IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 25 July 2023, accepted 16 August 2023, date of publication 21 August 2023, date of current version 25 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3306838

RESEARCH ARTICLE

A Novel Multi-Objective Evolutionary Algorithm to Address Turnover in the Software Project Scheduling Problem Based on Best Fit Skills Criterion

NATASHA NIGAR[1], MUHAMMAD KASHIF SHAHZAD[2], SHAHID ISLAM[1], OLUKAYODE OKI[3], (Member, IEEE), AND JOSE MANAPPATTUKUNNEL LUKOSE[3]

[1]Department of Computer Science, RCET, University of Engineering & Technology Lahore, Lahore 54890, Pakistan
[2]Ministry of Energy, Power Division, Government of Pakistan, Power Information Technology Company (PITC), Lahore 54000, Pakistan
[3]Department of Information Technology, Walter Sisulu University, Mthatha 5117, South Africa

Corresponding author: Natasha Nigar (natasha@uet.edu.pk)

ABSTRACT As a typical scheduling problem, the software project scheduling (SPS) under uncertain and dynamic events has gathered attention of academic researchers and practitioners because it causes tangible and intangible loss and will also influence the project schedule and cost if not addressed in time. In this paper, a SPS model based on skills acquired by employees and required by tasks criterion is proposed to tackle the issue of 'employee turnover' along with novel multi-objective evolutionary optimization algorithm to handle such dynamic events by employing the domain knowledge for population initialization. This is critical for the success of a software project as when an employee leaves, the project is rescheduled with remaining resources to assess if the project can be completed on time and under budget; otherwise, an alternative resource is hired. The proposed model and algorithm are evaluated on 18 dynamic benchmark and 6 real-world problem instances. The experimental results indicate that if existing employees are reassigned to the tasks, the project cost and duration increases on average 22% and 4% respectively. However, hiring new employees with recommended skills and proficiency based on proposed approach provides 9% decrease in project cost with no increase in duration and 15% decrease in cost against 11% referenced increase in project duration, in comparison to classical rescheduling results. The results demonstrate that proposed model and algorithm handles dynamic event effectively and help in identifying the right set of skills for hiring of an alternate resource while optimizing project duration and cost.

INDEX TERMS Software project scheduling, employee turnover, meta-heuristics, multi-objective optimization.

I. INTRODUCTION

The management of cloud-based software solutions (large-scale software projects) in a competitive market is time-consuming and error-prone [1]. As a result, ensuring effective resource allocation becomes essential for project managers to adhere to budget and schedule constraints [2], [3], [4]. This pertains to the software project scheduling (SPS) problem, wherein decisions are made concerning task assignments during the software development life cycle [5]. To tackle the SPS problem, most of the researches have focused on the development of exact and sub-optimal methods assuming a deterministic environment and availability of complete information [6], [7]. Nonetheless, these approaches may fail because of unique characteristics of software projects which are technology dependent. An essential characteristic of an effective scheduling system lies in its capacity to provide ample flexibility for efficiently and promptly responding to disruptions [8]. Therefore, the utilization of computer-aided tools becomes imperative for

The associate editor coordinating the review of this manuscript and approving it for publication was Shih-Wei Lin.

proficient project planning, particularly when managing volatile project parameters within schedule and budget constraints.

Employee turnover represents a volatile project parameter that has adverse impact on project schedule and budget. As evidenced by statistics, the employee turnover rate in some organizations has surged from the traditional 6∼10% to 25% [9]. The elevated turnover rates among IT professionals are attributed to a combination of market demand factors and internal organizational factors [10]. Perhaps, the employee turnover is not an evil doing, but core employee turnover during the software project development may significantly harm the enterprise, resulting in project failure. Based on these facts, staff turnover can significantly impact a software project's cost and duration, resulting in variations of up to 40% to 60% [10].

In cases where the existing staff possess the necessary skills to meet project requirements, a project manager may decide to reschedule them. This decision aims to minimize training expenses and hiring delays while ensuring the project is completed within budget. Moreover, it is not always easy for the manager to hire an employee with exactly the same skills as the departing employee. As resources with required skill set may not be readily available in the market; hence, manager may need some flexibility in employee's skills. Therefore, SPS under dynamic environment has emerged as an important problem for successful project completion.

In the last two decades, evolutionary algorithms (EAs) have become a class of popular methodologies for solving multi-objective optimization problems (MOPs). The SPS problem considering dynamic events and uncertainties, is one of those problems for which these EAs have shown promising results [11], [12], [13]. This is attributed to the population-based mechanism that enables them to discover multiple Pareto-optimal solutions in a single run. An evolutionary algorithm designed to address problems with multiple objectives is termed as multi-objective evolutionary algorithm (MOEA). The objective of a MOEA is to evolve a finite set of initial solutions (e.g., randomly-generated) over several generations, to a set of solutions that closely approximates the true Pareto-Front (PF) for a given problem in terms of convergence and diversity [14].

In literature, no study takes 'employee turnover' dynamic event (DE) into consideration, however, there exist some literature to predict and measure the likelihood of employee turnover rate [15]. To address this issue, we present a model to deal with employee turnover dynamic event. The model accommodates both employee rescheduling and recruitment scenarios. Additionally, we also propose a MOEA-based approach to address this real-world dynamic event in the context of the SPS problem. The suggested approach utilizes the ϵ-MOEA concept [16] and leverages domain knowledge to optimize and minimize project duration and cost objectives. In case of employees' rescheduling after the evolutionary search process and optimization, we present a set of solutions to the project manager for decision-making. The manager has the flexibility to determine whether they are satisfied with these solutions. Moreover, the suggested approach assists the manager in making new recruitment if existing employees, after employee turnover, are unable to meet project skill requirements or if the manager is not content with the rescheduling outcomes. This aspect of the approach integrates the manager's provided information into the search process for the SPS problem. After completing the search process, candidate solutions are presented to the decision maker, who selects the final schedule, including the minimum proficiency level required for new employees. Finding a quality software professional without any hiring delay can be a significant challenge for a software project manager. Such delays may ultimately impact the project's duration and cost. Hence, the purpose of this research is to propose an approach that addresses these identified issues. To sum up, this paper makes the following three main contributions:

- We present a model based on real-world situations to address the 'employee turnover' event, while considering both employee rescheduling and hiring scenarios.
- We propose a novel MOEA-based approach by employing the domain knowledge for population initialization to accommodate the dynamic event.
- We apply the proposed approach to 18 dynamic benchmark and 6 real-world data instances.

The subsequent sections of this paper are structured as follows: section II overviews the background and related work. Following this, the problem formulation in section III. Section IV provides the proposed methodology. The next section, section V, presents the mathematical model. Section VI discusses the experimental design and results. Finally, the last section concludes our work and outlines future directions.

II. LITERATURE REVIEW

A. BACKGROUND

1) STATE-OF-THE-ART APPROACH

Shen et al. [12] developed a mathematical model to address various objectives, accounting for dynamic events and uncertainties commonly encountered in software development projects. The model aims to present a dynamic version of the SPS problem, considering one uncertainty and three unforeseen events. The uncertainty in work efforts results from either inaccurate initial estimates or changes in job specifications made by the client, which may modify the projected work effort. Furthermore, the new requirements may also be raised by stakeholders during the course of the project. Additionally, the model accounts for dynamic events, such as employees leaving or returning after vacations, which can impact project execution. To handle these unforeseen events and uncertainty, a multi-objective approach built on ϵ-MOEA [16] is also proposed. This algorithm incorporates

TABLE 1. Employee turnover factors.

Key Criteria	Explanation
Managerial factors	Employees are more likely to stick with and continue working for a firm that should have a stable and welcoming workplace culture.
Working environment	Insufficient basic facilities can lead to a poor work-environment, resulting in employee dissatisfaction and turnover.
Pay	Lower pay is seen as one of the most important reasons influencing employee turnover.
Fringe benefit	Employee turnover is often caused by a lack of fringe benefits and employee incentives.
Career Promotion	Employees are motivated by career advancement, which involves increased pay and other benefits.
Job satisfactions	Dissatisfaction with one's employment is another important reason of employee turnover. When more appropriate staff are hired and essential steps to promote job satisfaction are implemented, the productivity of the organisation increases.
Clear job expectation	Turnover is not reduced unless job expectations for recently hired staff are not satisfied.
Alternative employment opportunity	One of the biggest challenges for organisations is when people quit the organisation if another better job is available.
Influence of co-workers	If an organisation has a high turnover rate. It will also cause an increase in the number of employees to decide to leave.

a scenario-based approach to deal with uncertainty and basic heuristic strategies for the dynamic events.

2) BASELINE APPROACH

A baseline algorithm is a complete rescheduling approach based on MOEA [17]. It has been attributed to create a completely new schedule from the beginning. It means that the initial population is produced at each scheduling point at random.

3) TURNOVER FACTORS

In the current era, there are growing concerns about the causes and implications of employee turnover in the software industry. This has led to the emergence of the concept of turnover culture, which helps us better understand turnover in the software business. Employee turnover is described as the rotation of employees within the labour market, transitioning between different businesses, occupations, and positions, as well as between employment and unemployment [18]. According to Price, [19] 'turnover' is the ratio of people who left an organization to the average number of employees in that organization over a specified time. The high turnover of IT experts can be attributed to a combination of internal organizational issues, such as poor job placements and limited career advancement opportunities, and external market demand factors, including attractive compensation and a significant need for software workers [10]. Therefore, the average length of employee tenure is about 13 months [9], indicating that employees are more committed to their careers and skills than to their employers [20]. Resignations pose significant challenges for project managers, especially when key employees who are well-versed in the project decide to leave. This departure can potentially cause delays or hinder the introduction of new technologies. Ultimately, such setbacks might lead to project suspensions, resulting in financial costs. According to [21], Table 1 presents the various factors that can contribute to employee turnover in any firm.

Turnover caused by employee resignations boosts expenses, diminishes output, disrupts teams, and results in knowledge loss. As a result, managing turnover efficiently is critical to project success.

B. RELATED WORKS

There are very few studies [11], [12], [22], [23] in the literature that address multi-objective SPS in a dynamic framework. The fact that "employee turnover" is not taken into account as a dynamic event in previous studies is particularly notable. Even though there is a lot of literature on this topic, the majority of studies have focused on the causes of employee turnover as well as predicting and evaluating the likelihood of employee turnover rate [15]. This is what motivates our research to develop a novel model and approach to handle this dynamic event.

In contrast to projects in other domains, software projects heavily rely on human resources as their primary assets, making them people-intensive operations [22]. As a result, an accurate and precise staff allocation to software activities is critical for software development organisations. In this regard, to address the dynamic nature of SPS problem, Shen et al. [12] presented a proactive-scheduling approach with inclusion of one uncertainty and three dynamic events (employee leave/return (from holiday), new task arrival). The extended version of their work [11], introduced a memetic algorithm with same aforementioned dynamic events. Under the same dynamic conditions,they [24] developed a mathematical model for large-scale projects. Cheng et al. [25] work involved addressing reworking job arrivals, dynamic skill proficiencies, and employee leave and return. Guo et al. [26] proposed a model to measure the dynamic skills of employees. However, no other event was considered. Xiao et al. [13] handled the dynamic resource rescheduling under the disruption-prone environment. Silva et al. [27] improved the particle swarm optimization (PSO) by adding dynamic strategies to solve the dynamic SPS considering new task

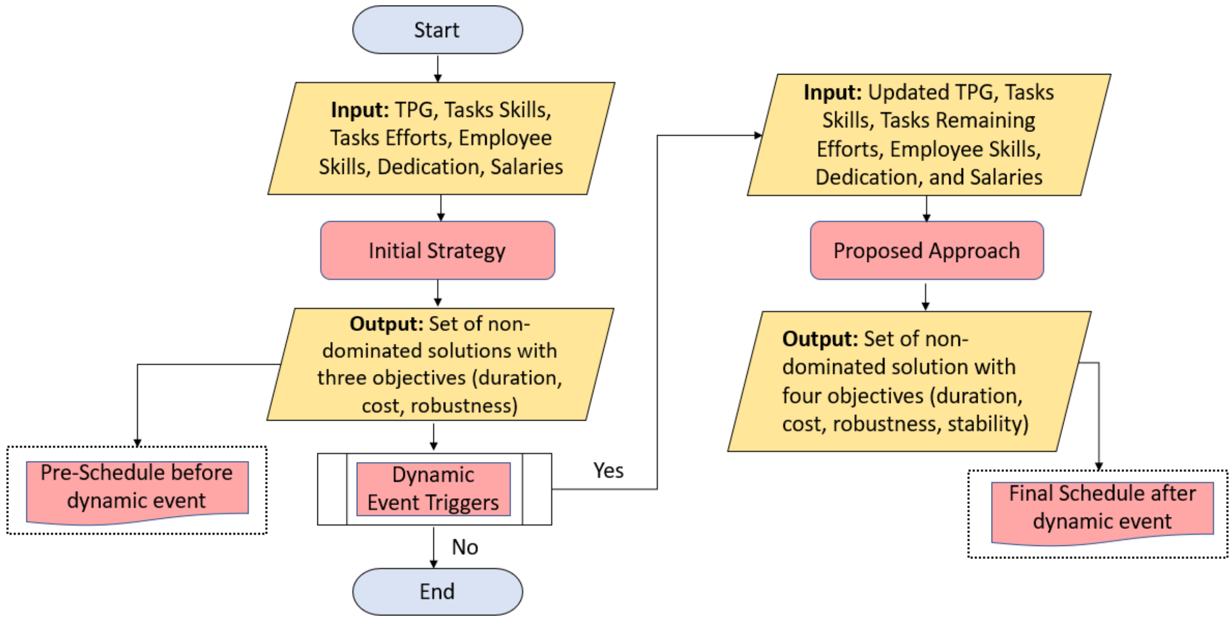


FIGURE 1. General framework for solving the DSPS problem using a rescheduling approach.

arrival event. Nigar et al. [23] presented a novel approach to deal with the addition of the new employee as an unplanned event. Besides it, they also proposed a model to show the influence of employee experience on the software project time and cost for the SPS problem [1].

Few studies [28], [29] also address the dynamic changes for resource constraint project scheduling (RCPS) problem, but that is out of scope for this research.

The preceding research emphasises the fact that dynamic techniques are limited to employee leaving and returning (from holiday). Since employee turnover has become a sort of culture in some organizations [9], where employees change employers by considering it as effective career strategy. As a result, it is critical to address this dynamic occurrence.

III. DYNAMIC SOFTWARE PROJECT SCHEDULING (DSPS)

A. GENERIC PROBLEM FORMULATION

Human resources are extremely important in the effective execution of software projects. As a result, guaranteeing appropriate and proper employees assignment to software tasks is an important necessity for software development organisations. Let E represent a group of workers engaged in a software project, where $E = \{1, 2, \ldots, m\}$, and m represents number of employees. An employee is denoted as e_i having skills set represented as $SK = \{s_1, s_2, \ldots, s_k\}$. The term $e^{skill} \subset SK$ represents skill of an employee, e^{norm_salary} and $e^{overwork\_salary}$ being the normal monthly salary and overwork salary respectively. The employee's maximum dedication to the project is represented as e^{max_ded}. The term e_{ij}^{prof} denotes employee's proficiency to perform a task.

Now let us talk about the modelling of software tasks. The set of tasks to be performed in the project is referred to as T. An individual is denoted as t_j, number of skills required by task as t^{skills}, and task effort as t^{effort} which is in person-month (PM). These tasks are carried out in accordance with a task precedence graph (TPG), which denotes the order in which they should be completed. The TPG is an acyclic directed graph $G(T, A)$ that illustrates dependency between tasks. In TPG, the set of tasks T are represented by the set of nodes, and arcs A show the precedence relation among the tasks. A general framework of rescheduling approach for DSPS problem is represented by Fig. 1.

B. SOLUTION REPRESENTATION

The output of SPS problem is represented by a dedication matrix $X = (x_{ij})$ of size m x n where $x_{ij} \geq 0$. The notation m and n represent number of employees and tasks respectively. Each element x_{ij} of the dedication matrix denotes the proportion of a full workday that an employee e_i spends on project-related task t_j. Hence, when $x_{ij} = 0.5$, it implies that employee spends half workday on task t_j. On the other hand, $x_{ij} = 0$ signifies that the employee will not perform that particular task. The dedication matrix is used to compute the duration, start, and end times for each activity. We can compute project cost using the dedication matrix and employees' salaries once the project duration has been determined.

IV. THE PROPOSED APPROACH

A. EMPLOYEE TURNOVER MODEL

Employee turnover is one of the primary reasons of software project failure [30]. It has the potential to harm the

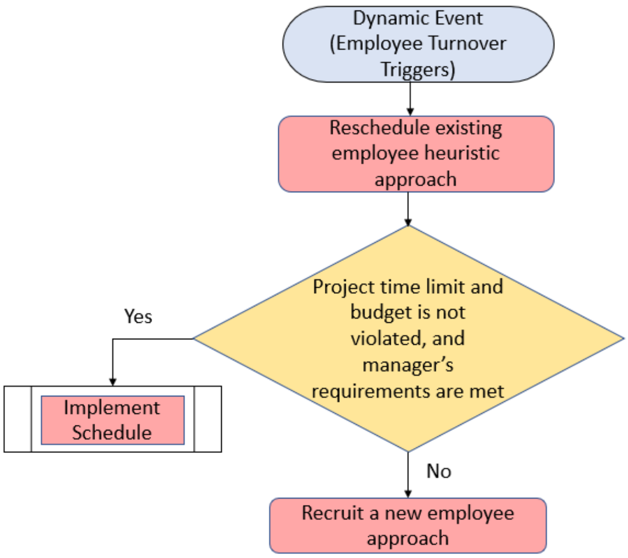


FIGURE 2. **Employee turnover model.**

organization's output, prosperity, and stability. As a result, it is increasingly important to do research on this issue in order to assist commercial organisations by recognising their difficulties, interpreting the data, and offering potential solutions. Employee resignation may result in both tangible and intangible losses for the software firm, as well as a substantial influence on project timeline and expense if not managed properly. Taking these factors into account, we provide a model (Fig. 2) based on a real-world situation in which a person decides to leave the software organisation during project execution.

This model is divided into two sections: i) employees rescheduling, ii) hiring of new employee. Initially, the model attempts to reassign current staff to affected activities if they can cover essential skills without exceeding the project's schedule and cost constraints. If the present employees fail to meet the project's requirements and are overworked, the model presents management with a range of potential candidates who can be hired under the budget. The workings of the model are discussed in the following sections:

1) RESCHEDULING APPROACH

In this approach, the existing employees are rescheduled. The designed heuristic strategy is based on real-world situation. Based on the heuristic, if an employee's resignation impacts certain tasks, reallocate a worker who possesses the needed skills for that task, provided that the employee excessive workload restriction is also met. Additionally, if two workers are assigned to the affected job, two distinct heuristic solutions are developed. This approach provides an informed decision-making support for the project manager. Algorithm 1 provides the pseudo code for reschedule heuristic.

Algorithm 1 Reschedule Heuristic

```
Input  : E_rem_set: remaining employees, T: set of
         available tasks, dedication matrix
Output: R_hs: Heuristic solutions with rescheduling of
         existing employees
for ∀T_j ∈ T do
    /* a subroutine findEffectedTasks() is called to
       find all the tasks affected by employee resign. */
    T_eff_set = Effected tasks
end
for ∀t_j ∈ T_eff_set do
    for ∀e_i ∈ E_rem_set do
        if e_ij^prof > 0 then
            X_ij : assign employee e_i to t_j with
              maximum dedication.
        end
        if Task_skill(X_ij) = 0 & Overwork(X_ij) = 0
         then
            /* a heuristic solution is generated. */
            X_ij : assign employee e_i to t_j with
              maximum dedication.
        end
        else
            /* Process is terminated and new
              employee hiring approach is adopted. */
            exit;
        end
    end
end
return R_hs
```

One probable rationale for the rescheduling strategy is that the leaving employee might be the least skilled for the tasks which wouldn't significantly delay the project execution.

Another factor is that employing a new employee may be more expensive. As per one estimate [31], it requires 18 months for a new employee to become optimally efficient. A huge influx of new workers on a software project will result in increased training and communication costs [10]. It is widely recognized that communication overhead increases proportionally to n^2, where n represents the size of the team [32], [33].

A set of non-dominated solutions is provided by EAs to solve any NP-hard problem [34]. As a result, following rescheduling, we present the team manager with a selection of viable options. It is up to the manager to decide whether or not he or she is satisfied with these solutions. It will allow managers to make more informed selections. A project manager may consider three sorts of solutions among all objectives: best duration, best cost, and best trade-off.

2) NEW EMPLOYEE HIRING APPROACH

If a leaving employee possesses certain abilities that no other employees possess, or if the manager is dissatisfied with rescheduling options, the organisation hires a new worker who possesses such skills. Recruiting competent software professionals while avoiding project delays may be a major problem for project managers.

To overcome these challenges, the software project manager is provided with information about the type of person they can recruit using this strategy. This technique will assist the project manager in quickly acquiring a new employee. We make an assumption that a new person is employed somewhere in the midst of the project; nevertheless, Brook's law [35], which suggests that adding workers to a late project will further delay it, does not apply. Another assumption is that a skill-related minimal competency value for an employee is '0.1'. The procedure is given below: **Step 1**: The information provided to the project managers i.e., threshold values: project duration and cost, are received as input. **Step 2**: The knowledge is extracted from the skills of the leaving employee and fed into the system. **Step 3**: The respective skill's proficiencies values are varied and optimized. We start this variation with the minimum proficiency value that a worker may have. Considering the number of important skills that substantially influence the project, reduce one skill's proficiency value at first and observe what happens. If there is success, then reduce the value of the second skill, and so on. We incrementally increase/decrease proficiency value based on all prior findings. As a consequence, we'll have a comprehensive list of all feasible options. Let a and b be two key skills that have a large influence on the project. Then, as shown in Fig. 3, we reduce the proficiency value of skill a first, and then lower the proficiency value of the second skill based on the outcome. **Step 4**: Upon completion of the search process and optimization, if the result falls within the threshold region, the solution is accepted. However, if the solution exceeds the threshold range, it is eliminated. At first, the threshold matches the preceding duration and cost settings. If a solution with the lowest feasible proficiency values cannot be found then increase the threshold and observe what happens. A software team manager may not want to surpass a specific amount of time and money. The threshold values will be determined by the software project manager.

B. MOEA-BASED METHOD FOR SPS PROBLEM

This section presents the evolutionary multi-objective (EMO) approach to handle the 'employee turnover' dynamic event. The suggested approach's mechanism is built on ϵ-MOEA [16]. The proposed MOEA method deals with: i) employees' rescheduling, ii) hiring new employee.

1) RESCHEDULING PROCEDURE

In this algorithm, the domain knowledge is used to generate the initial population. The generation will be more progressive if an initial population is created using population-based methods. This step is critical in EAs since the speed of convergence and the quality of the final solution are highly impacted by this. In general, EAs do not employ domain knowledge to build the initial population, and the most widely used technique is random solutions. Domain knowledge saves time by avoiding needless searches, leading to more promising results and improved EA convergence speed. Therefore, in our suggested method (Algorithm 2), the initial population P is generated, at scheduling point t' upon the occurrence dynamic event, using the following combination of solutions:

Algorithm 2 Rescheduling at Scheduling Point 't''

Input : Reschedule heuristic solutions R_{hs}, history solution H
Output: R_{es}: a set of non-dominated solutions with rescheduling of existing employees

```
P = ∪ (Recombination + Mutation(P), H, R_hs)
A ← non_dominated_sort(P)
j ← 1
repeat
    Function POP_SELECTION(P):
        return s
    Function ARCHIVE_SELECTION(A):
        return e
    Function VARIATION(s,e):
        return c1,c2
    P ← ∪ (P, c1, c2)
    A ← ∪ (A, c1, c2)
until j <= N
R_es ← non_dominated_sort(A)
return R_es
```

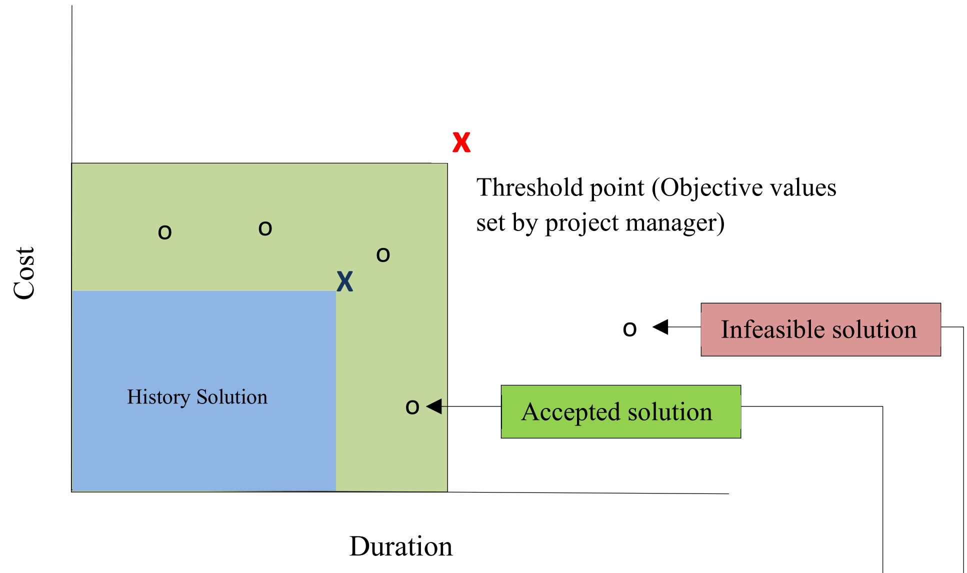
Case a: If an employee's resignation has an impact on certain tasks, the appropriate heuristic solutions are generated.

$$50\%Heuristic + 1\%Baseline + 49\%Random \quad (1)$$

In our case, 50% of the solutions consist of heuristics and their variations utilizing the mutation operator. Pre-schedule/baseline/history solutions (executed schedules) make up 1% of all solutions, while random solutions make up the remaining 49%. The inclusion of a baseline solution guides the search process towards a solution that that is more likely to be implemented. We use a combination of random initialization and a heuristic technique. Random initialization diversifies the solutions, whereas heuristics/baseline solutions converge the solutions in a rapid manner. **Case b:** No heuristic solution is produced. It suggests that the remaining employees have the necessary expertise for each assignment.

$$1\%History + 50\%HistoryVariants + 49\%Random \quad (2)$$

In Equation (2), pre-schedule/baseline/history solution (implemented schedule) make up 1%, its variants utilizing mutation operator make up 50%, and random solutions make up 49%.



Proficiency Variation Matrix

Skills	a	b	c	d
Proficiency	3.5	4.0	4.9	5.0
	0.1	4.0	4.9	5.0
	0.1	0.1	4.9	5.0

FIGURE 3. Working mechanism of proficiency variation.

Following the creation of the initial population, multi-objective evaluations are undertaken, and all the pareto non-dominated solutions are identified to create the archive population A. Two distinct solutions are chosen from P using the pop_selection technique, and two distinct solutions are chosen from A using the archive_selection procedure. To create offspring, variation operators are applied to the selected solutions. These off-springs are included in P and A until required number of evaluations are performed. When the stop requirements are fulfilled, all pareto non-dominated options from A are found and given to the project manager for selection of one optimal solution.

2) NEW EMPLOYEE HIRING PROCEDURE

The proposed MOEA for the hiring of new employee comprises the information supplied by project manager (PM) during the search phase. We also establish the necessary information for new employee (variation in employee skill competency levels, beginning with the minimum value). Following the search and optimisation processes, candidate solutions (which meet the threshold limit with one to two staff shuffles) are presented to the decision-makers. The decision maker then selects the best trade-off among these solutions by specifying a certain proficiency level for the new employee. The general flow chart for this method is illustrated in Fig. 4.

For 'new employee hiring' approach (Algorithm 3), the following set of solutions is used to create the first population:

$$1\% History + 50\% HistoryVariants + 49\% Random \quad (3)$$

In Equation (3), 1% is history solution with the information set up for new employee, 50% are its variants using mutation operator and 49% are random solutions.

Following that, multi-objective evaluations are performed to determine all of the pareto non-dominated solutions that comprise the archive population A. Using the pop_selection and archive_selection techniques, two individual solutions are chosen from P and A, respectively. Variation operators on the selected solutions yield two offspring. These off-springs are included in P and A until required number of evaluations are performed. When the stop criterion is fulfilled, all pareto non-dominated solutions from A are determined. If the project manager's requirements are satisfied, and the candidate solutions have the lowest proficiency values, then the decision maker chooses one of the candidate solutions. In the alternative, if the manager's requirements are met but the solutions do not have the lowest proficiency numbers,

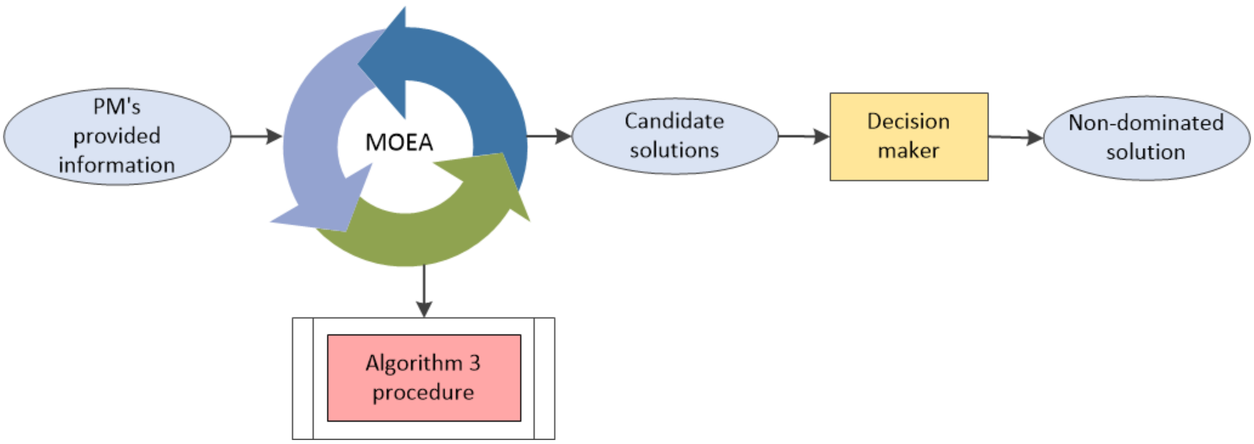


FIGURE 4. General mechanism of 'new employee hiring' MOEA.

we increase the threshold values based on the manager's preference. Otherwise, we just increase/decrease proficiency values step by step based on past results.

V. OPTIMIZED OBJECTIVES AND CONSTRAINTS

A. OBJECTIVE FUNCTIONS

This section defines the project objectives, namely the duration and cost. The objectives are optimised to assess how well our proposed strategy perform for a new unplanned event. Each objective's mathematical formulation and detail are according to [12].

$$\min F(t) = [f_1, f_2] \tag{4}$$

The project duration and cost objectives are denoted as f_1, f_2 respectively.

$$f_1(t) = duration = max(T_j^{end}) - min(T_j^{start}) \tag{5}$$

In Eq. (5), the duration objective denoted as $f_1(t)$ is the maximum elapsed time necessary to complete each available task. The terms T_j^{end} and T_j^{start} denotes the start and finish time of each task respectively. Furthermore, the duration of the entire project is the maximum completion time of the last task.

$$f_2(t) = cost = \sum_{e_i \in E\_ava\_set} e\_cost_i \tag{6}$$

In Eq. (6), $f_2(t)$ indicates the project cost, which is the total of all expenditures paid to available employees against their commitments to available tasks. The term e_cost denotes employee-paid expenditures, and E_ava_set represents the set of available employees. Let T_active_set symbolise the set of active tasks that are being developed at time t', where t' represents any month that the project is being developed. At time t', a task is said to be active if it has no previous incomplete task in the TPG. Hence, the costs paid to the employee e_i at t' month are calculated as follow:

$$if \sum_{j \in T\_active\_set} x_{ij} \leq 1.0$$
$$e\_cost_i = e_i^{norm\_salary} \cdot t' \cdot \sum_{j \in T\_active\_set} x_{ij} \tag{7}$$

$$if \sum_{j \in T\_active\_set} x_{ij} \geq 1.0$$
$$e\_cost_i = e_i^{norm\_salary} \cdot t' \cdot 1 + e_i^{overwork\_salary} \cdot t' \cdot \left(\sum_{j \in T\_active\_set} x_{ij} - 1 \right) \tag{8}$$

where $e_i^{norm\_salary}$ represents employee's normal income, $e_i^{overwork\_salary}$ is overwork salary, x_{ij} denotes dedication of an employee e_i to task t_j. The term x_{ij} greater than 1 indicates that the employee overworks for the project, and overtime pay is applied to the employee's total pay.

B. CONSTRAINTS

The SPS problem is subject to following constraints [12]. Below (i)-(iii) are hard constraints while (iv) is a soft constraint.

i) All tasks allocation constraint: There should be no unassigned tasks. At least one available employee should be assigned to each task. If T_j_Dedication represents task dedication then:

$$\forall e_i \in \text{e\_available\_set(t)}, \sum T_j\text{\_Dedication} \neq 0 \tag{9}$$

ii) Task skills constraint: All employee assigned to a job must possess the necessary skill set.

$$T_j^{skills} \subseteq \cup e_i, \{Skills \mid X_{ij} > 0\} \tag{10}$$

iii) No overwork constraint: An employee should not devote more time to the project than his/her maximum

Algorithm 3 'New Employee Hiring' at Scheduling Point 't''

Input : Project manager's provided Information, resigning employee skills, history solution H, history variants HV

Output: R_{ns}: a set of non-dominated solutions with recommended skills for new employee

Set up some information for new employee e_{n+1}
$P = \cup$ (Recombination + Mutation(P), H, HV)
$A \leftarrow non\_dominated\_sort(P)$
$j \leftarrow 1$
repeat
  Function POP_SELECTION(P):
    return s
  Function ARCHIVE_SELECTION(A):
    return e
  Function VARIATION(s,e):
    return c1,c2
  $P \leftarrow \cup (P, c1, c2)$
  $A \leftarrow \cup (A, c1, c2)$
until $j <= N$
$Q \leftarrow non\_dominated\_sort(A)$
if $manager's_requirements == true$ **then**
  if Q : $lowest_proficiency_values$ **then**
    $S \leftarrow decision_maker(Q)$
  end
  else
    t_values += x. Go to Step 1.
  end
end
else
  Increase/decrease proficiency values step-wise depending on previous results. Go to Step 2.
end
return R_{hn}

dedication.

$$\sum_{j \in T} X_{ij} \leq e_i^{max_ded} \tag{11}$$

iv) Task head count constraint: According to the best practices of software engineering development, the number of employees assigned to a task should not exceed a certain limit. The terms $T_j^{max_headcount}$ and $T_j^{no\_of\_emp}$ represent task's maximum head count and number of employees allocated on it. Task headcount is calculated according to [30].

$$\forall T_j, T_j^{no\_of\_emp} \leq T_j^{max_headcount} \tag{12}$$

C. EMPLOYEE'S PROFICIENCY FORMULATION

The employee proficiency e_{ij}^{prof} for performing a task can be described as the employee's capacity level to do that work with values between [0,1]. According to Chang [36], an employee with a zero proficiency score (denoted as *prof*) for a certain skill required by a task indicates that the employee is unable to accomplish that activity. In reality, if an individual is not skilled in one skill but has other skills for a task, the person can still complete that activity. Therefore, in this work, we define employee to task proficiency as:

$$e_{ij}^{prof} = \sum_{k \in req_j} \frac{prof_i^k}{C} \tag{13}$$

In Eq. (13), req_j implies the skills required by the task along with proficiency score for each skill; C denotes a constant and we set '$C = 5$' according to [36]. We then calculate average proficiency as follows:

$$e_{ij}^{prof} = \frac{e_{ij}^{prof}}{total\_skills\_required\_by\_task} \tag{14}$$

VI. RESULTS AND DISCUSSION

A. EXPERIMENTAL SETUP

1) DSPS INSTANCES

We used 18 dynamic benchmarks and 6 real-world data instances in this investigation.

a: BENCHMARK

The 18 dynamic benchmark instances are extracted from Alba and Chicano's dataset [5]. This dataset was chosen because it includes variations of three important criteria (number of employees, number of tasks, and number of employee skills) for the SPS problem in a real-world setting. This dataset also includes key factors such as task maximum headcount, task effort uncertainty, part-time jobs, and employee overwork. In the project, it is assumed that 20% of employees are part-time with maximum dedications in the range [0.5, 1], 20% are overtime employees with maximum dedications generated at random from (1, 1.5), and the remaining employees are full-time with maximum dedication set to 1.0. Each employee is assigned a competency score for a skill; these values are sampled uniformly from (0, 5] randomly. Each employee's typical monthly income is sampled from a normal distribution with a mean of 10,000 and a standard deviation of 1,000. In the initial state of project, TPG is generated using the method in [5].

b: REAL-WORLD

Our experiments also make use of six real-world instances. Three of these instances are from business software construction projects for a departmental store [37] with a maximum of 10 employees and 15 tasks. We also utilise three real-world instances gathered from a software company. These data instances have 10,8,5 employees as well as 42,43, and 91 tasks, respectively.

The representation of data instances is like, 'T15_E10 _SK6-7', where the notation 'T15' denotes number of tasks, 'E10' represents employees and 'SK6-7' means skills that an employee could possess. The representation of six real instances is like 'Real_n_ Tx_Ey' where n ∈ {1,2,...6} and

TABLE 2. Employees properties for Real_2 data instance.

ID	Is Contract	Basic Salary	Overwork Salary	perHour Salary	nHours	Exp	Ded
1	N	3000	100	40	100	0.38	1.5
2	N	2600	0	35	100	0.9	1
3	N	2600	0	30	100	0.59	1
4	N	2400	0	34	100	0.84	0.75
5	N	2400	0	34	100	0.48	0.5
6	Y	0	0	100	100	0.06	1
7	Y	0	0	96	100	0.46	1
8	Y	0	0	94	100	0.32	1
9	Y	0	0	90	100	0.91	0.75
10	N	2600	0	36	100	0.9	0.5

TABLE 3. Tasks properties of Real_2 data instance.

Task ID	Status	Effort	HeadCount	Skills
1	1	0.5	3	1,3
2	1	1	3	1,3,5
3	1	1.6	4	1,2,3,5
4	1	0.5	4	1,2
5	1	1.2	4	1,3
6	1	1	4	3,4
7	1	0.6	3	2,4,5
8	1	0.8	3	2,3
9	1	0.5	3	2,4
10	1	1	3	4,5
11	1	1	3	1,3,5
12	1	0.5	3	2,4
13	1	1.6	4	2,3,4
14	1	1	3	1,2,3,5
15	1	2	4	2,3,4,5

TABLE 4. Employees properties for Real_3 data instance.

ID	Is Contract	Basic Salary	Overwork Salary	perHour Salary	nHours	Exp	Ded
1	N	3400	105	35	100	0.82	1.25
2	N	2400	102	34	100	0.69	1.5
3	Y	0	228	114	100	0.59	1.75
4	N	3200	129	43	100	0.8	1.5
5	N	2500	0	27	100	0.48	0.5
6	Y	0	0	50	100	0.63	0.5
7	N	4100	0	43	100	0.94	0.5
8	Y	0	0	79	100	0.32	0.25
9	Y	0	0	109	100	0.91	1
10	N	3500	0	38	100	0.9	0.25

x, y represents number of tasks and number of employees respectively. Table 2,4 and Table 3,5 show employee and task attributes respectively, as examples of real-world data cases.

2) PARAMETER SETTINGS

In Table 6, we present a set of parameter values for a rational comparison of algorithms.

In this study, 30 separate runs were performed for each method on each problem instance (18 benchmark and 6 real-world) to gather all of the findings.

B. PERFORMANCE ANALYSIS OF EXISTING ALGORITHMS

This section investigates whether existing methodologies can effectively deal with the 'employee turnover' dynamic event. For this event, we investigate both state-of-the-art (SOA) [12] and baseline algorithms. We evaluate for a

TABLE 5. Tasks properties of Real_3 data instance.

Task ID	Status	Effort	HeadCount	Skills
1	1	3	6	1,3,5
2	1	2	3	2
3	1	2	5	4,5
4	1	2	4	1,4,5
5	1	3	6	3,5
6	1	1	3	4
7	1	4	6	4,5
8	1	3	3	1,2
9	1	2	6	1
10	1	2	5	1,4
11	1	2	3	2,5
12	1	3	3	3

TABLE 6. Parametrization (L = Individual Length).

State-of-the-art algorithm [12]	
Population size	100 individuals
Crossover	SBX, pc = 0.9
Mutation	polynomial, pm = 1.0 / L
No of evaluations	10,000
Baseline Approach	
Population size	100 individuals
Crossover	SBX, pc = 0.9
Mutation	polynomial, pm = 1.0 / L
No of evaluations	10,000

wide range of data instances. In Fig. 5, the findings are represented as a clustered stacked graph. In the graph, one column represents one objective against one data instance. The first column represents cost and 2^{nd} is duration objective. We normalise objective values such that they are all on the same scale. The results indicate that neither of the algorithms exhibit a discernible pattern in handling new dynamic events. In certain cases, baseline outperforms state-of-the-art algorithm and vice versa. All these aspects necessitate the need for a novel method to handle 'employee turnover' event.

In Table 7, we also present objective values for real data instances against each algorithm for the 'employee turnover'. The findings show that baseline outperforms the state-of-the-art algorithm for the 'duration' objective in all real-world scenarios. For the 'cost' objective, the state-of-the-art algorithm outperforms baseline algorithm while providing the lowest cost.

EAs provide a set of non-dominated solutions when tackling any NP-hard problem. [34]. Therefore, in our experiments, only a single solution is presented from solution set according to decision-maker choice [38]. In a real setting, software project manager is provided with a set of non-dominated solutions found by EAs which are subject to manager's choice. The project manager may choose a solution based on the optimal duration, cost, or trade-off among all project objectives. However, in our experiments, having a human decision-maker is not a practical choice. Therefore, an automated decision-making approach [38] is used.

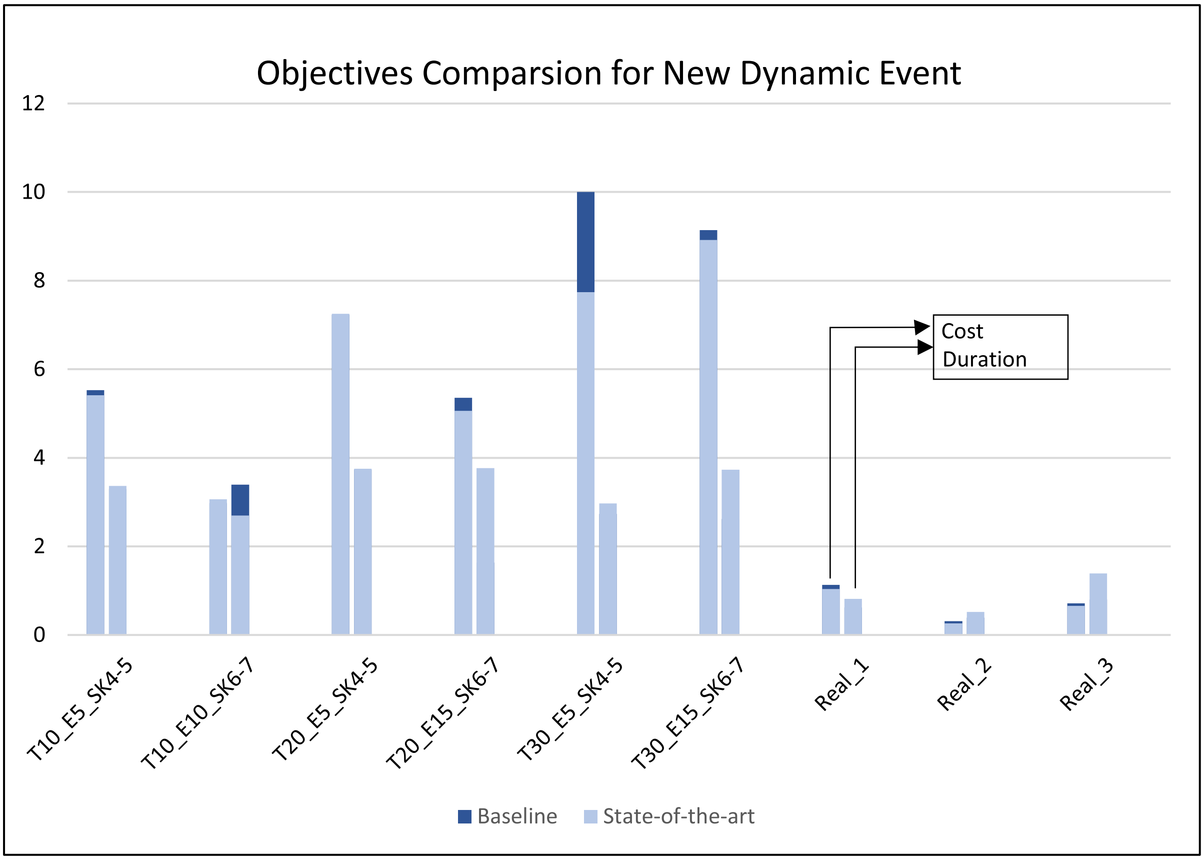


FIGURE 5. State-of-the-art and baseline algorithm performance for 'employee turnover' dynamic event.

TABLE 7. Duration and cost values comparison for real data instances. The selected solution against each instance is according to decision maker result [38].

	Real_1		Real_2		Real_3	
	SOA	Baseline	SOA	Baseline	SOA	Baseline
Duration	8.12E+00	**6.20E+00**	5.22E+00	**3.91E+00**	13.9E+00	**7.99E+00**
Cost	**1.52E+05**	1.65E+05	**3.81E+04**	4.55E+04	**9.55E+04**	1.04E+05

C. PROJECT DELAY BY RESCHEDULING EXISTING EMPLOYEES

This section explores whether reassigning existing employees using our suggested technique (when an employee quits) significantly put the project back schedule, assuming that existing employees have the necessary project skills. If an employee leaving affects a task, heuristic solutions are created at the scheduling point t'. These solutions are variations on current employees assignments that can do that activity by meeting the requisite skills. For example, if two employees are assigned to an affected task, two distinct solutions will be developed. In the alternative, if one employee quits and the remaining assigned employees can complete task skills, no heuristic solution is provided. The history solutions and variations are employed in this scenario.

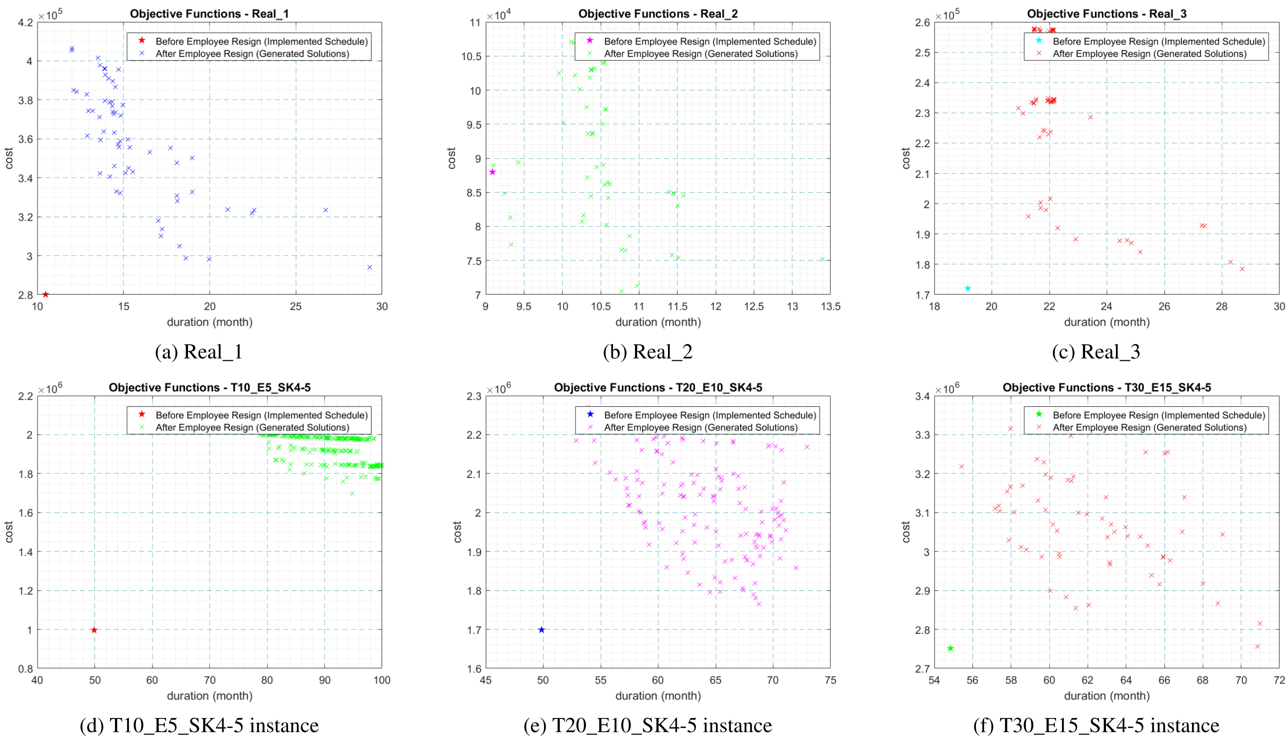
After rescheduling, we present results in Table 8 for duration and cost objectives for benchmark and real data instances. It can be observed that for all the data instances, after the 'employee turnover' dynamic event occurs, the duration is increased. Project duration and cost frequently go hand in hand. The duration of the two real instances, Real_1 and Real_3, has significantly increased, therefore, cost has also increased. Real_2 experiences less project delays and lower costs. Regarding benchmark data, both duration and cost has been increased with the exception of instance 'T30_E15_SK6-7'. Here, the duration objective exhibits a considerable increase, but the cost is reduced. This is because the departing employee had a significantly higher salary compared to others.; the same is true for Real_2. The results show that if existing employees are reassigned with the tasks, the project cost and duration increases on average 22% and 4% respectively as shown in Table 9. The visual representation of these results for some instances is also shown in Fig. 6. The manager has the option to decide whether they are content with the project's delay.

D. SKILL LEVEL FOR NEW HIRED EMPLOYEE

A replacement employee with the same skill set may be difficult to locate in the market if an employee resigns. Using our proposed technique, we analyse what the lowest proficiency value of a new recruiting employee may be if the software project manager increases both project time and cost

TABLE 8. Duration and cost values comparison for data instances after rescheduling. The selected solution is average of 30 independent runs.

	Before DE	After DE	Before DE	After DE	Before DE	After DE
	Real_1		Real_2		Real_3	
Duration	1.26E+01	1.98E+01	9.79E+00	1.00E+01	2.25E+01	3.12E+01
Cost	2.68E+05	2.91E+05	9.12E+04	8.97E+04	1.72E+05	1.85E+05
	T10_E5_SK4-5		T20_E5_SK4-5		T20_E10_SK4-5	
Duration	4.93E+01	9.69E+01	5.10E+01	9.85E+01	4.74E+01	6.02E+01
Cost	1.01E+06	1.62E+06	1.52E+06	2.17E+06	1.71E+06	1.92E+06
	T30_E10_SK4-5		T30_E15_SK4-5		T30_E15_SK6-7	
Duration	4.27E+01	5.73E+01	5.40E+01	6.18E+01	5.48E+01	6.83E+01
Cost	2.11E+06	2.27E+06	2.71E+06	2.86E+06	2.59E+06	2.56E+06



(a) Real_1 (b) Real_2 (c) Real_3

(d) T10_E5_SK4-5 instance (e) T20_E10_SK4-5 instance (f) T30_E15_SK4-5 instance

FIGURE 6. Rescheduling of existing employees for data instances.

TABLE 9. Duration and cost average percent values for real data instances after rescheduling.

	Real_1			Real_2			Real_3		Average	
	Before DE	After DE	%	Before DE	After DE	%	Before DE	After DE	%	%
Duration	1.26E+01	1.98E+01	36%	9.79E+00	1.00E+01	2%	2.25E+01	3.12E+01	28%	22%
Cost	2.68E+05	2.91E+05	8%	9.12E+04	8.97E+04	-2%	1.72E+05	1.85E+05	7%	4%

values. Alternatively, if manager keeps the same threshold as previous, then, what kind of employee could be hired. This is examined using three real instances namely, Real_4, Real_5, Real_6. In the event of an employee resignation, Real_4 has one missing skill (SK: 9) that none of the current employees have; Real_5 and Real_6 have two (SK: 8, 9) and one (SK: 9) missing skills respectively. During this experiment, we permit the shuffling of one to two employees, ensuring that people are not moved between teams and that team stability is maintained [39].

In Table 10, results of these three instances are shown. We begin with same duration and cost values and increment them step by step. Our aim is to determine the lowest proficiency value for each missing skill that a new hire may

TABLE 10. Skill's proficiency values for new hiring employee.

Real_4 (Skills of leaving employee: 1,2,9)				
Skill's Proficiency 0.1, 0.1, 0.1	Dur = same, Cost = same No solution found	Dur = same, Cost = 2% solution found	Dur = 5 days, Cost = same No solution found	Dur = 5 days, Cost = 0.6% solution found
Real_5 (Skills of leaving employee: 8,9)				
Skill's Proficiency 0.1, 0.1	Dur = same, Cost = 5% No solution found	Dur = same, Cost = 15% solution found	Dur = 5 days, Cost = same No solution found	Dur = 6 days, Cost = 13% solution found
Real_6 (Skills of leaving employee: 1,2,4,5,9)				
Skill's Proficiency 2, 2, 2, 2, 0.1	Dur = same, Cost = same No solution found	Dur = same, Cost = 23% solution found	Dur = 6 days, Cost = same No solution found	Dur = 6 days, Cost = 8% solution found

TABLE 11. Duration and cost average percent values with new employee hiring in comparison to rescheduling of existing employees.

New resources with right skills and proficiency	Instances	Real_4	Real_5	Real_6	Average
	Employees	10	8	5	8
	Tasks	42	43	91	59
	Project Duration (days)	45	65	26	45
Optimal increase in duration and cost	Duration Increase (days)	5	5	6	12%
	Cost Increase (%age)	1%	13%	8%	7%
Optimal increase in cost only	Duration Increase (days)	0	0	0	0%
	Cost Increase (%age)	2%	15%	23%	13%

have. The reason for this is because employees with the same skill level as the leaving employee may not be easily accessible in the market. For Real_4, algorithm fails to find any solution while keeping the duration and budget same. However, we find a solution with bit increase of cost with same duration. Another solution reflects increase in both duration and cost values. Real_5 yields identical findings as Real_4, i.e., no solution is found within the same duration and budget. Furthermore, it can be noted that there is significant increase in the cost value to find an employee with minimum skill's proficiency level. The results of Real_6 show that if we keep the duration same and missing skill's proficiency level to minimum, then, cost is increased significantly. It also demonstrates a trade-off between conflicting objectives that increasing the duration reduces the cost. To fulfil the criteria in this case, employees with greater proficiency values are required. It might be because this instance has 5 employees working on 91 tasks. The results reveal the fact that in comparison to rescheduling of existing employees, hiring new employees with recommended skills and proficiency provides 9% decrease in project cost with no increase in duration and 15% decrease in cost against 11% referenced increase in project duration as shown in Table 11.

VII. CONCLUSION AND FUTURE WORK

This study addresses the SPS problem in a dynamic environment, particularly dealing with 'employee turnover'. Employee resignations can lead to increased costs, decreased production, team disruptions, and loss of valuable knowledge. Therefore, effectively managing turnover is essential for project success. The paper introduces a model and an evolutionary approach which incorporates domain knowledge to generate the initial population for tackling this new dynamic event. Six instances of real-world problems and 18 dynamic benchmarks are used to evaluate the suggested approach.

The findings demonstrate that our suggested approach effectively addresses the dynamic event of 'employee turnover'. The model effectively handles both scenarios of employee resignations and new employee hiring. Upon an employee resignation if existing employees are reassigned with the tasks, the project cost and duration increases on average 22% and 4% respectively. We argue that replacing the leaving employee with an alternate resource having right set of skills and proficiency will optimize project duration and cost, affected from this disruption. The proposed dynamic SPS model and MOEA-based evolutionary approach identifies the alternate resource with required skills and proficiency. The results show that hiring new employees with recommended skills and proficiency based on proposed approach results in decreasing the cost by 9% for the fixed project duration; and 15% decrease in cost against 11%, as an optimal increase for rescheduled project duration.

For rescheduling, the project manager is provided with set of optimal solutions for informed decision making. We show that how our novel method provides insights into various skills for hiring a new employee and helps the software companies. In the future, we will take into account additional variables and dynamic events that may occur in the SPS problem, e.g., task's due-date, task removal, and task precedence change. Another interesting direction is the consideration of global software development.

ACKNOWLEDGMENT

This research was published originally as a thesis cited in [4].

REFERENCES

[1] N. Nigar, M. K. Shahzad, S. Islam, S. Kumar, and A. Jaleel, "Modeling human resource experience evolution for multiobjective project scheduling in large scale software projects," *IEEE Access*, vol. 10, pp. 44677–44690, 2022.

[2] K. M. Sallam, R. K. Chakrabortty, and M. J. Ryan, "A reinforcement learning based multi-method approach for stochastic resource constrained project scheduling problems," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114479.

[3] N. Nigar, "Model-based dynamic software project scheduling," in *Proc. 11th Joint Meeting Found. Softw. Eng.*, Aug. 2017, pp. 1042–1045.

[4] N. Nigar, "Multi-objective dynamic software project scheduling: An evolutionary approach for uncertain environments," Ph.D. dissertation, Comput. Sci., Univ. Birmingham, Birmingham, U.K., 2021.

[5] E. Alba and J. Franciscochicano, "Software project management with GAs," *Inf. Sci.*, vol. 177, no. 11, pp. 2380–2401, Jun. 2007.

[6] F. Xie, H. Li, and Z. Xu, "Multi-mode resource-constrained project scheduling with uncertain activity cost," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114475.

[7] M. N. Mahdi, M. H. M. Zabil, A. R. Ahmad, R. Ismail, Y. Yusoff, L. K. Cheng, M. S. B. M. Azmi, H. Natiq, and H. H. Naidu, "Software project management using machine learning technique—A review," *Appl. Sci.*, vol. 11, no. 11, p. 5183, 2021.

[8] V. Suresh and D. Chaudhuri, "Dynamic scheduling: A survey of research," *Int. J. Prod. Econ.*, vol. 32, no. 1, pp. 53–63, 1993.

[9] J. E. Moore and L. A. Burke, "How to turn aroundturnover culture'in it," *Commun. ACM*, vol. 45, no. 2, pp. 73–78, 2002.

[10] T. K. Abdel-Hamid, "A study of staff turnover, acquisition, and assimilation and their impact on software development cost and schedule," *J. Manage. Inf. Syst.*, vol. 6, no. 1, pp. 21–40, Jun. 1989.

[11] X.-N. Shen, L. L. Minku, N. Marturi, Y.-N. Guo, and Y. Han, "A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling," *Inf. Sci.*, vol. 428, pp. 1–29, Feb. 2018.

[12] X. Shen, L. L. Minku, R. Bahsoon, and X. Yao, "Dynamic software project scheduling through a proactive-rescheduling method," *IEEE Trans. Softw. Eng.*, vol. 42, no. 7, pp. 658–686, Jul. 2016.

[13] J. Xiao, L. J. Osterweil, Q. Wang, and M. Li, "Dynamic resource scheduling in disruption-prone software development environments," in *Proc. Int. Conf. Fundam. Approaches Softw. Eng.* Cham, Switzerland: Springer, 2010, pp. 107–122.

[14] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. Hoboken, NJ, USA: Wiley, 2001.

[15] J. Rong, L. Hongzhi, Y. Jiankun, F. Tao, Z. Chenggui, and L. Junlin, "A model based on information entropy to measure developer turnover risk on software project," in *Proc. 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol.*, Aug. 2009, pp. 419–422.

[16] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.

[17] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *J. Scheduling*, vol. 12, no. 4, pp. 417–431, Aug. 2009.

[18] S. M. Abbasi and K. W. Hollman, "Turnover: The real bottom line," *Public Personnel Manage.*, vol. 29, no. 3, pp. 333–342, Sep. 2000.

[19] J. L. Price, *The Study of Turnover*. Ames, IA, USA: Iowa State Press, 1977.

[20] S. Dibble, *Keeping Your Valuable Employees: Retention Strategies for Your Organization's Most Important Resource*. Hoboken, NJ, USA: Wiley, 1999.

[21] C. A. A. Mamun and M. N. Hasan, "Factors affecting employee turnover and sound retention strategies in business organization: A conceptual view," *Problems Perspect. Manage.*, vol. 15, no. 1, pp. 63–71, Mar. 2017.

[22] W.-N. Chen and J. Zhang, "Ant colony optimization for software project scheduling and staffing with an event-based scheduler," *IEEE Trans. Softw. Eng.*, vol. 39, no. 1, pp. 1–17, Jan. 2013.

[23] N. Nigar, M. K. Shahzad, S. Islam, O. Oki, and J. M. Lukose, "Multi-objective dynamic software project scheduling: A novel approach to handle employee's addition," *IEEE Access*, vol. 11, pp. 39792–39806, 2023.

[24] X. Shen, Y. Guo, and A. Li, "Cooperative coevolution with an improved resource allocation for large-scale multi-objective software project scheduling," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 106059.

[25] J. Cheng, J. Ji, Y.-N. Guo, and J. Ji, "Dynamic multiobjective software project scheduling optimization method based on firework algorithm," *Math. Problems Eng.*, vol. 2019, pp. 1–13, Jul. 2019.

[26] Y. Guo, J. Ji, J. Ji, D. Gong, J. Cheng, and X. Shen, "Firework-based software project scheduling method considering the learning and forgetting effect," *Soft Comput.*, vol. 23, no. 13, pp. 5019–5034, Jul. 2019.

[27] G. F. da Silva, L. Silva, and A. Britto, "Dynamic software project scheduling problem with PSO and dynamic strategies based on memory," in *Proc. Brazilian Conf. Intell. Syst.* Cham, Switzerland: Springer, 2020, pp. 79–94.

[28] S. Chand, H. Singh, and T. Ray, "Evolving heuristics for the resource constrained project scheduling problem with dynamic resource disruptions," *Swarm Evol. Comput.*, vol. 44, pp. 897–912, Feb. 2019.

[29] H. Chen, G. Ding, S. Qin, and J. Zhang, "A hyper-heuristic based ensemble genetic programming approach for stochastic resource constrained project scheduling problem," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114174.

[30] B. W. Boehm, "Software risk management: Principles and practices," *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, Jan. 1991.

[31] K. C. Laudon and J. P. Laudon, *Management Information Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[32] R. F. Scott and D. B. Simmons, "Predicting programming group productivity a communications model," *IEEE Trans. Softw. Eng.*, vol. SE-1, no. 4, pp. 411–414, Dec. 1975.

[33] M. L. Shooman, *Software Engineering: Design, Reliability, and Management*. New York, NY, USA: McGraw-Hill, 1980.

[34] F. Chicano, F. Luna, A. J. Nebro, and E. Alba, "Using multi-objective metaheuristics to solve the software project scheduling problem," in *Proc. 13th Annu. Conf. Genetic Evol. Comput.*, 2011, pp. 1915–1922.

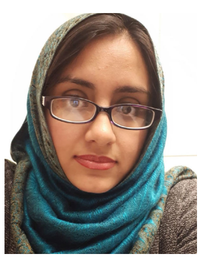
[35] F. P. Brooks, *The Mythical Man-Month*. White Plains, NY, USA: Longman Publishing Co., 1995.

[36] C. K. Chang, H.-Y. Jiang, Y. Di, D. Zhu, and Y. Ge, "Time-line based model for software project scheduling with genetic algorithms," *Inf. Softw. Technol.*, vol. 50, no. 11, pp. 1142–1154, Oct. 2008.

[37] D. Golenko-Ginzburg and A. Gonik, "Stochastic network project scheduling with non-consumable limited resources," *Int. J. Prod. Econ.*, vol. 48, no. 1, pp. 29–37, Jan. 1997.

[38] X.-N. Shen and X. Yao, "Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems," *Inf. Sci.*, vol. 298, pp. 198–224, Mar. 2015.

[39] M. Cohn, *Succeeding With Agile: Software Development Using S*. London, U.K.: Pearson, 2010.



NATASHA NIGAR received the Ph.D. degree from the School of Computer Science, University of Birmingham, U.K., in 2021. She is currently an Assistant Professor with the Department of Computer Science, University of Engineering & Technology Lahore, Pakistan. From 2008 to 2009, she was a Lab Engineer with the National University of Computer and Emerging Sciences, Lahore, Pakistan. She was a Software Engineer with Palmchip Pvt. Ltd., Lahore, from 2009 to 2011; and a Senior Software Quality Assurance Engineer with Netsol Technologies, Lahore, from 2011 to 2013. Her research interests include computational intelligence, optimization in dynamic and uncertain environments, and machine learning. She was awarded Faculty Development Program Scholarship to pursue the Ph.D. studies.

MUHAMMAD KASHIF SHAHZAD received the bachelor's degree in engineering from UET Lahore, Pakistan, in 2000, and the master's and Ph.D. degrees in industrial systems engineering from the University of Grenoble, France, in 2008 and 2012, respectively. He is currently a Chief Technical Officer (CTO) with the Ministry of Energy, Power Division, Government of Pakistan, Power Information Technology Company (PITC). His research interests include data models interoperability, advanced software engineering, technology, smart grid solutions, and engineering data management. He has vast experience of working in large scale European research and development projects IMPROVE and INTEGRATE. He specializes in designing and delivering technology driven smart grid solutions and working with USAID in developing solutions to improve Pakistan's power sector in deregulated market. He has 38 publications and two book chapters. Moreover, he has more than 20 years of professional experience designing, business process re-engineering, and managing large scale software development projects.

SHAHID ISLAM received the B.S. and M.S. degrees in computer science from the University of Engineering & Technology Lahore (UET Lahore), Pakistan, in 2003 and 2008, respectively. He is currently an Assistant Professor with the Rachna College, UET Lahore. His research interests include cloud computing, machine learning, semantic web, m-learning, and intelligent agent applications.

OLUKAYODE OKI (Member, IEEE) received the Ph.D. degree from the University of Zululand, South Africa, in 2019. He is currently a Lecturer with the Department of Information Technology, Walter Sisulu University, South Africa. He has authored more than 40 articles. His research interests include biologically inspired computation, ICT4D, communication networks, the IoT, machine learning, data analytics, and climate-smart agriculture. He has received several grants both for research and development and to attend conferences. He was a recipient of the South Africa National Research Foundation (NRF) Rated Researcher Award, an Honorary Rosalind Member of *London Journal* press, and a member of the IEEE South Africa Subsection.

JOSE MANAPPATTUKUNNEL LUKOSE received the Ph.D. degree from Fort Hare University, South Africa. He is currently the HoD and a Senior Lecturer with the Department of Information Technology, Walter Sisulu University, South Africa. His research interests include, but not limited to ICT in education and ICT for sustainable development.