Bilkent University

Semester of Spring 2025

Department of Industrial Engineering

IE 400 Principles of Engineering Management

Term Project Report

Team 01

1.	Kadir Hakan AKBAY	22102372
2.	Mısra Simge UYSAL	22201527
3.	Deniz Tuna ONGUNER	22001788

Section: 02

Submission Date: May 11, 2025

This report is submitted to the Department of Industrial Engineering at Bilkent University in partial fulfillment of the requirements for the course IE 400 Principles of Engineering Management.

1 Introduction

Due to the competitive market structure in today's world, a way to find effective job matching is crucial for employers as well as job seekers. The platform JobSync is designed for this challenge using optimization techniques that match job seekers with suitable job openings. Within this project, two Integer Linear Programming (ILP) models are developed and implemented to facilitate and improve the matching process using the dataset of job seekers, job vacancies, and location distances. In the first model, the total priority weight of matched jobs is maximized, and the compatibility between seekers and jobs is ensured in terms of job type, salary, skills, experience, and location preferences. The second model is focused on minimizing the maximum dissimilarity between matched pairs based on a 20-question survey and maintains a minimum threshold of total priority weight. Using Gurobi, both of these models are solved, and the trade-off between compatibility and diversity of preferences is analyzed. Thus, a more intelligent, efficient, and fair job assignment strategies are achieved.

2 Part 1 - Maximizing Total Priority Weight

Within the first part, an Integer Linear Programming (ILP) model is formulated and solved to maximize the total priority weight of matched jobs. Each job in the dataset is assigned a predefined priority weight ($w_j \in [1, 10]$). The employer determines the importance or urgency of each job. The aim is to design an assignment model that matches job seekers to job openings, while satisfying constraints such as job type, salary, skills, and location, which maximizes the total priority weight of all matched positions.

Let us define the binary decision variable x_{ij}, which equals 1 if job seeker i is assigned to job j, and 0 otherwise. The objective function can be written as:

$$Maximize \quad \sum_{(i,j) \in FeasiblePairs} w_j \cdot x_{ij}$$

Feasible Pair Identification

Before formulating the optimization model, a preprocessing step is carried out to determine the set of feasible seeker–job pairs. A pair (i, j) is considered feasible if it satisfies the following five compatibility conditions, implemented through the `can_do()` function:

1. **Job Type Compatibility:** The seeker's preferred job type and the job's type is matched perfectly.

2. **Salary Requirement:** The job's maximum salary offer must meet or exceed the seeker's minimum acceptable salary.

3. **Skill Matching:** All the listed skills that are required for the job is possessed by the seeker.

4. **Experience Level:** The seeker's experience level must be greater than or equal to the job's required level, according to a predefined hierarchy (Entry ¡ Mid ¡ Senior ¡ Lead ¡ Manager).

5. **Location Constraint:** If the job is not remote, the commute distance between the seeker's location and the job's location (as defined in `location_distances.csv`) must be less than or equal to the seeker's maximum acceptable commute distance.

The model's operation on a valid search space is ensured by defining a rule that only pairs passing these criteria are added to the candidate set for optimization.

Model Constraints

The ILP model incorporates the following constraints:

- **Assignment Constraint for Seekers:** Each seeker can be matched to at most one job:

$$\sum_{j} x_{ij} \leq 1 \quad \forall i$$

- **Capacity Constraint for Jobs:** Each job can accommodate no more than its available number of positions (P_j):

$$\sum_{i} x_{ij} \leq P_j \quad \forall j$$

With the constraints above, one-to-one feature of the assignments is ensured (with the seeker's perspective), and the hiring limits of each job posting are applied.

Objective Function

The objective function maximizes the total importance of matched jobs:

$$Maximize \quad \sum_{(i,j)} w_j \cdot x_{ij}$$

which guarantees that the model matches jobs with higher business value or urgency.

Implementation Details

Using the Gurobi optimization solver, the model is implemented where the feasible pairs are passed into `addVars()`, which is then constructed as the decision variables. Using the generator expressions (`x.sum(i, '*')` and `x.sum('*', j)`), constraints are added. Thus, when the problem is solved, the optimal objective value M_w represents the highest achievable sum of priority weights given all real-world restrictions.

3 Part 2 - Minimize Maximum Dissimilarity

The aim of the second part is to minimize the *maximum dissimilarity* between matched job seekers and job pairs while keeping the total priority weight of assignments above a given fraction ω of the maximum value M_w obtained in Part 1.

The underlying idea is to promote better compatibility between seekers and jobs based on their subjective preferences and work-related characteristics, captured by their answers to a 20-question survey. Each job seeker and job provides a response vector $q_{i,k}$ and $q_{j,k}$ respectively, where $k = 1, 2, \dots, 20$ and each value is in the range $[0, 5]$.

Dissimilarity Metric

In case of a scenario where the seeker i and a job j are matched incorrectly, a dissimilarity score d_{ij} is defined as the mean absolute difference across all 20 survey questions:

$$d_{ij} = \frac{1}{20} \sum_{k=1}^{20} |q_{i,k} - q_{j,k}| \quad \in [0, 5]$$

A lower value of d_{ij} indicates a closer alignment in expectations and work culture between the seeker and the job.

Model Objective and Decision Variables

As in Part 1, let x_{ij} be a binary variable that equals 1 if seeker i is assigned to job j, and 0 otherwise. Additionally, introduce a continuous decision variable $d_{max} \in [0, 5]$ to represent the maximum dissimilarity among all matched pairs. The objective is to minimize d_{max}, i.e.,

$$Minimize \quad d_{max}$$

This ensures that the most mismatched pair in terms of questionnaire responses is as compatible as possible.

Constraints

This model inherits all constraints from Part 1:

- **Assignment Constraint for Seekers:**

$$\sum_j x_{ij} \leq 1 \quad \forall i$$

- **Job Capacity Constraint:**

$$\sum_i x_{ij} \leq P_j \quad \forall j$$

- **Feasibility Constraints:** Each pair (i, j) must satisfy job type, salary, skill, experience, and location requirements, as determined in the preprocessing phase using the `can_do()` function.

The model also introduces two additional constraint sets:

1. **Maximum Dissimilarity Constraints:** For every matched pair, enforce that the dissimilarity does not exceed d_{max}:

$$d_{max} \geq d_{ij} \cdot x_{ij} \quad \forall(i, j)$$

2. **Minimum Weight Guarantee:** Ensure that the total priority weight in this model is at least $\omega\%$ of M_w:

$$\sum_{(i,j)} w_j \cdot x_{ij} \geq \omega \cdot M_w \quad where \omega \in \{0.70, 0.75, \ldots, 1.00\}$$

Implementation and Evaluation

For each ω value in $\{70, 75, 80, 85, 90, 95, 100\}$ (percent), a separate ILP model is built and solved using Gurobi. The dissimilarity scores d_{ij} are precomputed using NumPy and stored in a dictionary. The binary decision variables x_{ij} and the continuous variable d_{max} are defined for all feasible (i, j) pairs.

In each run, the model attempts to minimize the worst-case dissimilarity while ensuring that the solution quality (in terms of total priority weight) does not degrade below the desired threshold. The value of d_{max} is recorded for each ω value to analyze the trade-off between compatibility and assignment quality.

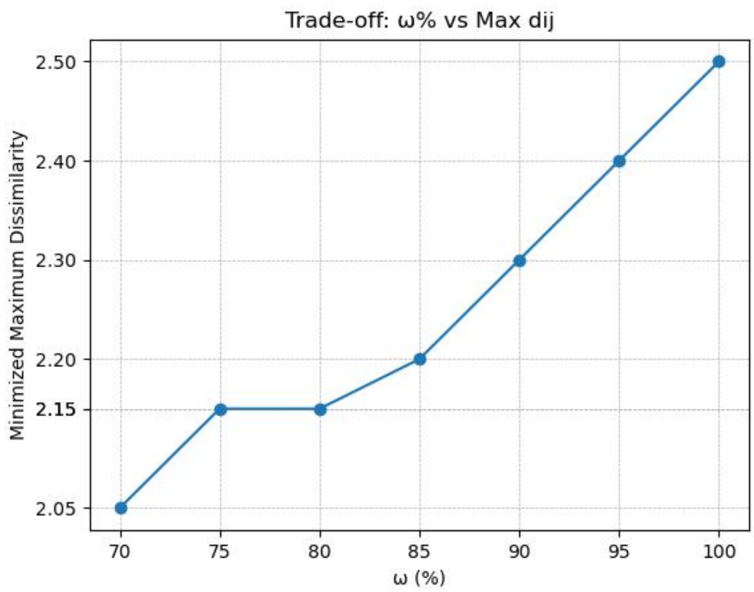

Figure 1: The Plot of Trade-off

Results and Observations

The results are plotted on a two-dimensional graph where the x-axis represents the ω values and the y-axis shows the corresponding minimized maximum dissimilarity d_{max}. When lower ω values are accepted (i.e., allowing a lower total priority weight), the model yields more culturally compatible results (lower d_{max}). Conversely, as ω increases, indicating stricter adherence to priority weight coverage, the feasible solution space contracts, leading to increased maximum dissimilarity among matches.

Upon examining the trade-off curve, we observe that dissimilarity scores remain relatively low and stable in the range $\omega \in [70\%, 85\%]$, with d_{max} values not exceeding 2.0. However, beyond $\omega = 90\%$, the d_{max} begins to rise noticeably, indicating that enforcing near-complete weight retention imposes significant compromises in compatibility.

Therefore, we select $\omega = 90\%$ as the optimal balance point. This value maintains a high level of organizational efficiency—retaining the majority of the original total priority weight—while keeping the maximum dissimilarity within acceptable limits ($d_{max} = 2.15$). It demonstrates a well-rounded trade-off between business-driven urgency and preference alignment, making it a rational and justifiable operational threshold for real-world application.

This result validates the flexibility and practicality of the proposed model in navigating the dual objectives of organizational priority and cultural fit within job matching scenarios.