The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

Productive and Profitable Cluster Hire

Parth Patel,[1] Kalyani Selvarajah,[1] Ziad Kobti,[1] Mehdi Kargar[2]
[1]University of Windsor, Windsor, Ontario, Canada
[2]Ryerson University,Ted Rogers School of Information Technology Management,
Toronto, Ontario, Canada.
(patel1in,selva111,kobti)@uwindsor.ca, kargar@ryerson.ca

Abstract

Cluster Hire is defined as a problem of hiring a group of experts to maximize profits with the ability to complete multiple projects simultaneously under a budget. It assumes that we have a set of projects which require skills and experts who possess various skills. The process of hiring a group of experts to complete a set of projects under the given conditions is proven to be the NP-hard problem. Individuals expect financial support (i.e.salary) which can be handled by a specific budget that we get, to work on the projects. In addition to maximizing the total profit, we are interested in hiring productive experts who can work many projects concurrently with effective result. Therefore, this paper examines the problem of hiring a cluster of experts, so that the total salary does not exceed more than a given budget and maximizes the total benefit of the projects that a highly productive team can cover collectively. We propose two greedy algorithms to solve this problem with different strategies. We illustrate the effectiveness of our approach by experimenting with the synthetic data sets. The results from a study of the synthetic dataset were compared with Bruteforce and Random Algorithm. It suggested that both our proposed ”Project greedy” and ”Expert greedy” algorithms performed well regarding both accuracy and run-time.

Introduction

Hiring a group of experts has become the most significant process in various applications. For instance, organizations, expert networks (e.g., LinkedIn, DBLP and GitHub) that connect skilled and experienced professionals, and freelancing websites (e.g., Upwork and Freelancer) are some real-world applications. The success of a task or project relies on the selection of adequate and effective team members. A suitable selection of members or experts should provide the required skills, which are essential for achieving the given tasks. At the same time, it should maximize the profit and enable an effective way of having the necessary costs including communication cost between members and expert cost (i.e. salary). The problem which is explained above is referred to as a Cluster Hire problem, and was introduced by Golshan et al.(Golshan, Lappas, and Terzi 2014) and has proved it is an NP-hard problem.

In reality, organizations or online job offering sites have many projects that need to be completed. Every project requires a set of skills, and a profit value is associated with it which is achieved on its completion. To work successfully on these projects, hiring effective experts is necessary. This study considers that every expert possesses a set of skills, capacity, productivity, and salary. The set of skills of an expert defines his or her expertise depending on experience and education. The capacity determines the number of times the expert can provide his/her expertise. The productivity value is evaluated based on the past performance of the expert. For example, among a group of researchers, the one that publishes the most is considered to be more productive. Clearly, we prefer to have a group of experts that have high productivity. To the best of our knowledge, no prior work has considered the productivity of the expert in cluster hire problem. A salary needs to be paid to the experts for providing their skills to perform a task. Therefore, a specific value of budget needs to be assigned to handle expert hiring. Our primary objective in this study is to hire as many experts as possible in which the sum of their hiring costs (i.e., salary) is under the given budget. Recently, Kargar et al.(Patel and Kargar 2017) proposed a Cluster Hire problem with communication cost. However, no one in the past research work considered the productivity of experts for forming teams.

In this work, we address the problem of the cluster hire in a group of experts to maximize profits and productivity. Since the proposed problem optimizes the two objective values, it turns out to be a bi-objective optimization problem. In addition to this, we consider two different version of Cluster Hire problem. One utilizes experts in order to participate in as many projects as possible at a time while another type focuses on the completion of the projects with as many experts as possible.

The main contribution of this study is to introduce the productivity score of the expert in the Cluster Hire problem. So this study focuses on how to hire a group of productive experts in order to complete a given set of projects. At the same time, the overall cost of hiring experts should be less than or equal to the given budget, and the profit needs to be maximized while no expert exceeds their capacity.

Related Works

Given a set of required skills to build teams of experts has been examined in many studies. Lappas et al.(Lappas, Liu, and Terzi 2009) first introduced the discovery of a team of experts from a social network. Then, the authors of (Kargar and An 2011) tested a new function called the sum of the distance to find the best teams. Later, Gajwar et al. (Gajewar and Das Sarma 2012) introduced another cost function based on the density of the induced sub-graph. The contribution by (Kargar, An, and Zihayat 2012), (Kargar, Zihayat, and An 2013) and (Zihayat, Kargar, and An 2014) are significant in order to have variant research of team formation problems.

Addition to this, the team formation problem was tackled by evolutionary computations in order to handle the complex expert network. The authors (Han et al. 2017) applied Genetic Algorithms to discover teams of experts and considered the geographical location of each member of the team while optimizing the approach. The authors (Selvarajah et al. 2017) used Cultural Algorithms in team formation problem for the first time. Recently, the authors (Selvarajah et al. 2018) considered the team formation problem in the health care setting and used Cultural algorithms to optimize multi-objectives.

The authors of (Anagnostopoulos et al. 2010) considered a set of experts where each expert is associated with a set of skills and a collection of projects arriving one at a time in an online form. (Golshan, Lappas, and Terzi 2014) proposed Cluster Hire problem for the first time, and followed the similar concept of (Anagnostopoulos et al. 2010). But, the difference was that they didn't choose projects from online and generated a single team that can perform many projects. Recently, the authors of (Patel and Kargar 2017) extended the work of (Golshan, Lappas, and Terzi 2014) by considering the previous collaboration among experts and optimizing the communication cost among experts. Probably our work is most related to (Patel and Kargar 2017). But the significant difference is that we didn't consider the communication cost between team members since we aim to get more productive experts which seem to be more important than previous collaboration in online freelancing work.

Problem Statement

Considering a set of n experts shown as $E = \{e_1, e_2, \ldots, e_n\}$ and a set of m skills shown as $S = \{s_1, s_2, \ldots, s_m\}$. Each expert e consists of a set of skills represented as $ES(e)$ where each skills is subset of S i.e., $\forall\ e \in E, ES(e) \subseteq S$. Each expert requires a salary $C(e)$ measured in dollar value to carry out different tasks. We assume that at most $Cap(e)$ times each expert can offer his or her expertise since an expert can not work more than their capacity. Given a set of projects $P = \{p_1, p_2, \ldots, p_k\}$ which requires a set of skills $PS(p_i) \subseteq S$ and $\forall\ p_i \in P$.

Given a set of experts E, a set of m skills S, and a set of k projects P, a team of experts $\mathcal{E} \subseteq E$ is able to complete a subset of projects $\mathcal{P} \subseteq P$. For all projects $\mathcal{P} \in P$, we assume that the team $\mathcal{E}$ covers $\mathcal{P}$ if the team $\mathcal{E}$ has the entire required skills for $\mathcal{P}$. In addition to this, for all experts $e_j \in \mathcal{E}$, we assume that $\mathcal{P}$ can provide his/her skills at most capacity $Cap(e)$ times.

Each expert e is also assigned a Productivity Score $PR(e)$. This score is determined based on the past performance of the expert. For example, among a group of researchers, the one that publishes the most is considered to be more productive. Clearly, we prefer to have a group of experts that have high productivity. The productivity of a group of experts is defined as follows:

Definition. ***(Productivity)*** *Given a group of experts E, the productivity of this group E is defined as follows:*

$$Productivity(\mathcal{E}) = \sum_{i=1}^{|\mathcal{E}|} PR(e_i)$$

On completion of each project, it brings profit $PF(p)$ for project p in dollar value. We need to select a set of projects in which the sum of the profit of entire projects needs to be maximized.

Definition. ***(Profit of Projects)*** *Given a set of projects P, the profit of finishing these projects is defined as follows:*

$$Profit(\mathcal{P}) = \sum_{p \in \mathcal{P}} PF(p)$$

We have a predefined budget B(measured in dollar value) to carry out a set of projects and to spent on hiring the experts. Our objective is to recruit as many experts as possible to cover the sum of their hiring costs (i.e., salary) within the given budget B. In addition to this, we aim to maximize the profit as well as the productivity of the group of experts. Therefore, our problem is a bi-objective optimization problem. It can be easily approached by converting it into a single objective problem. This can be achieved by introducing a trade-off parameter λ which varies between 0 and 1 and determines whether we want to assign more weight towards profit or productivity.

Problem. ***1*** *Given a set of n, number of experts E, a set of m skills S, a set of k projects P, a trade off λ between the profit and productivity, we are interested to choose a group of experts $\mathcal{E} \subseteq E$ and a set of projects $\mathcal{P} \subseteq S$ in which the following objective is maximized:*

$$PP(\mathcal{P}, \mathcal{E}) = (\lambda).Profit(\mathcal{P}) + (1 - \lambda)Productivity(\mathcal{E})$$

Additionally, the following budget constraint must be satisfied:

$$\sum_{e \in \mathcal{E}} C(e) \leq B$$

In the objective function, both the dollar values of the project's profit and the productivity of experts have different scales. Therefore, these values need to be normalized.

Discovering a team of experts to cover a set of projects $\mathcal{P} \subseteq P$ while maximizing $Profit(\mathcal{P})$ within the given budget B is proved to an NP-hard problem in (Golshan, Lappas, and Terzi 2014). Since the objective of Problem is linearly related to $Profit(\mathcal{P})$, then optimizing Problem is also an NP-hard problem.

Algorithms

This section describes two different algorithms to find a group of experts to cover a set of projects while maximizing the profit and productivity. The first algorithm selects an expert in each iteration and assigns the expert a set of projects according to his or her capabilities. The second algorithm selects a project in every iteration and selects the best group of experts for the project.

Expert Greedy Algorithm

Our first proposed greedy algorithm greedily chooses one expert in each iteration and assigns the experts to the pool of existing experts. While adding an expert to the pool, it checks whether the expert can cover any of the remaining projects or not. This algorithm tries to utilize expert as much as possible depending on the capacity of the expert while ensuring that the salary of the expert is within the specified budget B. It assigns a score to each pair of expert and projects and chooses the expert with the highest score in each iteration. The score is designing based on the *cheap salary* , *high profitable* project and *high productivity*.

A cheap expert might not have many skills in a profitable project, or the expert may not be very productive. To consider all these objectives, we design the following score function for each pair of projects p and experts e.

$$sc_e^p \leftarrow \lambda . \frac{PF(p). \min\{Skill(e,p),\ Cap(e)\}}{C(e)} + (1-\lambda).PR(e) \tag{1}$$

Here λ is the tradeoff parameter between profit and productivity where $Skill(e,p)$ is the number of required skills in project p that could be covered by the expert e. Note that after allocating an expert to the pool of existing experts, his/her capacity is updated based on the projects she participated.

The first part of equation 1 chooses a pair of expert and project in such a way that the project p has *high profit* and the expert e covers a maximum number of skills in p. This number is then, divided by the cost of the expert e which ensure that we consider the salary of the expert e. While selecting the expert, we choose the minimum value between the number of skills that expert e can be able to cover in p and the capacity of e. It is because we do not want to violate the capacity of the expert e and overload the expert with more projects.

Algorithm 1 is a solution to Problem 1 to find the best group of experts to cover a subset of projects while maximizing the profit and productivity. This algorithm receives the set of n experts, the set of m skills, the set of k projects, the tradeoff parameter λ and the budget B as the input. The output of the algorithm is the subset of projects $\mathcal{P}$ which are covered by the group of experts $\mathcal{E}$ while maximizing the objective of the problem and the sum of the salary of the experts in $\mathcal{E}$ is not more than the given budget B.

Algorithm 1 Cluster Hire with Expert Greedy Algorithm

Input: set of experts $E = \{e_1, e_2, \ldots, e_n\}$, set of skills $S = \{s_1, s_2, \ldots, s_m\}$, set of projects $P = \{p_1, p_2, \ldots, p_k\}$, λ, budget B, Capacity $C(e)$, Productivity $PR(e)$.
Output: subset of projects $\mathcal{P} \subseteq P$ and a group of experts $\mathcal{E} \subseteq E$ that maximize $PP(\mathcal{P}, \mathcal{E})$ under the given budget B.

$\mathcal{E} \leftarrow \emptyset, \mathcal{P} \leftarrow \emptyset, b \leftarrow 0$
while $b < B$ and $E/\mathcal{E} \neq \emptyset$ **do**
  $\mathcal{R} \leftarrow \{e \mid e \in E$ and $e \notin \mathcal{E}$ and $C(e) + b \leq B\}$
  for all $e \in \mathcal{R}$ **do**
    if e does not cover any required skills in $P/\mathcal{P}$ **then**
      remove e from $\mathcal{R}$
    end if
  end for
  if $\mathcal{R} = \emptyset$ **then**
    return $\mathcal{E}$ and $\mathcal{P}$
  end if
  for all $p \in P/\mathcal{P}$ **do**
    for all $e \in \mathcal{R}$ **do**
      if e covers at least one skill in p **then**
        $sc_e^p \leftarrow \lambda . \frac{PF(p). \min\{Skill(e,p),\ Cap(e)\}}{C(e)} + (1-\lambda).PR(e)$
      else
        $sc_e^p \leftarrow 0$
      end if
    end for
  end for
  $(e, p) \leftarrow \arg\max_{e \in \mathcal{R},\ p \in P/\mathcal{P}}\ sc_e^p$
  add e to $\mathcal{E}$
  assign skills of e to p based on rarest skill strategy
  update $Cap(e)$
  update $PS(p)$
  $b \leftarrow b + C(e)$
  if $|PS(p)| = 0$ **then**
    add p to $\mathcal{P}$
  end if
  while $Cap(e) > 0$ **do**
    $p' \leftarrow \arg\max_{p \in P/\mathcal{P}}\ score_e^p$
    assign skills of e to p' based on rarest skill strategy
    update $PS(p')$ according to $ES(e)$
    update $Cap(e)$
    if $|PS(p')| = 0$ **then**
      add p' to $\mathcal{P}$
    end if
  end while
end while
return $\mathcal{E}$ and $\mathcal{P}$

Project Greedy Algorithm

Our second proposed greedy algorithm is to find the group of experts to cover the subset of projects with the highest profit. It is designed based on the idea of selecting a project in each iteration. A score is assigned to each uncovered project and select the one with the highest score and added to the pool of projects in each iteration. The score of each project can be decided based on the *cheap salary*, *high profit* and *productive*.

Similar to the first strategy, the above scores are not necessarily compatible with each other. A low profitable project might need an expensive and productive set of experts, while a highly profitable project might need a moderately expensive expert. Hence, we design the scoring function that balance both combinations of all these objectives.

Algorithm 2 Cluster Hire with Project Greedy Strategy

Input: set of experts $E = \{e_1, e_2, \ldots, e_n\}$, set of skills $S = \{s_1, s_2, \ldots, s_m\}$, set of projects $P = \{p_1, p_2, \ldots, p_k\}$, λ, budget B, Capacity $C(e)$, Productivity $PR(e)$
Output: subset of projects $\mathcal{P} \subseteq P$ and a group of experts $\mathcal{E} \subseteq E$ that maximize $PP(\mathcal{P}, \mathcal{E})$ under the given budget B.

$\mathcal{E} \leftarrow \emptyset$, $\mathcal{P} \leftarrow \emptyset$, $b \leftarrow 0$
while $b < B$ and $P/\mathcal{P} \neq \emptyset$ **do**
  $P' \leftarrow P/\mathcal{P}$, $\mathcal{R} \leftarrow \{e \mid e \in E$ and $e \notin \mathcal{E}$ and $C(e) + b \leq B\}$
  if $\mathcal{R} = \emptyset$ **then**
    return $\mathcal{E}$ and $\mathcal{P}$
  end if
  for all $p \in P'$ **do**
    $E_p \leftarrow \emptyset$, $S_p \leftarrow PS(p)$, $\mathcal{R}' \leftarrow \mathcal{R}$
    while $S_p \neq \emptyset$ **do**
      for all $e \in \mathcal{R}'$ **do**
        if e covers at least one skill in p **then**
          evaluate the score sc_e using equation 1
        else
          $sc_e \leftarrow 0$
        end if
      end for
      $e \leftarrow \arg\max_{e \in \mathcal{R}'} \ sc_e$
      add e to E_p, **update** S_p
    end while
  end for
  for all $p \in P'$ **do**
    if $(\sum_{e \in E_p} C(e)) + b > B$ **then**
      remove p from P'
    end if
  end for
  if $P' = \emptyset$ **then**
    return $\mathcal{E}$ and $\mathcal{P}$
  end if
  $(p, E_p) \leftarrow \arg\max_{p \in P'} \ \lambda . \frac{PF(p)}{\sum_{e \in E_p} C(e)} + (1 - \lambda) . \sum_{e \in E_p} PR(e)$
  add p to $\mathcal{P}$, **assign** skills of experts in E_p to p
  for all $e \in E_p$ **do**
    add e to $\mathcal{E}$, **update** $Cap(e)$, $b \leftarrow b + C(e)$
    while $Cap(e) > 0$ **do**
      $s \leftarrow$ rarest skill in e which is required by a p in $P/\mathcal{P}$
      assign skill s to the most expensive p in $P/\mathcal{P}$
      update $Cap(e)$
    end while
  end for
end while
return $\mathcal{E}$ and $\mathcal{P}$

In each iteration, we find a set of experts Ep for each uncovered projects, to cover the required skills of p. For that, we use a modified version of the greedy weighted set cover algorithm. In the greedy set cover algorithm, a collection of skills is given (skills of each expert) in which each skill is associated with a price. The objective is to select a subset of sets to cover a given union set which is the set of skills required for a given project in this problem. In each iteration, the greedy weighted set cover algorithm maximizes the number of covered elements divided by the cost of the expert which is selected. We then add the productivity to the price per skill when selecting the next expert to cover a given project. We discover a set of experts for any remaining projects that can cover the required skills in the project in each iteration. For the set for project p, we begin with an empty set E_p. We then, select an expert to add to E_p that maximizes the following equation:

$$sc_e \leftarrow \lambda . \frac{\min\{Skill(e,p),\ Cap(e)\}}{C(e)} + (1-\lambda).PR(e) \quad (2)$$

The λ is the tradeoff parameter. The first part of equation 2 is chosen from the greedy set cover algorithm with a minor modification that takes the capacity of the expert in consideration. The other part of it calculates the productivity of the expert. After finding the set of experts E_p for all uncovered projects, we select one of the projects with the highest score and add to the pool of selected projects such that, the following equation is maximized.

$$\lambda . \frac{PF(p)}{\sum_{e \in E_p} C(e)} + (1 - \lambda) . \sum_{e \in E_p} PR(e) \quad (3)$$

The objective is to select the project that has a high profit, needs experts with low salary, and the set of experts responsible for performing the project's tasks have high productivity. Now, our proposed greedy Algorithm 2 returns a group of experts for performing a set of profitable projects with project greedy strategy. The input and output of algorithm 2 is similar to Algorithm 1.

Experiments

This section elaborates the performance of our proposed algorithms over the synthetic dataset which we generated randomly.

Synthetic Data

We generate synthetic data sets for our experiment. Our program (i.e. the coding) has been implemented in order to change the required numerical values to have the different type of dataset. For the expert details, we set the values for the number of experts; each expert randomly gets a specific number of skills and which will be assigned from the set of all skills. The capacity and productivity take values between a min and max randomly. For the project details, we set the values for the number of projects with the profit of completing the project. Each project randomly gets a specific number of required skills.

We randomly generated the dataset with the following values: the number of expert's skills randomly from 5 to 8, the productivity value is randomly from 1 to 10, the capacity

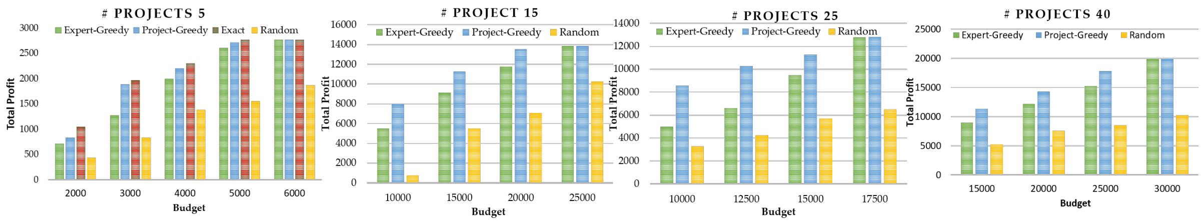

Figure 1: Comparison for Total profit vs. budget of Project Greedy, Expert Greedy with Exact and Random Algorithm

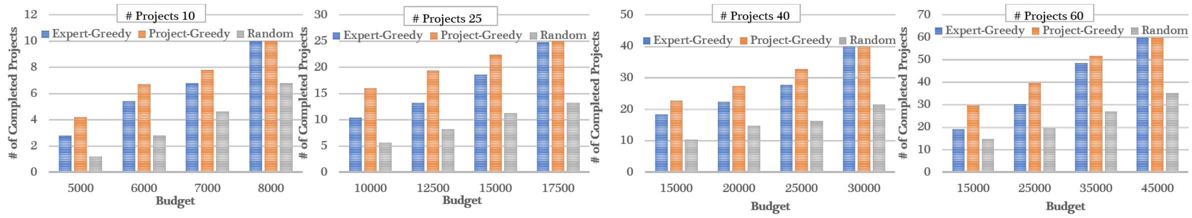

Figure 2: Comparison for the completed project vs. budget of Project Greedy, Expert Greedy with Random Algorithm

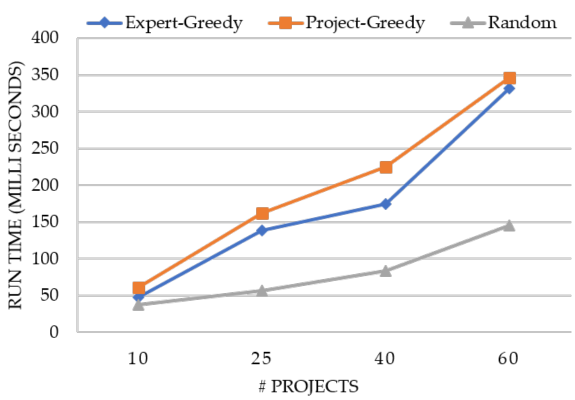

Figure 3: The run time of of Project Greedy, Expert Greedy and Random Algorithm when we have various number of projects.

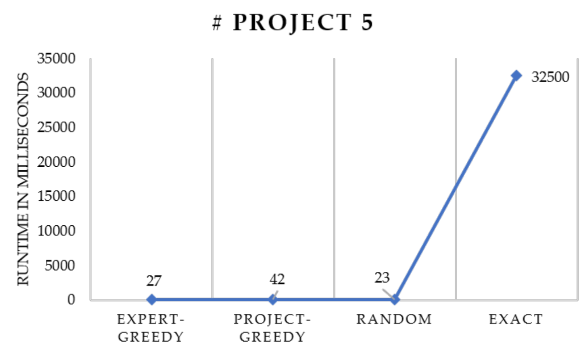

Figure 4: The run time of of Project Greedy, Expert Greedy, Random and Exact Algorithms when we have 5 projects.

of the experts is randomly from 3 to 6, and the salary of an expert is randomly from 500 to 550. We set the number of projects from 5 to 60. The profit of the project is generated randomly between 500 and 600. We run the experiments 10 times and record the average values. The default value of λ is assigned to 0.5 since we need to give priority to both the profit and productivity equally. Our experiments use the various range of values for the budget to see the total profit returned by each algorithm.

Performance Analysis

For the baseline comparison, we use the random algorithm. It selects a group of experts that can cover all required skills to complete the given projects without considering capacity, productivity, and profit. It only considers the Budget constraint and makes sure that the overall cost of hiring experts is less than or equal to given Budget B. We also compare the proposed algorithm with the exact algorithm for obtaining the results using an exhaustive search. We used Intel Core i7 2.6 GHz computer with 8 GB of RAM to implement our algorithms in Java.

We check the effect of the budget on the total profit of the projects as shown in fig 1. Each experiment is evaluated with k number of projects, in which $k = \{5, 15, 25, 40\}$. The graphs are plotted for total profit against various budget from 2000 to 30,000. The results indicate that Project Greedy achieves a higher overall profit than Expert Greedy when the budget is low. However, when the budget is high, both Project greedy and Expert greedy perform similarly. Both the project greedy and expert greedy outperformed the random algorithm.

Since the exhaustive search takes a very long time as the problem is NP-hard and the search space is exponential. We are able to run only for 5 projects with various budget as in fig1. However the run time was high compared to other algorithms as shown in fig 4.

Then, we tested the number of completed project vs budget as shown in fig 2 with default $\lambda = 0.5$. The result shows that both the project greedy and expert greedy behaves similarly as in the result from total profit vs. budget. The Project greedy completes more projects than Expert greedy when the budget is limited. For higher values of the budget, both algorithms complete the same number of projects or almost all projects are completed. At the same time, both Project greedy and Expert greedy outperform the Random algorithm. The exact algorithm couldn't perform for high number of projects because of run time.

Moreover, we checked the run time of the proposed two algorithms and random algorithms by varying the number of projects as shown in fig 3. Random algorithm took less time than the other two since it selects experts based on their skills. It did not minimize or maximize any objective. The project greedy took little more time than expert greedy algorithms as in the project greedy algorithm we have two iterations one internal and one external. External iteration selects the best project and the best expert associated with it. The internal iteration then finds remaining experts to complete the selected project in the external iteration. On the other hand, in expert greedy, we do not iterate internally.

Conclusions

Few researchers in the past have addressed the Cluster Hire problem which is to hire a group of experts to maximize profits with the ability to complete multiple projects within a given budget. This study examines the productivity of the experts for the projects which have maximum profit as an extension of previous works. Productivity is a significant concept since it considers the most efficient members to complete the project within budget. This study optimizes both profit and productivity. Therefore, it is a bi-objective problem and gives equal priority to both objectives by assigning 0.5 as a tradeoff value. To handle this NP-hard problem, we proposed two greedy algorithms in order to hire the best group of experts. For a certain value of budget, project greedy performs well than expert greedy. However, they both works similar to the high budget. The runtime is a little higher for project greedy compared to expert greedy. As a conclusion, the results from both algorithms are more close to the exact algorithm and better than the random algorithm.

In the future, we are planning to include communication cost as a another optimizing factor since it significant when considering teams. At the same time, we plan to consider Pareto-optimization to generate proper tradeoff values as well as optimize the problem.

References

Anagnostopoulos, A.; Becchetti, L.; Castillo, C.; Gionis, A.; and 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*, 599–608. ACM.

Gajewar, A., and Das Sarma, A. 2012. Multi-skill collaborative teams based on densest subgraphs. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, 165–176. SIAM.

Golshan, B.; Lappas, T.; and Terzi, E. 2014. Profit-maximizing cluster hires. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1196–1205. ACM.

Han, Y.; Wan, Y.; Chen, L.; Xu, G.; and Wu, J. 2017. Exploiting geographical location for team formation in social coding sites. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 499–510. Springer.

Kargar, M.; An, A.; and Zihayat, M. 2012. Efficient bi-objective team formation in social networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 483–498. Springer.

Kargar, M., and 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*, 985–994. ACM.

Kargar, M.; Zihayat, M.; and An, A. 2013. Finding affordable and collaborative teams from a network of experts. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, 587–595. SIAM.

Lappas, T.; Liu, K.; and 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*, 467–476. ACM.

Patel, M., and Kargar, M. 2017. Cluster hire in a network of experts.

Selvarajah, K.; Zadeh, P. M.; Kargar, M.; and Kobti, Z. 2017. A knowledge-based computational algorithm for discovering a team of experts in social networks.

Selvarajah, K.; Zadeh, P. M.; Kobti, Z.; Kargar, M.; Ishraque, M. T.; and Pfaff, K. 2018. Team formation in community-based palliative care. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*, 1–7. IEEE.

Zihayat, M.; Kargar, M.; and An, A. 2014. Two-phase pareto set discovery for team formation in social networks. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 2, 304–311. IEEE.