

# A Team Hiring Solution Based on Graph-Based Modelling of Human Resource Entities

Avinash Sharma, Jyotirmaya Mahapatra, Asmita Metrewar,  
Abhishek Tripathi, and Partha Dutta<sup>(✉)</sup>

Xerox Research Centre India, Bangalore, India

{Avinash.Sharma, Jyotirmaya.Mahapatra, Asmita.Metrewar,  
Abhishek.Tripathi3, Partha.Dutta}@xerox.com

**Abstract.** As modern organizations become more agile and support more complex business processes, acquiring the right set of talent is becoming crucial for their operations. One of the key talent acquisition problems is staffing a team that has requirement for multiple job descriptions, from a pool of external candidates. This team hiring problem may arise for (i) a new organization, (ii) a new group in an existing organization, or (iii) an existing group that faces high attrition level. This paper presents a Talent Acquisition Decision Support System (TADSS) that provides decision support for team hiring. The system first builds a weighted graph based model for the three types of Human Resource (HR) entities in the problem setup (jobs, employees and candidates), and the inter-relationship among them. Next, an algorithm based on spectral embedding of the HR Graph is used to select teams. The system then provides an interactive team selection and comparison interface based on the HR Graph. Simulation-based evaluations show the effectiveness of the proposed system in team formation.

**Keywords:** Human resource graph · Graph embedding

## 1 Introduction

**Motivation.** Talent Acquisition is one of the key challenges in modern organizations. Many organizations face increasingly complex business operations, people-intensive business process or high attrition level, which makes the talent acquisition problem especially cumbersome. Not surprisingly, significant research effort has been devoted to analyzing and automating parts of the talent acquisition process. In particular, sourcing of candidates and screening them automatically for interviews has been the focus of multiple recent commercial and research projects [5, 7].

Although sourcing and screening of individual candidates is one of the central topics in talent acquisition, most business operations need a team of employees with different job descriptions (JD). Creating a cohesive team is often necessary for efficient business operations. In this context, this paper studies the *team hiring* problem, where given a pool of candidates and a team with talent requirements for different job descriptions, the organization needs to select a team of one of more candidates for each

job description in the team requirement while ensuring a certain level of cohesiveness or affinity among the team members.

The team hiring problem has two main aspects. First, for each job description in the team, the available set of candidates needs to be ranked with respect to the role, to aid in screening by the hiring manager. Second, given a possible team formation, one needs to evaluate how effective the group of candidates will be as a team. Although the first aspect can be addressed by earlier works, such as [5, 7], the challenges to address the second aspect are three-folds: (i) Modelling the key Human Resource (HR) entities that are involved in the team hiring problem and their inter-relationships, (ii) exploiting information from *indirect* relationship between HR entities, which may have important information about team's affinities, and (iii) obtaining a cohesiveness score for a team based on the pairwise strength of relationship between the team members, in order to recommend one team formation over another.

**Contributions.** This work presents an interactive decision support system for selecting a team of candidates for a given set of job descriptions such that, not only are the selected candidates suitable for their respective job descriptions, but also the formed team is as cohesive or compatible as possible. The paper proposes a natural graph-based representation of the HR eco-system (called HR Graph), where the nodes are the HR entities (job descriptions, current employees and external candidates), and links capture the relationship between these entities (see Fig. 1). The advantage of such a graph-based modelling is that the information used for candidate ranking and team selection is not restricted to a specific job description and candidate resumes, but takes into account the other entities (e.g., existing employees, common connections or peers), and the strength of the connections or affinities between all these entities. Then, a spectral embedding of HR Graph-based algorithm is proposed to select teams with high cohesiveness.

This work complements the proposed HR Graph modelling information with an interactive team selection (Fig. 2) and comparison interfaces (Fig. 3). The team selection interface assists the user (e.g., hiring manager) in selecting one or more candidates for each job description in the team by providing a ranked list of candidates. The comparison interface comprises of intuitive decision scales which works as a model of cohesion scoring of different teams. The interface allows user to provide simultaneous feedback in terms of different affinity weightages. It also captures the information with regards to different decision biases and exceptions added in team selection.

Finally, the proposed system is evaluated with a numerical simulation using synthetic data generated for 1500 candidates. Results on team formation show that the system provides a good trade-off between the compatibility of selected candidates to job descriptions and the cohesiveness of the formed team.

**Related Work.** The social network of job candidates has been studied in sociology. For example, [6] studied the effect of a job seeker's social network, and strengths of her relationship (affinity) in the network on the employment outcome. However, these studies do not propose a method to exploit the social network for job search or team formation. Recent works have presented some methods on how to use online social networks to search for a job. For instance, [7, 8] use social network profiles of

candidates and employers to apply for jobs and find compatible jobs, whereas [9] ranks jobs for a candidate, based on the number of individuals in a candidate’s social network who are employees of the corresponding organization. However, none of the above works considers hiring a team of candidates while taking candidate affinities into account, and the affinities that arise due to either indirect connections or connections that are not captured in social network (e.g., relationship between two job descriptions).

Although, employment rates for social groups have been modelled in economic sociology [10], this body of work does not consider how to hire a team based on the studied models. More recent works have considered affinity between employees while creating teams in an organization [11]. Team formation using existing employees of an organization under various resource constraints has been extensively studied under workforce optimization in Operations Research [1]. These papers, however, rarely consider the problem of hiring external job candidates, and affinities arising out of indirect relationship between candidates.

## 2 System Overview

This section gives an overview of the Talent Acquisition Decision Support System (TADSS) proposed in this work. Next, the construction of HR Graph in TADSS is described.

The nodes in the HR Graph are the job descriptions in the team, job candidates, and the existing employees in the organization. One of the challenges for the HR Graph construction is determining the link weights between the nodes. Table 1 presents the different kind of relationship considered in HR Graph, and the parameters that determine the weights of these relationship links. The strength of relationship between two HR entities is captured by link weights: higher edge weight indicates stronger affinity.

**Table 1.** Lists of functions used for computing link weights

JD-JD	$f_{jj}$ (Job_Description)
candidate-candidate	$f_{cc}$ (Candidate_Profile, Social)
employee-employee	$f_{ee}$ (Employee_Profile, Job_Affiliation, Social)
JD-candidate	$f_{jc}$ (Historical_Decision, Candidate_Profile, Job_Description)
candidate-employee	$f_{ce}$ (Candidate_Profile, Employee_Profile, Social)
JD-employee	$f_{je}$ (Historical_Decision, Employee_Profile)

Edge weighing functions are described below.

1. *Employee-employee* weighing function ( $f_{ee}$ ) considers as parameters three meta-attributes, viz., Employee\_Profile, Job\_Affiliations and Social (Network Connections). Each of the meta-attributes has multi-level sub-attributes, e.g., Education as a sub-attribute of Employee\_Profiles. An example formulation of  $f_{ee}$  is given below, where [0.4, 0.4, 0.2] are attribute importance factors that always sum

to one:  $f_{ee}$  (Employee\_Profile, Job\_Affiliation, Social) =  $0.4 * \text{Employee\_Profile} + 0.4 * \text{Job\_Affiliation} + 0.2 * \text{Social}$ .

2. Candidate-candidate weighing function ( $f_{cc}$ ) formulation is very similar to  $f_{ee}$  with only two meta-attribute parameters and with different possible attribute importance factors, e.g., [0.7, 0.3].
3. JD-JD edge weighing function ( $f_{jj}$ ) considers similarity in content of input pair of JD's in terms of skills and experience required.
4. JD-candidate weighing function ( $f_{jc}$ ) considers Education, Experience and Skill overlap between JD and candidate profile. Although not considered in this work, historical hiring decisions, on candidates that were considered for the current JD or similar JDs, can also be used. Both these parameters can have real values and their weighted linear combination can be used to compute link weights.
5. JD-employee weighing function ( $f_{je}$ ) has similar definition with different weights.
6. Candidate-employee weighing function ( $f_{ce}$ ) computes a similarity score based on overlap in employee and candidate profiles as well as considering their social network connectivity, if available.

A detailed discussion on the practical aspects of the above edge weighing functions is presented in the section on HR Graph modelling parameters. Using these definitions, the HR Graph for TADSS can be induced which is then used for computing average connectivity (indicating the strength of relationship) among graph nodes. Figure 1 shows an example graph modeling of HRMS entities without link weights.

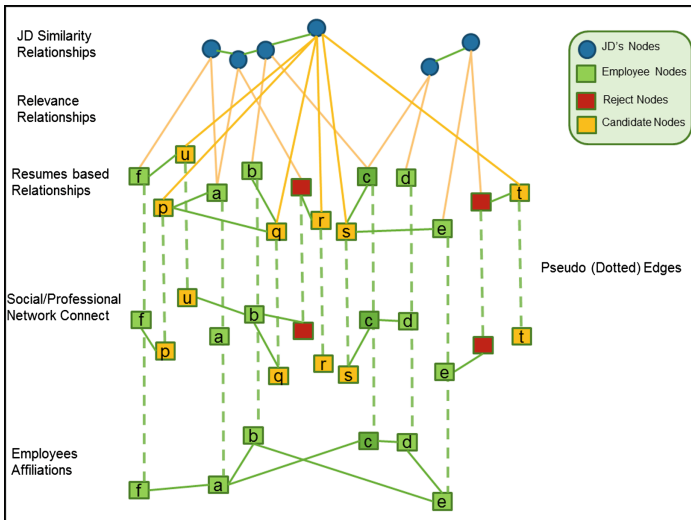


Fig. 1. Proposed graph representation

TADSS next computes an implicit representation of HR Graph known as *spectral or Laplacian embedding* [2]. The Laplacian embedding [3, 4] is a popular spectral

representation technique where each graph node is mapped to a K-dimensional space spanned by the first K non-null eigenvectors of the graph Laplacian matrix. As described later, TADSS uses the Laplacian embedding of HR Graph to select teams, and exploits the property that the lower Euclidean distances in the embedding space reflect stronger average connectivity between two graph nodes in the original graph.

### 3 TADSS User Interface

This section illustrates the TADSS user interface through a use case, and describes the associated algorithms to select teams. Consider the case of John who is a hiring manager in a mid-sized Information Technology (IT) services delivery firm. John has to staff and manage the IT support requirements for their new client. This use case scenario below explains how John interacts with TADSS to hire the best team configuration for a given project.

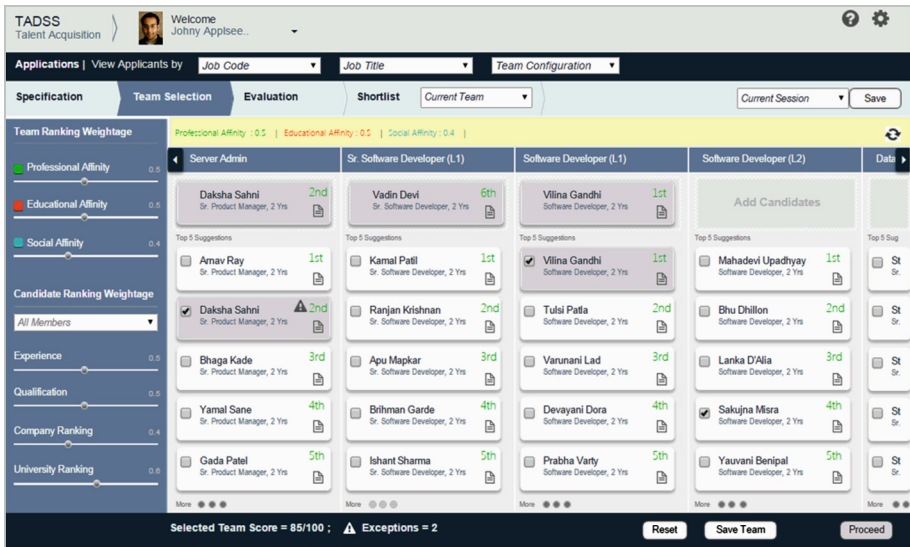


Fig. 2. Team selection interface

TADSS provides a four step wizard process for team creation. The first step, namely, the *specification interface* allows him to view, search, add and edit JD details against a large of pool of candidate profiles available with the company. He can use the application to specify his team configuration. In the second step, *team selection interface* as shown in Fig. 2, he can define Employee-Employee and JD-Employee weightage values for different HR entity attributes. Once John changes the affinity values, the preferred team-affinity weightages (Employee-Employee) and candidate ranking weightages (JD-Candidates) are used in the HR Graph construction.

Next top candidates with respect to each JD are found. The ranking algorithm considers John’s preference on education, experience and skill competencies of all candidates and provides a ranking solution. In the team selection interface, he can select the default suggested candidate or add any exception by overriding system-suggested team member. For a given session a user can save multiple team compositions. While the user is selecting team members, the total cohesiveness of the team is computed simultaneously and shown as *selected team score*.

John can select one or multiple candidates for a given JD as per the requirement of team and thereby generate multiple such team configurations. Once John has selected and saved multiple team configurations, he can proceed to *team evaluation interface* as shown in Fig. 3. He can compare the different team configurations based on the affinity score. The affinity score is computed using the HR Graph embedding based algorithm described later.

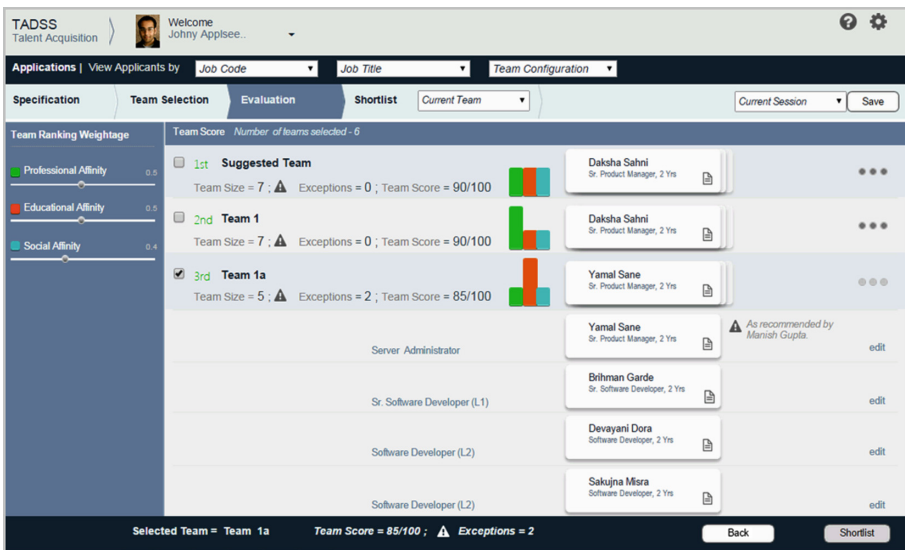


Fig. 3. Team evaluation interface

The team evaluation interface also provides the affinity visualization next to team cohesiveness score where he can compare the team capabilities. The suggested team and selected teams are shown as cluster packs which can be expanded and individual candidate resume can be seen along with any exception added to every candidate node. The interface can be further enhanced to include actions related to hiring decision workflow against every candidate card. The bar chart along with team score gives a quick preview of the cohesiveness value at various affinities. Next, the two main queries handled by the TADSS user interfaces are described.

**Candidate Ranking Query:** In this query, TADSS is asked to rank candidate profiles for a given job description. TADSS exploits the property that smaller Euclidian distance between two nodes in the Laplacian embedding of the HR Graph indicates stronger average connectivity (or stronger relationship) between the corresponding entities in the HR Graph. Thus, to answer the query for a given JD, TADSS simply ranks the candidates in increasing order of their distance from JD in the embedding space.

**Team Hiring Query:** In this query, TADSS system is asked to provide a cohesion score for a given set of candidates who are selected for a team. The system assumes the hypothesis that for forming a cohesive team, the members should have common or related academic background or past affiliation from previous jobs, or direct or indirect connections on social/professional media platforms. Based on the construction of the HR Graph, the closeness of the candidate nodes in the embedding space provides an indication of overall cohesion among them. Thus, this work proposes to compute the cohesion score as a 2-dimensional vector storing the standard deviation and average pairwise distance, of the embedded points associated with the set of candidates in the given team.

Let  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  be the  $K$ -dim Laplacian embedding of HR graph where  $\mathbf{x}_i = \{x_i^1, \dots, x_i^K\}$  represents  $K$  coordinates of a node. Let  $\mathbf{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_p\}$  be the  $p$  teams configured by the hiring manager. Each team instance is represented by a set of candidates (associated with respective JDs) as  $\mathbf{t}_i = \{\mathbf{x}_{\alpha(i,1)}, \dots, \mathbf{x}_{\alpha(i,m)}\}$  where the team has  $m$  members and  $\alpha(i, \cdot)$  is the index set of the respective selected candidates. Then the system computes the following for each team  $\mathbf{t}_i$ .

$$mean(\mathbf{t}_i) = \frac{1}{m} \sum_j \mathbf{x}_{\alpha(i,j)} \quad (1)$$

$$sd(\mathbf{t}_i) = \sqrt{\frac{\sum_j dist(\mathbf{x}_{\alpha(i,j)} - mean(\mathbf{t}_i))^2}{m}} \quad (2)$$

$$avg\_dist(\mathbf{t}_i) = \frac{1}{m^2} \sum_{l=1}^m \sum_{j=1}^m dist(\mathbf{x}_{\alpha(i,l)} - \mathbf{x}_{\alpha(i,j)}) \quad (3)$$

$$score(\mathbf{t}_i) = \begin{bmatrix} sd(\mathbf{t}_i) \\ avg\_dist(\mathbf{t}_i) \end{bmatrix} \quad (4)$$

On computing the cohesion score vector (where lower value in each dimension implies better team cohesion), the TADSS system either presents the vector as output to the query, or for ease of comprehension, a scalar normalized cohesion score value (as shown in Fig. 3) whose value increases with increased level of team cohesion, is obtained from the computed score vector.

## 4 Dataset Generation

This section describes the dataset generation for the numerical simulations. One of the challenges in talent acquisition analytics is the lack of public access to data due to privacy concerns. To overcome this challenge, this work uses synthetic data for simulations, and its generation procedure is described next.

The dataset generated has 20 job descriptions of 3 categories, from a professional network where all these job descriptions are related to J2EE domain. 1500 resume samples are generated, of which 1000 were chosen as candidates and the rest 500 as employees. The details are given in Table 2.

The structure of each resume is as follows:

1. Resume index
2. Education

List of degrees (masters, bachelors), Degree name, Degree category (management or technology), Institute, Year of graduation, and grade obtained

3. Experience

List of jobs, Position title, Organization name, Start year and End year

4. Educational skills
5. Experience skills (only for experienced candidates and employees)
6. Associated job description id (only for employees)

**Table 2.** Details of profile attributes of generated resumes

Candidate	Profile category	Position title	Resumes	Total
Experienced (60 %)	Manager/Lead (20 %)	Product manager (12 %)	180	300
		Technical lead (8 %)	120	
	Developer (20 %)	Software developer (12 %)	180	300
		Senior software engineer (8 %)	120	
	Tester (20 %)	Test engineer (12 %)	180	300
		Senior test engineer (8 %)	120	
Fresher (40 %)			600	600
<b>Total number of Resumes</b>				<b>1500</b>

**Education:** 60 % of the profiles have master's as their highest degree while rest have bachelor's as the highest degree. Educational degrees were divided into two categories: technology and management. All the degrees in technology category are assumed to be earned in Computer Science domain. Degrees which were considered include 4 types of bachelor's and 4 types of master's degrees in Computer Science, and 2 types of master's degrees in management. The schools and institutes are assigned by taking random samples from a list of 82 schools and 600 institutes, compiled from public



sources. All the bachelor's degrees are assumed to have a span of 4 years, while masters and those in management are assumed to have a span of 2 years. Obtained grades follow a normal distribution with mean of 87 and standard deviation of 3.

**Experience:** Random samples were generated for each of the 6 job profiles from a list of 42 companies in IT domain. Profiles in management category are assigned job experiences with a minimum of 3 and a maximum of 7 years whereas those in development and testing category have job experiences with a minimum of 1 and a maximum of 3 years. For the candidate profiles generated, 60 % had existing work experience, and remaining 40 % were fresher.

**Skills:** Two kinds of skills were assigned to the profiles, viz., educational and industrial skills, from 16 product management related skills, 39 development related skills and 39 testing related skills. All the development and testing skills are in the domain of J2EE. These lists were compiled by referring to similar job profiles from social professional networks. Each resume has a minimum of 5 and a maximum of 10 skills in educational and/or experience category. Random samples of skills from each category, conforming to the position titles of profiles were created. Fresher resumes are assigned a mix of testing and development skills.

## 5 HR Graph Modelling Parameters

The graph modelling was achieved by inducing an undirected weighted graph among 1520 nodes representing job descriptions (20), employee resumes (500) and candidate resumes (1000). The edge weights were computed using the functions in Table 1. The features considered while calculating edge weights were: (1) education, (2) experience, and (3) skill. Thus, for both candidate-candidate and job description-candidate edges, the total affinity, i.e., edge weight was calculated as:

$$w_{e_1, e_2} = weight_{edu} * affinity_{edu}(e_1, e_2) + weight_{exp} * affinity_{exp}(e_1, e_2) + weight_{skills} * affinity_{skills}(e_1, e_2)$$

where  $e_1$  and  $e_2$  were the two entities (either job description and candidate, or a pair of candidates). Each of these affinities was a weighted sum of sub-features, enumerated in subsequent sections. For JD-candidate edges, the feature weights were:  $weight_{edu} = 0.2$ ,  $weight_{exp} = 0.6$ ,  $weight_{skills} = 0.2$ . For candidate-candidate graph, the feature weights were:  $weight_{edu} = 0.3$ ,  $weight_{exp} = 0.4$ ,  $weight_{skills} = 0.3$ .

**Education Affinity:** For candidate-candidate edge weight, education affinity,  $affinity_{edu}$ , was calculated by finding match between degrees of the profiles in terms of degree category (technical or management), degree name, year of starting and graduation, and school/institute. Different levels of educational degrees were weighed differently. For JD-candidate graph, affinity was assigned the full value, i.e., 0.2, if the candidate's degree matched exactly to the required degree, 0.1 if it did not match but was under the same category (technology or management).

**Experience Affinity:** Experience affinity  $affinity_{exp}$  was calculated by finding match between position titles, company names, start and end year of job and years of experience of the two profiles in candidate-candidate graph. For JD-candidate graph, highest affinity was assigned to the most recent job position if it matches the title of the given job description, and relatively lower to the earlier ones. Sub-features, viz., organization name, position title and minimum years of experience were weighted as:  $weight_{organization} = 0.2$ ,  $weight_{positiontitle} = 0.2$ ,  $weight_{experience\_years} = 0.2$ .

**Skill Affinity:** Skill affinity  $affinity_{skills}$  was calculated as a weighted sum of matches for educational skills and experience skills. In the experiments, the weights were set as follows:  $weight_{educational\_skills} = 0.4$ ,  $weight_{experience\_skills} = 0.6$ . The similarity over skill sets of the two profiles or a job description and profile was calculated using Sorenson Dice coefficient [12].

All the affinities for an edge in candidate-candidate graph are normalized over the set of all the candidates using feature scaling, before summing them up to calculate the total affinity.

## 6 Experimental Results

The spectral embedding of the induced HR graph was obtained by computing the eigen-decomposition of associated graph Laplacian matrix where only  $K = 304$  (approximately 20 % of total number of eigenvectors, i.e., 1520) eigenvectors were computed.

Once the Laplacian embedding of the HR graph was computed, an example team configuration for a randomly selected company in the generated data was chosen with following details:

Job Profile	# of Positions
• Product manager	01
• Technical lead	01
• Senior software engineer	01
• Software developer	03
• Test engineer	02

For this specific team configuration, first relevance ranking of candidates for each job profile was computed by executing the candidate ranking query in TADSS using Laplacian embedding. Using this relevance ranking of candidates, 10 instances of team configuration were obtained, such that for the first team instance the top ranked candidates per job profile was chosen, and then the selection was shifted to one rank down for the next team instance. Next, team hiring query was executed to compute the team score for each of the 10 instances of team configuration and the obtained the respective score vector fields are listed in Table 3. It is interesting to note here that the score vector with the best value i.e., with minimum magnitude (highlighted in 3<sup>rd</sup> row) is actually

**Table 3.** Team score of difference team instances and respective overlap among academic background and past job experience of team members.

Team Instance	In-	Score		#Unique Acad. Institutes/#Total Acad. Institutes	#Unique Companies/#Total Companies
		$sd(t_i)$	$avg\_dist(t_i)$		
$t_1$		0.018393	0.015963	10/15=0.67	7/8=0.87
$t_2$		0.026681	0.028860	09/13=0.69	5/7=0.71
$t_3$		<b>0.011318</b>	<b>0.014273</b>	<b>10/16=0.62</b>	<b>3/6=0.50</b>
$t_4$		0.013921	0.017169	11/16=0.68	5/8=0.62
$t_5$		0.017852	0.022248	11/14=0.78	6/10=0.60
$t_6$		0.015173	0.019246	13/14=0.92	4/8=0.50
$t_7$		0.016809	0.021587	10/16=0.62	4/7=0.57
$t_8$		0.017947	0.022839	10/14=0.71	3/5=0.60
$t_9$		0.011646	0.014761	10/13=0.76	5/6=0.83
$t_{10}$		0.011699	0.014563	10/14=0.71	2/4=0.50

the third team instance instead of the first team instance where top relevance ranked candidates were chosen.

In order to establish further confidence on the score value, an approximate measure of cohesiveness was computed among candidates (selected in each team instance) based on their academic background as well as past experience. In the former case, the fraction of unique academic institution divided by the total number of academic institutes where candidates pursued their higher education was computed. Similarly in latter case, the fraction of unique companies divided by the total number of companies to which candidates were affiliated was considered. These cohesiveness measures appropriately reflect the importance of respective score values. However, these are very coarse approximation of cohesiveness as the HR Graph is a multi-relations model and there are more relationships which need to be explored for getting a better estimate of cohesiveness (e.g., employee to candidate cohesiveness based on academic background or past job history).

As mentioned earlier, this simulation shows an example where the team which is staffed with the top ranked candidates from the individual relevance ranking list, need not to be the most cohesive team. Thus, TADSS can help the hiring manager to choose a team with better cohesiveness as reflected by its team score.

## 7 Preliminary User Evaluation and Discussions

The different team compositions for a given set of JDs given by TADSS were used for a qualitative feedback of the *team evaluation interface*. Users were given extended view of each suggested team in the interface. They were also given resume documents of top 5 candidates with respect to each JD for team creation. It was observed that hiring managers implicitly used the information related to HR entities (professional,

educational and social) while coming up with team-member compositions. However, a team comparison view brings forth the importance of weightages provided by hiring managers. It was observed that users were referring the cohesiveness bar chart to understand the reasons for higher or lower affinity scores.

Converged teams require a lot of human decision making and TADSS aids HR practitioners and staffing specialists in making more informed decision in relatively less time. Such a system for creating team compositions can be a useful tool for hiring managers and project staffing experts. In case of hiring for a team, hiring managers may provide their own insights, based on past experiences, as user provided importance factor for different affinity factors. As part of future work, a larger and longitudinal user study of TADSS has been planned.

## References

1. Aksin, Z., Armony, M., Mehrotra, V.: The modern call center: a multi-disciplinary perspective on operations management research. In: *Production and Operations Management* (2007)
2. Chung, F.R.K.: *Spectral Graph Theory*. American Mathematical Society, Providence (1997)
3. Qiu, H., Hancock, E.R.: Clustering and embedding using commute times. *Pattern Anal. Mach. Intell.* **29**(11), 1873–1890 (2007)
4. Lehoucq, R.B., Sorensen, D.C., Yang, C.: *ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. Society for Industrial and Applied Mathematics, Philadelphia (1997)
5. Mehta, S., Pimplikar, R., Singh, A., Varshney, L., Visweswariah, K.: Efficient multifaceted screening of job applicants. In: *EDBT* (2013)
6. Hoye, G., Hooft, E.A.J., Lievens, F.: Networking as a job search behavior: a social network perspective. *J. Occup. Organ. Psychol.* **82**(3), 661–682 (2009)
7. <https://www.linkedin.com/company/bright.com>
8. Social networking job matching technology. United States Patent Application 20130013526
9. Method and apparatus for hiring using social networks. United States Patent Application 20110196802
10. Krauth, B.V.: A dynamic model of job networking and social influences on employment. *J. Economic Dyn. & Control* **28**(6), 1185–1204 (2004)
11. Determination of a contractor team. WIPO Patent Application WO/2013/187866
12. Dice, L.R.: Measures of the amount of ecologic association between species. *Ecology* **26**(3), 297–302 (1945)