A DYNAMIC MULTI-OBJECTIVE MODEL FOR FAIR WORKFORCE SCHEDULING WITH SYNCHRONIZATION AND ERGONOMIC CONSTRAINTS FOR LOGISTICS PLATFORMS

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
KOCAELİ UNIVERSITY

BY

PARMIS SHAHMALEKI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

KOCAELİ 2021

A DYNAMIC MULTI-OBJECTIVE MODEL FOR FAIR WORKFORCE SCHEDULING WITH SYNCHRONIZATION AND ERGONOMIC CONSTRAINTS FOR LOGISTICS PLATFORMS

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
KOCAELI UNIVERSITY

BY

PARMIS SHAHMALEKI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

Prof. Dr. Alpaslan FIĞLALI		
Supervisor,	**Kocaeli University**	**..........................**
Asst. Prof. Dr. Celal ÖZKALE		
Jury member,	**Kocaeli University**	**..........................**
Asst. Prof. Dr. Hatice ESEN		
Jury member,	**Kocaeli University**	**..........................**
Assoc. Prof. Dr. Gülşen AYDIN KESKİN		
Jury member,	**Balıkesir University**	**..........................**
Asst. Prof. Dr. Ahmet CİHAN		
Jury member,	**Düzce University**	**..........................**

Thesis Defense Date: 08.01.2021

ACKNOWLEDGMENT

Firstly, I wish to express my sincere gratitude and warmest thanks to my supervisor, Prof. Dr. Alpaslan Fığlalı for his continuous support, guidance, motivation, and supervision throughout my doctorate studies. His professional approach and practical knowledge helped me to complete my doctorate in the best possible way. It wouldn't be possible without his great support. Along with, I am grateful to Assist. Prof. Dr. Ahmet Cihan for his invaluable time, comments, suggestions and contributions to my research work. Besides my professors, I am really thankful to all my friends. I am also very thankful to all the academic personnel of the Industrial Engineering Department at Kocaeli University, Turkey. Last but not least, I am obliged to my mother Parvin and my father Ahmad for their unconditional love, support and prayers. I'm very much thankful to my brothers and sister for their love and support.

December, 2020 Parmis SHAHMALEKI

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

kj	:	Number of required teams for task j
dj	:	Processing time of task j
vj	:	Value of task j
TS	:	Total hours per shift
xij	:	1 if team i assigned to task j otherwise 0
yj	:	1 if task j selected for being processed otherwise 0
tij	:	Starting time of team i on task j
$setup_{jj'}$	:	if j and j' belong to the high ergonomic score group (risky tasks)
$s_{ijj'}$	:	1 if both task j and task j' processed by team$i$, 0 otherwise
$o_{ijj'}$	:	1 if task j processed before task j' by team$i$, 0 otherwise
$o_{ij'j}$	:	1 if taskj' processed before task$j$ by team$i$, 0 otherwise
$p_{ii'j}$	:	1 if both team i and team i' work on task j at the same time, 0 otherwise
C_j	:	Completion time of task j
Z	:	Objective function
EP_j	:	Ergonomic point of task j
TEP_i	:	Total Ergonomic point for team i
TWT_i	:	Total working time for team i

Abbreviations

AGV	:	Automated Guided Vehicle
BAPH	:	Branch and Price Heuristic
BPC	:	Branch and Price and Cut
CP	:	Constraint Programming
DOE	:	Design Of Experiments
EAWS	:	Ergonomic Assessment Worksheet
EMO	:	Evolutionary Multi-objective Optimization
EP	:	Ergonomics Point
GRASP	:	Greedy Randomized Adaptive Search Procedures
HHC	:	Home Health Care
JSI	:	Job Strain Index
MCDM	:	Multi Criteria Decision Making
MILP	:	Mixed Integer Linear Programming
MIP	:	Mixed Integer Programming
MPTRSP	:	Multi period Technician Routing and Scheduling Problem
MOMP	:	Multi-Objective Mathematical Problem
MRPSC	:	Manpower Routing Problem with Synchronization Constraints
NLE	:	NOISH Lifting Equation
OCRA	:	Occupational Repetitive Actions
REBA	:	Rapid Entire Body Assessment
RHS	:	Right Hand Side
RULA	:	Rapid Upper Limb Assessment

SA	:	Simulated Annealing
TOPSIS	:	The Technique for Order of Preference by Similarity to Ideal Solution
TTSP	:	Technician and Task Scheduling Problem
VRP	:	Vehicle Routing Problem
WSRP	:	Workforce Scheduling and Routing Problem

LOJİSTİK PLATFORMLAR İÇİN SENKRONİZASYON VE ERGONOMİK KISITLAR ALTINDA İŞGÜCÜNÜN ADİL ÇİZELGELENMESİ İÇİN DİNAMİK ÇOK AMAÇLI BİR MODEL

ÖZET

Bu çalışmada, çapraz sevkiyat platformlarının paketleme bölümünde ortaya çıkan bir çizelgeleme problemi incelenmektedir. Paketleme fonksiyonları sınırlı miktarda iş gücü ile gerçekleştirildiğinden, çapraz sevkiyatın dahili operasyonları arasında çok önemli bir aşamayı oluşturmaktadır. Bu problem, uygulamadaki işçi takımları arasındaki senkronizasyon ve ergonomik sınırlamalar gibi kısıtların yanısıra; çeşitli yönetimsel kısıtlamaları da içermektedir. Bu yüzden çapraz sevkiyat uygulamalarında, işgücünün planlanması karmaşık ve önemli bir problem olarak ortaya çıkmaktadır. Bu araştırmada, çapraz sevkiyatta , paketleme bölümündeki işgücünün planlanması için bir karar destek sisteminin tasarlanması ve uygulanması hedeflenmektedir. Otomotiv endüstrisine ait gerçek bir problem için dinamik bir model ve çözüm yöntemi önerilmektedir. Problemin NP-zor grupta yer alması nedeniyle, büyük ölçekli problemlerde uygun sürede optimum çözümün elde edilmesi mümkün değildir. Bu nedenle, çok amaçlı dinamik bir çözüm yöntemi ve yeni bir yapıcı sezgisel algoritma önerilmiştir. Önerilen çözüm yöntemleri, hem gerçek veriler, hem de oluşturulmuş veri kümeleri üzerinde uygulanmıştır. Manuel ve zaman alıcı mevcut planlama yöntemi ile karşılaştırıldığında, önerilen yöntemin makul sürelerde iyi sonuçlar ürettiği görülmektedir.

Anahtar Kelimeler: Adil Çizelgeleme, Ergonomi, Lojistik Platformlar, Personel Planlaması, Senkronizasyon Kısıtlamaları.

A DYNAMIC MULTI-OBJECTIVE MODEL FOR FAIR WORKFORCE SCHEDULING WITH SYNCHRONIZATION AND ERGONOMIC CONSTRAINTS FOR LOGISTIC PLATFORMS

ABSTRACT

This research studied a practical scheduling problem arising in the repackaging phase of cross-dock platforms. Packaging/repackaging is one of the significant concerns in cross-dock internal operations, where most of the tasks have been done by a number limited of teams. The problem also contains many practical constraints such as synchronization between teams and ergonomic aspects, as well as several managerial constraints. These conditions make internal workforce scheduling a complex and important issue for cross-docks. Implementation of the decision support system for planning and manpower scheduling in the repackaging phase of a cross-dock is encouraged us for this research. We try to model a real-world problem from automotive industry. Due to the non-deterministic polynomial time hardness (NP-hard) of the problem, finding a good solution is difficult. Therefore a dynamic solution method and a novel greedy construction heuristic algorithm are proposed. We apply these methods for real problem and generated instances. While the current process of planning and scheduling is manually and time consuming, our proposed algorithms generate good results for real problem instances in reasonable times.

Keywords: Fair Scheduling, Ergonomics, Logistic Platforms, Workforce Scheduling, Synchronization Constraints.

INTRODUCTION

Scheduling is the allocation of limited resources to a set of tasks over time and also is a decision-making process that tries to optimize one or more goals. This process is used regularly in many production, information, and service systems such as manufacturing systems, transportation, distribution systems, and other types of industries.

Scheduling as an essential tool in production and engineering has a significant impact on a system's efficiency. There are three main elements in scheduling, which are resources, tasks, and time. The resources may include machinery of the workstations, workforce (e.g., workers of production lines, teachers, nurses, and drivers) and, transportation equipment. On the other hand, the tasks may include processing operations (e.g., reaction, separation, blending, and packaging), service operations (e.g., teaching, driving, and surgery), or other activities like transportation, cleaning the place, and changeovers. The time at which the tasks have to be performed needs to be optimized, considering the required resources's availability and restrictions [1,2].

Various scheduling models and methods developed over the last five decades have been used for covering scheduling and sequencing problems related to different fields, including machine scheduling, airline scheduling, and project management.

The literature frequently emphasized the significant impact of scheduling and sequencing decisions on system performance. Human resource or manpower is one of the essential components of systems that have an essential impact on their productivity.

Workforce planning is an area of constantly increasing importance in an industrialized and knowledge-intensive society. For many organizations, labor cost is the chief direct cost components; therefore, reducing this cost by only a few percent via implementing a new personnel schedule could be very profitable [2].

Scheduling theory and human resource management are closely tied to each other in most business environments. Recognizing this fact can bring many opportunities for improvements in practical problems [3].

There is a large potential gain in applying optimization theory to practical workforce planning and scheduling problems. One of the main areas is the logistic platforms.

This thesis is about personnel scheduling in logistic platforms, especially in cross-dock or distribution centers.

1. WORKFORCE SCHEDULING IN CROSS-DOCKS

In a modern competitive market, companies try to globalize their activities to benefit from economies of scale and maintain competitiveness. Thus, most of them try to establish their facilities in different countries due to reducing their costs regarding the logistics and manufacturing operations. So international logistic flows significantly increase [4]. In these logistic patterns, for improving efficiency and lowering costs, reducing inventory in warehousing at every step of operations is a central concern [6]. Thereby, using cross-dock platforms is one of the main strategies that help companies cope with their global challenges.

Cross-docking tries to reduce warehouses to transshipment centers where the storage of products is limited or nonexistent, and its leading functions are receiving and shipping items [5].

In other words, a cross-dock is an intermediate node in the supply chain that reduces the cost of storing and inventory. In addition, Due to high fluctuating demand (customer's need), cross-docks (distribution centers) have to show high flexibility to be able to attract customers. Since buying machines and adapting them to the customer's need is rarely cost-effective; therefore, most of the cross-docks operations are carried out by human beings. So, the qualified workforce is in high demand and accounts for a significant part of total expenses in cross-docks.

In general, in a cross-dock, the working environment can be divided into three different areas. These three zones are inbound area, internal (treatment) area, and outbound area. Figure 1.1 depicts the general flow shop and operations of cross-dock.

In the cross-dock, at the inbound area, large incoming loads from different suppliers are unloaded, unpacked, disaggregated, and placed. Successively, items based on customer demands are sorted. Whenever repackaging is needed, items are transported to an intermediate area for deconsolidation, sorting, repackaging, and consolidation operations; otherwise, they are sent directly to the outbound zone. Once the items are consolidated in this area, the containers are built up and finally loaded onto outbound vehicles and sent to customers.

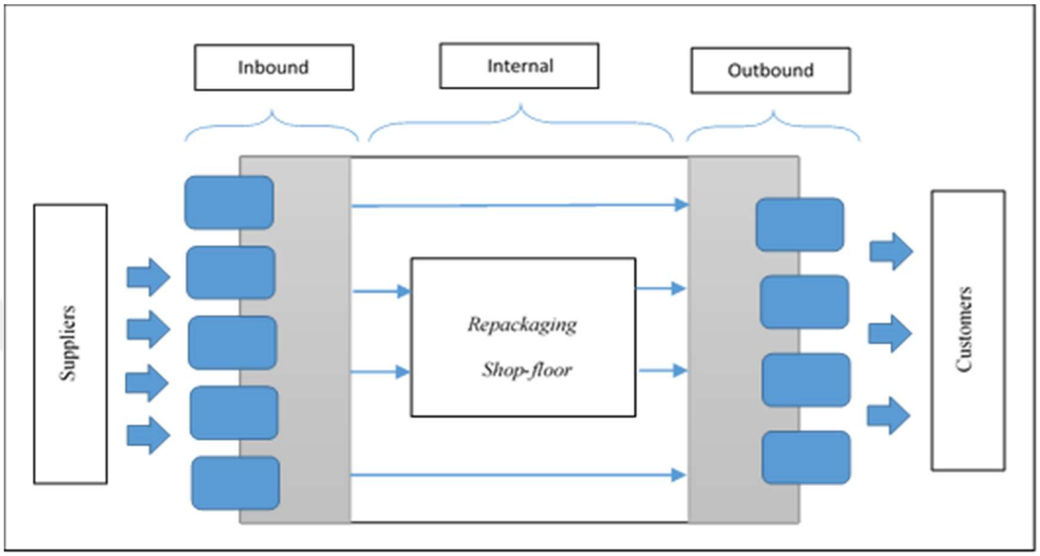

Figure 1.1. Cross-dock layout

This study focuses on the workforce in the intermediate work area and especially on the repackaging phase.

Generally, the main research subjects concerning cross-docking are considered as either strategical, tactical, or operational level problems [5]. Using this classification, the problem that we address here is considered to be an operational problem. Most of the intermediate work area activities are heavy physical activities and done by workers, so the workforce is one of the main factors that affect the productivity of the whole cross-dock. Since there is a limited number of workers (teams) that process tasks, so the efficient management of such workforce in the context of workforce allocation and scheduling becomes a priority.

When workload should be divided among workers, allocation, and scheduling processes are compiled.

A challenge is that the workload is variable: the number of arriving trucks and the number of orders to be prepared change every day. Besides, within the standard 8-hour shift, the emergency orders enter the system, or in other words, there is a dynamic environment.

Therefore, in order to minimize costs, to maximize the utilization of the available workforce, and to ensure a high level of satisfaction among the workers, sophisticated scheduling methods are required. On the day of operation, the objective is to assign as much work as possible to the available workforce while respecting various requirements and rules.

From one perspective, the under-researched problem can be considered as a combination of task scheduling problem and manpower allocation problem. In order to solve it, both sub-problems must be considered simultaneously because the separate solution of each of them will not necessarily lead to the optimal solution of the problem.

A clarified description of the manpower allocation problem for the repackaging phase in a cross-dock is as follows: different tasks on the shop floor demand varying numbers of teams. A planning center dispatches teams to satisfy this demand by considering the specified features of this real-world problem.

Furthermore, due to this, cross-dock's customer service strategy, finishing high priority tasks as early as possible is desirable in the planning period for the managers. It is noteworthy that the planning horizon is short, and daily schedules are needed. Therefore, our primary goal in this research is to create a daily schedule for tasks and workers so that the number of processed tasks will be maximized by taking into account the distinctive features of this real-world problem.

Considering workforce properties simultaneously with the sequencing of tasks, make the problem so intricate and so real.

This thesis is concerned with a subfield of scheduling focused on manpower planning in logistic platforms generally and Cross-Docks specifically. This problem originated from a logistic framework related to the automobile industry, but it can be generalized to a wide range of practical problems.

1.1. Contributions And Originality Of This Research

Under researched problem has special features that distinguish it from available models in the literature. The contributions of this thesis are as follows:

1- Combining synchronization between teams in a short-time horizon plan for the workforce.
2- Considering synchronization, ergonomic, and fairness constraints in a workforce scheduling problem.
3- Solving the multi-objective workforce scheduling problem through sequential solving, each step being modeled by a mixed-integer linear program solved to optimal.
4- Considering different and conflicting objective functions simultaneously and proposing a user-friendly method that dynamically generates Pareto solutions at each step. The user can change the direction of solutions via determining varied weights for objective functions.

This problem, with all its characteristics, has no longer been formerly addressed in the literature to the best of our knowledge.

The rest of this research is structured as follows. In section 2, the literature related to our work has been reviewed. We describe the model with details in section 3. We presented the solution methods in section 4. In section 5, the computational results are provided. Finally, in section 6, we conclude our paper and suggest possible opportunities for future research.

2. LITERATURE REVIEW

Our review will focus on four aspects: firstly, section 2.1 focuses on the personnel scheduling, its importance, and its application fields especially related to the logistics field. Secondly, in section 2.2, we investigate the synchronization concept in scheduling theory. Thirdly in section 2.3 we have a closer look at ergonomic concept and the researches that embedded this concept in scheduling and planning methods. Finally, section 2.4 focuses on the definition of fairness in scheduling theory and related formulas and studies.

2.1. Workforce Scheduling in Logistics Framework

Since business becomes more service-oriented and cost-conscious, workforce allocation and scheduling have become increasingly important. The complexity of allocation and scheduling of workforce varies based on the size and requirements of the organization but the main concern is utilizing the manpower resource efficiently and productively. An optimized workforce schedule can significantly benefit while obliging satisfying workplace agreements, shift equity, and other requirements [6].

In general personnel scheduling consists of three main steps: 1. determine the number of necessary staff, 2. assigning (allocating) staff to days (or shifts), and 3. Assigning tasks to staff (individual or members) [6]. These steps can be solved simultaneously or in sequence, depending on the context.

Ladier et al. [7] developed an employee scheduling decision support tool for a logistics platform consisting of three subproblems based on the type of decision: workforce dimensioning, weekly task allocation, and detailed daily rostering. They proposed a mixed-integer linear programming model for each subproblem and solved them sequentially so that the output of the previous model is the input for the next model.

M.Pour et al. [8] Proposed a mixed-integer programming model for the preventive signaling crew scheduling problem. This problem consists of many practical and managerial constraints. They developed a hybrid Constraint programming and MIP framework. They used CP to generate warm-start solutions for the MILP solver.

In contrast with the problem tackled in this paper, threshold frequencies for the services are not considered.

Serrano et al. [4] proposed an integer linear programming model as an alternative method for the current method for distribution and operation planning at a cross-dock platform. They applied their method to Renault Company.

Jutte et al. [9] studied the railway crew scheduling with various objective functions included minimizing net schedule cost, minimizing schedule unpopularity, and minimizing schedule unfairness. They embedded the unfairness aspect by adding penalty costs in the objective function. They analyzed the influence of heightened fairness on cost via applied a column generation-based solution algorithm. They defined and measured unfairness in a railway crew scheduling context.

Castillo-Salazar et al. [10] made a comprehensive survey on workforce scheduling and routing problems (WSRP). In this kind of problem, the workforce should perform tasks in various locations hence, transportation is inevitable. Various examples of these sorts of problems consist of technicians carrying out repairs at customers' locations, security guards performing rounds at different premises, and nurses visiting patients at home.

It is noticeable that there are only a few numbers of research related to workforce scheduling problems in logistic platforms.

2.2. Synchronization and Scheduling

2.2.1. Definition and concept

In the literature, for this concept, there are different keywords. The words that are used in this research are synchronization, simultaneous execution, cooperation, collaborative works, and team building.

Synchronization of workers and vehicles has played an important role in diverse industries, including logistics, health care, and airport ground handling [11]. In general, in the operation research field, the synchronization concept comes up when, for doing a task or an operation, more than one agent (vehicle, worker) should cooperate at the same time. Therefore, synchronization constraints are used to model these situations. These constraints impose a temporal dependency between agents [12]. This topic has started to be one of the hot research fields in vehicle routing problems (VRP).

This concept could be used to model more realistic problems in different fields such as VRPs, nurse scheduling, and home health care.

Drexl [12] reviewed the vehicle routing problems with multiple synchronization constraints comprehensively. He classified the synchronization in VRP into Task synchronization, Operation synchronization, Movement synchronization, Load synchronization, and Resource synchronization in such a way that each of these types has some subdivisions. Operation Synchronization category consists of three subcategories: Pure Spatial Operations, Operations with Precedence, and Exact Operations.

Based on the classification proposed by Drexl, our model can be put in the Operation Synchronization category and the Exact Operation subcategory.

Synchronization between agents arises in different contexts, where cooperation between agents is required to perform tasks. An example is a media company, which shoots programs at different locations. Synchronization and cooperation between directors, camerapersons, recorders, stage workers, and other staff for each program in each location are vital for this kind of company. Agents can be vehicles such as trucks, buses, airplanes, AGVs, or human beings such as doctors, nurses, or technicians.

One of the important issues in synchronization context when the agent is a human being is constructing a team of individuals to perform tasks. The composition of each team is affiliated with the requirements of the task.

Workforce can be homogeneous or heterogeneous. For performing some tasks, a team of different workers proficient in various skills (heterogeneous) is essential, for example, technicians in maintenance. In contrast, some tasks need more than one worker with the same ability and performance (homogenous); for example in home health care, sometimes more than one caregiver is necessary for conveying a patient. This situation obliged the job-teaming constraints to scheduling models.

Nowadays, in the industrial and business environment, grouping workers in the form of teams is common [13]. Some tasks, depending on their nature, require more than one workforce. The members of a team may have different skills. If a job requires a team of workers, everyone on the team should be present at the job location and initiate execution simultaneously.

Another example is the home health care sector. Some patients may require cooperation between personnel with different skills; therefore, they travel between different patients' locations in a working day.

This kind of problem may occur in hospitals where some tasks (surgery) demand collaborative work between doctors and nurses, and a composition of them may vary for various tasks.

Another context for this problem is where technicians with different skills need to perform service tasks. In other words, the combination of technicians with various skills is required to perform each task, such as the maintenance provider in maintenance companies or the service provider in airports.

Another classification is based on whether workers remaining together for the entirety of the working period. In some situations, a team's workers stay together for the whole working period [14], but in other situations, different workers or teams of workers collaborate temporarily for a task.

Labadie et al. [15] define two kinds of synchronization in VRP: simultaneous synchronization and precedence synchronization. According to simultaneous synchronization, related workers should start performing simultaneously when a task needs more than one worker. On another side, in precedence synchronization, a task may need more than one worker, but these workers should perform sequentially.

In our research, according to Labadie et al.'s categorization, the simultaneous synchronization fits the situation that is happening in the packaging section of the cross-dock.

2.2.2. Literature review for synchronization

References are listed in chronological order below:

Job-teaming constraints have been introduced by Li et al. [13]. They have studied a situation originating from the port of Singapore. Due to different demands in the yard of the port, the workforce allocation problem for service personnel was a big issue for managers. They formulate this problem as a multi-objective problem. The main objective is to minimize the number of servicemen required to do all tasks. They used a metaheuristic approach.

In Lim et al. [6]'s research, the manpower allocation with time windows for servicemen in a port in Singapore is addressed. They have considered various objective goals, consists of minimization of the number of required servicemen, travel distances, travel times, and waiting times. A Tabu-embedded Simulated Annealing algorithm and a squeaky wheel optimization with a local search algorithm were proposed. The effectiveness of their proposed method has been proven via experimental results based on real data.

Dohn et al. [15] consider a manpower allocation and scheduling problem for ground handling tasks in some European airports. They formulate this problem as an integer programming model. Both tasks and teams have been restricted by time windows. Besides, some tasks need cooperation between teams. The number of teams is restricted; therefore, the objective function is the maximization of the number of assigned tasks. The scheduling horizon is one working day (24 hours). A branch and price approach with column generation have been proposed for solving 12 realistic test instances. The proposed solving method shows promising results.

Zamorano and Stolletz [14] have studied the Multi-period Technician Routing and Scheduling Problem (MPTRSP). Their study originated from an external maintenance provider company where tasks have time windows and require different skill proficiencies. Workers have qualifications in different skills, and also their proficiency level in each skill is various from each other. Several parallel decisions have to be made: daily assignment of technicians into teams, assigning teams to tasks and assigning teams to daily routes. These decisions should be made in such a way that the total operation costs be minimized. They model a problem as a mixed-integer program, and due to it’s inefficiently for large instances, a branch-and-price algorithm has been conducted. The real data instances were evaluated, and the capability of the solution method was proven.

Cordeau et al. [16] considered the Technician and Task Scheduling Problem (TTSP) in a large telecommunication company where tasks vary in difficulty and have different skill requirements. Some tasks require more than one technician. On another side, technicians differ in proficiency levels in several skill domains. Technicians are paired into teams, and a team must stay together on a given day. Routing decisions are not considered in this problem. The objective function is minimizing makespan by considering precedence between tasks, task's skill requirements, workers availability, and other constraints. A construction heuristic and an adaptive neighborhood search heuristic for tackling large real instances are proposed. The results prove that the solution procedure is efficient.

Labadie et al. [17] have modeled the VRP with synchronization constraints as a mixed integer programming model. In their paper, the objective function was the minimization of total travel cost. A constructive heuristics and an iterated local search metaheuristic for solving small and medium-sized instances are presented.

Haddadene et al. [18] have studied the vehicle routing problem in HHC (home health care) structures with time windows where synchronization and precedence constraints are considered. A mixed-integer programming model is proposed. For producing high-quality solutions a greedy heuristic (GRASP), two local search strategies, two metaheuristics, and a hybrid metaheuristic approach are developed.

Afifi et al. [19] have developed a Simulated Annealing (SA) Algorithm for the VRP with time windows and synchronization constraints. They considered the minimization of traveling time as the objective function. Their algorithm considerably improved the best-known solutions for nine instances of the data sets in the literature.

Kim, Koo, and Park [20] have introduced a combination of VRP and scheduling of workers where each customer needs several tasks in a given order. The teams of workers should be transferred by vehicles. Although teams and vehicles are not bound to each other and could be scheduled separately, they should be synchronized. In their paper, the optimization criterion was finding the teams' efficient schedule and minimizing the total traveling cost. Besides proposing a mixed-integer programming model, they have developed a Particle Swarm Optimization algorithm.

Bredstrom and Ronnqvist [21] have studied the VRP and scheduling problem with synchronization constraints in dealing with an application in home health care. The synchronization constraints have been used to model the situation when some of the customers require simultaneous service. They proposed a branch and price algorithm for this problem when the objective function is minimizing the traveling time. In the generated instances maximum of 10 percent of customers need simultaneous visit.

Fink et al. [11] focused on operational ground handling planning in airports. They have studied “abstract vehicle routing problem with workers and vehicles synchronization”. Each task has a time window and maybe needs cooperation between more than one worke. In addition, the number and capacity of vehicles are limited. Two mathematical multi-commodity flow formulations based on the time-space network were suggested. A branch-and-price heuristic (BAPH) procedure was developed too. Based on Drexel categorization, they have modeled movement, resource, and task synchronizations.

Luo et al. [22] considered the manpower routing problem with synchronization constraints (MRPSC). A branch and price and cut (BPC) algorithm were proposed to find an exact solution. Experimental results show the effectiveness of this approach.

Cai et al. [23] investigated the manpower allocation problem with time windows for tasks and job teaming constraints. They propose a tree data structure for representing solutions. A novel Tabu search algorithm with new search operators based on the tree data structure is proposed.

Nasir and Kuo [24] suggested a decision support framework for creating a simultaneous schedule and route plans for caregivers and home delivery vehicles with considering synchronization between staff and vehicle’s visits, multiple visits to patients, multiple routes of vehicles, and pickup/delivery visits related precedence for vehicles. A MILP model and a hybrid genetic algorithm were developed.

None of the abovementioned researches exactly fit our studied model. In our study, synchronization between work teams in a short period with a large number of tasks on a shop floor is evaluated for the first time.

2.3. Ergonomics and Workforce Scheduling

2.3.1. Definition and concept

It is not enough to pay attention to the technical aspects alone to improve a system; also, it is necessary to consider environmental and ergonomic issues.The human-related characteristics should be considered. In integration between scheduling and human factors, these characteristics are affected by scheduling decisions.

Forasmuch as humans have different characteristics that distinguish them from other inanimate parts of the systems, contemplating these features lead to a more precise and realistic assessment [25]. So another important feature of our problem is the ergonomic aspect. As defined by International Ergonomics Association, "ergonomics" addresses the ways and methods to optimize the worker's well-being and overall system performance via improving the interaction between humans and other elements of the system [26].

The literature frequently emphasized the significant impact of scheduling and sequencing decisions on system performance. Besides, the human factor and ergonomics literature demonstrated the important role of sequencing the human tasks on human performance and well-being. So the interaction between scheduling and ergonomics may affect system performance positively. In their research, Carnehan et al. [27] showed that by increasing human performance via improving workers' ergonomic criteria such as reducing stress, fatigue, and work injury risk, the overall performance of the system enhanced consequently. However, typically in most scheduling problems, the effects of these human characteristics on human performance are ignored and several assumptions are used to simplify human behavior [3].

Boudreau et al. [3] summarized some assumptions that simplify human behavior in scheduling or, more generally, in the operation research field. These assumptions are as follows:

"1. People are not a major factor. Many models look at machines without people, so the human side is omitted entirely.

2. People are deterministic and predictable. People have perfect availability (no breaks, absenteeism, etc.). Task times are deterministic. Mistakes do not happen, or mistakes occur randomly. Workers are identical (work at the same speed, have the same values, and respond to the same incentives).

3. Workers are independent (not affected by each other, physically or psychologically).

4. Workers are "stationary." No learning, tiredness, or other patterns exist. Problem solving is not considered.

5. Workers are not part of the product or service. Workers support the "product" (e.g., by making it, repairing equipment, etc.) but are not considered explicitly as part of the customer experience. The impact of system structure on how customers interact with workers is ignored.

6. Workers are emotionless and unaffected by factors such as pride, loyalty, and embarrassment.

7. Work is perfectly observable. Measurement error is ignored. No consideration is given to the possibility that observation changes performance (Hawthorne effect)."

Obviously, the assumptions mentioned above are not reliable and applicable to optimize human performance, well-being, and productivity of the system.

Several factors can impact human performance directly or indirectly. Some of these factors are fatigue, stress, boredom, cumulative workload, and skill learning [28].

The workload can be divided into physical workload and mental workload. Also, the workload can be measured in different dimensions. Various workload assessment measures are used to assess the perceived demand of the task processed by the worker. Targeted working activities of these methods, as well as their desired level of details, provide differences in the application of them. The most frequently used ergonomic measurement methods are NLE (NIOSH Lifting Equation) [29], JSI (Job Strain Index) [30], RULA (Rapid Upper Limb Assessment) [31], REBA (Rapid Entire Body

Assessment) [32], EAWS (Ergonomic Assessment Worksheet)[33] and OCRA (Occupational Repetitive Actions) [34].

2.3.2. Literature review related to integrating ergonomics and workforce scheduling

The studies focusing on personnel scheduling and using the ergonomic considerations with fairness are summarized below:

Lodree, Jr. et al. [28] demonstrated the lack of collaboration between scheduling theory and human factor engineering. They proposed a framework for interdisciplinary connection between workforce assignment and scheduling with human factor engineering.

Hochdorffer et al. [35] presented a mathematical model for generating job rotation schedules. By considering the workers' qualifications, the workplace ergonomic, and workforce assignment, the complexity of the problem increases. The authors propose a linear programming based heuristic. Testing this method on real data from the assembly line of an automotive producer in Germany shows the effectiveness of the approach.

Otto and Battaia [36] focused on assembly line balancing, rotation schedule, and physical ergonomics risks in their survey article. They provided a comprehensive review of articles investigating the physical ergonomics risks for job rotation scheduling in line balancing concept. Also, they provide helpful insights and research directions for operation researchers, ergonomists, and production managers.

Yoon et al. [37] developed a mathematical model for generating job rotation schedules by considering reducing the cumulative workload in the automotive assembly line in Korea. Here the cumulative workload is related to successive use of the same body region. They try to reduce the daily workload variance between workers and prevent repeated high workload exposure on the same body region. They use rapid entire body assessment (REBA) for calculating workload. Their proposed model shows promising results in the ergonomic aspect, despite an increase in computational time.

Mutlu and Ozgurmus [38] have suggested a fuzzy linear programming model for balancing an assembly line with consideration physical workload of tasks. They applied this model in a textile company in Turkey. The results showed that the proposed model improved the assembly line's performance without influencing other main aspects of the problem.

2.3.3. Ergonomics assessment method

2.3.3.1. OCRA method

In this thesis, we focus on physical ergonomics risks, and we have used the OCRA method. OCRA [34] is an observational technique that allows quick evaluation of the exposure of upper limbs in repetitive works. A higher value of the OCRA index indicates higher ergonomics risks.

The OCRA checklist method has been used in different fields, for example, in poultry slaughterhouses [38], in animal facility operators [39], and packing line operators [40].

There are two kinds of OCRA method: the OCRA index and the OCRA checklist. The OCRA checklist is a simplified version of the OCRA index [41].

We used the OCRA checklist method for assessing risk measurement for “multitask” jobs in the repackaging section.

We use a five-level color system for facilitating the interpretation of the overall risk scores. This system reflects the ergonomic risks in categories. Green, yellow, light red, red, and purple indicates acceptable, very low, medium-low, medium, and high risk levels, respectively [40,42]. The OCRA checklist score is shown in Table 2.1.

Table 2.1. The OCRA checklist score

Checklist Score	Exposure Level
<= 7.5	No exposure
7.6-11.0	Very low exposure
11.1-14.0	Light exposure
14.1-22.5	Medium exposure
>=22.5	High exposure

The heaviness of each task is calculated via the OCRA method. As shown in the table above, if the ergonomics score of a task is above 22, it is considered a heavy (risky) task.

2.3.3.2. Literature review for the OCRA method

Paulsen et al. [43] examined the inter-rater reliability of two physical exposure assessment methods of the upper extremity, the Strain Index (SI) and Occupational Repetitive Actions (OCRA) Checklist.

Rosecrance et al. [44] conducted and compared the SI (Strain Index) and OCRA checklist method for evaluating the ergonomics risk of cheese processing tasks in a factory in Italy. Seven ergonomists assessed task-level physical exposures to the upper limb of workers performing 21 cheese-manufacturing tasks.

Tiacci and Mimmi [45] used the OCRA index as a method for ergonomics risk assessment to propose an approach to design asynchronous assembly lines in compliance with the ergonomics aspect in a company related to agricultural equipment. They suggested a genetic algorithm to tackle this problem.

Reis et al. [38] analyzed the risks associated with repetitive movements of workers' upper limbs performing meat processing tasks. The study was conducted in a slaughterhouse in Brazil. They used the OCRA checklist method.

Lasota et al. [40] used the OCRA checklist method for assessing risk measurement in the packaging operation in a factory in Poland. Their results showed the efficiency of this method for manual packaging activities.

2.3.4. Fairness and workforce scheduling

Fairness of work distribution among workers is an essential human resources issue. Assuming that dispensing the workloads fairly across colleagues in a work environment contributes to higher job satisfaction. Increasing job satisfaction, in turn, increases productivity and lowers the resignation rates of staff.

There are different ways to evaluate fairness in a schedule of workers. Some work schedules impose a high level of fatigue only on some of the workers, which raises unfair feelings among them.

Fairness criteria are divided into two distinguished categories: quantitative measures and qualitative measures. Quantitative criteria are measurable such as balancing time and balancing workloads. On another side, qualitative criteria are usually relevant to individual different preferences and availabilities.

In this research, we incorporate fairness in our model as balance constraints on working times. In other words, in our model, balancing work time between teams has been used as a measure of fairness that is a quantitative criterion.

When for a team, the amount of heavy work assigned to them is higher than the average per team, the schedule is perceived as unfair. This situation lowers the workers' job satisfaction and can decrease their performance and motivation [46]. Another important consequence of unfairness in a working environment is to increase the absenteeism and turnover rates of employees [47]. Even in one case in the Netherlands railways, the unfair distribution of duties among drivers led to widespread labor strikes [48].

Blochliger offers an initial idea about considering fairness in staff scheduling problems. Difference between the maximum and minimum outcomes of all employees and the standard deviation of all outcomes are suggested to measure fairness [49].

Simsons and Roberson [50] have done comprehensive research on unfairness in working environments and how an unfair work environment can affect the employees' productivity.

De Boer et al. [51] have studied the relationship between perceived unfairness and absenteeism at work among security guards.

Abbink et al. have successfully developed a model that integrates fairness aspects in the crew scheduling problem at railways in the Netherlands. In their model, the unpopular tasks comprise operating trains in unpopular areas and operating old rolling stock. The suggested operation research model helped in developing an alternative set of scheduling rules. This approach, called Sharing-Sweet-and-Sour, generated more variety in drivers' and conductors' duties and increased their satisfaction and efficiency. Moreover, due to implementing this model, personnel costs decreased approximately 1.2 percent per year [48].

Muhlenthaler and Wanka have introduced two new formulations for creating fair course timetables in the setting of a university. They included different notions of fairness. In addition, they proposed an optimization approach based on simulated annealing [52].

In the research done by Martin et al., they incorporate fairness constraints in the nurse rostering problem. They have developed an agent-based cooperative search framework. The fairness criteria were qualitative ones that consider the nurses' preferences. Existing real-world benchmark instances were used, and the overall results show that the proposed framework outperforms each metaheuristic run individually [53].

Stolletz and Brunner have proposed a reduced set covering formulation for solving the fortnightly physicians' scheduling problem with flexible shifts in a hospital. They incorporated fairness aspects in their model. They introduced two new fairness measures due to requirements in hospitals. They tried to distribute the working times and on-call services among physicians evenly. Despite the increase in computational time, the results showed the efficiency of this approach in realistic instances [54].

Jutte et al. have studied the crew scheduling problem arising in a large European railway freight. They considered the fairness aspects and unpopularity duties for drivers. They tried to solve this problem with three conflicting objective functions. They analyzed the interaction between objectives. The results show their proposed model can lower the unfairness of schedule while keeping a comparable level of schedule unpopularity and net schedule costs (two other objectives) [9].

Since tasks' assignment and scheduling are closely related to the distribution of workload among individuals, considering and analyzing the fairness of a schedule are inevitable in practice. In the proposed mathematical model, fairness constraints are used to minimize between-team variability in cumulative daily workload. In other words, Fair assignment of working hours has been modeled as the deviation of the actual share of working time assigned to one team from the overall average.

3. PROBLEM STATEMENT OF THE STUDY

Determining a model and a solution method to use in logistics platforms such as cross-docks and distribution centers to generate the workforce schedules is our primary goal. To be closer to an industrial context while defining the model and its assumptions, we monitored the scheduling and allocation process within a cross-dock where it was done manually.

This research focuses on the intermediate work area and especially on the repackaging phase in logistic platforms. Internal operations usually consist of deconsolidation, sorting, repackaging, and consolidation. At this phase, the packaging of parts is done by teams of workers and equipment. The repacked or prepared parts are loaded into pallets and successively in containers.

The preparation and repackage of goods is driven by international customers' orders. The daily schedule is produced a day before with consideration of the rolling horizon. The domain of scheduling is 8-hour shifts. However, the daily schedule can be redefined during the day and taking into account the possibly new orders (data, tasks) which may arrive in the meantime.

By taking a closer look at the scheduling and planning process, it can be seen that the whole procedure consists of two main steps:

In the first step, the tasks are prioritized based on specified predetermined criteria. The higher priority means greater importance and represents the partial importance of the tasks. These given priorities act as task weights in the objective function in the proposed model.

In the second step, the daily workforce schedule and task allocation are determined based on the available prioritized tasks at the moment.

As a convention, we will use the "task" word for each customer order for the rest of the thesis.

The workforce in this context is in the form of working teams. The number of teams and the number of individuals in each team is constant and does not change over time. Since the number of teams is limited, some tasks can be left unassigned. We try to assign a sequential order of tasks to teams to maximize the total (weighted) number of completed tasks by considering time fairness, ergonomics fairness, and synchronization constraints.

As a subordinate objective, tasks with higher priority should be tried to schedule earlier during the scheduling. Depending on general logistic platforms' customer-oriented service policy, this approach makes advantages on service levels and relations with international customers.

The assumptions made for the problem are as follows:

Assumptions:

• Processing times of tasks are definite and known in advance.
• The number of teams is fixed and known in advance.
• Preemption is not allowed.
• Ergonomic point (EP) of each task is stable during the working period and known in advance. They are calculated with the OCRA method by specialists.
• There is no precedence constraints between tasks.
• The number of workers in each team is fixed. Also, the number of workers in different teams is the same.
• Transportation times are negligible.
• The work shift is 8 hours.
• The workers are homogenous. In other words, they have the same ability, proficiency, and speed.
• Productivity of workers is stable during the working shift. The learning effect, fatigue are not considered explicitly.
• Tasks need the same skills to be processed.
• Some tasks need more than one team. The number of required teams for each task is definite and known in advance.
• Teams do not interrupt each other. If a team is free does not help other teams.
• Ergonomic point of each task is independent of its number of required teams.

• The value (priority level) of each task is definite and calculated by planners.

• Each team can be assigned to only one task at a time.

• Workload of a task distributes evenly among workers of a team assigned to that task.

This model relies on continuous-time.

The sequence of operations and the assignment of teams to tasks are not determined initially, and they are formed while solving the problem; so that the objective functions are optimized.

The model in it's entirely is explained in the following subsections.

3.1. Problem Definition

The problem is described as follows: There are N tasks to be done by M teams. A team is formed by a group of workers. The number of workers in each team is fixed, and workers have the same skills and experience. Teams are homogenous and identical. The number of teams is limited. Each team can process one task at a time. Each task is required to be processed by one or more teams, depending on its requirement. It means, in some situations, more than one team must collaborate to process one task. These teams should start a task at the same time; in other words there should be synchronization between teams.

The number of tasks varies daily. Since during a day the workload is varying over time, the environment is considered a dynamic one.

Each task has a predetermined priority (weight), deterministic processing time, due date, the number of required teams and heaviness score which is determined by the ergonomic score of the task.

On the other hand, as some of these tasks are ergonomically heavy, they must not be done consecutively by a team. (It is essential to eliminate sequential high workloads)

Transportation time is negligible compared to processing time. Therefore, the walking times for workers and transportation times for forklifts are neglected. Preemption is not allowed, which means that once a task is started, it must be finished even if a higher priority task arrives.

3.1.1. Objective Functions

There are four various objective functions. The main objective function should ensure that as many tasks as possible are completed within the time shift (planning horizon). The second one tries to minimize the total weighted completion times. This objective function is in line with the customer-oriented policy of the cross-dock. The third objective function aims to minimize between-team variability in cumulative daily workloads (workloads are related to ergonomic scores). Finally, the last objective function tries to distribute working time among teams as evenly as possible.

We clarify the mathematical model related to each objective function. Most constraints are the same for models related to different objective functions. Nevertheless, the required changes are applied where it is necessary.

3.1.2. Main Constraints

The two main hard constraints are synchronization between teams and eliminating sequential high workloads.

3.1.3. Mathematical Modeling

In this section, the main mathematical model with the first objective function is explained entirely. In this proposed model, the task scheduling problem and the workforce allocation are simultaneously modeled.

We refer to this model as MILP0. The notations given below are used:

M total number of teams

N total number of tasks

Indexes:

i, i' Teams $\mathrm{i}, \mathrm{i}' \in \mathrm{M}$ or $(\mathrm{i}, \mathrm{i}' = 1, \ldots, \mathrm{M})$

j, j' Tasks $\mathrm{j}, \mathrm{j}' \in N$ or $(\mathrm{j}, \mathrm{j}' = 1, \ldots, \mathrm{N})$

Parameters:

k_j Number of required teams for the task j

d_j Processing time of the task j

v_j Value of the task j or P_j priority level of the task j (relative importance of the task)

TS Total hours per shift

L A large number (L represents an arbitrarily large number to help bound some of the constraints)

Variables:

x_{ij} Equals 1 if team i assigned to task j otherwise 0

y_j Equals 1 if task j selected for being processed otherwise 0

t_{ij} Starting time of team i on task j

$setup_{jj'}$ if j and j' belong to the high ergonomic score group (risky tasks) it equals the predefined amount of resting time; otherwise equals 0.

$s_{ijj'}$ Equals 1 if both task j and task j' processed by team i, 0 otherwise

$o_{ijj'}$ Equals 1 if task j processed before task j' by team i, 0 otherwise (if team i work on task j before task j')

$o_{ij'j}$ Equals 1 if task j' processed before task j by team i, 0 otherwise (if team i work on task j' before task j)

$p_{ii'j}$ Equals 1 if both team i and team i' work on task j at the same time, 0 otherwise

C_j Completion time of the task j

Z Objective function

Based on the described assumptions, the mathematical model is formulated as follows:

MILP0:

Maximizing $\sum_{j=1}^{N} v_j . y_j$

Constraints:

$$\sum_{i=1}^{M} x_{ij} = k_j . y_j \qquad \forall j = 1, \dots, N \tag{3.1}$$

$$s_{ijj'} = x_{ij} . x_{ij'} \qquad \forall j, j' = 1, \dots, N, i = 1, \dots, M, j \neq j' \tag{3.2}$$

$$o_{ijj'} + o_{ij'j} = s_{ij\,'} \qquad \forall j, j' = 1, \dots, N, i = 1, \dots, M, j \neq j' \tag{3.3}$$

$$t_{ij'} - t_{ij} \geq d_j - L\left(1 - o_{ijj'}\right) + setup_{jj'} \quad \forall j, j' = 1, \dots, N, i = 1, \dots, M, j \neq j' \tag{3.4}$$

$$t_{ij} - t_{ij'} \geq d_{j'} - L\left(1 - o_{ij'j}\right) + setup_{jj'} \quad \forall j, j' = 1, \dots, N, i = 1, \dots, M, j \neq j' \tag{3.5}$$

$$p_{ii'j} = x_{ij} . x_{i'j} \qquad \forall i, i' = 1, \dots, M, j = 1, \dots, N, i \neq i' \tag{3.6}$$

$$t_{ij} - t_{i'j} \geq -L(1 - p_{ii'j}) \qquad \forall j, j' = 1, \dots, N, i = 1, \dots, M, j \neq j' \tag{3.7}$$

$$t_{i'j} - t_{ij} \geq -L(1 - p_{ii'j}) \qquad \forall j, j' = 1, \dots, N, i = 1, \dots, M, j \neq j' \tag{3.8}$$

$$C_j - t_{ij} \geq d_j \qquad \forall j = 1, \dots, N, i = 1, \dots, M \tag{3.9}$$

$$C_j \leq TS \qquad \forall j = 1, \dots, N \tag{3.10}$$

$$\sum_{j=1}^{N} x_{ij} . d_j \leq TS \qquad \forall i = 1, \dots, M \tag{3.11}$$

$$\sum_{j=1}^{N} y_j . d_j . k_j \leq M . TS \tag{3.12}$$

Constraints (3.1) ensure that if task i is selected, the number of assigned teams to this task should match exactly its requirement.

Constraints (3.2) and (3.3) ensure that there is some ordering among tasks assigning the same team for their execution. If task j and task j' both selected and assigned to the same team (team i), $s_{ijj'}$ equals 1, otherwise equals 0. In addition, if team i work on task j after task j $o_{ijj'}$ equals 1, otherwise $o_{ij'j}$ equals 1.

Constraint (3.4) and (3.5) ensure that sequences exist among starting times of tasks assigning the same team. Under these constraints, if, for example, task j is scheduled for team i immediately after task j', then the processing of task j cannot be started before the completion of processing of task j'.

Constraints (3.6), (3.7), and (3.8) ensure that synchronization exists among teams processing the same task. Under these constraints, if, for example, task j is assigned to team i and team i', both teams must start to process this task at the same time (simultaneously).

Constraints (3.9) determine the completion time of each team (completion time of each task), and constraints (3.10) ensure that the completion time of each team (completion time of each task) cannot exceed the shift time.

Constraints (3.11) ensure that the total number of tasks processed by each team cannot exceed the shift time (available time).

Constraints (3.12) ensure that the total number of tasks processed by all teams cannot exceed the total available time per shift.

To alter the proposed nonlinear mathematical model into the form of MILP, Constraint (3.2) and (3.6), with using linearization methods, changed to linear constraints. They changed as follows respectively:

$$x_{ij} + x_{ij'} - s_{ijj'} \leq 1 \quad \forall j, j' = 1, \ldots, N, i = 1, \ldots, M, j \neq j' \tag{3.13}$$

$$2s_{ijj'} \leq x_{ij} + x_{ij'} \tag{3.14}$$

$$x_{ij} + x_{i'j} - p_{ii'j} \leq 1 \quad \forall i, i' = 1, \ldots, M, j = 1, \ldots, N, i \neq i' \tag{31.5}$$

$$2p_{ii'j} \leq x_{ij} + x_{ij'} \tag{3.16}$$

With replacement, the constraints (3.13) and (3.14) with constraints (3.2) and (3.15) and (3.16) with constraint (3.6) in proposed mathematical model, a mixed-integer linear programming model for the problem is achieved.

MILP1:

By considering the customer-oriented policy, the orders with high priority are preferred to be completed as early as possible. So, the second objective function is defined as the minimization of total weighted completion times.

minimize $\sum_{j=1}^{N} v_j . C_j$

At this level, the constraints (3.1) are not required anymore. Because the selected tasks are already determined. However, the constraint (3.3) to (3.5), (3.7) to (3.16) remain the same.

MILP2***:*** **(**Modeling Ergonomic Fairness)

Workload (ergonomic loads) should be divided fairly as much as possible between teams to respect the safety and ergonomic principles.

There is varied sort of objective functions (or equations) to model the Ergonomic fairness, is this research, the use of absolute value function is preferred. Furthermore, some new necessary constraints are embedded in the model.

Parameters:

EP_j Ergonomic point of task *j*

Variables:

TEP_i Total Ergonomic point for team i

minimize $\sum_{i=1}^{M} a_i$

$$TEP_i = \sum_{j=1}^{N} EP_j . x_{ij} \qquad \forall i = 1, \ldots, M \tag{3.17}$$

$$\left| TEP_i - \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} EP_j . x_{ij}}{M} \right| \leq a_i \qquad \forall i = 1, \ldots, M \tag{3.18}$$

It can be seen that some of the constraints need to be linearized. If Z appears in the minimization objective, then a constraint $Z = |X - Y|$ can be linearized as follows: $Z = |X - Y| \leftrightarrow \begin{cases} Z \geq X - Y \\ Z \geq Y - X \end{cases}$ [7].

Constraints (3.18) are replaced by the following constraints:

$$TEP_i - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} EP_j.x_{ij}}{M} \leq a_i \tag{3.19}$$

$$TEP_i - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} EP_j.x_{ij}}{M} \geq -a_i \tag{3.20}$$

For this model, the constraints (3.1) are not required anymore. Because the selected tasks are already determined. However, the constraint (3.3) to (3.5), (3.7) to (3.16) remain the same.

MILP3: (Modeling Working Time Fairness)

An important part of planning for a manager is making sure every team has the right amount of work.

Different forms of modeling working time fairness exist in the literature, but here, similar the ergonomic fairness, using the absolute value function is preferred. The objective function is to minimize between-teams variance in cumulative shift work time (daily working time).

Parameters:

TWT_i Total working time for team i(number of hours worked by team i)

$$\text{minimize} \sum_{i=1}^{M} b_i$$

$$TWT_i = \sum_{j=1}^{N} d_j.x_{ij} \quad \forall i = 1, \ldots, M \tag{3.21}$$

$$\left|TWT_i - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} d_j.x_{ij}}{M}\right| \leq b_i \tag{3.22}$$

Since nonlinear constraints appear in the objective function and the objective function is minimization, the same linearization method is used. The new constraints are as following:

$$TWT_i - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} d_j.x_{ij}}{M} \leq b_i \tag{3.23}$$

$$TWT_i - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} d_j.x_{ij}}{M} \geq -b_i \tag{3.24}$$

4. SOLUTION METHODS

This research aims to build a detailed schedule for a given day for the workforce on a logistic platform's internal shop floor. We try to find good solutions in a reasonable time for this problem presented in the previous sections. The solution schedule is supposed to be obtained every morning or the night before to plan the upcoming day. Since workload is varying during the day, the problem is considered as a dynamic one. Therefore, to respond to high emergency orders received during the day, the daily schedule is redefined every two hours.

We propose two solution methods:

1. Dynamic multi-objective solution method
2. Heuristic algorithm

The heuristic algorithm produces feasible schedules considering minimizing total weighted completion time as a main objective function. This heuristic is a greedy construction heuristic, and it is fast and practical.

Secondly, the comprehensive dynamic method is proposed that considers all objective functions sequentially and simultaneously.

This hybrid framework consists two consecutive main phases, the first phase is a selection phase, and the second phase is the solution improvement or rescheduling phase.

In the selection phase, the MILP0 model is solved to optimality for a 2-hour interval. The Cplex solver is used to find an optimum solution regarding the first objective function (maximizing the total weighted number of completed tasks). The more tasks that are fulfilled, the better the plan is considered to be.

The obtained initial solution is solved via a hybrid method considering MILP2, MILP3, and MILP4 for further improvement in the rescheduling phase.

Each phase is clarified in more detail in the following sections.

Selection phase:

Under the condition that the workload is varying over time, the set of intervals are considered for the daily scheduling and allocation. For an 8-hour shift, four intervals are considered (It means that the duration of each interval is two hours).

The purpose of defining time intervals is to cover dynamic input data. In other words, the daily schedule is redefined (updated) every two hours with available prioritized tasks (it encompasses previous unassigned tasks and recently arrive tasks).

In the selection phase, the MILP0 model chooses the most profitable tasks among available tasks.

The output of this phase is the input of the next one.

There are three subordinate objective functions in the rescheduling phase:

1- Minimizing total weighted completion times

2- Minimizing the deviation of ergonomic risks (workload) between teams

3- Minimizing the deviation of work time between teams.

It is obvious that these objective functions are in conflict. (Figure 4.1)

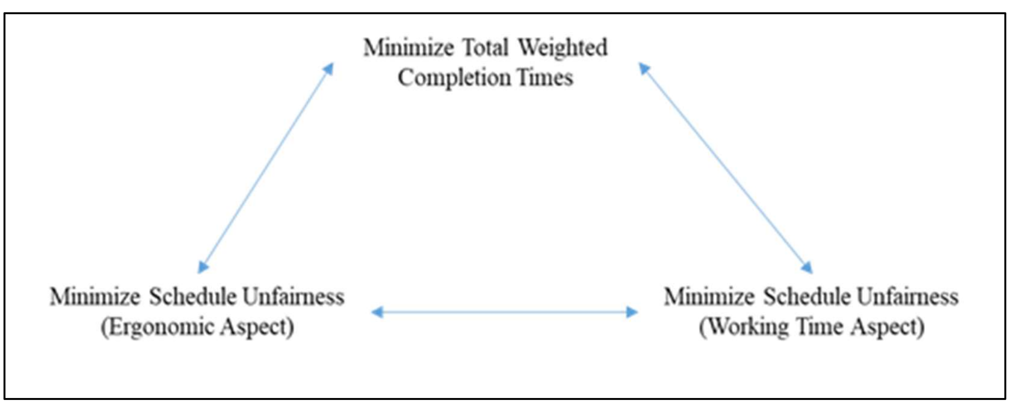

Figure 4.1. Conflicts of Objectives

Since there are three conflicting objective functions, using a multi-objective method for generating Pareto solutions is required.

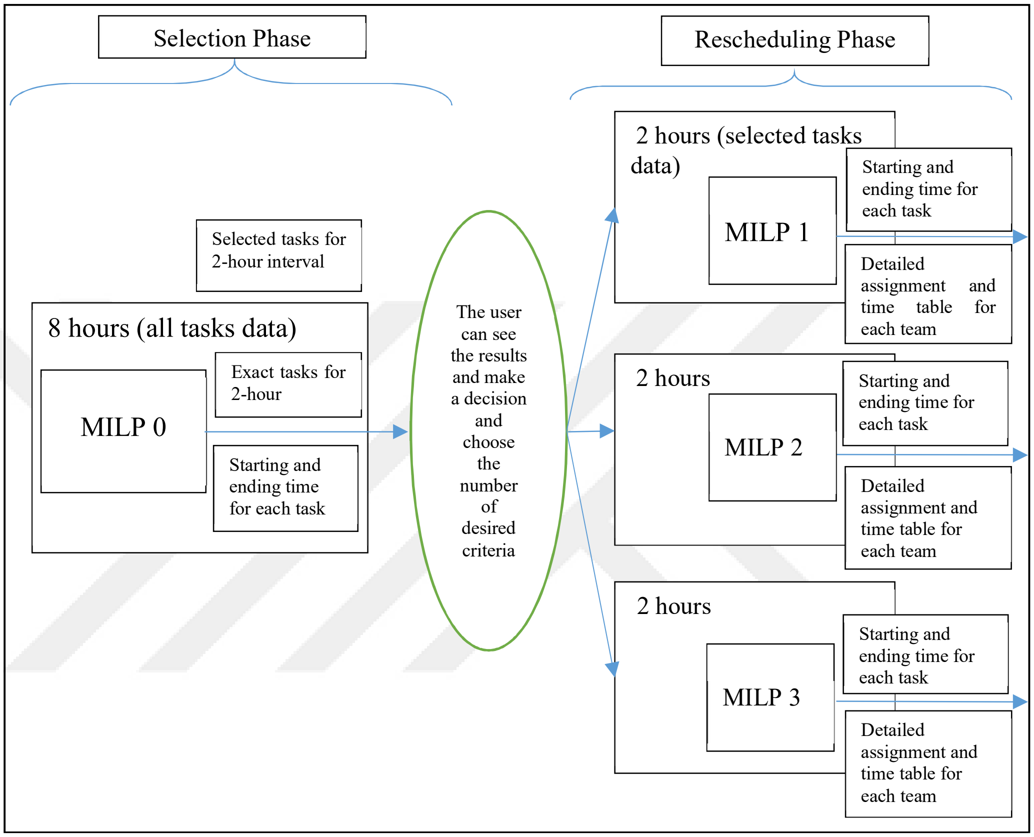

Figure 4.2. The workflow between Phase 1 and Phase 2 of Dynamic Solution Method

The relations between two distinctive phases are illustrated in Figure 4.2. The first mixed-integer linear programming model aims to create a daily schedule, giving each team their exact starting times and finish times and selecting the optimum set of tasks among the available tasks for the 2-hour interval. Then this feasible schedule is considered as an input for the second phase. In the second phase, using the first phase's output is tried to obtain better schedules (regarding other objective functions) if they exist.

4.1. Dynamic Multi-Objective Solution Method

4.1.1. Multi-objective mathematical programming

A multi-objective mathematical programming model (MOMP) includes more than one objective function, and most of the time, these objective functions are in conflict. Therefore there is no single optimal solution that optimizes all of the objectives simultaneously. Due to this conflict, finding a single solution that optimizes all objectives simultaneously is rare. In other words, optimizing one of the objective functions does not necessarily mean that the other ones are optimized too.

In such problems, the goal is searching and finding the most preferred solutions. In this framework, the concept of optimality is replaced by efficiency or Pareto optimality. The efficient solution, also known as Pareto solutions, non-dominated solutions, and non-inferior solutions, is the solution that cannot be improved in one objective function without deteriorating its performance in at least one of the rest [55].

Generally, the focus is on finding a set of solutions that define the best tradeoff between competing objective functions.

Identifying all Pareto solutions since it maximizes a decision maker's knowledge about the tradeoffs among different objectives for convenience of optimal decision making is desirable.

Determining all Pareto solutions since increasing a decision maker's knowledge about the interactions between various objective functions for convenience in optimal decision making is desirable. Nevertheless, unfortunately, until now, generating the entire set of Pareto solutions (non-dominated) has not been possible in many cases [56, [57].

Although there are multiple Pareto optimal solutions, in practice, only one solution has to be selected for implementation.

According to widely accepted classification, the multi-objective mathematical programming techniques can be classified into three categories, based on how and when to incorporate preferences from the decision-maker into the decision-making process: the a priori methods (prior to the search), the interactive methods (during the search) and the posteriori methods or generation methods (after the search).

In a priori methods, the decision-maker provides his/her preference information beforehand. These preferences can be set of goals or weights to the objective functions. Although the “a priori” methods seem attractive, in practice, it is very difficult for the decision-maker to know and to quantify his/her preferences in advance. The provided preferences can be so optimistic or pessimistic. Some well-known a priori methods comprise Goal Programming, Goal-Attainment Method, and Lexicographic Method.

The interactive methods are the most decision-maker intensive ones. In the interactive methods, the decision maker expresses his/her preferences iteratively and progressively directs the search process with his/her answers towards the most preferred solution. The disadvantage is that the decision-maker never sees the whole Pareto solutions or an approximation of it. Some of the interactive methods include the Method of Geoffrion-Dyer-Feinberg (GDF) [58], Tchebycheff Method [59], Reference Point Methods [60], Light Beam Search [61].

In the a posteriori or generation methods, the decision-maker is involved only after the problem is solved to select the most preferred solution among the obtained Pareto solutions. Here solving a MOP means generating the efficient solutions of the problem (all of them or a sufficient representation).

The a posteriori methods present more information to the decision-maker and strengthen his/her confidence for making the final choice via generating the whole efficient solutions.

The disadvantage of these methods is that they need more computation time and effort to generate Pareto optimal solutions. In addition, selecting a reasonable and compromise solution from such a large set is potentially intractable for the decision-maker. The disadvantage of these methods is their computational effort. In other words, the calculation of the efficient solutions is usually a time-consuming process. Some of the commonly used posteriori methods consist weighted sum method, epsilon constraint method [62], EMO algorithms [63].

For this problem, a posteriori method is considered. An augmented ε -constraint method is among the highly applied methods, and it produces good results.

4.1.2. The Conventional ε -constraint Method

For the following multi objective mathematical problem:

$$Min \quad \left(f_1(x), f_2(x), \dots, f_p(x)\right)$$

$$x \in S$$

Where x refers to the vector of decision variables, S is the feasible region and subscript p represents the number of competing objective functions.

A feasible solution x is said to be efficient, and the corresponding objective vector is said to be non-dominated, if there is no other feasible solution x' satisfying $f_i(x') \geq f_i(x)$ for every $i = 1,2, \dots, p$ with at least one strict inequality.

A general form of ε-constraint method for the mentioned problem is formulated as below:

$$Min\ f_1(x)$$

Subject to

$$f_2(x) \leq e_2$$

$$f_3(x) \leq e_3$$

$$\dots$$

$$f_p(x) \leq e_p$$

$$x \in S$$

In this method, one of the objective functions is optimized by considering the other objectives as constraints and incorporating them in the constraint part of the model.

Pareto optimal solutions are obtained by parametrical variations in the right-hand side (RHS) of the constrained objective functions $(e_2, e_3, \dots, e_p)$.

4.1.3. The augmented ε-constraint method

In this research, the augmented ε -constraint method presented by Mavrotas [64] is employed to generate the Pareto solutions of the multi-objective optimization problem.

The conventional e-constraint method has two drawbacks: first, the obtained solution may not be Pareto-optimal, and second, they may be weakly efficient. The augmented e-constraint method overcomes the first weakness by using the lexicographic optimization for organizing the payoff table according to the priorities of the objectives and optimizing the objective functions based on these priorities.

The AUGMECON method is the abbreviation name for this method.

In this method, inequality constraints of constrained objectives are transformed to Equality constraints by introducing non-negative slack variables or surplus variables and then augments the objective function with the summation of these slack or surplus variables. The mentioned model reformulated as follows:

$$max \quad f_1(x) + \delta(s_2/r_2 + s_3/r_3 + \dots + s_p/r_p)$$

$$f_2(x) + s_2 = e_2$$

$$f_3(x) + s_3 = e_3$$

$$\dots$$

$$f_p(x) + s_p = e_p$$

$$x \in S \text{ and } s_i \in R^+$$

Where δ is an adequately small number usually between 10^{-3} and 10^{-6} and r_i is the range of the ith objective.

The AUGMECON method only produces Pareto solution (efficient solutions). For proof, see Mavrotas [64].

The steps of augmented ε-constraint method are as the following:

1. Using the lexicographic method, the payoff table values are calculated.
2. One of the objective functions is selected as the main objective function of the problem.
3. The best and the worst value of each sub-objective function are extracted from the payoff table. (maximum and minimum value of the corresponding column in the payoff table for the maximization model)
4. The scope (range) of each of the sub-objective functions is calculated.
5. The range of sub-objective functions is divided into a predetermined number according to the number of Parto response desired (Each of the values obtained from this division for the objective function f_i, as one e_i is used.
6. Set the main objective function as the model objective function and other objective functions in the constraints.
7. The obtained model will be solved for each value of e_i. Answers obtained for each of the values $\varepsilon_2 \dots \varepsilon_k$, is one of the Pareto solutions to the problem.
8. Another objective function is considered as the main objective function of the model and the steps of the algorithm are repeated. The whole algorithm is repeated for each of the objective functions as the main objective function.
9. All generated Pareto solutions are reported.

The comprehensive diagram of augmented ε-constraint method is illustrated in Figure 4.3.

None of the obtained Pareto solutions can be said it is better than the others without additional information about the decision-makers' preferences. So each multi-criteria optimization approach should be a combination of optimization and decision support. The decision support system, with considering the preferences of decision-makers, helps them to select the most preferred single solution among generated Pareto solutions. In other words, after the optimization step and generating Pareto solutions, the next step, with considering the decision-makers' preferences, helps them to select the most preferred single solution among the whole obtained Pareto solutions.

In this research, in the second step (decision-support), using one of the MCDM methods (Multi-Criteria Decision Making) is considered. The derived Pareto solutions are considered as discrete alternatives, and by considering the decision-makers' preferences and using the TOPSIS method, the most desirable solution will be selected.

The proposed iterative hybrid method is described in the next section in more detail.

4.1.4. Description of the dynamic multi-objective solution method

The main goal of this research is to find good solutions in a reasonable time for the daily planning of the workforce assignment and task scheduling problems presented in the previous section. Since the current manual procedure is subjective and time-consuming, finding a more practical method is desirable. Besides, since the environment is dynamic, redefining the daily schedule is necessary. When the new emergency tasks enter the system, they should be considered, and the daily schedule should be redefined to include them. We propose a dynamic multi-objective method that includes three main phases: selection phase, reschedule& reassignment phase, and the decision making phase. In this procedure, a daily shift is divided into intervals. Each interval is solved separately, but they are connected, and there is a data flow between them. These phases are clarified in the following sections in more detail.

In the selection phase, the MILP0 model solved to optimality; it finds the optimum schedule for a 2-hour interval. MILP0 model's goal is to assign high-value tasks as much as possible. Input for this mathematical problem is all available tasks at the moment, and its the output is a 2-hour schedule. Each solution is constructed by assigning tasks to teams and schedule assigned tasks within the 2-hour interval such that every constraint is satisfied. It is noteworthy that the interval duration can be changed depending on the managers' thoughts and different circumstances. It can be shorter or longer, but the too short interval may prevent efficient scheduling; while, the too long interval can decrease the response speed to customers.

One of the most challenging parts of this problem is considering conflicting objective functions. As mentioned previously, besides the maximization of the value of completed tasks, there are three more crucial objectives that can affect the system's performance.

The generated schedule in the selection phase may not be a good schedule regarding fairness aspects or minimizing total weighted completion times (customer oriented policy). The fulfillment of the first objective function does not mean that the other objective functions are fulfilled too. In the rescheduling and reassignment phase, we are faced with three objective functions: minimizing the total weighted completion time, minimizing ergonomic deviation between teams, and minimizing the deviation of working time between teams. As previously explained, these objectives are in conflict; for example, decreasing schedule ergonomic deviation does not necessarily induce a reduction in schedule working time fairness. Therefore, to cope with this challenge, the augmented ε-constraint method is used. This method is applied successfully to a wide range of mixed-integer linear programming methods. Since there is no priority among objectives in this phase, all combinations of objective functions priority are used in the Lexicografic method to generate the pay-off table. The input of this phase is the selected tasks in the selection phase (previous phase). Then during the iterative process, the Pareto solutions are found.

In the decision-making phase, based on the obtained Pareto solutions and predetermined weights for objective functions the TOPSIS method is applied, and the Pareto solutions are ranked. The weights of objective functions can be determined using trial and error tests or interviews with managers and expert opinions. The output of this phase is a high ranked Pareto solution that is ready for applying in the workshop.

In each iteration of this algorithm, three mentioned phases are processed consecutively.

After the first iteration and generating the schedule for the first interval, the algorithm continues to generate a daily schedule for the next interval by considering the unassigned tasks and newly arrived tasks as input tasks. This procedure is continued and repeated until the termination criteria are met. The termination criteria include (a) 8-hour shift (available time) is over or (b) there is no unassigned task in the system.

The overall process of the proposed dynamic multi-objective solution method is depicted in Figure 4.4.

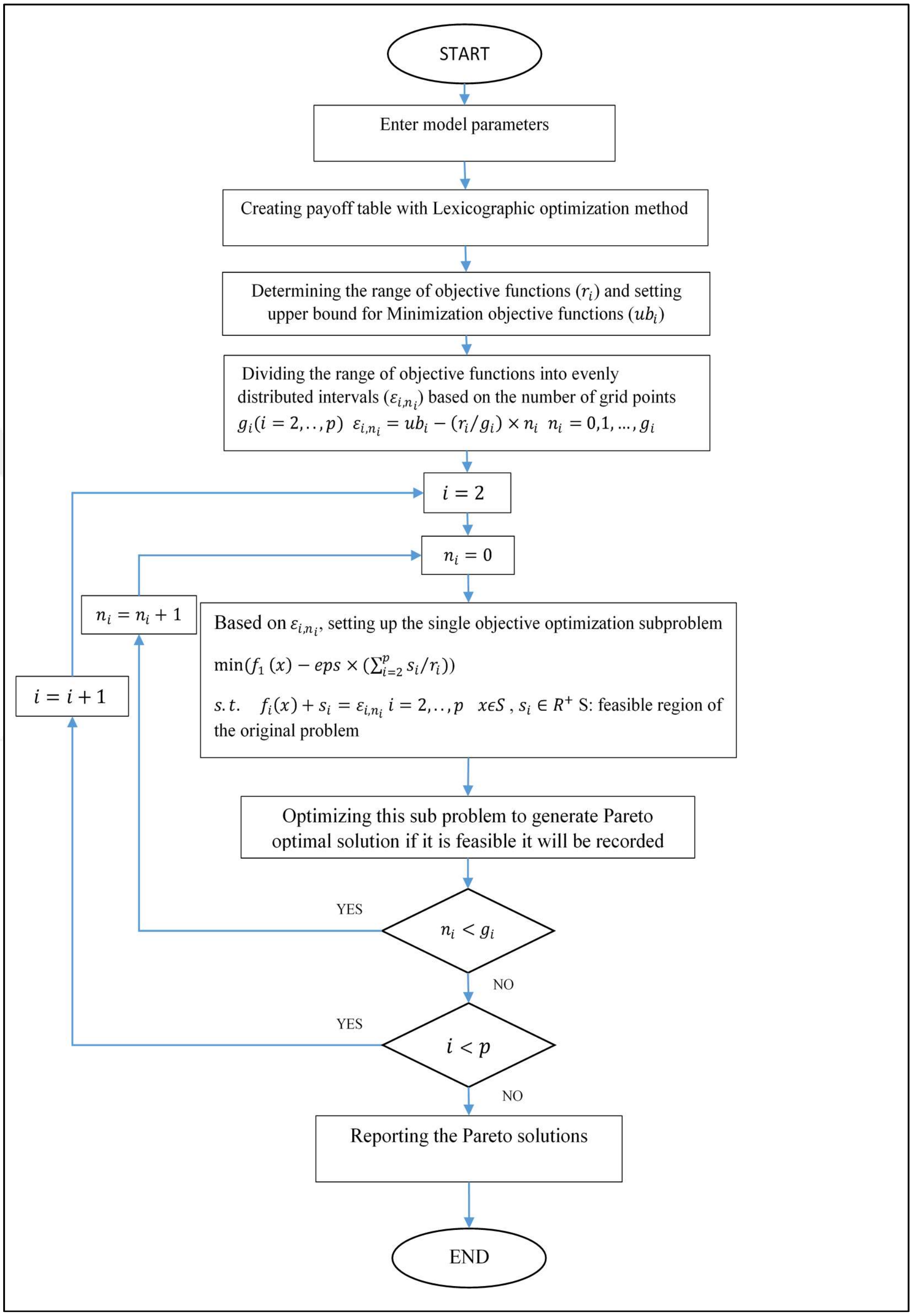

Figure 4.3. Augmented ε-constraint method Algorithm

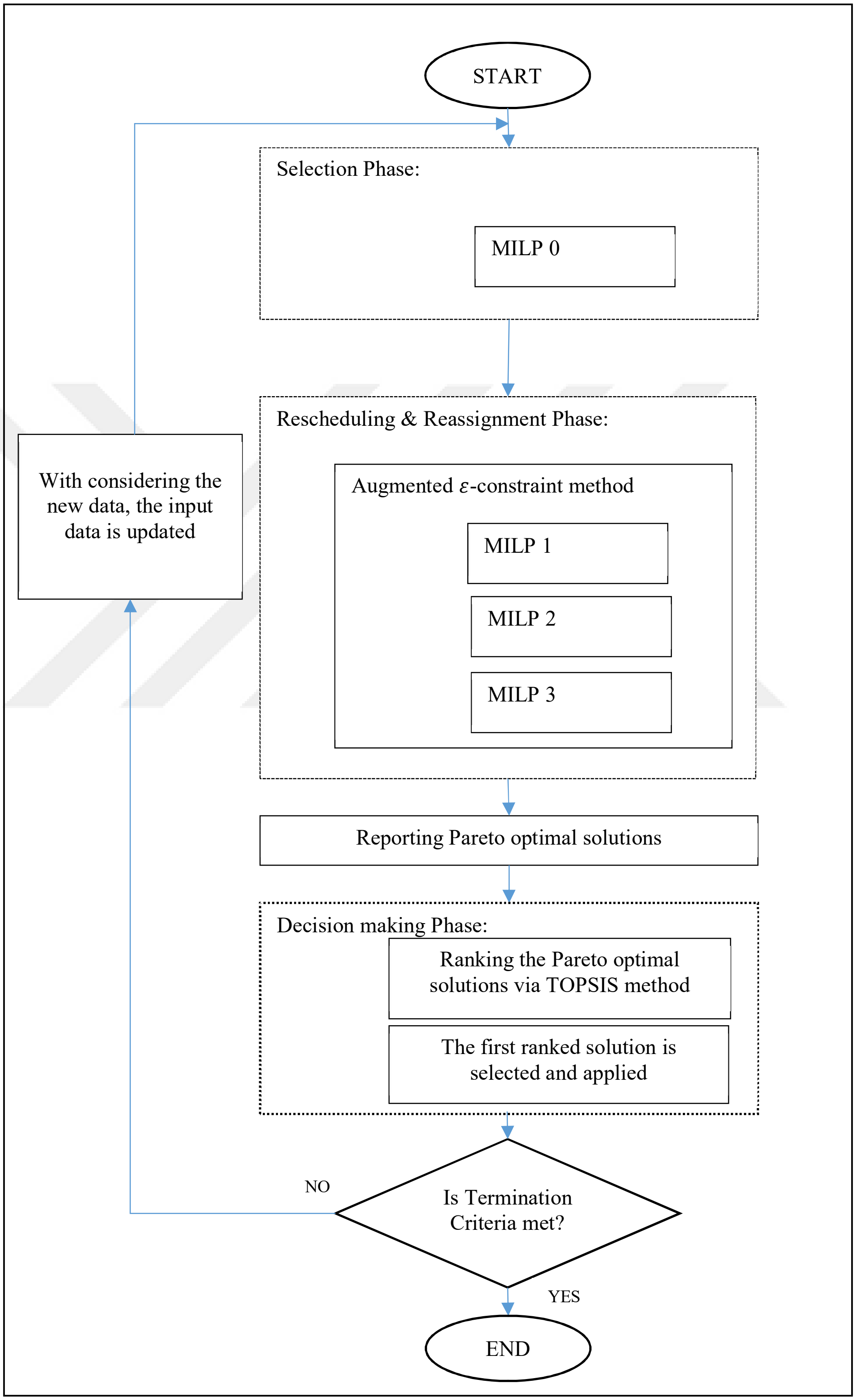

Figure 4.4. The overall scheme of the proposed Dynamic Solution Method

4.2. Heuristic Method

A greedy construction heuristic algorithm is proposed. This algorithm is applied to a problem instance as follows:

A list of gaps that is idle time interval is determined for each team. The assignment procedure starts from the first task in the set of $Unassigned$ tasks. Each task is scheduled at the earliest possible time by considering the ergonomic and technological constraints. If possible gaps are found, the task is scheduled, relevant gaps are updated immediately, and the task is placed in the Assigned tasks set. Nevertheless, if the feasible gaps aren't found, the task remains in the $Unassigned$ tasks set.

The main flow of the algorithm is given below:

1- In the first step, available tasks are sorted in descending order based on their priority levels determined previously.

2- The tasks are taken from the sorted list one by one, and a schedule is constructed incrementally.

3- All current intervals are searched to see if there is a sufficiently long interval for processing task j. Then, a set from all feasible intervals is formed.

4- Depending on the number of required teams for a task, the feasible combination of intervals is chosen and put in a $suitTj$ set. (All possible groups are collected into a $suitTj$ set.)

5- Among the feasible combinations in set $suitTj$, the one that gives the earliest start time for the task, is selected.

6- If there is no feasible interval or combination of intervals, the algorithm continues and takes the next task in the list.

7- The Algorithm continues until all possible tasks are assigned or the time is over.

The notations given below are used in the algorithm:

M total number of teams

N total number of tasks

TS planning period (typically an 8-hour shift)

Indexes

i, i' Teams $i, i' \in M$

j, j' Tasks $j, j' \in N$

$schedule$ is a $M.N$ matrix that is used to show the schedule

NRT_j refers the number of required teams to process task j.

st_j starting time of task j.

d_j is a duration of task j

In order to explain this algorithm in detail, let $gapT_i$ and $suitT_j$ be the sets of gaps for team i and suitable overlapping groups of gaps for task j, respectively.

The algorithm can be described in detail as follows:

Step1: Initialize two main sets, $Assigned\ tasks\ set = \{\}$ and $Unassigned\ tasks = \{1,2,\dots,N\}$, N is the total number of tasks.

Step 2: for each team i, initialize $gapT_i$ to $[0, TS]$,where TS is an available shift time and $suitT_j = \{\}$.

Step 3: for $p = 1$ to the total number of tasks until the termination criteria (all tasks are assigned or reaching maximum available time) are fulfilled do the following:

Step 3-1: consider the task j at position k in $Unassigned\ tasks\ set$

Step 3-2: determine intervals $[l, u]$ so that task j can be assigned to time slot.

Feasible intervals for task j have to be fulfilled by the time and the ergonomic criteria:

1. $u - l \geq d_j$, d_j is a duration of task j

2. If task j is heavy then tasks at position $schedule(i, u+1)$ and position $schedule(i, l-1)$ should be light tasks. In other words, if a task j is heavy, the tasks assigned before and after that task should be light. But if a task is light, it doesn't matter if the tasks before and after that task are heavy or light.

Step 3-3: Feasible Interval gaps for each team are arranged based on the starting time (lower bound) from earliest to the latest. (It is obvious that there is no overlap between these intervals).

Step 3-4: based on the NRT_j for a task, the feasible combination of intervals are determined and put in a $suitT_j$ set. In other words, $suitT_j$ is a set of possible groups of gaps that task j can be assigned. If $NRTj$ is 1, the $suitT_j$ consists of only feasible gaps, but when the $NRTj$ is more than 1, $suitT_j$ consists of feasible groups of gaps. It is noteworthy that the number of gaps in each group equals to the number of required team for this task. All teams in a group have a common time interval that is at least as long as the duration of the task.

Therefore, step 3-4 can be stated as follows: determine intervals $[l, u] = [l_{ik}, u_{ik}] \cap [l_{i'}, u_{i'}] \ldots \cap [l_{i''}, u_{i''}]$ so that $u - l \geq d_j$

Step 3-5: Among the intervals determined in step 3-4, select interval or a group of intervals with minimum$[0, TS]$. If there is more than one option with minimum, assign the task to the team or teams with smallest cumulative ergonomic score.

Step 3-6: Remove interval $[st_j, st_j + d_j]$ from related gaps.

In order to clarify the step 3-4 of the proposed algorithm, assume that $NRT_j = 1$, in this case, for each team, the feasible interval with the earliest starting time (lower bound) is selected and moved to $suitT_j$set. But if $1 < RNB_j \leq M$, then all possible combinations of feasible gaps are selected.

A compound is feasible if the overlap of all considered intervals is equal to or greater than the processing time of task j, in other words,

$$\min \underbrace{\left(u_{i,k}, u_{i'k'},, \ldots, u_{i''k''}\right)}_{the\ number\ is\ equal\ to RNB_j} - \max\left(l_{i,k}, l_{i'k'}, \ldots, l_{i''k''}\right) \geq d_j$$

If a compound is feasible, then it is placed in $(suitT_j)$ set of task j.

An acceleration mechanism may be useful in some cases.

If for an interval of a team a feasible combination is found then the search on next intervals of this team will be stopped and the search continues from the intervals of the next team. In other words, since we are trying to find the earliest starting time for each task, we will stop looking at the next intervals of the same team as soon as we find the first possible interval. Therefore useless searching will not be done.

If in the searching the upper bound of the considered interval is smaller than the lower bound of the interval, then searching in intervals of this team will stop because the intervals in each team are sorted based on their lower bound.

The intervals of each team are sorted based on lower bound (starting times) from earliest to the latest. In a case where a task required more than one team, in searching process between teams to find a suitable combination of intervals, if the upper bound of an interval of team i is smaller than a lower bound of an interval of team i' so the searching is stopped on the intervals of team i' and searching process continues from next teams' intervals.

The flowchart of the main steps of the proposed algorithm is shown in Figure 4.5.

As an illustration, let us consider a small instance with 12 tasks and three teams. Priorities (weights), Processing times, number of required teams, and ergonomic scores are given in Table 4.1.

Table 4.1. Data for the example

Task *j*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Weight(priority)	10000	9000	7000	5500	1000	750	200	80	20	15	10	5
Processing time	10	5	15	5	5	5	10	5	15	7	5	15
Number of required team	2	1	3	1	1	3	2	2	2	1	2	1
Ergonomic Score	25	25	10	25	10	20	25	25	25	15	15	15

The consecutive steps of the algorithm are shown in Table 4.2. k is a counter. The Gantt chart of the obtained solution is depicted in Figure 4.6.

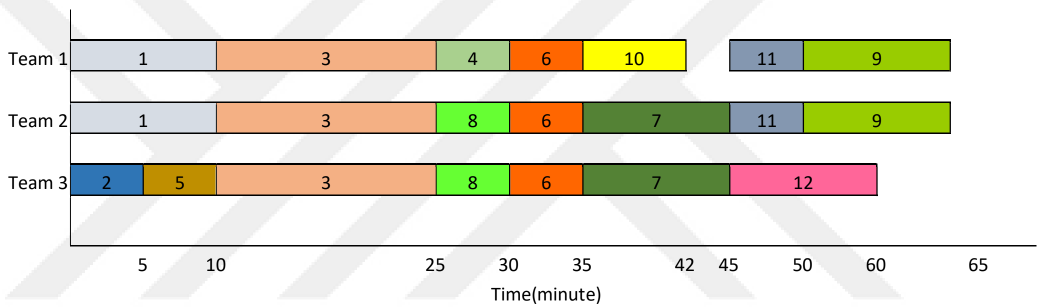

Figure 4.6. The Gantt chart of the obtained solution

After executing the algorithm, the tasks that remain in the unassigned set will be shifted to the next working period, and obviously, their priority will be changed.

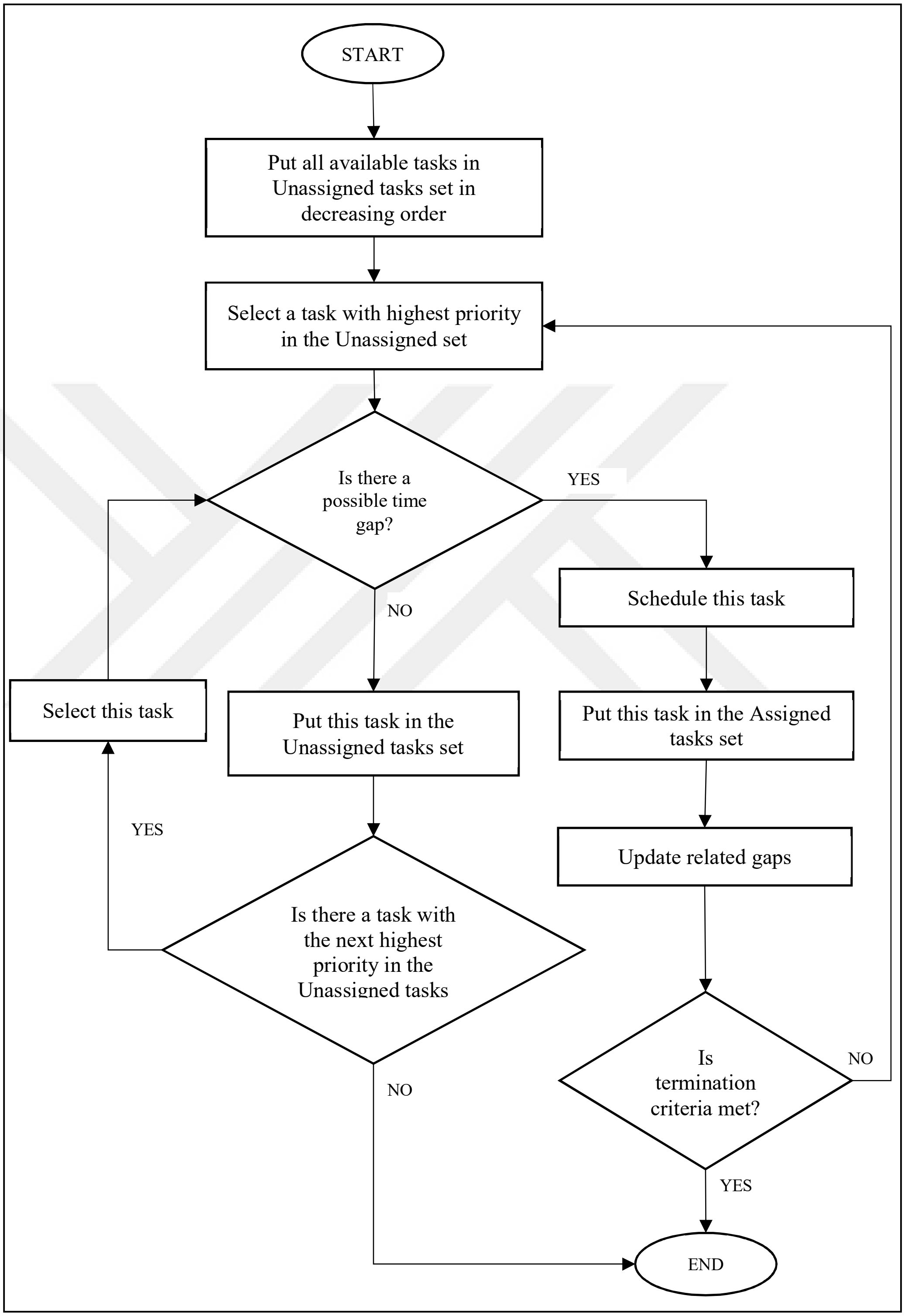

Figure 4.5. Flowchart of the proposed Heuristic algorithm

Table 4.2. Steps of the algorithm

k	Task	$[l,u]$	st	Team 1		Team 2		Team 3	
				Idle Gaps	**CES***	**Idle Gaps**	**CES**	**Idle Gaps**	**CES**
1	1	[0,10]	0	GapT1[10,120]	25	GapT2[10,120]	25	GapT3[0,120]	0
2	2	[0,5]	0	GapT1[10,120]	25	GapT2[10,120]	25	GapT3[5,120]	25
3	3	[10,25]	10	GapT1[25,120]	35	GapT2[25,120]	35	GapT3[5,10]∪[25,120]	35
4	4	[25,30]	25	GapT1[30,120]	60	GapT2[25,120]	35	GapT3[5,10]∪[25,120]	35
5	5	[5,10]	5	GapT1[30,120]	60	GapT2[25,120]	35	GapT3[25,120]	45
6	6	[30,35]	30	GapT1[35,120]	80	GapT2[25,30]∪[35,120]	55	GapT3[25,30]∪[35,120]	65
7	7	[35,45]	35	GapT1[35,120]	80	GapT2[25,30]∪[45,120]	80	GapT3[25,30]∪[45,120]	90
8	8	[25,30]	25	GapT1[35,120]	80	GapT2[45,120]	105	GapT3[45,120]	115
9	10	[35,42]	35	GapT1[42,120]	95	GapT2[45,120]	105	GapT3[45,120]	115
10	11	[45,50]	45	GapT1[42,45]∪[50,120]	110	GapT2[50,120]	120	GapT3[45,120]	115
11	9	[50,65]	50	GapT1[42,45]∪[65,120]	135	GapT2[65,120]	145	GapT3[45,120]	115
12	12	[45,60]	45	GapT1[42,45]∪[65,120]	135	GapT2[65,120]	145	GapT3[60,120]	130

*CES is the abbreviation of Cumulative Ergonomic Score

5. RESULTS AND DISCUSSION

Since the studied problem has not been investigated in the literature before, there is no benchmark or exact or approximate solution method developed for its solution.

In this section, the proposed solution methods detailed previously are analyzed to assess their performances in various situations.

Firstly, the solution models are tested under real circumstances. Real-world data provided by the planning department of an automotive distribution center in Kocaeli are used at this level. In section 5.1, described results are obtained using real data.

Secondly, generated instances within eight different categories are used to analyze the performance of the proposed dynamic method more comprehensively.

The proposed dynamic algorithm and constructive heuristic were coded in MATLAB 2014a. As well as the mixed-integer programming models are coded in GAMS 25.1. These algorithms were implemented in a personal laptop with a Core i5 processor, 2.40 GHz CPU, 4 GB RAM, and Microsoft Windows 10 64-bit operating system.

5.1. Numerical Results From the Real-data Instances

Data was collected in a logistics center. The data sets belong to daily shifts of 15 different days at a cross-dock platform are gathered and tested. In this platform, there are three homogenous teams. The number of tasks is between 40 and 60. The processing time of tasks has a normal distribution with an average of 24 minutes and a standard deviation of 5 minutes. All of the tasks were analyzed using different ergonomic parameters of the OCRA method; they scored and grouped into five main ergonomic categories (this process is described in section 2.3.3.1 in more detail). The tasks that belong to the fifth category are considered heavy tasks; thus, they shall not be performed consecutively.

Real problems are solved using the proposed dynamic algorithm and the constructive heuristic to measure the efficiency of the proposed algorithm. The results are compared with real schedules. Real schedules are schedules that were obtained manually at a cross-dock. It takes approximately 2 hours for an employee to make a daily schedule; thus, in tables for manual method time, 7200 seconds is considered.

OBJ1, OBJ2, and OBJ3 are abbreviations for considered objective functions in the rescheduling and reassignment phase in the dynamic algorithm. OBJ1 symbolizes the minimization of the total weighted completion times. OBJ2 stands for the minimization of the total ergonomic deviation among teams. OBJ3 figures out the last objective function, which minimizes working time deviation among teams.

It should be noted that while using the dynamic algorithm, the weights of the objective functions are considered equal.

The mutual comparison between the manual method, the dynamic method, and the heuristic method are shown in Tables 5.1, Table 5.2, and Table 5.3, respectively.

In Table 5-1, in the comparison between the dynamic algorithm and the manual method, it is seen that an average of 6% improvement is achieved in the total weighted completion time of tasks. 84% improvement in the balanced distribution of the workload and 62.78 % improvement in the fairness of working times among teams are obtained. In two instances, we encountered a situation that an obtained value for OBJ1 with the manual method is better than the dynamic method. This situation is because of selecting different tasks or assignments differently in two methods. Even in these cases, the OBJ2 and OBJ3 are noticeably better in the dynamic method. It is noteworthy that for calculating improvement percent for 15 instances and three objective functions, the below formula is used:

$$\text{Improvement}_{ij} = \left(\frac{\text{Manual method}_{ij} - \text{Dynamic method}_{ij}}{\text{Manual method}_{ij}}\right) \times 100 \;\; \text{for} \;\; i = 1, \dots 15, j = 1, ..3 \qquad (5.1)$$

Also, it can be seen from Table 5.1 that the proposed dynamic method generate daily schedule noticeably in much less time.

The comparison between the dynamic method and the heuristic method is shown in Table 5.2. The dynamic method improved solutions by 0.71%, 83.45%, and 58.07% regarding OBJ1, OBJ2, and OBJ3, respectively. The following formula is used for estimating improvement percent:

$$\text{Improvement}_{ij} = \left(\frac{\text{Heuristic method}_{ij} - \text{Dynamic method}_{ij}}{\text{Heuristic method}_{ij}}\right) \times 100 \ \text{ for } \ i = 1, \dots 15, j = 1, ..3 \tag{5.2}$$

As evident in Table 5.2, in some cases (shown in gray color) the heuristic method generated better solutions. However, these cases are rare, and they are due to selecting different tasks and different allocation of tasks in two methods. It should also be noticed that the heuristic produces feasible solutions in much less time than the dynamic method.

Finally, in Table 5.3, the comparison between the heuristic method and the manual method is made. The heuristic method improved OBJ1 by an average of 5.53%. On another side, it improved the average values of OBJ 2 and OBJ3 by 7.51 % and 8.87%, respectively. Note that the heuristic method decreases the planning time drastically, from hours to seconds, compared to the manual method. The following formula is used to calculate improvement percent.

$$\text{Improvement}_{ij} = \left(\frac{\text{Manual method}_{ij} - \text{Heuristic method}_{ij}}{\text{Manual method}_{ij}}\right) \times 100 \ \text{ for } \ i = 1, \dots 15, j = 1, ..3 \tag{5.3}$$

The heuristic method can obtain a feasible solution in less than one second in all instances. General comparison among the dynamic method, the heuristic method, and the manual one considering OBJ1, OBJ2, and OBJ3 is presented in Figure 5.1, Figure 5.2, and Figure 5.3 respectively. Real data from 15 different days at a cross-dock platform are tested.

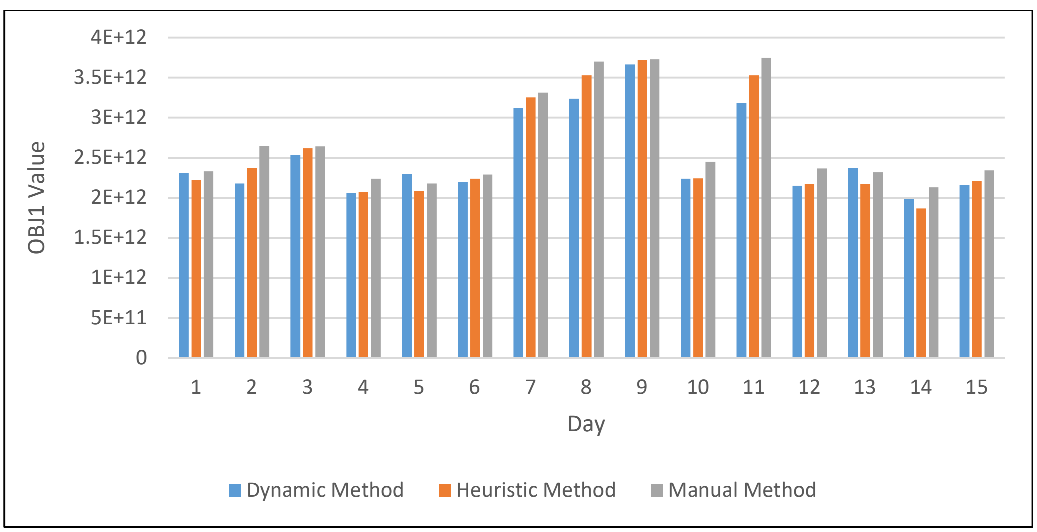

Figure 5.1. Comparison among Dynamic Method, Heuristic Method and Manual Method regarding OBJ1

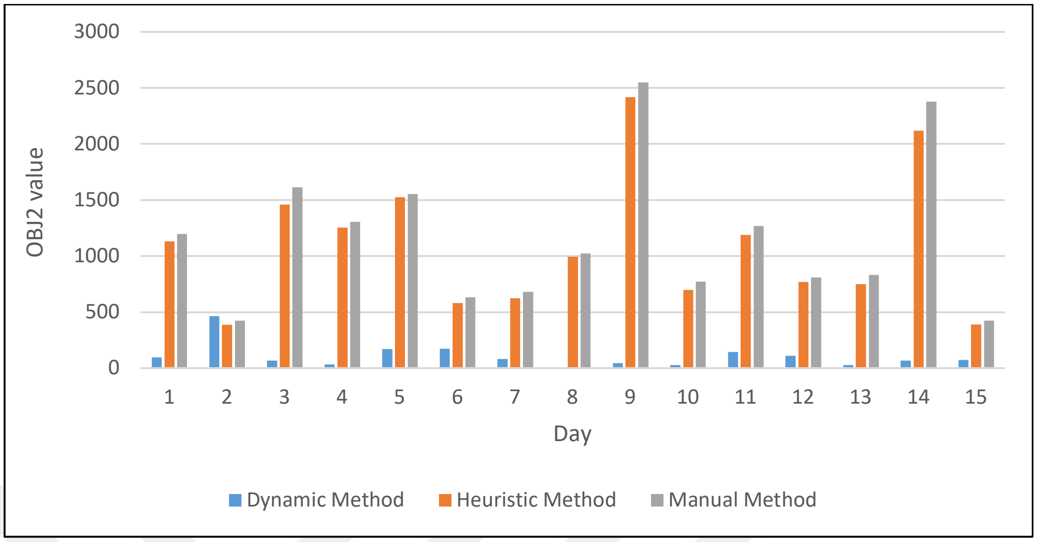

Figure 5.2. Comparison among Dynamic Method, Heuristic Method and Manual Method regarding OBJ2

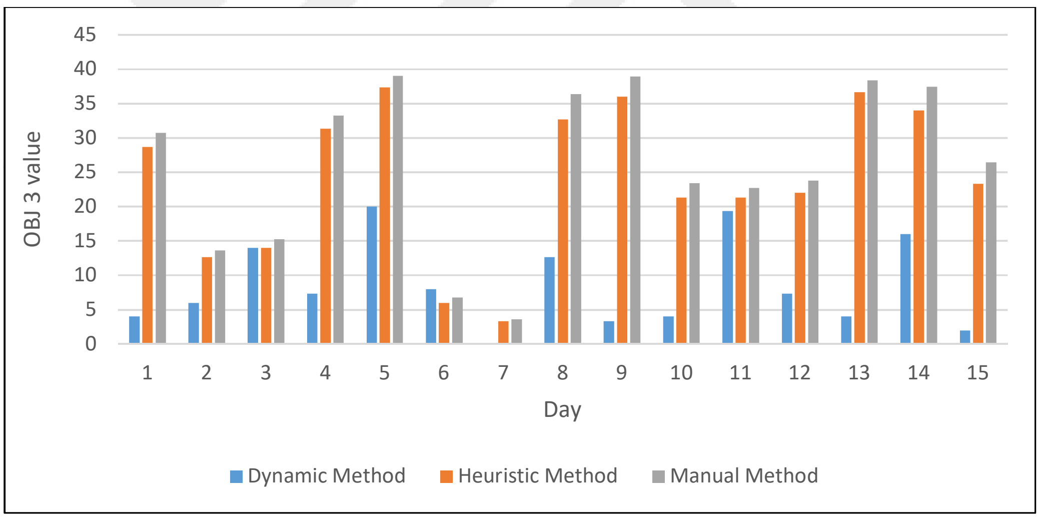

Figure 5.3. Comparison among Dynamic Method, Heuristic Method and Manual Method regarding OBJ3

From the above figures, it is evident that the dynamic method's performance is better than other methods generally.

Table 5.1. Comparison of Dynamic Method and Manual Method

	Dynamic Method (1/3,1/3,1/3)						Manual Method						% Improvement		
Instance	**OBJ 1**	**OBJ 2**	**OBJ 3**	**CPU Time (Sec)**	**Number of completed tasks**	**Total Time**	**OBJ 1**	**OBJ 2**	**OBJ 3**	**Time (Sec)**	**Number of completed tasks**	**Total Time**	**OBJ 1**	**OBJ 2**	**OBJ 3**
Day 1	2.31E+12	98.00	4.00	297.94	40	1422	2.33E+12	1196.13	30.73	7200	40	1411	0.96	91.81	86.98
Day 2	2.18E+12	464.00	6.00	681.51	41	1371	2.64E+12	466.47	13.62	7200	41	1371	17.53	0.53	55.95
Day 3	2.54E+12	67.33	14.00	183.12	40	1413	2.64E+12	1614.71	15.29	7200	39	1385	3.93	95.83	8.42
Day 4	2.06E+12	32.67	7.33	873.79	41	1420	2.24E+12	1307.39	33.24	7200	41	1420	7.95	97.50	77.94
Day 5	2.3E+12	169.33	20.00	356.68	39	1392	2.18E+12	1552.05	38.99	7200	39	1384	-5.50	89.09	48.71
Day 6	2.2E+12	173.33	8.00	329.94	41	1431	2.29E+12	633.47	8.06	7200	41	1431	4.13	72.64	0.75
Day 7	3.12E+12	80.67	0.00	6062.44	41	1419	3.31E+12	681.03	3.64	7200	41	1419	5.73	88.16	100.00
Day 8	3.24E+12	5.33	12.67	290.30	37	1394	3.7E+12	1022.43	36.38	7200	38	1413	12.48	99.48	65.19
Day 9	3.67E+12	44.00	3.33	415.15	36	1412	3.73E+12	2548.59	38.94	7200	35	1408	1.64	98.27	91.44
Day 10	2.24E+12	28.67	4.00	738.67	42	1395	2.45E+12	773.20	23.42	7200	42	1417	8.63	96.29	82.92
Day 11	3.18E+12	145.33	19.33	527.89	39	1415	3.75E+12	1270.35	22.74	7200	39	1415	15.07	88.56	14.98
Day 12	2.15E+12	110.00	7.33	311.20	39	1417	2.37E+12	809.67	23.77	7200	37	1365	8.98	86.41	69.15
Day 13	2.38E+12	28.00	4.00	172.42	42	1425	2.32E+12	831.76	38.36	7200	42	1425	-2.50	96.63	89.57
Day 14	1.99E+12	66.67	16.00	4389.31	43	1413	2.13E+12	2378.17	37.46	7200	43	1413	6.73	97.20	57.28
Day 15	2.16E+12	72.00	2.00	330.36	38	1401	2.34E+12	425.69	26.43	7200	38	1401	7.74	83.09	92.43
												Average	**6.23**	**85.43**	**62.78**

Tabel 5.2 Comparison of Dynamic Method and Heuristic Method

	Dynamic Method (1/3,1/3,1/3)						Heuristic Method						% Improvement		
Instance	**OBJ 1**	**OBJ 2**	**OBJ 3**	**CPU Time (Sec)**	**Number of completed tasks**	**Total Time**	**OBJ 1**	**OBJ 2**	**OBJ 3**	**CPU Time (Sec)**	**Number of completed tasks**	**Total Time**	**OBJ1**	**OBJ2**	**OBJ3**
Day 1	2.31E+12	98.00	4.00	297.94	40	1422	2.23E+12	1133.33	28.67	0.94	40	1399	-3.71	91.35	86.05
Day 2	2.18E+12	464.00	6.00	681.51	41	1371	2.37E+12	386.00	12.67	0.71	42	1393	8.13	-20.21	52.63
Day 3	2.54E+12	67.33	14.00	183.12	40	1413	2.62E+12	1460.67	14.00	0.82	40	1386	3.09	95.39	0.00
Day 4	2.06E+12	32.67	7.33	873.79	41	1420	2.07E+12	1256.67	31.33	0.44	39	1376	0.44	97.40	76.60
Day 5	2.3E+12	169.33	20.00	356.68	39	1392	2.09E+12	1523.33	37.33	0.47	38	1397	-10.12	88.88	46.43
Day 6	2.2E+12	173.33	8.00	329.94	41	1431	2.24E+12	581.33	6.00	0.76	39	1368	1.94	70.18	-33.33
Day 7	3.12E+12	80.67	0.00	6062.44	41	1419	3.25E+12	622.67	3.33	0.65	38	1369	4.02	87.04	100.00
Day 8	3.24E+12	5.33	12.67	290.30	37	1394	3.53E+12	995.33	32.67	0.68	36	1340	8.31	99.46	61.22
Day 9	3.67E+12	44.00	3.33	415.15	36	1412	3.72E+12	2417.33	36.00	0.68	36	1395	1.42	98.18	90.74
Day 10	2.24E+12	28.67	4.00	738.67	42	1395	2.24E+12	698.67	21.33	0.79	41	1376	0.22	95.90	81.25
Day 11	3.18E+12	145.33	19.33	527.89	39	1415	3.53E+12	1188.67	21.33	0.71	37	1348	9.88	87.77	9.38
Day 12	2.15E+12	110.00	7.33	311.20	39	1417	2.18E+12	770.00	22.00	0.68	38	1380	1.01	85.71	66.67
Day 13	2.38E+12	28.00	4.00	172.42	42	1425	2.17E+12	750.00	36.67	0.67	43	1393	-9.42	96.27	89.09
Day 14	1.99E+12	66.67	16.00	4389.31	43	1413	1.87E+12	2118.00	34.00	0.67	44	1350	-6.53	96.85	52.94
Day 15	2.16E+12	72.00	2.00	330.36	38	1401	2.21E+12	389.33	23.33	0.62	37	1364	1.99	81.51	91.43
												Average	**0.71**	**83.45**	**58.07**

Table 5.3 Comparison of Manual Method and Heuristic Method

	Manual Method						**Heuristic Method**						**% Improvement**		
Instance	**OBJ 1**	**OBJ 2**	**OBJ 3**	**Time (Sec)**	**Number of completed tasks**	**Total Time**	**OBJ 1**	**OBJ 2**	**OBJ 3**	**CPU Time**	**Number of completed tasks**	**Total Time**	**OBJ 1**	**OBJ 2**	**OBJ 3**
1	2.33E+12	1196.13	30.73	7200	40	1411	2.23E+12	1133.33	28.67	0.94	40	1399	4.50	5.25	6.70
2	2.64E+12	466.47	13.62	7200	41	1371	2.37E+12	386.00	12.67	0.71	42	1393	10.24	17.25	7.01
3	2.64E+12	1614.71	15.29	7200	39	1385	2.62E+12	1460.67	14.00	0.82	40	1386	0.87	9.54	8.42
4	2.24E+12	1307.39	33.24	7200	41	1420	2.07E+12	1256.67	31.33	0.44	39	1376	7.54	3.88	5.74
5	2.18E+12	1552.05	38.99	7200	39	1384	2.09E+12	1523.33	37.33	0.47	38	1397	4.20	1.85	4.26
6	2.29E+12	633.47	8.06	7200	41	1431	2.24E+12	581.33	6.00	0.76	39	1368	2.23	8.23	25.56
7	3.31E+12	681.03	3.64	7200	41	1419	3.25E+12	622.67	3.33	0.65	38	1369	1.78	8.57	8.33
8	3.7E+12	1022.43	36.38	7200	38	1413	3.53E+12	995.33	32.67	0.68	36	1340	4.55	2.65	10.22
9	3.73E+12	2548.59	38.94	7200	35	1408	3.72E+12	2417.33	36.00	0.68	36	1395	0.22	5.15	7.55
10	2.45E+12	773.20	23.42	7200	42	1417	2.24E+12	698.67	21.33	0.79	41	1376	8.43	9.64	8.92
11	3.75E+12	1270.35	22.74	7200	39	1415	3.53E+12	1188.67	21.33	0.71	37	1348	5.76	6.43	6.19
12	2.37E+12	809.67	23.77	7200	37	1365	2.18E+12	770.00	22.00	0.68	38	1380	8.05	4.90	7.46
13	2.32E+12	831.76	38.36	7200	42	1425	2.17E+12	750.00	36.67	0.67	43	1393	6.33	9.83	4.41
14	2.13E+12	2378.17	37.46	7200	43	1413	1.87E+12	2118.00	34.00	0.67	44	1350	12.45	10.94	9.23
15	2.34E+12	425.69	26.43	7200	38	1401	2.21E+12	389.33	23.33	0.62	37	1364	5.86	8.54	11.71
												Average	**5.53**	**7.51**	**8.78**

5.2. Numerical Results from Generated Instances

The previous section shows that the daily schedules for real size problems can be solved properly and in a reasonable time. In the current section, we aim to assess and analyze the proposed method regarding some of the input parameters that can affect its performance.

The length of an interval is set to 2 hours, which means we consider four intervals for a daily shift.

First of all, we focus on the average processing time (or the number of available tasks). For this reason, several numerical examples are generated. These instances are classified into three sets, small, medium, and large size instances regarding the average processing time of a task.

The three different problem instances and their characteristics are presented in Table 5.4. They vary from one another with respect to the average processing time of a task and the number of available tasks per day. It is assumed that processing times have a normal distribution function, but ergonomic score and number of tasks have uniform distribution functions.

Furthermore, for simulating the real circumstance, every 2-hour new generated tasks enter the system dynamically. The number of these dynamic tasks is produced based on a uniform distribution between zero to three.

Table 5.4. Charactristic of Parameters (Their distribution functions, average, and standard deviation)

Parameter	Small instance	Medium Instance	Large instance
Processing time	N(20,5)	N(24, 5)	N(28, 5)
Number of Tasks	U(55-65)	U(55-65)	U(55-65)
Ergonomic Score	U(5,25)	U(5,25)	U(5,25)
Number of dynamic tasks	U(0-3)	U(0-3)	U(0-3)

The TOPSIS method is one of the main parts of the proposed dynamic method. Since defining objective functions' weights is an essential part of the TOPSIS method, so for covering a wide range of weights sufficiently, Steuer's method [65] is used to generate different sets of weights.

The Steuer's Contracting Cone Method is iterative multiple objective decision-making that is usually used to generate Pareto solutions. In Steuer's method (Steuer's Contracting Cone Method), if there are p objective functions, this method generates 2p+1 trial solutions each time. The initial set of weights are as follows:

Table 5.5. Steuer's Method

λ_1	= the first extreme vector	$= (1,0,0, \dots ,0)$
λ_2	= the second extreme vector	$= (0,1,0, \dots ,0)$
$\vdots$	$\vdots$	$\vdots$
λ_p	= the P_{th} extreme vector	$= (0,0,0, \dots ,1)$
λ_{p+1}	$= ((\lambda_2 + \lambda_3 + \cdots + \lambda_p + \lambda_{p+1})/p)$	$= (1/p^2, r, r, \dots , r)$
λ_{p+2}	$= ((\lambda_1 + \lambda_3 + \lambda_4 + \cdots + \lambda_p + \lambda_{p+1})/p)$	$= (r, 1/p^2, r, r, \dots , r)$
λ_{p+3}	$= ((\lambda_1 + \lambda_2 + \lambda_4 + \cdots + \lambda_p + \lambda_{p+1})/p$	$= (r, r, 1/p^2, r, \dots , r)$
$\vdots$	$\vdots$	$\vdots$
λ_{2p+1}	$= (1/p(\lambda_1 + \lambda_2 + \cdots + \lambda_p))$	$= (1/p, 1/p, \dots ,1/p)$
		Where $r = (p+1)/p^2$

According to the table, in our problem that there are three objective functions, then the set of weights are as bellow:

$\lambda_1 = (1,0,0)$

$\lambda_2 = (0,1,0)$

$\lambda_3 = (0,0,1)$

$\lambda_4 = (1/9,4/9,4/9)$

$\lambda_5 = (4/9 , 1/9, 4/9)$

$\lambda_6 = (4/9,4/9, 1/9)$

$\lambda_7 = (1/3,1/3,1/3)$

Furthermore, in order to see the model's ability to solve problems of different properties, instances have been produced in eight different categories. This categorization is based on the number of initial emergency tasks, the number of heavy tasks, and the number of tasks requiring only one team. It is important to control how the method behaves when these parameters change. This categorization is depicted in Figure 5.4.

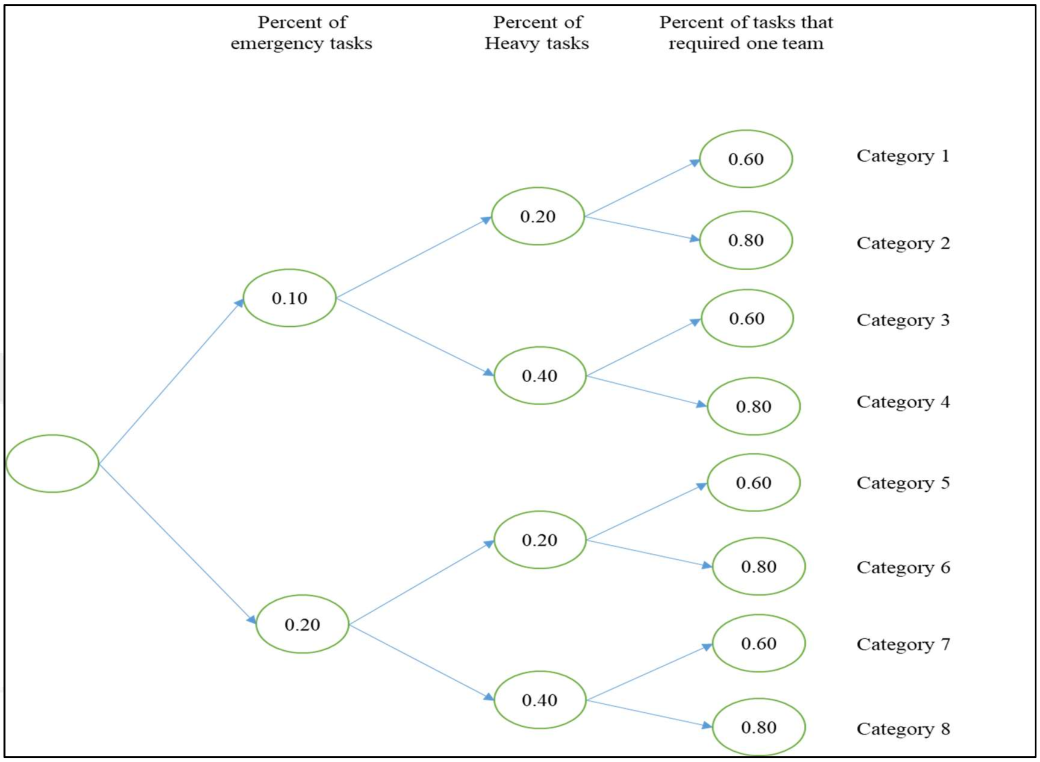

Figure 5.4. Categories of generated instances

Summarily, for each small, medium, and large instance, eight categories with seven different sets of objective function weights are produced. The results are presented in Table5.6, Table 5.7 and Table 5.8.

The results show that the proposed dynamic algorithm works properly in different situations.

Also, it can be seen from the results that the running time (CPU time) for category 7 in all instances is the highest one. This increase can be due to the more complexity that the higher percentage of heavy tasks in parallel with a lower percent of 1-team tasks impose on the model.

With increasing processing time average, the number of selected and completed tasks in a daily shift decreases; therefore, the dynamic algorithm's running time decreases noticeably.

Table 5.6. Results for short instances (Processing time average = 20 minutes)

Category	λ1					λ2				
	OBJ 1	OBJ 2	OBJ 3	CPU Time*	NCT*	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.02E+12	68.7	8.0	693	47	2.048E+12	7.3	22.0	577	47
2	1.967E+12	558.0	27.3	5317	50	2.038E+12	186.0	5.3	4866	50
3	2.755E+12	270.7	12.7	3333	45	2.858E+12	88.7	5.3	1608	45
4	2.899E+12	550.7	2.0	9375	48	2.954E+12	10.7	2.0	9791	48
5	2.012E+12	384.0	1.3	888	46	2.096E+12	10.0	25.3	1710	46
6	1.905E+12	320.7	21.3	1122	51	1.941E+12	59.3	17.3	811	51
7	2.761E+12	454.7	10.7	20990	48	2.777E+12	13.3	7.3	11115	48
8	2.705E+12	265.3	15.3	4952	49	2.754E+12	93.3	14.7	2337	49

Category	λ3					λ4				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.026E+12	108.7	2.0	1373	47	2.026E+12	62.7	4.0	1359	47
2	1.995E+12	356.0	7.3	5281	50	2.002E+12	30.0	7.3	17811	50
3	2.776E+12	166.7	3.3	3456	45	2.815E+12	110.7	4.7	1500	45
4	2.927E+12	348.7	0.0	9437	48	2.905E+12	52.7	2.0	8787	48
5	2.036E+12	70.0	1.3	1534	46	2.114E+12	10.0	8.7	1263	46
6	1.924E+12	645.3	1.3	842	51	1.924E+12	195.3	3.3	821	51
7	2.772E+12	83.3	1.3	14665	48	2.774E+12	11.3	2.7	15839	48
8	2.707E+12	305.3	8.7	3267	49	2.695E+12	89.3	8.7	2679	49

Category	λ5					λ6				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.026E+12	51.3	4.0	1377	47	2.026E+12	12.7	16.0	1304	47
2	2.024E+12	30.0	7.3	7748	50	2.006E+12	20.0	15.3	17434	50
3	2.815E+12	110.7	4.7	1516	45	2.851E+12	84.7	5.3	1668	45
4	2.921E+12	348.7	0.0	9533	48	2.905E+12	52.7	2.0	10584	48
5	2.056E+12	56.0	1.3	837	46	2.063E+12	50.0	5.3	1114	46
6	1.924E+12	195.3	3.3	822	51	1.941E+12	59.3	17.3	793	51
7	2.773E+12	52.7	1.3	13410	48	2.777E+12	29.3	2.7	10401	48
8	2.712E+12	71.3	8.7	3349	49	2.755E+12	87.3	8.7	2319	49

Category	λ7				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.026E+12	62.7	4.0	1369	47
2	2.002E+12	30.0	7.3	17555	50
3	2.815E+12	110.7	4.7	1440	45
4	2.905E+12	52.7	2.0	9065	48
5	2.082E+12	88.0	1.3	1385	46
6	1.906E+12	99.3	7.3	2876	51
7	2.774E+12	11.3	2.7	12378	48
8	2.695E+12	89.3	8.7	2673	49

*NCT is the abbreviation of Number of Completed Tasks
*CPU Time unit is second

Table 5.7. Results for Medium instances (Processing time average=24 minutes)

Category	λ1					λ2				
	OBJ 1	OBJ 2	OBJ 3	CPU* Time	NCT*	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.05E+12	264.7	10.7	649	39	2.08E+12	17.3	15.3	573	39
2	1.84E+12	166.0	16.7	775	43	1.9E+12	67.3	14.7	826	43
3	2.47E+12	409.3	6.0	1556	42	2.5E+12	9.3	4.0	1455	42
4	2.62E+12	655.3	11.3	2671	43	2.64E+12	703.3	14.7	2860	43
5	2.15E+12	220.7	2.0	1608	38	2.25E+12	70.7	24.0	1575	38
6	1.89E+12	1070.7	27.3	1233	40	1.95E+12	806.7	19.3	1201	40
7	2.8E+12	110.0	9.3	4732	39	2.83E+12	36.0	16.7	4674	39
8	2.42E+12	1120.7	19.3	3431	45	2.48E+12	934.7	29.3	3315	45

Category	λ3					λ4				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.07E+12	195.3	1.3	630	39	2.08E+12	138.7	3.3	671	39
2	1.83E+12	441.3	1.3	431	43	1.83E+12	299.3	1.3	804	43
3	2.47E+12	724.7	0.0	1551	42	2.48E+12	21.3	2.0	1454	42
4	2.68E+12	381.3	12.7	2740	43	2.68E+12	695.3	12.7	2852	43
5	2.14E+12	794.7	6.7	1873	38	2.22E+12	240.7	10.7	1951	38
6	1.89E+12	1394.7	15.3	1249	40	1.94E+12	806.7	19.3	1158	40
7	2.82E+12	652.0	3.3	4721	39	2.86E+12	296.0	4.7	4760	39
8	2.49E+12	1370.7	7.3	4408	45	2.45E+12	1310.7	1.3	3300	45

Category	λ5					λ6				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.08E+12	443.3	1.3	669	39	2.08E+12	60.7	5.3	626	39
2	1.83E+12	372.0	20.7	732	43	1.83E+12	299.3	1.3	807	43
3	2.48E+12	574.7	0.0	1467	42	2.48E+12	21.3	2.0	1450	42
4	2.68E+12	695.3	12.7	2860	43	2.68E+12	735.3	8.7	2868	43
5	2.18E+12	652.7	7.3	1954	38	2.25E+12	70.7	24.0	1570	38
6	1.94E+12	806.7	19.3	1158	40	1.94E+12	806.7	19.3	1170	40
7	2.86E+12	296.0	4.7	4734	39	2.86E+12	296.0	4.7	4721	39
8	2.5E+12	1370.7	7.3	4422	45	2.47E+12	640.7	9.3	3352	45

Category	λ7				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.08E+12	138.7	3.3	688	39
2	1.83E+12	299.3	1.3	810	43
3	2.48E+12	21.3	2.0	1458	42
4	2.68E+12	695.3	12.7	2855	43
5	2.22E+12	240.7	10.7	1953	38
6	1.94E+12	806.7	19.3	1164	40
7	2.86E+12	296.0	4.7	4713	39
8	2.45E+12	1310.7	1.3	3331	45

*NCT is the abbreviation of Number of Completed Tasks
*CPU Time unit is second

Table 5.8. Results for Large instances (Processing time average=28 minutes)

Category	λ1					λ2				
	OBJ 1	OBJ 2	OBJ 3	CPU Time*	NCT*	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.75E+12	1338.0	9.3	607	39	2.87E+12	202.0	16.7	550	39
2	2.77E+12	260.7	8.7	555	39	2.8E+12	152.7	3.3	308	39
3	3.77E+12	1129.3	16.0	485	34	3.8E+12	1237.3	15.3	536	35
4	3.71E+12	646.7	12.0	215	36	3.73E+12	1204.7	9.3	312	35
5	2.88E+12	1021.3	4.7	422	37	2.94E+12	181.3	4.7	378	37
6	2.86E+12	735.3	3.3	736	42	2.89E+12	68.7	4.0	624	41
7	4E+12	1176.0	12.0	677	33	4.16E+12	534.0	40.7	717	32
8	3.59E+12	1397.3	20.0	700	38	3.68E+12	199.3	13.3	445	39

Category	λ3					λ4				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.82E+12	94.0	1.3	539	39	2.82E+12	118.0	11.3	533	39
2	2.78E+12	1787.3	3.3	513	39	2.84E+12	263.3	4.7	495	39
3	3.8E+12	979.3	18.0	442	34	3.83E+12	418.7	4.0	425	35
4	3.77E+12	1230.0	8.0	312		3.82E+12	1230.0	12.0	352	36
5	2.93E+12	1049.3	3.3	386	37	2.93E+12	137.3	4.7	417	37
6	2.85E+12	910.0	6.0	604	41	2.96E+12	262.7	0.0	681	41
7	4.03E+12	692.0	4.7	751	32	4.05E+12	1032.0	1.3	700	32
8	3.6E+12	729.3	11.3	676	39	3.6E+12	697.3	13.3	576	39

Category	λ5					λ6				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.84E+12	526.0	1.3	520	39	2.86E+12	188.0	11.3	523	39
2	2.79E+12	464.7	2.7	442	39	2.8E+12	152.7	3.3	301	39
3	3.84E+12	502.7	1.3	549	35	3.83E+12	1481.3	6.0	435	35
4	3.77E+12	1230.0	8.0	317	36	3.78E+12	586.0	2.7	242	36
5	2.93E+12	40.7	3.3	360	37	2.94E+12	181.3	4.7	381	37
6	2.91E+12	736.0	10.0	583	41	2.94E+12	120.7	2.0	684	41
7	4.04E+12	1020.0	4.7	770	32	4.05E+12	880.0	28.7	604	32
8	3.6E+12	697.3	13.3	577	39	3.68E+12	199.3	13.3	451	39

Category	λ7				
	OBJ 1	OBJ 2	OBJ 3	CPU Time	NCT
1	2.84E+12	118.0	11.3	508	39
2	2.84E+12	263.3	4.7	521	39
3	3.83E+12	418.7	4.0	439	35
4	3.82E+12	1230.0	12.0	332	36
5	2.93E+12	137.3	4.7	425	37
6	2.96E+12	262.7	0.0	689	41
7	4.05E+12	1032.0	1.3	681	32
8	3.6E+12	697.3	13.3	611	39

*NCT is the abbreviation of Number of Completed Tasks
*CPU Time unit is second

5.3. Design of Experiment

Two full factorial design of experiments are designed to analyze the impact of the different parameters on the execution time. The design of experiments are conducted in Minitab 19 software.

Pareto charts of the effects are used to compare the relative magnitude and the statistical significance of both main and interaction effects.

In the first DOE, the percent of emergency tasks (A), the percent of heavy tasks (B), the percent of tasks that require only one team (C), and the size of the instances (D) are considered independent factors. The dependent factor is execution time. The result is shown in Figure 5.5 and Figure 5.6.

Minitab plots the effects in the decreasing order of their absolute values. The reference line on the chart indicates (red line) which effects are significant.

It can be seen that other effects and interaction effects are statistically significant except for effects (A) and (C).

In addition, the largest effect is (AC) because it extends the farthest.

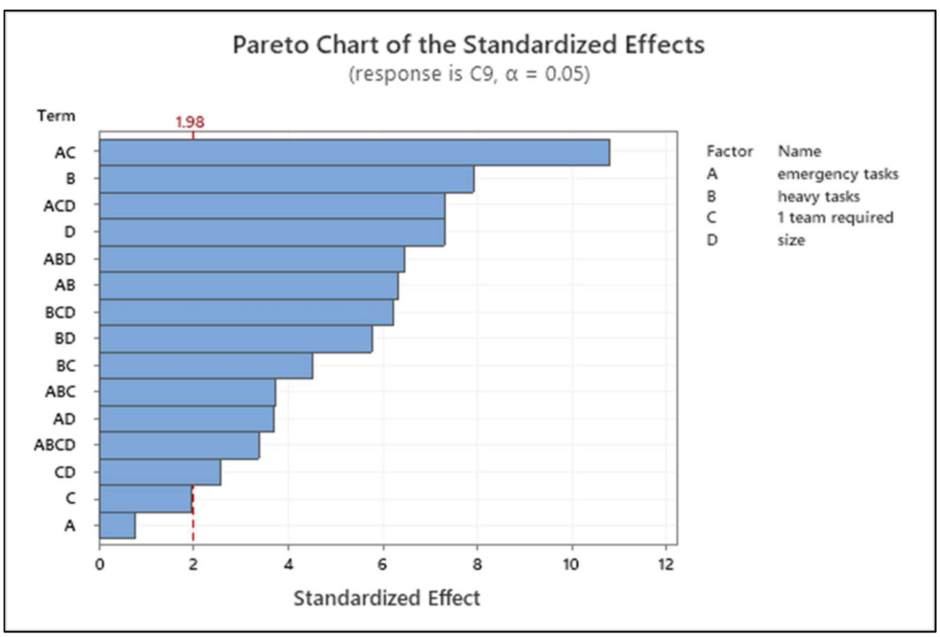

Figure 5.5. Pareto chart of the first DOE

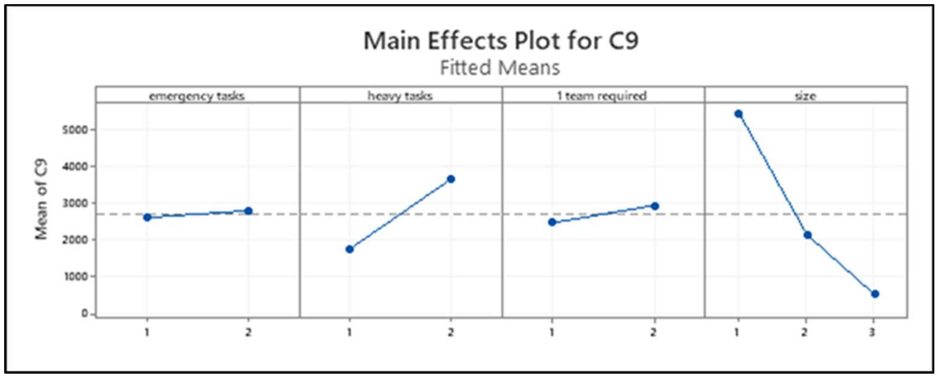

Figure 5.6. The main effect plot of the first DOE

In the second DOE, the number of emergency tasks (A), the number of heavy tasks (B), the percent of tasks requiring only one team (C), and the weights of the objective functions (D) are considered independent factors. The dependent factor is execution time. The result is shown in Figure 5.7 and 5.8.

It can be seen that only effects and interaction effects (AC), (B), (AB) and (ABC) are statistically significant.

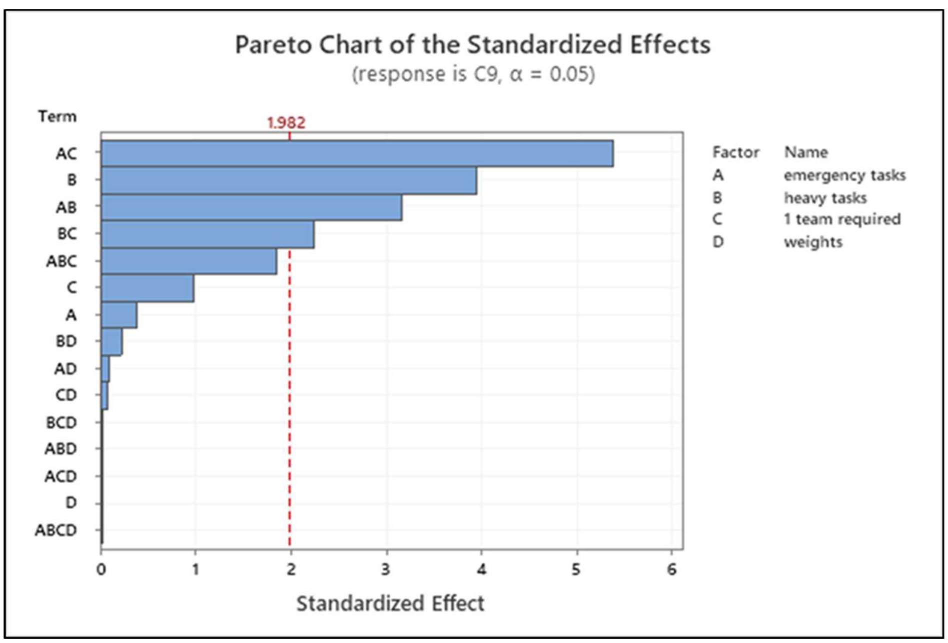

Figure 5.7. Pareto chart of the second DOE

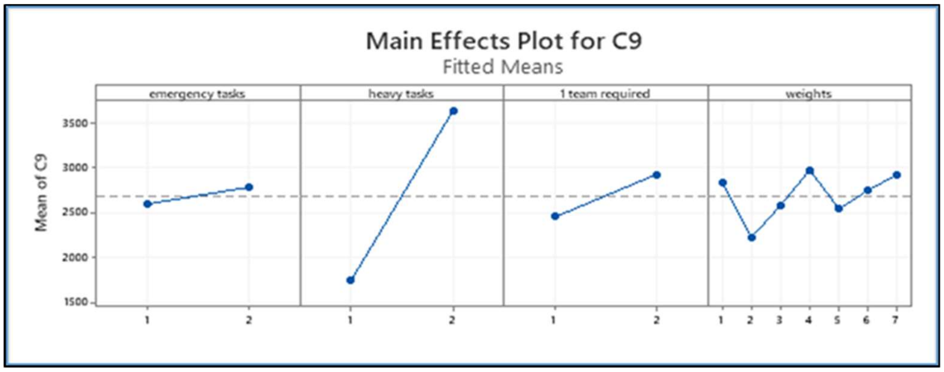

Figure 5.8. The main effect plot of the second DOE

5.4. Tuning Weights in the TOPSIS Method

In the decision-making phase of the proposed dynamic algorithm, the TOPSIS method helps decision-makers rank Pareto solutions and select the most appropriate one concerning weights of objective functions. So, it is necessary to observe how selecting different sets of weights for objectives can directly affect the results. For this reason, we run the proposed dynamic algorithm for 4-hour intervals separately for different instances. In other words, we run the algorithm once for a 4-hour interval without dynamic inputs. The results are presented in Table 5.9. The optimum solution for each objective is bolded. The results show the straight relation between the weights of objective functions and the selected final Pareto solution.

5.5. Tuning Grid Points In Augmented ε -constraint Method

The augmented ε-constraint method is used to find the Pareto solution set; one of this method's essential elements is the number of grid points. Three sets of grid points are tested to analyze the effect of the number of grid points on the performance of the augmented ε-constraint method and dynamic method, respectively.

Two different instances with different parameters are tested, and the results are shown in Table 5.10 and 5.11.

Since there are three objective functions in the rescheduling and reassignment phase of the proposed dynamic algorithm, the augmented ε-constraint method considers three objective functions, so the grid point sets consist of three numbers.

For the first instance, Table 5.10, for the 10*10*10 grid points, 15 Pareto solutions are obtained within 29623 seconds. The number of obtained Pareto solutions for 5*5*5 grid points is 11 within 1495 seconds, while with the 3*3*3 grid points, only seven Pareto solutions are produced during 999 seconds. It is noteworthy that the generated Pareto solutions for different sets of grid point may not be the same; these solutions are highlighted in the table.

For the second instance, the 10*10*10, 5*5*5, and 3*3*3 sets of grid points generated 15, 14, and 12 Pareto solutions. Their running times (CPU time) are 1738, 586, and 311 seconds respectively.

In the trade-off between the algorithm's running time and the number of obtained Pareto solutions, the grid point equals five is selected as an appropriate one.

For solving all instances in this research, the 5*5*5 grid point set is applied.

5.6. Tuning the Length of Intervals

Since the considered environment is dynamic, and new data enter the system dynamically (gradually), to cope with this challenge and process these new emergency tasks, refining the prepared daily schedule is necessary. Most of the time, there are three rest times for workers on a shift, two short breaks, and one lunch break. So in this study, the length of an interval for refining schedule is set to 2 hours. Hence for an 8-hour shift, four intervals are considered totally, which means that reschedule is performed once each two hours. It is necessary to mention that the length and the number of intervals for an 8-hour shift can be adjusted and changed related to conditions and management requirements; furthermore, the length of the intervals in a day can be the same or not.

5.7. Weights of Objective Functions

This research is aiming to automate the decision making procedure, which contains generating the daily schedules. Extra, the generated daily schedules should be better or similar to the one that the manager achieves manually. Different weights setting for objective functions lead to different final schedules. So, the Steuers method is used to define different weights to analyze the impact of diverse weight setting on final results.

Objective functions' weights indicate the relative importance of them. These weights can be changed per shift or even per interval depending on the system's demand or manager's requirements at the time. For instance, when there are many emergency tasks in the system, the OBJ1 gets higher importance than other functions to handle this situation. However, in a normal situation, when the system is not under pressure, maybe the higher weights for OBJ2 and OBJ3 are more desirable.

Table 5.9. Results of different weights in 4-hour shift

		Sample 1			Sample 2		
		OBJ 1	OBJ 2	OBJ 3	OBJ 1	OBJ 2	OBJ 3
(1,0,0)	λ1	**9.84E+10**	93.33	5.33	**1.12E+11**	263.33	12.67
(0,1,0)	λ2	1.06E+11	**9.33**	12.67	1.27E+11	**2.67**	14.67
(0,0,1)	λ3	9.86E+10	419.33	**1.33**	1.12E+11	526.67	**1.33**
(1/9,4/9,4/9)	λ4	9.95E+10	92.67	2.67	1.26E+11	51.33	1.33
(4/9,1/9,4/9)	λ5	9.87E+10	172.67	**1.33**	1.24E+11	66.67	**1.33**
(4/9,4/9,1/9)	λ6	1.05E+11	42.67	4.67	1.14E+11	10.67	9.33
(1/3,1/3,1/3)	λ7	9.95E+10	92.67	2.67	1.26E+11	51.33	1.33
		Sample 3			Sample 4		
		OBJ 1	OBJ 2	OBJ 3	OBJ 1	OBJ 2	OBJ 3
(1,0,0)	λ1	**1.47E+11**	464.67	23.33	**9.3E+10**	728.00	14.00
(0,1,0)	λ2	1.67E+11	**9.33**	15.33	1.06E+11	**18.00**	18.00
(0,0,1)	λ3	1.48E+11	207.33	**1.33**	9.3E+10	116.00	**8.00**
(1/9,4/9,4/9)	λ4	1.51E+11	65.33	1.33	9.3E+10	84.00	10.00
(4/9,1/9,4/9)	λ5	1.51E+11	65.33	**1.33**	9.3E+10	116.00	**8.00**
(4/9,4/9,1/9)	λ6	1.5E+11	27.33	4.67	9.3E+10	84.00	10.00
(1/3,1/3,1/3)	λ7	1.51E+11	65.33	1.33	9.3E+10	84.00	10.00
		Sample 5			Sample 6		
		OBJ 1	OBJ 2	OBJ 3	OBJ 1	OBJ 2	OBJ 3
(1,0,0)	λ1	**1.21E+11**	298.67	11.33	**9.48E+10**	577.33	2.00
(0,1,0)	λ2	1.39E+11	**2.67**	23.33	1.18E+11	**6.67**	14.00
(0,0,1)	λ3	1.26E+11	473.33	**1.33**	9.48E+10	677.33	**0.00**
(1/9,4/9,4/9)	λ4	1.26E+11	26.67	5.33	9.96E+10	139.33	2.00
(4/9,1/9,4/9)	λ5	1.27E+11	153.33	**3.33**	9.71E+10	346.67	**0.00**
(4/9,4/9,1/9)	λ6	1.26E+11	26.67	5.33	9.53E+10	29.33	8.00
(1/3,1/3,1/3)	λ7	1.26E+11	26.67	5.33	9.96E+10	139.33	2.00
		Sample7			Sample 8		
		OBJ 1	OBJ 2	OBJ 3	OBJ 1	OBJ 2	OBJ 3
(1,0,0)	λ1	**7.86E+10**	504.00	31.33	**1.16E+11**	260.00	36.67
(0,1,0)	λ2	9.21E+10	**2.00**	26.67	1.43E+11	**18.00**	40.67
(0,0,1)	λ3	7.87E+10	698.00	**6.67**	1.16E+11	170.00	**6.67**
(1/9,4/9,4/9)	λ4	9.19E+10	82.00	12.67	1.16E+11	110.00	18.67
(4/9,1/9,4/9)	λ5	7.94E+10	182.00	**8.67**	1.16E+11	170.00	**6.67**
(4/9,4/9,1/9)	λ6	7.92E+10	24.00	19.33	1.43E+11	18.00	40.67
(1/3,1/3,1/3)	λ7	9.19E+10	82.00	12.67	1.16E+11	110.00	18.67

Table 5.10 Results of different grid points for the first instance

Number of Grid Points	10*10*10			5*5*5			3*3*3		
	OBJ 1	OBJ 2	OBJ 3	OBJ 1	OBJ 2	OBJ 3	OBJ 1	OBJ 2	OBJ 3
Pareto solution 1	179618074887	244	15.3333	179618455449	244	15.3333			
Pareto solution 2	179618074887	286	4.6667	179618074887	286	4.6667	179618074887	286	4.6667
Pareto solution 3	179618411737	188	18.6667	179618417606	188	18.6667			
Pareto solution 4	179782081711	208	6.6667	179782081711	208	6.6667			
Pareto solution 5	179782083571	244	4.6667	179782083571	244	4.6667	184915698660	244	2.6667
Pareto solution 6	179920828016	86	4.6667	179920828016	86	4.6667	179920828016	86	4.6667
Pareto solution 7	179921120220	406	2.6667						
Pareto solution 8	179921132284	374	2.6667						
Pareto solution 9	183914354539	344	2.6667						
Pareto solution 10	184035406598	244	2.6667	184854446826	244	2.6667			
Pareto solution 11	184242418124	208	3.3333						
Pareto solution 12	188079305557	436	1.3333	188079305557	436	1.3333	188079305557	436	1.3333
Pareto solution 13	188079306717	324	1.3333	188079306717	324	1.3333	189183184913	324	1.3333
Pareto solution 14	188459290301	62	2.6667	188670204321	62	2.6667	188624987490	62	2.6667
Pareto solution 15	195738110698	48	10.6667	195738110698	48	10.6667	195738110698	48	10.6667
CPU Time (sec)	29623.133636			1459			999.381867		

Table 5.11 Results of different grid points for the second instance

Number of Grid Points	**10*10*10**			**5*5*5**			**3*3*3**		
	OBJ 1	**OBJ 2**	**OBJ 3**	**OBJ 1**	**OBJ 2**	**OBJ 3**	**OBJ 1**	**OBJ 2**	**OBJ 3**
Pareto solution 1	1.43908E+11	578	20	1.43908E+11	578	20	1.43908E+11	578	20
Pareto solution 2	1.43908E+11	254	20	1.43908E+11	254	20	1.43908E+11	254	20
Pareto solution 3	1.43908E+11	52	34	1.43908E+11	52	34	1.43908E+11	52	34
Pareto solution 4	1.44009E+11	188	26	1.44009E+11	188	26			
Pareto solution 5	1.44009E+11	442	12	1.44009E+11	442	12	1.44009E+11	442	12
Pareto solution 6	1.44035E+11	94	12	1.44035E+11	94	12	1.44035E+11	94	12
Pareto solution 7	1.44221E+11	492	10	1.44221E+11	492	10			
Pareto solution 8	1.44221E+11	52	30						
Pareto solution 9	1.44653E+11	144	10	1.44653E+11	144	10	1.44932E+11	144	10
Pareto solution 10	1.66808E+11	4	26	1.66808E+11	4	26	1.67486E+11	4	26
Pareto solution 11	1.67104E+11	94	10	1.67104E+11	94	10	1.67671E+11	94	10
Pareto solution 12	1.70239E+11	358	8	1.70239E+11	358	8	1.70239E+11	358	8
Pareto solution 13	1.70276E+11	54	12	1.70276E+11	54	12	1.70276E+11	54	12
Pareto solution 14	1.70279E+11	592	4	1.70279E+11	592	4	1.70279E+11	592	4
Pareto solution 15	1.70495E+11	244	4	1.70495E+11	244	4	1.70495E+11	244	4
CPU Time (sec)	1737.889690			586.103869			310.966114		

6. CONCLUSION

The distribution center links overseas industrial sites with domestic suppliers. In terms of guaranteeing a high service quality and rate, distribution, operation planning, and scheduling play an important role. In the under-researched distribution center, task scheduling and workforce assignment in the repackaging section are key activities. Due to the current manual process for scheduling in this section, finding an alternative algorithm has been identified as an improvement opportunity.

In this research, the workforce scheduling problem at the repackaging section under real conditions is discussed. In this problem, some works require cooperation between more than one team simultaneously, and heavy tasks should not be performed consecutively. The main goal is to find the best work schedule and workforce allocation to maximize the number of processed tasks in a work shift with a limited number of work teams. Besides, finding a schedule by considering customer-oriented policy of the company, fairness aspect, and ergonomics features are desirable.

Due to the high complexity of the problem, a dynamic multi-objective solution method and a greedy heuristic algorithm that provides appropriate schedules were suggested.

After coding these algorithms with MATLAB 2014a and GAMS 25.1 and comparing their results with the available information, the efficiency of these algorithms was measured. The results show that acceptable answers are obtained in a shorter time. Therefore, it is recommended to implement these algorithms in the distribution center.

In future research, a more accurate model in favor of the ergonomic aspect of the problem can be considered. Furthermore, metaheuristic methods can be used for solving this problem. One opportunity for future study can apply the proposed dynamic hybrid method to similar problems. On another side, although this model is designed for cross-dock platforms, it can be adopted to other human-centered systems.

REFERENCES

[1] M. Pinedo, *Scheduling*, vol. 29. Springer, 2012.

[2] Van Den Bergh J., Beliën J., De Bruecker P., Demeulemeester E., De Boeck L., Personnel Scheduling: A Literature Review, *European Journal of Operational Research*, 2013, **226**(3), 367–385.

[3] Boudreau J., Hopp W., McClain J. O., Joseph Thomas L., On the Interface between Operations and Human Resources Management, *Manufacturing & Service Operations Management.*, 2003, **5**(3), 179–202.

[4] Serrano C., Delorme X., Dolgui A., Distribution and Operation Planning at a Cross-dock Platform: A Case of Study at Renault, *4th IEEE International Conference on Advanced Logistics and Transport (IEEE ICALT 2015*), Valenciennes, France, 20-22 May 2015.

[5] Fanti M. P., Stecco G., Ukovich W., Solving Scheduling Problems in Distribution Centers by Mixed Integer Linear Programming Formulations, *IFAC Proceedings Volumes (IFAC-PapersOnline)*, Trieste, Italy, 24-27 August 2011.

[6] Lim A., Rodrigues B., Song L., Manpower Allocation with Time Windows, *Journal of the Operational Research Society*, 2004, **55**(11), 1178–1186.

[7] Ladier A. L., Alpan G., Penz B., Joint Employee Weekly Timetabling and Daily Rostering: A Decision-support Tool for a Logistics Platform, *European Journal of Operational Research*, 2014, **34**(1), 278–291.

[8] M. Pour S., Drake J. H., Ejlertsen L. S., M. Rasmussen K., Burke E. K., A Hybrid Constraint Programming/Mixed Integer Programming Framework for the Preventive Signaling Maintenance Crew Scheduling Problem, *European Journal of Operational Research,* 2018, **269**(1), 341–352.

[9] Jütte S., Müller D., Thonemann U. W., Optimizing Railway Crew Schedules with Fairness Preferences, *Journal of Scheduling*, 2017, **20**(1), 43–55.

[10] Castillo-Salazar J. A., Landa-Silva D., Qu R., Workforce Scheduling and Routing Problems: Literature Survey and Computational Study, *Annals of Operations Research*, 2016, **239**(1), 39–67.

[11] Fink M., Desaulniers G., Frey M., Kiermaier F., Kolisch R., Soumis F., Column Generation for Vehicle Routing Problems with Multiple Synchronization Constraints, *European Journal of Operational Research,* 2019, **272**(2), 699–711.

[12] Drexl M., Synchronization in Vehicle Routing - A Survey of VRPs with Multiple Synchronization Constraints, *Transportation Science*, 2012, **46** (3), 297–316.

[13] Li Y., Lim A., Rodrigues B., Manpower Allocation with Time Windows and Job-teaming Constraints, *Naval Research Logistics*, 2005, **52**(4), 302–311.

[14] Zamorano E., Stolletz R., Branch-and-price Approaches for the Multiperiod Technician Routing and Scheduling Problem, *European Journal of Operational Research*, 2017, **257**(1), 55–68.

[15] Dohn A., Kolind E., Clausen J., The Manpower Allocation Problem with Time Windows and Job-teaming Constraints: A Branch-and-price Approach, *Computers and Operations Research*, 2007, **36**(4), 1145–1157.

[16] Cordeau J. F., Laporte G., Pasin F., Ropke S., Scheduling Technicians and Tasks in a Telecommunications Company, *Journal of Scheduling*, 2010, **13**(4), 393–409.

[17] Labadie N., Prins C., Yang Y., Iterated Local Search for a Vehicle Routing Problem with Synchronization Constraints, *Proceedings of the 3rd International Conference on Operations Research and Enterprise Systems (CORES 2014)*, Loire Valley, France, 6-8 March 2014.

[18] Ait Haddadene S. R., Labadie N., Prodhon C., A GRASP × ILS for the Vehicle Routing Problem with Time Windows, Synchronization and Precedence Constraints, *Expert Systems with Applications*, 2016, **66**, 1339–1351.

[19] Afifi S., Dang D. C., Moukrim A., A Simulated Annealing Algorithm for the Vehicle Routing Problem with Time Windows and Synchronization Constraints, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, **7997** , 259–265.

[20] Kim B. I., Koo J., Park J., The Combined Manpower-vehicle Routing Problem for Multi-staged Services, *Expert Systems with Applications*, 2010, **37**(12), 8424–8431.

[21] Bredstrom D., Ronnqvist M., A Branch and Price Algorithm for the Combined Vehicle Routing and Scheduling Problem With Synchronization Constraints, *SSRN Electronic Journal*, 2011.

[22] Luo Z., Qin H., Zhu W., Lim A., Branch-and-price-and-cut for the Manpower Routing Problem with Synchronization Constraints, *Naval Research Logistics*, 2016, **63**(2) ,138–171.

[23] Cai Y., Zhang Z., Guo S., Qin H., Lim A., A Tree-Based Tabu Search Algorithm for the Manpower Allocation Problem with Time Windows and Job-Teaming Constraints, *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, Beijing, China, 3-9 August 2013.

[24] Nasir J. A., Kuo Y. H., A Decision Support Framework for Home Health Care Transportation with Simultaneous Multi-vehicle Routing and Staff Scheduling Synchronization, *Decision Support Systems*, 2020, **138**, 113361.

[25] Wickens C. D., Gordon S. E., Liu Y., *An Introduction to Human Factors Engineering*, Longman, New York, 1998.

[26] https://iea.cc/ (Accessed Dec. 02, 2020).

[27] Carnahan B. J., Redfern M. S., Norman B., Designing Safe Job Rotation Schedules Using Optimization and Heuristic Search, *Ergonomics*, 2000, **43**(4), 543–560.

[28] Lodree E. J., Geiger C. D., Jiang X., Taxonomy for Integrating Scheduling Theory and Human Factors: Review and Research Opportunities, *International Journal of Industrial Ergonomics*, 2008, **39**(1), 39-51.

[29] Waters T. R., Putz-Anderson V., Garg, A., Fine L. J., Revised NIOSH Equation for the Design and Evaluation of Manual Lifting Tasks, *Ergonomics*, 1993, **36**(7), 749-776.

[30] Moore J. S., Garg A., The Strain Index: A Proposed Method to Analyze Jobs for Risk of Distal Upper Extremity Disorders, *American Industrial Hygiene Association Journal*, 1995, **56**(5), 443-458.

[31] McAtamney L., Nigel Corlett E., RULA: A Survey Method for the Investigation of Work-related Upper Limb Disorders, *Applied Ergonomics*, 1993, **24**(2), 91-99.

[32] Hignett S., McAtamney L., Rapid Entire Body Assessment (REBA), *Applied Ergonomics*, 2000, **31**(2), 201-205.

[33] https://www.eaws.it/ (Accessed Dec. 02, 2020).

[34] Occhipinti E., OCRA: A Concise Index for the Assessment of Exposure to Repetitive Movements of the Upper Limbs, *Ergonomics*, 1998, **41**(9), 1290-1311.

[35] Hochdörffer J., Hedler M., Lanza G., Staff Scheduling in Job Rotation Environments Considering Ergonomic Aspects and Preservation of Qualifications, *Journal of Manufacturing Systems*, 2018, **46**, 103-114.

[36] Otto A., Battaïa O., Reducing Physical Ergonomic Risks at Assembly Lines by Line Balancing and Job Rotation: A Survey, *Computers and Industrial Engineering*, 2017, **111**, 467-480.

[37] Yoon S. Y., Ko J., Jung M. C., A Model for Developing Job Rotation Schedules that Eliminate Sequential High Workloads and Minimize Between-worker Variability in Cumulative Daily Workloads: Application to Automotive Assembly Lines, *Applied Ergonomics*, 2016, **55**, 8-15.

[38] Reis D., Moro A., Ramos E., Reis P., Upper Limbs Exposure to Biomechanical Overload: Occupational Risk Assessment in a Poultry Slaughterhouse, *Advances in Intelligent Systems and Computing*, 2016, **489**, 275-282.

[39] Occhionero V., Ghersi R., Prandini L., Korpinen L., Gobba F., The Biomechanical Overload of the Upper Limb: A Neglected Occupational Hazard in Animal Facility Operators, *Ergonomics*, 2017, **60**(3), 366-374.

[40] Lasota A. M., Ergonomic Evaluation of Physical Risk for Packing Line Operators, *Logistics and Transport*, 2015, **26**(2).

[41] Occhipinti E., Colombini D., A Checklist for Evaluating Exposure to Repetitive Movements of the Upper Limbs Based on the OCRA Index, *International Encyclopedia of Ergonomics and Human Factors, Second Edition - 3 Volume Set*, CRC Press, 2006.

[42] Colombini D., Occhipinti E., Álvarez-Casado E., The Revised OCRA Checklist Method, *Factors Humans,* Barcelona, Spain, 2013.

[43] Paulsen R., Gallu T., Gilkey D., Reiser R., Murgia L., Rosecrance J., The Inter-rater Reliability of Strain Index and OCRA Checklist Task Assessments in Cheese Processing, *Applied Ergonomics*, 2015, **51**, 199-204.

[44] Rosecrance J., Paulsen R., Murgia L., Risk Assessment of Cheese Processing Tasks Using the Strain Index and OCRA Checklist, *International Journal of Industrial Ergonomics*, 2017, **61**, 142-148.

[45] Tiacci L., Mimmi M., Integrating Ergonomic Risks Evaluation Through OCRA Index and Balancing/Sequencing Decisions for Mixed Model Stochastic Asynchronous Assembly Lines, *Omega (United Kingdom)*, 2018, **78**, 112-138.

[46] Bard J. F., Purnomo H. W., Preference Scheduling for Nurses Using Column Generation, *European Journal of Operational Research*, 2005, **164**(2), 510-534.

[47] Ouelhadj D., Martin S., Smet P., Ozcan E., Berghe V., *Fairness in Nurse Rostering*, University of Portsmouth, 2012.

[48] Abbink E., Fischetti M., Kroon L., Timmer G., Vromans M., Reinventing Crew Scheduling at Netherlands Railways, *Interfaces*, 2005, **35**(5), 393-401.

[49] Blöchliger I., Modeling Staff Scheduling Problems. A Tutorial, *European Journal of Operational Research*, 2004, **158**(3), 533-542.

[50] Simons T., Roberson Q., Why Managers Should Care About Fairness: The Effects of Aggregate Justice Perceptions on Organizational Outcomes, *Journal of Applied Psychology*, 2003, **88**(3), 432-443.

[51] De Boer E. M., Bakker A. B., Syroit J. E., Schaufeli W. B., Unfairness at Work as a Predictor of Absenteeism, *Journal of Organizational Behavior*, 2002, **23**(2), 181-197.

[52] Mühlenthaler M., Wanka R., Fairness in Academic Course Timetabling, *Annals of Operations Reearch*, 2013, **239**(1), 171-188.

[53] Martin S., Ouelhadj D., Smet P., Vanden Berghe G., Özcan E., Cooperative Search for Fair Nurse Rosters, *Expert Systems with Applications*, 2013, **40**(16), 6674-6683.

[54] Stolletz R., Brunner J. O., Fair Optimization of Fortnightly Physician Schedules with Flexible Shifts, *European Journal of Operational Research*, 2012, **219**(3), 622-629.

[55] Aghaei J., Amjady N., Shayanfar H. A., Multi-objective Electricity Market Clearing Considering Dynamic Security by Lexicographic Optimization and Augmented Epsilon Constraint Method, *Applied Soft Computing*, 2011, **11**(4), 3846-3858.

[56] Ben Abdelaziz F., Multiple Objective Programming and Goal Programming: New Trends and Applications, *European Journal of Operational Research*, 2007, **177**(3), 1520–1522.

[57] Zhang W., Reimann M., A Simple Augmented ∈-constraint Method for Multi-objective Mathematical Integer Programming Problems, *European Journal of Operational Research*, 2014, **234**(1), 15-24.

[58] Geoffrion A. M., Dyer J. S., Feinberg A., An Interactive Approach for Multi-Criterion Optimization, with an Application to the Operation of an Academic Department, *Management Science*, 1972, **19**(4), 357–368.

[59] Steuer R. E., Choo E. U., An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming, *Mathematical Programming*, 1983, **26**(3), 326-344.

[60] Wierzbicki A. P., *The Use of Reference Objectives in Multiobjective Optimization*, Springer, Berlin, Heidelberg, 1980.

[61] Jaszkiewicz A., Słowiński R., The 'Light Beam Search' Approach - An Overview of Methodology and Applications, *European Journal of Operational Research*, 1999, **113**(2), 300-314.

[62] Gass S., Saaty T., The Computational Algorithm for the Parametric Objective Function, *Naval Research Logistics Quarterly*, 1955, **2**(1–2), 39-45.

[63] Deb K., *Multi-objective optimization using evolutionary algorithms*, **16,** John Wiley & Sons, 2001.

[64] Mavrotas G., Effective Implementation of the ε-constraint Method in Multi-Objective Mathematical Programming Problems, *Applied Mathematics and Computation*, 2009, **213**(2), 455-465.

[65] Zionts S., Multiple Criteria Mathematical Programming: an Overview and Several Approaches, *Multiple Criteria Decision Methods and Applications*, Springer Berlin Heidelberg, 1985.

PUBLICATIONS AND WORKS

[1] Ahmadizar F., **Shahmaleki P.**, Groupshop Scheduling with Sequence-dependent Setup and Transportation Times, *Applied Mathematical Modeling,* 2014, **38**, 5080-5091.

[2] **Shahmaleki P.**, Fıglalı A., Workforce Scheduling with Synchronization Constraints and Ergonomic Aspects at Cross Dock Platforms, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 2021, **25**, 113-128.

[3] Shahmaleki P., Mahzoon M., **Shahmaleki P.**, Designing Controller System for Integrated Solar Power Plant Using Fuzzy Method, *Proceeding of the ASME International Mechanical Engineering Conference and Exposition (IMECE)*, Boston, MA, USA, 2008.

[4] Shahmaleki P., **Shahmaleki P.**, Hybrid Solar Power Plant, *National Conference of Improvement*, Shiraz, Iran, 2010.

[5] Sivri MS., Özgün S., **Shahmaleki P.**, Interval Valued Hesitant Fuzzy Analytic Hierarchy Process and Its Application to Group Decision Making , *The 5th International Fuzzy Systems Symposium (FUZZYSS'17)*, Ankara, Turkey, 2017.

[6] Yıldırım S.T., Keskin F.S., Baynal K., **Shahmaleki P.**, Effects of Perlite and Bottom Ash on Properties of Cement Binder Mortar , *22nd Congress on Thermal Science and Technology*, Kocaeli, Turkey, 11-14 September 2019.

CURRICULUM VITAE

Parmis Shahmaleki received her Bachelor of Science in Industrial Engineering from the Isfahan University of Technology, Isfahan, Iran and Master of Science in Industrial Engineering from Kordestan University, Sanandaj, Iran. She completed her Doctor of Philosophy (PhD) in Industrial Engineering at Kocaeli University, Turkey. She is working in the field of operation research, her research area includes Scheduling, Supply chains, Decision Making and Metaheuristics.