

Optimizing Candidate-Job Fit Using a Constraint-Aware Genetic Algorithm Framework

T. N. Malini

Department of Management Studies, Nitte Meenakshi Institute of Technology, Bangalore, India
malini.tn@nmit.ac.in

D. B. Srinivas

Department of Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India
srinivas.db@nmit.ac.in (corresponding author)

K. Lakshminarayana

Master of Business Administration, Visvesvaraya Technological University, Mudelahalli, Karnataka, India
lakhminarayana@vtu.ac.in

Received: 21 July 2025 | Revised: 30 August 2025, 12 September 2025, and 16 September 2025 | Accepted: 18 September 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: <https://doi.org/10.48084/etasr.13543>

ABSTRACT

Organizations aiming to hire the most suitable candidate for a specific job position face the complex challenge of distinguishing the best option among a large pool of candidates while also adhering to multiple constraints such as cost, diversity, role-specific quota, and candidate quality. Driven to solve this challenging problem, this work presents a data-driven optimization framework that leverages a Genetic Algorithm (GA) to automate and enhance the candidate selection process. Our methodology evaluates each candidate based on a fit score, derived from weighted attributes such as skills, experience, and alignment with job requirements, while key constraints, including budget limits, a minimum percentage of female candidates, required role coverage (e.g., engineers), and minimum fit thresholds, are integrated directly into the fitness evaluation. Solutions that violate these constraints are ignored, ensuring that the model retains only feasible and high-quality selections. Collectively, the GA operates on binary selection vectors representing subsets of candidates and iteratively evolves toward an optimal solution using fitness-based selection, crossover, and mutation operators. To evaluate GA performance, the model was tested using synthetic Human Resource (HR) data and successfully identified a subset of candidates that maximized overall fit while satisfying all constraints, validating the effectiveness of the proposed approach.

Keywords-genetic algorithm; candidate-job fit; human resource; budget; skills match score

I. INTRODUCTION

In today's increasingly dynamic and hybrid work environments, organizations face increased pressure to make rapid yet strategic talent acquisition decisions. The complexity of this challenge lies not only in identifying high-quality candidates but also in ensuring that the decision aligns with broader organizational objectives such as budget constraints, diversity and inclusion benchmarks, and role-specific workforce planning. Traditional recruitment pipelines, which often rely heavily on manual processes and subjective assessments, frequently fail to scale under these multidimensional demands.

On the other hand, Artificial Intelligence (AI) and optimization-driven methods have shown potential in assisting with identifying the optimal candidates based on the established constraints. Among these, Genetic Algorithms (GAs) have been particularly effective in solving complex search, selection, and scheduling problems under multiple constraints [1].

Inspired by the principles of natural selection, GAs evolve a population of candidate solutions iteratively toward an optimal set. In the context of talent acquisition, each candidate can be represented as a component of a solution vector, where their inclusion is encoded as binary values (selected or not). This representation enables optimization of candidate-job fit under

multiple simultaneous organizational requirements. However, integrating fairness, diversity, and transparency into such optimization processes remains an open challenge.

Recent literature demonstrates significant progress in algorithmic approaches to Person-Job Fit (PJF) modelling. For instance, authors in [2] proposed PJFCANN, which employs co-attention neural networks leveraging both candidate profiles and recruitment histories to enhance feature representation and improve fit accuracy. Similarly, the Multi-Field Features representation and Interaction (MUFFIN) model in [3] incorporates self-attention mechanisms, capturing interactions across textual, numerical, and categorical features to refine resume-job matching. Another notable contribution is the Topic-based Ability-aware Person-Job Fit Neural Network (TAPJFNN) introduced in [4], which uses hierarchical attention and word-level semantic representations to generate interpretable matches based on historical application data. Complementing these approaches, the Search History-enhanced PJF model (SHPJF) in [5] incorporates resumes, job descriptions, and user search behavior to personalize candidate-job alignment.

In parallel, evolutionary optimization techniques have continued to evolve. Enhanced GA variants, such as the Genetic Algorithm with Mutated-Crossover (GAMC) framework [6], have demonstrated improved performance, while other GA-based methods have been successfully applied to resource allocation and multi-criteria selection problems [7]. Building on these foundations, hybrid models have also emerged. For instance, the Genetic Algorithm and Decision-Making Trial and Evaluation Laboratory (GA-GDEMATEL) approach [8] integrates GAs with decision-making frameworks, whereas Reinforcement Learning-assisted Genetic Programming (RL-GP) [9] combines reinforcement learning with genetic programming to support dynamic optimization.

Beyond evolutionary methods, deep learning and graph-based techniques have further expanded the capabilities of recruitment analytics. For example, the Deep Cycle Learning (DCL) framework [10] employs Convolutional Neural Networks (CNNs) with noise injection for robust matching, while Bidirectional Encoder Representations from Transformers (BERT)-based approaches [11] leverage contextual embeddings for candidate shortlisting. Furthermore, graph-based approaches [12] explicitly model bidirectional preferences between employers and candidates, capturing directed interactions to achieve better mutual fit.

Finally, several studies emphasize the importance of aligning candidate attributes with organizational culture, as cultural mismatches are known to negatively impact employee retention [13, 14]. Related work on salary benchmarking, including latent factor models and big-data-driven frameworks such as the Good Fit Job Candidate Algorithm (GFC-A), highlights the influence of compensation and personality traits on candidate success [15, 16]. More recently, machine learning frameworks have reframed candidate selection as a cost optimization problem, balancing efficiency with fairness considerations [17].

After evaluating the limitations of previous models, this study proposes a Constraint-Aware Genetic Algorithm (CAGA) framework designed to optimize candidate-job fit while addressing real-world organizational constraints. Each candidate is evaluated based on a composite fit score, derived from multiple dimensions such as skill match (technical alignment with job requirements), experience level, cultural and organizational alignment, and performance indicators. The optimization objective is to maximize the total fit score of the selected candidates, subject to constraints such as budget, diversity, and job role. Additionally, visual analytics are integrated into the optimization pipeline, offering transparency that helps Human Resources (HR) decision-makers to select the most suitable candidate.

II. PROPOSED FRAMEWORK

A. Framework

The proposed CAGA framework, illustrated in Figure 1, comprises three principal components: Input Data, GA Process, and Metrics & Output.

The Input Data component defines the essential candidate characteristics, such as fit scores (how well each candidate matches the job), estimated hiring costs, gender (for diversity constraints), role (functional area), and optional attributes, such as psychometric scores or location preferences, which serve as raw inputs for the model. These data points are fed into the GA Process component, where an evolutionary loop generates and evaluates candidate combinations, selects high-performing solutions, applies crossover and constraint filtering, and iterates until an optimal subset is found. Finally, the Metrics & Output component provides interpretability and monitoring capabilities by reporting: the selected candidate subset, average and best fitness values per generation, solution diversity, and the final total fit score and hiring cost.

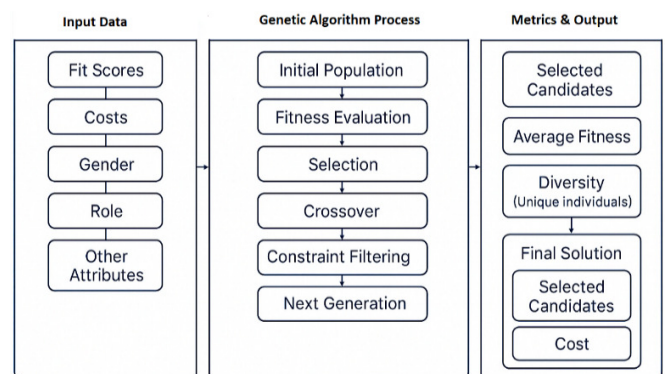


Fig. 1. CAGA framework.

B. Constraint-Aware Genetic Algorithm

In the CAGA framework, each candidate subset is represented as a chromosome, where each gene corresponds to one candidate's selection state. The algorithm begins by creating an initial population of N potential solutions (chromosomes), where each chromosome is a binary vector of length m , where m is the number of available candidates. A

diverse initial population ensures the algorithm explores different combinations of candidate selections from the start, avoiding premature convergence. The algorithm then iterates through G generations, progressively refining the population.

In every generation, each chromosome is evaluated to measure its quality and check whether it meets all required constraints. If a chromosome meets all constraints, it receives a fitness value equal to the sum of the selected candidates' fitness scores; otherwise, it receives zero (or a penalty), discouraging invalid selections. Once all chromosomes are evaluated, the algorithm selects the most promising individuals as parents, typically using methods such as tournament selection that prefer higher-fitness solutions. Crossover and mutation are then applied to generate new offspring. Crossover combines parent chromosomes to explore promising combinations, while mutation introduces small random changes to preserve diversity. Then, a new population is formed for the next generation, often retaining the best individuals through elitism to ensure valuable solutions are not lost. After completing all generations, the algorithm returns the best feasible chromosome found, representing the optimal set of selected candidates. Table I describes the key components used in the algorithm.

TABLE I. KEY COMPONENTS OF THE ALGORITHM

| Component | Description |
|---------------------|--|
| Chromosome | A binary vector (e.g., [1, 0, 1, 0, ...]), where 1 means that the candidate is selected. |
| Population | A set of chromosomes representing different combinations of selected candidates. |
| Fitness Function | Calculates the total fit score if constraints are met; otherwise, assigns zero fitness. |
| Selection | Selects the fittest individuals to pass genes to the next generation. |
| Crossover | Combines two parent chromosomes to create new offspring by swapping segments. |
| Mutation | Randomly flips bits in a chromosome to introduce diversity. |
| Constraint Handling | Ensures that only candidate combinations meeting all criteria are considered viable. |

1) Pseudocode Representation of the CAGA Algorithm

```

Step 1: Initialize population of  $N$  binary chromosomes of length  $m$ 
Step 2: for generation = 1 to  $G$  do
Step 3: for all individuals  $z \in$  population do
Step 4: Compute
    a) Total cost:  $C(z) = \sum z_i \cdot c_i$ 
    b) Total fit:  $F(z) = \sum z_i \cdot \text{FitScore}_i$ 
    c) Female ratio:  $R_f(z) = \frac{\sum z_i \cdot g_i}{\sum z_i}$ 
    d) Role count:  $R_r(z) = \sum z_i \cdot \mathbb{1}(r_i = \text{target})$ 
Step 5: if Constraints (4)–(9) are satisfied
    Then
Step 6: Fitness( $z$ )  $\leftarrow F(z)$ 
    else

```

```

Step 7: Fitness( $z$ )  $\leftarrow 0$ 
Step 8: end if
Step 9: end for
Step 10: Select parents based on fitness (e.g., tournament selection)
Step 11: Apply crossover and mutation to generate Offspring.
Step 12: Replace population using elitism strategy
Step 13: end for
Step 14: return best feasible individual  $z^*$ 

```

C. Model

The model evaluates each candidate i through a fit score, which is the weighted sum of key evaluation dimensions:

$$\text{FitScore}_i = w^T \cdot x_i = \sum_{j=1}^n w_j x_{ij} \quad (1)$$

where:

- $w = [w_1, w_2, \dots, w_n]^T$ is the weight vector.
- $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$ is the characteristic vector of candidate i .
- $w_j \geq 0$, and $\sum_{j=1}^n w_j = 1$.

If $n = 5$, then (1) the fit score expands as:

$$\text{FitScore}_i = w_1 S_i + w_2 I_i + w_3 C_i + w_4 E_i + w_5 A_i \quad (2)$$

where S_i is the skills match score (0-1), I_i is the interview score (0-1), C_i is the cultural fit score (0-1), E_i is the experience fit score (0-1), A_i is the salary alignment score (0-1), and w_j are normalized attribute weights such that $\sum_{j=1}^5 w_j = 1$.

1) Objective Function

Let $z_i \in \{0,1\}$ denote a binary decision variable indicating whether the candidate i is selected. The optimization goal is to maximize the total fit score of the selected candidates using:

$$\text{Max}_z \sum_{i=1}^m z_i \cdot \text{FitScore}_i \quad (3)$$

subject to the following organizational and fairness constraints.

- Budget constraint:

$$\sum_{i=1}^m z_i \cdot c_i \leq B \quad (4)$$

where c_i is the cost (e.g., salary) of the candidate i , and B is the total hiring budget.

- Diversity constraint:

$$\frac{\sum_{i=1}^m z_i \cdot g_i}{\sum_{i=1}^m z_i} \geq \delta \quad (5)$$

where g_i is a gender flag (e.g., 1 for female, 0 for male), and δ is the desired diversity ratio (e.g., 0.4).

- Role requirement constraint:

$$\sum_{i:r_i=k} z_i \geq R_k \quad (6)$$

where r_i is the role of candidate i (e.g., engineer = 1, designer = 2), and R_k is the required number of hires for the role k .

- Minimum fit score threshold:

$$z_i = 1 \Rightarrow \text{FitScore}_i \geq \theta \quad (7)$$

or equivalently,

$$z_i \cdot (\text{FitScore}_i - \theta) \geq 0 \quad (8)$$

- Weight normalization constraint:

$$\sum_{j=1}^n w_j = 1, \quad w_j \geq 0 \forall j \quad (9)$$

III. RESULTS AND DISCUSSION

The proposed CAGA framework was implemented and evaluated using Python, while experimental validation was conducted on the resume-job description matching dataset provided by NETSOL Technologies Inc. [18]. This publicly available dataset consists of 1,031 resume-job description pairs. Specifically, there are i) 648 matched pairs, where the resume appropriately corresponds to the job description, ii) 201 mismatched pairs, where the resume does not align with the job requirements, and iii) 142 invalid entries, containing incomplete or missing resume or job description information. The remaining 40 samples were excluded during preprocessing due to missing supplemental input information.

A. Model Evaluation

Figure 2 presents the CAGA evolution results of the best fitness and average fitness values over successive generations implemented on the NETSOL Technologies Inc. dataset. In the initial generations 0 to 12, the average fitness begins at the relatively low value of 3.0 and steadily increases, and gradually approaches the best fitness curve. During this phase, the algorithm evaluates a wide range of candidate combinations and progressively eliminates low-quality solutions.

In contrast, the best fitness at generation 0 is at a comparatively high value of 3.95, suggesting that a near-optimal solution is already present in the initial population. From generation 12 onward, both the best and average fitness curves stabilize at values close to 4.0. This plateau signifies algorithmic convergence, indicating that further generations yield minimal improvement and that the population has largely converged toward an optimal or near-optimal solution set.

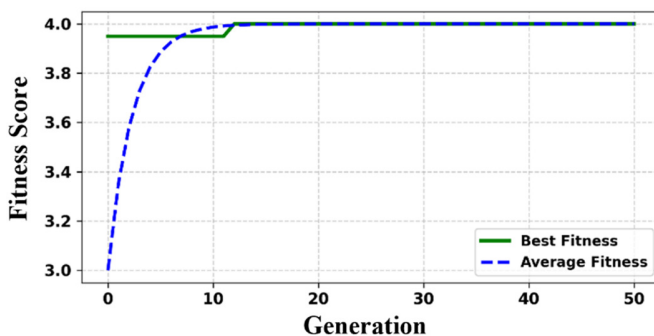


Fig. 2. GA fitness evolution over generations.

B. Comparison with the Greedy Algorithm

An additional step in our analysis included comparing the proposed model with the Greedy algorithm, which was selected as a baseline due to its simplicity and widespread use in resource allocation and candidate selection problems. Both algorithms were evaluated using the same dataset and identical constraints to ensure a fair comparison.

Figure 3 compares their performance across multiple runs using two evaluation metrics: SumFit and feasibility percentage. The SumFit metric represents the aggregate abnormalized fitness value, computed as the sum of individual candidate-job fit scores for all selected pairs in a solution.

Across all runs, CAGA consistently achieves higher SumFit values than the Greedy approach. Specifically, the CAGA SumFit curve exhibits a steady upward trend, increasing from approximately 88 in the initial run to nearly 98 in the final run. In contrast, the Greedy algorithm produces noticeably lower SumFit values, ranging from 80 to 88, with slower and less consistent improvement. This outcome reflects the Greedy method's reliance on locally optimal decisions, which limits its ability to explore and exploit better global solutions.

CAGA also demonstrates significantly higher and more stable feasibility percentages, remaining above 95% across most runs, while the Greedy algorithm exhibits lower and more variable feasibility percentages, ranging between 55% and 70%. Such variability suggests that Greedy selections frequently violate constraints or result in suboptimal feasible solutions. Overall, the proposed CAGA algorithm offers a promising framework for candidate selection problems.

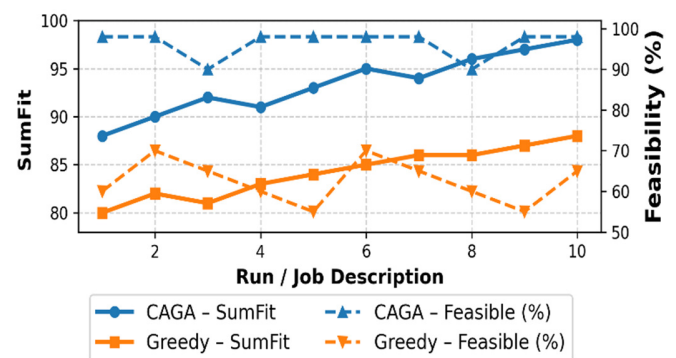


Fig. 3. Candidate selection performance comparison.

IV. CONCLUSION

This paper presents a scalable Constraint-Aware Genetic Algorithm (CAGA)-based optimization framework designed to enhance decision-making in talent acquisition, particularly in hybrid work environments. By encoding candidate selections as binary vectors and optimizing a composite fit score, the proposed model effectively identifies high-quality candidates while simultaneously satisfying real-world constraints, including budget limitations, gender diversity targets, minimum role representation, and per-candidate fit thresholds. The primary strength of the framework lies in its integrated fitness function, which not only evaluates the overall quality of

a candidate subset but also embeds organizational priorities and fairness considerations. This ensures that only feasible and equitable hiring decisions are generated.

The framework is validated using synthetic candidate data, demonstrating its ability to converge toward cost-effective, diverse, and high-performing candidate selections. Furthermore, the proposed approach is highly adaptable, making it suitable for integration into Human Resource (HR) decision-support systems and enterprise recruitment platforms.

Future work may extend the framework by incorporating machine learning techniques to predict candidate fit scores using historical hiring data or Natural Language Processing (NLP)-based resume analysis. Integration with interactive enterprise Human Resource Management System (HRMS) platforms would further enhance the practical applicability of the framework in real-time recruitment and workforce planning environments.

REFERENCES

- [1] D. Li, L. Raymond, and P. Bergman, "Hiring as Exploration." arXiv, Nov. 2024, <https://doi.org/10.48550/arXiv.2411.03616>.
- [2] Z. Wang, W. Wei, C. Xu, J. Xu, and X.-L. Mao, "Person-job fit estimation from candidate profile and related recruitment history with co-attention neural networks." arXiv, Jun. 2022, <https://doi.org/10.48550/arXiv.2206.09116>.
- [3] M. He, D. Shen, T. Wang, H. Zhao, Z. Zhang, and R. He, "Self-Attentional Multi-Field Features Representation and Interaction Learning for Person–Job Fit," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 1, pp. 255–268, Feb. 2023, <https://doi.org/10.1109/TCSS.2021.3134458>.
- [4] C. Qin *et al.*, "An Enhanced Neural Network Approach to Person-Job Fit in Talent Recruitment," *ACM Transactions on Information Systems*, vol. 38, no. 2, pp. 1–33, Apr. 2020, <https://doi.org/10.1145/3376927>.
- [5] Y. Hou *et al.*, "Leveraging Search History for Improving Person-Job Fit," in *Database Systems for Advanced Applications*, vol. 13245, A. Bhattacharya, J. Lee Mong Li, D. Agrawal, P. K. Reddy, M. Mohania, A. Mondal, V. Goyal, and R. Uday Kiran, Eds. Cham: Springer International Publishing, 2022, pp. 38–54.
- [6] S. A. Naz, A. Dilshad, and A. Sikander, "Person-Job Allocation Optimization using Genetic Algorithm with Mutated-Crossover (GAMC)," *Journal of Independent Studies and Research Computing*, vol. 20, no. 2, Dec. 2022, <https://doi.org/10.31645/JISRC.22.20.2.9>.
- [7] A. Shakya and S. Paudel, "Job-Candidate Matching using ESCO Ontology," *Journal of the Institute of Engineering*, vol. 15, no. 1, pp. 1–13, Jan. 2019, <https://doi.org/10.3126/jie.v15i1.27699>.
- [8] P.-H. Nguyen, "GA-GDEMATEL: A Novel Approach to Optimize Recruitment and Personnel Selection Problems," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–17, Sept. 2022, <https://doi.org/10.1155/2022/3106672>.
- [9] Y. Guo, H. Wang, L. He, W. Pedrycz, P. N. Suganthan, and Y. Song, "A Reinforcement Learning-assisted Genetic Programming Algorithm for Team Formation Problem Considering Person-Job Matching." arXiv, Apr. 2023, <https://doi.org/10.48550/arXiv.2304.04022>.
- [10] Z. Zhang, Y. Luo, Y. Wen, and X. Zhang, "CycleResume: A Cycle Learning Framework with Hybrid Attention for Fine-Grained Talent-Job Fit," in *Artificial Intelligence*, vol. 13606, L. Fang, D. Povey, G. Zhai, T. Mei, and R. Wang, Eds. Cham: Springer Nature Switzerland, 2022, pp. 260–271.
- [11] E. Abdollahnejad, M. Kalman, and B. H. Far, "A Deep Learning BERT-Based Approach to Person-Job Fit in Talent Recruitment," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, Dec. 2021, pp. 98–104, <https://doi.org/10.1109/CSCI54926.2021.00091>.
- [12] L. A. A. Shaalan, and F. M. Y. Hamed, "The Impact of Person–Job Fit on Employee Engagement," *The Business and Management Review*, vol. 15, no. 02, Aug. 2024, <https://doi.org/10.24052/BMR/V15NU02/ART-17>.
- [13] A. Dimopoulos, D. Evaggelos, and K. Zafiroopoulos, "Person to Organization Fit & Person to the Job Fit Impact on Employment Interview Decisions. An Exploratory Field Study in Greece," *International Journal of Human Resource Studies*, vol. 11, no. 1, Jan. 2021, Art. no. 130, <https://doi.org/10.5296/ijhrs.v11i1.18259>.
- [14] Q. Meng, K. Xiao, D. Shen, H. Zhu, and H. Xiong, "Fine-Grained Job Salary Benchmarking with a Nonparametric Dirichlet Process–Based Latent Factor Model," *INFORMS Journal on Computing*, vol. 34, no. 5, pp. 2443–2463, Sept. 2022, <https://doi.org/10.1287/ijoc.2022.1182>.
- [15] M. F. Uddin, "A Proposed Good Fit Job Candidate Algorithm (GFC-A) Utilizing Big Data and Career Data," in *2022 8th International Conference on Information Technology Trends (ITT)*, Dubai, United Arab Emirates, May 2022, pp. 172–176, <https://doi.org/10.1109/ITT56123.2022.9863941>.
- [16] S. Wild and M. Last, "Cost-Oriented Candidate Screening Using Machine Learning Algorithms," in *Recent Challenges in Intelligent Information and Database Systems*, vol. 1716, E. Szczerbicki, K. Wojtkiewicz, S. V. Nguyen, M. Pietranik, and M. Krótkiewicz, Eds. Singapore: Springer Nature Singapore, 2022, pp. 737–750.
- [17] S. S. K. Aithal, S. Santhosh, M. A. Shenoy, and S. S. Kumar, "Machine Learning based Ideal Job Role Fit and Career Recommendation System," in *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, Feb. 2023, pp. 64–67, <https://doi.org/10.1109/ICCMC56507.2023.10084315>.
- [18] *Task: Resume Matching with Job Descriptions*. (2023), SIDDIGAN TM. [Online]. Available: <https://www.kaggle.com/code/siddigantm/task-resume-matching-with-job-descriptions/input>.