9-2018

JobComposer: Career path optimization via multicriteria utility learning

Richard J. OENTARYO
Singapore Management University, roentaryo@smu.edu.sg

Xavier Jayaraj Siddarth ASHOK
Singapore Management University, xaviera@smu.edu.sg

Ee-peng LIM
Singapore Management University, eplim@smu.edu.sg

Philips Kokoh PRASETYO
Singapore Management University, pprasetyo@smu.edu.sg

Citation

OENTARYO, Richard J.; ASHOK, Xavier Jayaraj Siddarth; LIM, Ee-peng; and PRASETYO, Philips Kokoh. JobComposer: Career path optimization via multicriteria utility learning. (2018). *Workshop on Data Science for Human Capital Management (DSHCM2018), Dublin, Ireland, Europe, 2018 September 14*. 1-16.

Available at: https://ink.library.smu.edu.sg/sis_research/4257

JobComposer: Career Path Optimization via Multicriteria Utility Learning [⋆]

Richard J. Oentaryo, Xavier Jayaraj Siddarth Ashok, Ee-Peng Lim, and Philips Kokoh Prasetyo

[1] Mclaren Applied Technologies, 7 Temasek Boulevard, Singapore 038987
richard.oentaryo@mclaren.com
[2] Living Analytics Research Centre, Singapore Management University, Singapore 178902
{xaviera,eplim,pprasetyo}@smu.edu.sg

Abstract. With online professional network platforms (OPNs, e.g., LinkedIn, Xing, etc.) becoming popular on the web, people are now turning to these platforms to create and share their professional profiles, to connect with others who share similar professional aspirations and to explore new career opportunities. These platforms however do not offer a long-term roadmap to guide career progression and improve workforce employability. The career trajectories of OPN users can serve as a reference but they are not always optimal. A career plan can also be devised through consultation with career coaches, whose knowledge may however be limited to a few industries. To address the above limitations, we present a novel data-driven approach dubbed JobComposer to automate career path planning and optimization. Its key premise is that the observed career trajectories in OPNs may not necessarily be optimal, and can be improved by learning to maximize the sum of payoffs attainable by following a career path. At its heart, JobComposer features a decomposition-based multicriteria utility learning procedure to achieve the best tradeoff among different payoff criteria in career path planning. Extensive studies using a city state-based OPN dataset demonstrate that JobComposer returns career paths better than other baseline methods and the actual career paths.

Keywords: Career planning · multicriteria optimization · job transition

1 Introduction

Motivation. At some point in our life, we may raise questions such as "What is my career goal?" or "How can I achieve my career goal?" Developing a career plan is essential for the career success of individuals, giving them a sense of direction to follow and a way to track how their career is progressing. Career planning would also help workforce improve their employability and develop the right skills, especially in the face of uncertainties due to rapid shifts in global economy and technology.

The success of career planning depends on how much insights one has on the set of possible career trajectories. To gain such insights, one may conduct his or her own

[⋆] This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

research or seek consultation and counseling from some career coach(es). Both approaches are however challenging as they have to cope with dynamically-changing job and workforce landscape. To offer individuals career planning insights at the societal scale, there is a critical need to automate career planning as much as possible, thus motivating this research.

Fortunately, with the fast-rising adoption of online professional network (OPN) sites (such as LinkedIn, Xing and Viadeo), people start creating and sharing their professional profiles with others, using their profiles to connect with others as well as to explore new career opportunities. It is also possible to leverage on these OPNs to gain deeper and more comprehensive insights on career trajectories.

Objectives. In this paper, we therefore develop a new data-driven approach dubbed `JobComposer` for automated career path planning and optimization. The centerpiece of this approach is a multicriteria utility learning procedure that seeks to identify an optimal strategy for composing a career path emanating from a given origin job, by taking into account the trade-off among different payoff criteria simultaneously. Our goal is to assist job seekers and career coaches in making more informed career decisions and guidance, respectively, based on a data-centric view of job and workforce landscape.

To the best of our knowledge, the problem of optimizing (improving) existing career paths is new, and our `JobComposer` is the first attempt to address this problem via utility learning. Previous works on OPN-based career path recommendation [11,9] favor the more popular career paths that people take, implying an inherent assumption that common career paths are ideal. However, this is not necessarily true in reality, and so the utility values of these popular career paths may be suboptimal (see Section 5).

`JobComposer` works based on the premise that observed career paths may be suboptimal and can be further improved. The suboptimality can be due to the "incompleteness" of observed career paths as their users may discontinue further career for various reasons. For example, a user could decide to retire at an early age, or fail to acquire the required skills to advance to the next higher level job. Another possible reason of suboptimality is the subjectivity in career decisions made by users. For instance, an head of engineering who is no longer interested in senior management, may relinquish her position to become an engineer. Finally, the optimality of a career path involves multiple criteria, e.g., salary, job level, etc.. To realize this, `JobComposer` assembles career paths out of all possible job transitions found in observed career trajectories, and employs a new multicriteria utility learning procedure that jointly takes into account multiple payoff criteria when optimizing a career path. This allows us to capture the tradeoff among different—possibly competing—goals in career path planning. As individuals select their preferred criteria, `JobComposer` would return the corresponding optimized career paths. However, these career paths are not personalized as the preferred criteria differ from user features (e.g. skills, education level, etc.).

Contributions. We summarize our main contributions as follows: (a) We present a new problem formulation of career path optimization as a multicriteria utility learning task using career trajectories observed in OPN data, which distinguishes our work from previous works [18,9] that do not consider a set of payoff criteria in finding good career paths; (b) We develop an efficient decomposition-based iterative procedure that divides the multicriteria utility learning problem into multiple scalar optimization subproblems and optimizes them simultaneously, which has been shown to be efficient than non-decomposition-based methods; and (c) We conduct extensive experiments to evaluate the efficacy and validity of our approach in both quantitative and qualitative aspects.

Paper outline. In Section 2, we first survey related works on job transition analyses and career recommendation. Section 3 describes our dataset. We then elaborate the proposed `JobComposer` approach in Section 4. Section 5 presents the details of our empirical study using a city state-based OPN dataset. Section 6 concludes this paper.

2 Related Work

Research on job and workforce movements has been around for decades [17,10,13,5,6], which often involves analyzing various aspects of workforce such as age and wage growth. These studies, however, traditionally relied on surveys, census, and other data such as tax lists and population registers, which require time-consuming and costly (manual) efforts to collect. Moreover, the findings tend to be focused on selected workforce segments, and cannot be easily replicated in a larger population.

With the rise of OPN, there is now a growing interest to conduct research on online user data in order to understand job transition behavior and career growth. Wang *et al.* [18] proposed a hierarchical Bayesian model to predict the probability of a user making a job transition at a certain time. Cheng *et al.* [3] modeled job transition activities to rank influential companies. State *et al.* [16] analyzed the migration trends of professional workers into the US. Xu *et al.* [20] combined work experiences from OPNs and check-in records from location-based social networks to predict job change occasions. More recently, Chaudhury *et al.* [2] analyzed the growth patterns of the ego-network of new employees in companies. Xu *et al.* [21] generated and analyzed job transition networks to identify talent circles. Kapur *et al.* [7] employed the PageRank [14] algorithm on a graph of job transitions among companies to estimate the desirability ranking of the companies, and use this ranking to compute the ranking of universities.

Beyond job transition analysis, a few studies have been conducted that make use of the career paths (trajectories) of OPN users. Xu *et al.* [22] developed a sequence alignment method for quantifying the professional similarity between two career paths. Lou *et al.* [11] devised a career path planning method that utilizes the shortest path algorithm on a job hop graph, where the edge weight is the negative logarithm of the job transition probability. Most recently, Liu *et al.* [9] developed a multi-source learning framework that combines information from multiple social networks (i.e., sources) and models the career path of an individual.

While our work is most closely related to [11,9], it is noteworthy that the proposed `JobComposer` approach is based on a fundamentally different premise. In particular, the methods in [11,9] were designed to obtain a good fit to common/popular career trajectories observed in the OPN data. As such, they inherently assume that the common trajectories are ideal, which may not always hold in reality. In contrast, `JobComposer` works based on the premise that the observed career trajectories may be suboptimal and have room for improvements. `JobComposer` also goes further by incorporating multi-criteria utility-based learning and path generation features. As such, it is able to achieve a good balance among multiple goals involved in a career path planning process.

Table 1. Statistics of our OPN dataset

Count metric	Value
No. of user profiles	455,477
No. of career trajectories	57,784
No. of (unique) jobs	255,691
No. of (unique) job transitions	265,533

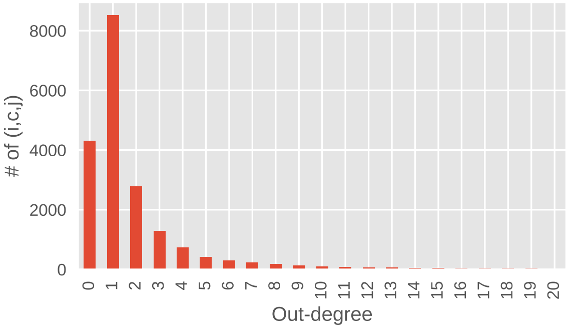

Fig. 1. Distribution of out-degree of jobs

3 Dataset

Dataset construction. We study career trajectories data collected from one of the largest OPNs in the city state of Singapore. We first crawled the directory of all public user accounts maintained by the target OPN. The dataset was constructed in November 2016. These public user profiles cover professional attributes of users including their education and career trajectory information. Each career trajectory consists of the sequence of jobs of a user and the respective companies. Companies come with different sizes and are assigned to the size categories: [2-10], [11-50], [51-200], [201-1000], [1001-5000], [5001-10000], and [10001+]. Each company also belongs to an industry.

Data cleaning. A job here refers to a job title in a certain sized company of an industry, as opposed to specific job title in some company. This allows the resultant career paths to be more appropriate for guiding users as opposed to specific job in specific company. To facilitate a proper empirical study, we perform two data cleaning steps. Firstly, we only include those jobs s, whose job titles j_s have a minimum support of 100 users. This removes jobs with unusual and misspelled titles. Secondly, we remove users who did not specify their education qualifications. We also remove all work experiences before one graduates from university or colleges. Table 1 summarizes the count statistics of the final dataset that we obtain after cleaning, and use throughout our study.

Out-transition degree of jobs. To determine the likelihood of out-transition of jobs in our dataset, we plot the out-transition degree of jobs in Figure 1 against the career paths observed. From the figure, we can observe that there are jobs with an out-degree of 0. These represent the last jobs in the career paths. These are usually the CEO and other very senior jobs. There are also many more jobs with out-degree greater than 1.

4 Proposed Approach

In this section, we first introduce our problem formulation that forms the basis of our JobComposer approach. Next, we describe the core ideas of JobComposer, namely: decomposition-based multicriteria utility learning and multicriteria utility selection. Finally, we present several payoff criteria used in the context of our OPN data.

4.1 Problem Formulation

First of all, we define a *career path* as a sequence of jobs taken by a person in a chronological order. Here, two adjacent/consecutive jobs in the sequence are assumed to have non-overlapping time period. Each *job s* is represented by a tuple (i_s, c_s, j_s), which denotes a job title j_s for a company size category c_s in industry i_s. Note that each company size category c_s represents the company type in terms of employee size, e.g., '1-50 employees' typically denotes a small startup, whereas '10,000+ employees' usually refers to a large, multi-national corporation (MNC).

Based on this definition, we formulate the following problem: "*Given a person's current (origin) job, how can we find an optimal career path that maximizes the total payoff possibly obtained by following that path*?" Under this formulation, we then treat career path as a stochastic process that undergoes transitions from one job (state) to another job (state). For computational simplicity, we particularly focus on a type of stochastic processes called *Markov process*, which exhibits a Markov property such that the next value of the process depends on the current value, but is conditionally independent of the previous values of the stochastic process [4]. In the context of career path planning, this means that a person's future job is stochastically independent of his/her jobs in the past, given his/her current job.

It is worth noting that our formulation can be viewed as a special instance of *Markov Decision Process* (MDP) [15]. One key difference is that the MDP formulation additionally includes the concept of action, which allows an agent to make a choice/decision that would affect the evolution of the agent's states over time. However, in our career planning context, only one action exists for the transition of one state (job) to another, which is hopping from one job j to another job j'. In this case, the next state is a deterministic (non-stochastic) function of the current state and action, and thus MDP reduces to a Markov process. Extended formulation of the current career planning problem to MDP is beyond the scope of this paper and would be left for future work.

Given the Markov process formulation, how do we then define the optimality of a career path beginning from some origin/source job s? Firstly, we consider a person who has to decide his/her path of destination jobs $\{d_1, \ldots, d_t, \ldots, d_T\}$, in order to maximize the discounted sum of future payoffs $r(d_{t-1}, d_t)$ that he/she can possibly get starting from s. The optimal value for this maximization process is given by the *value function*:

$$V(s) = \max_{\{d_1,\ldots,d_t,\ldots,d_T\}} \left[r(s, d_1) + \gamma r(d_1, d_2) + \ldots + \gamma^{T-1} r(d_{T-1}, d_T) \right] \quad (1)$$

where $s = d_0$ and $\gamma \in (0, 1]$ is the discount factor. The discount factor controls how myopic (or far-sighted) the selection of a career path is. In an economic context where the payoff is the amount of dollars made, a discount factor $\gamma < 1$ can be interpreted in terms of the idea that a dollar today is worth more than a dollar tomorrow.

Subsequently, the formula (1) can be rearranged to:

$$\begin{aligned} V(s) &= \max_{d_1} \left[r(s,d_1) + \max_{\{d_2,\ldots,d_T\}} \left[\gamma r(d_1,d_2) + \ldots + \gamma^{T-1} r(d_{T-1},d_T) \right] \right] \\ &= \max_{d_1} \left[r(s,d_1) + \gamma \max_{\{d_2,\ldots,d_T\}} \left[r(d_1,d_2) + \ldots + \gamma^{T-2} r(d_{T-1},d_T) \right] \right] \\ &= \max_{d} \left[r(s,d) + \gamma V(d) \right] \end{aligned} \tag{2}$$

where we equate $d = d_1$ for notational simplicity. This recursive definition is also known as the *Bellman equation*, which underpins dynamic programming (DP) theory [1]. Next, we define a *utility function* $U(s,d)$ for transition from job s to d such that:

$$V(s) = \max_{d} U(s,d) \tag{3}$$

This in turn leads to the following recursive formulation:

$$\begin{aligned} U(s,d) &= r(s,d) + \gamma V(d) \\ &= r(s,d) + \gamma \max_{d'} U(d,d') \end{aligned} \tag{4}$$

In the recursive formulation, d' denotes the next destination job that maximizes the overall utility. To solve equation (4), a simple way is to turn the equality symbol "$=$" into an assignment operator "$\leftarrow$" and keep iterating the assignments. This leads to the following *value iteration* procedure: For all job transitions (s,d), where s and d are the source and destination jobs respectively, repeat until terminating criterion:

$$U(s,d) \leftarrow r(s,d) + \gamma \max_{d'} U(d,d') \tag{5}$$

The above asynchronous DP update procedure is guaranteed to converge to the optimal utility $U^*(s,d)$, independent from the initial value of $U(s,d)$. We can prove the convergence by showing that, for large iteration k, $\lim_{k\to\infty} U(s,d)^{(k)} = U^*(s,d)$. To this end, we first quantify the distance between $U(s,d)^{(k)}$ and the optimal $U^*(s,d)$ using L_∞-norm, i.e., $||U^{(k)} - U^*||_\infty = \max_{(s,d)} |U(s,d)^{(k)} - U(s,d)^*|$. We then let $\mathcal{T}$ be the backup operator such that $\mathcal{T}(U(s,d)) = r(s,d) + \gamma \max_{d'} U(d,d')$. With this, we can show that:

$$\begin{aligned} ||U^{(k+1)} - U^*||_\infty \le ||\mathcal{T}(U^{(k)}) - \mathcal{T}(U^*)||_\infty \le \gamma ||U^{(k)} - U^*||_\infty \le \cdots \\ \le \gamma^{k+1} ||U^{(0)} - U^*|| \to 0 \end{aligned} \tag{6}$$

4.2 Decomposition-Based Multicriteria Utility Learning

The value iteration described in (5) operates based on a single type of payoff criterion. In practical career planning, however, one may be interested in multiple payoff criteria, such as job level gain, duration of stay in a company, income gain, etc. To account for

Algorithm 1 Decomposition-Based Multicriteria Utility Learning (MUL/D)

```
Inputs:
  Set of all job transition observations $\mathcal{D} = \{(s,d)\}$,
  Discount factor $\gamma \in (0,1]$,
  Maximum iteration $T_{max}$,
  Total number of weight combinations $N$
Output:
  Utility values $\{U(s,d,\boldsymbol{\lambda}^j) | \forall (s,d) \in \mathcal{D}, \forall \boldsymbol{\lambda}^j \in \boldsymbol{\lambda}\}$

  /* Initialization */
  Generate all $N$ possible weight combinations $\boldsymbol{\lambda} = [\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^j, \ldots, \boldsymbol{\lambda}^N]$
  Set $U(s,d,\boldsymbol{\lambda}^j) \leftarrow 0$ for all weight combinations $\boldsymbol{\lambda}^j \in \boldsymbol{\lambda}$ and job transitions $(s,d) \in \mathcal{D}$
  /* Utility update */
  repeat
    for each weight combination $\boldsymbol{\lambda}^j \in \boldsymbol{\lambda}$ do
      for each job transition $(s,d) \in \mathcal{D}$ do
        $r(s,d,\boldsymbol{\lambda}^j) \leftarrow \sum_{i=1}^{M} \lambda_i^j f_i(s,d)$
        $U(s,d,\boldsymbol{\lambda}^j) \leftarrow r(s,d,\boldsymbol{\lambda}^j) + \gamma \max_{d'} U(d,d',\boldsymbol{\lambda}^j)$
      end for
    end for
  until maximum iteration $T_{max}$
```

multiple career planning goals, we generalize the single-criterion approach described in (5) and propose a new multicriteria utility learning procedure.

To this end, we first decompose the multicriteria utility problem into a number of scalar optimization subproblems by considering a weighted combination of the different criteria. Let $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_i, \ldots, \lambda_M]^T$ denote a weight vector, such that $\lambda_i \geq 0, \forall i \in \{1, \ldots, m\}$ and $\sum_{i=1}^{M} \lambda_i = 1$, where M is the total number of criteria. Accordingly, we extend the payoff and utility functions by adding a third dimension that corresponds to a particular weight vector $\boldsymbol{\lambda}$. That is, the value iteration formula now becomes:

$$U(s,d,\boldsymbol{\lambda}) \leftarrow r(s,d,\boldsymbol{\lambda}) + \gamma \max_{d'} U(d,d',\boldsymbol{\lambda}) \tag{7}$$

where the payoff function $r(s,d,\boldsymbol{\lambda})$ is defined by:

$$r(s,d,\boldsymbol{\lambda}) = \sum_{i=1}^{M} \lambda_i f_i(s,d) \tag{8}$$

where the payoff function $r(s,d,\boldsymbol{\lambda})$ is defined by: $f_i(s,d)$ is the i^{th} payoff criterion for transition (s,d). We further elaborate the payoff criteria we have used in Section 4.4.

The expression given in (8) corresponds to the so-called *weighted summation* scalarization method [12]. Given a fixed weight vector λ, $r(s,d,\boldsymbol{\lambda})$ is a linear combination of $f_i(s,d)$. The value iteration procedure for $U(s,d,\boldsymbol{\lambda})$ will thus retain the same convergence property as equation (6). An empirical demonstration of the convergence trait of our multicriteria utility learning procedure is presented in Section 5.

Algorithm 1 summarizes our decomposition-based multicriteria utility learning procedure, or MUL/D in short. The original multicriteria utility career path optimization is broken down into multiple weighted criteria single utility optimization problems, each with a unique weight combination. Here, the number of criteria is M. The total number of possible weight combinations N is controlled by a positive (integer) parameter H. More precisely, $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^j, \ldots, \boldsymbol{\lambda}^N$ are all the weight vectors in which each individual weight λ_i^j $(1 \leq i \leq M-1)$ takes a value from $\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\}$. The final weight component, λ_M^j, is determined by:

$$\lambda_M^j = 1 - \sum_{k=1}^{M-1} \lambda_k^j \tag{9}$$

With this definition, the total number of possible weight vectors N for M different criteria is given by:

$$N = (H+1)^{M-1} \tag{10}$$

As an example, suppose we have three criteria (i.e., $M = 3$) and $H = 10$. In this case, the degree of freedom is $M - 1 = 2$, owing to the constraint $\sum_{i=1}^{M} \lambda_i = 1$. That is, the weight for the last (third) criterion can be computed as $\lambda_3 = 1 - \lambda_1 - \lambda_2$. The total number of weight vectors or weight combinations would therefore be $(H + 1)^{M-1} = 11^2 = 121$. By adjusting H alone, we can control both learning overhead and granularity of the optimal solutions.

4.3 Multicriteria Utility Selection

Once we learn through Algorithm 1 the utility values $U(s, d, \boldsymbol{\lambda_j})$ for all possible weight vectors $\lambda_j \in \boldsymbol{\lambda}$, the next question is: *How do we pick a weight vector λ^* that gives the best tradeoff/balance among different criteria*? To answer this, we introduce the idea that the best tradeoff/balance is one that allows best overall improvement of actual observed paths taken by individuals. With this, we present a new metric known as the *product of improvement means* (PIM) defined as:

$$\text{PIM} = \min_i \left(\text{sgn}(\mu_i)\right) \times \prod_{i=1}^{M} |\mu_i| \tag{11}$$

where sgn() is the sign function (i.e., $sgn(\mu) = +1$ when $\mu > 0$ and -1 otherwise), and μ_i is the mean of improvement of the optimized payoff criterion i over to the actual payoff criterion i:

$$\mu_i = \frac{1}{K} \sum_{k=1}^{K} \left(\sum_{(s,d) \in \mathbf{P}^{\text{optimized}}(k,s_0)} f_i(s,d) - \sum_{(s,d) \in \mathbf{P}^{\text{actual}}(k,s_0)} f_i(s,d) \right) \tag{12}$$

Here $\mathbf{P}^{\text{optimized}}(k, s_0)$ and $\mathbf{P}^{\text{actual}}(k, s_0)$ are the set of all job transitions in the optimized career path produced by `JobComposer` and those in the actual path taken by an individual k, respectively. In this case, both paths start from the same origin job s_0. The distinction between the two paths will be further illustrated in Section 5.

The term $\min_i(\text{sgn}(\mu_i))$ in (11) implies that, if any one of μ_i is negative, then the overall PIM value will be negative as well. This is to ensure that each improvement mean μ_i has a positive value, which is intuitive and desirable. Without this constraint, it is possible that two of the criteria give negative improvement means, but the overall PIM is positive due to double-negation of the (negative) signs of the corresponding μ_i during the product operation.

For a given weight vector λ^j, we have a PIM value PIM(λ^j). Accordingly, we identify the best tradeoff among different payoff criteria by picking a weight vector λ^* such that the following condition is met:

$$\lambda^* = \arg\max_{\lambda^j \in \boldsymbol{\lambda}} \text{PIM}(\lambda^j) \tag{13}$$

4.4 Payoff Criteria

In this work, we devise a few examples of payoff criterion $f_i(s,d)$, which tell us how rewarding a job transition is. These are payoff criteria that are derived from the career trajectories observed in our OPN dataset (which we will elaborate later). Our multicriteria utility learning approach is however not limited to these criteria. Depending on the available datasets, one could use criteria such as salary, medical benefits, etc.. Let p and s (or d) denote a person and a job, respectively. Our three criterion examples are defined as follows:

– **Duration cost** ($D_{s,d}$): This is our simplest payoff criterion; for a job pair (s,d), it refers to the *average time lapse* between the start date and end date of the source job s, over all people who move from job s to job d. In other words, a job transition which takes less time duration to the destination job is viewed as more favorable than another job transition which takes more time duration. Formally, we define the duration cost as:

$$D_{s,d} = \frac{\sum_{p \in \mathbf{K}_{s,d}} (end_{p,s} - start_{p,s})}{|\mathbf{K}_{s,d}|} \tag{14}$$

where $\mathbf{K}_{s,d}$ is the set of all people who hop from job s to job d, and $start_{p,s}$ and $end_{p,s}$ are the start and end dates of a job s that a person p takes, respectively. We note that the duration cost is computed based on the start/end dates of a source (instead of destination) job s, as our primary interest is a person's duration of stay at the source job before hopping to the destination job.

– **Level gain** ($L_{s,d}$): This refers to the difference between the levels of two jobs s and d, where the "level" of job s is estimated by computing the *average work experience* over all people who have job s in their career trajectories. Here, we assume that a job transition that gains much job level is favorable. In our data, there is no explicit job level information. The job level gain $L_{s,d}$ is thus defined by:

$$L_{s,d} = L_d - L_s = \frac{\sum_{p \in |\mathbf{K}_d|} w_{p,d}}{|\mathbf{K}_d|} - \frac{\sum_{p \in |\mathbf{K}_s|} w_{p,s}}{|\mathbf{K}_s|} \tag{15}$$

where $\mathbf{K}_s$ is the set of all people who ever took job s, and $w_{p,s}$ is the work experience of person p at job s. he work experience is the duration since the last graduation date of a person till the time at which he/she finishes a particular job. Intuitively, a job with longer average work experience implies that a longer time is required to achieve that position, and so we can expect it to be a high-level job. Examples of job with high level score L_s according to our OPN data are "Professor" and "Managing Director", whereas examples of job with low L_s are "Intern" and "Teaching Assistant". Also note that a positive (negative) gain $L_{s,d}$ can be loosely viewed as a "promotion" ("demotion")—in terms of the level of work experience required. We note that, although there is no ground truth available in our OPN data, our manual inspections show that the level gain provides a reasonable proxy for a promotion or demotion.

- **Desirability gain** ($R_{s,d}$). This is defined as the difference between logarithm of the desirability of two jobs s and d, where the "desirability" is computed using the *PageRank* algorithm [14,7]. In this case, we reinterpret the set of all job transitions $\mathcal{D}$ as a directed (unweighted) graph, where each node corresponds to a job s and each edge a job transition (s, d). PageRank views inbound edges as "votes", and the key idea is that "votes" from important nodes should carry more weight than "votes" from less important nodes [14]. As such, a job node that has high PageRank score reflects a "desirable" point where the flow of job transitions is heading to. We compute the desirability gain $R_{s,d}$ as:

$$R_{s,d} = \log(P_d) - \log(P_s) \tag{16}$$

where P_s is the PageRank score of job s. In this work, we employ a *weighted* version of PageRank [8], whereby the transition probabilities for each (source) node is proportional to the (out-)edge weights divided by the weighted out-degree of the node. In the context of this work, the weighted PageRank can be viewed as a measure of *global competitiveness*. That is, a job with high PageRank reflects a "desirable" destination where the flow of job transitions is heading to. In this case, we use hop volume as the edge weight, based on the intuition that the volume matters in determining where the flow goes to. To prevent inflated PageRank due to sink nodes (i.e., nodes with zero out-degree), we also allow a jump to any random node in the graph with a probability of α. Unless specified, we use $\alpha = 0.15$ by default in this work.

Using the above criteria, we construct a three-dimensional payoff vector $\boldsymbol{f}(s, d) = [f_1(s, d), f_2(s, d), f_3(s, d)] = [-D_{s,d}, L_{s,d}, R_{s,d}]$. In contrast to $L_{s,d}$ and $R_{s,d}$, we use the negated value of $D_{s,d}$, because we view the duration as a cost to be minimized.

5 Empirical Study

This section presents our empirical study on the dataset presented in Section 3. We first describe the evaluation metrics and baseline methods used in our study. We then present our quantitative and qualitative analyses. Our analyses aim to demonstrate that one can optimize career paths starting from any job considering the different criteria altogether, and to illustrate some optimized career paths through some case examples.

5.1 Metrics and Baselines

Combined metrics. We evaluate for each actual path P in our dataset in Figure 1 (57,784 of them), the quality of optimized path returned by our proposed multicriteria utility method compared with the quality of optimized path returned by some baseline method combining all the different payoff criteria. We define this path-specific metric based on *product of improvement means* (PIM). The path-specific PIM value comparing the path returned by a method denoted by *opt* for an actual path P is defined as:

$$\text{PIM}_P^{\text{opt}} = \min_i \text{sgn}(\mu_{P,i}) \times \prod_{i=1}^{M} |\mu_{P,i}| \tag{17}$$

where

$$\mu_{P,i} = \sum_{(s,d)\in\mathbf{P}^{\text{opt}}(P,s_0)} f_i(s,d) - \sum_{(s,d)\in\mathbf{P}^{\text{actual}}(P,s_0)} f_i(s,d) \tag{18}$$

We then compare our method with a baseline by computing $\text{PIM}_P^{\text{MUL/D}} - \text{PIM}_P^{\text{baseline}}$. This metric returns a positive value when our method returns a optimized path superior than that returned by the baseline method, and negative value otherwise.

Baseline methods. In this work, we evaluate our MUL/D-based `JobComposer` approach against several baselines. They consists of greedy based methods, methods that optimize single-criterion utility only for the purpose of finding the best performance one can achieve for each criterion, and a method assuming equal weights for different criteria. Specifically, the baseline methods include the following:

- **Greedy most common path**: Greedy path generation by following job transitions with the most number of people.
- **Greedy shortest duration path**: Greedy path generation by following job transitions with the lowest $D_{s,d}$.
- **Greedy level gain path**: Greedy path generation by following job transitions with the highest $L_{s,d}$.
- **Greedy desirability gain path**: Greedy path generation by following job transitions with the highest $R_{s,d}$.
- **Single-criterion utility path** $(-D_{s,d})$: Utility learning with negated $D_{s,d}$ as payoff criterion.
- **Single-criterion utility path** $(L_{s,d})$: Utility learning with $L_{s,d}$ as payoff criterion.
- **Single-criterion utility path** $(R_{s,d})$: Utility learning with $R_{s,d}$ as payoff criterion.
- **Equally weighted utility path**: Utility learning via linear scalarization of payoff criteria: $w_1 L_{s,d} - w_2 D_{s,d} + w_3 R_{s,d}$

The greedy methods are designed to select the job that maximizes some single criterion at every new job transition. Since they construct the overall path by making incremental decisions, they may not return the optimal path even for the criterion they try to optimize. The single criterion methods can address this shortcoming, but do not optimize for multiple criteria. The multi-criteria utility method to be compared with ours is the Equally weighted utility path method.

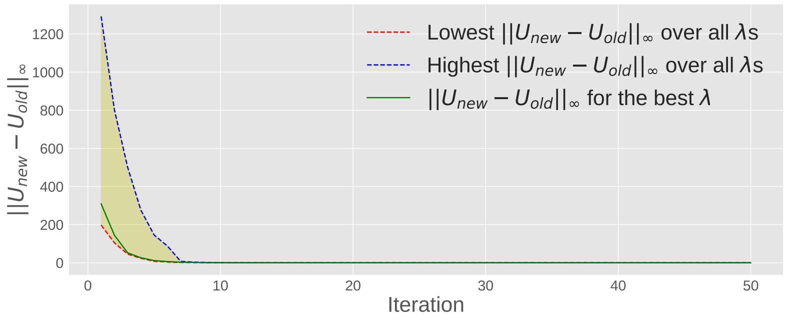

Fig. 2. Convergence of our multicriteria utility method

Parameter settings. For all the utility-based baselines (including ours), we set the discount factor and maximum iteration as $\gamma = 0.7$ and $T_{max} = 50$, respectively, which give the best results overall. For the equally weighted utility method, as we do not have one preferred criterion , we equalize the weights as: $w_1 = w_2 = 1$ and $w_3 = \beta = 500$. We set an equal weight for $D_{s,d}$ and $L_{s,d}$, since they have the same unit (i.e., months). Meanwhile, we use $w_3 = \beta$ to bring $R_{s,d}$ after to (roughly) the same scale. To test for the statistical significance of our method relative to the baselines, we use the *Wilcoxon signed-rank test* [19], which is a non-parametric statistical significance test for two paired samples, which cannot be assumed to be normally distributed. A low p-value below a threshold (typically 0.05 or 0.01) indicates that they are significantly different.

5.2 Quantitative Analysis

Convergence analysis. We firstly examine the convergence property of our MUL/D algorithm. In particular, We would like to show that our MUL/D method converges within a small number of iterations. The shaded region in Figure 2 shows the evolution of the L_∞-norm of the utility changes between two successive iterations $||U^{(k+1)} - U^{(k)}||_\infty$ of all possible weights in $\boldsymbol{\lambda}$ over iterations. The solid curve in the shaded region represents the L_∞-norm of λ^* as mentioned in (11), whereas the lower and upper dashed curves represent minimum and maximum L_∞-norm respectively. In terms of utility changes, we can see that our method converges to a stationary point within 10 iterations, regardless of the choice of λ. This provides an empirical justification for the convergence trait as per (6).

Comparison with baselines. First, we show the mean and median of $PIM_P^{\text{MUL/D}} - PIM_P^{\text{Baseline}}$ of actual paths when they are optimized by our method and some baseline method in Figure 3. The results show that our method outperforms all the baselines for most of the actual paths as $PIM_P^{\text{MUL/D}} - PIM_P^{\text{Baseline}}$ values are largely positive. In particular, our method is very much better than single-criterion utility method based on job level L, greedy desirability gain method and greedy most common method.

Next, Table 2 summarizes how well our method fares against the baselines according to the improvement in combined criteria measured by $PIM_P^{\text{optimized}} - PIM_P^{\text{actual}}$,

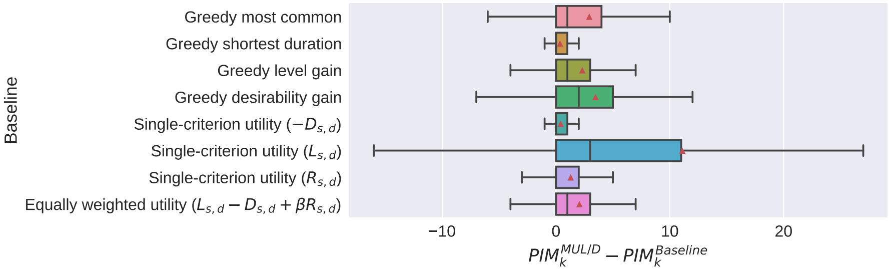

Fig. 3. Path-based combined metric improvements for different baseline methods

Table 2. Benchmarking results of different path planning methods

Method	**PIM**	$D^{\text{actual}} - D^{\text{optimized}}$		$L^{\text{optimized}} - L^{\text{actual}}$		$R^{\text{optimized}} - R^{\text{actual}}$	
		Mean	p**-value**	**Mean**	p**-value**	**Mean**	p**-value**
Greedy most common	-74.01	-88.80	0.00e+00 (-)	11.67	0.00e+00 (-)	-0.07	0.00e+00 (-)
Greedy shortest duration	-19.46	23.06	0.00e+00 (+)	2.67	0.00e+00 (-)	-0.32	0.00e+00 (-)
Greedy level gain	-255.37	-28.39	0.00e+00 (-)	26.84	0.00e+00 (+)	-0.34	0.00e+00 (-)
Greedy desirability gain	-1057.25	-57.38	0.00e+00 (-)	19.19	3.34e-04 (-)	0.96	0.00e+00 (+)
Single-criterion utility ($-D_{s,d}$)	-198.45	61.30	0.00e+00 (+)	-6.55	0.00e+00 (-)	-0.49	0.00e+00 (-)
Single-criterion utility ($L_{s,d}$)	-2478.66	-90.45	0.00e+00 (-)	81.52	0.00e+00 (+)	-0.34	0.00e+00 (-)
Single-criterion utility ($R_{s,d}$)	-240.31	-36.65	0.00e+00 (-)	5.96	0.00e+00 (-)	1.10	0.00e+00 (+)
Equally weighted utility	-577.01	-42.83	0.00e+00 (-)	12.31	0.00e+00 (-)	1.09	0.00e+00 (+)
Multicriteria utility	241.73	16.36	-	19.51	-	0.76	-

(+): significantly better than multicriteria utility; (-): significantly worse than multicriteria utility

and in single criteria (i.e., duration cost, level gain, or desirability gain) of the optimized career path returned by the method against that of the observed career path. $PIM^{\text{optimized}}$ and PIM^{actual} denote the PIM value of optimized path and actual path respectively. For single criteria such as duration cost. The duration cost of a path is defined as $D_{\mathbf{P}} = \sum_{(s,d)\in\mathbf{P}} D_{s,d}$, and $D_P^{\text{actual}} - D_P^{\text{optimized}}$ denotes the improvement in duration cost. Other single criteria-based improvement can be defined in a similar manner. As shown in Table 2, our MUL/D approach returns the best combined performance over different criteria measured by the average improvement of PIM. It is the only method that return positive PIM value. It is also the only method that returns positive mean and median improvement in every criteria. The greedy and single-criterion methods tend to do well only on one specific criterion, but not the others. Hence, they could return positive gains in one criterion but negative ones in other criteria.

Compared to the equally weighted utility method, our MUL/D approach gives better mean for $PIM_P^{\text{optimized}} - PIM_P^{\text{actual}}$, $D^{\text{actual}} - D^{\text{optimized}}$ and $L^{\text{optimized}} - L^{\text{actual}}$, though the former is superior in terms of $R^{\text{optimized}} - R^{\text{actual}}$. This result is reasonable as the equally weighted utility method is a reasonably strong baseline. Table 2 also includes the two-sided p-values of the Wilcoxon test. It is evident that all p-values are very small (i.e., $\ll 0.01$). Hence, the performance differences between our method with the other baselines are statistically significant. While above mentioned baselines are conceptually weaker, they are not necessarily easy to beat in practice.

Table 3. Examples of career path optimization results

#1	**Actual path**: (Telecom., 10,001+ employees, Technical Officer) → (Telecom., 5001-10K employees, Engineer) → (Telecom., 10,001+ employees, Associate Eng.)
	Greedy most common path: (Telecom., 10,001+ employees, Tech. Officer) → (International Trade & Development, 5001-10,000 employees, Tech. Manager)
	Multicriteria utility path: (Telecom., 10,001+ employees, Tech. Officer) → (Telecom., 5001-10K employees, Eng.) → (Telecom., 5001-10K employees, Project Leader)
	Metric improvement for greedy most common path: $D^{\text{actual}} - D^{\text{greedy most common}} =$ -72.00, $L^{\text{greedy most common}} - L^{\text{actual}} =$ 107.15, $R^{\text{greedy most common}} - R^{\text{actual}} =$ -0.06
	Metric improvement for multicriteria utility path: $D^{\text{actual}} - D^{\text{multicriteria utility}} =$ 70.00, $L^{\text{multicriteria utility}} - L^{\text{actual}} =$ 100.73, $R^{\text{multicriteria utility}} - R^{\text{actual}} =$ 0.04
#2	**Actual path**: (Banking, 10,000+ employees, Analyst) → (IT & Services, 5001-10,000 employees, Analyst) → (IT & Services, 5,000 - 10,000 employees, Team Leader) → (Banking, 10,000+ employees, Credit Risk Analyst)
	Greedy most common path: (Banking, 10,000+ employees, Analyst) → (Banking, 10,000+ employees, Associate) → (Banking, 10,000+ employees, Assistant Vice President) → (Banking, 10,000+ employees, Vice President) → (Banking, 10,000+ employees, Director) → (Financial Services, 10,000+ employees, Executive Director) → (Banking, 10,000+ employees, Managing Director)
	Multicriteria utility path: (Banking, 10,000+ employees, Analyst) → (Banking, 10,000+ employees, Consultant) → (IT & Services, 10,000+ employees, Technical Consultant) → (IT & Services, 10,000+ employees, Project Manager) → (Financial Services, 10,000+ employees, Assistant Vice President)
	Metric improvement for greedy most common path: $D^{\text{actual}} - D^{\text{greedy most common}} =$ -165.24, $L^{\text{greedy most common}} - L^{\text{actual}} =$ 160.56, $R^{\text{greedy most common}} - R^{\text{actual}} =$ 0.66
	Metric improvement for multicriteria utility path: $D^{\text{actual}} - D^{\text{multicriteria utility}} =$ 56.00, $L^{\text{multicriteria utility}} - L^{\text{actual}} =$ 78.89, $R^{\text{multicriteria utility}} - R^{\text{actual}} =$ 1.09
#3	**Actual path**: (Semiconductors, 10,001+ employees, Process Eng.) → (Semiconductors, 201-500 employees, Process Eng.) → (Semiconductors, 5001-10,000 employees, Equip. Eng.) → (Semiconductors, 201-500 employees, Field Appln. Eng.)
	Greedy most common path: (Semiconductors, 10,001+ employees, Process Engineer) → (Semiconductors, 10,001+ employees, Senior Process Eng.)
	Multicriteria utility path: (Semiconductors, 10,001+ employees, Process Eng.) → (Banking, 10,001+ employees, Rel. Manager) → (Banking, 10,001+ employees, Associate Director) → (Banking, 10,001+ employees, Managing Director)
	Metric improvement for greedy most common path: $D^{\text{actual}} - D^{\text{greedy most common}} =$ 79.75, $L^{\text{greedy most common}} - L^{\text{actual}} =$ 21.68, $R^{\text{greedy most common}} - R^{\text{actual}} =$ 0.59
	Metric improvement for multicriteria utility path: $D^{\text{actual}} - D^{\text{multicriteria utility}} =$ 67.42, $L^{\text{multicriteria utility}} - L^{\text{actual}} =$ 105.44, $R^{\text{multicriteria utility}} - R^{\text{actual}} =$ 1.39

Note: The format of a job is (i, c, j), where i is the industry code, c is the company size category, and j is the job title.

Runtime analysis. In the above quantitative analysis, MUL/D required less than one and a half hour to complete the analysis of 265,000+ job transitions. With parallel computation, the time required can be further reduced significantly. We nevertheless leave this detailed runtime analysis to future work.

5.3 Qualitative Analysis

We also study a few career path examples in detail in order to qualitatively evaluate the career path recommendations made by `JobComposer`. Specifically, we look at examples whereby $D^{\text{actual}} - D^{\text{multicriteria utility}} \geq 50$ months, $L^{\text{multicriteria}} - L^{\text{actual}} \geq 70$ months, and $R^{\text{multicriteria utility}} - R^{\text{actual}} \geq 0$. These are the examples that can show significant improvement according to our chosen multiple criteria. Table 3 shows three such examples. For comparison, we also show the corresponding recommendations made by the greedy most common path method. It can be seen that our multicriteria utility learning approach yields better metric improvements (at least two metrics) in comparison to the greedy most common path method.

It is interesting see how our method can devise optimized paths that are sensible and yet yield high improvements. The first examples illustrates a person who begins his career as a Technical Officer in Telecommunications Industry with a large company having more than 10,000 employees and remains in the same industry for entry-level engineering roles with similar company sizes. In this case, though the greedy most common method provides a higher level gain by switching to a lower company size in the Internet Industry, it takes 72 additional months and the desirability gain is less. In contrast, our method suggests to remain in the same Telecommunications Industry but in a relatively smaller company (5000-10,000 employees) ending up as a project leader, saving 70 months and having higher desirability gain.

In the second example, an Analyst in a very large bank, following the path suggested by our method (which advises to not immediately switch to a different Industry) saves 56 months and will end up as an Assistant Vice President in Financial Services Industry (higher level and desirability gains). Though the path suggested by greedy most common method has much higher level gain, the person has to wait for around 165 months to achieve it and also the desirability gain is much lesser than that of our method.

The last example shows how a person working in a large Semiconductor company can improve his career by switching to Banking Industry as a Relationship Manager, ending up as a Managing Director of a large bank (higher level and desirability gain) and saves around 67 months. Following the greedy most common method, the person instead of switching to different smaller companies in the same industry, could get a promotion in the same industry of same company size, saving 80 months. Comparing to our method, the greedy most common method has slightly better duration gain, but it has much lesser level and desirability gains.

In sum, our qualitative study shows that `JobComposer` produces sensible career path recommendations, with better quality than those of the greedy most common path, also justifying our earlier hypothesis that the common paths may not be always optimal and can be improved. Nevertheless, it must be noted that all the paths recommended in Table 3 only show what is possible, but do not necessarily guarantee that the next job in the path can always be achieved. Constraining the path to available openings and estimating the probability of successfully obtaining a career is beyond the scope of this paper, but is certainly the direction of our future work.

6 Conclusion

In this paper, we put forward a new data-driven approach for automated career path planning and optimization called `JobComposer`. To realize `JobComposer`, we first formulate career path optimization as a utility learning problem on top of a Markov process. Based on this formulation, we then develop a novel decomposition-based multicriteria utility learning and multicriteria utility selection procedures to devise a career path that optimizes multiple criteria simultaneously and identify the best tradeoff among different criteria, respectively. Comprehensive quantitative and qualitative studies using a city state-based OPN data have demonstrated the efficacy of our approach.

While `JobComposer` offers a powerful methodology for career path planning, there remains room of improvements. To endow a greater level of personalization within its career optimization processes, we would like to develop a more advanced method that utilizes a richer set of user features (e.g., skills, education level, etc.). Additionally, we plan to augment relevant auxiliary data—such as employees' salaries—and conduct studies with a more comprehensive set of payoff criteria.

References

1. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton, NJ, USA (1957)
2. Chaudhury, A., Kim, M., Tiwari, M.: Importance of first steps in a community for growth, diversity, and engagement. In: WWW. pp. 19–20 (2016)
3. Cheng, Y., Xie, Y., Chen, Z., Agrawal, A., Choudhary, A., Guo, S.: JobMiner: A real-time system for mining job-related patterns from social media. In: KDD. pp. 1450–1453 (2013)
4. Elson, C., Rozanov, Y.: Markov Random Fields. Springer (2012)
5. Fuller, S.: Job mobility and wage trajectories for men and women in the united states. American Sociological Review **73**(1), 158–183 (2008)
6. Joseph, D., Boh, W.F., Ang, S., Slaughter, S.: The career paths less (or more) traveled: A sequence analysis of IT career histories, mobility patterns, and career success. MIS Quarterly **36**(2), 427–452 (2012)
7. Kapur, N., Lytkin, N., Chen, B.C., Agarwal, D., Perisic, I.: Ranking universities based on career outcomes of graduates. In: KDD. pp. 137–144 (2016)
8. Langville, A.N., Meyer, C.D.: A survey of eigenvector methods for web information retrieval. SIAM Review **47**(1), 135–161 (2005)
9. Liu, Y., Zhang, L., Nie, L., Yan, Y., Rosenblum, D.S.: Fortune teller: Predicting your career path. In: AAAI. pp. 201–207 (2016)
10. Long, J., Ferrie, J.: Labour mobility. The Oxford Encyclopedia of Economic History (2003)
11. Lou, Y., Ran, R., Zhao, Y.: A machine learning approach for future career planning. Technical report, Stanford University (2010)
12. Miettinen, K.: Nonlinear multiobjective optimization. Kluwer Academic Publishers (1999)
13. Moscarini, G., Thomsson, K.: Occupational and job mobility in the US. The Scandinavian Journal of Economics **109**(4), 807–836 (2007)
14. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (1999)
15. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, NY, USA, 1st edn. (1994)
16. State, B., Rodriguez, M., Helbing, D., Zagheni, E.: Migration of professionals to the US: Evidence from LinkedIn data. In: SocInfo. pp. 531–543 (2014)
17. Topel, R.H., Ward, M.P.: Job mobility and the careers of young men. Working Paper 2649, National Bureau of Economic Research (July 1988)
18. Wang, J., Zhang, Y., Posse, C., Bhasin, A.: Is it time for a career switch? In: Proceedings of the International Conference on World Wide Web. pp. 1377–1388 (2013)
19. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**(6), 80–83 (1945)
20. Xu, H., Yu, Z., Xiong, H., Guo, B., Zhu, H.: Learning career mobility and human activity patterns for job change analysis. In: ICDM. pp. 1057–1062 (2015)
21. Xu, H., Yu, Z., Yang, J., Xiong, H., Zhu, H.: Talent circle detection in job transition networks. In: KDD. pp. 655–664 (2016)
22. Xu, Y., Li, Z., Gupta, A., Bugdayci, A., Bhasin, A.: Modeling professional similarity by mining professional career trajectories. In: KDD. pp. 1945–1954 (2014)