Journal of Industrial Engineering International (2019) 15:53–71
https://doi.org/10.1007/s40092-018-0282-6

ORIGINAL RESEARCH

An improved particle swarm optimization with a new swap operator for team formation problem

Walaa H. El-Ashmawi[1] · Ahmed F. Ali[1] · Mohamed A. Tawhid[2]

Received: 17 November 2017 / Accepted: 11 July 2018 / Published online: 28 July 2018

Abstract

Formation of effective teams of experts has played a crucial role in successful projects especially in social networks. In this paper, a new particle swarm optimization (PSO) algorithm is proposed for solving a team formation optimization problem by minimizing the communication cost among experts. The proposed algorithm is called by improved particle optimization with new swap operator (IPSONSO). In IPSONSO, a new swap operator is applied within particle swarm optimization to ensure the consistency of the capabilities and the skills to perform the required project. Also, the proposed algorithm is investigated by applying it on ten different experiments with different numbers of experts and skills; then, IPSONSO is applied on DBLP dataset, which is an example for benchmark real-life database. Moreover, the proposed algorithm is compared with the standard PSO to verify its efficiency and the effectiveness and practicality of the proposed algorithm are shown in our results.

Keywords Particle swarm optimization · Team formation problem · Social networks · Single-point crossover · Swap operator

Introduction

The team formation (TF) problem plays a crucial role in many real-life applications ranging from software project development to various participatory tasks in social networks. In such applications, collaboration among experts is required. There are a number of experts associated with their capabilities (i.e., skills) and a collaborative task (i.e., project) that requires set of skills needed to be accomplished. The problem is how to find the effective team of experts that covers all the required skills for a given task with least communication cost. It is known that this problem is NP-hard problem (Lappas et al. 2009); hence, it will be interesting to develop heuristic search methods to solve it.

It is well known that the swarm-based algorithms such as particle swarm optimization (PSO) (Vallade and Nakashima 2013) are capable of reaching solutions quickly and efficiently because they have the ability to generate different outputs from the same sample inputs. It is a heuristic method that based on execution of various alternative solutions via iterations to find the best solution. Another adaptive heuristic method is genetic algorithm (GA) (Holland 1975; Kalita et al. 2017). It is based on the natural law of evolution through the natural selection and the exchange of genetic information. Generally speaking, the goal of optimization methods is to find adequate incorporation of a set of parameters to achieve the most satisfaction (e.g., minimum or maximum) that depends on the requirement of the problem.

Therefore, the main objective of this research is to form the effective team of experts with minimum communication cost by using a hybrid improved PSO with a new swap operator and the main operator of GA (i.e., crossover

✉ Mohamed A. Tawhid
mtawhid@tru.ca

Walaa H. El-Ashmawi
w.hashmawi@ci.suez.edu.eg

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

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

2 Department of Mathematics and Statistics, Faculty of Science, Thompson Rivers University, Kamloops, BC V2C 0C8, Canada

operator). We call the proposed algorithm an improved particle optimization with a new swap operator (IPSONSO).

The problem in Karduck and Sienou (2004) is defined as the process anterior to the forming stage of the group development theory. The key problem is the selection of best candidates that fulfills the requirement specification for achieving the goal. Most of existing team formation based on approximation algorithms (Anagnostopoulos et al. 2012; Kargar et al. 2013) considers different communication costs such as diameter and minimum spanning tree (Lappas et al. 2009) and sum of distance from team leader (Kargar and An 2011).

A generalization of the team formation problem is given (Appel et al. 2014; Li and Shan 2010; Li et al. 2015) by assigning each skill to a specific number of experts. Consideration of the maximum load of experts according to different tasks is taken from Anagnostopoulos et al. (2010) without taking into consideration the minimum communication cost for team formation.

On the other side of team formation problem, a minimal research work has been done based on meta-heuristic algorithms such as PSO and GA (Haupt and Haupt 2004). These algorithms have been successfully applied in an optimization method as in Blum and Roli (2003), Pashaei et al. (2015), Sedighizadeh and Masehian (2009) for many real-world applications.

A group formation method using genetic algorithm is presented in Zhang and Si (2010), where the members for each group are generated based on the students' programming skill. A genetic algorithm in team formation is used in Nadershahi and Moghaddam (2012) based on Belbin team role that categorized individuals in nine roles regarding their specialty and attitude toward team working.

A team formation problem is presented in Gutiérrez et al. (2016) based on sociometric matrix in which a mathematical programming model for maximizing the efficiency understood relationships among people who share a multidisciplinary work cell is considered. A variable neighborhood local search meta-heuristic is applied in Gutiérrez et al. (2016) to solve team formation problem and showed the most efficient in almost all cases, but in our work, the global search meta-heuristic considered with least minimum communication cost among all the locals is the most efficient all over the search.

A team formation is considered in Huang et al. (2017) based on the available work time and set of skills for each expert in order to build the effective team. Each expert is associated with a skill level indicating his competence in this skill. In our research, all experts that have the ability to perform the skill are attentive to share in a collaborative group in order to achieve the goal.

A mathematical framework for dealing the team formation problem is proposed in Farasat and Nikolaev (2016) explicitly incorporating social structure among experts where a LK-TFP heuristic is used to perform variable-depth neighborhood search and compared the results with standard genetic algorithm. In our paper by given a pool of individuals, an improved PSO algorithm for team formation problem is proposed and compared the results with standard PSO.

Finally, in Fathian et al. (2017) a mathematical model is proposed to maximize team reliability by considering the probability of unreliable experts that may leave the team with a probability and prepare a backup for each unreliable one. In that case, for each team, associated team members in the two sets, namely, main and backup members, should be presented and is effective only in some specific situations. In contrast to our research, among all the available team members, the most feasible one is chosen in the team formation that has no incentive to leave the team.

The rest of the paper is organized as follows. Section 2 illustrates the definition of team formation problem. Section 3 introduces the formulation of proposed algorithm and how it works. Section 4 discusses the experimental results of the proposed algorithm. Finally, Sect. 5 concludes the work and highlights the future work.

Team formation problem

The team formation problem in social network can be formulated as finding a set of experts from a social network graph $G(V, E)$ to accomplish a given task (i.e., project) in which a number of experts n exist such that, $V = \{v_1, v_2, \ldots, v_n\}$ and a set of m skills $S = \{s_1, s_2, \ldots, s_m\}$, which represent their abilities to a given task. Each expert v_i is associated with a set of specific skills $s(v_i), s(v_i) \subset S$. The set of experts that have the skill s_k is denoted as $C(s_k)$, (i.e., $C(s_k) \subset V$). A given task T is formed by a set of required skills (i.e., $T = \{s_i, \ldots, s_j\} \subseteq S$) that can be applied by a set experts forming a team. A set of possible teams that can achieve a given task is denoted as $X, X = x_1, x_2, \ldots, x_k$. Therefore, the task ($T \subseteq \bigcup_{v_i \in x_k} s(v_i)$). The collaboration cost (i.e., communication cost) between any two experts (e.g., v_i and v_j) is denoted by $e_{ij} \in E$ that can be computed according to Eq. 1.

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

The goal is to find a team with least communication cost among team members $CC(x_k)$ according to Eq. 2.

$$CC(x_k) = \sum_{i=1}^{|x_k|} \sum_{j=i+1}^{|x_k|} e_{ij} \quad (2)$$

where $|x_k|$ is the cardinality of team x_k.

The team formation problem can be considered as an optimization problem by forming a feasible team x^* among a set of possible teams which covers the required skills for a given task with minimum communication cost among team's experts, and x^* can be obtained by the following

$$\text{Min}_{(x_i \in X)} CC(x_i) = \sum_{l=1}^{|x_i|} \sum_{j=l+1}^{|x_i|} e_{ij} \quad (3)$$

subjectto

$$\forall v_i, v_j : e_{ij} \in [0, 1] \quad (4)$$
$$\forall s_i \in T, \exists C(s_i) \geq 1$$

where the communication cost between any pair of experts within the range 0 and 1 and for each required skill in the given task, there exists at least one expert that have the required skill. All the skills should be achieved for a given task to obtain a feasible team x^*.

The notations of the team formation problem are summarized in Table 1.

Remark Set covering problem is one of the traditional problems in complexity theory and computer science. Set covering problem is regarded as one of the most important discrete optimization problems because it can be formulated as a model for various real-life problems, e.g., vehicle routing, resource allocation, nurse scheduling problem, airline crew scheduling, facility location problem. The name of problem, set covering problem, arises from covering the rows of an m-row/n-column zero-one matrix with a subset of columns at minimal cost set (Beasley and Chu 1996). Covering problem can be modeled as follows:

$$\text{Min} \sum_{j=1}^{n} c_j x_j \quad (5)$$

Table 1 Notations of team formation problem

Notation	Definition
V	A set of experts
$G(V, E)$	Experts social network
S	Set of skills
T	A task with required skills
X	Experts team
$C(s_i)$	A set of experts skilled in s_i
$s(v_i)$	Skill of expert v_i
e_{ij}	Communication cost between experts v_i and v_j

subjectto

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad j = 1, \ldots, m \quad (6)$$

$$\forall x_j \in (0, 1) \qquad j = 1, \ldots, n \quad (7)$$

Equation (5) is the objective function of set covering problem, where x_j is decision variable and c_j denotes to weight or cost of covering j column. Equation (6) is a constraint to assure that each row is covered by at least one column where a_{ij} is constraint coefficient matrix of size $m \times n$ whose elements consist of either "1" or "0." Also, Eq. (7) is the integrality constraint in which the value is expressed as follows

$$x_j = \begin{cases} 1, & \text{if} j \in S; \\ 0, & \text{otherwise.} \end{cases}$$

Despite the fact that it may look to be an easy problem from the objective functions and constraints of the problem, set covering problem is a combinational optimization problem and NP-complete decision problem (Lappas et al. 2009).

As mentioned in the literature, e.g., Kargar and An (2011), team formation problem is a special instance of the minimum set cover problem.

An example of the team formation problem

We describe an example of the team formation problem in Fig. 1.

In Fig. 1, a network of experts $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ is considered where each expert has a set of skills S and there is a communication cost between every two adjacent experts v_i, v_j, which is represented as a weight of edge (v_i, v_j) (e.g., $w(v_1, v_2) = 0.2$). The communication cost between non-adjacent experts is represented by the shortest path between them.

The aim is to find team X of experts V with the required skills S with a minimum communication cost. In Fig. 1, two teams with the required skills $X_1 = \{v_1, v_2, v_3, v_4\}$ and $X_2 = \{v_2, v_4, v_5, v_6\}$ are obtained.

The proposed algorithm

In the following subsections, the main processes of the standard particle swarm optimization (PSO), single-point crossover, and the improved swap operator are highlighted and invoking them in the proposed algorithm is described.

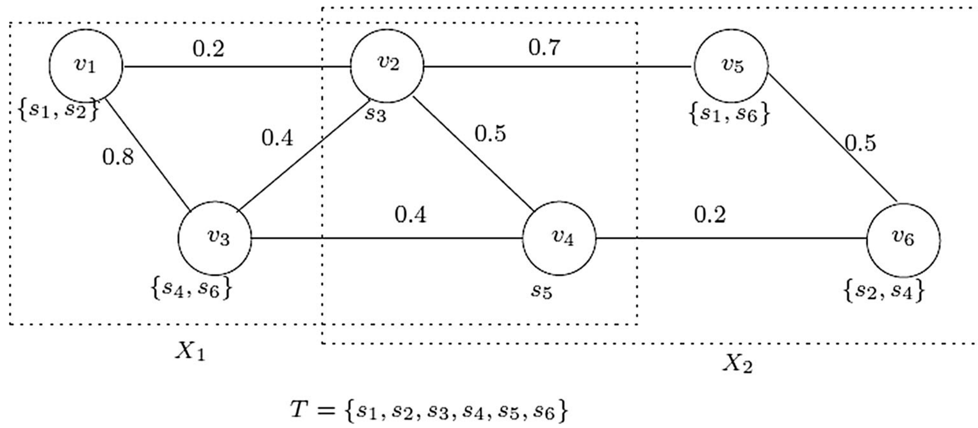


Fig. 1 An example of team formation problem

Particle swarm optimization

Particle swarm optimization (PSO) is a population-based meta-heuristic method developed by Kennedy and Eberhart in 1995 (Eberhart et al. 2001). The main process of the PSO is shown in Fig. 2. The PSO population is called swarm *SW*, the swarm contains particles (individuals), and each particle is represented by *n*-dimensional vectors as shown in Eq. 8

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in SW. \tag{8}$$

Each particle has a velocity, which is generated randomly as shown in Eq. 9.

$$\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{in}). \tag{9}$$

The best personal (P_{best}) and global positions (g_{best}) of each particle are assigned according to Eq. 10.

$$\mathbf{p}_i = (p_{i1}, p_{i2}, \ldots, p_{in}) \in SW. \tag{10}$$

At each iteration, each particle updates its personal position (P_{best}) and the global position (g_{best}) among particles in the neighborhood as shown in Eqs. 11 and 12, respectively.

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)}, \quad i = \{1, \ldots, SW\} \tag{11}$$

$$\mathbf{v}_i^{(t+1)} = \mathbf{v}_i^{(t)} + c_1 r_{i1} \times (\mathbf{p}_{best_i}^{(t)} - \mathbf{x}_i^{(t)}) + c_2 r_{i2} \times (\mathbf{g}_{best} - \mathbf{x}_i^{(t)}). \tag{12}$$

where c_1 and c_2 are the cognitive and social parameters, respectively. r_1 and r_2 are random vector $\in [0, 1]$. The process are repeated till termination criteria are satisfied.

Single-point crossover

Crossover is the one of the most important operators in GA. It creates one or more offspring from the selected parents. The single-point crossover (Goldberg 1989) is one of the most used operators in GA. The process starts by selecting a random point *k* in the parents between the first gene and the last gene. The two parents are swamping all the genes between the point *k* and the last gene. The process of the single-point crossover is shown Fig. 3.

A new swap operator

A swap operator (SO) in Wang et al. (2003), Wei et al. (2009) an Zhang and Si (2010) consists of two indices *SO*(*a*, *b*), which applied on the current solution to make a new solution. For example, if we have a solution $S = (1-2-3-4-5)$, $\mathrm{SO} = (2, 3)$; then, the new solution $S' = S + \mathrm{SO}(2,3) = (1-2-3-4-5) + \mathrm{SO}(2,3) = (1-3-2-4-5)$. A collection of one or more swap operators *SO*(*s*), which can apply sequentially, is called swap sequence (SS). SS applies on a solution by maintaining all its $\mathrm{SS} = (\mathrm{SO}_1, \mathrm{SO}_2, \ldots, \mathrm{SO}_n)$ to produce a final solution.

In our proposed algorithm, the proposed swap operator $\mathrm{NSO}(a, b, c)$ contains three indices: the first one argument *a* is the $skill_{id}$, and the second and the third arguments *b*, *c* are the current and the new experts' indices, respectively, which are selected randomly and they have the same $skill_{id}$ where $b \neq c$. For example, $\mathrm{NSO}(2, 1, 3)$ means for $skill_{id} = 2$ there is a swap between the $expert_{id} = 1$ and $expert_{id} = 3$.

Improved Particle Swarm Optimization with New Swap Operator (IPSONSO)

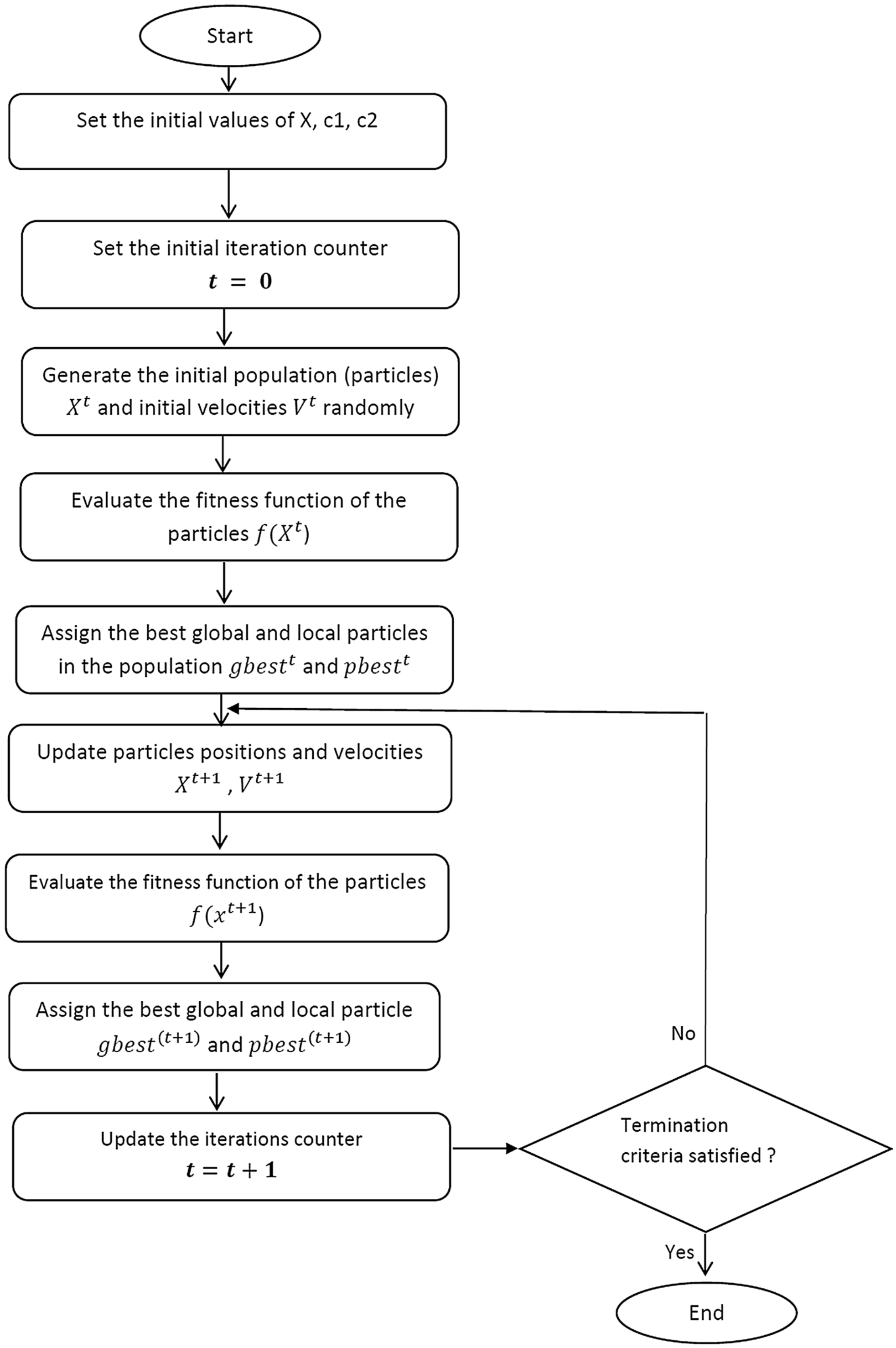
In this subsection, the main structure of the proposed IPSONSO is explained and shown in Algorithm 1.

In the following subsections, the proposed IPSONSO is applied and explained how to solve team formation problem.

Initialization and representation

IPSONSO starts by setting the initial values of its main parameters such as the population size *P*, social and

Fig. 2 Particle swarm operator processes



cognitive coefficients c_1, c_2 and the maximum number of iterations $\max_{itr}$. Given a project Pr contains a set of skills s_i, $i = \{1, 2, \ldots, d\}$, where d is the number of the requested skills in the project. IPSONSO initializes the positions and the velocities of all particles randomly, where each particle represents a vector of random skills to form the project and the velocity is a sequence of random swap operators, that represented by a new swap operator $\mathrm{NSO}(x, y, z)$, where x

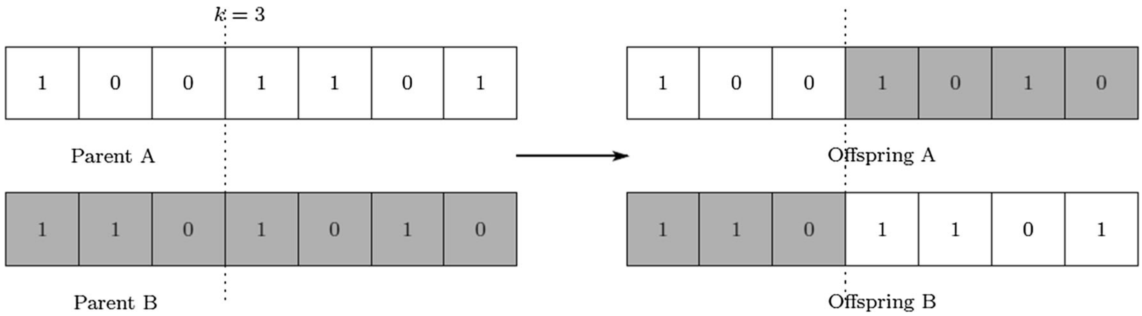


Fig. 3 Single-point crossover

Algorithm 1 Improved Particle Swarm Optimization with New Swap Operator (IPSONSO)

Set the initial values of swarm size SW.
Set $t := 0$.
Generate randomly $\mathbf{x}_i^{(t)}$, $\mathbf{v}_i^{(t)}$, $i = 1, \ldots, SW$ { where SW is the population (swarm) size}.
Evaluate the fitness function $f(\mathbf{x}_i^{(t)})$.
Set $\mathbf{g}_{best}^{(t)}$ {the best global solution in the swarm}.
Set $\mathbf{p}_{best_i}^{(t)}$ {the best personal solution in the swarm}.
repeat
  for $(i = 0; i < SW; i{+}{+})$ **do**
    $\mathbf{v}_i^{(t+1)} = \mathbf{v}_i^{(t)} \oplus \alpha \otimes (\mathbf{p}_{best_i}^{(t)} - \mathbf{x}_i^{(t)}) \oplus \beta \otimes (\mathbf{x}_{cross}^{(t)} - \mathbf{x}_i^{(t)})$
    $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)}, \quad i = \{1, \ldots, SW\}$ {Update particles positions}.
  end for
  for $(i = 0; i < SW; i{+}{+})$ **do**
    Apply crossover to $\mathbf{g}_{best}$ and current solution $\mathbf{x}_i^{(t+1)}$ {single point crossover}.
    Update $\mathbf{x}_i^{(t+1)}$ according to Equations 13, 14.
  end for
  if $f(\mathbf{x}_i^{(t+1)}) \leq f(\mathbf{p}_{best_i}^{(t)})$ **then**
    $\mathbf{p}_{best_i}^{(t+1)} = \mathbf{x}_i^{(t+1)}$. {Minimization problem}.
  else
    $\mathbf{p}_{best_i}^{(t+1)} = \mathbf{p}_{best_i}^{(t)}$.
  end if
  if $\mathbf{x}_i^{(t+1)} \leq f(\mathbf{g}_{best}^{(t)})$ **then**
    $\mathbf{g}_{best}^{(t+1)} = \mathbf{x}_i^{(t+1)}$.
  else
    $\mathbf{g}_{best}^{(t+1)} = \mathbf{g}_{best}^{(t)}$.
  end if
  Set $t = t + 1$ {Iteration counter is increasing}.
until Termination criteria are satisfied.
Report the best particle.

---

is the $skill_{id}$ and y, z are the indices of experts that have the skill from experts' list $C(s_i) = \{1, 2, \ldots, E_i\}$.

Particle evaluation

The relationship between experts is represented by a social network, where nodes represent experts and edges represent the communication cost (i.e., weight) between two experts. The weight between expert i and expert j is represented in Eq. 1.

The least communication cost among team members $CC(x_k)$ can be computed according to Eq. 2. The particle with minimum weight among all evaluated particles is considered as a g_{best} (global best particle), where the local best is assigned for each particle as $pbest$.

Particle velocity update

The initial particles' velocities contain a set of random new swap operators (NSO(s)). Each particle updates its velocity as shown in Eq. 13.

The single-point crossover operator is used to produce new individuals by combining sub-individuals from the current individual and the global best individual (g_{best}) in the whole population. After applying the crossover operator, two new individuals are obtained with mixed expert assignments from each other. Finally, one team configuration will be selected randomly $\mathbf{x}_{cross}^{(t)}$

$$\mathbf{v}_i^{(t+1)} = \mathbf{v}_i^{(t)} \oplus \alpha \otimes (\mathbf{p}_{best_i}^{(t)} - \mathbf{x}_i^{(t)}) \oplus \beta \otimes (\mathbf{x}_{cross}^{(t)} - \mathbf{x}_i^{(t)}). \tag{13}$$

where α, β are random numbers between [0,1] and the mark "$\oplus$" is a combined operator of two swap operators. The mark "$\otimes$" means the probability of α that all swap operators are selected in the swap sequences $(\mathbf{P}_{best_i}^{(t)} - \mathbf{x}_i^{(t)})$ and the probability of β that all swap operators are selected in the swap sequences $(\mathbf{x}_{cross}^{(t)} - \mathbf{x}_i^{(t)})$ to include in the updated velocity.

Particle position update

Particle positions are updated according to Eq. 14 by applying the sequences of the new swap operators [NSO(s)] to the current particle in order to obtain the new particle with a new position. All previous process are repeated till reaching to the maximum number of iterations.

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} \oplus \mathbf{v}_i^{(t+1)}, \quad i = \{1, \ldots, SW\} \tag{14}$$

Example of IPSONSO for team formation problem

In the following example, we consider a given project Pr which requires a set of skills to be accomplished, i.e., $=\{Network, Analysis, Algorithm\}$. Also, assume there exist a set of 5 experts (a,b,c,d,e) associated with their skills as follows: $s(a) = \{Network, Algorithm, Search\}$, $s(b) = \{Algorithm, Classification, Network\}$, $s(c) = \{Detection, Analysis\}$, $s(d) = \{Analysis, Graph\}$, $s(e) = \{Network, Analysis\}$.

The relationship between experts is represented by a social network where the nodes represent experts and the edges represent the communication cost (i.e., weight) between two experts as shown in Fig. 4.

The weight between experts can be computed as shown in Eq. (1).

Some of teams that have the required skills can be formed such as $T_1 = \{a, c\}$, $T_2 = \{a, d\}$, $T_3 = \{a, e\}$, $T_4 = \{b, c\}$, $T_5 = \{a, b, c\}$ and $T_6 = \{a, d, e\}$. The communication cost of the formed teams is defined as follows: $C(T_1) = \infty$, $C(T_2) = 0.8$, $C(T_3) = 0.8$, $C(T_4) = \infty$, $C(T_5) = 0.66$, $C(T_6) = 1.6$

A particle in IPSONSO algorithm is an array list of size 1×3, where the first needed skill is "Network," the second one is "Analysis," and the third skill is "Algorithm" as shown in Fig. 5. Figure 5 represents the possible values for each index of a particle in the IPSONSO algorithm. As for required $skill_{id} = 1$, there are three experts that have this skill, i.e., (a,b,e).

In the following subsection, the main steps of the proposed algorithm are highlighted when it is applied on the random dataset as described in Sect. 3.5 and shown in Figs. 4 and 5.

- *Initialization* In the IPSONSO algorithm, the initial population (particles) and their velocities are generated randomly. Each velocity is a swap sequence (i.e., sequence set of swap operators) that represented by a tuple $<x, y, z>$ where x is the $skill_{id}$ and y and z are the indices of the current and the new experts, respectively. An example of the initialization of two particles A, B is shown in Table 2.
- *Particles evaluation* The communication cost for each particle is computed as: $C(A) = \infty$, and $C(B) = 1.55$.
- *Particle positions and velocities update* The particle with minimum weight among all evaluated particles is considered as a g_{best} (particle B in our example), and the local best is assigned for each particle as pbest. In each iteration, the updated velocities and particle positions are computed as shown in Equations 13 and 14, respectively.

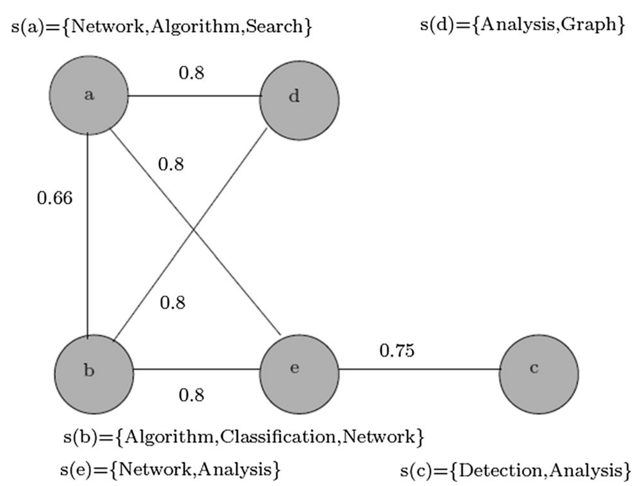


Fig. 4 The relationship between experts

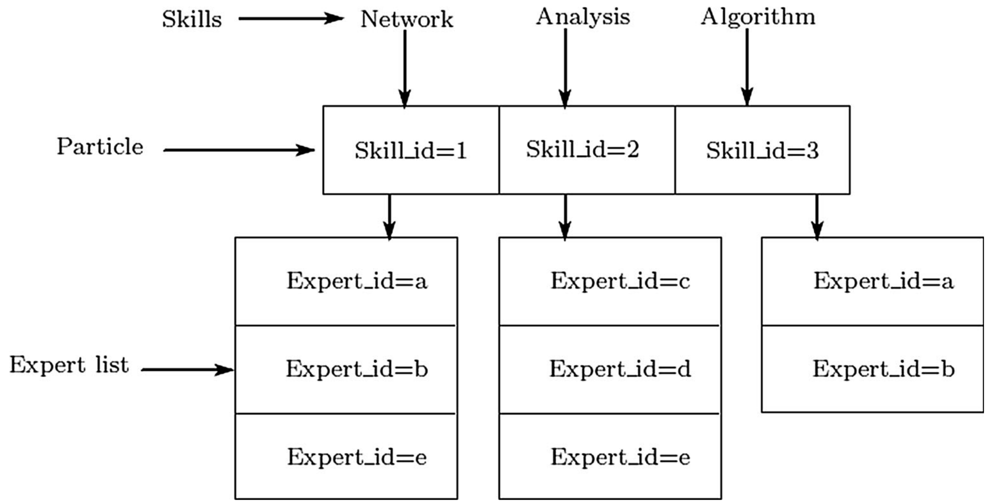


Fig. 5 Particle representation in the IPSONSO algorithm

Table 2 Example of two particles and their velocities

Particle_id	Particle	Velocity
A	(a,c,a)	(1,1,2),(2,1,2)
B	(e,c,b)	(1,2,1),(3, 1,0)

- *Crossover* The single-point crossover is applied between the g_{best} and particle A as shown in Fig. 6. The particle with minimum weight is chosen as a result of crossover, in our example $C(A1) = 1.55$ and $C(A2) = 0.66$. Therefore, the $\mathbf{x}_{cross}^{(t)}$ particle is A_2 = (a,c,b).
- **Velocity update.** The velocity of particle A is calculated as follows. $v^{(t)} = ((1,1,2),(2,1,2)) \oplus (3,1,0) = ((1,1,2),(2,1,2),(3,1,0))$.
- **Particle position update.** $A^{(t=1)} = (a,c,a) + ((1,1,2),(2,1,2),(3,1,0)) = (e,c,a) + ((2,1,2),(3,1,0)) = (e,e,a) + ((3,1,0)) = (e,e,a)$
- **Particles evaluation.** Particle A (a,c,a) is updated to (e,e,a), and its communication cost is $C(A) = 0.8$.

The same processes are applied for particle B. The next iteration, a pbest, is updated for particle A that changed from ∞ to 0.8, and the same g_{best} can be updated according to the particle that has a minimum communication cost. After a number of iterations, the

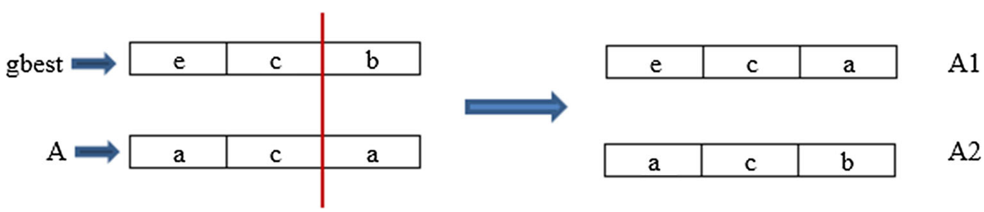


Fig. 6 Example of single-point crossover

Table 3 Parameter setting

Exp. no.	No. of iterations	No. of initial population	No. of skills	No. of experts
1	5	5	5	10
2	5	5	10	20
3	10	5	15	30
4	10	5	20	40
5	10	5	25	50
6	10	10	30	60
7	20	10	35	70
8	20	10	40	80
9	20	10	45	90
10	20	10	50	100

Table 4 Comparison between SPSO and IPSONSO on random data (numerical results)

Exp. no.	No. of skills		SPSO	ISPONSO
1	5	Min	0.32	0.13
		Max	2.03	2
		Mean	1.0558	**0.8534**
		SD	**0.35811918**	0.383973
2	10	Min	1.76	1.3
		Max	3.98	3.77
		Mean	2.8764	**2.5746**
		SD	**0.537916**	0.599637
3	15	Min	3.47	3.2
		Max	6.49	6.36
		Mean	5.108	**4.435**
		SD	**0.7681651**	0.8339218
4	20	Min	5.82	4.47
		Max	8.97	8.45
		Mean	5.108	**4.435**
		SD	7.3376	**6.7102**
5	25	Min	7.13	6.6
		Max	11.41	11.14
		Mean	9.6674	**8.9232**
		SD	**0.995715**	1.005622
6	30	Min	9.68	8.29
		Max	14.06	13.53
		Mean	11.8968	**11.0582**
		SD	**1.162541972**	1.355357758
7	35	Min	12	10.36
		Max	16.54	15.81
		Mean	13.806	**12.6634**
		SD	**1.0651186**	1.2172462
8	40	Min	14.27	12.69
		Max	18.25	17.44
		Mean	16.298	**14.9864**
		SD	**0.9159182**	1.1730264
9	45	Min	16.41	13.33
		Max	21.21	19.85
		Mean	18.7134	**17.1106**
		SD	**1.0226567**	1.292797
10	50	Min	18.04	12.81
		Max	23.66	22.91
		Mean	20.7332	**19.0344**
		SD	**1.213784**	1.755581

most feasible team is formed so far for required skills (i.e., the global best particle g_{best} so far).

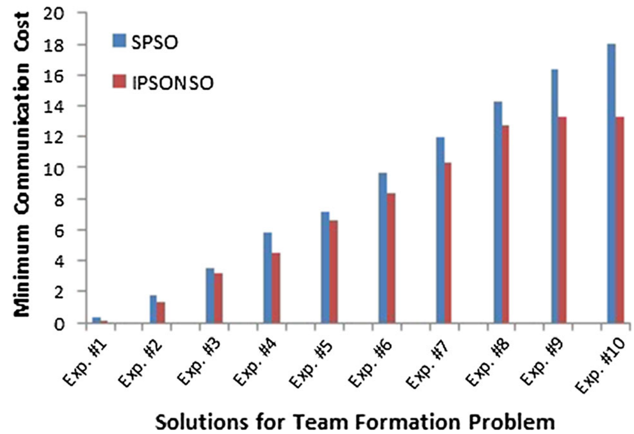


Fig. 7 Comparison between SPSO and IPSONSO on random dataset

Numerical experiments

Ten experiments are performed on random dataset as described in Sect. 3.5 with different skills and expert numbers to evaluate the performance of the proposed algorithm that focuses on iteratively minimizing the communication cost among team members. The proposed algorithm is compared against the standard PSO (SPSO). Also, the performance of the proposed algorithm is investigated on real-life DBLP dataset. The experiments are implemented by Eclipse Java EE IDE V-1.2 running on Intel(R) core i3 CPU- 2.53 GHz with 8 GB RAM and (Windows 7).

Parameter setting

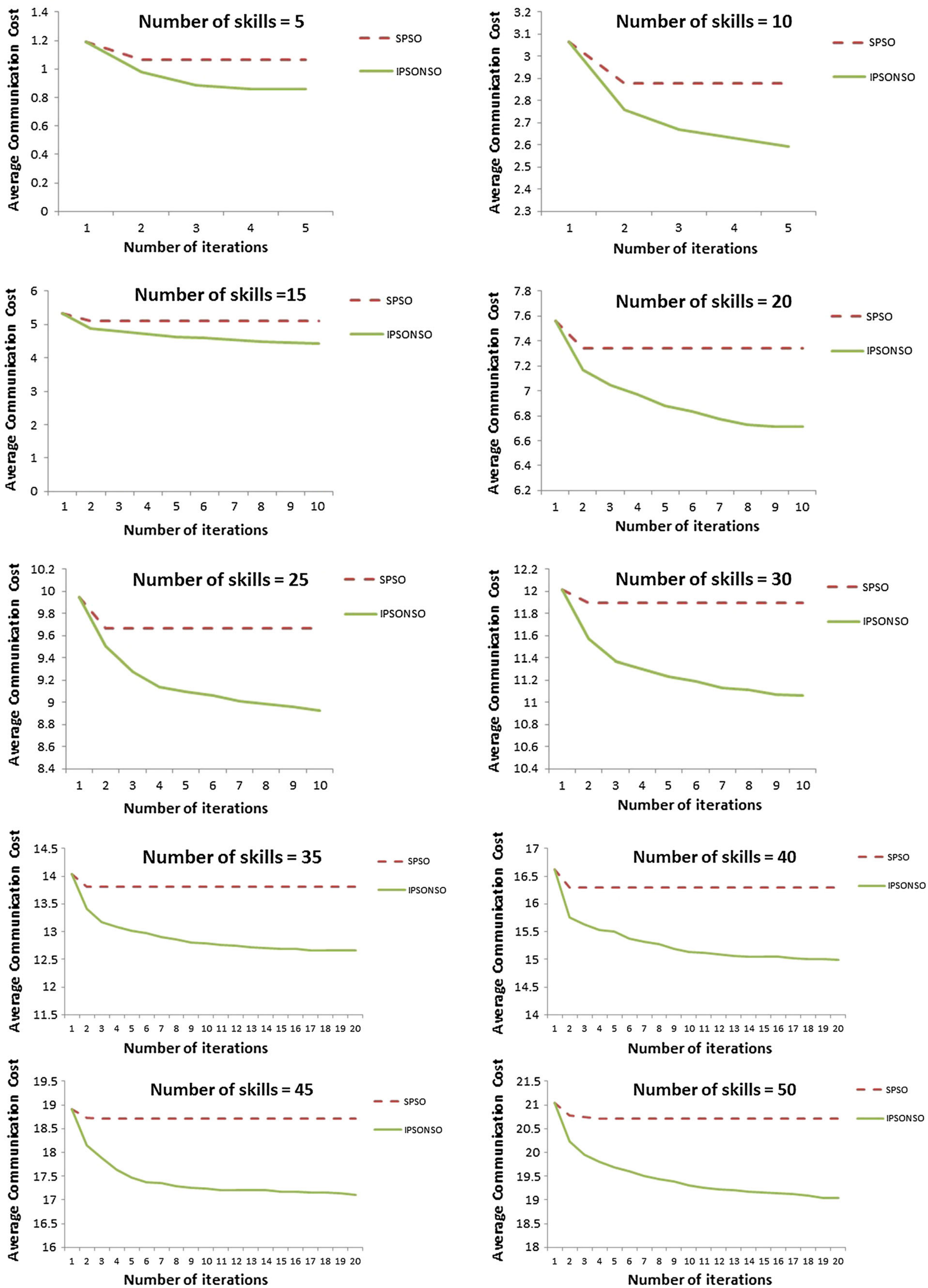
In this subsection, the parameter setting of the proposed algorithm is highlighted, which is used in the ten experiments for a random dataset. The parameters are reported in Table 3.

Random dataset

In this subsection, the performance of the proposed algorithm is investigated on random dataset which is described in Sect. 3.5. The proposed algorithm is applied on different numbers of experts and skills. The results of the proposed algorithm are reported on the subsequent subsections.

Comparison between SPSO and IPSONSO on random data

The first test of the proposed algorithm is to compare it against the standard PSO (SPSO) to verify its efficiency. The results are reported in Table 4. In Table 4, the minimum (min), maximum (max), average (mean) and the standard deviation (SD) of the results are reported over 50 random runs. The best results are reported in bold font. The



Number of skills = 5
Number of skills = 10
Number of skills =15
Number of skills = 20
Number of skills = 25
Number of skills = 30
Number of skills = 35
Number of skills = 40
Number of skills = 45
Number of skills = 50
SPSO
IPSONSO
Average Communication Cost
Number of iterations

◀ **Fig. 8** Comparison between SPSO and IPSONSO on random dataset on average communication cost

Table 5 Comparison between SPSO and IPSONSO on DBLP data (numerical results)

Exp. no.	No. of skills		SPSO	ISPONSO
1	2	Min	0.5	0.5
		Max	100	98
		Mean	19.7952	**18.8732**
		SD	20.977518	**20.144472**
2	4	Min	4.76	4.45
		Max	110.5	109.22
		Mean	40.1284	**36.65652174**
		SD	36.84511284	**36.41816124**
3	6	Min	13.45	12.81
		Max	138.97	137.72
		Mean	41.714	**39.5888**
		SD	36.8864513	**35.07954964**
4	8	Min	20.15	15.73
		Max	249.82	249.82
		Mean	40.9676	**38.5218**
		SD	40.49548227	**39.51807614**
5	10	Min	29.12	28
		Max	197.98	175.9
		Mean	74.9074	**69.911**
		SD	51.40512866	**46.8074525**

The best results are given in bold font

results in Table 4 and Fig. 7 show that the proposed algorithm is better than the standard PSO.

Also, the performance of the SPSO and the IPSONSO is shown in Fig. 8 by plotting the number of iterations versus the communication costs. The solid line represents the results of the proposed algorithm, while the dotted line represents the results of the standard PSO (SPSO). The results in Fig. 8 show that the proposed algorithm can obtain minimum communication cost faster than the standard PSO.

DBLP: real-life data

In this work, the DBLP datasets are used, which has been extracted from DBLP XML released on July 2017. The DBLP dataset is one of the most popular open bibliographic information about computer science journals and different proceedings that can be extracted in the form of XML document type definition (DTD). The following steps are applied to construct 4 tables as follows.

1. Author (name, paper_key), 6054672 records.
2. Citation (paper_cite_key, paper_cited_key), 79002 records.
3. Conference (conf_key,name,detail), 33953 records.
4. Paper (title, year, conference,paper_key), 1992157 records.

Our attention is focused on papers that have been published only in year 2017 (22364 records). Then, the DBLP dataset is restricted to the following 5 fields of computer science: databases (DB), theory (T), data mining (DM), artificial intelligence (AI), and software engineering (SE).

In order to construct the DBLP graph, the following steps are applied.

- The expert set consists of the authors who have at least three papers in DBLP (77 authors have published papers >3).
- Two experts are connected if they share papers' skills. The communication cost c_{ij} of expert i and j is estimated as shown in Eq. (1).
- The most important shared skills are considered among the experts extracted from the titles of 267 papers by using StringTokenizer in java.

It worth to mention that the papers of the major 10 conferences in computer science (with 1707 records) are included. Five experiments are conducted, and the average results are taken over 50 runs. The number of skills selected randomly from the most shared skills among authors with initial population is 3 and 10 number of iterations.

Comparison between SPSO and IPSONSO on DBLP dataset

In this subsection, the proposed algorithm is compared against the standard PSO (SPSO) with different numbers of experts and skills for DBLP dataset by reporting the maximum (max), average (mean) and standard deviation (SD) in Table 5.

Also, in Fig. 9, the results of the standard PSO (SPSO) and the proposed IPSONSO are presented by plotting the number of iterations versus the CI of average communication cost. The solid line represents the results of the proposed IPSONSO, while the dotted line represents the results of the SPSO. The result in Fig. 9 shows that the performance of the proposed algorithm is better than the performance of SPSO.

Confidence interval (CI)

A confidence interval (CI) measures the probability that a population parameter falls between two set values (upper and lower bound). It constructed at a confidence level (C) such as 95% (i.e., 95% CI). The 95% confidence interval

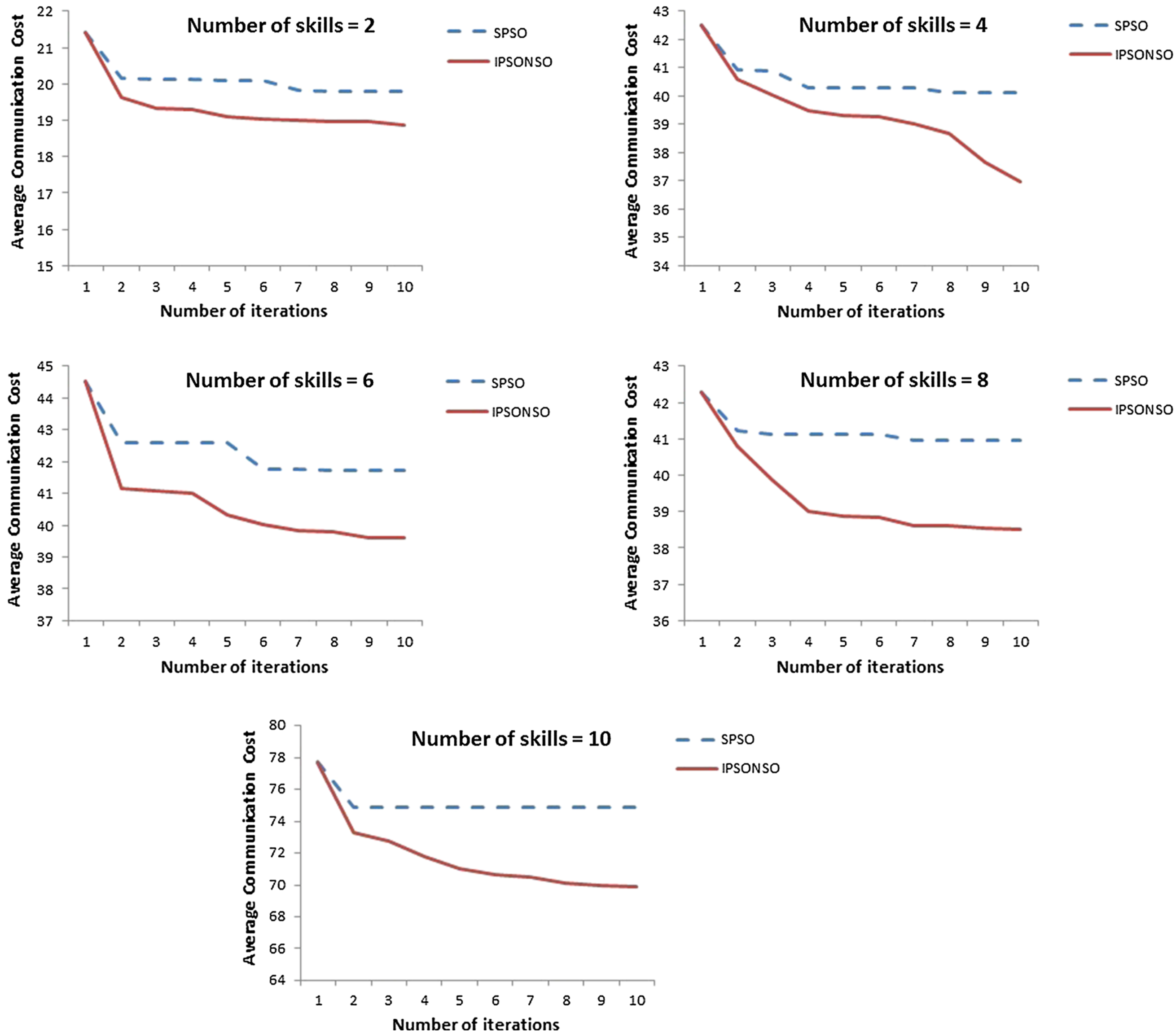


Fig. 9 Comparison between SPSO and IPSONSO on DBLP data on average communication cost

Table 6 CI on average communication cost for experiments 1 and 2

Iteration no.	Exp.1		Exp.2	
	SPSO	IPSONSO	SPSO	IPSONSO
1	1.190816 ± 0.12189	1.190816 ± 0.12189	3.06551 ± 0.162115	3.06551 ± 0.162115
2	1.062653 ± 0.09937	0.98102 ± 0.11673	2.88 ± 0.150476	2.759184 ± 0.151917
3	1.062653 ± 0.09937	0.883265 ± 0.11042	2.88 ± 0.150476	2.669184 ± 0.150022
4	1.062653 ± 0.09937	0.861837 ± 0.10781	2.88 ± 0.150476	2.629184 ± 0.141775
5	1.062653 ± 0.09937	0.856122 ± 0.10739	2.88 ± 0.150476	2.592857 ± 0.133992

uses the sample's mean and standard deviation by assuming a normal distribution. CI can be computed as follows.

$$\mathrm{CI} = mean \pm confidence \tag{15}$$

(γ, SD, sample size), where γ depends on the confidence level (i.e., $\gamma = 1 - C$), SD is the standard deviation of the sample, and sample size is the size of population. In case of

Table 7 CI on average communication cost for experiments 3 and 4

Iteration no.	Exp.3		Exp.4	
	SPSO	IPSONSO	SPSO	IPSONSO
1	5.3402 ± 0.20933	5.3402 ± 0.20933	7.5582 ± 0.216405	7.5582 ± 0.216405
2	5.108 ± 0.212921	4.8724 ± 0.205975	7.3376 ± 0.216517	7.1686 ± 0.236298
3	5.108 ± 0.212921	4.8024 ± 0.206984	7.3376 ± 0.216517	7.0484 ± 0.240729
4	5.108 ± 0.212921	4.7168 ± 0.205289	7.3376 ± 0.216517	6.9728 ± 0.242955
5	5.108 ± 0.212921	4.6232 ± 0.211876	7.3376 ± 0.216517	6.882 ± 0.249804
6	5.108 ± 0.212921	4.594 ± 0.212772	7.3376 ± 0.216517	6.8336 ± 0.251335
7	5.108 ± 0.212921	4.545 ± 0.216073	7.3376 ± 0.216517	6.7708 ± 0.260164
8	5.108 ± 0.212921	4.4838 ± 0.222785	7.3376 ± 0.216517	6.7248 ± 0.257636
9	5.108 ± 0.212921	4.4504 ± 0.22932	7.3376 ± 0.216517	6.7158 ± 0.257715
10	5.108 ± 0.212921	4.435 ± 0.231147	7.3376 ± 0.216517	6.7102 ± 0.258372

Table 8 CI on average communication cost for experiments 5 and 6

Iteration no.	Exp.5		Exp.6	
	SPSO	IPSONSO	SPSO	IPSONSO
1	9.9452 ± 0.307161	9.9452 ± 0.307161	12.0114 ± 0.315103	12.0114 ± 0.315103
2	9.6674 ± 0.275993	9.5062 ± 0.274675	11.8968 ± 0.322234	11.577 ± 0.336986
3	9.6674 ± 0.275993	9.2752 ± 0.273347	11.8968 ± 0.322234	11.3678 ± 0.358297
4	9.6674 ± 0.275993	9.1354 ± 0.278254	11.8968 ± 0.322234	11.3032 ± 0.357704
5	9.6674 ± 0.275993	9.095 ± 0.277179	11.8968 ± 0.322234	11.2354 ± 0.360561
6	9.6674 ± 0.275993	9.0596 ± 0.26873	11.8968 ± 0.322234	11.1862 ± 0.361863
7	9.6674 ± 0.275993	9.0152 ± 0.266973	11.8968 ± 0.322234	11.1274 ± 0.380261
8	9.6674 ± 0.275993	8.9874 ± 0.272385	11.8968 ± 0.322234	11.1124 ± 0.38018
9	9.6674 ± 0.275993	8.9634 ± 0.276761	11.8968 ± 0.322234	11.074 ± 0.378704
10	9.6674 ± 0.275993	8.9232 ± 0.278739	11.8968 ± 0.322234	11.0582 ± 0.375679

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

The performance (%) between the compared algorithms can be computed in Eq. (16).

$$Performance(\%) = \frac{(Avg_{(\mathrm{SPSO})} - Avg_{(\mathrm{ISPSONSO})})}{Avg_{(\mathrm{SPSO})}} \tag{16}$$

where $Avg_{(\mathrm{SPSO})}$ and $Avg_{(\mathrm{ISPSONSO})}$ are the average results obtained from SPSO and IPSONSO algorithms, respectively.

Confidence interval (CI) for random data

In the following tables, the CI of average communication cost is presented for 10 experiments on random generated data. The results in Table 6 show the average communication cost for experiments 1 and 2. In Table 6, the results of IPSONSO decrease iteratively to the number of iterations than SPSO with achieving better performance ranged from 8% in the second iteration to 19% in the last iteration for experiment 1, while the percentage of the improved results ranged from 4 to 10% when it is compared with SPSO in experiment 2.

The results of experiments 3 and 4 are reported in Table 7. In Table 7, the results of IPSONSO are better and more efficient than SPSO with average communication cost and went down from 5 to 13% during iterations and the average communication cost of proposed IPSONSO minimized by percentage ranged from 2 to 9% at the end of iterations when compared with SPSO results (Table 7).

In Table 8, for experiment 5, the performance of average communication cost of IPSONSO solution is improved within the range 2–8% when it is compared with SPSO along with the number of iterations and the proposed IPSONSO has proven its efficiency for team formation with minimum communication cost in the range from 3 to 7% better than SPSO.

In Table 9, the results of experiment 7 show that the IPSONSO achieves better performance results and reaches to 8% than SPSO with respect to the average communication cost along the number of iterations, and for experiment 8, the average communication cost of the proposed

Table 9 CI on average communication cost for experiments 7 and 8

Iteration no.	Exp.7		Exp.8	
	SPSO	IPSONSO	SPSO	IPSONSO
1	14.03 ± 0.306137	14.03 ± 0.306137	16.6226 ± 0.259503	16.6226 ± 0.259503
2	13.806 ± 0.29523	13.4114 ± 0.313878	16.298 ± 0.253875	15.7592 ± 0.270985
3	13.806 ± 0.29523	13.178 ± 0.315661	16.298 ± 0.253875	15.6228 ± 0.300302
4	13.806 ± 0.29523	13.0792 ± 0.311896	16.298 ± 0.253875	15.5332 ± 0.299227
5	13.806 ± 0.29523	13.0116 ± 0.321766	16.298 ± 0.253875	15.379 ± 0.287824
6	13.806 ± 0.29523	12.9668 ± 0.328314	16.298 ± 0.253875	15.3226 ± 0.297032
7	13.806 ± 0.29523	12.907 ± 0.334154	16.298 ± 0.253875	15.2762 ± 0.300185
8	13.806 ± 0.29523	12.8642 ± 0.336283	16.298 ± 0.253875	15.1864 ± 0.312258
9	13.806 ± 0.29523	12.8058 ± 0.326201	16.298 ± 0.253875	15.138 ± 0.318151
10	13.806 ± 0.29523	12.7858 ± 0.332691	16.298 ± 0.253875	15.1126 ± 0.32227
11	13.806 ± 0.29523	12.7596 ± 0.330194	16.298 ± 0.253875	15.0896 ± 0.32491
12	13.806 ± 0.29523	12.7416 ± 0.335089	16.298 ± 0.253875	15.0586 ± 0.332303
13	13.806 ± 0.29523	12.7124 ± 0.337857	16.298 ± 0.253875	15.0522 ± 0.332689
14	13.806 ± 0.29523	12.6984 ± 0.338417	16.298 ± 0.253875	15.0484 ± 0.331174
15	13.806 ± 0.29523	12.6928 ± 0.339173	16.298 ± 0.253875	15.0422 ± 0.329128
16	13.806 ± 0.29523	12.6826 ± 0.340435	16.298 ± 0.253875	15.0148 ± 0.326707
17	13.806 ± 0.29523	12.6668 ± 0.338979	16.298 ± 0.253875	15.0044 ± 0.323845
18	13.806 ± 0.29523	12.6656 ± 0.338097	16.298 ± 0.253875	15.0024 ± 0.323594
19	13.806 ± 0.29523	12.6656 ± 0.338097	16.298 ± 0.253875	14.9864 ± 0.32514
20	13.806 ± 0.29523	12.6634 ± 0.337397		

Table 10 CI on average communication cost for experiments 9 and 10

Iteration no.	Exp.9		Exp.10	
	SPSO	IPSONSO	SPSO	IPSONSO
1	18.914 ± 0.281336	18.914 ± 0.281336	21.0362 ± 0.354561	21.0362 ± 0.354561
2	18.7324 ± 0.286087	18.1518 ± 0.354953	20.78 ± 0.343793	20.2314± 0.409779
3	18.7134 ± 0.283461	17.8892 ± 0.367884	20.7436 ± 0.337998	19.9418 ± 0.38964
4	18.7134 ± 0.283461	17.632 ± 0.367912	20.7032 ± 0.336438	19.7968 ± 0.387752
5	18.7134 ± 0.283461	17.4772 ± 0.354376	20.7032 ± 0.336438	19.6926 ± 0.392841
6	18.7134 ± 0.283461	17.3748 ± 0.351345	20.7032 ± 0.336438	19.5954 ± 0.394692
7	18.7134 ± 0.283461	17.3468 ± 0.348526	20.7032 ± 0.336438	19.5032 ± 0.411394
8	18.7134 ± 0.283461	17.2908 ± 0.343192	20.7032 ± 0.336438	19.4398 ± 0.406
9	18.7134 ± 0.283461	17.2624 ± 0.344025	20.7032 ± 0.336438	19.3798 ± 0.436885
10	18.7134 ± 0.283461	17.2414 ± 0.348188	20.7032 ± 0.336438	19.2966 ± 0.453195
11	18.7134 ± 0.283461	17.2102 ± 0.352974	20.7032 ± 0.336438	19.2538 ± 0.45292
12	18.7134 ± 0.283461	17.2102 ± 0.352974	20.7032 ± 0.336438	19.2226 ± 0.448156
13	18.7134 ± 0.283461	17.2012 ± 0.354871	20.7032 ± 0.336438	19.2032± 0.455017
14	18.7134 ± 0.283461	17.2 ± 0.355861	20.7032 ± 0.336438	19.164 ± 0.47142
15	18.7134 ± 0.283461	17.1768 ± 0.355253	20.7032 ± 0.336438	19.15 ± 0.47069
16	18.7134 ± 0.283461	17.1748 ± 0.355428	20.7032 ± 0.336438	19.1342 ± 0.468485
17	18.7134 ± 0.283461	17.162 ± 0.357137	20.7032 ± 0.336438	19.1218 ± 0.468969
18	18.7134 ± 0.283461	17.1552 ± 0.353732	20.7032 ± 0.336438	19.089 ± 0.47331
19	18.7134 ± 0.283461	17.1416 ± 0.348797	20.7032 ± 0.336438	19.0414 ± 0.486003
20	18.7134 ± 0.283461	17.1106 ± 0.358338	20.7032 ± 0.336438	19.0344 ± 0.486613

IPSONSO is reduced by 8% over the 20 iterations when it is compared with SPSO.

In Table 10, the results of experiment 9 show that the average communication cost of IPSONSO performance

Table 11 CI on average communication cost for 2 and 4 skills in DBLP dataset

Iteration no.	2 skills		4 skills	
	SPSO	IPSONSO	SPSO	IPSONSO
1	21.4008 ± 6.582066	21.3952 ± 6.582045	42.4982 ± 11.18499	42.4982 ± 10.66317
2	20.1608 ± 5.967286	19.5534 ± 5.752612	40.9432 ± 10.61271	40.5952 ± 10.5503
3	20.1104± 5.962677	19.3314 ± 5.713089	40.8654 ± 10.6168	40.0532 ± 10.26711
4	20.1008 ± 5.963655	19.198 ± 5.711924	40.3038 ± 10.30576	39.4688 ± 10.26176
5	20.0746 ± 5.966061	19.09 ± 5.632937	40.3038 ± 10.30576	39.329 ± 10.26794
6	20.0246 ± 5.966061	19.038 ± 5.631029	40.3038 ± 10.30576	39.2714 ± 10.17108
7	19.8324 ± 5.829156	18.9912 ± 5.586283	40.3038 ± 10.30576	39.0216 ± 10.12742
8	19.7972 ± 5.814393	18.9704 ± 5.585088	40.1294 ± 10.2129	38.6834 ± 10.11918
9	19.7972 ± 5.814393	18.9528 ± 5.586755	40.1294 ± 10.2129	37.6684 ± 9.575164
10	19.56196 ± 5.970663	18.66717 ± 5.733383	40.1284 ± 10.21276	36.95652 ± 15.33421

Table 12 CI on average communication cost for 6 and 8 skills in DBLP dataset

Iteration no.	6 skills		8 skills	
	SPSO	IPSONSO	SPSO	IPSONSO
1	44.523 ± 10.74433	44.523 ± 10.74433	42.2874 ± 11.81881	42.2874 ± 11.81881
2	42.5924 ± 10.38519	41.1532± 10.16382	41.2334 ± 11.51306	40.7914 ± 11.29907
3	42.588 ± 10.38615	41.0684± 10.17006	41.1312 ± 11.46521	39.8808 ± 11.10441
4	42.588 ± 10.38615	41.0032 ± 10.15651	41.1312 ± 11.46521	38.9952 ± 10.96132
5	42.588 ± 10.38615	40.3032 ± 10.02055	41.1312 ± 11.46521	38.874 ± 10.96293
6	41.7536 ± 10.254	40.016 ± 9.943788	41.1312 ± 11.46521	38.8442 ± 10.96677
7	41.7536 ± 10.254	39.8334 ± 9.866452	40.9676 ± 11.22457	38.612 ± 10.94327
8	41.714 ± 10.22421	39.8042 ± 9.846164	40.9676 ± 11.22457	38.5958 ± 10.94476
9	41.714 ± 10.22421	39.594 ± 9.72425	40.9676 ± 11.22457	38.5546 ± 10.9505
10	41.714 ± 10.22421	39.5888 ± 9.723376	40.9676 ± 11.22457	38.5218 ± 10.95365

Table 13 CI on average communication cost for 10 skills in DBLP dataset

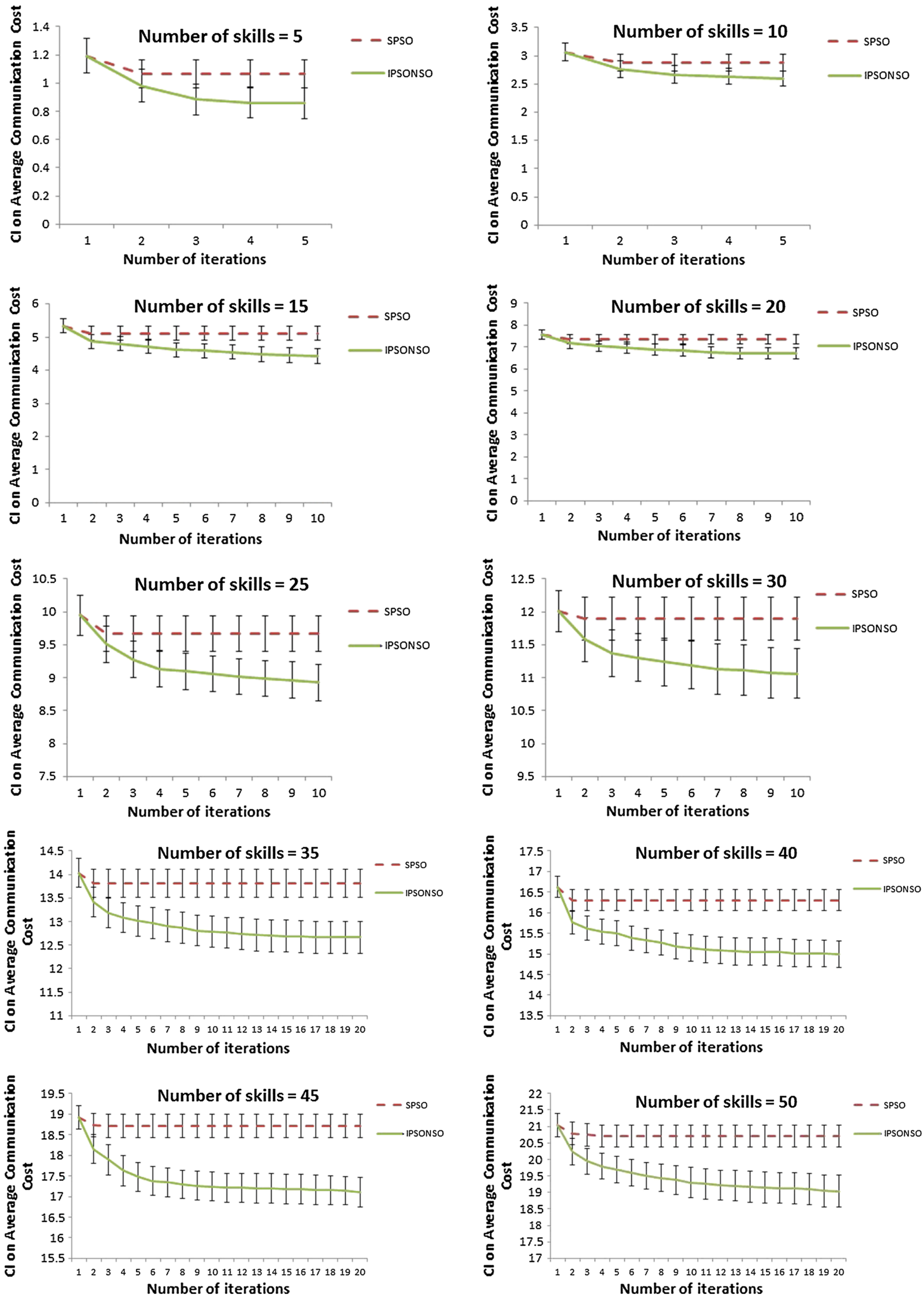
Iteration no.	10 skills	
	SPSO	IPSONSO
1	77.7736 ± 14.33009	77.641 ± 14.35403
2	74.9074 ± 14.24851	73.3012 ± 13.59475
3	74.9074 ± 14.24851	72.7496 ± 13.45998
4	74.9074 ± 14.24851	71.7948 ± 13.35133
5	74.9074 ± 14.24851	71.0142 ± 13.19331
6	74.9074 ± 14.24851	70.6348 ± 13.08217
7	74.9074 ± 14.24851	70.4948 ± 13.08498
8	74.9074 ± 14.24851	70.1218 ± 13.04883
9	74.9074 ± 14.24851	69.9818 ± 12.99535
10	74.9074 ± 14.24851	69.911 ± 12.97413

results is improved from 3 to 9% iteratively with respect to number of iterations when it is compared with the SPSO solution and the results of experiment 10 show that the average communication cost of IPSONSO is reduced iteratively and achieved better performance than SPSO by 8% with respect to the large number of experts and skills.

In Fig. 10, the CI of the proposed algorithm is presented against the standard PSO for different skill numbers by plotting the number of iterations against the CI on average communication cost. The solid line represents the results of the proposed algorithm, while the dotted line represents the standard PSO. The results in Fig. 10 show that the proposed algorithm is better than the standard PSO.

Confidence interval (CI) of SPSO and IPSONSO for DBLP dataset

In this subsection, the CI of SPSO and IPSONSO for DBLP dataset is reported with different numbers of skills as shown in Tables 11, 12 and 13. The results in Table 11 show that the average communication cost of IPSONSO achieves better results than SPSO over the number of iterations. The percentage of improved results is up to 5


Number of skills = 5
Number of skills = 10
Number of skills = 15
Number of skills = 20
Number of skills = 25
Number of skills = 30
Number of skills = 35
Number of skills = 40
Number of skills = 45
Number of skills = 50
SPSO
IPSONSO
CI on Average Communication Cost
Number of iterations

◀**Fig. 10** Confidence interval of SPSO and IPSONSO on random data

and 8% for 2 and 4 skills, respectively, when it is compared with SPSO (Fig. 11).

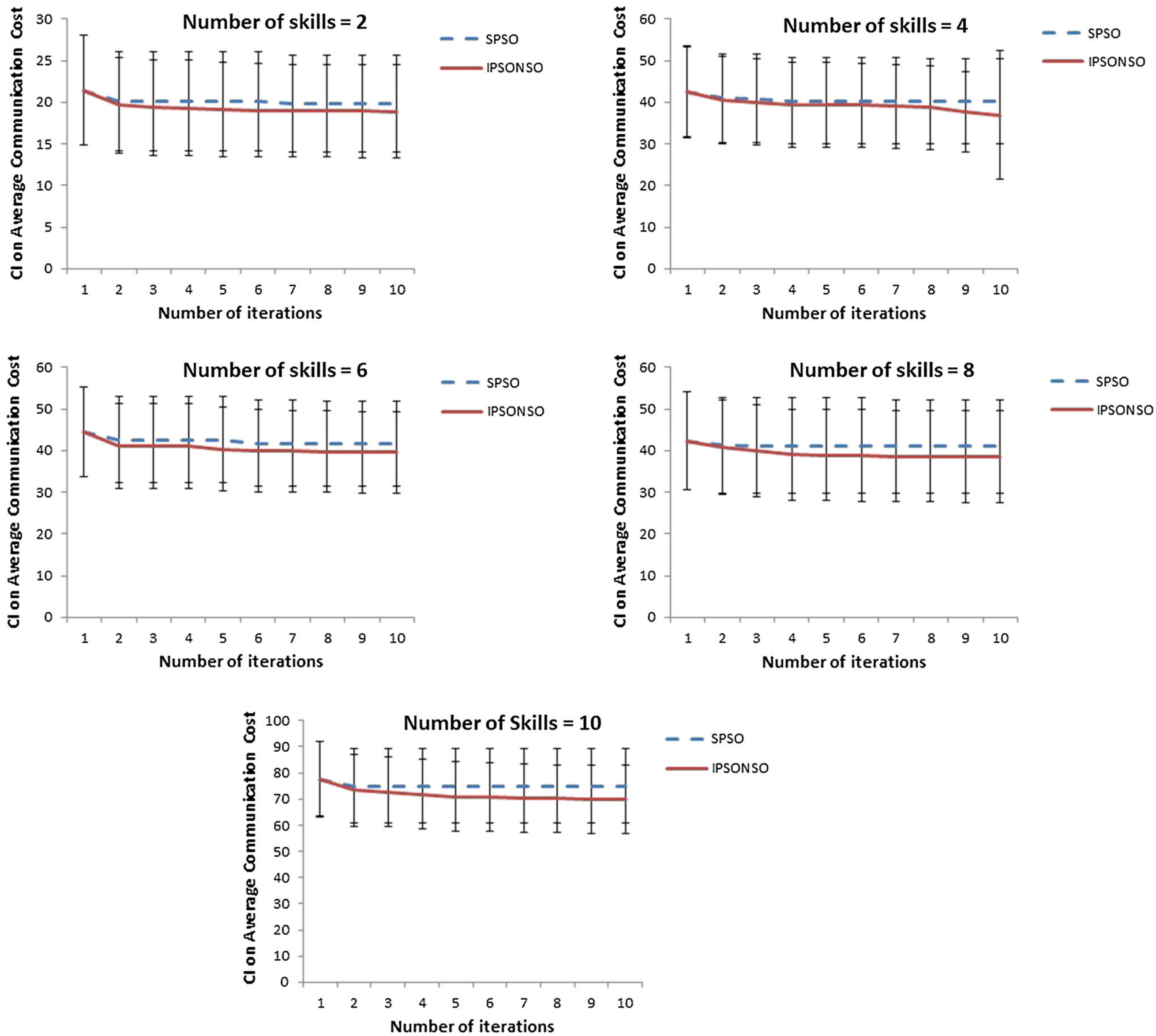


Fig. 11 Confidence interval of SPSO and IPSONSO on DBLP dataset

Table 14 Average processing time of SPSO and IPSONSO on DBLP dataset

No. of skill	SPSO	IPSONSO
2	3.3965	3.6651
4	11.0645	12.1595
6	15.5445	18.2278
8	25.5274	28.9133
10	8.6025	11.5458

In Table 12, the results of the PSO and IPSONSO are reported for 6 and 8 skills. The results in Table 12 show that the IPSONSO obtains better and more efficient results than SPSO with average communication cost and goes down from 3 to 5% during iterations for 6 skills, while it costs up to 6% better than SPSO for 8 skills.

Finally, the IPSONSO algorithm achieves better performance results ranged from 2 to 7% than SPSO with respect to the average communication cost along the number of iterations and number of skills.

The results in Tables 11, 12 and 13 and Fig. 11 show that the performance of the proposed algorithm is better the performance of the standard PSO algorithm.

We can conclude from the previous tables and figure that the performance of the proposed algorithm is better than the performance of the standard PSO.

Average processing time of SPSO and IPSONSO on DBLP dataset

The average processing time (in seconds) of the SPSO and IPSONSO is reported in Table 14 over 30 runs. The time for forming a team by using the proposed algorithm IPSONSO increases almost linearly with number of skills with average processing time ranged from 8 to 34% more time than SPSO due to some processing factors such as the crossover and swap sequence operator.

Conclusion and future work

Team formation problem is the problem of finding a group of team members with the requirement skills to perform a specific task. In this study, a new particle swarm optimization algorithm is investigated with a new swap operator to solve team formation problem. The proposed algorithm is called improved particle optimization with new swap operator (IPSONSO). In IPSONSO algorithm, a new swap operator $NSO(x, y, z)$ is proposed, where x is the $skill_{id}$ and y, z are the indices of experts that have the skill from experts' list. Invoking the single-point crossover in the proposed algorithm can exploit the promising area of the solutions and accelerate the convergence of it by mating the global best solution with a random selected solution. The performance of proposed algorithm is investigated on ten experiments with different numbers of skills and experts and five experiments for real-life DBLP dataset. The results of the proposed algorithm show that it can obtain a promising result in reasonable time. In the future work, combination of the proposed algorithm with other swarm intelligence algorithms is considered to accelerate the convergence of it and avoid the premature convergence. It is worthwhile to test our proposed algorithm over various benchmark problems of nonlinear mixed integer programming problems.

Acknowledgements The research of the third author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Anagnostopoulos A, Becchetti L, Castillo C, Gionis A, Leonardi S (2010) Power in unity: forming teams in large-scale community systems. In: Proceedings of the 19th ACM international conference on Information and knowledge management, pp 599–608

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

Beasley JE, Chu PC (1996) A genetic algorithm for the set covering problem. Eur J Oper Res 94:392–404

Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv (CSUR) 35(3):268–308

Eberhart RC, Shi Y, Kennedy J (2001) Swarm intelligence. The Morgan Kaufmann series in evolutionary computation. Morgan Kaufmann, Waltham

Farasat A, Nikolaev AG (2016) Social structure optimization in team formation. Comput Oper Res 74:127–142

Fathian M, Saei-Shahi M, Makui A (2017) A new optimization model for reliable team formation problem considering experts collaboration network. IEEE Trans Eng Manag 64:586–593

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading

Gutiérrez JH, Astudillo CA, Ballesteros-Pérez P, Mora-Melià D, Candia-Véjar A (2016) The multiple team formation problem using sociometry. Comput Oper Res 75:150–162

Haupt RL, Haupt SE (2004) Practical genetic algorithms. Wiley, London

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

Kalita K, Shivakoti I, Ghadai RK (2017) Optimizing process parameters for laser beam micro-marking using genetic algorithm and particle swarm optimization. Mater Manuf Process 32(10):1101–1108

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. Society for Industrial and Applied Mathematics, pp 587–595

Karduck A, Sienou A (2004) Forming the optimal team of experts for collaborative work. In: Artificial intelligence applications and innovations, pp 267–278

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: 2010 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

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

Sedighizadeh D, Masehian E (2009) Particle swarm optimization methods, taxonomy and applications. Int J Comput Theory Eng 1(5):486

Vallade B, Nakashima T (2013) Improving the performance of particle swarm optimization algorithm with a dynamic search space. In: ADVCOMP: the seventh international conference on advanced engineering computing and applications in sciences, pp 43–48

Wang KP, Huang L, Zhou CG, Pang W (2003) Particle swarm optimization for traveling salesman problem. In: 2003 international conference on machine learning and cybernetics, vol 3, pp 1583–1585

Wei X, Jiang-wei Z, Hon-lin Z (2009) Enhanced self-tentative particle swarm optimization algorithm for TSP. J N China Electr Power Univ 36(6):69–74

Zhang JW, Si WJ (2010) Improved enhanced self-tentative PSO algorithm for TSP. In: 2010 sixth international conference on natural computation (ICNC), vol 5, pp 2638–2641