An improved Jaya algorithm with a modified swap operator for solving team formation problem

Walaa H. El-Ashmawi^{1,2} · Ahmed F. Ali¹ · Adam Slowik³

Published online: 12 May 2020
© The Author(s) 2020

Abstract

Forming a team of experts that can match the requirements of a collaborative task is an important aspect, especially in project development. In this paper, we propose an improved Jaya optimization algorithm for minimizing the communication cost among team experts to solve team formation problem. The proposed algorithm is called an improved Jaya algorithm with a modified swap operator (IJMSO). We invoke a single-point crossover in the Jaya algorithm to accelerate the search, and we apply a new swap operator within Jaya algorithm to verify the consistency of the capabilities and the required skills to carry out the task. We investigate the IJMSO algorithm by implementing it on two real-life datasets (i.e., digital bibliographic library project and StackExchange) to evaluate the accuracy and efficiency of proposed algorithm against other meta-heuristic algorithms such as genetic algorithm, particle swarm optimization, African buffalo optimization algorithm and standard Jaya algorithm. Experimental results suggest that the proposed algorithm achieves significant improvement in finding effective teams with minimum communication costs among team members for achieving the goal.

Keywords Jaya algorithm · Team formation problem · Social networks · Genetic algorithm · Modified swap operator

1 Introduction

Team formation problem (TFP) considers an important role in many real-life applications and in social networks which are extending from software project development to different collaborative tasks. There is a community set of experts associated with a diverse skill sets in social networks. The goal is forming teams that cover the incoming tasks, in which each task requires a set of skills that must be covered by a team

members. The problem is how to form a team that should have small communication cost, according to the underlying social network. This problem can be formulated as NP-hard problem (Lappas et al. 2009) that required the development of meta-heuristic algorithms to solve it.

Most of the published papers in team formation are using approximation algorithms (Anagnostopoulos et al. 2012; Kargar et al. 2013), which consider diameter and minimum spanning tree as a communication cost (Lappas et al. 2009) or sum of distance from each member to team leader (Kargar and An 2011). The authors in Appel et al. (2014); Li and Shan (2010); Li et al. (2015) generalized the problem by allocating each skill to a determine number of experts. Others consider the maximum number of experts depending on different tasks (Anagnostopoulos et al. 2012) without using communication cost for team formation.

The authors in Gutiérrez et al. (2016) presents a team formation problem based on sociometric matrix. They considered a variable neighborhood search (VNS) the most efficient in almost all cases. The authors in Huang et al. (2017) considered a team formation based on the work time availability and skills for each expert in order to form an effective team. The authors in Farasat and Nikolaev (2016) proposed a mathematical model that maximizes team reliability depend-

Communicated by V. Loia.

✉ Adam Slowik
aslowik@ie.tu.koszalin.pl

Walaa H. El-Ashmawi
w.hashmawi@ci.suez.edu.eg ;
walaa.hassan@miuegypt.edu.eg

Ahmed F. Ali
ahmed_fouad@ci.suez.edu.eg

¹ Department of Computer Science, Faculty of Computers & Informatics, Suez Canal University, Ismailia, Egypt

² Faculty of Computer Science, Misr International University, Cairo, Egypt

³ Department of Electronics & Computer Science, Koszalin University of Technology, Koszalin, Poland

ing on the probability of unreliable experts which can exit from the team and preparing a backup for each unreliable member in the team.

Particle swarm optimization (PSO) and genetic algorithm (GA) have been applied in a small number of research to solve (TFP) (Haupt and Haupt 2004). These algorithms obtained promising results when they applied to solve real-world problems (Blum and Roli 2003) and (Pashaei et al. 2015; Sedighizadeh and Masehian 2009). The authors in Zhang and Si (2010) presented a group formation based on genetic algorithm, where the members are assigned to each group based on the programming skill for students. The authors in Nadershahi and Moghaddam (2012) used a genetic algorithm in team formation on the bases of Belbin team role which are using nine roles to form a team based on members specialty and attitude toward team working. The authors in Fathian et al. (2017) proposed a framework for treating social structure among experts in the team formation problem.

The authors in Han et al. (2017) combine the communication cost and geographical proximity into a unified objective function to solve TFP. They applied their algorithm to optimize the proposed objective function using a genetic algorithm. In the paper (Awal and Bharadwaj 2014), the genetic algorithm-based approach is applied to optimize computational collective intelligence in Web-based social networks.

Although the above research work of meta-heuristic algorithms in TFP has reached a solution to the problem, the TFP with consideration of social network is still relatively limited. Due to a large number of experts in a social network, the formation of feasible teams with various skill set requires an efficient optimization algorithm. The Jaya algorithm has proved its efficiency in solving optimization problems due to several advantages such as parameterless and victorious nature of it (Pandey 2016).

The main objective of this research lies on forming a most feasible team of experts for task achievement by minimizing the communication cost among team members. We propose an improved Jaya algorithm with a modified swap operator and single-point crossover to guarantee that the whole population moving toward the global optimum. The proposed algorithm is called an improved Jaya with a modified Swap Operator (IJMSO).

The paper is organized as follows. Section 2 describes the Jaya algorithm. In Sect. 3, we present the definition of the team formation problem. Section 4 presents the proposed IJMSO algorithm in details. In Sect. 5, we discuss the experimental results of the proposed model against some existing methods. Finally, we conclude and the future work made up Sect. 6.

2 Jaya algorithm

Jaya algorithm is a population-based meta-heuristic algorithm proposed by Rao in 2016 (Rao 2016). Due to its efficiency, many researchers have applied it on their works such as in Trivedi et al. (2016); the authors use a Jaya algorithm to solve the economic dispatch problem to achieve optimal cost of the micro-grid. In Rao et al. (2016), the authors use a Jaya algorithm in the dimensional optimization of a micro-channel heat sink.

The authors in Rao and Saroj (2017) use a method which is called self-adaptive multi-population-based Jaya (SAMP-Jaya) to control the search process based on the problem landscape using a Jaya algorithm. The authors in Rao and More (2017) proposed an improved Jaya algorithm called self-adaptive Jaya for optimal design of selected thermal devices such as heat pipe and cooling tower. The Jaya algorithm has proved its efficiency in the economic optimization problems as in Rao and Saroj (2017a, b), in which the authors use an elitist-Jaya algorithm to minimize the setup and the operational cost of shell-and-tube heat exchanger design. At the same time, researchers in Rao and Rai (2017a) and (Rao and Rai 2017b) use Jaya algorithm in welding process where the objective is the optimal selection of submerged arc welding process parameters. In addition, Jaya can be used in combination with other methods in identification and detection problems as in Zhang et al. (2016) where the authors proposed a system that can determine tea category from the captured images using a digital camera. They use fractional Fourier entropy for feature extraction then fed to a feed-forward neural network with optimal obtained weights from Jaya algorithm. In Wang et al. (2017), the authors use a Jaya algorithm for training the classifier of neural network to detect the abnormal breasts in mammogram images which captured by digital mammography. The authors in Dede (2018) applied Jaya algorithm for steel grillage structures. In the paper (Grzywinski et al. 2019), the Jaya algorithm is used for the problem of the optimum mass of braced dome structures with natural frequency constraints. Design variables of the bar cross-sectional area and coordinates of the structure nodes were used for size and shape optimization, respectively. Also, the authors in Rao et al. (2016) proposed a Jaya algorithm multi-objective Jaya (MOJaya algorithm) for solving multi-objective optimization models for machining processes as it considered an important aspects of plasma and the same as in Rao et al. (2016) which considered a surface grinding optimization process. It has recently used in the engineering problems as in Rao and Waghmare (2017) where the authors use a Jaya algorithm to test the performance of four mechanical design problems. As in Rao and More (2017), an improved Jaya algorithm proposed to minimize the energy expenses of cooling tower.

All of the previous works that have been done based on Jaya optimization algorithm or a modified version of standard Jaya concentrated on the mechanical/engineering and economic problems. In addition, most of these problems are solved in a continuous domain, and to our knowledge, no research paper till now considered the Jaya algorithm in social life such as team formation problem as one of the discrete optimization problems.

3 Team formation problem (TFP)

Consider a number of n experts $V = \{v_1, v_2, \dots, v_n\}$ exist in a social network $SN(V, E)$ to achieve a specific task. E is the set of edges connecting the experts (i.e., weight). $e(v_i, v_j) \in E$ which represents the communication cost between expert v_i and expert v_j . A set of m skills is given as follows, $S = \{s_1, s_2, \dots, s_m\}$ in which each expert belongs to a skill set $s(v_i), s(v_i) \subset S$. The subset of experts with the skill s_k is represented as $C(s_k)$ (i.e., $C(s_k) \subset V$). A task T is a subset of the required skills to perform (i.e., $T = \{s_i, \dots, s_j\} \subseteq S$) by a set of experts to form a team X . Therefore, the task ($T \subseteq \bigcup_{v_i \in X} s(v_i)$). Our goal is to find a set of most feasible experts that form a feasible team among all possible teams with minimum communication cost among team members. This problem can be considered an optimization problem. We summarize the notations of the TFP in Table 1 and give an example of it in Fig. 1.

In Fig. 1, we consider a social network of experts $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, and each individual (expert) has a set of skills S with a communication cost (weight) between every two experts v_i, v_j (e.g., $e(v_1, v_2) = 0.2$).

The main objective of this paper is to form team X of experts V that have required skills S by minimizing the communication cost among its individuals. In Fig. 1, we got two teams with the required skills $X_1 = \{v_1, v_2, v_3, v_4\}$ and $X_2 = \{v_2, v_4, v_5, v_6\}$.

Table 1 Definitions of TFP's parameters

Parameter	Definition
V	Experts set
$SN(V, E)$	Social network of experts
S	Skill set
T	A task that has required skills
X	Team of experts
$C(s_k)$	Experts set with skill s_k
$s(v_i)$	Skill of expert v_i
$e(v_i, v_j)$	Communication cost between experts v_i and v_j

4 The proposed algorithm

In the following subsections, we present the main processes of the proposed algorithm and we describe how it works.

4.1 Principles of Jaya algorithm

Jaya algorithm is a population-based meta-heuristic algorithm proposed by Rao in 2016 (Rao 2016). Jaya is a simple algorithm because it has not parameters to set like the other meta-heuristic algorithm and mainly developed for solving a continuous optimization problems. In this subsection, the steps of Jaya algorithm are presented, as shown in Fig. 2, and described as follows.

- **Initialization.** The algorithm starts by generating the initial population randomly $X_{j,k}^{t+1}$, $j = 1, \dots, m$, m is the number of problem variables, $k = 1, 2, \dots, SS$, and SS is the population size.
- **Population evaluating.** At iteration t , the solutions in the population are evaluating where the best (X_{best}^t) and worst (X_{worst}^t) solutions are assigned.
- **Solutions updating.** Each solution in the population is updating based on the best and worst solutions as shown in Eq. 1:

Fig. 1 An example of TFP with communication cost

Fig. 2 Standard Jaya algorithm for minimization of objective function

$$X_{j,k}^{t+1} = X_{j,k}^t + r_{1,j}^t[(X_{j,best}^t) - |X_{j,k}^t|] - r_{2,j}^t[(X_{j,worst}^t) - |X_{j,k}^t|] \quad (1)$$

where $r_{1,j}^t$ and $r_{2,j}^t$ are two random numbers in the range $[0, 1]$. If the new solution $X_{j,k}^{t+1}$ is better than the current solution $X_{j,k}^t$, then the new solution becomes the current solution.

- **Termination criteria** The previous steps are repeated until termination criteria satisfied.

4.2 A modified swap operator (MSO)

In this work, we modified the swap operator (SO) in Wang et al. (2003); Wei et al. (2009); Zhang and Si (2010) which has two indices $SO(a, b)$ to be $MSO(a, b, c)$ with three indices. For example, suppose you have a solution $S = (1 - 2 -$

$3 - 4 - 5)$, the applied swap operator is $SO = (1, 3)$, and then, the obtained solution will be $S' = S + SO(1, 3) = (1 - 2 - 3 - 4 - 5) + SO(1, 3) = (3 - 2 - 1 - 4 - 5)$.

In IJMSO algorithm, the modified swap operator $MSO(a, b, c)$ has three indices: Index a is the $skill_{id}$, and indices b and c are the current and the new experts' indices, respectively, which are selected randomly; each of them has the same $skill_{id}$ and the arguments $b \neq c$. For example, $MSO(2, 1, 3)$ means for $skill_{id} = 2$ swap the $expert_{id} = 1$ with $expert_{id} = 3$.

The advantage of using MSO is exchanging experts (second and third indices) that share the same skill (the first index) within teams. This can guarantee that validity of the solution in terms of each expert's skill set. MSO plays a vital role for solution updating process in the standard Jaya algorithm because it is proposed for solving continues optimization problems not for solving discrete optimization problems.

4.3 Improved Jaya algorithm with a modified swap operator (IJMSO)

This section discusses the improved version of the Jaya algorithm for solving a discrete problem (i.e., team formation problem). The main structure of the proposed algorithm is presented in Algorithm 1, and its steps are summarized as follows.

- **Initialization.** In IJMSO, the initial population are generated randomly $X_{j,k}^t$, where $j = 1, \dots, m$, m is the dimension of the problem and $k = 1, \dots, SS$, and SS is the population size.
- **Solution evaluation.** The objective function $f(X^t)$ for each solution is calculated, and the best and the worst solutions are assigned in the population. The communication cost between two experts can be computed as shown in Eq. 2.

$$e_{ij} = 1 - \frac{s(v_i) \cap s(v_j)}{s(v_i) \cup s(v_j)} \quad (2)$$

where $s(v_i)$ and $s(v_j)$ are the skill set of expert v_i and expert v_j . TFP is an optimization problem and it can be defined as shown in Eq. 3.

$$\text{Min } f(x_i) = \sum_{i=1}^{x_i} \sum_{j=i+1}^{x_i} e_{ij} \quad (3)$$

where $(x_i \in X)$ is a cardinality of team X and $e_{ij} \in [0, 1]$ is a weight (communication cost) between each two adjacent experts in each solution.

- **Single-point crossover.** In order to improve the current solution, we obtained the overall best solution X_{best}^t and the current solution (X^t) and we select the best solution from the obtained offspring
- **Solution update.** The position of each solution in the population is updated according to Eq. 4, which represent the main conversion from continuous domain to discrete domain through using different operators and modified swap operator.

$$X_{j,k}^{t+1} = X_{j,k}^t \oplus r_{1,j}^t \otimes [(X_{j,\text{cross}}^t) - (X_{j,k}^t)] - r_{2,j}^t \otimes [(X_{j,\text{worst}}^t) - (X_{j,k}^t)] \quad (4)$$

where “ $\oplus$ ” is a combining operator of two swap operators. The mark “ $\otimes$ ” means the probability of $r_{1,j}^t$ that all swap operators are selected in the swap sequences $(X_{j,\text{cross}}^t) - (X_{j,k}^t)$ and the probability of $r_{2,j}^t$ that all swap operators are selected in the swap sequences $(X_{j,\text{worst}}^t) - (X_{j,k}^t)$.

If the new solution is better than the current solution,

then we accept the new solution; otherwise, we select the current solution to be the new solution.

- **Termination criteria.** The overall process are repeated until number of iterations.

Algorithm 1 Improved Jaya algorithm with a modified swap operator (IJMSO)

```

1: Initialize the population size SS
2: Initialize the iteration counter,  $t := 0$ 
3: for ( $j = 1; j \leq m; j++$ ) do
4:   for ( $k = 1; k \leq SS; k++$ ) do
5:     Generate the initial population randomly  $X_{j,k}^t$  { $m$  is the number of problem variables}
6:   end for
7: end for
8: Evaluate the fitness function of the population  $f(X^t)$ 
9: Assign the best ( $X_{\text{best}}^t$ ) and the worst ( $X_{\text{worst}}^t$ ) solutions
10: repeat
11:   for ( $j = 1; j \leq m; j++$ ) do
12:     for ( $k = 1; k \leq SS; k++$ ) do
13:       Apply crossover on the best solution  $X_{\text{best}}^t$  and current solution  $X_k^t$  {single-point crossover}
14:       Select the best obtained offspring solution  $X_{\text{cross}}^t$ 
15:       Update all solutions in the population as shown in Eq. 4 {Update solutions}
16:       if  $f(X_{j,k}^{t+1}) \leq f(X_{j,\text{best}}^t)$  then
17:          $X_{j,\text{best}}^t = X_{j,k}^{t+1}$ 
18:       else
19:          $X_{j,\text{best}}^t = X_{j,\text{best}}^t$ 
20:       end if
21:     end for
22:   end for
23:   Increase the iteration counter,  $t = t + 1$ 
24: until Termination criteria are satisfied
25: Get the best solution

```

4.4 An illustrative example of IJMSO for TFP

Given a task T that requires a set of skills to be archived, i.e., $T = \{\text{publications, phd, conference}\}$. Suppose we have a set of five experts (e.g., a, b, c, d and e) associated with their skills as follows:

$s(a) = \{\text{publications, conference, research}\},$
 $s(b) = \{\text{conference, funding, publications}\},$
 $s(c) = \{\text{journals, phd, research}\},$
 $s(d) = \{\text{publications, cv}\},$
 $s(e) = \{\text{conference, phd}\}.$

The nodes in the social network represent experts which connect to each other with communication cost (weight) as shown in Fig. 3.

The five teams with the required skills can be formed as follows. $X_1 = \{a, e\}$, $X_2 = \{a, c\}$, $X_3 = \{d, e\}$, $X_4 = \{d, e, b\}$ and $X_5 = \{a, c, b\}$. The objective functions of the

Fig. 3 The relationship between experts **Fig. 4** The representation of the solutions in the IJMSO algorithm

formed teams are as follows: $f(X_1) = 0.2$, $f(X_2) = 0.17$, $f(X_3) = \infty$, $f(X_4) = 0.4$ and $f(X_5) = 0.5$. If we consider all possible teams that can be formed, the most feasible team X^* is X_2 (i.e., the one that has minimum communication cost among team members).

According to the example in Fig. 4, three required skills are needed to accomplish a task. A solution in IJMSO algorithm is an array list of size 1×3 where the first needed skill is “publications,” the second one is “phd” and the third skill is “conference.” In Fig. 4, we represent the possible values for each index of a solution in the IJMSO algorithm. As for required skill_id=1, there are three experts that have this skill (i.e., a, b and d).

– **Initialization.** In the IJMSO algorithm, initial population is generated randomly as illustrated in Table 2.

– **Solution evaluation.** The solution is evaluated by calculating the summation of all communication cost between all experts in it as shown in Eqs. 2, 3. Table 3 shows an example of solution evaluation process. The solution with the overall minimum weight represents a gbest (the global best solution), while the solution

Table 2 An example of initial population

Solution_id	Solution
A	(a,c,e)
B	(d,e,a)
C	(d,c,b)
D	(a,c,b)

Table 3 Solution evaluation

Solution_id	Solution	f(x)	
A	(a,c,e)	0.57	gworst
B	(d,e,a)	0.4	
C	(d,c,b)	0.2	gbest
D	(a,c,b)	0.5	

with overall maximum weight represents a gworst (global worst solution).

– **Solution updating.** In each iteration, the solution is updated and computed according Eq. 4. In the above example and based on Table 3, “gbest” is solution C and “gworst” is solution A. In each iteration,

Fig. 5 Single-point crossover between the best solution and solutions A and B

the solution is updated and computed according to the update Eq. 4 as follows.

The main steps for updating **individual A** is represented as follows:

1. Consider $r_1 = 1$ and $r_2 = 0.7$
2. A single-point crossover applied as in Fig. 5a
3. The one with minimum communication cost is chosen as a result of crossover; in this case, $f(A_1) = 0.2$ and $f(A_2) = 0.5$. Therefore, the X_{cross}^t solution is $A_1 = (d, c, e)$
4. Compute the difference for both parts in Eq. 4 according to the MSO procedure
5. For “gbest,” $A_1 - A : \text{MSO}(1, 2, 0)$
6. For “gworst,” $A - A = 0$ (i.e., identical solutions)
7. Solution A is updated as follows $A = (a, c, e) \oplus (\text{MSO}(1, 2, 0)) = (a, c, e)$
8. Individual A(a, c, e) is updated to (a, c, e)
9. The communication cost of it is $f(A) = 0.57$

The main steps for updating **individual B** is represented as follows

1. Consider $r_1 = 1$ and $r_2 = 0.7$
2. A single-point crossover applied with the same procedure with an individual B as shown in Fig. 5b
3. The one with minimum communication cost is chosen as a result of crossover; in this case, $f(B_1) = 0.4$ and $f(B_2) = 0.4$. Therefore, the X_{cross}^t solution is $B_1(d, c, e)$
4. Compute the difference for both parts in Eq. 4 according to the MSO procedure
5. For “gbest” part $B_1 - B : \text{MSO}(2, 1, 0)$
6. For “gworst” part, $A - B = \text{MSO}(1, 0, 2), \text{MSO}(2, 0, 1), \text{MSO}(3, 2, 0)$. It means for $\text{skill}_{\text{id}} = 1$ exclude $\text{expert}_{\text{id}} = 2$ and replace it with another expert cho-

sen randomly. $A - B = \text{MSO}(1, 0, 1), \text{MSO}(2, 0, 0), \text{MSO}(3, 2, 1)$

7. Solution B is updated as follows: $B = (d, e, a) \oplus (\text{MSO}(2, 1, 0), \text{MSO}(1, 0, 1), \text{MSO}(2, 0, 0)) = (d, c, a) \oplus (\text{MSO}(1, 0, 1), \text{MSO}(2, 0, 0)) = (a, c, a) \oplus (\text{MSO}(2, 0, 0)) = (a, c, a)$
8. Individual B(d, e, a) is updated to (a, c, a)
9. The communication cost of it is $F(B) = 0.17$

The same procedure is applied for solution C and D.

According to that example, the next iteration for solution A is still the same, but for solution B changed from 0.4 to 0.17 and the same “gbest” can be updated according to the solution that has a minimum communication cost.

- **Termination criteria.** The overall steps are repeated until satisfied number of iterations which result the most feasible team is formed so far for required skills (i.e., the global best solution “gbest” so far).

5 Numerical experiments

In order to examine the efficiency and accuracy of the proposed IJMSO algorithm, a set of experiments were to reduce the communication cost among team members. The IJMSO algorithm was compared with the standard GA, PSO, ABO and standard Jaya. Also, we investigate the performance of the IJMSO algorithm on DBLP and StackExchange datasets. The experiments were implemented coding by Java, running on Intel(R) core i7 CPU- 2.80 GHz with 8 GB RAM and (Windows 10).

5.1 Parameter setting

The parameter setting of the IJMSO algorithm for all experiments is presented in Table 4. In Table 4, we test the proposed

Table 4 Parameters of IJMSO algorithm

Exp. no.	No. of iterations	No. of individuals in initial population	No. of skills
1	5	4	2
2	10	4	4
3	15	6	6
4	20	6	8
5	25	8	10

algorithm on 5 experiments with different number of iterations, population size and number of skills. The population size for all algorithm is the same to make a fair comparison. The probabilities of crossover (P_c) and the mutations (P_m) in GA are 0.6 and 0.01, respectively. The acceleration constants C_1 and C_2 in PSO algorithm are set to 2. The learning parameters lp_1 and lp_2 , in ABO algorithm, are a random number between 0 and 1. Also, the unit of time parameter λ is set to 1 for the balance between exploration and exploitation.

In the following subsection, we highlight two real-life datasets: DBLP and StackExchange.

5.2 DBLP dataset

In this work, we have used the DBLP dataset and building four tables from it as follows.

1. The first table is called (Author) with two attributes (name and paper_key) and it contains 6054672 records.
2. The second table is called (Citation) with two attributes (paper_cite_key and paper_cited_key) and it contains 79002 records.
3. The third table is called (Conference) with three attributes (conf_key, name and detail) and it contains 33953 records.
4. The last table is called (Paper) with four attributes (title, year, conference and paper_key) and it contains 1992157 records.

In DBLP, the extracted papers are published in year 2017 only (22364 records) and we construct the new dataset with five fields in computer science as follows: databases (DB), theory (T), data mining (DM), artificial intelligence (AI) and software engineering (SE).

We have applied the following steps to construct the DBLP graph as follows.

- The set of experts contains the authors with at least three published papers in DBLP. There are 77 authors that have published papers more than three.

- If there are two experts have sharing papers' skills, then they become connected and their communication cost is calculated as shown in Eq. 2.
- We have considered the most important shared skills between experts extracted from the title of 267 papers.

In our test, we consider the top ten conferences papers in computer science field with 1707 records. Five experiments were conducted and the average results are taken over 10 runs.

5.2.1 Comparison between IJMSO and other meta-heuristic algorithms with DBLP dataset

We test the performance of IJMSO algorithm by comparing it with four meta-heuristic algorithms, which are GA (Holland 1975), PSO (Eberhart et al. 2001), ABO (Odili and Kahar 2015) and the standard Jaya (Rao 2016) algorithms. The results of the five algorithms in terms of the communication cost are given in Table 5. The results in Table 5 show the best (min), worst (max), average (mean) and the standard deviation (St.d) over 10 random runs. We report the overall best result of the five algorithms in **bold face**. From Table 5, the IJMSO algorithm has achieved a least minimum and average communication cost in all experiments. Also, in Fig. 6, we plot the number of iterations versus the communication costs. The results of IJMSO algorithm are represented by solid line, while the other dotted lines represent the results of the other meta-heuristic algorithms. The results in Fig. 6 show that the communication cost of the proposed IJMSO algorithm are decreased faster than the other compared meta-heuristic algorithms. For example (at number of skills = 2), the IJMSO fitness value decreased by 11% at the end of iterations, while (at number of skills = 8) it decreased from 2% at the second iteration to 15% at the last iteration.

5.3 StackExchange dataset

We used another real-life dataset to investigate the proposed algorithm which is called the StackExchange dataset that has been obtained from Academia Stack Exchange (June 2017). The constructed tables are listed below as follows:

1. Posts (Id, PostTypeId, AcceptedAnswerId, CreationDate, Score, ViewCount, Body, OwnerUserId, LastEditorUserId, LastEditDate, LastActivityDate, Title, Tags, AnswerCount, CommentCount, FavoriteCount), 131200 records
2. Users (Id, Reputation, CreationDate, DisplayName, LastAccessDate, WebsiteUrl, Location, AboutMe, Views, UpVotes, DownVotes, Age, AccountId), 55301 records
3. Tags (WikiPostId, ExcerptPostId, Count, TagName, Id), 400 records.

Table 5 Comparison between IJMSO and other meta-heuristic algorithms with DBLP dataset

Exp. no.	No. of skills		GA	PSO	ABO	Jaya	IJMSO
1	2	Best	0.85	0.85	0.63	0.72	0.5
		Worst	0.96	0.97	0.93	0.96	0.93
		Mean	0.915	0.923	0.875	0.889	0.722
		St.d	0.0392	0.0333	0.0891	0.0707	0.1996
2	4	Best	5.05	5.19	5.07	4.88	4.57
		Worst	5.76	5.66	5.63	5.5	5.23
		Mean	5.383	5.405	5.25	5.163	4.805
		St.d	0.2440	0.1816	0.1594	0.1809	0.2445
3	6	Best	13.61	13.12	13	12.83	11.66
		Worst	14.23	14.21	13.74	13.72	12.95
		Mean	13.961	13.792	13.369	13.212	12.61
		St.d	0.2206	0.2912	0.2794	0.2947	0.3599
4	8	Best	25.66	25.32	24.82	23.53	22.22
		Worst	26.82	26.86	26.26	25.37	24.79
		Mean	26.045	25.946	25.475	24.692	23.552
		St.d	0.3636	0.5116	0.4512	0.5611	0.8486
5	10	Best	40.95	41.5	39.44	38.05	37.04
		Worst	42.79	42.11	42.11	40.94	40.5
		Mean	41.93	41.80167	40.86	39.635	38.732
		St.d	0.5284	0.2513	0.8419	0.9158	1.2010

4. PostLinks (Id, CreationDate, PostId, RelatedPostId, LinkTypeId), 9380 records.
5. PostHistory (Id, PostHistoryTypeId, PostId, RevisionGUID, CreationDate, UserID, Text), 180620 records.
6. Votes (CreationDate, VoteTypeId, PostId, Id), 703546 records.
7. Comments (Id, PostId, Score, Text, CreationDate, UserId), 158764 records.
8. Badges (TagBased, Class, Date, Name, UserId, Id), 116925 records.

We focus on extracting an expert set and skill set from the tables according to the following points:

- The expert set consists of users that have at least 10 posts (i.e., distinct tags) in academia.stackexchange (192 users)
- If there are two experts with share post' tags (skills), then they become connected. The communication cost e_{ij} of expert i and j is evaluated as shown in Eq. 2.
- We have considered the most important shared skills such as “publications,” “phd” and “conference” between experts extracted from the tags of distinct posts' title by users using StringTokenizer in Java.

5.3.1 Comparison between IJMSO and other meta-heuristic algorithms with StackExchange dataset

The second test of the IJMSO algorithm is applied by comparing it against five meta-heuristic algorithm on StackExchange dataset. In Table 6, we compare the IJMSO algorithm against the GA, PSO, ABO and the standard Jaya algorithm. In Table 6, we report the best (min), worst (max), average (mean) and the standard deviation (St.d) results over 10 random runs. We report the best result of the five algorithms in **bold font**. The results in Table 6 show that the proposed IJMSO algorithm has achieved the least communication cost.

In Fig. 7, we plot the number of iterations versus the communication costs. The results of IJMSO algorithm are represented by solid line, while the other dotted lines represent the results of the other meta-heuristic algorithms. The results in Fig. 7 show that increasing the iteration number will decrease the values of communication cost of the IJMSO algorithm faster than the other compared meta-heuristic algorithms. For example (at number of skills = 4), the IJMSO fitness value is minimized by 17% at the end of iterations, while with more number of iterations, the minimization in communication cost ranged from 1% to 14% throughout different experiments.

Fig. 6 Comparison between IJMSO and other meta-heuristic algorithms with DBLP dataset

5.4 The confidence interval (CI) test

A confidence interval (CI) test is used to measure the probability that a population parameter will fall between upper and lower bounds. It formed at a confidence level (C) such as 95%. The 95% confidence interval is using data mean and standard deviation values by assuming a normal distribution. It can be defined as shown in Eq. 5:

$$CI = \text{mean} \pm \text{confidence} \quad (5)$$

A confidence can be computed according to the following three parameters (γ , St.d and pattern size), where γ are calculated based on the confidence level (i.e., $\gamma = 1 - C$), St.d is the standard deviation of the pattern and pattern

size is the size of population. The value of 95% CI means $\gamma = (1 - 0.95) = 0.05$, and CI is used to approximate the mean of the population.

The performance (%) of the compared algorithms can be calculated as shown in Eq. 6:

$$\text{Performance}(\%) = \frac{(\text{Avg}_{(GA, PSO, ABO, Jaya)} - \text{Avg}_{(IJMSO)})}{\text{Avg}_{(GA, PSO, ABO, Jaya)}} \quad (6)$$

where $\text{Avg}_{(GA, PSO, ABO, Jaya)}$ and $\text{Avg}_{(IJMSO)}$ are the average results obtained from the standard GA, PSO, ABO, Jaya and IJMSO algorithms, respectively.

Table 6 Comparison between IJMSO and other meta-heuristic algorithm with StackExchange dataset

Exp. no.	No. of skills		GA	PSO	ABO	Jaya	IJMSO
1	2	Best	0.81	0.81	0.81	0.78	0.77
		Worst	0.88	0.86	0.85	0.86	0.82
		Mean	0.83	0.84	0.82	0.82	0.79
		St.d	0.0263	0.0163	0.0125	0.0231	0.0164
2	4	Best	4.3	4.25	4.28	4.24	4.06
		Worst	5.14	5.2	5.16	5.07	4.59
		Mean	4.83	4.87	4.74	4.53	4.28
		St.d	0.3673	0.3417	0.3695	0.2975	0.1477
3	6	Best	12.11	12.05	11.53	11.40	10.50
		Worst	13.12	13.18	12.80	12.32	11.79
		Mean	12.759	12.76	12.05	11.95	11.35
		St.d	0.3565	0.3909	0.3370	0.2666	0.4161
4	8	Best	21.89	23	20.45	20.67	19.43
		Worst	24.24	24.25	23.19	23.09	22.63
		Mean	23.31	23.68	22.23	22.08	21.54
		St.d	0.6262	0.4273	0.9435	0.7386	1.0354
5	10	Best	36.92	36.57	35.98	35.38	33.69
		Worst	42	42.11	42.11	36.84	38.22
		Mean	38.40	38.55	37.22	36.144	35.48
		St.d	1.3947	1.8753	1.8337	0.5002	1.1868

5.4.1 Confidence interval (CI) of IJMSO and the other meta-heuristic algorithm for DBLP dataset

The confidence interval (CI) of IJMSO and other meta-heuristic algorithms for DBLP dataset is shown in Table 7. In Table 7, the 95% confidence interval on average communication cost of the proposed IJMSO algorithm has achieved better results minimized from 21% (at number of skills=2) to 8% (at number of skills=10) when compared with GA and results minimized from 22% (at number of skills=2) to 7% (at number of skills=10) when compared with PSO. When compared with ABO, the proposed improved algorithm beats it within range minimized from 17% (at number of skills=2) to 5% (at number of skills=10). Although the Jaya algorithm has better results than GA and PSO within range 3% - 5% for all the experiments and reached up to 3% better than ABO for all experiments except at number of skills equals 2, the ABO is better than Jaya by 2%. In general, the IJMSO has achieved results minimized from 19% (at number of skills=2) to 2% (at number of skills=10) when compared with standard Jaya algorithm.

5.4.2 Confidence interval (CI) of IJMSO and other meta-heuristic algorithm for StackExchange dataset

Another test of CI for the IJMSO and other meta-heuristic algorithms for StackExchange dataset are shown in Table 8. In Table 8, the 95% confidence interval on average commu-

nication cost of the proposed IJMSO algorithm has achieved better results minimized ranged from 5% (at number of skills=2) to 8% (at number of skills=10) when compared with GA and PSO. When compared with ABO, the proposed IJMSO algorithm beats it within range 3 - 10% at different number of skills. Although the Jaya algorithm has better results than GA, PSO and ABO reached to 7%, the IJMSO has achieved results minimized up to 4% (at number of skills=2), 6% (at number of skills=4), 5% (at number of skills=6) and 2% (at number of skills equal 8 and 10) when compared against the standard Jaya algorithm.

5.5 The running time of IJMSO and the other algorithms

The aim of this paper is minimizing the average communication cost not minimizing the running time. Therefore, the running time has not a significant impact in this work; in particular, we considered a TFP is an assignment problem, not a constrained assignment problem such as scheduling problem. Tables 9 and 10 show the best (min), worst (max) and the mean (average) of the running time (in seconds) for the five algorithms for forming a feasible team in each experiment. In general, the average running time increased with more iterations.

In Table 9, the average running time is different in each experiment. For the number of skills equals 2, the proposed algorithm has obtained a better time than ABO with 1%. For

Fig. 7 Comparison between IJMSO and other meta-heuristic algorithms with StackExchange dataset

Table 7 CI for IJMSO and the other algorithms with DBLP dataset

No. of skills	GA	PSO	ABO	Jaya	IJMSO
2	0.915 ± 0.0243	0.923 ± 0.0206	0.875 ± 0.0552	0.889 ± 0.0438	0.722 ± 0.1237
4	5.383 ± 0.1512	5.405 ± 0.1125	5.25 ± 0.0988	5.163 ± 0.1121	4.805 ± 0.1516
6	13.961 ± 0.1367	13.792 ± 0.1805	13.369 ± 0.1731	13.212 ± 0.1826	12.61 ± 0.2230
8	26.045 ± 0.2254	25.946 ± 0.3171	25.475 ± 0.2796	24.692 ± 0.3477	23.552 ± 0.5259
10	41.93 ± 0.3275	41.80167 ± 0.1557	40.86 ± 0.5218	39.635 ± 0.5676	38.732 ± 0.7444

Table 8 CI for IJMSO and the other algorithms with StackExchange dataset

No. of skills	GA	PSO	ABO	Jaya	IJMSO
2	0.836 ± 0.0163	0.84 ± 0.0101	0.823 ± 0.0077	0.827 ± 0.0143	0.794 ± 0.0102
4	4.835 ± 0.2276	4.873 ± 0.2118	4.742 ± 0.2290	4.530 ± 0.1843	4.280 ± 0.0915
6	12.759 ± 0.2209	12.763 ± 0.2423	12.058 ± 0.2089	11.959 ± 0.1652	11.347 ± 0.2579
8	23.307 ± 0.3881	23.687 ± 0.2648	22.237 ± 0.5848	22.081 ± 0.4578	21.545 ± 0.6417
10	38.407 ± 0.8644	38.55 ± 1.1623	37.226 ± 1.1365	36.144 ± 0.3100	35.484 ± 0.7355

Table 9 Running time (in second) of IJMSO and the other algorithms with DBLP dataset

Exp. no.	No. of skills		GA	PSO	ABO	Jaya	IJMSO
1	2	Best	0.22	0.56	0.25	0.60	0.56
		Worst	0.42	0.98	0.51	0.80	1.13
		Mean	0.29	0.74	0.34	0.70344	0.69
2	4	Best	1.40	2	1.40	2.10	2.20
		Worst	1.90	4.8	1.80	2.8	2.50
		Mean	1.55	2.62	1.56	2.31	2.35
3	6	Best	9.13	10.53	8.93	10.80	10.73
		Worst	15.66	16.40	14.46	14.2	13.46
		Mean	11.01	12.34	10.46	12.20	11.76
4	8	Best	21.88	24.56	21.88	24.85	24.70
		Worst	27.29	29.77	25.66	33.93	33.50
		Mean	23.88	26.64	23.76	27.75	26.91
5	10	Best	57.96	59.80	56.28	60.65	58.60
		Worst	64.24	69.36	64.15	70.24	71.77
		Mean	61.30	63.39	60.54	64.09	63.67

10 skills, the running time of IJMSO is minimized by 11% when compared with PSO. Although the proposed algorithm does not beat all the compared algorithms in the average running time in different number of skills, it beats them in terms of minimizing the communication cost during the number of iterations for solving the team formation problem. More running time of the proposed algorithm is due to the improvements (e.g., the crossover operator and the modified swap operators).

In Table 10, the average running time is different in each experiment. For the number of skills equals 2, the proposed algorithm has obtained a better time that ABO with 1%. For 10 skills, the running time of IJMSO minimized by 11% when compared with PSO. Although the proposed algorithm does not beat all the compared algorithms in the average running time in different number of skills, it beats them in terms of minimizing the communication cost during the number of iterations for solving the team formation problem. More running time of the proposed algorithm is due to the improvements (e.g., the crossover operator and the modified swap operators).

6 Conclusion and future work

In team formation problem, a group of experts connected with their skills to perform a specific task. In this paper, we propose a new meta-heuristic algorithm which is called Jaya algorithm by using a single-point crossover and invoking a modified swap operator to accelerate the search of it. The proposed algorithm is called an improved Jaya algorithm with a modified swap operator (IJMSO). In IJMSO, we present a modified swap operator $MSO(a,b,c)$, where a is the $skill_{id}$ and b and c are the indices of experts that have the skill from experts' list. The performance of IJMSO algorithm was tested on two real-life dataset (DBLP and StackExchange) and compared against four meta-heuristic algorithms (GA, PSO, ABO and Jaya). The obtained results of IJMSO algorithm show that it was faster than the other algorithms. In the future work, we will increase the number of the real-life datasets and we will combine the IJMSO algorithm with more new swarm intelligence algorithms (SI) to improve the performance of it.

Table 10 Running time (in second) of IJMSO and the other algorithms with StackExchange dataset

Exp. no.	No. of skills		GA	PSO	ABO	Jaya	IJMSO
1	2	Best	1.072	1.53	1.07	1.46	1.08
		Worst	3.45	3.44	3.52	3.39	3.73
		Mean	1.47	1.85	1.94	1.87	1.91
2	4	Best	11.33	11.46	11.15	11.74	10.87
		Worst	34.80	35.48	37.20	36.89	37.87
		Mean	14.59	14.99	17.54	17.40	17.81
3	6	Best	56.84	58.85	56.98	58.04	58.01
		Worst	65.77	174.62	177.69	73.84	184.40
		Mean	59.22	72.81	71.50	61.92	96.54
4	8	Best	135.83	141.86	135.85	140.76	141.51
		Worst	148.00	157.24	449.11	465.46	437.40
		Mean	142.24	146.38	175.39	204.76	230.49
5	10	Best	358.77	369.32	361.53	359.27	389.24
		Worst	396.02	1136.48	382.75	1124.89	1094.82
		Mean	383.10	534.88	374.79	459.04	472.21

Acknowledgements This study was not funded by any grant.

Compliance with ethical standards

Conflict of interest Author Walaa H. El-Ashmawi declares that she has no conflict of interest. Author Ahmed F. Ali declares that he has no conflict of interest. Author Adam Slowik declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Anagnostopoulos A, Becchetti L, Castillo C, Gionis A, Leonardi S (2012) Online team formation in social networks. In: Proceedings of the 21st international conference on World Wide Web, pp 839–848
- Appel AP, Cavalcante VF, Vieira MR, de Santana VF, de Paula RA, Tsukamoto SK (2014) Building socially connected skilled teams to accomplish complex tasks. In: Proceedings of the 8th workshop on social network mining and analysis. Article no. 8
- Awal GK, Bharadwaj KK (2014) Team formation in social networks based on collective intelligence—an evolutionary approach. *Appl Intell* 41(2):627–648
- Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv (CSUR)* 35(3):268–308
- Dede T (2018) Jaya algorithm to solve single objective size optimization problem for steel grillage structures. *Steel Compos Struct* 26(2):163–170
- Eberhart RC, Shi Y, Kennedy J (2001) Swarm intelligence (The morgan kaufmann series in evolutionary computation). Morgan Kaufmann Publishers, Burlington
- Farasat F, Nikolaev AG (2016) Social structure optimization in team formation. *Comput Oper Res* 74:127–142
- Fathian M, Makui Saei-Shahi MA (2017) A new optimization model for reliable team formation problem considering experts' collaboration network. *IEEE Trans Eng Manag* 64(4):586–593
- Grzywinski M, Dede T, Ozdemir YI (2019) Optimization of the braced dome structures by using Jaya algorithm with frequency constraints. *Steel Compos Struct* 30(1):47–55
- Gutiérrez JH, Astudillo CA, Ballesteros-Pérez P, Mora-Meliá D, Candia-Véjar A (2016) The multiple team formation problem using sociometry. *Comput Oper Res* 75:150–162
- Han Y, Wan Y, Chen L, Xu G, Wu J (2017) Exploiting geographical location for team formation in social coding sites In: Proceedings of Pacific-Asia conference on knowledge discovery and data mining, (LNAI), vol 10234, Springer, Heidelberg, pp 499–510
- Haupt RL, Haupt SE (2004) Practical genetic algorithms. Wiley, New York
- Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
- Huang J, Sun X, Zhou Y, Sun H (2017) A team formation model with personnel work hours and project workload quantified. *Comput J* 60(9):1382–1394
- Kargar M, An A (2011) Discovering top-k teams of experts with/without a leader in social networks. In: Proceedings of the 20th ACM international conference on information and Knowledge Management, pp 985–994
- Kargar M, Zihayat M, An A (2013) Finding affordable and collaborative teams from a network of experts. In: Proceedings of the 2013 SIAM international conference on data mining, pp 587–595

- Lappas T, Liu K, Terzi E (2009) Finding a team of experts in social networks. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 467–476
- Li CT, Shan MK (2010) Team formation for generalized tasks in expertise social networks. In: Proceedings of IEEE second international conference on social computing (SocialCom), pp 9–16
- Li CT, Shan MK, Lin SD (2015) On team formation with expertise query in collaborative social networks. *Knowl Inf Syst* 42(2):441–463
- Nadershahi M, Moghaddam RT (2012) An application of genetic algorithm methods for team formation on the basis of Belbin team role. *Arch Appl Sci Res* 4(6):2488–2496
- Odili JB, Kahar MNM (2015) African buffalo optimization (ABO): a new meta-heuristic algorithm. *J Adv Appl Sci* 3(03):101–106
- Pandey HM (2016) Jaya a novel optimization algorithm: What, how and why?. In: Proceedings of 6th international conference-cloud system and big data engineering (Confluence), pp 728–730
- Pashaei K, Taghiyareh F, Badie K (2015) A recursive genetic framework for evolutionary decision-making in problems with high dynamism. *Int J Syst Sci* 46(15):2715–2731
- Rao RV, Rai DP, Balic J (2016) Surface grinding process optimization using Jaya algorithm. In: computational intelligence in data mining, vol 2, Springer, Berlin, pp 487–495
- Rao RV (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int J Ind Eng Comput* 7(1):19–34
- Rao RV, More KC (2017) Design optimization and analysis of selected thermal devices using self-adaptive Jaya algorithm. *Energy Convers Manag* 140:24–35
- Rao RV, More KC (2017) Optimal design and analysis of mechanical draft cooling tower using improved Jaya algorithm. *Int J Refrig* 82:312–324
- Rao RV, Rai DP (2017) Optimization of submerged arc welding process parameters using quasi-oppositional based Jaya algorithm. *J Mech Sci Technol* 31(5):2513–2522
- Rao RV, Rai DP (2017) Optimisation of welding processes using quasi-oppositional-based Jaya algorithm. *J Exp Theor Artif Intell* 29(5):1099–1117
- Rao RV, Saroj A (2017) A self-adaptive multi-population based Jaya algorithm for engineering optimization. *Swarm Evolut Comput* 37:1–26
- Rao RV, Saroj A (2017) Constrained economic optimization of shell-and-tube heat exchangers using elitist-Jaya algorithm. *Energy* 128:785–800
- Rao RV, Saroj A (2017) Economic optimization of shell-and-tube heat exchanger using Jaya algorithm with maintenance consideration. *Appl Therm Eng* 116:473–487
- Rao RV, Waghmare GG (2017) A new optimization algorithm for solving complex constrained design optimization problems. *Eng Optim* 49(1):60–83
- Rao RV, More KC, Taler J, Oclon P (2016) Dimensional optimization of a micro-channel heat sink using Jaya algorithm. *Appl Therm Eng* 103:572–582
- Rao RV, Rai DP, Ramkumar J, Balic J (2016) A new multi-objective Jaya algorithm for optimization of modern machining processes. *Adv Prod Eng Manag* 11(4):271–286
- Sedighizadeh D, Masehian E (2009) Particle swarm optimization methods, taxonomy and applications. *Int J Comput Theory Eng* 1(5):486–502
- Trivedi NI, Purohit SN, Jangir P, Bhoje MT (2016) Environment dispatch of distributed energy resources in a microgrid using Jaya algorithm. In: Proceedings of 2nd international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB), pp 224–228
- Wang KP, Huang L, Zhou CG, Pang W (2003) Particle swarm optimization for traveling salesman problem. In: Proceedings of international conference on machine learning and cybernetics, vol 3, pp 1583–1585
- Wang S, Rao RV, Chen P, Zhang Y, Liu A, Wei L (2017) Abnormal breast detection in mammogram images by feed-forward neural network trained by Jaya algorithm. *Fundam Inform* 151(1–4):191–211
- Wei X, Jiang-wei Z, Hon-lin Z (2009) Enhanced self-tentative particle swarm optimization algorithm for TSP. *J North China Electr Power Univ* 36(6):69–74
- Zhang JW, Si WJ (2010) Improved enhanced self-tentative PSO algorithm for TSP. In: Proceedings of sixth international conference on natural computation (ICNC), vol 5, pp 2638–2641
- Zhang Y, Yang X, Cattani C, Rao RV, Wang S, Phillips P (2016) Tea category identification using a novel fractional Fourier entropy and Jaya algorithm. *Entropy*, vol 18(3), Article no. 77

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.