Machine-Cell and Part-Family Formation via Neurodynamics-Driven Constrained Binary Matrix Factorization

Hongzong Li, *Member, IEEE* and Jun Wang, *Life Fellow, IEEE*

***Abstract*—The formation of part families and their corresponding machine cells is a critical phase in the design of a cellular manufacturing system. This paper presents a constrained binary matrix factorization approach to machine-cell and part-family formation. A constrained binary matrix factorization problem is formulated for machine-cell and part-family formation to minimize the number of exceptional elements. The constrained binary matrix factorization is further reformulated to a quadratic unconstrained binary optimization problem by reducing the quartic objective function of the binary matrix factorization problem to a quadratic one and penalizing the violation of constraints. A neurodynamics-driven algorithm is proposed to solve the reformulated quadratic problem by leveraging several Boltzmann machines for searching solutions and a particle swarm optimization rule to reinitialize the neuronal states upon their local convergence to escape from local solutions and move toward global optimal ones. Experimental results on eighteen benchmark datasets are presented to showcase the superior performance of the proposed approach in terms of four criteria.**

***Index Terms*—Cellular manufacturing; machine-cell and part-family formation; binary matrix factorization; quadratic unconstrained binary optimization (QUBO); collaborative neurodynamic optimization; Boltzmann machine.**

I. INTRODUCTION

In manufacturing industries, the design and optimization of manufacturing systems, such as cellular manufacturing systems, plays a critical role in improving productivity and minimizing costs [1]–[3]. The formation of machine cells and part families is an essential undertaking in the development of cellular manufacturing systems [4]–[18]. It involves two critical tasks: forming part families and forming machine cells [5]. Part-family formation is concerned with grouping parts with similar characteristics and processing requirements into families. Machine-cell formation involves grouping machines into cells based on their capabilities and compatibility with part families. In a cellular manufacturing system, each part family is processed in its corresponding machine cell to minimize material handling costs and maximize the productivity of the manufacturing system. Fig. 1 illustrates a conceptual diagram of machine-cell and part-family formation. Various machines (e.g., lathes, millers, drills, grinders) and parts initially associated with processing requirements. The objective is to simultaneously partition machines into machine cells and parts into part families such that each part family can be processed predominantly within its corresponding machine cell, minimizing exceptional elements and improving manufacturing efficiency. To manufacture a batch of products based on customers' orders, a production planner needs to provide a solution to form machine cells and form part families for a cellular manufacturing system. Once the customers' demands change, the production planner must provide new solutions to meet the dynamic manufacturing environments.

This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region of China under Grant 11203721.

H.-Z. Li is with the Hong Kong Generative AI R&D Center, Hong Kong University of Science and Technology, Hong Kong (email: lihongzong@ust.hk).

J. Wang is with the Department of Computer Science and the Department of Data Science, City University of Hong Kong, Hong Kong. (email: jwang.cs@cityu.edu.hk).

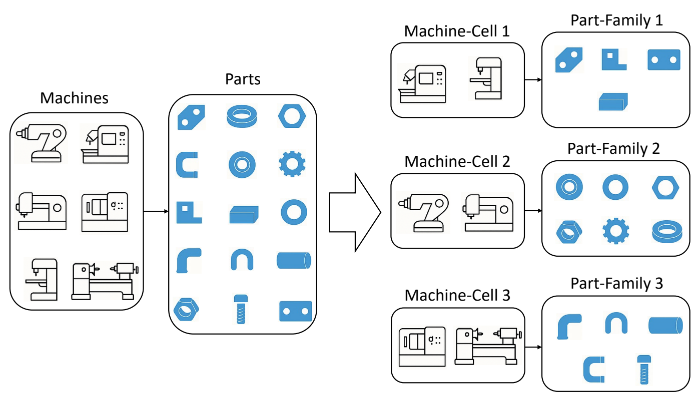


Fig. 1: Conceptual illustration of machine-cell and part-family formation in cellular manufacturing.

In the literature, there are three procedures of machine-cell and part-family formation: forming machine cells first and deducing part families, forming part families first and deducing machine cells, or forming machine cells and part families simultaneously. Over the past three decades, many methods have been developed for machine-cell and part-family formation, and they are mainly divided into two classes: clustering-based and optimization-based methods. Clustering-based methods include the rank-order clustering algorithm [4], the linear assignment clustering algorithm [6], [7], and the hierarchical clustering algorithm [16]. The clustering problem is known to be NP-complete. Optimization-based methods are subdivided into exact methods, heuristic methods, meta-heuristic methods, and neural network-based methods. Exact methods include the branch and cut algorithm [9], the primal and dual simplex algorithm [11], and the branch-and-bound algorithm [14]. Heuristic methods include the fuzzy goal programming algorithm [8], the heuristic part assignment algorithm [12], the multi-choice goal programming algorithm [13], and the p-median model-based algorithm [15]. Evolutionary methods include the ant colony optimization algorithm [10], the genetic algorithm [17], and the bacterial foraging algorithm [18]. Neural network methods include the ART1 [19]–[21]. In recent decades, artificial intelligence became a core enabler of intelligent manufacturing systems [22].

In his seminal papers [23], [24], John Hopfield foresaw that recurrent neural networks can collectively serve as powerful computational models. Specifically, the Hopfield networks are developed for linear programming and combinatorial optimization [24], [25]. Ever since then, a variety of neurodynamic optimization models have been developed for solving numerous optimization problems [26]–[31]. Despite the progress, it is acknowledged that an individual neurodynamic model faces challenges in effectively addressing combinatorial optimization problems because gradient-driven neurodynamic models are prone to be trapped in local minima. In recent years, the collaborative neurodynamic optimization (CNO) approach has emerged as a hybrid intelligence framework integrating neurodynamic optimization with evolutionary optimization methods to address diverse and intricate challenges in optimization. As demonstrated in [32], [33], CNO approaches are almost surely convergent to the global optimal solutions of optimization problems. CNO-driven computationally intelligent problem solvers appear in many applications, including nonnegative matrix factorization [34], Boolean matrix factorization [35], bicriteria sparse nonnegative matrix factorization [36], binary matrix factorization [37], etc. In addition, several CNO approaches are developed for solving general quadratic unconstrained binary optimization (QUBO) problems [38] as well as specific QUBO problems such as vehicle-task assignment [39], hash-bit selection [40], and capacitated clustering [41].

In this paper, we propose a CNO-driven binary matrix factorization approach to machine-cell and part-family formation. We first formulate a constrained binary matrix factorization problem with a quartic pseudo-Boolean objective function for machine-cell and part-family formation. We then prove that the quartic pseudo-Boolean function can be equivalently reduced to a quadratic one, and it is an upper bound of the number of exceptional elements in the given machine-part incident matrix. Next, we reformulate the problem as a matrix-valued QUBO problem via the penalization of constraint violation. We develop a CNO-driven algorithm based on multiple Boltzmann machines with repeated state reinitialization to solve the reformulated QUBO problem. The novelties and contributions of this work are outlined as follows:

i. The proposed constrained binary matrix factorization approach enables to form of machine cells and part families simultaneously.
ii. The quartic function of factorization errors is theoretically proven to be equivalent to a quadratic one as an upper bound of the number of exceptional elements in the machine-part incident matrix.
iii. The proposed CNO-driven algorithm for solving the reformulated problem with the reformulated quadratic objective function is experimentally demonstrated to perform statistically better than a CNO-driven algorithm for binary matrix factorization with the quartic objective function and the best-known results hitherto in terms of three performance criteria.

The remaining paper is structured as follows. Section II introduces essential preliminaries about a problem statement, performance criteria, and neurodynamic optimization. Section III discusses the problem formulation and reformulation. Section IV describes the proposed neurodynamics-driven algorithm. Section V elaborates on the experimental results in eighteen instances. Finally, Section VI provides the concluding remarks.

II. Preliminaries

A. Solution Representation

A machine-part incidence matrix $V \in \{0,1\}^{n\times m}$ encodes the relationships between machines and parts in a manufacturing system, where n is the number of machines, m is the number of parts, and $v_{ij} = 1$ if part j needs to be processed by machine i, and $v_{ij} = 0$, otherwise.

Let x_{ik} denote a binary decision variable to encode the assignment status of machine i to cell k, with $x_{ik} = 1$ if being assigned and $x_{ik} = 0$ otherwise. Similarly, let y_{kj} denote a binary decision variable to encode the assignment status of part j to family k, with $y_{kj} = 1$ if being assigned and $y_{kj} = 0$ otherwise.

Let $\mathcal{C}_k$ denote the index set of machine cell k, $\mathcal{F}_k$ denote the index set of part family k, and r denote the given number of machine cells or part families. Based on the encoding scheme above, machine cells and part family can be encoded using two indicator matrices X and Y as follows:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1r} \\ x_{21} & x_{22} & \cdots & x_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nr} \end{bmatrix}, \quad Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{r1} & y_{r2} & \cdots & y_{rm} \end{bmatrix},$$

where

$$x_{ij} = \begin{cases} 1 & \text{if machine } i \in \mathcal{C}_j, \\ 0 & \text{otherwise.} \end{cases} \qquad y_{ij} = \begin{cases} 1 & \text{if part } j \in \mathcal{F}_i, \\ 0 & \text{otherwise.} \end{cases}$$

Machine cells and the part families can be decoded from X and Y as follows:

$$\mathcal{C}_k = \{i | x_{ik} = 1,\ i = 1, ..., n\}, \quad k = 1, ..., r, \tag{1}$$

$$\mathcal{F}_k = \{j | y_{kj} = 1,\ j = 1, ..., m\}, \quad k = 1, ..., r, \tag{2}$$

B. Performance Criteria

The performance evaluation for machine-cell and part-family formation is commonly based on several criteria. For example, the number of exceptional elements (EE) refers to the number of parts in part families that need to be processed by machines outside machine cells associated with corresponding part families. In a permutated machine-part incidence matrix, it is the total number of 1's outside of blocked submatrices. EE is defined as follows:

$$\text{EE} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r} v_{ij}|x_{ik} - y_{kj}|, \tag{3}$$

where v_{ij} is the element at the i-th row and the j-th column of a given machine-part incidence matrix V.

The percentage of exceptional elements (PE) quantifies the ratio of exceptional elements to unity elements within V [4]:

$$\text{PE} = \frac{\text{EE}}{\text{UE}}, \tag{4}$$

where UE denotes the total count of ones in V, i.e., UE=$\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}$.

Bond energy (BE) is known as the measure of effectiveness for assessing the compactness of a permuted matrix $\tilde{V}$ [42]:

$$\text{BE} = \sum_{i=1}^{n-1}\sum_{j=1}^{m} \tilde{v}_{ij}\tilde{v}_{i+1,j} + \sum_{i=1}^{n}\sum_{j=1}^{m-1} \tilde{v}_{ij}\tilde{v}_{i,j+1}. \tag{5}$$

Machine utilization (MU) characterizes the frequency of visits to machines within cells [43]:

$$\text{MU} = \frac{\text{UE} - \text{EE}}{\sum_{k=1}^{r} h_k p_k}, \tag{6}$$

where h_k and p_k represent the number of machines in the k-th cell and the number of parts in the k-th family, respectively.

Grouping efficiency (GE) is defined as grouping efficiency [43]:

$$\text{GE} = w\frac{\text{UE} - \text{EE}}{\sum_{k=1}^{r} h_k p_k} + (1-w)\left(1 - \frac{\text{EE}}{nm - \sum_{k=1}^{r} h_k p_k}\right), \tag{7}$$

where $w \in [0, 1]$ is a weighting parameter. Normally $w = 0.5$. Note that GE=MU if $w = 1$.

C. Neurodynamic Optimization

1) Boltzmann Machine: The Boltzmann Machine (BM) is a type of stochastic neural network where each state x_i is updated based on an acceptance probability as follows [44]:

$$u(t) = Wx(t) - \theta, \tag{8}$$

$$P(x_i(t) = 1) = \frac{1}{1 + \exp(-u_i(t)/T(t))}, \tag{9}$$

where $u \in \Re^n$ denotes the net-input vector, $x \in \Re^n$ denotes the state vector, $W \in \Re^{n\times n}$ denotes the connection weight matrix, $\theta \in \Re^n$ denotes the threshold vector, and $T(t)$ denotes a positive temperature parameter at t-th iteration, updated according to $T = T_0\eta^t$, where T_0 denotes an initial temperature and $\eta \in (0, 1)$ is a cooling factor.

The BM is shown to be convergent to at least a local minimum of the following QUBO problem [44]:

$$\min \quad -\frac{1}{2}x^T Wx + \theta^T x, \qquad \text{s.t. } x \in \{0, 1\}^n. \tag{10}$$

A Boltzmann Machine with a momentum term (BMm) is expressed in [41] as follows:

$$u(t+1) = u(t) + Wx(t) - \theta, \tag{11a}$$

$$x_i(t) = \begin{cases} 1, & \text{if } \frac{1}{(1+\exp(-u_i(t)/T))} > \text{rand}, \\ 0, & \text{otherwise.} \end{cases} \tag{11b}$$

With the addition of the momentum term $u(t)$ in the BM dynamic equation, the BMm in (11) takes its historical effect into account and enriches its dynamic behaviors. It is shown that all neuronal states in the BMm in (11) can be activated synchronously and are convergent to local or near optima [41].

2) Collaborative Neurodynamic Optimization: In analogy with scattered searches in swarm intelligence, a CNO approach utilizes a population of individual neurodynamic optimization models to probe local optima. Additionally, it integrates a meta-heuristic rule, such as particle swarm optimization, to update initial neuronal states for the escape from local minima and the exploration of global optima. A mutation operator may be used to maintain a certain level of the diversity of initial neuronal states to prevent premature convergence.

Existing collaborative neurodynamic optimization (CNO) approaches utilize various neurodynamic models, including projection neural networks (e.g., [45]–[47]), discrete Hopfield networks [37]–[40], and Boltzmann machines [35], [38]. Almost all of the CNO algorithms [35], [37]–[40] use a particle swarm optimization rule in [48] as follows:

$$\psi_i(t) = c_0\psi_i(t-1) + c_1r_1(p_i^* - p_i(t-1)) + c_2r_2(p^* - p_i(t-1)), \tag{12a}$$

$$\text{if } (r_3 < S(\psi_i(t))), \text{then } p_i(t) = 1, \text{else } p_i(t) = 0, \tag{12b}$$

where p_i denotes the present position of the i-th particle, ψ_i denotes the velocity determining the searching direction, p_i^* denotes the present best solution of the i-th particle, p^* denotes the present best solution of a solution set, c_0 is an inertia weight, c_1 is a cognitive learning factor, c_2 is a social learning factor, and $r_1, r_2 \in [0, 1]$ are random constants, and $S(\cdot)$ is a sigmoid limiting transformation.

In a CNO approach, the diversity of initial states is essential for effective search, often enhanced by mutation operations to mitigate premature convergence. The diversity of initial states is quantified as follows:

$$\delta(p) = \frac{1}{Nn}\sum_{i=1}^{N} \|p^{(i)} - p^*\|_2, \tag{13}$$

where N is the population size (i.e., the total number of neurodynamic models), n is the dimension of a solution, $p^{(i)}$ is the initial states of the i-th neurodynamic model, and p^* is the present best solution among the entire population.

Bit-flip mutation, a commonly used mutation operator for combinatorial optimization [49], is expressed as:

$$p_j = \begin{cases} \neg p_j & \text{if } \kappa \le P_m, \\ p_j & \text{otherwise,} \end{cases} \tag{14}$$

where $\neg p_j$ denotes the negation of p_j, $\kappa \in [0, 1]$ is a random number, and P_m is a preset mutation probability.

CNO approaches based on BMs are developed for combinatorial optimization, such as capacitated clustering [41] and quadratic unconstrained binary optimization [38].

III. Problem Formulation and Reformulation

A. Problem Formulation

Let's consider the following ideal assignments without exceptional elements:

- If $v_{ij} = 1$, then for $k \in \{1, \ldots, r\}$, assign the i-th machine in $\mathcal{C}_k$ and assign the j-th part in $\mathcal{F}_k$ (i.e., $x_{ik} = y_{kj} = 1$, and $x_{il} = y_{lj} = 0$, $\forall l \neq k$). As a result, $\sum_{k=1}^{r} x_{ik} y_{kj} = 1$.
- If $v_{ij} = 0$, then $\forall k$, either machine i is not assigned in $\mathcal{C}_k$ or part j is not assigned in $\mathcal{F}_k$ (i.e., $\nexists k$ such that $i \in \mathcal{C}_k$ and $j \in \mathcal{F}_k$). As a result, $\sum_{k=1}^{r} x_{ik} y_{kj} = 0$.

Combining both cases yields

$$v_{ij} = \sum_{k=1}^{r} x_{ik} y_{kj}, \quad i = 1, ..., n, \quad j = 1, ..., m. \tag{15}$$

The equality in (15) does not hold, in the presence of exceptional element(s). It does not hold either if a part in a family does not need to be processed on every machine in its corresponding machine cell. In such scenarios, a norm of $XY - V$ may be used to measure the errors of cell-family imperfect match.

Based on the discussions above, a constrained binary matrix factorization problem is formulated for machine-cell and part-family formation as follows:

$$\min_{X,Y} \quad ||XY - V||_F^2 \tag{16a}$$

$$\text{s.t.} \quad Xe_r = e_n, \tag{16b}$$

$$Y^T e_r = e_m, \tag{16c}$$

$$X \in \{0,1\}^{n \times r}, \; Y \in \{0,1\}^{r \times m}, \tag{16d}$$

where $||\cdot||_F$ is the Frobenius norm, and $e_n = [1, 1, ..., 1]^T \in \mathfrak{R}^n$ is an n-vector of ones. Constraint (16b) requires the sum of elements in each row of X to be one to ensure each machine being assigned to one and only one cell. Constraint (16c) requires the sum of elements in each column of Y to be one to ensure each part being assigned to one and only one family.

Consider an incidence matrix V without exceptional elements to form four machine cells and part families (i.e., $r = 4$) in [50]:

$$V = \begin{array}{r|cccccccccccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ \hline 1 & 1 & & & 1 & & & 1 \\ 2 & & 1 & 1 & & 1 & & & 1 & & 1 \\ 3 & & 1 & 1 & & 1 & & & 1 & & 1 \\ 4 & 1 & & & 1 & & & 1 \\ 5 & & & & & & & & & & & & & 1 & 1 & 1 & & 1 & 1 & & 1 \\ 6 & 1 & & & 1 & & & 1 \\ 7 & & & & & & 1 & & & 1 & & 1 & 1 & & & & 1 & & & 1 \\ 8 & & & & & & 1 & & & 1 & & 1 & 1 & & & & 1 & & & 1 \\ 9 & & & & & & & & & & & & & 1 & 1 & 1 & & 1 & 1 & & 1 \\ 10 & & & & & & 1 & & & 1 & & 1 & 1 & & & & 1 & & & 1 \end{array},$$

where elements of zeros are left blank, $n = 10$, and $m = 20$.

Factorized X and Y are given as follows:

$$X = \begin{array}{r|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 \\ 2 & & 1 \\ 3 & & 1 \\ 4 & 1 \\ 5 & & & 1 \\ 6 & 1 \\ 7 & & & & 1 \\ 8 & & & & 1 \\ 9 & & & 1 \\ 10 & & & & 1 \end{array}, \tag{17}$$

$$Y = \begin{array}{r|cccccccccccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ \hline 1 & 1 & & & 1 & & & 1 \\ 2 & & 1 & 1 & & 1 & & & 1 & & 1 \\ 3 & & & & & & & & & & & & & 1 & 1 & 1 & & 1 & 1 & & 1 \\ 4 & & & & & & 1 & & & 1 & & 1 & 1 & & & & 1 & & & 1 \end{array}, \tag{18}$$

The index sets for machine cells are decoded according to (1) based on (17); i.e., $\mathcal{C}_1 = \{1, 4, 6\}$, $\mathcal{C}_2 = \{2, 3\}$, $\mathcal{C}_3 = \{5, 9\}$, and $\mathcal{C}_4 = \{7, 8, 10\}$. Similarly, the index sets for part families are decoded according to (2) based on (18); i.e., $\mathcal{F}_1 = \{1, 4, 7\}$, $\mathcal{F}_2 = \{2, 3, 5, 8, 10\}$, $\mathcal{F}_3 = \{13, 14, 15, 17, 18, 20\}$, and $\mathcal{F}_4 = \{6, 9, 11, 12, 16, 19\}$. Based on $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{C}_4$, $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$, and $\mathcal{F}_4$, V is permuted to become the following ideal incidence matrix $\hat{V}$:

$$\hat{V} = \begin{array}{r|cccccccccccccccccccc} & 1 & 4 & 7 & 2 & 3 & 5 & 8 & 10 & 13 & 14 & 15 & 17 & 18 & 20 & 6 & 9 & 11 & 12 & 16 & 19 \\ \hline 1 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 \\ 6 & 1 & 1 & 1 \\ 2 & & & & 1 & 1 & 1 & 1 & 1 \\ 3 & & & & 1 & 1 & 1 & 1 & 1 \\ 5 & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 \\ 9 & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & & & & & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 \\ 8 & & & & & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 \\ 10 & & & & & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

To visualize the resulting formulation, Fig. 2 shows V and $\hat{V}$ with four cells and no exceptional element in the ideal case.

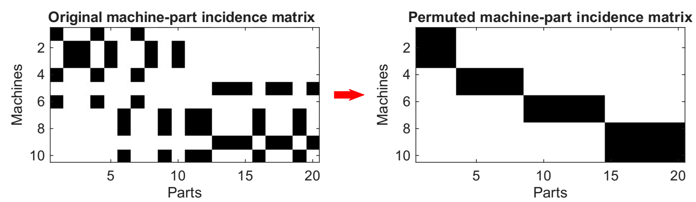


Fig. 2: The original machine-part incidence matrix and permuted machine-part incidence matrix, where dots represent 1 elements and 0 elements are left blank.

Consider a nonideal case with 40 machines, 100 parts, and ten groups (i.e., $n = 40$, $m = 100$, and $r = 10$) where exceptional elements are inevitable [51]. Fig. 3 illustrates the machine-part incidence matrix factorization and permutation processes.

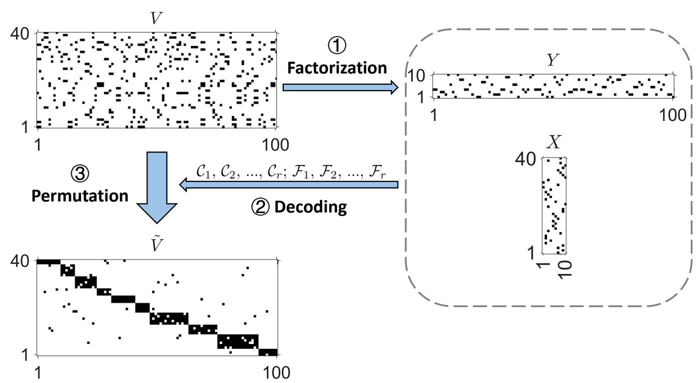


Fig. 3: The original machine-part incidence matrix V is factorized into two indicator matrices X and Y, and then V is permuted to become the block-diagonalized incidence matrix $\tilde{V}$ based on the decoded information from X and Y.

B. Problem Reformulation

Let $f(X, Y) = ||XY - V||_F^2$. It is rewritten in its element form:

$$f(X, Y)$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{m}\left\{\sum_{k=1}^{r}\sum_{l=1}^{r}(x_{ik}x_{il}y_{kj}y_{lj})-2\sum_{k=1}^{r}v_{ij}x_{ik}y_{kj}+v_{ij}^2\right\}. \tag{19}$$

Lemma 1. *If $x_{ik}\in\{0,1\}$, $y_{kj}\in\{0,1\}$, $\sum_{k=1}^{r}x_{ik}=1$, and $\sum_{k=1}^{r}y_{kj}=1$, then*

$$\sum_{k=1}^{r}\sum_{l=1}^{r}x_{ik}x_{il}y_{kj}y_{lj}=\sum_{k=1}^{r}x_{ik}y_{kj}, \tag{20}$$

$$|x_{ik}-y_{kj}|=x_{ik}(1-y_{kj})+y_{kj}(1-x_{ik}). \tag{21}$$

Proof. By utilizing the distributive property of addition and recombining the terms, Eq. (20) is converted to a square of a single summation as follows:

$$\sum_{k=1}^{r}\sum_{l=1}^{r}x_{ik}x_{il}y_{kj}y_{lj}=\sum_{k=1}^{r}x_{ik}y_{kj}\sum_{l=1}^{r}x_{il}y_{lj}=\left(\sum_{k=1}^{r}x_{ik}y_{kj}\right)^2.$$

For x_{ik} and y_{kj} satisfying $\sum_{k=1}^{r}x_{ik}=1$ and $\sum_{k=1}^{r}y_{kj}=1$, $(\sum_{k=1}^{r}x_{ik}y_{kj})^2=1$ and $\sum_{k=1}^{r}x_{ik}y_{kj}=1$ if and only if $\exists k, x_{ik}=y_{kj}=1$, otherwise $(\sum_{k=1}^{r}x_{ik}y_{kj})^2=0$ and $\sum_{k=1}^{r}x_{ik}y_{kj}=0$. Then, $(\sum_{k=1}^{r}x_{ik}y_{kj})^2=\sum_{k=1}^{r}x_{ik}y_{kj}$. As a result,

$$\sum_{k=1}^{r}\sum_{l=1}^{r}x_{ik}x_{il}y_{kj}y_{lj}=\sum_{k=1}^{r}x_{ik}y_{kj},$$

To prove equation (21), consider the two cases for the two binary variables x_{ik} and y_{kj}:

- If $x_{ik}=y_{kj}$, then $|x_{ik}-y_{kj}|=0$. Both $x_{ik}(1-y_{kj})$ and $y_{kj}(1-x_{ik})$ are also 0.
- If $x_{ik}\neq y_{kj}$, then $|x_{ik}-y_{kj}|=1$.
  - If $x_{ik}=1$ and $y_{kj}=0$, then $x_{ik}(1-y_{kj})=1$ and $y_{kj}(1-x_{ik})=0$. As a result, $x_{ik}(1-y_{kj})+y_{kj}(1-x_{ik})=1$.
  - If $x_{ik}=0$ and $y_{kj}=1$, then $x_{ik}(1-y_{kj})=0$ and $y_{kj}(1-x_{ik})=1$. As a result, $x_{ik}(1-y_{kj})+y_{kj}(1-x_{ik})=1$.

As a result, in both cases, $x_{ik}(1-y_{kj})+y_{kj}(1-x_{ik})$ is equal to $|x_{ik}-y_{kj}|$. □

Theorem 1. *If the constraints in* (16) *hold, then the quartic pseudo-Boolean function in $f(X,Y)$ can be equivalently reduced to a quadratic one.*

Proof. By substituting (20) into (19) and combining like terms, $f(X,Y)$ becomes

$$\phi(X,Y)=\sum_{i=1}^{n}\sum_{j=1}^{m}\left\{(1-2v_{ij})\sum_{k=1}^{r}x_{ik}y_{kj}+v_{ij}^2\right\}. \tag{22}$$

□

Remark 1. The theorem above reveals that the quartic objective function $f(X,Y)$ can be equivalently quadratized by leveraging the constraints without adding any auxiliary variables or extra constraints, in contrast to most existing methods [52]–[57]

Theorem 2. *$\phi(X,Y)$ is an upper bound of the number of exceptional elements (EE).*

Proof. Substituting (21) into (3), EE in (3) becomes:

$$\begin{aligned}&\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r}v_{ij}|x_{ik}-y_{kj}|\\&=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r}\{v_{ij}x_{ik}(1-y_{kj})+v_{ij}y_{kj}(1-x_{ik})\},\\&=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r}\{v_{ij}x_{ik}-2v_{ij}x_{ik}y_{kj}+v_{ij}y_{kj}\}.\end{aligned}$$

Then, the difference between $\phi(X,Y)$ and EE is expressed as:

$$\begin{aligned}&\phi(X,Y)-\mathrm{EE}\\&=\sum_{i=1}^{n}\sum_{j=1}^{m}\left\{(1-2v_{ij})\sum_{k=1}^{r}x_{ik}y_{kj}+v_{ij}^2\right\}\\&\quad-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r}\{v_{ij}x_{ik}-2v_{ij}x_{ik}y_{kj}+v_{ij}y_{kj}\},\\&=\sum_{i=1}^{n}\sum_{j=1}^{m}\left\{\sum_{k=1}^{r}\left[x_{ik}y_{kj}-v_{ij}x_{ik}y_{kj}-\frac{1}{2}v_{ij}x_{ik}-\frac{1}{2}v_{ij}y_{kj}\right]+v_{ij}\right\}.\end{aligned}$$

For simplicity, for each pair of indices (i,j), let

$$q_{ij}=\sum_{k=1}^{r}\left\{x_{ik}y_{kj}-v_{ij}x_{ik}y_{kj}-\frac{1}{2}v_{ij}x_{ik}-\frac{1}{2}v_{ij}y_{kj}\right\}+v_{ij}.$$

In view of the constraints (i.e., $\sum_{k=1}^{r}x_{ik}=1$, $\sum_{k=1}^{r}y_{kj}=1$, $x_{ik}\in\{0,1\}$ and $y_{kj}\in\{0,1\}$), it follows that:

$$q_{ij}=\begin{cases}1-v_{ij} & \text{if } \exists k, x_{ik}=y_{kj}=1,\\ 0 & \text{if otherwise.}\end{cases}$$

Thus, the difference between $\phi(X,Y)$ and EE is nonnegative:

$$\phi(X,Y)-\mathrm{EE}=\sum_{i=1}^{n}\sum_{j=1}^{m}q_{ij}\geq 0.$$

As a result, $\phi(X,Y)$ is an upper bound of EE. □

Two quadratic penalty functions are defined as follows for handling the constraints in (16b) and (16c):

$$p_a(X)=\frac{1}{2}\|Xe_r-e_n\|_2^2=\frac{1}{2}\sum_{i=1}^{n}(\sum_{j=1}^{r}x_{ij}-1)^2,$$

$$p_b(Y)=\frac{1}{2}\|Y^Te_r-e_m\|_2^2=\frac{1}{2}\sum_{j=1}^{m}(\sum_{k=1}^{r}y_{kj}-1)^2,$$

A penalty function is formulated based on $p_a(X)$ and $p_b(Y)$ as follows:

$$\begin{aligned}p(X,Y)&=p_a(X)+p_b(Y)\\&=\frac{1}{2}\sum_{i=1}^{n}(\sum_{j=1}^{r}x_{ij}-1)^2+\frac{1}{2}\sum_{j=1}^{m}(\sum_{k=1}^{r}y_{kj}-1)^2.\end{aligned} \tag{23}$$

By combining the penalty function (23) with objective function (22), a penalized objective function is defined: $\phi_\rho(X,Y) = \phi(X,Y) + \rho p(X,Y)$, where ρ is a positive penalty parameter. As such, problem (16) is reformulated to a QUBO problem with the penalized objective function:

$$\min \ \phi_\rho(X,Y) \tag{24}$$

$$\text{s.t. } X \in \{0,1\}^{n\times r}, \ Y \in \{0,1\}^{r\times m}. \tag{25}$$

It is known that problems (16) and (24) are equivalent in terms of their optimal solutions, provided that the penalty parameter ρ is set with a sufficiently large value [58].

IV. Algorithm Description

In this section, we describe the CNO-based algorithm based on BMm (11) to solve QUBO problem (24). As the dynamic equation of BMm (11) is composed of the negative gradient of a given objective function, we drive the partial derivatives of $\phi_\rho(X,Y)$ in (24) with respect to x_{ij} and y_{jk} as follows: for $i = 1,2,\dots,n; k = 1,2,\dots,r; j = 1,2,\dots,m:$

$$\frac{\partial \phi_\rho(X,Y)}{\partial x_{ik}} = \sum_{j=1}^{m}(1-2v_{ij})y_{kj} + \rho\Big(\sum_{q=1}^{r} x_{iq} - 1\Big), \tag{26}$$

$$\frac{\partial \phi_\rho(X,Y)}{\partial y_{kj}} = \sum_{i=1}^{n}(1-2v_{ij})x_{ik} + \rho\Big(\sum_{q=1}^{r} y_{qk} - 1\Big). \tag{27}$$

Based on (26) and (27), the neurodynamic equation and activation functions of BMms for updating X and Y in (24) are customized, respectively, as follows: for $i = 1,2,\dots,n; k = 1,2,\dots,r:$

$$u_{ik}^{X}(t) = \sum_{j=1}^{m}(1-2v_{ij})y_{kj}(t-1) + \rho\left(\sum_{q\neq k} x_{iq}(t-1) - 1/2\right), \tag{28a}$$

$$x_{ik}(t) = \begin{cases} 1, & \text{if } 1/\left(1+\exp(-u_{ik}^{X}(t)/T)\right) > \text{rand}, \\ 0, & \text{otherwise.} \end{cases} \tag{28b}$$

For $k = 1,2,\dots,r; j = 1,2,\dots,m:$

$$u_{kj}^{Y}(t) = \sum_{i=1}^{n}(1-2)v_{ij}x_{ik}(t-1) + \rho\left(\sum_{q\neq k} y_{qj}(t-1) - 1/2\right), \tag{29a}$$

$$y_{kj}(t) = \begin{cases} 1, & \text{if } 1/\left(1+\exp(-u_{kj}^{Y}(t)/T)\right) > \text{rand}, \\ 0, & \text{otherwise.} \end{cases} \tag{29b}$$

Algorithm 1 details the neurodynamics-driven constrained binary matrix factorization approach to machine-cell and part-family formation (CNO-MP). A population of BMms is utilized for scattered searches in Steps 3 - 5. The individual best solutions $X^{(i)}$ and $Y^{(i)}$ are identified in Steps 6 - 9. The best solution among the BMms is determined in Steps 11 - 18. The initial states of BMms are re-positioned using the particle swarm optimization update rule in Steps 19 - 21. The diversity is measured in Step 22, and if it falls below the threshold Δ, a bit-flip mutation is executed in Steps 23 - 25. The information on machine cells and part families is decoded in Step 27. The code of CNO-MP is available in Github[1].

Algorithm 1: CNO-MP algorithm

Input: N, $X^{(i)}(0) \in \{0,1\}^{n\times r}$ and $Y^{(i)}(0) \in \{0,1\}^{r\times m}$, $\psi_X^{(i)} \in [-1,1]^{n\times r}$ and $\psi_Y^{(i)} \in [-1,1]^{r\times m}$, for $i = 1,...,N$, T_0, η, c_0, c_1, c_2, Δ, termination criterion M.

Output: machine cells and part families.

1 **while** $l \le M$ **do**
2   **for** $i = 1$ **to** N **do**
3   **repeat**
4   Update $X^{(i)}(t)$ and $Y^{(i)}(t)$ using BMm according to (28) and (29), respectively;
5   **until** *convergence*;
6   **if** $\phi_\rho(\bar{X}^{(i)}, \bar{Y}^{(i)}) < \phi_\rho(X^{(i)}, Y^{(i)})$ **then**
7       $X^{(i)} \leftarrow \bar{X}^{(i)}$;
8       $Y^{(i)} \leftarrow \bar{Y}^{(i)}$;
9   **end**
10   **end**
11   $i^* = \arg\min_i\{..., \phi_\rho(X^{(i)}, Y^{(i)}), ...\}$;
12   **if** $\phi_\rho(X^{(i^*)}, Y^{(i^*)}) < \phi_\rho(X^*, Y^*)$ **then**
13     $l \leftarrow 0$;
14     $X^* \leftarrow X^{(i^*)}$;
15     $Y^* \leftarrow Y^{(i^*)}$;
16   **else**
17     $l \leftarrow l + 1$;
18   **end**
19   **for** $i = 1$ **to** N **do**
20   Update velocity and initial neuronal states $X^{(i)}(0)$ and $Y^{(i)}(0)$ according to (12) ;
21   **end**
22   Calculate the diversity of the swarm δ in (13);
23   **if** $\delta < \Delta$ **then**
24   Perform the bit-flip mutation in (14);
25   **end**
26 **end**
27 The information on machine cells and part families is decoded according to (1) and (2);
28 **return** the decoded information on machine cells and part families.

V. Experimental Results

A. Experiment Setups

The experiments are based on eighteen datasets with their major parameters listed in Table I. The performance of CNO-MP is compared with the results using CNO-BMF [37] based on the quartic objective function $f(X,Y)$ in (19) and the best-known results from the references in Table I.

In this study, the hyper-parameters N (i.e., population size) and M (i.e., termination criteria) in Algorithm 1 are determined via 25-run Monte Carlo tests with random

[1] https://github.com/HongzongLI-CS/CNO-MP

TABLE I: The major information of the 18 benchmark incidence matrices and the hyper-parameter values used in the experiments.

#	$n \times m$	r	N	M	References
1	5 × 7	2	2	2	[5]
2	5 × 7	2	2	2	[59]
3	5 × 7	2	2	5	[59]
4	7 × 11	2	10	10	[60]
5	10 × 10	3	2	3	[61]
6	15 × 10	3	2	2	[62]
7	8 × 20	3	2	2	[43]
8	10 × 20	4	2	2	[50]
9	23 × 20	2	2	5	[63]
10	24 × 40	7	2	2	[64]
11	24 × 40	7	2	2	[64]
12	24 × 40	7	2	5	[64]
13	24 × 40	7	2	5	[64]
14	24 × 40	7	60	50	[64]
15	24 × 40	7	40	100	[64]
16	24 × 40	7	15	20	[64]
17	30 × 41	3	10	3	[65]
18	40 × 100	10	2	2	[51]

initialization on the eighteen datasets. Figs. 7 and 8 depict the box-plots of the results of the Monte Carlo tests using CNO-MP over 25 runs with varied initial states on the eighteen datasets. As shown in Figs. 7 and 8, the median is represented by a center bar within each box. The upper and lower quartiles ($q_n(0.75)$ and $q_n(0.25)$) are indicated by the top and bottom of each box, respectively. The whiskers depict the highest and lowest values observed in the tests. The values of M and N are set to values at which the deviation of the objective function value becomes zero. Table I records the hyper-parameters values used in the experiments.

In the experiments, the parameters of CNO-MP are set as follows. The mutation probability P_m in (14) is set to a sufficiently small value (i.e., 0.01), and the diversity threshold Δ is set to 0.004, same as the values used in many studies; e.g., [37], [66]. In the particle swarm optimization rule in (12), $c_0 = 1$, $c_1 = c_2 = 2$, as typically used values [67].

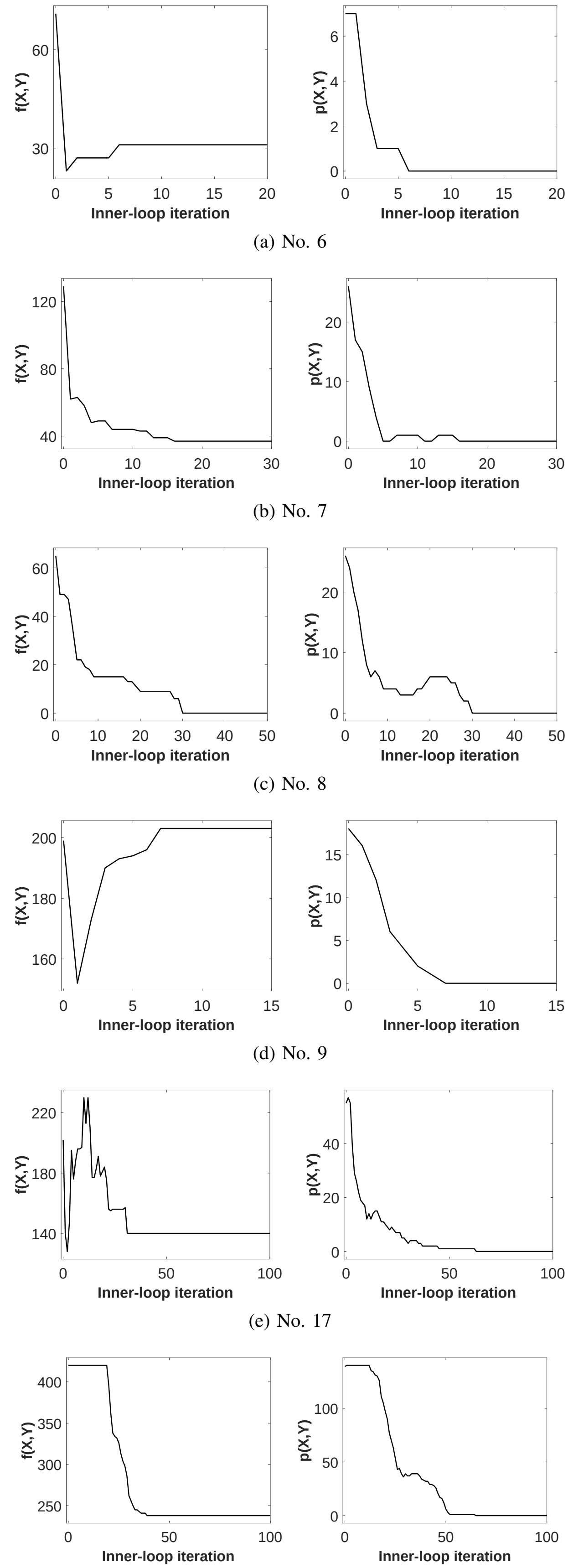
B. Neurodynamic Behaviors

Fig. 4 depicts six snapshots of the convergent behaviors of the objective function $f(X,Y)$ in (16a) and the penalty function $p(X,Y)$ in (23) in the inner loop of CNO-MP on the six datasets. As shown in Fig. 4, the objective and penalty functions reach stationary points within 70 iterations, and the values of the penalty function decrease to zero, indicating that BMms converge to feasible solutions with random initial neuronal states.

Fig. 5 depicts the convergent behaviors of $f(X,Y)$ resulting from CNO-MP on the six datasets, where the red envelopes depict the objective functions of group-best solutions, i.e., $f(X^*,Y^*)$. It shows that CNO-MP converges within 40 iterations.

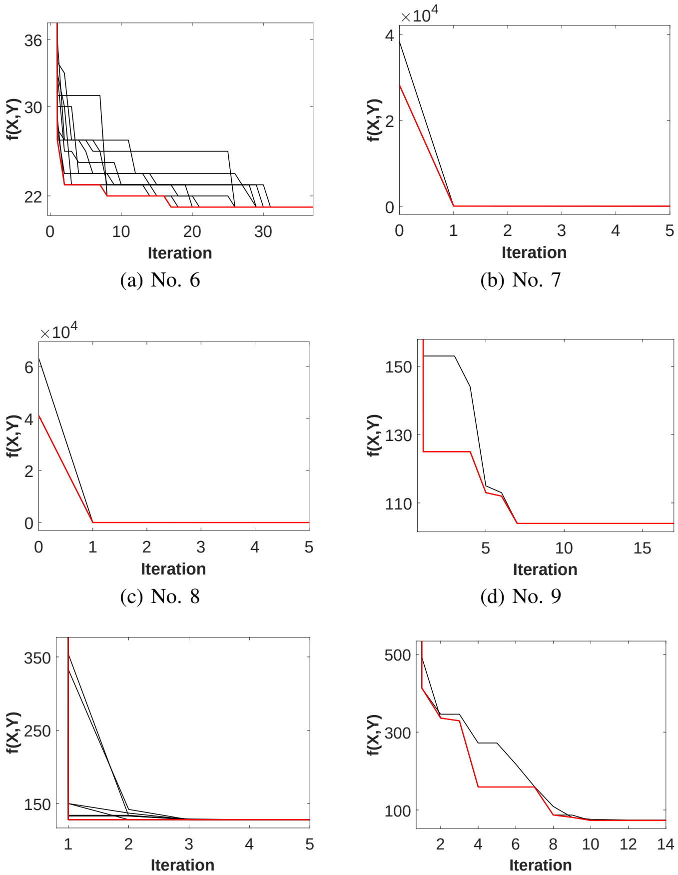
C. Performance Comparisons

Table II summarizes the Monte-Carlo study with the objective function values, four criteria values resulting from



(a) No. 6 (b) No. 7 (c) No. 8 (d) No. 9 (e) No. 17 (f) No. 18

Fig. 4: Snapshots of the objective function values of $f(X,Y)$ in (16a) and the penalty function values of $p(X,Y)$ in (23) in the inner loop of CNO-MP on the six datasets.



(a) No. 6 (b) No. 7 (c) No. 8 (d) No. 9 (e) No. 17 (f) No. 18

Fig. 5: The convergent behaviors of CNO-MP.

CNO-BMF with the quartic objective function $f(X,Y)$ in (19) and CNO-MP with the quadratic objective function $\phi(X,Y)$ in (22) over 50 runs with random initialization on the eighteen benchmark datasets, where the best results are boldfaced. As shown in Table II, CNO-MP with the quadratic objective function outperforms CNO-BMF with the quartic objective function in terms of the objective function values on all of the datasets and most of the criteria values. Fig. 6 depicts the average computational times of CNO-BMF and CNO-MP over 50 runs with random initialization on the 18 benchmark datasets. It shows that the computational times of CNO-MP are smaller than those of CNO-BMF across all 18 datasets, demonstrating the superior efficiency of CNO-MP. In addition, the computational times of CNO-MP are not proportionally larger on large-sized problems, indicating the high efficiency and scalability of CNO-MP.

Table III summarizes the Monte-Carlo study with the four criteria resulting from CNO-MP over 50 runs with random initialization with quadratic objective function on the eighteen datasets, where the best-known results are documented in the literature in Table I. The second and third columns of Table III show that the values of the objective function are always larger than or equal to the values of EE across the eighteen datasets, echoing the theoretical result in Theorem 2. As shown in Table III, the $f(X,Y)$ values and EE values resulting from CNO-MP on datasets #7 and #8 are equal, implying that PE values are minimal in view that $f(X,Y)$ is an upper bound of EE. As shown in the remaining eight columns of Table III, out of the total 72 performance indexes examined, 16 index values are better than, and 37 index values are equal to the index values of the best-known results reported in the literature.

Table IV tabulates the counts of best results achieved using CNO-MP and twelve baselines across the four metrics (PE, BE, MU, and GE) on the 18 datasets. It shows that CNO-MP achieves the best results with 53 best metric values in total, more than doubled the second-best method (ZODIAC in [51]) with 21 best values. Specifically, CNO-MP achieves the best-known results in the literature in terms of PE on 12 datasets out of 18 datasets (i.e., 66.7%), in terms of BE on eight datasets (i.e., 44.4%), in terms of MU on 17 datasets (i.e., 94.4%), and in terms of GE on 16 datasets (i.e., 88.9%).

VI. Concluding Remarks

This paper proposes a neurodynamics-driven constrained binary matrix factorization approach to machine-cell and part-family formation. By minimizing the Frobenius norm of factorization errors, the proposed approach can decode the information of machine cells and part families from factorized matrices. To facilitate the solution process, the formulated binary matrix factorization problem is equivalently reformulated to a quadratic unconstrained binary optimization problem via polynomial-degree reduction and constraint-violation penalization. To solve the reformulated problem, the proposed approach leverages the hill-climbing local search capability of Boltzmann machines for scattered searches and the global search capability of collaborative neurodynamic optimization to seek global optima. The experimental results demonstrate that the proposed approach statistically outperforms an existing neurodynamics-driven binary matrix factorization approach and the best-known results in the literature. Further investigations may aim to enhance the efficiency and scalability of the neurodynamics-driven constrained binary matrix factorization approach via machine learning, develop bi-level approaches to determine the number of machine cells and part families according to factorization or manufacturing performance metrics, and adapt the proposed approach for other applications such as supply chain management.

References

[1] S. Huang and Y. Yan, "Design of delayed reconfigurable manufacturing system based on part family grouping and machine selection," *International Journal of Production Research*, vol. 58, no. 14, pp. 4471–4488, 2020.

[2] R. YounesSinaki, A. Sadeghi, H. Mosadegh, N. Almasarwah, and G. Suer, "Cellular manufacturing design 1996–2021: A review and introduction to applications of industry 4.0," *International Journal of Production Research*, vol. 61, no. 16, pp. 5585–5636, 2023.

[3] F. K. Konstantinidis, N. Myrillas, K. A. Tsintotas, S. G. Mouroutsos, and A. Gasteratos, "A technology maturity assessment framework for industry 5.0 machine vision systems based on systematic literature review in automotive manufacturing," *International Journal of Production Research*, 2024, in press.

[4] J. R. King, "Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm," *International Journal of Production Research*, vol. 18, no. 2, pp. 213–232, 1980.

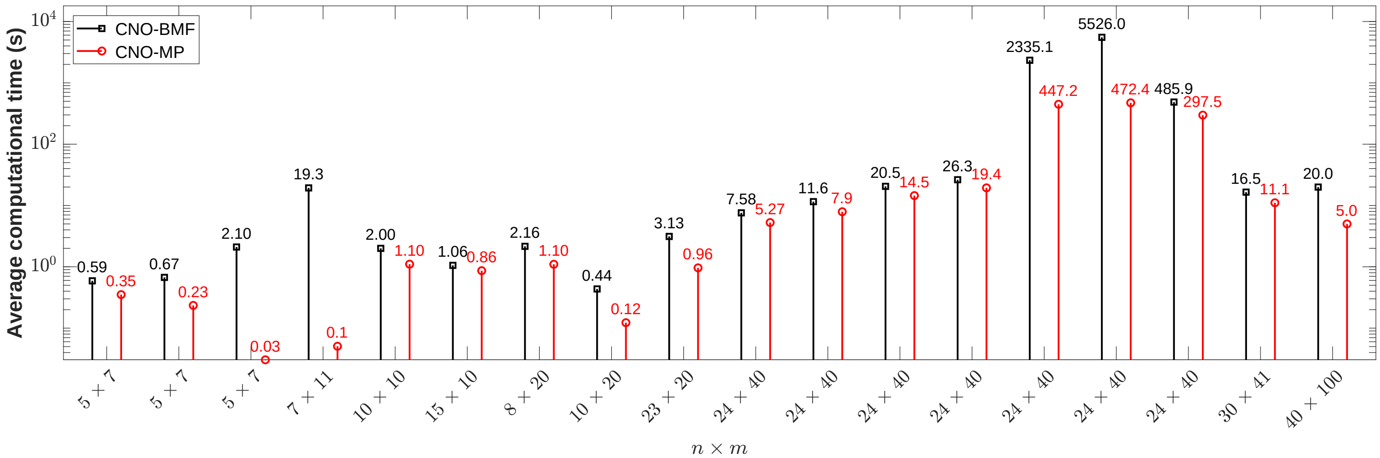


Fig. 6: The computational times of CNO-BMF and CNO-MP on the 18 benchmark datasets.

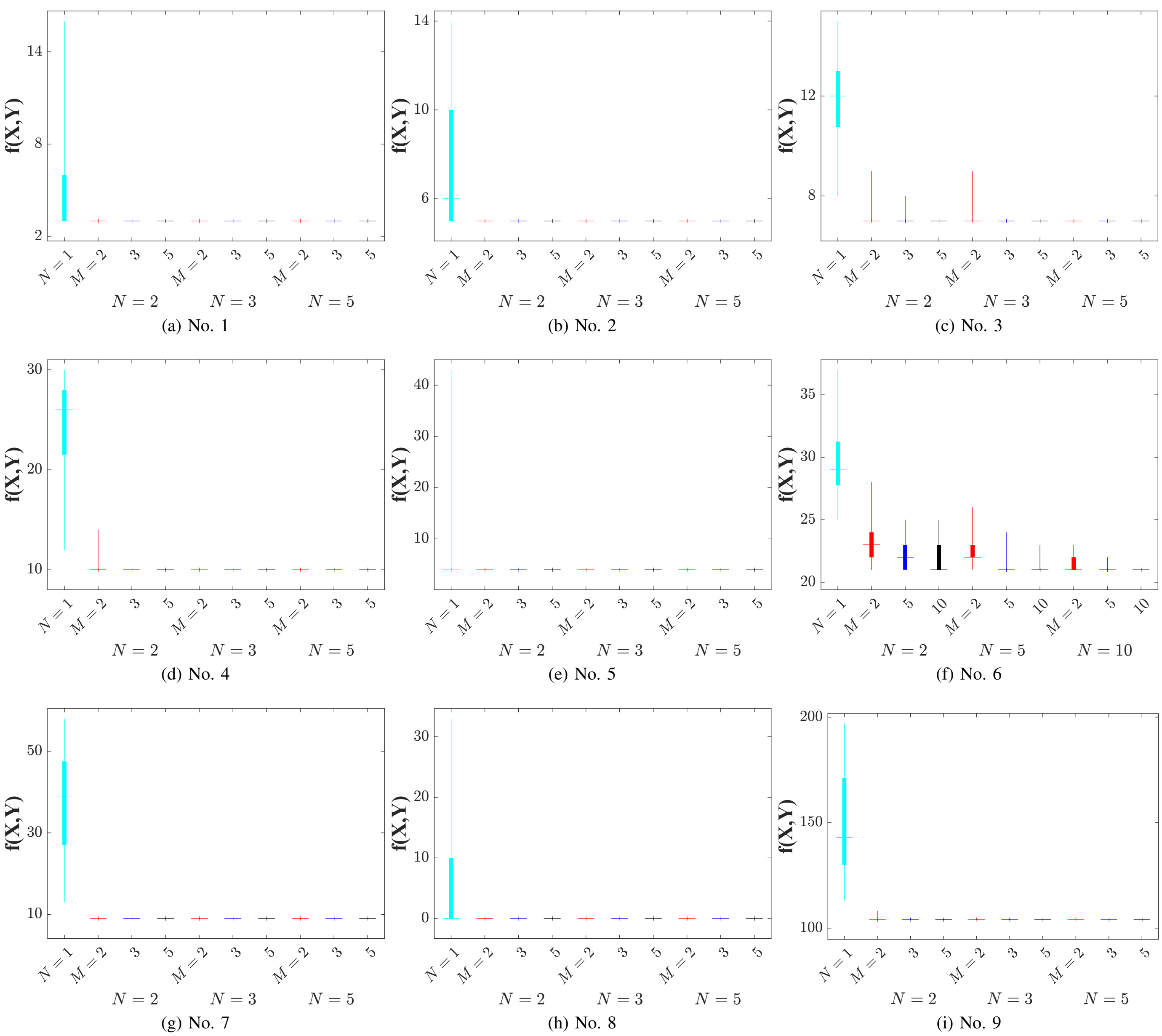
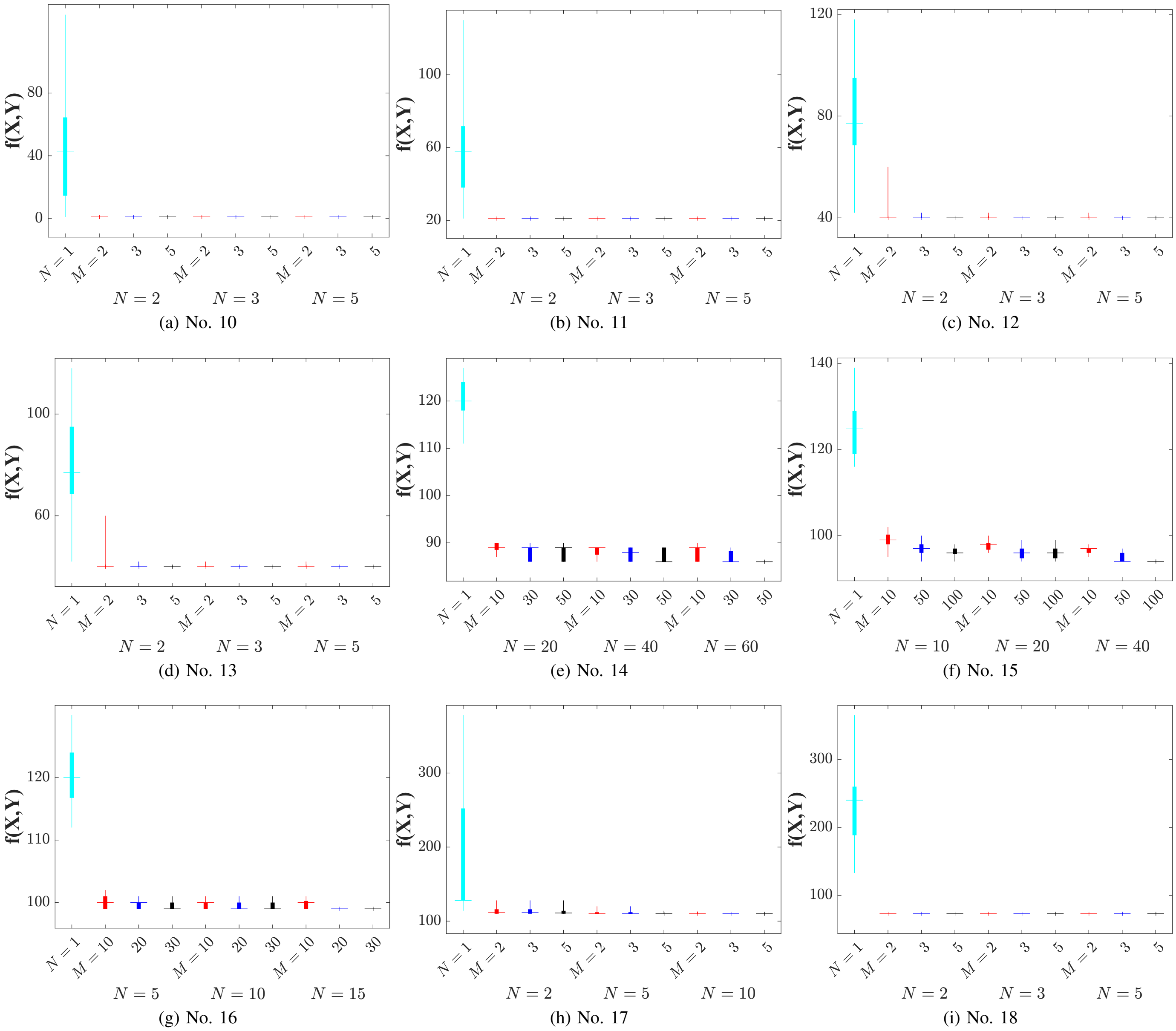


Fig. 7: The box-plots of the Monte Carlo test results on the nine datasets using CNO-MP with several values of N and M.



(a) No. 10 (b) No. 11 (c) No. 12

(d) No. 13 (e) No. 14 (f) No. 15

(g) No. 16 (h) No. 17 (i) No. 18

Fig. 8: The box-plots of the Monte Carlo test results on the nine datasets using CNO-MP with several values of N and M.

TABLE II: Monte-Carlo test results with average values of objective function value and four criteria achieved using CNO-BMF and CNO-MP over 50 runs with random initialization on the 18 datasets

#	$f(X,Y)\downarrow$		PE $\downarrow$		BE $\uparrow$		MU $\uparrow$		GE $\uparrow$	
	CNO-BMF	CNO-MP	CNO-BMF	CNO-MP	CNO-BMF	CNO-MP	CNO-BMF	CNO-MP	CNO-BMF	CNO-MP
1	**3.00 ± 0.00**	**3.00 ± 0.00**	**0.0000 ± 0.0000**	**0.0000 ± 0.0000**	**14.0000 ± 0.0000**	**14.0000 ± 0.0000**	**0.8235 ± 0.0000**	**0.8235 ± 0.0000**	**0.9118 ± 0.0000**	**0.9118 ± 0.0000**
2	5.28 ± 1.02	**5.00 ± 0.00**	0.1300 ± 0.0250	**0.1250 ± 0.0000**	14.7200 ± 1.4000	**15.0000 ± 0.0000**	0.8136 ± 0.0331	**0.8235 ± 0.0000**	0.8485 ± 0.0286	**0.8562 ± 0.0000**
3	7.76 ± 1.01	**7.00 ± 0.00**	0.2340 ± 0.0535	**0.2000 ± 0.0000**	**20.0400 ± 1.3687**	19.2000 ± 1.0000	0.8329 ± 0.0199	**0.8421 ± 0.0000**	0.7769 ± 0.0259	**0.7961 ± 0.0000**
4	12.20 ± 2.65	**10.00 ± 0.00**	0.0717 ± 0.1017	**0.0000 ± 0.0000**	21.2400 ± 2.5212	**23.0000 ± 0.0000**	0.6792 ± 0.0331	**0.7059 ± 0.0000**	0.8274 ± 0.0310	**0.8529 ± 0.0000**
5	**4.00 ± 0.00**	**4.00 ± 0.00**	**0.0000 ± 0.0000**	**0.0000 ± 0.0000**	**63.0000 ± 0.0000**	**63.0000 ± 0.0000**	**0.9200 ± 0.0000**	**0.9200 ± 0.0000**	**0.9600 ± 0.0000**	**0.9600 ± 0.0000**
6	21.04 ± 0.20	**21.00 ± 0.00**	**0.1304 ± 0.0000**	**0.1304 ± 0.0000**	**14.5600 ± 0.5066**	14.4800 ± 0.5099	0.5258 ± 0.0027	**0.5263 ± 0.0000**	0.7244 ± 0.0016	**0.7247 ± 0.0000**
7	**9.00 ± 0.00**	**9.00 ± 0.00**	**0.1475 ± 0.0000**	**0.1475 ± 0.0000**	78.5600 ± 0.7681	**78.6400 ± 0.7000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**0.9583 ± 0.0000**	**0.9583 ± 0.0000**
8	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.0000 ± 0.0000**	**0.0000 ± 0.0000**	**68.0000 ± 0.0000**	**68.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
9	105.92 ± 3.96	**104.00 ± 0.00**	**0.7368 ± 0.0739**	0.7464 ± 0.0452	64.3200 ± 1.2819	**65.2800 ± 1.4866**	0.5701 ± 0.0406	**0.5853 ± 0.0169**	0.6840 ± 0.0165	**0.6911 ± 0.0048**
10	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.0000 ± 0.0000**	**0.0000 ± 0.0000**	**194.0000 ± 0.0000**	**194.0000 ± 0.0000**	**0.9924 ± 0.0000**	**0.9924 ± 0.0000**	**0.9962 ± 0.0000**	**0.9962 ± 0.0000**
11	62.40 ± 17.78	**21.00 ± 0.00**	0.2646 ± 0.1006	**0.0769 ± 0.0000**	131.5600 ± 17.0858	**164.0800 ± 1.8466**	0.7735 ± 0.0647	**0.9160 ± 0.0000**	0.8663 ± 0.0376	**0.9520 ± 0.0000**
12	62.08 ± 19.89	**40.00 ± 0.00**	0.2937 ± 0.1406	**0.1527 ± 0.0000**	121.2800 ± 18.1487	**139.7200 ± 1.7205**	0.7969 ± 0.0653	**0.8473 ± 0.0000**	0.8759 ± 0.0381	**0.9116 ± 0.0000**
13	126.36 ± 7.33	**40.00 ± 0.00**	0.8574 ± 0.0805	**0.1527 ± 0.0000**	42.9600 ± 11.4727	**139.4000 ± 1.8930**	0.6036 ± 0.1208	**0.8473 ± 0.0000**	0.7413 ± 0.0603	**0.9116 ± 0.0000**
14	**86.00 ± 0.00**	**86.00 ± 0.00**	**0.3923 ± 0.0224**	0.3963 ± 0.0143	**78.0400 ± 2.7911**	77.6190 ± 2.2243	0.6935 ± 0.0096	**0.6950 ± 0.0065**	0.8166 ± 0.0033	**0.8171 ± 0.0023**
15	96.48 ± 1.71	**94.00 ± 0.00**	0.6751 ± 0.0408	**0.6252 ± 0.0034**	61.0800 ± 3.1348	**61.4500 ± 2.9105**	**0.8510 ± 0.0578**	0.8023 ± 0.0042	**0.8769 ± 0.0267**	0.8556 ± 0.0019
16	105.72 ± 4.39	**99.00 ± 0.00**	0.7868 ± 0.0314	**0.7497 ± 0.0130**	43.7600 ± 5.0438	**46.0000 ± 3.4008**	0.8935 ± 0.0699	**0.9589 ± 0.0431**	0.8917 ± 0.0353	**0.9268 ± 0.0208**
17	110.29 ± 0.69	**110.00 ± 0.00**	0.7959 ± 0.0360	**0.7769 ± 0.0232**	59.0833 ± 2.1653	**59.8000 ± 1.8028**	**0.7745 ± 0.0769**	0.7356 ± 0.0382	**0.8446 ± 0.0368**	0.8260 ± 0.0181
18	319.12 ± 122.11	**73.00 ± 0.00**	0.5030 ± 0.2245	**0.0857 ± 0.0000**	358.2400 ± 128.1475	**576.9600 ± 1.4283**	0.6222 ± 0.1968	**0.9121 ± 0.0000**	0.7827 ± 0.1099	**0.9510 ± 0.0000**

TABLE III: Monte-Carlo test results with average values of objective function, the number of exceptional elements, and four criteria achieved using CNO-MP over 50 runs with random initialization and their best-known values in the literature on the 18 datasets

#	$f(X,Y)$	EE	PE ↓		BE ↑		MU ↑		GE ↑	
			CNO-MP	best-known	CNO-MP	best-known	CNO-MP	best-known	CNO-MP	best-known
1	3.00 ± 0.00	0.00 ± 0.00	**0.0000 ± 0.0000**	**0.0000** [5]	**14.0000 ± 0.0000**	**14.0000** [5]	**0.8235 ± 0.0000**	**0.8235** [5]	**0.9118 ± 0.0000**	**0.9118** [5]
2	5.00 ± 0.00	2.00 ± 0.00	**0.1250 ± 0.0000**	**0.1250** [59]	**15.0000 ± 0.0000**	**15.0000** [59]	**0.8235 ± 0.0000**	**0.8235** [59]	**0.8562 ± 0.0000**	**0.8562** [59]
3	7.00 ± 0.00	4.00 ± 0.00	0.2000 ± 0.0000	**0.1250** [59]	**19.2000 ± 1.0000**	16.0000 [59]	**0.8421 ± 0.0000**	0.8235 [59]	0.7961 ± 0.0000	**0.8562** [59]
4	10.00 ± 0.00	3.00 ± 0.00	**0.0000 ± 0.0000**	**0.0000** [61]	**23.0000 ± 0.0000**	21.0000 [61]	**0.7059 ± 0.0000**	**0.7059** [61]	**0.8529 ± 0.0000**	**0.8529** [61]
5	4.00 ± 0.00	0.00 ± 0.00	**0.0000 ± 0.0000**	**0.0000** [62]	**63.0000 ± 0.0000**	**63.0000** [62]	**0.9200 ± 0.0000**	**0.9200** [62]	**0.9600 ± 0.0000**	**0.9600** [62]
6	21.00 ± 0.00	0.00 ± 0.00	**0.1304 ± 0.0000**	**0.1304** [60]	14.4800 ± 0.5099	**18.0000** [60]	**0.5263 ± 0.0000**	**0.5263** [60]	**0.7247 ± 0.0000**	**0.7247** [60]
7	9.00 ± 0.00	9.00 ± 0.00	**0.1475 ± 0.0000**	**0.1475** [43]	**78.6400 ± 0.7000**	78.0000 [43]	**1.0000 ± 0.0000**	**1.0000** [43]	**0.9583 ± 0.0000**	**0.9583** [43]
8	0.00 ± 0.00	0.00 ± 0.00	**0.0000 ± 0.0000**	**0.0000** [50]	**68.0000 ± 0.0000**	**68.0000** [50]	**1.0000 ± 0.0000**	**1.0000** [50]	**1.0000 ± 0.0000**	**1.0000** [50]
9	104.00 ± 0.00	83.60 ± 5.07	0.7464 ± 0.0452	**0.1140** [63]	65.2800 ± 1.4866	**78.0000** [7]	**0.5853 ± 0.0169**	0.4280 [68]	**0.6911 ± 0.0048**	0.6667 [7]
10	1.00 ± 0.00	0.00 ± 0.00	**0.0000 ± 0.0000**	**0.0000** [51]	194.0000 ± 0.0000	**198.0000** [51]	0.9924 ± 0.0000	**1.0000** [51]	0.9962 ± 0.0000	**1.0000** [51]
11	21.00 ± 0.00	10.00 ± 0.00	**0.0769 ± 0.0000**	**0.0769** [51]	**164.0800 ± 1.8466**	163.0000 [51]	**0.9160 ± 0.0000**	**0.9160** [51]	**0.9520 ± 0.0000**	**0.9520** [51]
12	40.00 ± 0.00	20.00 ± 0.00	**0.1527 ± 0.0000**	**0.1527** [51]	139.7200 ± 1.7205	**143.0000** [51]	**0.8473 ± 0.0000**	**0.8473** [51]	**0.9116 ± 0.0000**	**0.9116** [7]
13	40.00 ± 0.00	20.00 ± 0.00	**0.1527 ± 0.0000**	**0.1527** [51]	139.4000 ± 1.8930	**142.0000** [51]	**0.8473 ± 0.0000**	0.8284 [51]	**0.9116 ± 0.0000**	**0.9116** [51]
14	86.00 ± 0.00	51.52 ± 1.86	0.3963 ± 0.0143	**0.3664** [68]	77.6190 ± 2.2243	**95.0000** [68]	**0.6950 ± 0.0065**	0.6434 [68]	**0.8171 ± 0.0023**	0.7928 [68]
15	94.00 ± 0.00	81.90 ± 0.45	0.6252 ± 0.0034	**0.4046** [68]	61.4500 ± 2.9105	**76.0000** [68]	**0.8023 ± 0.0042**	0.5909 [51]	**0.8556 ± 0.0019**	0.7635 [7]
16	99.00 ± 0.00	97.46 ± 1.69	0.7497 ± 0.0130	**0.4351** [68]	46.0000 ± 3.4008	**68.0000** [68]	**0.9589 ± 0.0431**	0.5290 [51]	**0.9268 ± 0.0208**	0.7292 [7]
17	110.00 ± 0.00	99.44 ± 2.97	0.7769 ± 0.0232	**0.0234** [68]	59.8000 ± 1.8028	**68.0000** [7]	**0.7356 ± 0.0382**	0.2850 [65]	**0.8260 ± 0.0181**	0.6388 [7]
18	73.00 ± 0.00	36.00 ± 0.00	**0.0857 ± 0.0000**	**0.0857** [51]	576.9600 ± 1.4283	**577.0000** [51]	**0.9121 ± 0.0000**	**0.9121** [51]	**0.9510 ± 0.0000**	**0.9510** [51]

TABLE IV: Counts of best results achieved using CNO-MP and twelve baselines on 18 datasets

Method	# of Best				
	PE	BE	MU	GE	total
CNO-MP (herein)	12	8	17	16	53
ROC [5]	1	1	1	1	4
MACE [59]	2	2	2	2	6
PFA [61]	1	1	1	1	4
DCA [62]	1	1	1	1	4
Two clustering [60]	1	1	1	1	4
Ideal-seed [43]	1	1	1	1	4
P-median [50]	1	1	1	1	4
OKD [63]	1	0	0	0	1
ZODIAC [51]	5	5	7	4	21
Subconstructing [65]	0	0	1	0	1
LA [7]	0	2	0	5	2
TPC [68]	4	3	2	1	7

[5] J. R. King and V. Nakornchai, "Machine-component group formation in group technology: Review and extension," *International Journal of Production Research*, vol. 20, no. 2, pp. 117–133, 1982.

[6] J. Wang, "A linear assignment clustering algorithm based on the least similar cluster representatives," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 29, no. 1, pp. 100–104, Jan. 1999.

[7] ——, "Formation of machine cells and part families in cellular manufacturing systems using linear assignment algorithm," *Automatica*, vol. 39, pp. 1607–1615, 2003.

[8] I. Mahdavi, A. Aalaei, M. M. Paydar, and M. Solimanpur, "Multi-objective cell formation and production planning in dynamic virtual cellular manufacturing systems," *International Journal of Production Research*, vol. 49, no. 21, pp. 6517–6537, 2011.

[9] B. Elbenani and J. A. Ferland, "An exact method for solving the manufacturing cell formation problem," *International Journal of Production Research*, vol. 50, no. 15, pp. 4038–4045, 2012.

[10] D. Li, M. Li, X. Meng, and Y. Tian, "A hyperheuristic approach for intercell scheduling with single processing machines and batch processing machines," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 2, pp. 315–325, 2014.

[11] W. Nunkaew and B. Phruksaphanrat, "Lexicographic fuzzy multi-objective model for minimisation of exceptional and void elements in manufacturing cell formation," *International Journal of Production Research*, vol. 52, no. 5, pp. 1419–1442, 2014.

[12] Y. Won and R. Logendran, "Effective two-phase p-median approach for the balanced cell formation in the design of cellular manufacturing system," *International Journal of Production Research*, vol. 53, no. 9, pp. 2730–2750, 2015.

[13] M. M. Paydar and M. Saidi-Mehrabad, "Revised multi-choice goal programming for integrated supply chain design and dynamic virtual cell formation with fuzzy parameters," *International Journal of Computer Integrated Manufacturing*, vol. 28, no. 3, pp. 251–265, 2015.

[14] I. E. Utkina, M. V. Batsyn, and E. K. Batsyna, "A branch-and-bound algorithm for the cell formation problem," *International Journal of Production Research*, vol. 56, no. 9, pp. 3262–3273, 2018.

[15] S. Ibrahim and B. Jarboui, "A general variable neighborhood search approach based on a p-median model for cellular manufacturing problems," *Optimization Letters*, vol. 16, no. 1, pp. 137–151, 2022.

[16] I. Zaabar, V. Polotski, L. Bérard, B. El-Ouaqaf, Y. Beauregard, and M. Paquet, "A two-phase part family formation model to optimize resource planning: a case study in the electronics industry," *Operational Research*, vol. 22, p. 4441–4469, 2022.

[17] R. Motahari, Z. Alavifar, A. Z. Andaryan, M. Chipulu, and M. Saberi, "A multi-objective linear programming model for scheduling part families and designing a group layout in cellular manufacturing systems," *Computers & Operations Research*, vol. 151, p. 106090, 2023.

[18] C. Liu, J. Wang, M. Zhou, and T. Zhou, "Intelligent optimization approach to cell formation and product scheduling for multifactory cellular manufacturing systems considering supply chain and operational error," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 8, pp. 4649–4660, 2023.

[19] T. W. Liao and L. Chen, "An evaluation of ART1 neural models for GT part family and machine cell forming," *Journal of Manufacturing Systems*, vol. 12, no. 4, pp. 282–290, 1993.

[20] P. Venkumar and A. Noorul Haq, "Manufacturing cell formation using modified ART1 networks," *The International Journal of Advanced Manufacturing Technology*, vol. 26, pp. 909–916, 2005.

[21] M.-S. Yang and J.-H. Yang, "Machine-part cell formation in group technology using a modified ART1 method," *European Journal of Operational Research*, vol. 188, no. 1, pp. 140–152, 2008.

[22] J. Wang and A. Kusiak, Eds., *Computational Intelligence in Manufacturing Handbook*. CRC Press, 2000.

[23] 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.

[24] J. J. Hopfield and D. W. Tank, "Computing with neural circuits - a model," *Science*, vol. 233, no. 4764, pp. 625–633, 1986.

[25] 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.

[26] M. Wang, Y. Xie, and S. Qin, "An adaptive memristor-programming neurodynamic approach to nonsmooth nonconvex optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 11, pp. 6874–6885, 2023.

[27] Y. Zhao, X. Liao, and X. He, "Distributed smoothing projection neurodynamic approaches for constrained nonsmooth optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 2, pp. 675–688, 2023.

[28] X. Liao, Y. Zhao, and X. Zhou, "Neurodynamic flow approach for convex and quasi-convex optimization on Riemannian manifolds

with diagonal metrics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 4, pp. 1995–2007, 2024.

[29] L. Luan, S. Qin, J. Sheng, and X. Jiang, "Generalized second-order neurodynamic approach for distributed optimal allocation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 6, pp. 3369–3380, 2024.

[30] H. Wen, X. He, T. Huang, and J. Yu, "Neurodynamic algorithms with finite/fixed-time convergence for sparse optimization via ℓ_1 regularization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 1, pp. 131–142, 2024.

[31] Y. Xia, Q. Liu, J. Wang, and A. Cichocki, "A survey of neurodynamic optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 4, pp. 2677–2696, 2024.

[32] 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.

[33] H. Che and J. Wang, "A collaborative neurodynamic approach to global and combinatorial optimization," *Neural Networks*, vol. 114, pp. 15 – 27, 2019.

[34] ——, "A nonnegative matrix factorization algorithm based on a discrete-time projection neural network," *Neural Networks*, vol. 103, pp. 63–71, 2018.

[35] X. Li, J. Wang, and S. Kwong, "Boolean matrix factorization based on collaborative neurodynamic optimization with Boltzmann machines," *Neural Networks*, vol. 153, pp. 142–151, 2022.

[36] H. Che, J. Wang, and A. Cichocki, "Bicriteria sparse nonnegative matrix factorization via two-timescale duplex neurodynamic optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4881–4891, Aug. 2023.

[37] H. Li, J. Wang, N. Zhang, and W. Zhang, "Binary matrix factorization via collaborative neurodynamic optimization," *Neural Networks*, vol. 176, p. 106348, Aug. 2024.

[38] H. Li and J. Wang, "A collaborative neurodynamic algorithm for quadratic unconstrained binary optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 9, no. 1, pp. 228–239, Feb. 2025.

[39] J. Wang, J. Wang, and Q.-L. Han, "Multi-vehicle task assignment based on collaborative neurodynamic optimization with discrete Hopfield networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5274–5286, Dec. 2021.

[40] X. Li, J. Wang, and S. Kwong, "Hash bit selection via collaborative neurodynamic optimization with discrete Hopfield networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5116–5124, Oct. 2022.

[41] H. Li and J. Wang, "Capacitated clustering via majorization-minimization and collaborative neurodynamic optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 6, pp. 6679–6692, Jun. 2024.

[42] W. T. McCormick Jr, P. J. Schweitzer, and T. W. White, "Problem decomposition and data reorganization by a clustering technique," *Operations Research*, vol. 20, no. 5, pp. 993–1009, 1972.

[43] M. P. Chandrasekharan and R. Rajagopalan, "An ideal seed non-hierarchical clustering algorithm for cellular manufacturing," *International Journal of Production Research*, vol. 24, no. 2, pp. 451–463, 1986.

[44] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1983, pp. 448–453.

[45] Z. Xia, Y. Liu, and J. Wang, "A collaborative neurodynamic approach to distributed global optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 5, pp. 3141–3151, May 2023.

[46] Z. Chen, J. Wang, and Q.-L. Han, "Chiller plant operation planning via collaborative neurodynamic optimization," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, vol. 53, no. 8, pp. 4623–4635, 2023.

[47] Y. Jiang, Z. Peng, and J. Wang, "Safety-certified multi-target circumnavigation with autonomous surface vehicles via neurodynamics-driven distributed optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 4, pp. 2092–2103, Apr. 2024.

[48] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[49] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.

[50] G. Srinlvasan, T. Narendran, and B. Mahadevan, "An assignment model for the part-families problem in group technology," *International Journal of Production Research*, vol. 28, no. 1, pp. 145–152, 1990.

[51] M. P. Chandrasekharan and R. Rajagopalan, "ZODIAC—an algorithm for concurrent formation of part-families and machine-cells," *International Journal of Production Research*, vol. 25, no. 6, pp. 835–850, 1987.

[52] C. Buchheim and G. Rinaldi, "Efficient reduction of polynomial zero-one optimization to the quadratic case," *SIAM Journal on Optimization*, vol. 18, no. 4, pp. 1398–1413, 2008.

[53] M. Anthony, E. Boros, Y. Crama, and A. Gruber, "Quadratization of symmetric pseudo-Boolean functions," *Discrete Applied Mathematics*, vol. 203, pp. 1–12, 2016.

[54] ——, "Quadratic reformulations of nonlinear binary optimization problems," *Mathematical Programming*, vol. 162, no. 1, pp. 115–144, 2017.

[55] E. Boros, Y. Crama, and E. Rodríguez-Heck, "Compact quadratizations for pseudo-Boolean functions," *Journal of Combinatorial Optimization*, vol. 39, pp. 687–707, 2020.

[56] A. Verma and M. Lewis, "Optimal quadratic reformulations of fourth degree pseudo-Boolean functions," *Optimization Letters*, vol. 14, no. 6, pp. 1557–1569, 2020.

[57] S. Elloumi, A. Lambert, and A. Lazare, "Solving unconstrained 0-1 polynomial programs through quadratic convex reformulation," *Journal of Global Optimization*, vol. 80, p. 231–248, 2021.

[58] 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.

[59] P. H. Waghodekar and S. Sahu, "Machine-component cell formation in group technology: Mace," *International Journal of Production Research*, vol. 22, no. 6, pp. 937–948, 1984.

[60] A. Kusiak and W. S. Chow, "Efficient solving of the group technology problem," *Journal of Manufacturing Systems*, vol. 6, no. 2, pp. 117–124, 1987.

[61] C. Mosier and L. Taube, "The facets of group technology and their impacts on implementation—a state-of-the-art survey," *Omega*, vol. 13, no. 5, pp. 381–391, 1985.

[62] H. M. Chan and D. A. Milner, "Direct clustering algorithm for group formation in cellular manufacture," *Journal of Manufacturing Systems*, vol. 1, no. 1, pp. 65–75, 1982.

[63] K. R. Kumar, A. Kusiak, and A. Vannelli, "Grouping of parts and components in flexible manufacturing systems," *European Journal of Operational Research*, vol. 24, no. 3, pp. 387–397, 1986.

[64] M. P. Chandrasekharan and R. Rajagopalan, "Groupability: An analysis of the properties of binary data matrices for group technology." *International Journal of Production Research*, vol. 27, no. 6, pp. 1035–1052, 1989.

[65] K. R. Kumar and A. Vannelli, "Strategic subcontracting for efficient disaggregated manufacturing," *International Journal of Production Research*, vol. 25, pp. 1715–1728, 1986.

[66] H. Li and J. Wang, "Collaborative annealing power k-means++ clustering," *Knowledge-Based Systems*, vol. 255, no. 109593, 2022.

[67] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2018.

[68] Z. Yan, J. Wang, and J. Fan, "Machine-cell and part-family formation in cellular manufacturing using a two-phase clustering algorithm," in *Proc. of IFAC World Congress*, vol. 19, no. 1, 2014, pp. 2605–2610.

Hongzong LI received the B.E. degree in automation from Northeastern University, Shenyang, Liaoning, China, in 2020, and the Ph.D. degree in computer science from City University of Hong Kong, Kowloon, Hong Kong, in 2024.

He is currently a postdoctoral fellow with the Generative AI Research and Development Center, the Hong Kong University of Science and Technology, Hong Kong. His current research interests include optimization, computational intelligence, and neural networks.

Jun Wang (Life Fellow) received his B.S. and M.S. degrees in 1982 and 1985 from Dalian University of Technology, Dalian, China, and his Ph.D. degree in 1991 from Case Western Reserve University, Cleveland, Ohio, USA. He held various academic positions at Dalian University of Technology, Case Western Reserve University, University of North Dakota, and the Chinese University of Hong Kong, Hong Kong. He also held various short-term or part-time visiting positions at the U.S. Air Force Armstrong Laboratory, Dayton, Ohio, USA; RIKEN Brain Science Institute, Tokyo, Japan; Huazhong University of Science and Technology, Wuhan, China; Shanghai Jiao Tong University, Shanghai, China; Dalian University of Technology, Dalian, China; and Swinburne University of Technology, Melbourne, Australia. He is currently a chair professor at City University of Hong Kong, Hong Kong. He was a recipient of several awards such as the Research Excellence Award from the Chinese University of Hong Kong (2008-2009), Outstanding Achievement Award from Asia-Pacific Neural Network Assembly (2011), *IEEE Transactions on Neural Networks* Outstanding Paper Award (2011), Neural Networks Pioneer Award from the IEEE Computational Intelligence Society (2014), and Norbert Wiener Award from the IEEE Systems, Man and Cybernetics Society (2019). He is the Editor-in-Chief of the *IEEE Transactions on Artificial Intelligence* since 2025 and was the Editor-in-Chief of the *IEEE Transactions on Cybernetics* (2014-2019). He served as the General Chair of the 13th/25th International Conference on Neural Information Processing (2006/2018) and the IEEE World Congress on Computational Intelligence (2008). He was a distinguished lecturer of the IEEE Systems, Man, and Cybernetics Society (2017-2022) and IEEE Computational Intelligence Society (2010-2012, 2014-2016).