IE400 - Spring 2025

Project Report

Kutay Buyruk - 22102559
Sıla Özel - 22102825
Halil Tataroğlu - 22102198

Team 20

1. Introduction

JobSync seeks to automate the pairing of job seekers with open positions by respecting organizational priorities and candidate–job compatibility. This program is developed by Kutay Buyruk, Sıla Özel, and Halil Tataroğlu for the IE400 term project. We develop two integer‑linear programming models implemented in Python with Gurobi 11.0.

- **Part 1 – Priority Model**: It maximizes the total priority weight of filled positions
- **Part 2 - Fairness Model**: It minimizes the worst culture‑fit dissimilarity while still keeping at least ω% of the total priority value from Part 1.

2. Data Overview

File	Key Fields
seekers.csv	Skills, Desired_Job_Type, Min_Desired_Salary, Experience_Level, Location, Max_Commute_Distance, Questionnaire
jobs.csv	Num_Positions *Pj*, Priority_Weight *wj*, Job_Type, Salary_Range, Required_Skills, Required_Experience_Level, Location/Is_Remote, Questionnaire
location_distances.csv	Symmetric kilometre matrix between cities A–F

First data is parsed and seekers are indexed by $i \in I$ and jobs by $j \in J$.

3. Part 1 - Priority‑Weighted Matching Model

3.1. Decision Variables

x_{ij} will be a binary decision variable, 1 if seeker i is assigned to job j; 0 otherwise

3.2. Formulation

$$max \sum_{i \in I} \sum_{j \in J} w_j x_{ij}$$

$s.t.$

$\sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I$, this will ensure one job per seeker

$\sum_{j \in J} x_{ij} \le P_j \quad \forall j \in J$, this will ensure job capacity does not exceed

$x_{ij} \le C_{ij} \qquad \forall j \in J, \forall i \in I$, this will ensure compatibility which will be discussed in Section 3.3

$x_{ij} \in \{0,1\} \quad \forall i, j$

C_{ij} will be 1 if only seeker i satisfies all fundamental requirements of job j.

3.3. Compatibility

Compatibility flags are calculated before optimization to reduce the model size. And this section will discuss our design choices.

1. **Job Type** – We are looking for the **exact match** here
2. **Salary** – Our design ensures that the **upper bound of the range of the salary of the job is larger or equal to the minimum desired salary**[1].
3. **Skills** – **Required skills are a subset of seeker skills** to ensure the seeker can provide at least these skills.
4. **Experience** – **Ordinal mapping** is used *Entry (1) < Mid (2) < Senior (3) < Lead (4) < Manager (5)*.
5. **Location / Commute** – Either the job is **remote or distance ≤ seeker's maximum commute distance**, where distance is looked up in the city-distance matrix.

3.4. Results

Solution yield max value as **Mw: 296.0**.

[1] As we were unsure about the design here in the Appendix section results of a different design choice will be shared as well

4. Part 2 - Minimize Maximum Dissimilarity

In Part 2, a dissimilarity score (d_{ij}) is used to define the questionnaire response distance between compatible seeker i and job j. Then, an ILP model is defined to find an assignment to minimize the worst (maximum) dissimilarity score among all matched pairs. All rules from Part 1 apply. Additionally, a parameter ω is introduced in Section 4.2.

4.1. Decision Variables

- x_{ij} is binary, defined as before.
- z is continuous, defining the worst (maximum) dissimilarity among matched pairs.

4.2. Parameters

The total priority weight achieved by the matches in this model is at least ω% of the maximum value M_ω found in Part 1.

The dissimilarity score is pre-computed as

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

4.3. Mathematical Formulation

The problem can be formulated into an ILP model as follows:

$$min\ z$$

$$s.t.\ \sum_j x_{ij} \leq\ 1 \qquad \forall i \quad (one\ job\ per\ seeker)$$

$$\sum_i x_{ij} \leq\ P_j \qquad \forall j \quad (capacity)$$

$$x_{ij} \leq\ C_{ij} \qquad \forall i, j \ \ (compatibility)$$

$$\sum_i \sum_j w_j x_{ij} \geq\ \omega M_\omega \qquad (priority\ threshold)$$

$$z\ \geq\ d_{ij} x_{ij} \qquad \forall (i,\ j)\ where\ C_{ij} =\ 1\ (max\ -\ dissim\ def.)$$

$$x_{ij} \in \{0,\ 1\},\ z\ \geq\ 0$$

- The $z \geq d_{ij} x_{ij}$ constraints are only generated for pairs with $C_{ij} = 1$, reducing the number of potential rows.
- The objective is linear because *z* is bounded above by 5 and multiplied by binary *x* only within the constraints, not in the objective.

4.4. Analysis of Changing ω as Parameter

Choosing the value of the parameter ω is a subject of experimentation. Increasing the ω value forces the model to progressively prioritize job priorities, whereas decreasing the ω value forces the model to prioritize the candidate's fit to the job in terms of questionnaire fit. Therefore, a tradeoff between job priorities and candidate fit must be made. To analyze how changing ω affects the objective value, the model is run multiple times with ω set to values 70, 75, 80, 85, 90, 95, 100. The results are presented below.

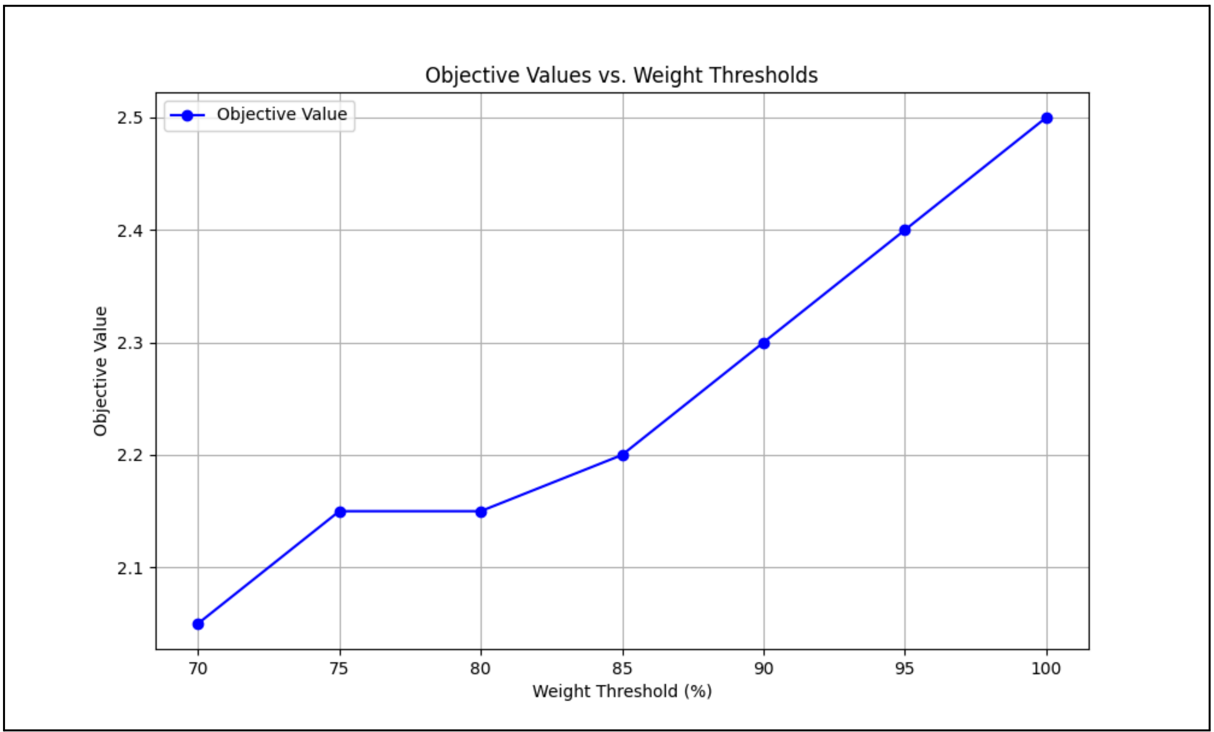

Figure 1: Trade-off between priority preservation and match quality[2]

From the figure, it can be seen that increasing the weight threshold monotonically increases the objective value (which is aimed to be minimized.) Upon closer inspection, the objective value increases very slowly, such that an increase of 30 absolute percentage points in weight threshold, from 70% to 100%, only increases the worst dissimilarity score by only 0.45 points. This means that, for the applicant with the worstly fit questionnaire, the distance between their questionnaire answers and job answers are only an extra half points away.

[2] Detailed table is available on Appendix section.

Although it seems desirable to increase ω since the relative increase at objective value is minimal, it may be a good strategy to choose a smaller value since the objective value is already quite high (2.05 at ω = 70.) Since the objective value stays relatively flat up to ω = 80%, it is a good point of trade-off to select **ω = 80%.**

5. Conclusion

Our two-stage optimization first maximizes the organizational priority, then improves intra-match quality with a retaining ratio ω. In the part 1 design we followed on salary might yielded a less strict result which is **Mw = 296.0**. Also in part 2, 80% was selected as the point as objective value stays relatively flat up to ω **= 80%** so we selected that point.

6. Appendix

6.1. Results of a Model With Different Design for Salary

If (job's salary range's lower bound >= desired minimum salary of the seeker) is set as a constraint in part 1 results will be the following.

Part 1:
Mw: 119.0

Part 2:

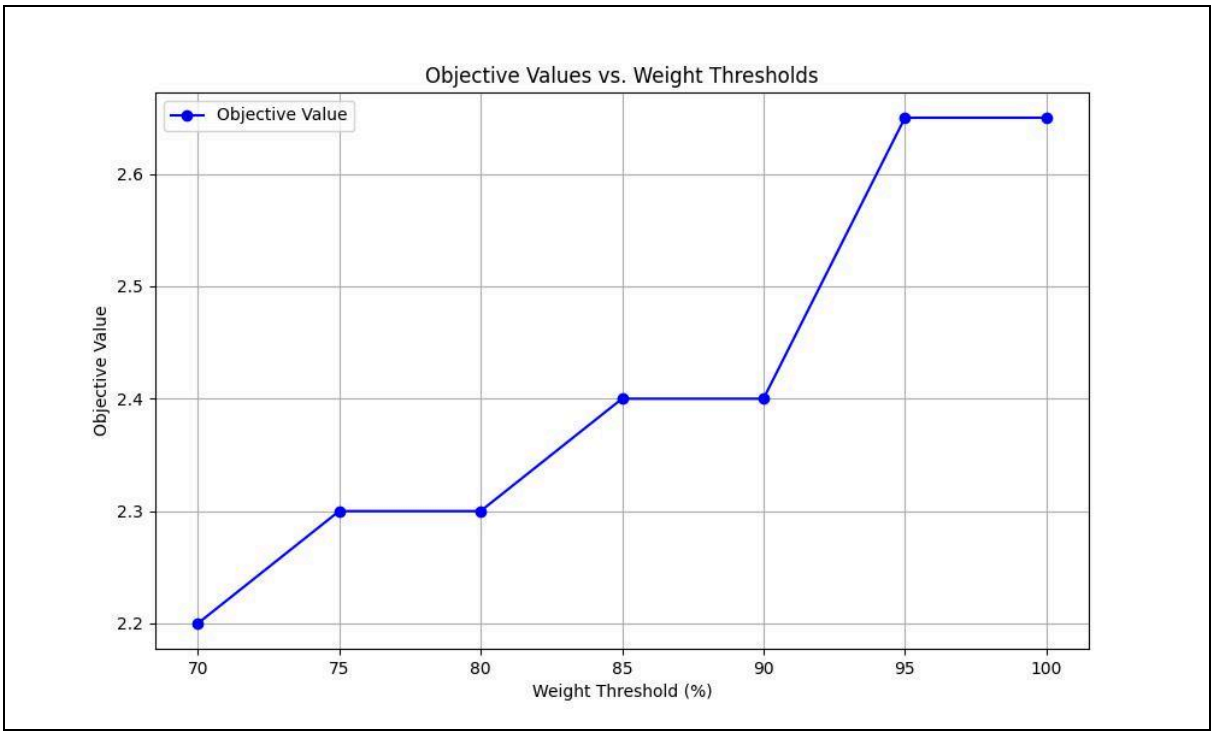

Figure 2: Trade-off between priority preservation and match quality for the alternative design

Here **ω = 90%** can be optimal choice as increase is not quite dominant until 90% but there is a rapid increase after 90%.

6.2. Exact Values of Part B

Weight Threshold	Objective Value
70	2.05
75	2.15
80	2.15

85	2.2
90	2.3
95	2.4
100	2.5

Table 1: Data Table of Figure 1