2023-2024 Academic Year

Worker Assignment & Training Problem Based on Competency Levels

Abderrahmane BENADDER // Buse ÖZCAN// Gizem BAKTIR// Rabia YAVUZ// Selin DÜNYA

Academic Supervisor: Rahime Şeyma BEKLİ & Betül KAYIŞOĞLU// Company Supervisor: Ramazan KULAÇOĞLU & Hasan ÜLGER

ABDULLAH GÜL UNIVERSITY

INDUSTRIAL ENGINEERING DEPARTMENT OF INDUSTRIAL ENGINEERING DEPARTMENT OF INDUSTRIAL ENGINEERING

1 ABOUT COMPANY

FNSS Defense Systems Inc., a Turkish defense company established in 1988, specializes in customized defense products like tanks and armored vehicles. Headquartered in Gölbaşı, Ankara, it serves the Turkish Armed Forces and international customers. FNSS focuses on R&D, producing low-volume, high-customization products with long lead times. It employs over 1000 people, with 65% white-collar workers, nearly 90% engineers, and 50% in R&D. The company operates on 8-hour shifts, with some departments working one shift and others two.

2 CURRENT SYSTEM & PROBLEM DEFINITION

The project implementation focuses on the assembly production area, which consists of nearly 50 stations and 70 employees. Each station is allocated based on project requirements and involves various tasks. Some stations handle short processing times, requiring skilled employees to move between stations during their shifts.therefore the cost increases. It was thought that this machine would be able to prevent waste generation by changing the type in less time.

Production planning is managed daily by foremen who manually assign tasks based on worker competencies. This method, however, is inefficient, causing delays, reduced work quality, and customer dissatisfaction. The lack of training for new hires and the presence of low-ability workers further complicate production, leading to mistakes and inefficiencies. The project's main goal is to address these issues by optimizing worker assignments and training through mathematical modeling.

OBJECTION OF THE PROJECT

The project's objective is to efficiently and strategically assign tasks to workers, considering their competency levels. It aims to ensure proper task allocation while prioritizing worker training to diversify abilities. By enhancing worker competency in various tasks and implementing effective training, FNSS seeks to develop a skilled workforce that can improve productivity and quality across operations.

3 METHODOLOGY

Indices

i : worker index (1, 2,…,35)

t : task index (0, 1,…,90)

N1(t) : task index without dummy tasks (1, 2,…,89)

k : ability type (1, 2,…,12)

Alias

(t,q)

Parameters

a_t : Worker requirement of task t

b_{ik} : Ability score of worker i in ability type k

s_t : Duration of task t

n_{tk} : Ability requirement of task t in ability type k

p_t : Penalty cost for task t

r_t : Training requirement

m_{tq} : Task consecutiveness

d_{tk} : Ability requirement of task t in ability type k as binary

Decision Variables

x_{it} : {*If worker i is assigned to task t ,1 ; otherwise, 0*}

y_{itk} : {*If worker i is trained for task t for ability type k ,1 ; otherwise, 0*}

w_{itq}: {*If worker i is assigned to task/trainig q right after task t, 1 ; otherwise, 0*}

v_t : *Total missing worker number*

$u_{i,t}$: *Subtour elimination*

Objective Function

$$\text{Max } z = \propto_1 \left(\sum_i\sum_k\sum_t b_{ik}x_{it}d_{tk}\right) + \propto_2 \left(\sum_i\sum_k\sum_t (n_{tk} - y_{itk}b_{ik}d_{tk})\right) - \propto_3 \sum_t (v_t p_t) \quad (20)$$

s.t.

$$\sum_i x_{it} \le a_t \quad \forall_t \quad (1)$$

$$\sum_k y_{itk} \le 12(1 - x_{it}) \quad \forall_{i,t} \quad (2)$$

$$b_{ik} \ge n_{tk}x_{it} \quad \forall_{i,t,k} \quad (3)$$

$$\sum_i(\sum_k y_{itk}) = r_t \quad \forall_t \quad (4)$$

$$b_{ik}y_{itk} \le n_{tk} - 1 \quad \forall_{i,t,k} \quad n_{tk}\neq 0 \quad (5)$$

$$b_{ik}y_{itk} \le n_{tk} \quad \forall_{i,t,k} \quad n_{tk} = 0 \quad (6)$$

$$a_t - \sum_i x_{it} = v_t \quad \forall_t \quad (7)$$

$$w_{itq} - m_{tq} \le 0 \quad \forall_{i,t,q} \quad (8)$$

$$\sum_q w_{itq} \le x_{it} + \sum_k y_{itk} \quad \forall_{t,i} \quad (9)$$

$$\sum_t w_{itq} \le x_{iq} + \sum_k y_{iqk} \quad \forall_{q,i} \quad (10)$$

$$\sum_t w_{itq} = x_{iq} + \sum_k y_{iqk} \quad \forall_{q,i} \quad (11)$$

$$\sum_t w_{iqt} = x_{iq} + \sum_k y_{iqk} \quad \forall_{q,i} \quad (12)$$

$$w_{itt} = 0 \quad \forall_{i,t} \quad (13)$$

$$\sum_t w_{it0} = 0 \quad \forall_i \quad (14)$$

$$\sum_q w_{i90q} = 0 \quad \forall_i \quad (15)$$

$$\sum_q w_{i0q} = 1 \quad \forall_i \quad (16)$$

$$\sum_q w_{iq90} = 1 \quad \forall_i \quad (17)$$

$$u_{it} - u_{iq} + card(t)\, w_{itq} \le card(t) - 1 \quad \forall_{i,t,q} \quad (18)$$

$$x_{it},\ y_{itk},\ w_{itq} \in \{0,1\} \quad (19)$$

$$v_t\ ,\ u_{i,t} > 0 \quad (20)$$

Explanation of the Constraints:

1- It ensures that the number of workers assigned to task t meets the required worker number for task t.

2- It ensures that since there are 12 distinct ability types when workers are assigned to training, the ability level that is insufficient for the required k type is taken into account. Additionally, a worker cannot be assigned to a task t as a trainee and primary worker at the same time.

3- It ensures that only workers who meet or above the ability level required by task t can be assigned to specific task t.

4- It ensures that the number of trainees assigned to task t exactly meets the required trainee number for task t.

5- It ensures that only workers who are beloved to the ability level required by task t can be assigned to training, excluding the ability requirement of task 0.

6- It ensures that only workers who are below the ability level required by task t can be assigned to training, specifically written for task 0.

7- It ensures the monitoring of the total missing worker number for task t.

8- It ensures the sequence of assigned tasks by following the consecutiveness table.

9, 10- It ensures that the number of right-after tasks assigned to worker i, for both trainees and the primary worker, does not surpass the total number of right-after tasks.

11, 12- It ensures that the number of right-after tasks assigned to worker i, for both trainees and the primary worker, is exactly equal to the total number of right-after tasks.

13- It ensures that after task t is completed, no assignment can be made to the same task again.

14- It ensures that no tasks can be assigned before the 0th task.

15- It ensures that no tasks can be assigned after the 90th task.

16- It ensures that all tasks starting from the 0th task will be assigned to other tasks.

17- It ensures that all tasks are eventually finished by assigning them to the 90th task.

18- It ensures the Miller-Tucker-Zemlin subtour elimination constraint that subtours don't emerge in the order in which workers are assigned tasks, guaranteeing a cogent assignment solution that eliminates unnecessary repetitions.

19, 20- It represents the variable domain.

$$\text{Max } z = (1/2085)\left(\sum_i\sum_k\sum_t b_{ik}x_{it}d_{tk}\right) + (1/432)\left(\sum_i\sum_k\sum_t (n_{tk} - y_{itk}b_{ik}d_{tk})\right) - (0.\,2/375110000)\sum_t (v_t p_t)$$

Max Absolute Scaling was applied to normalize the values of the three distinct objectives. This method allowed the model to scale the objectives to a common range, preventing any single objective from disproportionately influencing the optimization process. Additionally, a Bayesian algorithm was employed to effectively handle uncertainty and variability, leveraging prior knowledge and probabilistic models. By systematically exploring the solution space, the Bayesian approach ensured a balanced and optimal assignment of workers and training schedules, leading to accurate and feasible solutions based on company feedback and optimization results. Above is the company-specific objective function with weights.

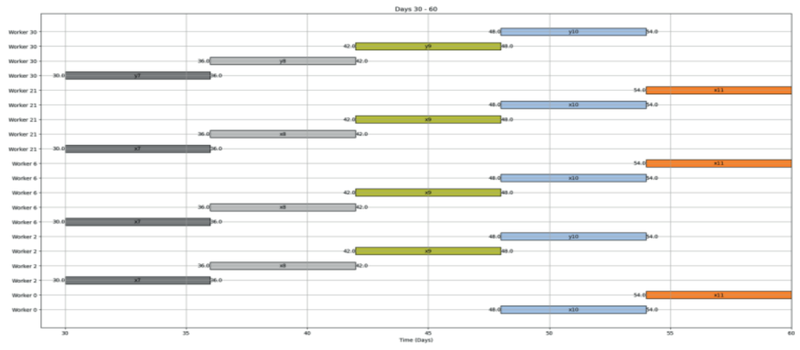
4 PYTHON IMPLEMENTATION

Python, using Gurobi software, solves a mathematical model efficiently due to Python's robust library support and Gurobi's capability to handle complex problems. Python's integration with Gurobi allows for defining variables, constraints, and optimizing models effectively. Results, displayed using Gurobi functions like setObjective and optimize, highlight assigned tasks ('x') and training ('y') variables. Visualized through Gantt charts, tasks are scheduled daily and monthly, aiding workforce management and training planning for optimizing efficiency and employee development.

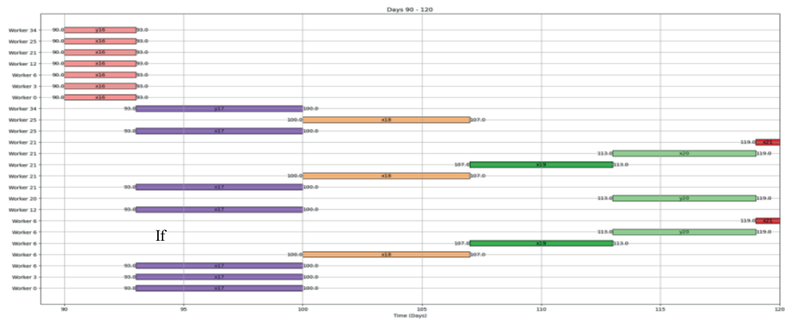
5 RESULTS

The company requested a 6-month workforce optimization model. Due to the complexity (35 workers, 90 tasks), separate daily, weekly, and monthly Gantt charts were used to show assignments clearly. 'X' bars denote actual work, 'Y' bars represent training sessions, enhancing model clarity and usability.

4.1.Daily Assignment



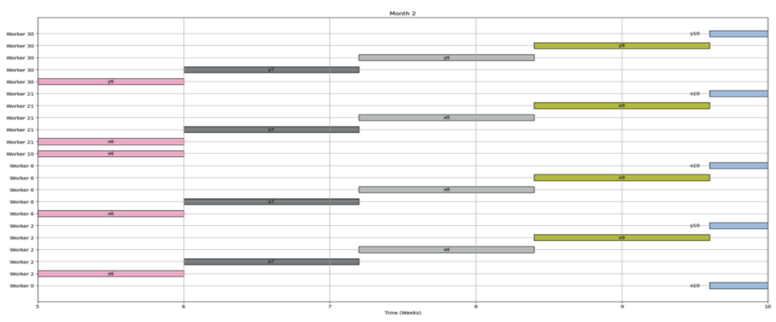
If a worker is assigned to a task for training, there must be a worker competent in that task who can teach him the task. Therefore, the worker must be accompanied by a worker who is competent in that task for training as long as the duration of the task. Looking at the gantt chart below, workers 25,21,12, and 6 start their work which is work14 on day 72 and are planned to finish this task on day 79. At the same time, it was seen that the 34th worker was assigned to the 14th task alongside the expert workers as training on the 72th day. More clearly, it is planned to train the 34th worker alongside the 25,21 and 21 workers for 8 days. In summary, all the tasks to be done are divided into days, making it easier to understand.



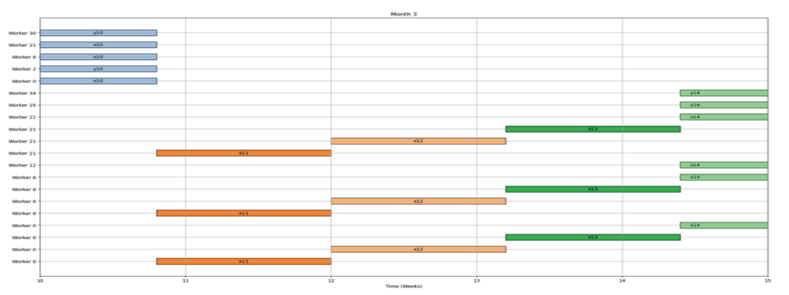
As seen in the graph above, it has been ensured that workers with appropriate competencies have been assigned to tasks in the best possible way, and it is concluded that the required number of workers have been assigned to tasks that require training.

4.2.Weekly & Monthly Assignment

If the company wants to see its weekly and monthly worker assignment plan, it can easily analyze it from the monthly Gantt chart. The x-axis indicates weeks, and the y-axis indicates workers. All Gantt charts are displayed divided by months. One month is calculated as four weeks, but the x-axis is divided by five to show that the work from the previous month continues in other months.



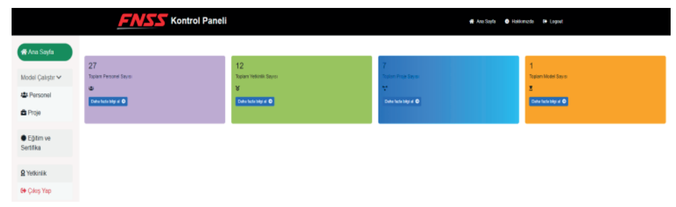
It is a priority to train some employees (e.g., new graduates, those who start working directly after a long-term internship) in certain tasks. For example, worker30 is trained in different tasks for 1 month, including tasks 6, 7, 8, 9 and 10 (consecutiveness tasks). Moreover, it is seen that some employees are experts in many tasks and do their own task repeatedly. For example, worker6 does his/her own task repeatedly using his/her competencies for 1 month.



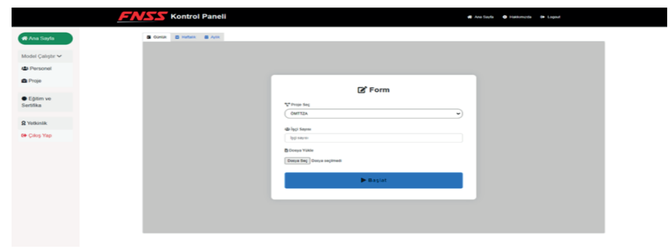
It is intended to ensure that only employees who match the competencies of the tasks are assigned by the model. For instance, since 11, 12, 13 tasks are consecutive, workers are not assigned to these 3 tasks for training and only competent workers are assigned in the model.

6 INTERFACE

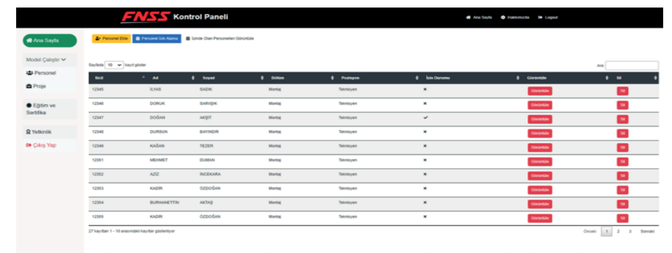
The project uses Django for rapid web development with PostgreSQL for robust database management. Django structures the app, PostgreSQL manages data, HTML provides UI basics, JavaScript adds interactivity, Django Templating System integrates dynamic content, and Cron automates tasks, enhancing development efficiency and user experience.



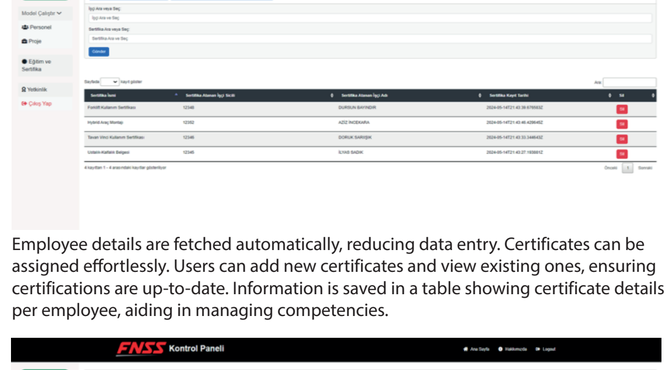
The home screen provides essential business performance metrics, including total personnel, competencies, projects, and working models.



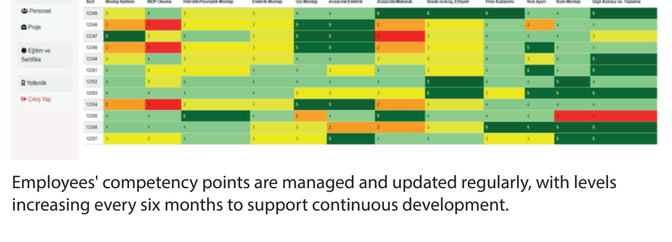
Users can activate, review, and adjust projects as needed, ensuring effective workflow management and informed decision-making to optimize business processes.



Users can add new personnel and assign permissions for leave, which are displayed in a personnel table. A system also ensures project continuity by automatically assigning replacements when employees take leave, supporting business efficiency and project timelines.



Employee details are fetched automatically, reducing data entry. Certificates can be assigned effortlessly. Users can add new certificates and view existing ones, ensuring certifications are up-to-date. Information is saved in a table showing certificate details per employee, aiding in managing competencies.



Employees' competency points are managed and updated regularly, with levels increasing every six months to support continuous development.

7 CONCLUSION & DISCUSSION

The project developed a competency-based worker and trainee assignment system, enhancing workforce efficiency and task performance, as confirmed by increased productivity feedback. Challenges like worker resistance were tackled by standardizing evaluations and providing support. Recommendations include FNSS adding provisions for worker shortages in training requirements and raising competency levels of all involved employees by one level. Salary increases for key employees and trainers were suggested to motivate and sustain training processes. Future improvements could leverage real-time data analytics, predictive modeling, and machine learning for better accuracy and responsiveness, with continuous evaluation to meet FNSS's needs.

8 REFERENCES

Attia, E., Duquenne, P., & Lann, J. L. (2014). Considering skills evolutions in multi-skilled workforce allocation with flexible working hours. International Journal of Production Research, 52(15), 4548-4573.

Beed, R. S., Sarkar, S., & Roy, A. (2021). Hierarchical Bayesian approach for improving weights for solving multi-objective route optimization problem. International Journal of Information Technology, 13(4), 1331-1341.

Lian, J., Liu, C., Li, W., & Yin, Y. (2010). Seru Seisan-an innovation of the production management Mode in Japan. Asian Journal of Technology Innovation, 18(2), 89-113.

INDUSTRIAL ENGINEERING

FNSS