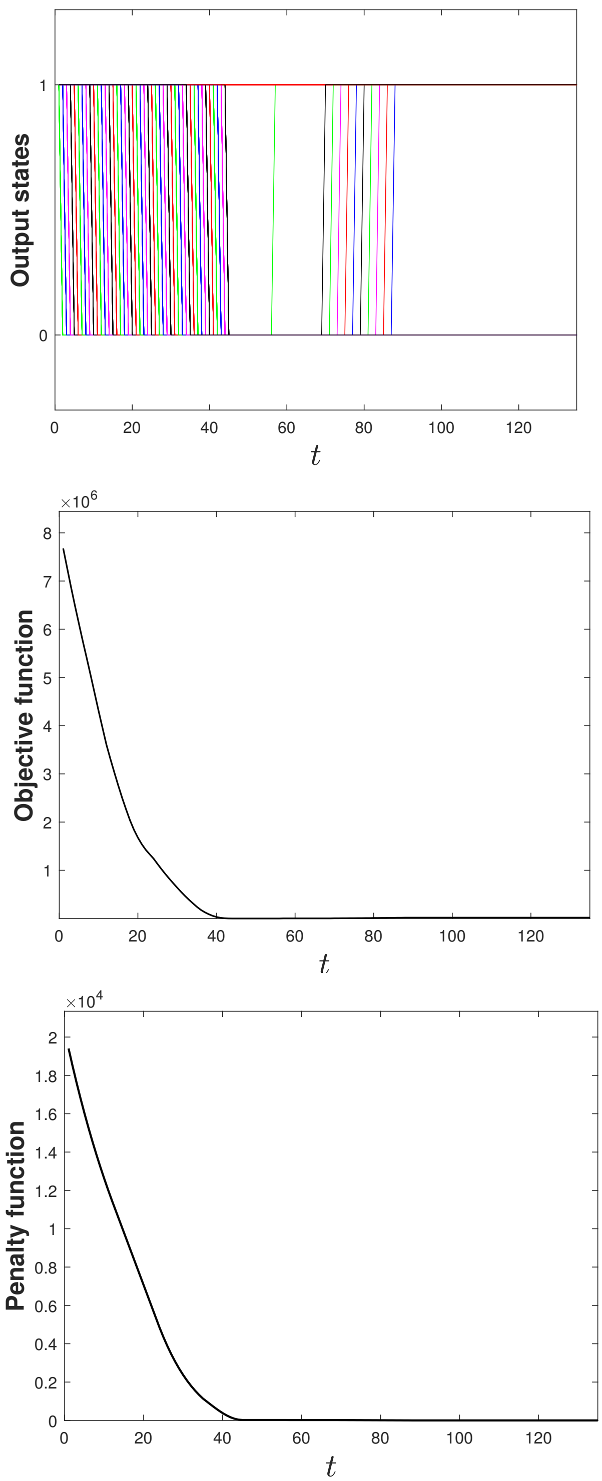

Fig. 11. Snapshots of neuronal states, objective function values and penalty function values of DHN (Algorithm 2) on the data set ulysses22 with $M = 30$ and $N = 3000$.

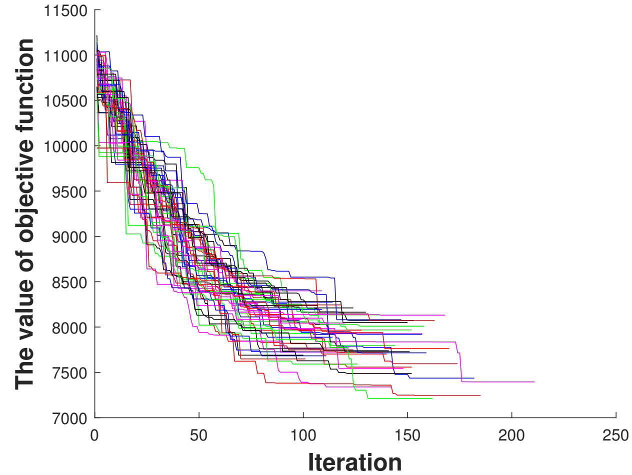

Fig. 12. The convergent behavior of the CNO approach on the data set ulysses22 with $M = 30$ and $N = 3000$.

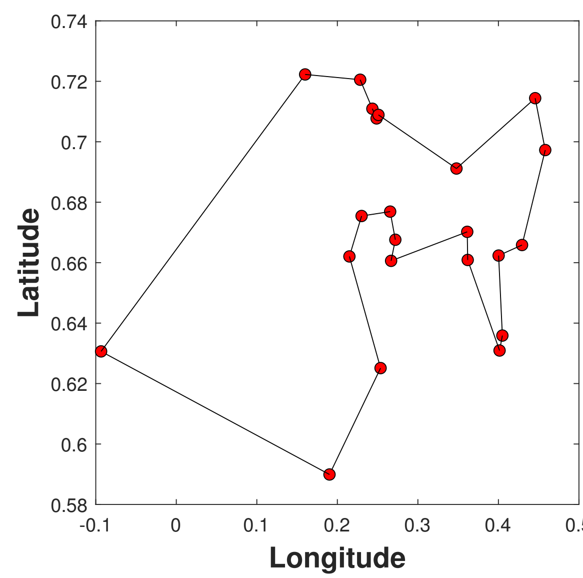

Fig. 13. Optimal tour with total distance 7013 obtained by using the CNO approach on the data set ulysses22 with $M = 30$ and $N = 3000$.

[13] ——, "A general projection neural network for solving monotone variational inequalities and related optimization problems," *IEEE Transactions on Neural Networks*, vol. 15, no. 2, pp. 318–328, 2004.

[14] ——, "A recurrent neural network for solving nonlinear convex programs subject to linear constraints," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 379–386, 2005.

[15] S. Liu and J. Wang, "A simplified dual neural network for quadratic programming with its KWTA application," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1500–1510, 2006.

[16] Y. Xia, G. Feng, and J. Wang, "A novel neural network for solving nonlinear optimization problems with inequality constraints," *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1340–1353, 2008.

[17] X. Hu and J. Wang, "An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application," *IEEE Transactions on Neural Networks*, vol. 19, no. 12, pp. 2022–2031, 2008.

[18] Q. Liu and J. Wang, "A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming," *IEEE Transactions on Neural Networks*, vol. 19, no. 4, pp. 558–570, 2008.

[19] Q. Liu, T. Huang, and J. Wang, "One-layer continuous-and discrete-time projection neural networks for solving variational inequalities and related optimization problems," *IEEE Transactions on Neural Networks*

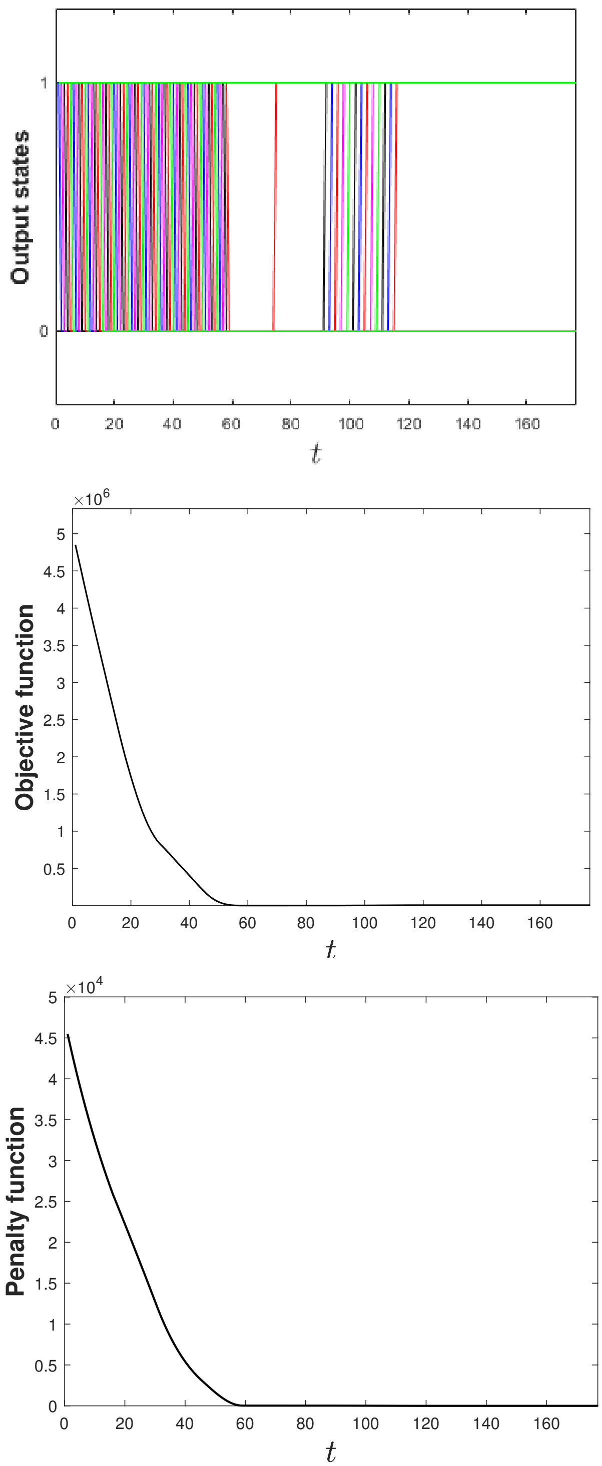

Fig. 14. Snapshots of neuronal states, objective function values and penalty function values of DHN (Algorithm 2) on the data set bays29 with $M = 30$ and $N = 3000$.

and Learning Systems, vol. 25, no. 7, pp. 1308–1318, July 2014.

[20] Y. Xia and J. Wang, "A bi-projection neural network for solving constrained quadratic optimization problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 214–224, 2016.

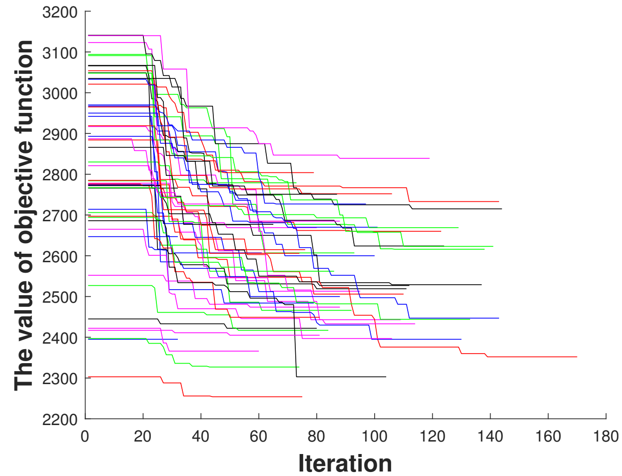

Fig. 15. The convergent behavior of the CNO approach on the data set bay29 with $M = 30$ and $N = 3000$.

[21] Y. Xia, J. Wang, and W. Guo, "Two projection neural networks with reduced model complexity for nonlinear programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2020–2029, 2020.

[22] Q. Liu and J. Wang, "Finite-time convergent recurrent neural network with a hard-limiting activation function for constrained optimization with piecewise-linear objective functions," *IEEE Transactions on Neural Networks*, vol. 22, no. 4, pp. 601–613, 2011.

[23] ——, "A one-layer projection neural network for nonsmooth optimization subject to linear equalities and bound constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 812–824, 2013.

[24] X. Hu and J. Wang, "Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1487–1499, 2006.

[25] Z. Guo, Q. Liu, and J. Wang, "A one-layer recurrent neural network for pseudoconvex optimization subject to linear equality constraints," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1892–1900, 2011.

[26] Q. Liu and J. Wang, "A projection neural network for constrained quadratic minimax optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2891–2900, 2015.

[27] X. Le and J. Wang, "A two-time-scale neurodynamic approach to constrained minimax optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 620–629, 2017.

[28] S. Zhang, Y. Xia, and J. Wang, "A complex-valued projection neural network for constrained optimization of real functions in complex variables," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3227–3238, 2015.

[29] Q. Liu, S. Yang, and J. Wang, "A collective neurodynamic approach to distributed constrained optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 8, pp. 1747–1758, 2017.

[30] S. Yang, Q. Liu, and J. Wang, "A collaborative neurodynamic approach to multiple-objective distributed optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 981–992, April 2018.

[31] S. Qin, X. Le, and J. Wang, "A neurodynamic optimization approach to bilevel quadratic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2580–2591, Nov 2017.

[32] Z.-B. Xu, G.-Q. Hu, and C.-P. Kwong, "Asymmetric Hopfield-type networks: theory and applications," *Neural Networks*, vol. 9, no. 3, pp. 483–501, 1996.

[33] J. Wang, "Discrete Hopfield network combined with estimation of distribution for unconstrained binary quadratic programming problem," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5758 – 5774, 2010.

[34] Z. Yan, J. Wang, and G. Li, "A collective neurodynamic optimization ap-

proach to bound-constrained nonconvex optimization," *Neural Networks*, vol. 55, pp. 20–29, 2014.
[35] Z. Yan, J. Fan, and J. Wang, "A collective neurodynamic approach to constrained global optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1206–1215, 2017.
[36] H. Che and J. Wang, "A collaborative neurodynamic approach to global and combinatorial optimization," *Neural Networks*, vol. 114, pp. 15 – 27, 2019.
[37] H. Che and J. Wang, "A two-timescale duplex neurodynamic approach to mixed-integer optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 36–48, 2021.
[38] J. Fan and J. Wang, "A collective neurodynamic optimization approach to nonnegative matrix factorization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2344–2356, 10 2017.
[39] X. Li, J. Wang, and S. Kwong, "A discrete-time neurodynamic approach to sparsity-constrained nonnegative matrix factorization," *Neural Computation*, vol. 32, no. 8, pp. 1531–1562, 2020.
[40] F. Glover, G. Kochenberger, and Y. Du, "Quantum bridge analytics i: a tutorial on formulating and using qubo models," *4OR*, vol. 17, no. 4, pp. 335–371, 2019.
[41] E. Goles-Chacc, F. Fogelman-Soulié, and D. Pellegrin, "Decreasing energy functions as a tool for studying threshold networks," *Discrete Applied Mathematics*, vol. 12, no. 3, pp. 261–277, 1985.
[42] B. Cernuschi-Frías, "Partial simultaneous updating in Hopfield memories," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 4, pp. 887–888, 1989.
[43] D.-L. Lee, "New stability conditions for Hopfield networks in partial simultaneous update mode," *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 975–978, 1999.
[44] J. Muñoz-Pérez, A. Ruiz-Sepúlveda, and R. Benítez-Rochel, "Parallelism in binary hopfield networks," in *Advances in Computational Intelligence*, J. Cabestany, I. Rojas, and G. Joya, Eds. Springer Berlin Heidelberg, 2011, pp. 105–112.
[45] M. Peng, N. K. Gupta, and A. F. Armitage, "An investigation into the improvement of local minima of the hopfield network," *Neural networks*, vol. 9, no. 7, pp. 1241–1253, 1996.
[46] H. Che and J. Wang, "A two-timescale duplex neurodynamic approach to biconvex optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 8, pp. 2503–2514, 2019.
[47] M.-F. Leung and J. Wang, "A collaborative neurodynamic approach to multiobjective optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5738 – 5748, 2018.
[48] G. A. Kochenberger, F. Glover, B. Alidaee, and C. Rego, "A unified modeling and solution framework for combinatorial optimization problems," *OR Spectrum*, vol. 26, no. 2, pp. 237–250, 2004.
[49] C. A. Floudas, P. M. Pardalos, C. Adjiman, W. R. Esposito, Z. H. Gümüs, S. T. Harding, J. L. Klepeis, C. A. Meyer, and C. A. Schweiger, *Handbook of test problems in local and global optimization*. Springer Science & Business Media, 2013, vol. 33.
[50] C. Floudas, P. Pardalos, and et al., *Handbook of Test Problems in Local and Global Optimization*. Kluwer Academic Publishers, 2000.