Team Formation in Business Process Context

Yang Yu*, Weijun Chen, Jing Yang
School of Data and Computer Science
Sun Yat-sen University
Guangzhou, China
Email: yuy@mail.sysu.edu.cn, {chenwj96, yangj357}@mail2.sysu.edu.cn

***Abstract*—High performing teams may benefit the execution efficiency of business processes. In executing a process, the performance of the team is concerned with the individual expertise of team members and the handover relations between executors of adjacent tasks of the process. Team formation problem in the presence of business process can be defined as finding a group of individuals to execute all tasks of the process. Considering individual expertise and handover relations, a method called Bayesian Network based Team Formation (BN-TF) is proposed to address the team formation problem in business process context. Using BN-TF, a team formation problem is first modeled as a Most Probable Explanation (MPE) problem in Bayesian network based on the structural information of the related business process. For solving the transformed MPE problem, we design an improved genetic algorithm called Forward-Backward Greedy Genetic Algorithm (FBG-GA). Experimental results on simulation data verify that BN-TF indeed produces teams that satisfy the requirements while improve the execution efficiency of the business process. Compared with existing methods, RarestFirst and CoverSteiner, which focus on minimizing team communication cost, BN-TF shows improvement in terms of individual expertise and handover relations.**

Keywords—team formation; business process; Bayesian network

I. Introduction

Team formation problem is defined as finding a group of individuals to accomplish a job (project) requiring a set of skills for completion. Based on this definition, team formation problem has been further extended to several variants which focus on aspects such as coordination, load balancing, etc. Consider the situation where the business process of the given job is a known condition (as well as its skill requirements), we discover that previous researches hardly look into team formation problem in the presence of business process. In this paper, we study team formation in business process context, which aims to find a group of resources (experts) to execute all tasks in a business process, and to improve the execution efficiency of the business process.

Former researches on team formation recognize individual expertise of candidates and social relations among them as two key factors contributing to the effectiveness of the assembled team. Studies on event-logs mining have demonstrated that both factors can be modeled and measured by means of process mining techniques. Social relations, or coordination among candidates, are the research focus in recent years, which are studied mainly through social network analysis. In the context of a business process, social relations may exist in the handover of two adjacent actors. For example, Mike and Jim both participate to finish a job and are assigned to execute two successive tasks of the job process respectively. Then once Mike accomplishes his task, it will be Jim who takes over Mike's work and start to execute his own task, and so the job process continues. This is where the handover relation exists. Furthermore, we assume empirically that Mike and Jim may have lower overhead in both communication and the handover of work, if they have cooperated as adjacent actors enough times before. We consider handover relation as a more appropriate description of how team members collaborate in the presence of a business process.

In this paper, a method named Bayesian Network based Team Formation (BN-TF) is proposed to address team formation problem in business process context. Taking both individual expertise and handover relations into consideration, team formation problem is first modeled as Most Probable Explanation (MPE) problem in Bayesian network. We then propose a modified genetic algorithm named Forward-Backward Greedy Genetic Algorithm (FBG-GA) to solve the transformed problem. Experiments on simulation data are conducted to verify the effectiveness of the whole procedure. The results indicate that BN-TF indeed form a team which satisfy the requirements of the given job, while improving the execution efficiency of the business process.

Roadmap: In Section 2 we review the related work on team formation. Section 3 introduces two key concepts which are used through this paper. The formal definition of the problem and our methods for solution are described in Section 4. Experiments on simulation data and analysis of experimental results are demonstrated in Section 5. The final section presents a brief conclusion of the paper.

II. Related Work

There is a considerable amount of work on team formation problem conducted in the field of operation research (OR) [2, 3]. A line of trend is to transform the problem into integral linear programming problem and solve. The main focus of these studies is to find an optimal match between people and functional requirements, which are usually limited to the expertise of individuals.

Recent researches on team formation take into account social relations among potential team members [4, 5]. Lappas, Liu and Terzi [5] explicitly consider collaboration as a contributing factor to the effectiveness of a team, and introduce team formation problem in social networks. Later studies generalize the problem [6]. Team formation problem in the context of social networks is further extended to be considering individual expertise, coordination cost and workload balance simultaneously [7]. The similarity between these studies is that,

This work is Supported by the National Natural Science Foundation of China under Grant No.61572539; the Research Foundation of Science and Technology Major Project in Guangdong Province under Grant Nos.2015B010106007, 2016B010110003; the Research Foundation of Science and Technology Plan Project in Guangdong Province under Grant No.2016B050502006; the Research Foundation of Science and Technology Plan Project in Guangzhou City under Grant No.201620160403001.

social relations in teams are modeled as social network graphs, and the research methods are therefore based on network analysis mostly.

However, in the context of business process, social relations among individuals are more likely to be influenced by the structure of the process in a workflow instance. If two individuals of a team are involved in the process as executors of two successive tasks respectively, then we may expect more interactions between these two, in comparison with other team members which are not directly associated. Hence graph based methods seem insufficient for modeling social relations within business processes. Yet studies on team formation problem hardly looked from the view of business process. References [8, 10] focus on social relations existed in the handover of work in processes, and regard the handover relation as an important factor in team formation, but do not include individual expertise as a contributing factor.

Lin et al. [1] consider individual expertise and handover relations simultaneously, and concludes that skillful individuals combined with smooth handover relations lead to higher execution efficiency of business process. They proposed a method, namely TF-HMM, to address team formation problem in workflow process context. They adopt Hidden Markov Model (HMM) for problem modeling and apply Viterbi algorithm for problem solving. The effectiveness of TF-HMM is demonstrated in [1]. But TF-HMM is only capable of dealing with processes with sequential patterns. For those with branch and parallel patterns, the TF-HMM approach fails to model appropriately, thus inapplicable for general use. The BN-TF method presented in this paper provides a solution to solve team formation problem, given business processes with either sequential or non-sequential patterns.

III. Preliminaries

Previous studies [1, 14] show that both the level of individual expertise and social relations in business processes (namely, specialty and handover in the following sections) can be measured through mining and analysis on event-log data. The definitions of these two key contributing factors are illustrated as follows.

A. Specialty

$Execute\_Frequency(r_i, ac_j)$ represents the number of times recorded in event logs, that activity ac_j successfully executed by individual r_i for all cases. The ratio of activity ac_j to all activities successfully executed by individual r_i is defined as the level of specialty individual r_i to activity ac_j, denoted by $Specialty(r_i, ac_j)$. A higher value of $Specialty(r_i, ac_j)$ indicates that individual r_i is more skillful at executing activity ac_j. In practice, individuals who possess certain expertise are often related with particular business roles.

$$Specialty(r_i, ac_j) = \frac{Execute\_Frequency(r_i, ac_j)}{\sum_{ac_k \in AC} Execute\_Frequency(r_i, ac_k)} \quad (1)$$

B. Handover

$Handover\_Frequency(r_i, r_j)$ represents the number of times recorded in event logs, that r_i handed the work to r_j, and r_j successfully executed. The ratio of successful handovers between r_i and r_j to all successful handovers r_i handed is defined as the handover from r_i to r_j, denoted by $Handover(r_i, r_j)$. A higher value of $Handover(r_i, r_j)$ indicates that individuals r_i and r_j cooperate well in terms of handovers, which may suggest a close working relationship between r_i and r_j.

$$Handover(r_i, r_j) = \begin{cases} 1.0, & \text{for } r_i = r_j \\ \frac{Handover\_Frequency(r_i, r_j)}{\sum_{r_k \in R} Handover\_Frequency(r_i, r_k)}, & \text{for } r_i \neq r_j \end{cases} \quad (2)$$

IV. Problem Modeling and Solution

A. Problem Definition and Modeling

Team formation problem in business process context is defined as follows. Given a job and a business process $BP = (P, T, F)$ corresponding to the job, and a set of resources $R = \{r_1, r_2, \cdots, r_n\}$, select a subset of R to form a team, which meet the requirements of the job and maximize the $Specialty(r_i, ac_j)$ and $Handover(r_i, r_j)$ to improve the execution efficiency of process.

B. Modeling

Business processes are modeled as Bayesian network for the need of transforming the problem and more effective solution. Bayesian network contains a directed acyclic graph (DAG) and every node of the graph is associated with a probability table $P(X|\pi(X))$. For a business process represented in Petri net model, we demonstrate how to transform it into a Bayesian network.

Given a Petri net representing a business process $BP = (P, T, F)$, where P is a set of places, T is a set of transitions, and $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs. If there are cycles in the Petri net (i.e. repetitions of certain tasks in the business process), then we remove the places and arcs which connect the first and the last transition in the loop, for later converting it into a directed acyclic graph.

Let $G = (C, F, L)$ be a directed acyclic graph of Bayesian network. C and F represent sets of nodes, $L \subseteq (C \times F) \cup (C \times C)$ represents set of directed edges.

Step 1. For the original Petri net, remove all AND-split, AND-join, places nodes and the edges associated with them. For each removed node, add an edge which connects the precursor and successor of the removed node. A directed acyclic graph with only activity nodes (converted from transitions in petri net) and directed edges is then obtained.

Step 2. Rename all activity nodes t_i as c_i and obtain node set $C = \{c_1, c_2, \cdots, c_k\}$. Denote the set of resources $R = \{r_1, r_2, \cdots, r_n\}$. Associate C with R, and denote $c_i = r_j$ as assigning activity t_i to r_j.

Step 3. For each node c_i, add a new node f_i and a directed edge from c_i to f_i. A set of nodes F is generated. The range of each node f_i in F is the set of activities $AC = \{ac_1, ac_2, \cdots, ac_m\}$.

A directed acyclic graph of Bayesian network is obtained, which consists of sets of nodes, C and F, and directed edges.

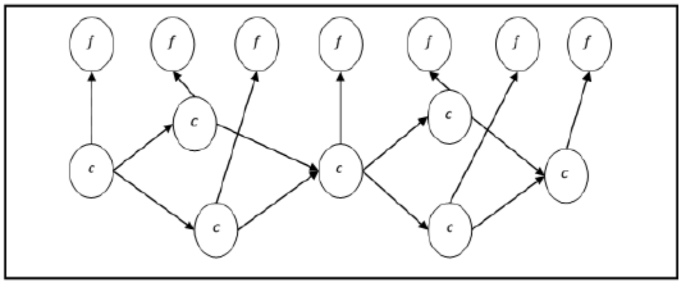

Fig. 1. A directed acyclic graph resulting from transformation.

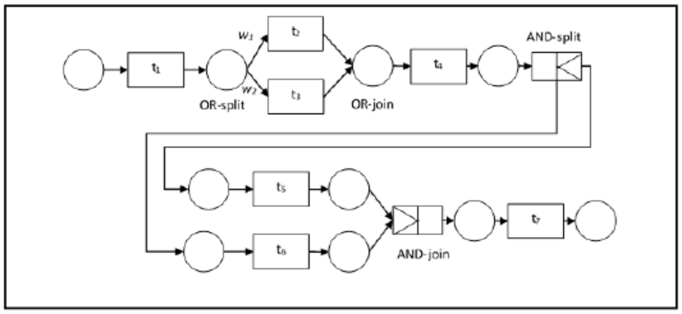

Fig. 2. The original Petri net.

Fig.1 shows a directed acyclic graph transformed from the original Petri net in Fig. 2.

The directed acyclic graph of Bayesian network contains two sets of nodes, F and C. We make the following rules to assign the probability table corresponding to each of the two sets.

Each node in F has only one precursor node $\pi(f_i) = c_i$, , so the probability table of F can be defined as:

$$P(f_i = ac_l | c_i = r_j) = \frac{Specialty(r_j, ac_l)}{\sum_p^{1\le p\le m} Specialty(r_j, ac_p)} \tag{3}$$

The larger $P(f_i = ac_l | c_i = r_j)$ means that resource r_j is more skillful on ac_l, r_j is more likely to be selected to execute the task in node c_i.

The number of precursor nodes $\pi(c_i)$ is uncertain. With an auxiliary function $Q(node, r)$, the probability table of C can be defined as

$$P\left(c_i = r_j | \pi(c_i)\right) = \frac{Q(t_i.preNode, r_j)}{\sum_{each\ r \in R} Q(t_i.preNode, r)} \tag{4}$$

where function $Q(node, r) : (T \cup P) \times R \to [0,1]$ is defined as follows:

$$Q(node, r) = \begin{cases} \frac{1}{|R|}, & node\ is\ the\ first\ place, \\ Handover(executor(node), r), & node\ is\ transition, \\ Q(node.preNode, r), & node\ is\ place\ or\ split, \\ \sum_{each\ p\ in\ node.preNode} p.w \cdot Q(p, r), & node\ is\ OR\ join, \\ \min_{each\ p\ in\ node.preNode} Q(p, r), & node\ is\ AND\ join \end{cases} \tag{5}$$

For function $Q(node, r)$, the parameter $node \in (T \cup P)$, where T represents transitions and P for places in Petri net. The parameter r is a value of c_i and the parameter $node$ is the precursor node of t_i, t_i is a transition in Petri net, corresponding to c_i in Bayesian network.

The following is the explanation of function $Q(node, r)$:

- If $node$ is the first place, it has no precursor. Since $node$ is the precursor of t_i, t_i must be the first task in process. There are no handovers, so each resource in R is selected with equal probability.
- If $node$ is a transition, it represents a task, so there is a handover between the executor of the task and r.
- If $node$ is a place, which is not the first one, nor an AND-split or an OR-split. According to the structural rule of Petri net, there is only a branch before, so the value of function can be solved recursively.
- If $node$ is OR-join, there are several branches before and the precursor executor in each branch should be considered. Let $pre.w$ denotes the selected probability of each branch, which is accessible from statistics of event logs, so the value of function can be calculated by the weighted average of the value of each branch.
- If $node$ is AND-join, there are also several branches before, but the minimal value of branches is only considered because it is the bottleneck of handover relations.

Now that directed acyclic graph and probability table is generated, team formation problem can be redefined as: Given a business process $BP = (P, T, F)$, the values of nodes in F is known because $f_i = t_i$, the goal is to find the values of nodes in C , which is also the $Team = \{r'_1, r'_2, \cdots, r'_k\} \subseteq R$, and maximize the objective function:

$$Team = ArgMAX_{for\ r_i'\ in\ R, 1\le i\le k}$$

$$\left(\prod_i^{1\le i\le k} P(f_i = t_i | c_i = r_i') \cdot \prod_i^{1\le i\le k} P\left(c_i = r_i' | \pi(c_i)\right)\right) \tag{6}$$

In the objective function, the larger $P(f_i = t_i | c_i = r_i')$ means r_i' is more skillful on executing t_i, the larger $P\big(c_i = r_i' | \pi(c_i)\big)$ means r_i' has a better handover relations with its precursors $\pi(c_i)$. To ensure global optimum, the product of probabilities is used in objective function.

The values of nodes in F are given and the values of nodes in C are to be solved. This is a typical Most Probable Explanation problem in Bayesian network.

C. *Solution*

MPE problem can be solved in an iterative fashion applying genetic algorithms. We provide an improved genetic algorithm called Forward-Backward Greedy Genetic Algorithm (FBG-GA) for a faster solution. In FBG-GA, an improved crossover operator is used for acceleration in convergence, which reduce the overall time of finding a satisfying solution. The following parts show the details of FBG-GA.

1) Encoding: Solutions need to be encoded. Firstly, sort all nodes of C in Bayesian network in topological order. Then associate every node c_i with an identifier of the resource, representing that the task in this node is executed by the corresponding resource. Use a list consisting of identifiers to encode a solution. In genetic algorithm, the list is usually called chromosome.

2) Initialization: Initial population is randomly generated. Since all nodes are sorted, let every node c_i randomly associate with an identifier of the resource. When all nodes are associated, a chromosome is generated. A population consists of a number of chromosomes. The size of the population is set to 100 in this paper.

3) Selection: Selection operator selects the fitter chromosome as next generation. Fitness function is used for selecting the fitter solutions. Considering both specialty and handover have much effect on the performance of a team, function (7) is used for measuring the fitness of the chromosome. It is also the objective function previously defined.

$$fitness(r'_1, r'_2, \cdots, r'_k) = \prod_i^{1 \le i \le k} P(f_i = t_i | c_i = r'_i) \cdot \prod_i^{1 \le i \le k} P(c_i = r'_i | \pi(c_i)) \quad (7)$$

4) Crossover: Crossover operator is a process of taking more than one parent chromosomes and producing a child chromosome from them. To accelerate convergence and generate fitter chromosomes, a Forward-Backward Greedy strategy for crossover is used here.

To begin with, randomly select two parent chromosomes from population,$X = (x_1, x_2, \cdots, x_k)$, $Y = (y_1, y_2, \cdots, y_k)$, and then use Forward Greedy strategy and Backward Greedy strategy respectively to get two child chromosomes.

The steps of Forward Greedy algorithm are as follows.

Step 1. Initialize a child chromosome to an empty list Z.

Step 2. Scan two parent chromosomes from left to right, compare the gene at the same position, if x_i equals to y_i then add x_i to Z, else go to the step 3.

Step 3. Now $Z = (z_1, z_2, \cdots z_{i-1})$, compare $g_1 = P(c_i = x_i | c_1 = z_1, c_2 = z_2, \cdots c_{i-1} = z_{i-1}) \cdot P(f_i = t_i | c_i = x_i)$ with $g_2 = P(c_i = y_i | c_1 = z_1, c_2 = z_2, \cdots c_{i-1} = z_{i-1}) \cdot P(f_i = t_i | c_i = y_i)$, if $g_1 > g_2$ then add x_i to Z, else add y_i to Z.

Step 4. Go to the step 2, loop until the scan is over. A child chromosome $Z = (z_1, z_2, \cdots, z_k)$ is generated, $z_i = x_i$ or $z_i = y_i$.

In Forward Greedy algorithm, all genes before z_i are determined when it begins to choose a gene to be z_i. Expression $P(c_i = x_i | \pi) \cdot P(f_i = t_i | c_i = x_i)$ has the same meaning of the fitness function. So the child chromosome can improve the fitness relatively quicker.

The steps of Backward Greedy algorithm are as follows.

Step 1. Initialize a child chromosome to a list *Z* of size *k*.

Step 2. Scan two parent chromosomes from right to left, compare the gene at the same position, if x_i equals to y_i then set $z_i = x_i$, else go to the step 3.

Step 3. Now $Z = (N/A_1, N/A_2, \cdots, N/A_i, z_{i+1}, z_{i+2}, \cdots z_k)$, N/A_i denotes the initial value of z_i, compare $g_1 = P(c_{i+1} = z_{i+1}, c_{i+2} = z_{i+2}, \cdots c_k = z_k | c_i = x_i) \cdot P(f_i = t_i | c_i = x_i)$ with $g_2 = P(c_{i+1} = z_{i+1}, c_{i+2} = z_{i+2}, \cdots c_k = z_k | c_i = y_i) \cdot P(f_i = t_i | c_i = y_i)$, if $g_1 > g_2$ then set $z_i = x_i$, else set $z_i = y_i$.

Step 4. Go to the step 2, loop until the scan is over. A child chromosome $Z = (z_1, z_2, \cdots, z_k)$ is generated, $z_i = x_i$ or $z_i = y_i$.

In Backward Greedy algorithm, all genes after z_i are determined when it begins to choose a gene as z_i. So conditional probability of the current node and its successor nodes is used to estimate the effects of adding a new gene.

Compared with traditional crossover operator, Forward-Backward Greedy algorithm keeps the better gene from parent chromosome, improves the fitness of child chromosomes and accelerates convergence effectively. The probability of crossover is set to 0.6 in this paper.

5) Mutation: To maintain genetic diversity, mutation operator is used. Firstly randomly choose a gene c_i in the chromosome, and t_i is executed by c_i at this node. Then randomly choose a resource which is able to execute t_i, and replace the older gene with the identifier of this resource. In this paper, only one gene probably mutate in every chromosome and the probability of mutation is set to 0.02.

V. EXPERIMENTAL EVALUATION

A. Dataset

Simulation experiments are conducted to verify the proposed algorithms. We create simulated data based on a common fact that a resource with more experience in executing specific activities is more skillful, and hence more likely to be chosen to execute the same activities. Roulette wheel selection is adopted to avoid the issue that only a few resources would be selected repeatedly. Table I shows the parameter setups for creating simulation data, which are based on the characteristics of business processes in practice. Fig. 3 shows an example business process with 7 activities in simulation data.

B. Verify the feasibility of BN-TF

We verify our algorithm through a comparative experiment. RarestFirst algorithm and CoverSteiner algorithm proposed to solve team formation problem in social networks context [5] are adopted for comparison. The two algorithms aim to form satisfactory team with minimized communication cost among team members, either measured by diameter cost or minimum spanning tree cost, respectively.

Teams are formed and evaluated by both metrics respectively: Average Specialty value and average Handover value. For a team, the average values of Specialty and Handover are defined as follows.

Average Specialty is obtained by taking the mean value of:

$$\sum_{for\ each\ activity\ ac_j, and\ its\ actor\ r_i} Specialty(r_i, ac_j) \quad (8)$$

Average Handover is obtained considering all pairs of handover individuals in the process, i.e. take the mean value of:

$$\sum_{for\ each\ pair\ of\ r_i, r_j\ requiring\ handover} Handover(r_i, r_j) \quad (9)$$

The steps of the experiment are as follows.

Step 1. Select 20 jobs randomly from the dataset, each corresponding with a business process from the dataset.

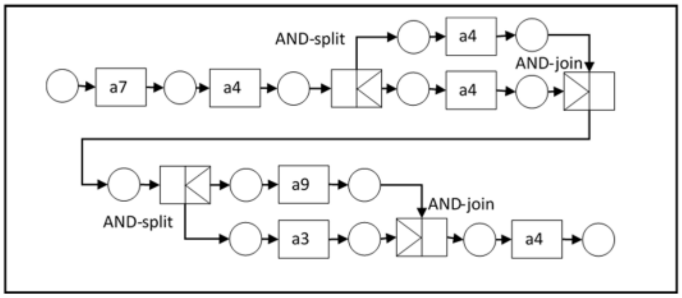

Fig. 3. An example business process.

TABLE. I. SPECIFICATION FOR CREATING SIMULATION DATA

Data Item	Parameter	Value
Business process	Structure	Selection structure : 15% Parallel structure : 15% Sequence structure : 70%
	Amount	20
	Amount of Activities	7~16
Activity	Amount	50
Resource	Amount	100
Case	Amount	3000
Event	Amount	26695

Step 2. Apply BN-TF to model these 20 business processes, and FBG-GA to solve the transformed problem. Obtain the resulting 20 teams.

Step 3. For the same jobs, apply RarestFirst to obtain 20 teams.

Step 4. For the same jobs, apply CoverSteiner to obtain 20 teams.

Step 5. For the 3 resulting teams corresponding to a same job, compare the average values of Specialty and Handover.

Fig. 4 is the comparison of average Specialty value. In all cases, BN-TF achieves a significant increase compared with RarestFirst and CoverSteiner. Results demonstrate the effectiveness of BN-TF in optimizing the Specialty of a team.

Result shown in Fig. 5 suggests that BN-TF has a slight advantage over RarestFirst and CoverSteiner in the comparison of average Handover. By further investigating the results of RarestFirst and CoverSteiner, we discovered that the same resources are likely to be the actors of successive tasks in these teams, since both algorithms tend to form teams with relatively smaller size. Nevertheless, BN-TF still demonstrates its effectiveness in optimizing the Handover of a team.

The results above suggest that BN-TF effectively optimizes both individual expertise (Specialty) and handover relations (Handover). As mentioned in [1], the higher value of Specialty and Handover makes the higher efficiency of work execution of the team. Furthermore, BN-TF combines with the structure of business process, which makes the team more suitable for the process context. The team resulting from BN-TF contributes to the improvement of the execution efficiency of business process with a slight increase in communication cost. The effectiveness of BN-TF in team formation problem is proven.

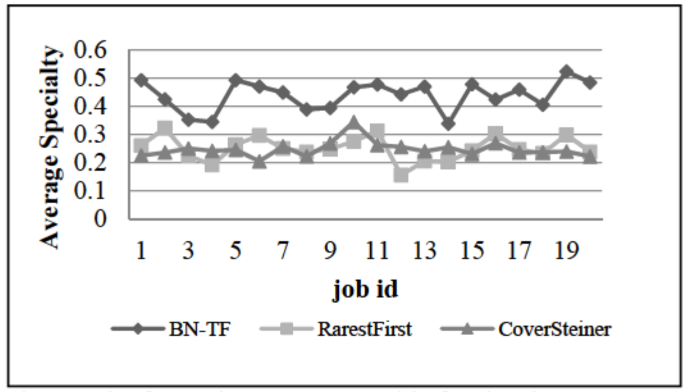

Fig. 4. Results of comparison on average Specialty for selected teams.

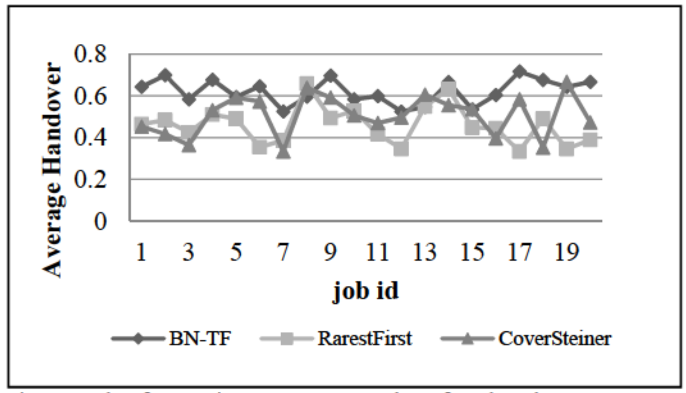

Fig. 5. Results of comparison on average Handover for selected teams.

C. Evaluate the performance of applying FBG-GA

Another experiment is conducted to evaluate the performance of FBG-GA, compared with the traditional genetic algorithm. The major difference between FBG-GA and the traditional genetic algorithm is that FBG-GA uses Forward-Backward Greedy strategy in crossover operator while the traditional genetic algorithm uses one-point or two-point crossover strategy.

The steps are as follow:

Step 1. Select 20 jobs randomly from the dataset, each corresponding with a business process from the dataset.

Step 2. Apply BN-TF to model these 20 business processes.

Step 3. Use either FBG-GA, one-point crossover genetic algorithm or two-point crossover genetic algorithm for solving the transformed problem.

Step 4. Compare the value of fitness through generations.

Fig. 6 shows the time performance of applying the three genetic algorithms. FBG-GA gets an acceptable fitness in 20th generation, while one-point crossover genetic and two-point crossover get a satisfied result in 200th generation. It is obvious that FBG-GA has a faster convergence speed than the traditional genetic algorithms. Furthermore, the value of fitness resulted from FBG-GA is also higher than other two algorithms.

We conclude that FBG-GA performs better in terms of computing overhead, and also results in more satisfying fitness value.

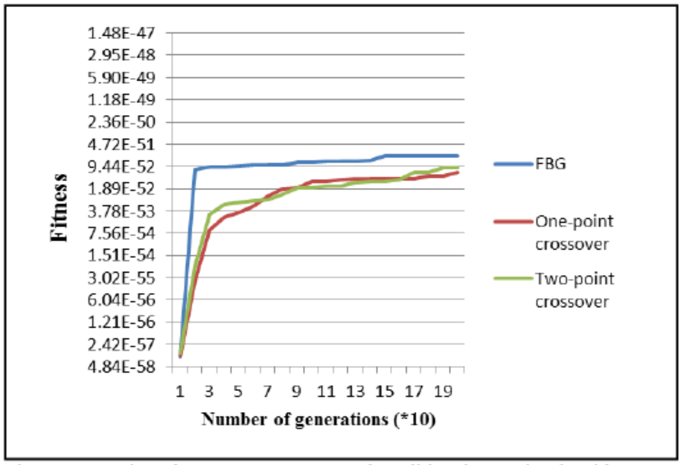

Fig. 6. Comparison between FBG-GA and traditional genetic algorithms, on fitness value.

VI. CONCLUSION

Given a business process consisting of different tasks, we studied the problem of forming a team of skillful experts which meet the requirements of all tasks, while ensuring smooth handover between individuals performing successive tasks. We defined two metrics, Specialty and Handover, to measure the level of expertise and how well team members may cooperate in handover of work. A method called BN-TF is proposed in this paper to address team formation problem in business process context, which starts by modeling a given business process as a Bayesian network, and then transform team formation problem into MPE problem. We developed an improved genetic algorithm called FBG-GA for solving the MPE problem more effectively in terms of time complexity. Through experiments on simulation, we demonstrated that the proposed algorithms produce more effective teams by comparison with existing methods, RarestFirst and CoverSteiner. We also compared FBG-GA with traditional types of genetic algorithms and proved its efficiency.

For effective team formation, it is also important to include workload of resources into consideration. Overloading on individual resources may lead to less efficiency of the assembled team in actual production. The presented study has not yet considered the issue of balancing the workload. Future work may extend the current research by taking into account more aspects as workload balancing, resource capacities, etc.

REFERENCES

[1] Lin S, Luo Z, Yu Y, et al. Effective Team Formation in Workflow Process Context[C]//Cloud and Green Computing (CGC), 2013 Third International Conference on. IEEE, 2013: 508-513.

[2] Baykasoglu A, Dereli T, Das S. Project team selection using fuzzy optimization approach[J]. Cybernetics and Systems: An International Journal, 2007, 38(2): 155-185.

[3] Zzkarian A, Kusiak A. Forming teams: an analytical approach[J]. IIE transactions, 1999, 31(1): 85-97.

[4] Wi H, Oh S, Mun J, et al. A team formation model based on knowledge and collaboration[J]. Expert Systems with Applications, 2009, 36(5): 9121-9134.

[5] Lappas T, Liu K, Terzi E. Finding a team of experts in social networks[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 467-476.

[6] Li C T, Shan M K. Team formation for generalized tasks in expertise social networks[C]//Social Computing (SocialCom), 2010 IEEE Second International Conference on. IEEE, 2010: 9-16.

[7] Anagnostopoulos A, Becchetti L, Castillo C, et al. Online team formation in social networks[C]//Proceedings of the 21st international conference on World Wide Web. ACM, 2012: 839-848.

[8] Van Der Aalst W M P, Reijers H A, Song M. Discovering social networks from event logs[J]. Computer Supported Cooperative Work (CSCW), 2005, 14(6): 549-593.

[9] Bomba D T, Prakash R. A description of handover processes in an Australian public hospital[J]. Australian Health Review, 2005, 29(1): 68-79.

[10] Kumar A, Dijkman R, Song M. Optimal resource assignment in workflows for maximizing cooperation[M]//Business process management. Springer Berlin Heidelberg, 2013: 235-250.

[11] McDonald D W. Recommending collaboration with social networks: a comparative evaluation[C]//Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2003: 593-600.

[12] Wolf T, Schröter A, Damian D, et al. Mining task-based social networks to explore collaboration in software teams[J]. IEEE Software, 2009, 26(1): 58-66.

[13] Sorkhi M, Hashemi S. Effective team formation in collaboration networks using vertex and proficiency similarity measures[J]. AI Communications, 2015, 28(4): 637-654.

[14] Kargar M, An A. Discovering top-k teams of experts with/without a leader in social networks[C]//Proceedings of the 20th ACM international conference on Information and knowledge management. ACM, 2011: 985-994.