Hindawi
Complexity
Volume 2019, Article ID 8405036, 13 pages
https://doi.org/10.1155/2019/8405036

Research Article

A Gradient-Based Interior-Point Method to Solve the Many-to-Many Assignment Problems

Nitish Das and P. Aruna Priya

Department of Electronics and Communication Engg., SRM Institute of Science and Technology, Kattankulathur 603203, Chennai, India

Correspondence should be addressed to Nitish Das; nitishdas.n@ktr.srmuniv.ac.in

Received 7 May 2019; Accepted 10 July 2019; Published 29 July 2019

Academic Editor: Alejandro F. Villaverde

The many-to-many assignment problem (MMAP) is a recent topic of study in the field of combinatorial optimization. In this paper, a gradient-based interior-point method is proposed to solve MMAP. It is a deterministic method which assures an optimal solution. In this approach, the relaxation of the constraints is performed initially using the cardinality constraint detection operation. Then, the logarithmic barrier function (LBF) based gradient descent approach is executed to reach an accurate solution. Experiments have been performed to validate the practical implementation of the proposed algorithm. It also illustrates a significant improvement in convergence speed for the large MMAPs (i.e., if group size, $\alpha \geq 80$) over state-of-the-art literature.

1. Introduction

Recently, the many-to-many assignment problem (MMAP) has drawn the attention of the researchers in the field of combinatorial optimization. It is the topic of interest because MMAP serves as an authentic model for the activities such as group role assignment, job scheduling, and machine routing, in an organizational structure [1, 2].

The performance of an organizational structure depends on its capability of handling a shared task [3]. Role-based collaboration (RBC) plays a vital role in managing a shared task efficiently [3, 4]. In an RBC system, the rating of each constituent agent is performed. Then, a role is assigned to an agent into a particular group based on their rating value [5, 6]. If there exists a one-to-one mapping between a task and an agent, then it is called a linear assignment problem (LAP) or one-to-one assignment problem [7]. Hence, RBC problem is termed as one-to-many assignment problem [6, 7]. Similarly, if a single agent can handle many tasks which are not assigned to any other agent and vice-versa, then a LAP turns into a MMAP by adding the constraint as mentioned earlier [6, 7]. MMAP is also known as the generalized assignment problem.

RBC is the widely investigated topic and many exhaustion techniques are proposed to solve it. If an agent is free of tasks in a group, then it can be assigned to any other group to perform a specific task. MMAP takes care of the condition as mentioned earlier. Therefore, MMAP is more appropriate to simulate the real world role assignment problem in an organizational structure as compared with RBC problem [6, 7]. To the best of author's knowledge, only a few approaches are proposed to solve MMAP in the literature. Hence, this study is confined to create a new framework to solve MMAP.

The Kuhn-Munkres (K-M) technique is the most fundamental technique to solve any combinatorial optimization problem. It works well with the rectangular cost matrices where rows are considered as agents and columns are considered as tasks. A back-tracking method is added to the K-M algorithm to solve MMAP [6, 7]. This approach consists of two stages which are as follows: (a) preparation stage; (b) processing stage. In the preparation stage, normalization of the cost matrix is performed for MMAP using the constraints. Then, in the processing stage, reduction of the cost matrix is conducted to allocate agents to tasks efficiently.

In literature, Lagrangian bound-based algorithms are explored to solve k-cardinality assignment problem, which is a conjugate dual of the MMAP [8, 9]. In the Lagrangian relaxation approach, the problem is divided into two subproblems which are as follows: "easy" problem and "complicating"

problem. The "easy" problem is referred to optimizing the objective function by relaxing the constraints associated with it. The "complicating" problem is stipulated for optimizing the objective function with constraints. In this technique, relaxation of the constraints is performed by embedding a penalty function into the objective function. Hence, the violation of the penalty function is depressed. In the case of the maximization problem, the fixed-multiplier term of objective function sets the upper bound. Further, the semi-Lagrangian technique is developed [8]. It works well when applied to the objective function with the inequality constraints.

The convergence speed and accuracy of an algorithm serve as the most prominent parameters while solving MMAP. Hence, a technique with high accuracy and convergence rate is desired.

The fundamental heuristic algorithms which operate on the evolutionary principle like the genetic algorithm (GA), particle swarm optimization (PSO), and differential evolution (DE) are explored in the literature to solve a combinatorial optimization problem. GA's convergence rate decreases while reaching the global optimum, which is a severe drawback [10, 11]. DE and PSO offer high convergence speed but demonstrate a premature convergence, that is a compromise with the accuracy of the obtained solution [12, 13]. On the other hand, the deterministic techniques like interior-point method, and Lagrangian approach have shown promising results for combinatorial problems as they reach the global optimum with high accuracy and convergence rate [14]. The mentioned deterministic approaches are beneficial in addressing a combinatorial as well as a discrete optimization problem [15].

The interior-point method is investigated extensively in the literature for solving convex optimization problems (i.e., both linear and nonlinear problems) with inequality constraints [16]. This method follows a linear programming (LP) model which contains a uniquely defined objective function with constraints [17, 18]. The constraints of the objective function are twice continuously differentiable. The interior-point technique performs the relaxation of the constraints of the LP model as a set of boundary conditions surrounding a specific solution region. From the geometric point of view, it moves towards a solution from either the exterior of the solution region or the interior of the solution region [16, 17]. The barrier method, such as the logarithmic barrier function (LBF) approach, is the fundamental type of interior-point technique explored in the literature [16].

Therefore, an interior-point method is adopted to solve MMAP. In the proposed technique, the relaxation of the constraints is performed initially using the cardinality constraint detection operation. Then, the LBF-based gradient descent approach is executed to obtain an optimal solution.

Experiments are performed to substantiate the feasibility of the proposed algorithm and its comparative performance analysis with the existing literature [7] is made. It requires only a limited number of iterations to perform the complete search. Therefore, the proposed algorithm outperforms for the large MMAPs (i.e., if group size, $\alpha \geq 80$) as compared with [7]. The computation time for each of its iteration is larger than [7]. Therefore, it offers a comparatively large convergence period w.r.t. [7] for the small MMAPs. Thus, the practical implication of the proposed algorithm is to solve the machine routing and scheduling problem in Manufacturing-based organizations [1, 2].

The rest of the paper is regulated as follows. Section 2 consists of problem formulation. The framework for the gradient-based interior-point method is described in Section 3. Section 4 presents the experimental setup for the proposed work and its comparative analysis with the state-of-the-art literature. In the end, conclusions are devised in Section 5.

2. Research Problem Formulation

A graph $G_{match} = (V_{match}, E_{match})$, where V_{match} represents the nodes and E_{match} denotes the edges, is described as a bipartite graph, if it satisfies the criteria such as $V_{match} = X_{match} \cup Y_{match}$ and $X_{match} \cap Y_{match} = \varpi$, where ϖ is a null set, and $E_{match} \subseteq X_{match} \times Y_{match}$ [7].

A bipartite graph G_{match} is described as a weighted bipartite graph if its every edge (i.e., E_{match}) consists of a weight (i.e., $w(i, j)$) [7]. The weight of one-to-one matching (i.e., M_{match}) is described in (1).

$$w\left(M_{match}\right) = \sum_{E_{match} \in M_{match}} w\left(E_{match}\right) \tag{1}$$

A matching (i.e., M_{match}) is called a perfect matching, if each node of X_{match} is assigned to one of the Y_{match} nodes. Therefore, a linear assignment problem (LAP) is defined as determining a perfect matching in G_{match} which is maximally weighted [7]. In this paper, it is considered that there are α agents and β tasks.

Let an ability limit vector of α agents be Λ. Here, an element Λ_i indicates the number of tasks that can be maximally assigned to an agent i (i.e., $0 \leq i < \alpha$) [7].

Let a task range vector of β tasks be Γ. Here, an element Γ_j represents the quantity of task j (i.e., $0 \leq j < \beta$) which are needed to be assigned [7].

Let a potential matrix, ϱ, be an $\alpha \times \beta$ matrix, where $\varrho_{ij} \in [0, 1]$ denotes a potential value of an agent i for the task j. Hence, $\varrho_{ij} = 1$ represents the highest value, while 0 represents the least value for a particular element.

Let an assignment matrix, ω, be defined as $\alpha \times \beta$ matrix, where ω_{ij} opts only the binary labels (i.e., 0 or 1). Hence, $\omega_{ij} = 1$ which denotes agent i is assigned to task j, whereas $\omega_{ij} = 0$ symbolizes agent i is not assigned to task j.

Let an agent i (i.e., $0 \leq i < \alpha$) can execute many but different tasks. Similarly, let a task j (i.e., $0 \leq i < \beta$) can be assigned to many but different agents. A LAP turns into a many-to-many assignment problem by adding the two constraints, as mentioned earlier. From literature [7], a many-to-many assignment problem is thus defined mathematically by (2).

$$\max\left\{\lambda = \sum_{i=0}^{\alpha-1} \sum_{j=0}^{\beta-1} \left(\varrho_{ij} \times \omega_{ij}\right)\right\} \tag{2}$$

with the constraints as given in (3), (4), and (5).

$$\omega_{ij} \in \{0,1\} \quad where \ (0 \le j < \beta) \ and \ (0 \le i < \alpha) \tag{3}$$

$$\sum_{j=0}^{\beta-1} \omega_{ij} \le \Lambda_i \quad where \ (0 \le i < \alpha) \tag{4}$$

$$\sum_{i=0}^{\alpha-1} \omega_{ij} = \Gamma_j \quad where \ (0 \le j < \beta) \tag{5}$$

3. Methodology

Initially, the potential matrix ϱ of dimension $\alpha \times \beta$ is created. Hence, each element of ϱ denotes the efficiency of a particular agent (i.e., out of α agents) performing a specific task (i.e., out of β tasks). Then, Λ and Γ are defined according to the problem constraints. In literature [7], it is proven in Theorem 6 (i.e., Decidable theorem) that MMAP becomes unsolvable, when the inequality defined by (6) occurs.

$$\sum_{i=0}^{\alpha-1} \Lambda_i < \sum_{j=0}^{\beta-1} \Gamma_j \tag{6}$$

Thus, it becomes mandatory to perform cardinality constraint detection (CCD) to avoid the condition described in (6). Therefore, CCD is performed using (7).

$$\sum_{i=0}^{\alpha-1} \Lambda_i \ge \sum_{j=0}^{\beta-1} \Gamma_j \tag{7}$$

The cardinality of Λ is always greater than or equal to the cardinality of Γ after performing CCD. Hence, matrix ϱ of dimension $\alpha \times \beta$ is expanded into a matrix ϱ_extend of dimension $\eta \times \eta$, where $\eta = \sum_{i=0}^{\alpha-1} \Lambda_i$ using Λ and Γ. The new columns of ϱ_extend are appended with zeroes. It is expressed in (8), where an agent i contains Λ_i rows in ϱ_extend and the task j contains Γ_j columns in ϱ_extend.

$$\max \left\{ \lambda = \sum_{i=0}^{\eta-1} \sum_{j=0}^{\eta-1} \left(\varrho\_extend_{ij} \times \omega_{ij} \right) \right\} \quad where \ \eta = \sum_{i=0}^{\alpha-1} \Lambda_i \tag{8}$$

From the literature [9], the concept of conjugate duality is used to transform the objective function for maximization (i.e., (8)) into a minimization problem. A dual-parameter μ is defined by (9) because the elements of ϱ_extend vary from 0 to 1.

$$\mu_{ij} = 1 - \varrho\_extend_{ij} \tag{9}$$

By applying (9) into (8), the objective function reduces to (10).

$$\min \left\{ \lambda_{dual} = \sum_{i=0}^{\eta-1} \sum_{j=0}^{\eta-1} \left(\mu_{ij} \times \omega_{ij} \right) = cost\,(\omega) \right\} \tag{10}$$

Let a graph G_{map} defined by $G_{map} = (V_{map}, E_{map})$, characterizes Equation (10). Thus, η denotes the total number of nodes in it. In the graph G_{map}, E_{map} (i.e., μ_{ij}) symbolizes the edge weights between the specific nodes (i.e., between an agent i and a task j), and V_{map} (i.e., column vectors of ω) denotes the set of nodes. Therefore, each node $\in V_{map}$, represents a specific state code as ω_{ij} can take only binary labels.

In the field of circuit design, the hypercube embedding based techniques, such as HARD [19] and Improved-FSMIM [20], are developed for hardware optimization. These techniques have never been examined to solve a generalized assignment problem such as MMAP. To the best of author's knowledge, the proposed work makes the first attempt to address the MMAP by employing the hypercube embedding process, which proves its novelty.

Let a hypercube be characterized as $\phi_\nu = (V_\phi, E_\phi)$, where ν is the dimension, V_ϕ is the set of vertices of ϕ_ν, and E_ϕ is the set of edges of ϕ_ν [21]. The total number of elements present in E_ϕ and V_ϕ, are defined by (11) and (12).

$$\left|E_\phi\right| = \left|V_\phi\right| \times \frac{\nu}{2} = 2^{\nu-1} \times \nu \tag{11}$$

$$\left|V_\phi\right| = 2^\nu \tag{12}$$

The hypercube embedding is applied to minimize (10). It is executed from graph G_{map} into the hypercube ϕ_ν [21, 22]. It is symbolized as $\omega : V_{map} \longrightarrow V_\phi$. It is a 1 to 1 mapping function. Hence, if a node i of graph G_{map} is represented by a particular binary state code, then the corresponding vertex of the hypercube (i.e., k_i) is denoted by the same code. Therefore, η-binary ν-vectors are defined by (13).

$$k_i \in \{k : k \in \{0,1\}^\nu\} \quad \text{where } i \in V_{map} \ \text{(i.e., column vectors of } \omega) \tag{13}$$

In the hypercube ϕ_ν, $dist_{ij}$ (where $i, j \in V_{map}$) symbolizes the Hamming distance between the vertices k_i and k_j. The pictorial representation for evaluation of $dist_{ij}$ is given in [20]. Its mathematical representation is shown in Equation (14), where ϑ_{ij} is the instantaneous value of k_{ij}. ϑ_{ij} value varies between -1 and 1 (a detailed description is provided in Section 3.1). Therefore, the cost function of this embedding process is given in (15) and (16).

$$dist_{ij} = \sum_{\xi=1}^{\nu} \left(\vartheta_{i\xi} - \vartheta_{j\xi} \right)^2 \quad where, \ k_{ij} = \begin{cases} 1 & if \ \vartheta_{ij} \ge 0 \\ 0 & if \ \vartheta_{ij} < 0 \end{cases} \tag{14}$$

$$cost\,(\omega) = \sum_{i=0}^{\eta-1} \sum_{j=0}^{\eta-1} \left(\mu_{ij} \times dist_{ij} \right) \tag{15}$$

$$cost\,(\omega) = \sum_{i=0}^{\eta-1} \sum_{j=0}^{\eta-1} \sum_{\xi=1}^{\nu} \left(\mu_{ij} \times \left(\vartheta_{i\xi} - \vartheta_{j\xi} \right)^2 \right) \tag{16}$$

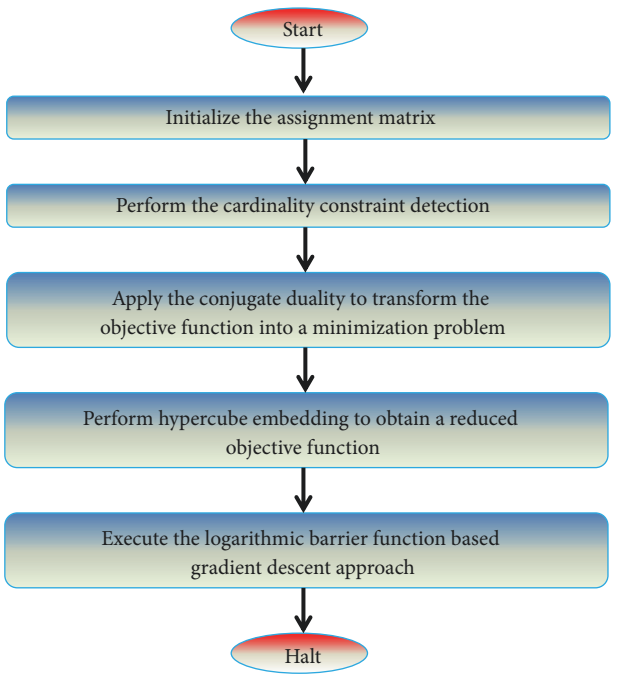

FIGURE 1: Flow chart for the proposed gradient-based interior-point method.

Thus, minimizing the cost function presented in (16) becomes the ultimate goal for solving the MMAP.

The convergence speed and accuracy of an algorithm serve as the most prominent parameters while solving the MMAP. Hence, a technique with high accuracy and convergence rate is chosen to minimize the cost function given in (16).

The basic heuristic algorithms which operate on the evolutionary principle like the genetic algorithm (GA), particle swarm optimization (PSO), and differential evolution (DE) are explored in the literature to solve a combinatorial optimization problem. GA's convergence rate decreases while reaching the global optimum, which is a severe drawback [10, 11]. DE and PSO offer high convergence speed but demonstrate a premature convergence, that is a compromise with the accuracy of the obtained solution [12, 13]. On the other hand, the deterministic techniques like interior-point method and Lagrangian approach have shown promising results for combinatorial problems as they reach the global optimum with high accuracy and convergence rate [14]. The mentioned deterministic approaches are beneficial in addressing a combinatorial as well as a discrete optimization problem [15]. The barrier method, such as the logarithmic barrier function (LBF) approach, is the fundamental type of interior-point technique explored in literature [16]. Hence, the LBF-based gradient descent technique is selected to minimize the cost function given in (16). The steps involved in the proposed gradient-based interior-point approach are presented in Figure 1.

3.1. Logarithmic Barrier Function Based Gradient Descent Approach. An interior-point method, such as logarithmic barrier function method (LBF), is widely investigated to solve discrete optimization problems [14, 23]. It is a deterministic technique that guarantees a feasible solution. The cost function for MMAP is formulated using LBF. Then, its minimization is performed using the gradient-projection technique in an iterative manner. In the LBF method, optimization of the objective function is accomplished in the continuous space domain. The obtained analytic solution is then discretized to produce the exact discrete solution [24]. Thus, the objective function for LBF, which is subjected to inequality constraints, is defined by (17).

$$\begin{aligned} \min \quad & f(\vartheta) \\ s.t., \quad & constraint_i(\vartheta) \geq 0, \\ & i = 1, 2, \ldots, u \end{aligned} \tag{17}$$

LBF to reduce the objective function (as given in (16)) is presented in (18). Its representation at the iteration $iter\_t$ is shown in (19). In (18) and (19), the second term acts as a barrier for any move that excludes constraint (i.e., $constraint_i(\vartheta)$) [25].

$$\min \quad \rho(\vartheta, \chi) = f(\vartheta) + \chi \sum_{i=1}^{u} \log_e (constraint_i(\vartheta)) \tag{18}$$

$$\begin{aligned} \min \quad & \rho\left(\vartheta, \chi^{iter\_t}\right)_{iter\_t} \\ & = f(\vartheta) + \chi^{iter\_t} \sum_{i=1}^{u} \log_e (constraint_i(\vartheta)) \end{aligned} \tag{19}$$

At the initial stage, LBF chooses $\chi^0 > 0$ and a feasible ϑ^0. It selects $\chi^{iter\_t+1} = \theta \cdot \chi^{iter\_t}$, where $\theta < 1$. The process continues until χ^{iter_t} arrives at a substantially small value.

A line search method is essential to reduce (19) w.r.t. ϑ iteratively. The first-order gradient descent approach is pertinent for this purpose [26]. In this method, weight vectors (i.e., the model parameters) are determined to reduce the objective function [27, 28]. Its mathematical representation is given in (20).

$$\begin{aligned} \min \quad & \rho(\vartheta, \chi) \\ s.t., \quad & \tau(\vartheta) = 0 \end{aligned} \tag{20}$$

In the first-order gradient descent method, a particular iteration (i.e., $iter\_t$) is defined by (21), where ψ represents the step size. A small positive real number is selected as ψ value [26].

$$\vartheta \longleftarrow \vartheta - \psi(\nabla \rho(\vartheta, \chi)) \tag{21}$$

From (21), small moves (i.e., with a step size ψ) are made in negative direction of the gradient. At the next iteration (i.e., $\vartheta^{(iter\_t+1)}$), (22) is employed to determine ϑ value over the constraint surface.

$$\vartheta \longleftarrow [\vartheta]^{\tau(\vartheta)=0} \tag{22}$$

The convergence norm for the proposed gradient-based interior-point method is given in (23), where σ (ranges from 0 to 1) is the upper bound.

$$\left|\vartheta^{(iter\_t+1)} - \vartheta^{(iter\_t)}\right| < \sigma \tag{23}$$

Hence, the hypercube embedding problem is interpreted as determining η-binary ν-vectors that minimizes (16). It is given in (24).

$$\begin{aligned} &\vartheta_i \in \left\{\vartheta : \vartheta \in \Re^{\nu} \;\&\; \|\vartheta\|^2 = 1\right\}; \\ &\text{where } i \in V_{map} \text{ and } \|\vartheta\|^2 \text{ indicates the normalized value of } \vartheta \end{aligned} \tag{24}$$

Thus, (16) is deduced using Hamming distance between vertices of the hypercube (i.e., ϕ_ν). It is presented in (25).

$$\begin{aligned} cost\,(\omega) &= \sum_{i=0}^{\eta-1}\sum_{j=0}^{\eta-1}\left(\mu_{ij} \times \|\vartheta_i - \vartheta_j\|^2\right) \\ &\text{where, } \vartheta_i = [\vartheta_{i1}, \vartheta_{i2}, \ldots, \vartheta_{i\nu}]^T \\ \|\vartheta_i - \vartheta_j\|^2 &= \sum_{\xi=1}^{\nu}\left(\vartheta_{i\xi} - \vartheta_{j\xi}\right)^2 \end{aligned} \tag{25}$$

In this optimization problem, any two vertices of the hypercube (i.e., ϕ_ν) should not be defined by the same binary state code (i.e., $\vartheta_i - \vartheta_j \neq 0$). This condition serves as a constraint for this optimization process. It is given in (26).

$$constraint\,(\vartheta) = \|\vartheta_i - \vartheta_j\|^2 > 0 \tag{26}$$

Equations (25) and (26) are applied into (18) to reduce the cost function. The reduced cost function is given in (27).

$$\begin{aligned} &\min \quad \rho(\vartheta, \chi) \\ &= \sum_{i=0}^{\eta-1}\sum_{j=0}^{\eta-1}\left(\mu_{ij} \times \|\vartheta_i - \vartheta_j\|^2\right) \\ &\quad + \chi\sum_{i=0}^{\eta-1}\sum_{j=0}^{\eta-1}\log_e\left(\|\vartheta_i - \vartheta_j\|^2\right) \end{aligned} \tag{27}$$

Therefore, the entity $\tau(\vartheta)$ (from (22)) is defined by (28).

$$\begin{aligned} &\tau(\vartheta) \\ &= \left[\left(\|\vartheta_1\|^2\right.\right. \\ &\left.\left. - 1\right), \left(\|\vartheta_2\|^2 - 1\right), \ldots, \left(\|\vartheta_\eta\|^2 - 1\right)\right]^T \\ &s.t. \quad \vartheta = \left[\vartheta_1, \vartheta_2, \ldots, \vartheta_\eta\right]^T \\ &\vartheta_i = [\vartheta_{i1}, \vartheta_{i2}, \ldots, \vartheta_{i\nu}]^T; \\ &\text{where each } \vartheta_i\,(1 \le i \le \eta) \end{aligned} \tag{28}$$

From (21), the computation of the derivative term (i.e., $\nabla\rho(\vartheta, \chi)$) is essential to approach towards the gradient descent direction. The required derivative term is deduced by substituting (29), (30), (31), and (32) into (27). Hence, $\nabla\rho(\vartheta, \chi)$ is defined by (33).

$$\frac{\partial}{\partial\vartheta_i}\left(\|\vartheta_i - \vartheta_j\|^2\right) = 2\left(\vartheta_i - \vartheta_j\right) \tag{29}$$

$$\frac{\partial}{\partial\vartheta_j}\left(\|\vartheta_i - \vartheta_j\|^2\right) = 2\left(\vartheta_j - \vartheta_i\right) \tag{30}$$

$$\frac{\partial}{\partial\vartheta_i}\left\{\log_e\left(\|\vartheta_i - \vartheta_j\|^2\right)\right\} = \frac{2\left(\vartheta_i - \vartheta_j\right)}{\|\vartheta_i - \vartheta_j\|^2} \tag{31}$$

$$\frac{\partial}{\partial\vartheta_j}\left\{\log_e\left(\|\vartheta_i - \vartheta_j\|^2\right)\right\} = \frac{2\left(\vartheta_j - \vartheta_i\right)}{\|\vartheta_i - \vartheta_j\|^2} \tag{32}$$

$$\begin{aligned} &\nabla\rho(\vartheta, \chi) \\ &= 2\begin{bmatrix} \sum_{j=1}^{\eta}\left\{\mu_{1j} \times \left(\vartheta_1 - \vartheta_j\right)\right\} + \chi\sum_{j=1}^{\eta}\left\{\frac{\left(\vartheta_1 - \vartheta_j\right)}{\|\vartheta_1 - \vartheta_j\|^2}\right\} \\ \sum_{j=1}^{\eta}\left\{\mu_{2j} \times \left(\vartheta_2 - \vartheta_j\right)\right\} + \chi\sum_{j=1}^{\eta}\left\{\frac{\left(\vartheta_2 - \vartheta_j\right)}{\|\vartheta_2 - \vartheta_j\|^2}\right\} \\ \cdots \\ \cdots \\ \cdots \\ \sum_{j=1}^{\eta}\left\{\mu_{\eta j} \times \left(\vartheta_\eta - \vartheta_j\right)\right\} + \chi\sum_{j=1}^{\eta}\left\{\frac{\left(\vartheta_\eta - \vartheta_j\right)}{\|\vartheta_\eta - \vartheta_j\|^2}\right\} \end{bmatrix} \end{aligned} \tag{33}$$

By applying (28) into (22), ϑ in its normalized form is given in (34).

$$[\vartheta]^{\tau(\vartheta)=0} = \begin{bmatrix} \frac{\vartheta_1}{\|\vartheta_1\|} \\ \frac{\vartheta_2}{\|\vartheta_2\|} \\ \cdots \\ \cdots \\ \frac{\vartheta_\eta}{\|\vartheta_\eta\|} \end{bmatrix} \tag{34}$$

At the end of the gradient descent iterative process (i.e., when (23) is satisfied), discretization of the analytic solution is performed. Hence, a set of optimal state codes (i.e., ω^*) is determined by discretizing $\widehat{\vartheta}_{ij}$ (i.e., value of ϑ at the iteration σ) by (35).

$$\widehat{k}_{ij} = \begin{cases} 1 & if\ \widehat{\vartheta}_{ij} \ge 0 \\ 0 & if\ \widehat{\vartheta}_{ij} < 0 \end{cases} \tag{35}$$

The pseudocode for the proposed gradient-based interior-point method is conferred in Algorithm 1.

Input: Λ_i, Γ_j and potential matrix (i.e., ϱ)
Output: ω^* (i.e., the final solution)
begin
Initialization: $\omega \longleftarrow$ *One hot encoded codes*; $\chi \longleftarrow$ *initial_*χ;
Cardinality constraint detection is performed using Equation (7);
Expansion of the potential matrix (i.e., ϱ) is executed using Equation (8);
Conversion of the objective function into a minimization problem
(i.e., Equation (10)) is conducted using Equation (9);
Hypercube embedding is performed & objective function is defined by Equation (16);
while ($\chi >$ *final_*χ) **do**/∗ **Start the LBF-based gradient projection method** ∗/
repeat
for *iter_t* $\longleftarrow 1$ *to* σ
$\vartheta^{iter\_t} \longleftarrow \vartheta^{(iter\_t-1)} - \psi\{\nabla\rho(\vartheta, \chi)\}$;
/∗ by Equation (21) and Equation (33)∗/
end
return $\widehat{\vartheta}_{ij}$ (i.e., ϑ value when *iter_t* $= \sigma$);
evaluate $\widehat{k}_{ij} = \{1, \text{ if } \widehat{\vartheta}_{ij} \geq 0;\ 0, \text{ if } \widehat{\vartheta}_{ij} < 0\}$/∗*by Equation (35)*∗/
Compute *cost* for the new value of ω by Equation (16);
case: $cost(\omega^{iter\_t-1}) \geq cost(\omega^{iter\_t})$ **then**
update, $\omega^* \longleftarrow \omega^{iter\_t}$;
case: $cost(\omega^{iter\_t-1}) < cost(\omega^{iter\_t})$ **then**
update, $\omega^* \longleftarrow \omega^{iter\_t-1}$;
end case
until the gradient based interior point method converges
$\chi \longleftarrow \theta \cdot \chi$;
end
return ω^*;
end

Algorithm 1: A Gradient based interior point method.

4. Numerical Results and Discussions

The fundamental goal of this numerical study is to substantiate the feasibility of the proposed algorithm and to demonstrate the comparison with the state-of-the-art literature [7]. All optimizations are performed using Eclipse Kepler JDK 1.7 package [7]. The workstation configuration to perform computations is as follows: Intel(R) Core i7 (6th Gen), 16 GB RAM and 3.5 GHz CPU.

The proposed approach is a deterministic approach. Therefore, its convergence period for optimal assignment is constant. Experiments are performed between two random groups (where $\varrho_{ij} \in [0, 1]$) to validate its feasibility. For Λ_i range, such as $[1, 3]$, $[1, 4]$, and $[1, 6]$, the Γ_j range is set to $[1, 2\alpha/\beta]$, where *mid_value* $\longleftarrow 2$, to present the comparison with [7].

Similarly, for Λ_i range $[1, 5]$, the Γ_j range is set to $[1, 3\alpha/\beta]$, where *mid_value* $\longleftarrow 3$, to present the comparison with [7]. Simulations are performed to evaluate the cost for optimal assignment and convergence time for different combinations of α, β, and Λ_i range. The cost for a particular assignment matrix is evaluated using (16). The convergence plot for different combinations of α, β, and Λ_i range is presented in Figure 2. A parameter, *success rate*, is also defined which provides the average of successful assignments within 100 iterations. It is presented in Table 1. The *success rate* ranges from 95.78 to 100%. It indicates that the proposed algorithm takes only a limited number of iterations to converge.

Another set of analysis is conducted to determine the overall performance (for α ranging from 20 to 300) of the proposed algorithm. Experiments are performed between 100 random groups (where $\varrho_{ij} \in [0, 1]$). Simulations are run to evaluate the cost for optimal assignment and convergence time for three combinations (i.e., $(\alpha, \beta \longleftarrow \alpha)$, $(\alpha, \beta \longleftarrow \alpha/2)$, and $(\alpha, \beta \longleftarrow \alpha/3)$) with various Λ_i range. The cost for a particular assignment matrix is evaluated using (16). The convergence analysis of the proposed algorithm for $\Lambda_i \in [1, 3]$ and $[1, 5]$ are shown in Figure 3. For Λ_i range, such as $[1, 3]$, $[1, 4]$, $[1, 6]$, and $[1, 7]$, the Γ_j range is set to $[1, 2\alpha/\beta]$, *where mid* − *value* $\longleftarrow 2$. Similarly, for Λ_i range, $[1, 5]$, the Γ_j range is set to $[1, 3\alpha/\beta]$, *where*, *mid* − *value* $\longleftarrow 3$, to present the comparison with [7]. The performance measures using the convergence time for various Λ_i range is presented in Tables 2 and 3. An analysis of computation time required for the optimal assignment is presented in Figure 4. Therefore, the performance of the proposed algorithm in comparison with [7] when $\Lambda_i \in [1, 3]$ and $[1, 5]$ are shown in Figure 5.

The proposed technique optimally defines the most feasible solution region. Hence, only a limited number of iterations are required to perform the complete search. Therefore, the proposed algorithm outperforms for the large MMAPs (i.e., if group size, $\alpha \geq 80$) as compared with [7].

Table 1: The convergence times and *success rate* for different combinations of α, β, and Λ_i range.

α	β	Λ_i range	*success rate* %	Converg. time (ms)	α	β	Λ_i range	*success rate* %	Converg. time (ms)
5	5	[1, 3]	100	0.087	5	5	[1, 5]	99.91	0.18
5	10	[1, 3]	100	0.114	5	10	[1, 5]	98.87	0.194
5	20	[1, 3]	99.39	0.101	5	20	[1, 5]	98.65	0.242
5	50	[1, 3]	100	0.119	5	50	[1, 5]	97.34	0.237
10	5	[1, 3]	99.27	0.147	10	5	[1, 5]	98.21	0.252
10	10	[1, 3]	99.23	0.141	10	10	[1, 5]	98.2	0.27
10	20	[1, 3]	99.12	0.156	10	20	[1, 5]	98.06	0.301
10	50	[1, 3]	100	0.177	10	50	[1, 5]	98.03	0.371
15	5	[1, 3]	99.04	0.151	15	5	[1, 5]	97.97	0.354
15	10	[1, 3]	98.97	0.193	15	10	[1, 5]	97.89	0.386
15	20	[1, 3]	98.94	0.197	15	20	[1, 5]	98.48	0.382
15	50	[1, 3]	98.47	0.28	15	50	[1, 5]	97.54	0.347
5	5	[1, 4]	98.34	0.096	5	5	[1, 6]	97.41	0.203
5	10	[1, 4]	98.32	0.11	5	10	[1, 6]	97.38	0.239
5	20	[1, 4]	98.18	0.106	5	20	[1, 6]	96.88	0.217
5	50	[1, 4]	98.69	0.117	5	50	[1, 6]	96.67	0.319
10	5	[1, 4]	97.95	0.123	10	5	[1, 6]	98.81	0.267
10	10	[1, 4]	97.83	0.147	10	10	[1, 6]	96.6	0.336
10	20	[1, 4]	97.56	0.18	10	20	[1, 6]	97.62	0.293
10	50	[1, 4]	98.44	0.156	10	50	[1, 6]	96.5	0.337
15	5	[1, 4]	97.29	0.185	15	5	[1, 6]	96.3	0.384
15	10	[1, 4]	99.25	0.163	15	10	[1, 6]	95.78	0.406
15	20	[1, 4]	97.1	0.159	15	20	[1, 6]	96.23	0.445
15	50	[1, 4]	96.17	0.29	15	50	[1, 6]	96.16	0.452

Table 2: Performance measures in terms of convergence time for various Λ_i range.

α	β	Convergence time (ms) for various Λ_i range				
		[1, 3]	[1, 4]	[1, 5]	[1, 6]	[1, 7]
20	α	1.5	1.53	3.05	3.35	5.01
	$\alpha/2$	3.91	2.5	3.91	6.67	6.7
	$\alpha/3$	2.56	4.19	6.74	8	9.46
40	α	2.49	2.39	11.96	22.27	25.47
	$\alpha/2$	15.09	15.43	30.2	25.42	28.67
	$\alpha/3$	6.24	8.73	51.99	38.21	67.68
60	α	3.47	11.54	29.37	43.08	215.78
	$\alpha/2$	45.08	58.75	93.77	35.53	222.17
	$\alpha/3$	10.4	13.1	103.97	109.42	226.11
80	α	5.66	25.8	63.88	148.13	260.1
	$\alpha/2$	64.6	87.19	126.34	138.86	327.09
	$\alpha/3$	22.41	26.41	211.65	214.94	378.63
100	α	10.93	54.1	124.86	336.19	517.8
	$\alpha/2$	104.36	158.86	286.24	417.04	594.34
	$\alpha/3$	28.03	39.7	314.03	319.75	600.45
120	α	21.94	74.21	175.46	437.63	747.46
	$\alpha/2$	175.39	271.75	404.09	653.25	1017.74
	$\alpha/3$	37.65	66.33	567.18	528.19	1190.04
140	α	46.59	133.76	298.22	732.7	1412.76
	$\alpha/2$	226.04	349.16	571.05	840.7	1337.31
	$\alpha/3$	50.37	103.49	805.83	826.76	2330.55

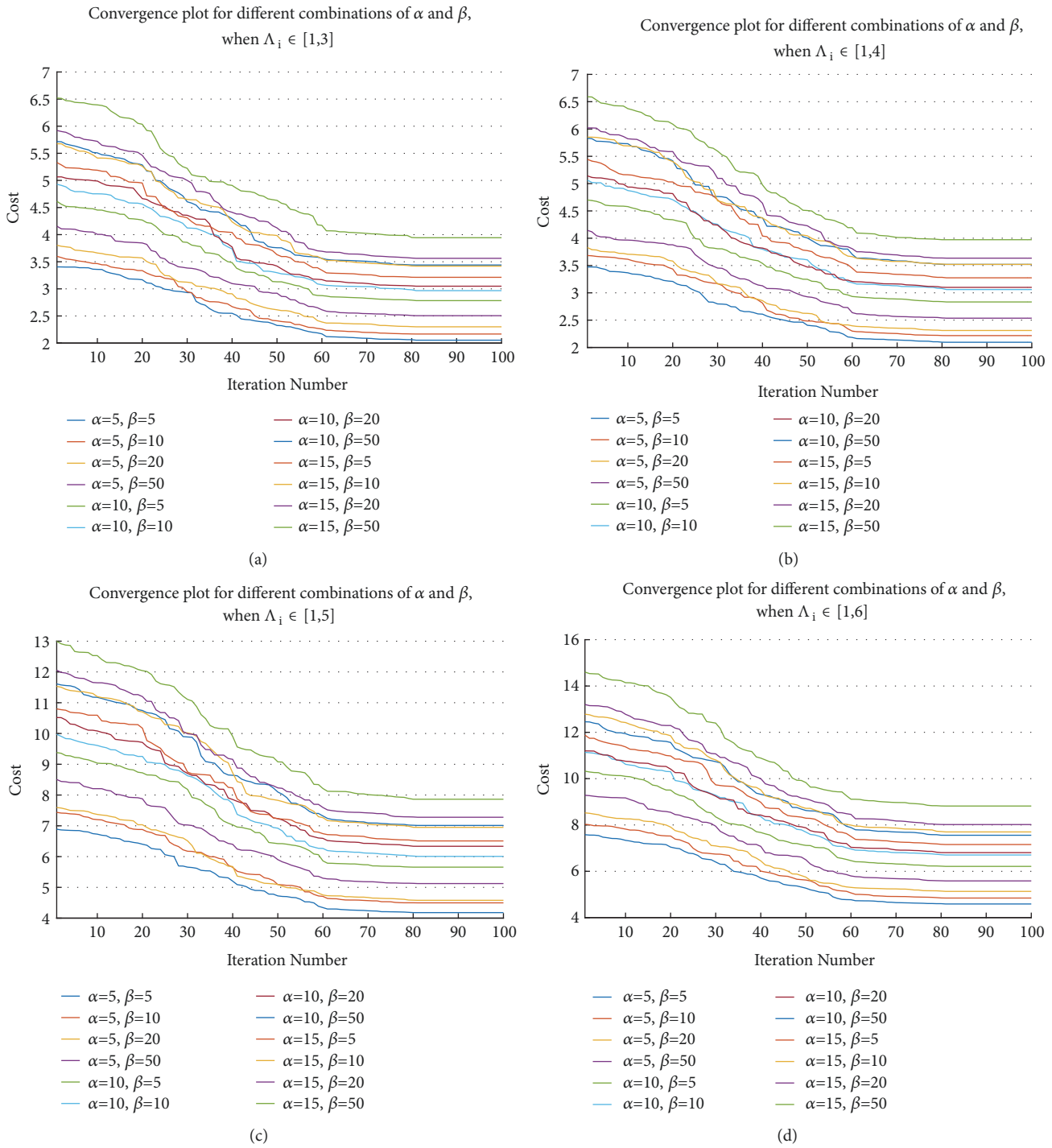

Figure 2: Convergence plot for different combinations of α, β, and Λ_i range.

The comparative analysis presented in Figure 5 indicates that each iteration of the proposed technique requires a substantial computation time as compared with [7]. Therefore, it offers a comparatively large convergence period w.r.t. [7] for the small MMAPs. It is a limitation of the proposed approach.

The problem of machine routing and scheduling is the primary issue in manufacturing-based industries. The mentioned problems become severe in the multisite manufacturing organizations when the number of operators working on a shared task is large [1, 2]. Therefore, the practical implication

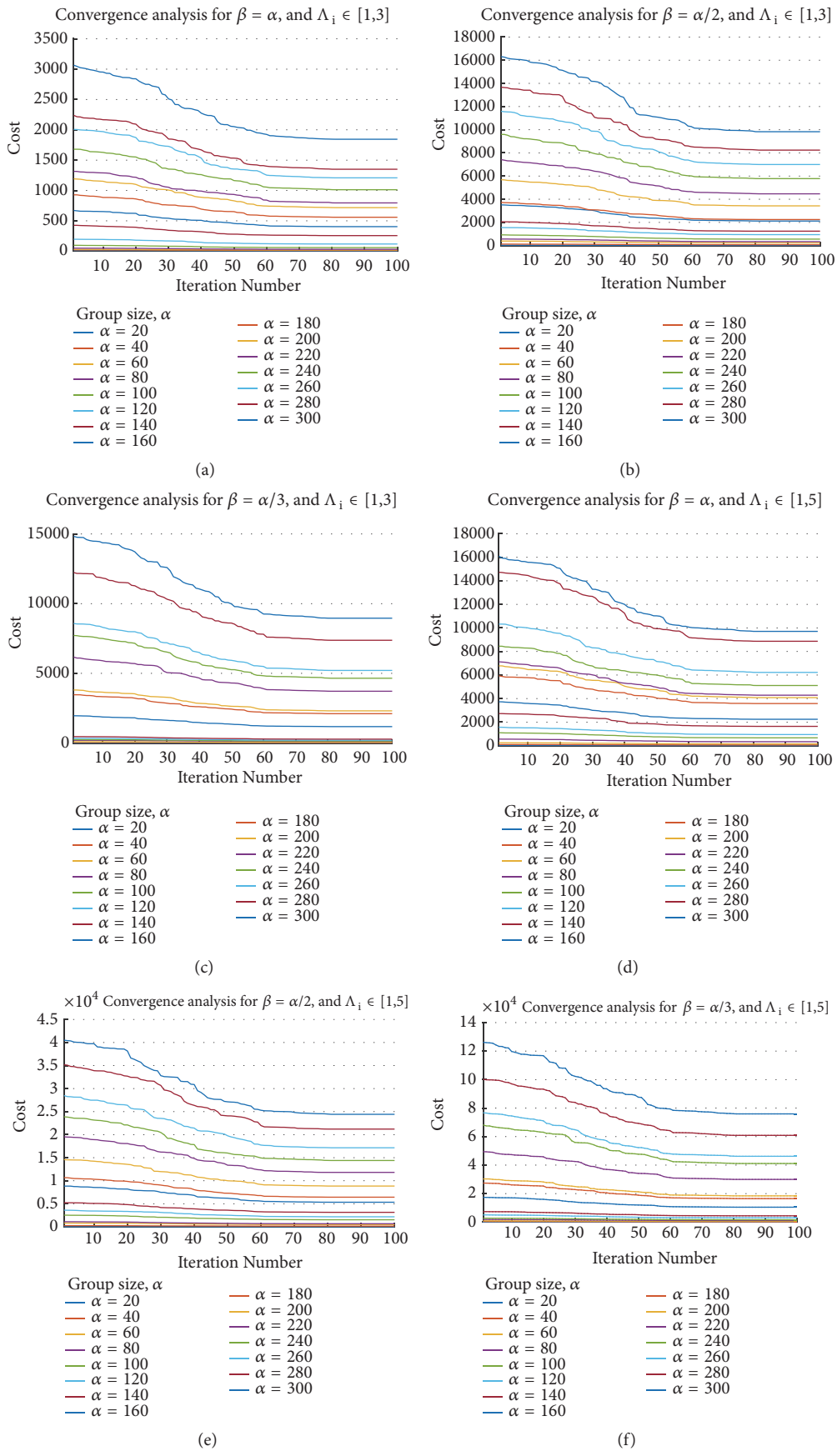

Figure 3: Convergence analysis of the proposed algorithm for $\Lambda_i \in [1,3]$ & $[1,5]$.

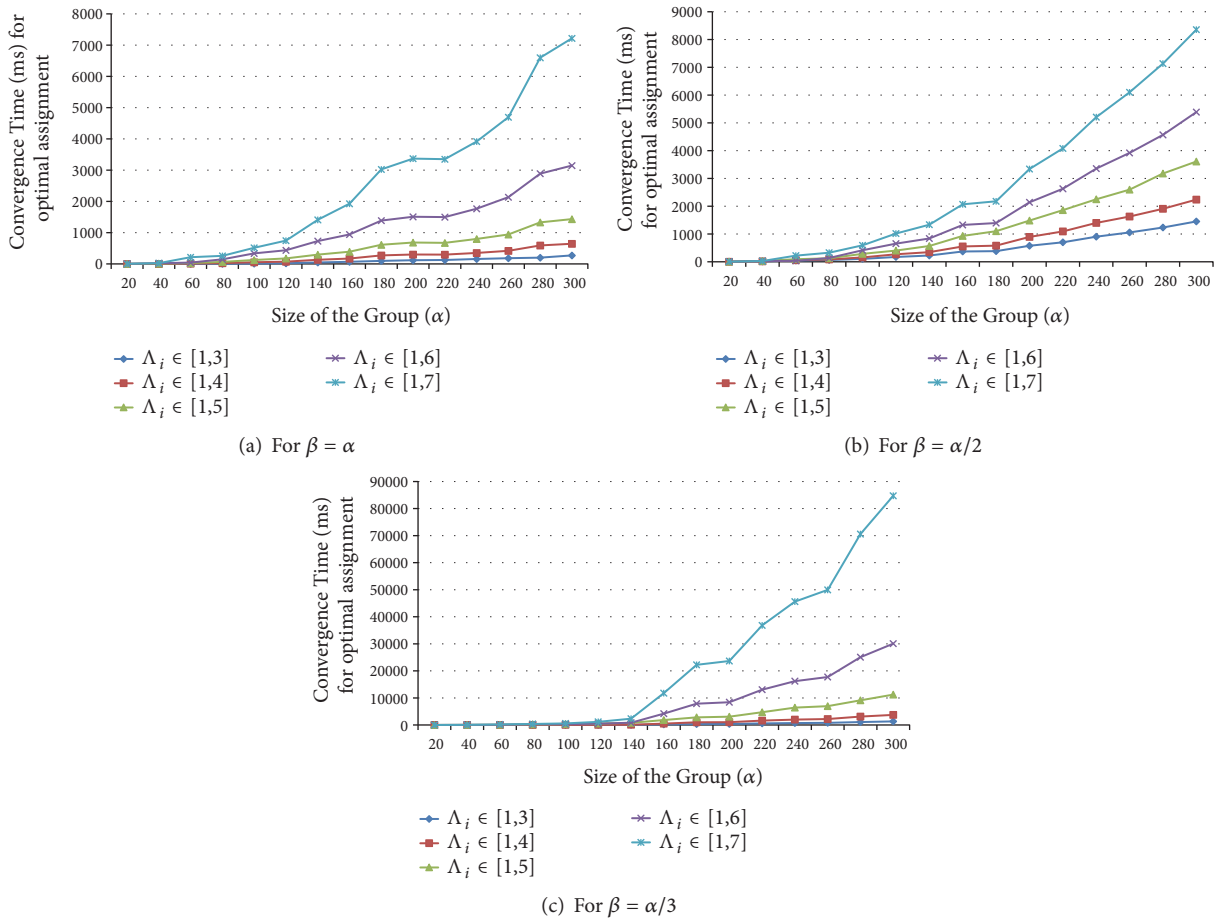

Figure 4: Performance evaluation of the proposed algorithm in terms of convergence time for various Λ_i range.

Table 3: Performance measures in terms of convergence time for various Λ_i range.

α	β	Convergence time (ms) for various Λ_i range				
		$[1,3]$	$[1,4]$	$[1,5]$	$[1,6]$	$[1,7]$
160	α	70.81	172.6	392.73	943.3	1928.53
	$\alpha/2$	371.03	554.58	934.28	1329.13	2070.21
	$\alpha/3$	206.33	521.98	1834.04	4182.59	11774.68
180	α	95.94	271.51	614.03	1387.2	3025.63
	$\alpha/2$	385.72	583.6	1103.61	1394.25	2179.26
	$\alpha/3$	361.22	1002.62	2829.37	7899.4	22248.78
200	α	120.55	302.94	685.8	1510.65	3368.68
	$\alpha/2$	575.91	894.46	1486.22	2143.75	3339.83
	$\alpha/3$	387.91	1048.01	3094.74	8403.4	23672.39
220	α	125.25	298.88	675.04	1497.18	3350.68
	$\alpha/2$	703.4	1093.11	1860.01	2628.96	4081.13
	$\alpha/3$	585.1	1633.06	4717.64	13059.29	36823.85
240	α	158.35	352.81	799.02	1768.21	3916.86
	$\alpha/2$	903.46	1399.31	2250.16	3355.76	5206.06
	$\alpha/3$	726.87	2020.59	6428.67	16192.45	45616.99
260	α	182.88	419.81	941.7	2135.22	4694.84
	$\alpha/2$	1058.77	1631.79	2595.69	3919.32	6101.87
	$\alpha/3$	788.25	2218.62	6989.23	17728.56	49945.42
280	α	202.38	591.12	1329.22	2889.11	6598.76
	$\alpha/2$	1234.98	1907.92	3179.06	4567.74	7133.22
	$\alpha/3$	1105.22	3115.55	9123.93	25077.61	70553.6
300	α	272.68	646.28	1434.9	3148.24	7216.9
	$\alpha/2$	1452.65	2239.31	3608.83	5386.14	8353.25
	$\alpha/3$	1324.04	3753.62	11225.06	30067.39	84718.73

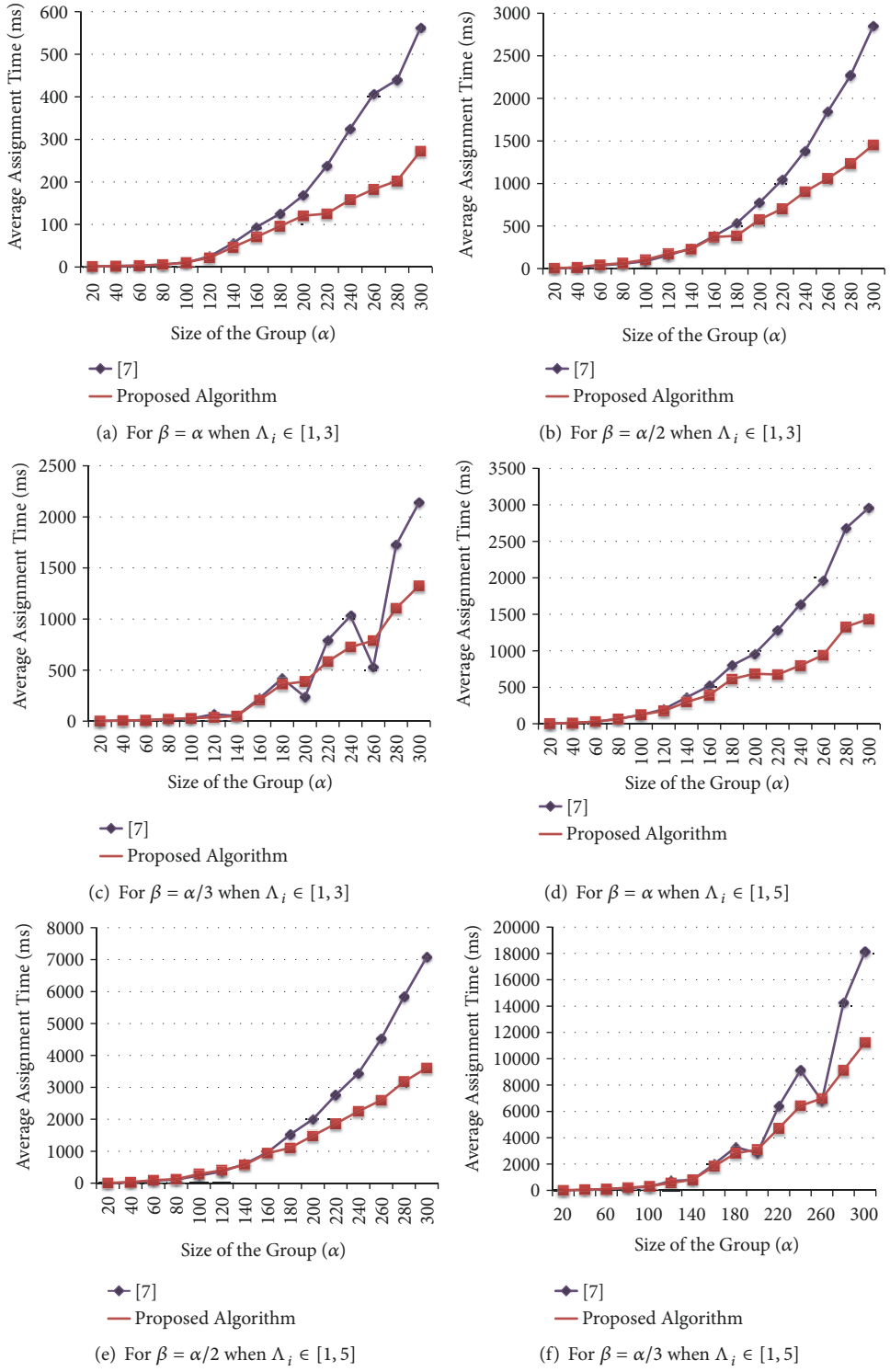

FIGURE 5: The performance of the proposed algorithm in comparison with the state-of-the-art literature when $\Lambda_i \in [1, 3]$ & $[1, 5]$.

of the proposed algorithm is to solve the machine routing and scheduling problem in Manufacturing-based organizations. The proposed technique can also be investigated to create reconfigurable FSM-based architecture efficiently [29].

5. Conclusion

The framework for a gradient-based interior-point method to solve the many-to-many assignment problem (MMAP)

is proposed in this study. It is a deterministic approach which assures an optimal solution. At the initial stage of this approach, the relaxation of the constraints is performed using the cardinality constraint detection operation. Then, the logarithmic barrier function (LBF) based gradient descent approach is executed to obtain an optimal solution for MMAP. The proposed technique requires only a limited number of iterations to perform the complete search. Therefore, the proposed algorithm outperforms for the large MMAPs (i.e., if group size, $\alpha \geq 80$) as compared with the existing literature. The practical implication of the proposed algorithm is to solve the machine routing and scheduling problem in manufacturing-based organizations [1, 2]. The proposed technique can also be investigated to create reconfigurable FSM-based architecture efficiently [29].

In organizational structures, the role assignment problem is more often strictly limited to LAPs, RBCs, or MMAPs. Hence, a unified algorithm is required that can handle the issues mentioned earlier. The proposed technique will be examined further for creating a unified method to efficiently solve LAPs, RBCs, or MMAPs by adding an adaptive operator (see [30]) in it.

Data Availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Disclosure

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research work is performed in the Department of Electronics and Communication Engg., SRM Institute of Science and Technology, Kanchipuram Dist. 603203, Chennai, India.

References

[1] Y. Hadad and B. Keren, "A revised method for allocating the optimum number of similar machines to operators," *International Journal of Productivity and Performance Management*, vol. 65, no. 2, pp. 223–244, 2016.

[2] M. A. Beheshtinia, A. Ghasemi, and M. Farokhnia, "Supply chain scheduling and routing in multi-site manufacturing system (case study: a drug manufacturing company)," *Journal of Modelling in Management*, vol. 13, no. 1, pp. 27–49, 2018.

[3] Y. Sheng, H. Zhu, X. Zhou, and W. Hu, "Effective approaches to adaptive collaboration via dynamic role assignment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 1, pp. 76–92, 2016.

[4] X. Zhu, X. Fangxiong, H. Zhu, W. Qilin, X. Zhou, and H. Wenting, "Group role assignment with flexible formation based on the genetic algorithm," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, pp. 2661–2666, Canada, October 2017.

[5] H. Zhu and Y. Zhu, "Group role assignment with agents' busyness degrees," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, pp. 3201–3206, Canada, October 2017.

[6] H. Zhu, D. Liu, S. Zhang, S. Teng, and Y. Zhu, "Solving the group multirole assignment problem by improving the ILOG approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3418–3424, 2017.

[7] H. Zhu, D. Liu, S. Zhang, Y. Zhu, L. Teng, and S. Teng, "Solving the Many to Many assignment problem by improving the Kuhn–Munkres algorithm with backtracking," *Theoretical Computer Science*, vol. 618, pp. 30–41, 2016.

[8] I. Belik and K. Jörnsten, "A new Semi-Lagrangean Relaxation for the k-cardinality assignment problem," *Journal of Information and Optimization Sciences*, vol. 37, no. 1, pp. 75–100, 2016.

[9] P. T. Thach and T. V. Thang, "Conjugate duality for vector-maximization problems," *Journal of Mathematical Analysis and Applications*, vol. 376, no. 1, pp. 94–102, 2011.

[10] G. Pavai and T. V. Geetha, "New crossover operators using dominance and co-dominance principles for faster convergence of genetic algorithms," *Soft Computing*, vol. 23, no. 11, pp. 3661–3686, 2019.

[11] S. Ganjefar and M. Tofighi, "Optimization of quantum-inspired neural network using memetic algorithm for function approximation and chaotic time series prediction," *Neurocomputing*, vol. 291, pp. 175–186, 2018.

[12] H. Huang, L. Lv, S. Ye, and Z. Hao, "Particle swarm optimization with convergence speed controller for large-scale numerical optimization," *Soft Computing*, pp. 1–17, 2018.

[13] R. Knobloch, J. Mlýnek, and R. Srb, "The classic differential evolution algorithm and its convergence properties," *Applications of Mathematics*, vol. 62, no. 2, pp. 197–208, 2017.

[14] P. Armand and R. Omheni, "A mixed logarithmic barrier-augmented Lagrangian method for nonlinear optimization," *Journal of Optimization Theory and Applications*, vol. 173, no. 2, pp. 523–547, 2017.

[15] W. Murray and K.-M. Ng, "An algorithm for nonlinear optimization problems with binary variables," *Computational optimization and applications*, vol. 47, no. 2, pp. 257–288, 2010.

[16] R. W. Cottle and M. N. Thapa, "Interior-point methods," in *Linear and Nonlinear Optimization*, C. C. Price, J. Zhu, and F. S. Hillier, Eds., pp. 517–536, Springer Nature, New York, NY, USA, 2017.

[17] N. Ploskas and N. Samaras, "Interior point methods," in *Linear Programming Using MATLAB*, P. M. Pardalos and D.-Z. Du, Eds., pp. 491–540, Springer Nature, Cham, Swizerland, 2017.

[18] D. G. Luenberger and Y. Ye, "Interior-point methods," in *Linear and Nonlinear Programming*, C. C. Price, J. Zhu, and F. S. Hillier, Eds., pp. 115–143, Springer International Publishing, Switzerland, 4th edition, 2016.

[19] I. Ahmad, "HARD: A hypercube embedding algorithm for state assignment of finite state machines," *Computers and Electrical Engineering*, vol. 29, no. 2, pp. 327–356, 2003.

[20] N. Das and P. Aruna Priya, "FPGA implementation of an improved reconfigurable FSMIM architecture using logarithmic barrier function based gradient descent approach," *International Journal of Reconfigurable Computing*, vol. 2019, 17 pages, 2019.

[21] K. Kabyl, A. Berrachedi, and É. Sopena, "A note on the cubical dimension of new classes of binary trees," *Czechoslovak Mathematical Journal*, vol. 65, no. 1, pp. 151–160, 2015.

[22] M. Liu and H.-M. Liu, "Vertex-fault-tolerant cycles embedding on enhanced hypercube networks," *Acta Mathematicae Applicatae Sinica*, vol. 32, no. 1, pp. 187–198, 2016.

[23] R. Gárciga Otero and A. Iusem, "A proximal method with logarithmic barrier for nonlinear complementarity problems," *Journal of Global Optimization*, vol. 64, no. 4, pp. 663–678, 2016.

[24] L. Menniche and D. Benterki, "A logarithmic barrier approach for linear programming," *Journal of Computational and Applied Mathematics*, vol. 312, pp. 267–275, 2016.

[25] R. Shen, Z. Meng, C. Dang, and M. Jiang, "Algorithm of barrier objective penalty function," *Numerical Functional Analysis and Optimization*, vol. 38, no. 11, pp. 1–17, 2017.

[26] I. Chakroun, T. Haber, and T. J. Ashby, "SW-SGD: the sliding window stochastic gradient descent algorithm," *Procedia Computer Science*, vol. 108, pp. 2318–2322, 2017.

[27] A. Senov and O. Granichin, "Projective approximation based gradient descent modification," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3899–3904, 2017.

[28] B. Kheirfam and M. Haghighi, "A wide neighborhood interior-point algorithm for linear optimization based on a specific kernel function," *Periodica Mathematica Hungarica*, 2018.

[29] N. Das and P. A. Priya, "FPGA implementation of reconfigurable finite state machine with input multiplexing architecture using hungarian method," *International Journal of Reconfigurable Computing*, vol. 2018, Article ID 6831901, 15 pages, 2018.

[30] E. Özcan, J. H. Drake, C. Altıntaş, and S. Asta, "A self-adaptive Multimeme Memetic Algorithm co-evolving utility scores to control genetic operators and their parameter settings," *Applied Soft Computing*, vol. 49, pp. 81–93, 2016.