Chapter 4
A Multi-Objective Home Healthcare Delivery Problem

4.1. Introduction

In the presence of multiple and often conflicting objectives or goals, finding the right balance among multiple competitive goals in one of the most important tasks. Excess importance given to one goal will lead to problems in achieving the others. Decision-maker not only needs to find a suitable balance for their conflicting objectives but also needs to incorporate the requirements of various stakeholders in the decision-making process. Achieving said balance becomes much harder when the relative importance (weightage) of the various objectives is not straightforward. In addition to this, the relative importance of various objectives of a stakeholder is generally found to be dynamic in nature that changes with day to day needs of the organization as well as the evolving cultural and social environment. A fixed but unfairly assumed importance for stakeholders' objectives will consistently add to the unfair treatment of one party over the other. To ensure the long-term sustainability of any organization, it is necessary for management to cater to the legitimate requirements of all its stakeholders, such as investors, employees, and customers (Vidal et al., 2020). To overcome this issue, in this chapter, the categorization of objectives is done based on the competitive goals of various stakeholders. Additionally, we have tried to develop a framework for incorporating this methodology by proposing a multi-objective mathematical model for a single-period home healthcare delivery problem.

Hence, the current chapter brings the generalized home healthcare model presented in Chapter 3 to a multi-objective setting. Further, we have tried to meticulously categorize the various objectives introduced in the literature so far and provide the framework for obtaining a set of solutions that cover a wide variety of relative importance. In addition to that, a major focus of this study is to improve the quality of schedule for the patients requiring multiple visits in a single day. In the existing literature, little attention has been given to the quality of the schedule, and often a feasible schedule that satisfies the basic time window constraints is sought. In the case of multiple visits, this aspect of the scheduling decision needs to be further improved. We have provided a detailed definition for the "quality of schedule" based on the minimum interference with the patient's daily routine and penalized the deviation from it in the objective function of the proposed mixed-integer programming model. To further accommodate the multiple visits with complicated precedence requirements, we present a mathematical model that deals with the patient-specific inconvenient time window and tries to avoid it as much as possible. To achieve this, we implement the provided inconvenient time window as a soft constraint and minimize its violation as one of the objectives. Violation of the proposed soft constraint and the penalty on the improper schedule is cumulatively termed as the 'cost of improper scheduling' and optimized as an objective in the mixed-integer programming model. The complexity of the problem being considered necessitates multi-objective or multi-criteria modeling of a problem, where competitive objectives are treated similarly to generate a pool of solutions that aptly represent the trade-off among the conflicting objectives.

Fikar and Hirsch (2017), in their review of the single-period HHC problem, considered twenty-five journal articles. Out of which, only nine have modeled the

HHC problem as a single objective optimization problem, which points toward the capability of multi-objective formulation to accommodate requirements of complex HHC setup. A major challenge with multi-objective models is the handling of multiple considered objectives. In most cases, soft constraint violation and the Weighed sum method have been used to bring down the effective number of objectives. Trautsamwieser and Hirsch (2011) utilize the weighted sum approach to bundle the seven different objective criteria into one single objective function. Similarly, Hiermann et al. (2015) have considered 13 different criteria that are eventually divided into three separate groups. The Weighted sum approach is used within the group to combine the effect of similar criteria, and then the normalized value of each group is combined to build the final objective function. By following the existing literature, it is clear that the HHC delivery problem has the potential to incorporate multiple novel ideas and, in turn, can contain enormous complexity. Hence, the necessity to develop the appropriate method to handle the complexity of the problem-at-hand is always felt. In order to solve the proposed multi-objective home healthcare delivery problem, the efficient implementations of Reference-point based non-dominated sorting genetic algorithm (NSGA-III), Multi-objective particle swarm optimization (MOPSO), and Multi-objective grey wolf optimizer (MOGWO) is developed. The method does not need to decide the weightage of different objectives a-priory as in the case of the weighted sum method that is predominantly used in the literature related to HHC delivery. In addition to this, the algorithm can produce a diverse pool of non-dominated solutions in a single run. The well-established Taguchi method is utilized to fine-tune the various parameters of the algorithm based on the quality of the obtained Pareto front approximation.

The organization of the rest of the chapter is as follows. First, the considered multi-objective home healthcare delivery problem is described in Section 4.2. with Section 4.3 presenting the mathematical model. In Section 4.4, an illustrative example is used to demonstrate the capability of the developed MIP model. Section 4.5 details the various solution approaches and their operators. Alongside the experiment setup and the details of the selected Pareto-front quality indicators, Section 4.6 includes the results and analysis of the computational experiment. The approximate Pareto front obtained for a medium size problem instance is further analyzed in order to establish the relationship between patient convenience and net profit. Finally, Section 4.7 concludes the chapter by summarizing the findings.

4.2. Problem description

In this section, we present the details of a multi-objective home healthcare delivery problem faced by a service provider for assigning, scheduling, and routing of the required caregivers. At any given instance, a number of medical procedures, such as surgical dressing, catheterization, and nebulization, are to be offered to the patients at their homes by the considered service provider. Each procedure can be characterized by the service time and the revenue associated with the single visit of that procedure. Dependencies between two procedures may necessitate the need to maintain a specified time gap. For example, procedure A cannot be performed within one hour of the completion of procedure B. Similarly, a predetermined time gap is to be inserted between the consecutive visits of the same procedure that requires multiple visits. For example, the second visit of a procedure that needs exactly two visits has to be scheduled 8 hours after the first visit (5 hours gap is implemented for procedures with visit-frequency 3). These procedures are offered to a set of patients spread over a geographical area. Locations of the patients and

travel time are assumed to be known in advance. A patient may request one or more procedures to be delivered to his/her location. Depending on the patient-procedure combination, the required number of visits for a selected request needs to be scheduled. Similarly, there may be a case where a patient-procedure combination cannot be performed with only one caregiver (e.g., to lift the patient). In that case, based on the requirement, a sufficient number of caregivers are scheduled. Assignment of the caregiver to the visit for selected procedures is made while considering the multiple compatibility criteria. Patient's preference regarding the caregiver's gender, as well as their capability to communicate with each other (at least one common language), is ensured during the assignments. The proficiency of the caregiver in administering the assigned procedure is also taken into account during this stage. The skill level of the caregiver must match or exceed the skill level required for the assigned visits.

In addition to the compatibility check, several other criteria are also taken into consideration before assigning visits to a caregiver. During the outbreak of a disease such as the novel coronavirus (COVID-19), strict restriction to limit unnecessary contacts is also implemented. During team-ups (visits requiring multiple caregivers), interactions between every patient and caregiver are tracked and kept under the advised limit. The caregiver's availability at a specific time is also considered while scheduling. A caregiver is only available in the morning or evening shifts, where a shift can be uniquely classified by the start and end time along with the earliest and latest start time for a fixed length mandatory break. The routes for the caregivers, designed to adhere to these time-window requirements, should also start and end at the designated location called hubs. Hubs are assumed to be spread throughout the considered geographical region. In addition to 8 hours

of the regular workload, a maximum of 4 hours of overtime work can be added to the end of a caregiver's shift. Overtime work is compensated at double the rate of regular work. The distribution of overtime is to be done in an egalitarian manner to provide equal opportunity of gaining higher pay to each caregiver. Similar to the availability of caregivers, the patients' availability at a specific time should also be taken into consideration. If possible, visits to a patient need to be scheduled to avoid the inconvenient time window provided by the patient. The desirability of a schedule to the patient largely depends on the number of interruptions during the day. Hence, the visits are scheduled in groups where multiple procedures are scheduled close to each other. These groups are termed as 'sessions' for further discussion.

The problem described above is, in essence, a multi-depot heterogeneous fleet vehicle routing problem with time windows (MD-HVRPTW) with additional side constraints. A solution to the problem should provide the selection, assignment, scheduling, and routing decisions. Compatibility criteria along with the limited contact restrictions for making visit-caregiver assignments, are implemented as a hard constraint. Similarly, time-gap requirements due to dependent procedures (precedence) or visit-frequency are also treated as hard restrictions in the proposed model. In contrast, the caregiver's workload requirement is implemented as a soft constraint with a suitable strict upper limit. Equal distribution of regular workload (with fixed pay) and overtime work (with the opportunity of gaining a higher pay rate) is optimized in one of the objective functions. Additionally, the inconvenient time window provided by patients is only used for scheduling as a last resort. The extent to which a solution respects the provided time window requirement is optimized as a part of a separate objective function. The remaining part of the same

objective function deals with the requirement of suitably organized visits where minimum interruptions to the patient's day-to-day routine are awarded. A modified definition for makespan is used to qualify the schedule in this regard. Finally, as described in the introduction section, objective functions are developed to separately cater to the requirements and the goals of competing stakeholders. While one of the objectives optimizes the balanced workload distribution among caregivers, a legitimate requirement for caregivers, the other tries to find the most convenient schedule for the patients. The next two objectives deal with the well-documented managerial objectives of optimizing the cost of daily operations. Net profit, as a net inflow of the money, is optimized along with the 'loss of the employed labor' as a separate objective. While both of the later objectives can be expressed in monetary terms, their relative importance for a specific service provider cannot be established without additional analysis. Hence, they are modeled as a separate objective function.

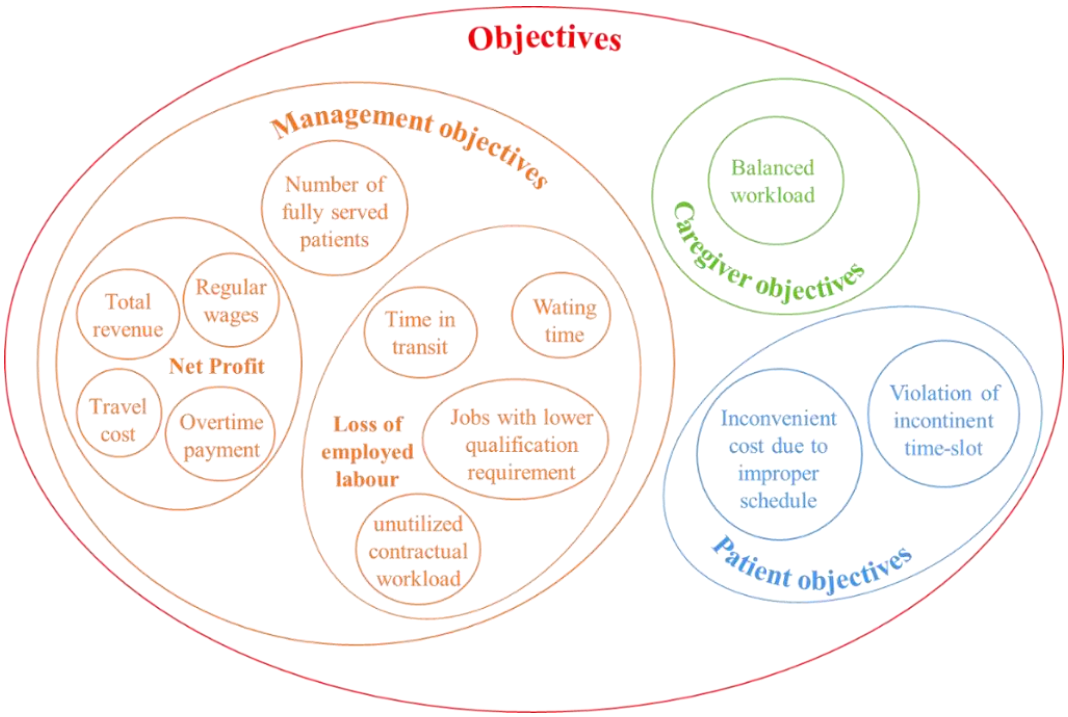


Figure 4.1: Objective functions considered in the proposed model.

Finally, the mathematical model adheres to the conclusion made in Chapter 3 and adopts the policy of 'partial accommodation'. Under the said policy, the complete fulfillment of all the requests for the selected patient is not deemed necessary. Hence, a separate objective function is used to maximize the number of fully served patients. **Figure 4.1** presents a pictorial representation of the various stakeholders' objectives and their components. It should be clear from the above discussion that the proposed mathematical model is a multi-objective mixed-integer programming model. Further details of the individual objective, along with the method of their implementation with constraint definition, are explained in the subsequent text. It should be noted that for the sake of consistency, we have followed the same notation and modelling style as the previous chapter.

4.3. Mathematical model [M]

Sets and indices:

M	Set of patients, indexed by i, j
H	Set of hubs, indexed by h
N	Set of all nodes, $N = H \cup M$
P_1	Set of all available medical procedures to patients, indexed by p, q
P_0	Mandatory reporting at the start and end of the shift (dummy task).
P	Set of all tasks, $P = P_0 \cup P_1$
V	Set of visits, indexed by v, u
K	Set of caregivers, indexed by k, l
G	Set of genders, indexed by g
L	Set of languages, indexed by e
S	Set of shifts, indexed by s

C Set of sessions, indexed by *c*

Input parameters:

Patient-related input parameters:

P'_{ip} $\begin{cases} 1, \text{ if procedure } p \text{ is requested by patient } i. \\ 0, \text{ otherwise.} \end{cases}$

V_{ip} Number of visits for procedure *p* is needed for patient *i* (visit frequency).

R_{ip} Number of caregivers needed to perform procedure *p* for patient *i*.

F'_{ip} Minimum skill-level needed to administer procedure *p* to patient *i* (on a scale of 1 to 10).

L'_{ie} $\begin{cases} 1, \text{ if patient } i \text{ speaks the language } e. \\ 0, \text{ otherwise.} \end{cases}$

G'_{ig} $\begin{cases} 1, \text{ if patient } i \text{ is comfortable with caregiver of gender } g. \\ 0, \text{ otherwise.} \end{cases}$

$\bar{T}'_i$ Start of the inconvenient time window (ITW) for patient *i*.

$\bar{\bar{T}}'_i$ End of the inconvenient time window (ITW) for patient *i*.

λ' Maximum allowed distinct contact for a patient.

$\tilde{T}_{ip}$ Minimum time gap between two visits of same procedure *p* for patient *i* (in minutes).

Caregiver-related input parameters:

F''_{kp} Skill-level of staff k in performing procedure p (from 1 to 10)

L''_{ke} $\begin{cases} 1, \text{ if caregiver } k \text{ speaks language } e. \\ 0, \text{ otherwise.} \end{cases}$

G''_{kg} $\begin{cases} 1, \text{ if caregiver } k \text{ is a person of gender } g. \\ 0, \text{ otherwise.} \end{cases}$

λ'' Maximum allowed contacts for a caregiver.

H''_{kh} $\begin{cases} 1, \text{ if caregiver } k \text{ starts and ends shift at hub } h. \\ 0, \text{ otherwise.} \end{cases}$

S''_{ks} $\begin{cases} 1, \text{ if caregiver } k \text{ is available in shift } s. \\ 0, \text{ otherwise.} \end{cases}$

W_K Maximum overtime allowed for staff *k* (in minutes).

C_k^d Daily cost of employing staff k for a regular shift.

C_k^h Hourly cost of employing staff k for a regular shift.

C_k^o Overtime payment for staff k (if required)

Procedure-related input parameters:

T_p Average time required to perform procedure *p* (in minutes).

$\ddot{T}_{pq}$ Minimum required time gap if procedure *q* is to be performed after procedure *p* (in minutes).

Ω_p Revenue attached to the procedure *p* (in monetary terms).

Other parameters:

t_{ij} Total time required to travel between node *i* to node *j* (in minutes).

C^t Travel cost (monetary unit/minute)

$\bar{T}_s$ Starting time for shift *s*.

$\bar{\bar{T}}_s$ Ending time for shift *s*.

θ Mandatory break duration (in minutes).

$\bar{\vartheta}_s$ Earliest lunch break starting time for shift *s*.

$\bar{\bar{\vartheta}}_s$ Latest lunch break starting time for shift *s*.

B Sufficiently large positive number.

Decision Variables:

p_i $\begin{cases} 1, \text{ if all requested procedures of patient } i \text{ is selected for home healthcare delivery.} \\ 0, \text{ otherwise.} \end{cases}$

q_{ip} $\begin{cases} 1, \text{ if procedure } p \text{ for patient } i \text{ is selected for home healthcare delivery.} \\ 0, \text{ otherwise.} \end{cases}$

v_{ipu} $\begin{cases} 1, \text{ if } u^{th} \text{ visit for procedure } p \text{ is to be made for patient } i. \\ 0, \text{ otherwise.} \end{cases}$

a_k $\begin{cases} 1, \text{ if caregiver } k \text{ is selected to make the visits.} \\ 0, \text{ otherwise.} \end{cases}$

y_{ipuk} $\begin{cases} 1, \text{ if } u^{th} \text{ visit for procedure } p \text{ to patient } i \text{ is to be made by the caregiver } k. \\ 0, \text{ otherwise.} \end{cases}$

$x_{ipujqvk}$ $\begin{cases} 1, \text{ if for a given caregiver } k, u^{th} \text{ visit for procedure } p \text{ to node } i \text{ is scheduled just before } v^{th} \text{ visit for procedure } q \text{ to node } j, \text{ where } (j \neq i \text{ or } q \neq p \text{ or } u \neq v). \\ 0, \text{ otherwise.} \end{cases}$

z_{ipukl} $\begin{cases} 1, \text{ if } k^{th} \text{ caregiver and } l^{th} \text{ caregiver are assigned to } u^{th} \text{ visit for procedure } p \text{ of patient } i \text{ together.} \\ 0, \text{ otherwise.} \end{cases}$

η'_{ik} $\begin{cases} 1, \text{ if } k^{th} \text{ caregiver visits patient } i \text{ at least once.} \\ 0, \text{ otherwise.} \end{cases}$

η''_{kl} $\begin{cases} 1, \text{ if } k^{th} \text{ caregiver and } l^{th} \text{ caregiver have come in contact at least once.} \\ 0, \text{ otherwise.} \end{cases}$

$\xi_{ipujqvk}$ $\begin{cases} 1, \text{ if lunch break time for HHC staff } k \text{ is scheduled between } u^{th} \text{ visit for procedure } p \text{ to patient } i \text{ and } v^{th} \text{ visit for procedure to patient } j, \text{ where } (j \neq i \text{ or } q \neq p \text{ or } u \neq v). \\ 0, \text{ otherwise.} \end{cases}$

π_{ipuk} Procedure starting time for u^{th} visit for procedure p to patient i by the caregiver k.

$\ddot{\pi}_{ipuk}$ Arrival time for u^{th} visit for procedure p to patient i by the caregiver k.

$\overleftarrow{\pi}_{ipuk}$ Waiting time between arrival and starting the service for u^{th} visit of procedure p to patient i by the caregiver k.

$\hat{\pi}_k$ Break starting time for the caregiver k.

$d^+{}_k$ Positive deviation from shift length for staff k.

$d^-{}_k$ Negative deviation from shift length for staff k.

D^+ Maximum of $d^+{}_k$ values among all the staff.

D^- Maximum of $d^-{}_k$ values among all the staff.

τ_{ipu} Scheduled time for u^{th} visit of procedure p to patient i.

$\overleftarrow{\sigma}_{ipuqv}$ $\begin{cases} 1, & \text{if } u^{th} \text{ visit of procedure } p \text{ is scheduled before the } v^{th} \text{ visit of procedure } q \text{ for patient } i. \\ 0, & \text{otherwise.} \end{cases}$

$\vec{\sigma}_{ipuqv}$ $\begin{cases} 1, & \text{if } u^{th} \text{ visit of procedure } p \text{ is scheduled after the } v^{th} \text{ visit of procedure } q \text{ for patient } i. \\ 0, & \text{otherwise.} \end{cases}$

$\overleftarrow{\delta}_{ipu}$ $\begin{cases} 1, & \text{if start time for } u^{th} \text{ visit for procedure } p \text{ to patient } i \text{ is scheduled before the ITW- start time.} \\ 0, & \text{otherwise.} \end{cases}$

$\vec{\delta}_{ipu}$ $\begin{cases} 1, & \text{if finish time for } u^{th} \text{ visit for procedure } p \text{ to patient } i \text{ is scheduled after the ITW- end time.} \\ 0, & \text{otherwise.} \end{cases}$

$\overleftrightarrow{\delta}_{ipu}$ $\begin{cases} 1, & \text{if } u^{th} \text{ visit for procedure } p \text{ to patient } i \text{ is scheduled completely inside the ITW.} \\ 0, & \text{otherwise.} \end{cases}$

$\breve{\delta}_{ipu}$ $\begin{cases} 1, & \text{if ITW falls completely inside the scheduled time for } u^{th} \text{ visit for procedure } p \text{ to patient } i. \\ 0, & \text{otherwise.} \end{cases}$

f_{ipu} Overlap of service time for visit u of procedure p for patient i with patient's inconvenient time slot, if the overlap occurs only with the front of patient's inconvenient time slot.

r_{ipu} Overlap of service time for visit *u* of procedure *p* for patient *i* with patient's inconvenient time slot, if the overlap occurs only with the rear of patient's inconvenient time slot.

O_{ipu} Overlap of service time for visit *u* of procedure *p* for patient *i* with patient's inconvenient time slot.

m_{ipq} $\begin{cases} 1, \text{ if a selected procedure } p \text{ has a higher or equal visit-frequency than selected procedure } q \text{ for patient } i. \\ 0, \text{ otherwise.} \end{cases}$

$\bar{m}_{ip}$ $\begin{cases} 1, \text{ if a selected procedure } p \text{ has the highest visit-frequency than any other selected procedures for patient } i. \\ 0, \text{ otherwise.} \end{cases}$

n_{ic} $\begin{cases} 1, \text{ if any visit for patient } i \text{ is scheduled inside session } s. \\ 0, \text{ otherwise.} \end{cases}$

$\bar{n}_{ipuc}$ $\begin{cases} 1, \text{ if } u^{th} \text{ visit for procedure } p \text{ to patient } i \text{ is made in session } c. \\ 0, \text{ otherwise.} \end{cases}$

$\bar{\tau}_{ic}$ Session starting time of session *c* for patient *i*.

$\bar{\bar{\tau}}_{ic}$ Session finish time of session *c* for patient *i*.

$\overleftrightarrow{\tau}_{ic}$ Length of the session *c* for patient *i*.

Objective functions:

Objective 1: Net profit

$$(\text{MAX})\ Z_1 = \sum_{i\in M}\sum_{p\in P} \Omega_p * V_{ip} * q_{ip} - \sum_{k\in H}\sum_{i\in N}\sum_{p\in P}\sum_{u\in V}\sum_{j\in N}\sum_{q\in P}\sum_{v\in V} C^t * t_{ij} * x_{ipujqvk} - \sum_{k\in K} C_k^d * a_k - \sum_{k\in K} C_k^o * d_k^+ \quad (4.1)$$

Objective 2: Loss of employed labor

$$(\text{MIN})\ Z_2 = \sum_{k\in K}\sum_{i\in N}\sum_{p\in P}\sum_{u\in V}\sum_{j\in N}\sum_{q\in P}\sum_{v\in V} C_k^h * t_{ij} * x_{ipujqvk} + \sum_{k\in K}\sum_{i\in M}\sum_{p\in P}\sum_{u\in V} C_k^h * \hat{\pi}_{ipuk} + \sum_{k\in K} C_k^h * d_k^- + \sum_{k\in K}\sum_{i\in M}\sum_{p\in P}\sum_{u\in V} C_k^h * T_p * \left(\frac{F_{kp}'' - F_{ip}'}{F_{kp}''}\right) * y_{ipuk} \quad (4.2)$$

Objective 3: Number of fully accommodated patients

$$(\text{MAX})\ Z_3 = \sum_{i\in M} p_i \quad (4.3)$$

Objective 4: Balanced workload among staff

$$(\text{MIN})\ Z_4 = D^+ + D^- \quad (4.4)$$

Objective 5: Patient's inconvenience due to improper scheduling

$$(\text{MIN})\ Z_5 = \sum_{i\in M}\sum_{c\in C} \overleftrightarrow{\tau}_{ic} + \sum_{i\in M}\sum_{p\in P}\sum_{u\in V} O_{ipu} \quad (4.5)$$

Subject to,

Assignment constraints:

$$q_{ip} \le P'_{ip} \qquad \forall\, i \in M, p \in P \quad (4.6)$$

$$\sum_{p\in P} P'_{ip} - \sum_{p\in P} q_{ip} \le B * (1 - p_i) \qquad \forall\, i \in M \quad (4.7)$$

$$\sum_{u\in V} v_{ipu} = q_{ip} * V_{ip} \qquad \forall\, i \in M,\ p \in P \quad (4.8)$$

$$v_{ip0} = 0 \qquad \forall\, i \in M, p \in P \qquad (4.9)$$

$$v_{ipu} - v_{ipu-1} \leq 0 \qquad \forall\, i \in M, p \in P, u \in V \backslash \{0,1\} \qquad (4.10)$$

$$\sum_{k \in K} y_{ipuk} = v_{ipu} * R_{ip} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.11)$$

$$y_{ipuk} \leq a_k \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (4.12)$$

$$F'_{ip} - F''_{kp} \leq B * (1 - y_{ipuk}) \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (4.13)$$

$$y_{ipuk} \leq \sum_{g \in G} \left(G'_{ig} * G''_{kg} \right) \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (4.14)$$

$$y_{ipuk} \leq \sum_{e \in L} (L'_{ie} * L''_{ke}) \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (4.15)$$

Limited contact constraints:

$$\sum_{p \in P} \sum_{u \in V} y_{ipuk} \leq \mathrm{B} * \eta'_{ik} \qquad \forall\, i \in M, k \in K \qquad (4.16)$$

$$\left(y_{ipuk} + y_{ipul} \right) - z_{ipukl} \leq 1 \quad \forall i \in M, p \in P, u \in V, (k,l) \in K, k < l \qquad (4.17)$$

$$\sum_{i \in M} \sum_{p \in P} \sum_{u \in V} z_{ipukl} \leq \mathrm{B} * \eta''_{kl} \qquad \forall\, k, l \in K, k < l \qquad (4.18)$$

$$\sum_{k \in K} \eta'_{ik} \leq \lambda' \qquad \forall\, i \in M \qquad (4.19)$$

$$\sum_{i \in M} \eta'_{ik} + \sum_{l \in K} \eta''_{kl} \leq \lambda'' \qquad \forall\, k \in K \qquad (4.20)$$

Routing constraints:

$$\sum_{p=0}\sum_{u=0}\sum_{j\in M}\sum_{q\in P}\sum_{v\in V} x_{hpujqvk} = a_k * H''_{kh} \qquad \forall\, k \in K, h \in H \qquad (4.21)$$

$$\sum_{i\in M}\sum_{p\in P}\sum_{u\in V}\sum_{q=0}\sum_{v=0} x_{ipuhqvk} = a_k * H''_{kh} \qquad \forall\, k \in K,\ h \in H \qquad (4.22)$$

$$\sum_{j\in N}\sum_{q\in P}\sum_{v\in V} x_{ipujqvk} = y_{ipuk} \qquad \forall\, i \in M, p \in P,\ u \in V, k \in K \qquad (4.23)$$

$$\sum_{j\in N}\sum_{q\in P}\sum_{v\in V} x_{jqvipuk} = y_{ipuk} \qquad \forall\, i \in M, p \in P,\ u \in V, k \in K \qquad (4.24)$$

$$x_{hpu\hat{h}qvk} = 0 \qquad \forall (h, \hat{h}) \in H, (p, q) \in P, (u, v) \in V, k \in K \qquad (4.25)$$

$$\pi_{ipuk} + T_p + t_{ij} + \left(\theta * \xi_{ipujqvk}\right) + \overleftarrow{\pi}_{jqvk} - \pi_{jqvk} \leq \mathrm{B} * \left(1 - x_{ipujqvk}\right)$$

$$\forall i \in N, j \in M, (p, q) \in P, (u, v) \in V, k \in K \qquad (4.26)$$

Break and workload constraints:

$$\bar{T}_s - \pi_{h00k} \leq \mathrm{B} * (1 - a_k * H''_{kh} * S''_{ks}) \qquad \forall\, k \in K, h \in H, s \in S \qquad (4.27)$$

$$\xi_{ipujqvk} \leq x_{ipujqvk} \qquad \forall (i, j) \in N, (p, q) \in P, (u, v) \in V, k \in K \qquad (4.28)$$

$$\sum_{i\in N}\sum_{p\in P}\sum_{u\in V}\sum_{j\in N}\sum_{q\in P}\sum_{v\in V} \xi_{ipujqvk} = a_k \qquad \forall\, k \in K \qquad (4.29)$$

$$\left(\pi_{ipuk} + T_p + t_{ij} - \hat{\pi}_k\right) * \xi_{ipujqvk} = 0$$

$$\forall\, (i, j) \in N, (p, q) \in P, (u, v) \in V, k \in K \qquad (4.30)$$

$$\hat{\pi}_k - \bar{\bar{\vartheta}}_s * S''_{ks} \leq 0 \qquad \forall\, k \in K, s \in S \qquad (4.31)$$

$$\bar{\vartheta}_s * S''_{ks} - \hat{\pi}_k \leq 0 \qquad \forall\, k \in K, s \in S \qquad (4.32)$$

$$\sum_{i\in N}\sum_{p\in P}\sum_{u\in V}\sum_{j\in N}\sum_{q\in P}\sum_{v\in V} t_{ij} * x_{ipujqvk} + \sum_{i\in M}\sum_{p\in P}\sum_{u\in V}\left(\overleftarrow{\pi}_{ipuk} + T_p\right) * y_{ipuk}$$

$$+ (\theta * a_k) - d_k^+ + d_k^-$$

$$= \sum_{s\in S}\left(\bar{\bar{T}}_s - \bar{T}_s\right) * S''_{ks} * a_k \qquad \forall\, k \in K \qquad (4.33)$$

$$d_k^+ \le W_k * a_k \qquad \forall\, k \in K \qquad (4.34)$$

$$d_k^+ \le D^+ \qquad \forall\, k \in K \qquad (4.35)$$

$$d_k^- \le D^- \qquad \forall\, k \in K \qquad (4.36)$$

Scheduling constraints:

$$\left(\pi_{ipuk} - \tau_{ipu}\right) * y_{ipuk} = 0 \qquad \forall\, i \in M, p \in P,\, u \in V, k \in K \qquad (4.37)$$

$$\left(\ddot{\pi}_{ipuk} + \overleftarrow{\pi}_{ipuk} - \pi_{ipuk}\right) * y_{ipuk} = 0 \qquad \forall\, i \in M, p \in P,\, u \in V, k \in K \qquad (4.38)$$

$$\tau_{ipu} \le \mathrm{B} * v_{ipu} \qquad \forall\, i \in M, p \in P,\, u \in V \qquad (4.39)$$

$$\left(\tau_{ip(u-1)} + \tilde{T}_{ip}\right) - \tau_{ipu} \le \mathrm{B} * \left(1 - v_{ipu}\right)$$

$$\forall\, i \in M,\, p \in P, u \in V\backslash\{0,1\} \qquad (4.40)$$

$$\tau_{ipu} + T_p + \overleftarrow{T}_{pq} - \tau_{iqv} \le \mathrm{B} * \left(1 - \overleftarrow{\sigma}_{ipuqv}\right)$$

$$\forall\, i \in M, (p,q) \in P, (u,v) \in V \qquad (4.41)$$

$$\tau_{iqv} + T_q - \tau_{ipu} \le \mathrm{B} * \left(1 - \vec{\sigma}_{ipuqv}\right) \qquad \forall\, i \in M, (p,q) \in P, (u,v) \in V \qquad (4.42)$$

$$\overleftarrow{\sigma}_{ipuqv} + \vec{\sigma}_{ipuqv} = v_{ipu} * v_{iqv}$$

$$\forall\, i \in M, (p,q) \in P, (u,v) \in V, (p \ne q \text{ or } u \ne v) \qquad (4.43)$$

Makespan constraints:

$$(V_{ip} * q_{ip}) - (V_{ir} * q_{ir}) + 1 \leq B * m_{ipr} \qquad \forall\, i \in M, (p, r) \in P \qquad (4.44)$$

$$\sum_{r \in P} m_{ipr} - |P| + 1 \leq \bar{m}_{ip} \qquad \forall\, i \in M, p \in P \qquad (4.45)$$

$$\sum_{c \in C} n_{ic} - (V_{ip} * q_{ip}) \leq B * (1 - \bar{m}_{ip}) \qquad \forall\, i \in M, p \in P \qquad (4.46)$$

$$n_{i(c-1)} \geq n_{ic} \qquad \forall\, i \in M, c \in C \backslash \{0\} \qquad (4.47)$$

$$\sum_{c \in C} \bar{n}_{ipuc} = v_{ipu} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.48)$$

$$\sum_{p \in P} \sum_{u \in V} \bar{n}_{ipuc} \leq B * n_{ic} \qquad \forall\, i \in M, c \in C \qquad (4.49)$$

$$\bar{\tau}_{ic} + \bar{\bar{\tau}}_{ic} \leq B * n_{ic} \qquad \forall\, i \in M, c \in C \qquad (4.50)$$

$$\bar{\tau}_{ic} - \tau_{ipu} \leq B * \left(1 - \bar{n}_{ipuc}\right) \qquad \forall\, i \in M, p \in P, u \in V, c \in C \qquad (4.51)$$

$$\tau_{ipu} + T_p - \bar{\bar{\tau}}_{ic} \leq B * \left(1 - \bar{n}_{ipuc}\right) \qquad \forall\, i \in M, p \in P, u \in V, c \in C \qquad (4.52)$$

$$\overleftrightarrow{\tau}_{ic} = \bar{\bar{\tau}}_{ic} - \bar{\tau}_{ic} \qquad \forall\, i \in M, c \in C \qquad (4.53)$$

Inconvenient timeslot constraints:

$$\tau_{ipu} - \bar{T}_i' \leq \mathrm{B} * \left(1 - \overleftarrow{\delta}_{ipu}\right) \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.54)$$

$$\bar{\bar{T}}_i' - \tau_{ipu} - T_p \leq \mathrm{B} * \left(1 - \vec{\delta}_{ipu}\right) \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.55)$$

$$\overleftarrow{\delta}_{ipu} + \vec{\delta}_{ipu} + \check{\delta}_{ipu} - \overleftrightarrow{\delta}_{ipu} = v_{ipu} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.56)$$

$$\check{\delta}_{ipu} + \overleftrightarrow{\delta}_{ipu} \leq v_{ipu} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.57)$$

$$\left(\tau_{ipu} + T_p - \bar{T}_i'\right) - \mathrm{B} * \left(1 - \check{\delta}_{ipu}\right) \leq f_{ipu}$$

$$\forall\, i \in M, p \in P, u \in V \qquad (4.58)$$

$$(\bar{\bar{T}}_i' - \tau_{ipu}) - \mathrm{B} * \left(1 - \vec{\delta}_{ipu}\right) \leq r_{ipu} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.59)$$

$$f_{ipu} + r_{ipu} + \left(\check{\delta}_{ipu} - \overleftrightarrow{\delta}_{ipu}\right) * T_p \leq O_{ipu} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.60)$$

$p_i \in \{0,1\}$ $\forall\, i \in M$ (4.61)

$q_{ip} \in \{0,1\}$ $\forall\, i \in M, p \in P$ (4.62)

$v_{ipu} \in \{0,1\}$ $\forall\, i \in M, p \in P, u \in V$ (4.63)

$a_k \in \{0,1\}$ $\forall\, k \in K$ (4.64)

$y_{ipuk} \in \{0,1\}$ $\forall\, i \in M, p \in P, u \in V, k \in K$ (4.65)

$x_{ipujqvk} \in \{0,1\}$ $\forall\, (i,j) \in N, (p,q) \in P, (u,v) \in V, k \in K$, (j≠i or q≠p or u≠v) (4.66)

$z_{ipukl} \in \{0,1\}$ $\forall\, i \in M, p \in P, u \in V, (k,l) \in K$ (4.67)

$\eta'_{ik} \in \{0,1\}$ $\forall\, i \in M, k \in K$ (4.68)

$\eta''_{kl} \in \{0,1\}$ $\forall\, (k,l) \in K$ (4.69)

$\tau_{ipu} \geq 0$ $\forall\, i \in M, p \in P, u \in V$ (4.70)

$\pi_{ipuk} \geq 0$ $\forall\, i \in N, p \in P, u \in V, k \in K$ (4.71)

$\ddot{\pi}_{ipuk} \geq 0$ $\forall\, i \in M, p \in P, u \in V, k \in K$ (4.72)

$\tilde{\pi}_{ipuk} \geq 0$ $\forall\, i \in M, p \in P, u \in V, k \in K$ (4.73)

$\hat{\pi}_k \geq 0$ $\forall k \in K$ (4.74)

$\xi_{ipujqvk} \in \{0,1\}$ $\forall\, (i,j) \in N, (p,q) \in P, (u,v) \in V, k \in K$, (j≠i or q≠p or u≠v) (4.75)

$\check{\sigma}_{ipuqv} \in \{0,1\}$ $\forall\, i \in M, (p,q) \in P, (u,v) \in V$ (4.76)

$$\vec{\sigma}_{ipuqv} \in \{0,1\} \qquad \forall\, i \in M, (p,q) \in P, (u,v) \in V \qquad (4.77)$$

$$d^{+}_{k} \geq 0 \qquad \forall k \in K \qquad (4.78)$$

$$d^{-}_{k} \geq 0 \qquad \forall k \in K \qquad (4.79)$$

$$D^{+} \geq 0 \qquad (4.80)$$

$$D^{-} \geq 0 \qquad (4.81)$$

$$\overleftarrow{\delta}_{ipu} \in \{0,1\} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.82)$$

$$\vec{\delta}_{ipu} \in \{0,1\} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.83)$$

$$\overleftrightarrow{\delta}_{ipu} \in \{0,1\} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.84)$$

$$\breve{\delta}_{ipu} \in \{0,1\} \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.85)$$

$$f_{ipu} \geq 0 \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.86)$$

$$r_{ipu} \geq 0 \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.87)$$

$$O_{ipu} \geq 0 \qquad \forall\, i \in M, p \in P, u \in V \qquad (4.88)$$

$$m_{ipr} \in \{0,1\} \qquad \forall\, i \in M, (p,r) \in P \qquad (4.89)$$

$$\bar{m}_{ip} \in \{0,1\} \qquad \forall\, i \in M, p \in P \qquad (4.90)$$

$$n_{ic} \in \{0,1\} \qquad \forall\, i \in M, c \in C \qquad (4.91)$$

$$\bar{n}_{ipus} \in \{0,1\} \qquad \forall\, i \in M, p \in P, u \in V, c \in C \qquad (4.92)$$

$$\bar{\tau}_{ic} \geq 0 \qquad \forall\, i \in M, c \in C \qquad (4.93)$$

$$\bar{\bar{\tau}}_{ic} \geq 0 \qquad \forall\, i \in M, c \in C \qquad (4.94)$$

$$\overleftrightarrow{\tau}_{ic} \geq 0 \qquad \forall\, i \in M, c \in C \qquad (4.95)$$

Objective (4.1) includes the total inflow and outflow of the money to calculate the net profit that is to be maximized. Daily wage, overtime payment, and cost of travel are calculated and subtracted from the total generated revenue to calculate the net profit. To further increase the utilization of already employed

skilled labor, Objective (4.2) calculated the total time assigned to each staff for performing non-technical works such as travel and waiting. The first two components of Objective (4.2) are used to calculate the cumulative travel and waiting time for all the active staff. There may be a case where the total workload assigned to an active staff is less than their shift length. In that case, a negative deviation from the shift length is calculated as the third component for Objective (4.2). Finally, the extent to which a task assigned to a caregiver utilizes the capability of the caregiver is calculated as the fourth component. With the decrease in the gap between the skill level of the caregiver and the required skill level for a task, this component decreases to zero. It is worth noting that although the components in Objective (4.2) are considered in monetary terms, these costs represent an – "opportunity cost" or indirect cost. Therefore, they are not integrated with Objective (4.1) where the profit is defined as the net cash flows.

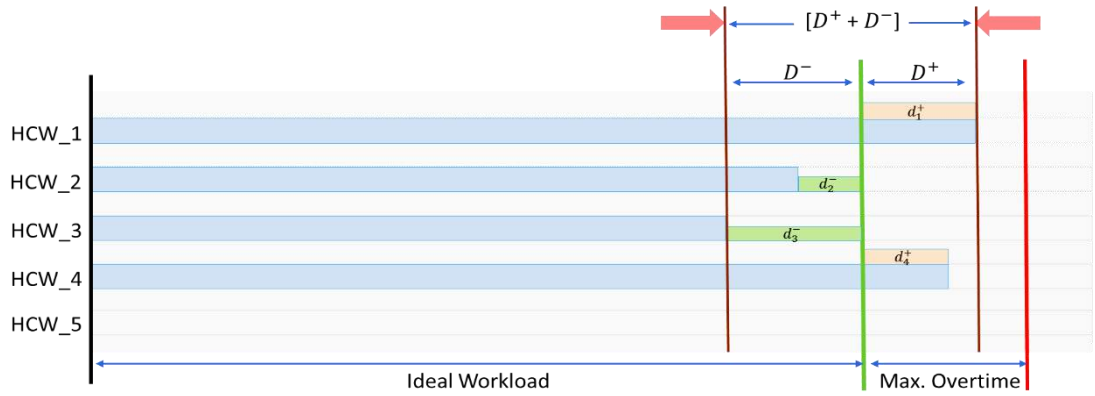


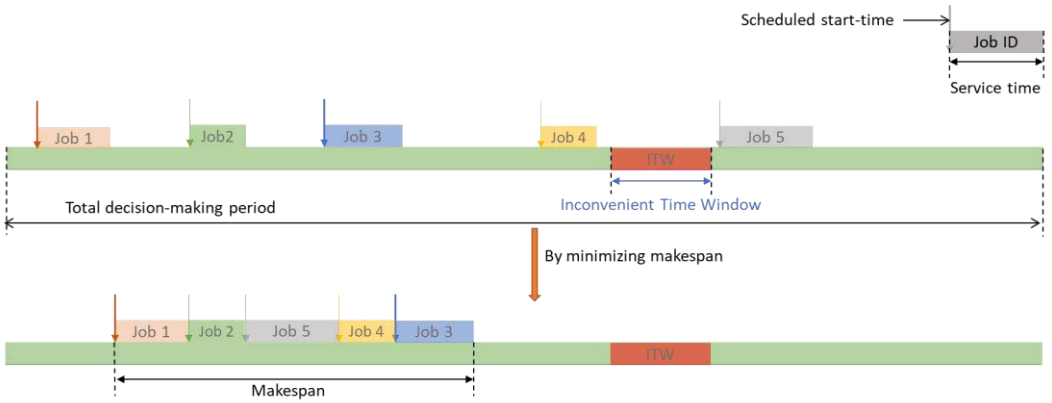
Figure 4.2: Workload assignment for healthcare workers.

Next, Objective (4.3) calculates the number of fully served patients and maximizes it under the policy of partial accommodation as explained in Chapter 3. Objective (4.4) optimizes the workload balance among the active staff. As seen in **Figure 4.2**, positive as well as negative deviation from the ideal workload is

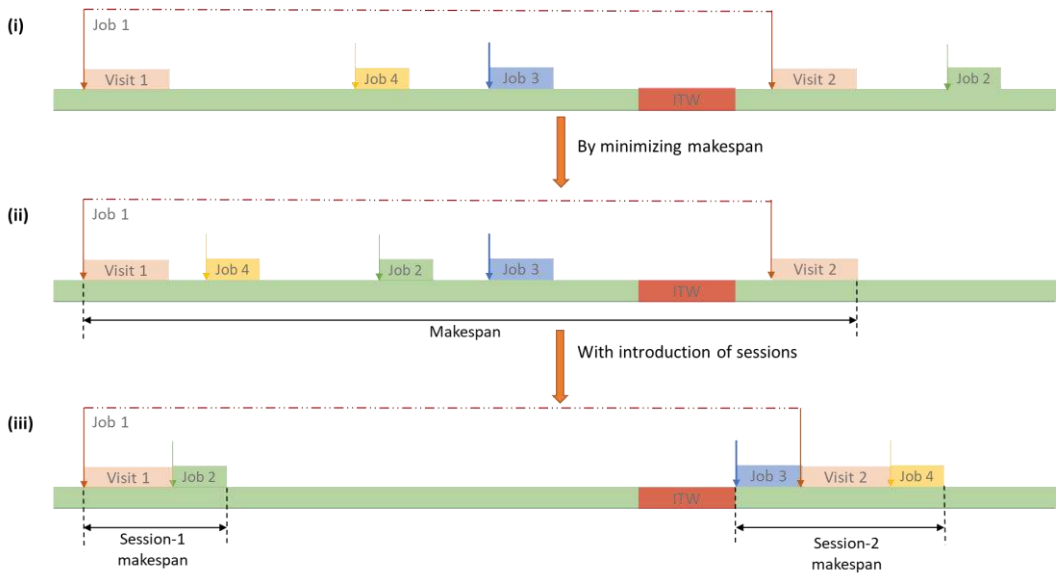
calculated for each staff, whichever is applicable. The objective function is used to minimize the sum of maximum negative and maximum positive deviation from the ideal workload. Hence finding a more favorable workload distribution for the caregivers.

Finally, Objective (4.5) is used to calculate the cumulative patient inconvenience due to improper scheduling. The first component of the objective function is used to reduce the unnecessarily spread-out visit schedule. If possible, a patient should not be disturbed multiple times during the day. This can be easily achieved by minimizing the makespan, where the makespan is defined as the difference between the start time of the first scheduled visit and the finish time of the last scheduled visit, as shown in **Figure 4.3(a)**. However, with the necessary requirement of scheduling the necessary time gap between the consecutive visits of a procedure, this strategy will have no effect on the procedures scheduled between the said interrelated visits **Figure 4.3(b)**. As the time gap between related visits cannot be reduced beyond a point, the introduction of separate sessions for each visit of the selected procedure with the highest visit- frequency becomes a necessity. **Figure 4.3(c)** shows that the problem will persist with fewer number of sessions than the prescribed count. Two possible schedules for a set of the selected requests, which contains a job that needs three separate visits, have been presented. In schedule (ii), where only two sessions are used, it can be seen that at least two visits of job three will be assigned to one of the sessions. In this case, some of the jobs can be improperly scheduled within these visits without affecting the total makespan. The proposed model identifies the appropriate number of sessions for each patient and minimizes the cumulative makespan of the sessions as the first component of the objective. The second component of the objective function

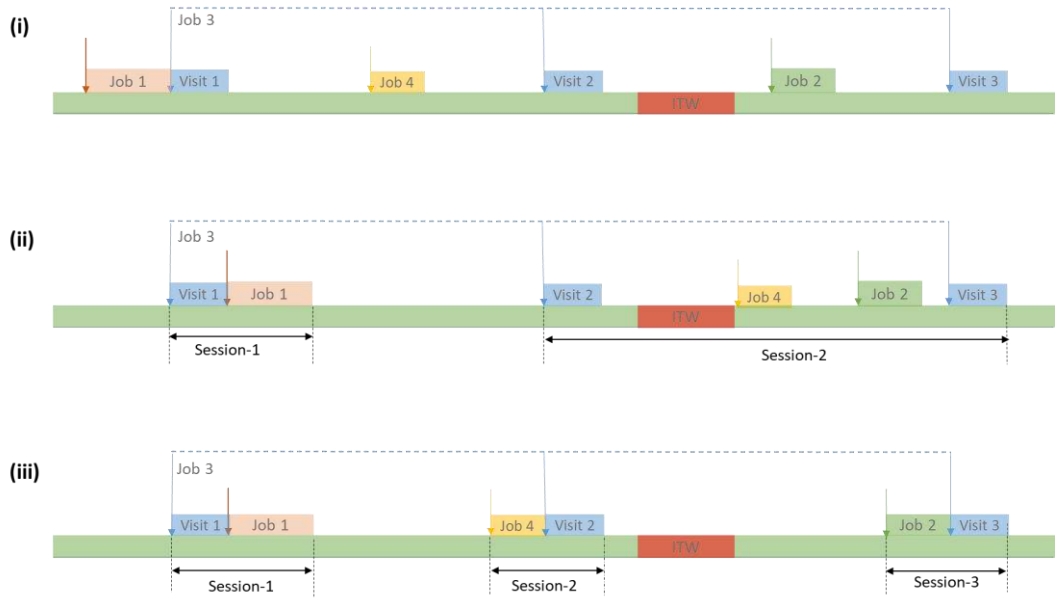
minimizes the sum of the total overlap of all the scheduled visits with the patient's inconvenient time windows. The model considers the two possible cases regarding the length of service time and the inconvenient time window, whether the service time for the procedure is greater or smaller than the patient's specified inconvenient time windows. Based on the identified case, the model is capable of calculating the accurate overlap using the forward (f_{ipu}) and/or rear overlap (r_{ipu}) along with the necessary correction. Needs and the method of calculating these values are more clearly explained alongside the relevant constraint definitions.



(a)



(b)



(c)

Figure 4.3: Possible schedules for various cases of multiple visits to a single patient.

Constraints (4.6) to (4.15) are categorized as assignment constraints. Constraint (4.6) ensures that the only requested procedures get selected for the home delivery. Constraint (4.7) identifies the patients with all the requests selected to be fulfilled. Constraints (4.8) to (4.10) are used to initiate the required number of visits for all the selected procedure requests. Constraint (4.11) assigns the required caregiver for each visit. Restrictions related to skill level are handled with Constraints (4.12) and (4.13) by ensuring that visits are only assigned to an on-duty staff with a skill level equal to or greater than the level required for the patient-procedure combination. Constraints (4.14) and (4.15) ensure the gender and language compatibility of patient-caregiver assignments. Constraints (4.16) to (4.20) are used to implement the 'limited contact' restriction during an outbreak of a disease. Constraint (4.16) keeps the count of caregivers coming in contact with the patients and vice versa. It is important to note that the contact between a specific patient-caregiver combination is only counted once, irrespective of the frequency

of their interaction. Similarly, Constraints (4.17) and (4.18) only count the unique interactions between two caregivers. Similar to patient-caregiver interaction, the possible repeated contacts between two specific caregivers are also only added once to the count. Finally, Constraints (4.19) and (4.20) keep the maximum number of interactions for an individual below their prescribed limits. The next group of constraints is used to ensure the feasible routing of caregivers. Constraints (4.21) and (4.22) start and end the route for a staff at the same node, the assigned hub. Constraints (4.23) and (4.24) act as the standard flow conservation constraints for the routes. Constraints (4.25) and (4.26) remove the possibility of movement between the hubs and the chances of sub-tour formation, respectively. Constraint (4.27) synchronizes the start of the caregiver's shift with the departure from their assigned hub. Constraints (4.28) to (4.30) identify exactly one link from the caregiver's route where the lunch break will be scheduled. Ultimately, Constraints (4.31) and (4.32) place the start time for the lunch break between the earliest and latest lunch break starting time associated with the caregiver's shift. Constraint (4.33) calculates the positive and negative deviations of the allocated workload from the contractual load for an available caregiver. It should be noted that the deviations will not be counted if the caregiver is not selected for service. Constraint (4.34) restricts the positive deviation (d_k^+) for staff within the maximum allowed overload. Constraints (4.35) and (4.36) find the maximum positive and negative deviation values for all the active staff, respectively. Constraint (4.37) synchronizes the procedure starting time for each staff assigned to a visit with the scheduled time of that visit. Similarly, Constraint (4.38) calculates the arrival time of a staff using the procedure starting time and the allotted waiting time. Constraints (4.39) and (4.40) schedule the repeated visits of a procedure (for procedures requiring multiple

visits) with an appropriate gap. Constraints (4.41) to (4.43) take care of the necessary disjunction between the visits (if applicable based on the procedures) and simultaneously eliminate the chances of assigning the same time for the two separate visits of the same patient.

As explained earlier, the makespan constraints (4.44) to (4.53) are used to provide conveniently scheduled visits for the patients. These constraints are used to club assignments as much as possible in order to avoid multiple scattered visits throughout the day. Constraints (4.44) and (4.45) are used to find the procedure with the maximum visit frequency among the selected procedures for a particular patient. The model uses the visit frequency of this procedure to determine the appropriate number of sessions to be activated for each patient. Constraints (4.46) to (4.49) assign every scheduled visit of a patient to one of the active sessions. Finally, Constraints (4.50) to (4.53) separately calculate the length of the active sessions, the summation of which is ultimately to be minimized using Objective (4.5). Finally, Constraints (4.54) to (4.60) are used to calculate the violation of the patient-specified inconvenient time window.

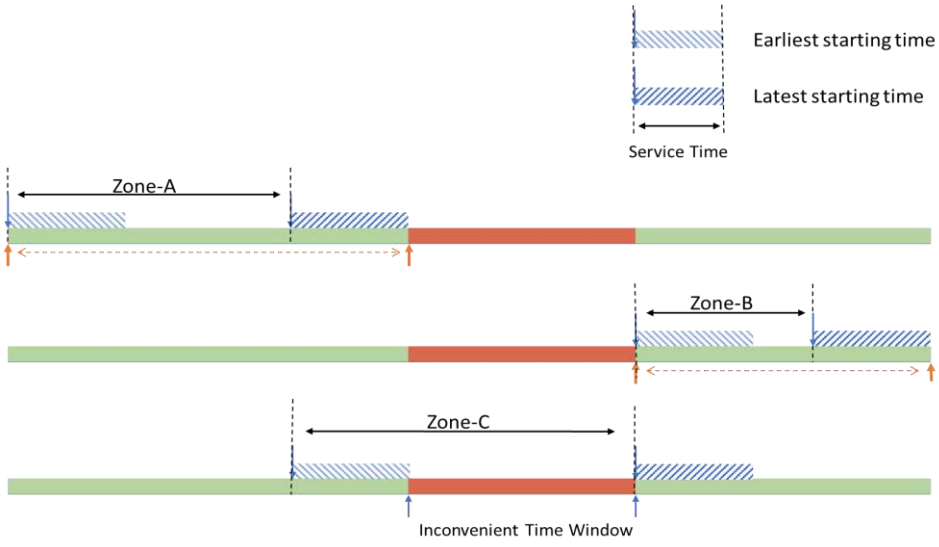
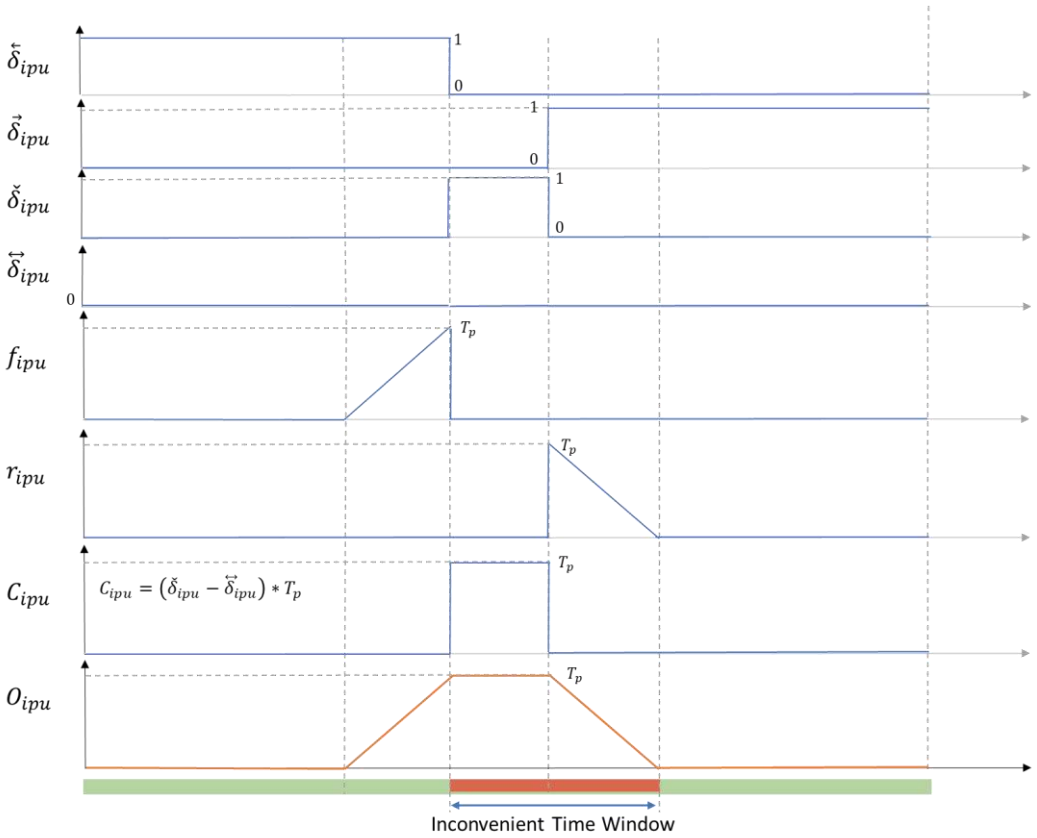
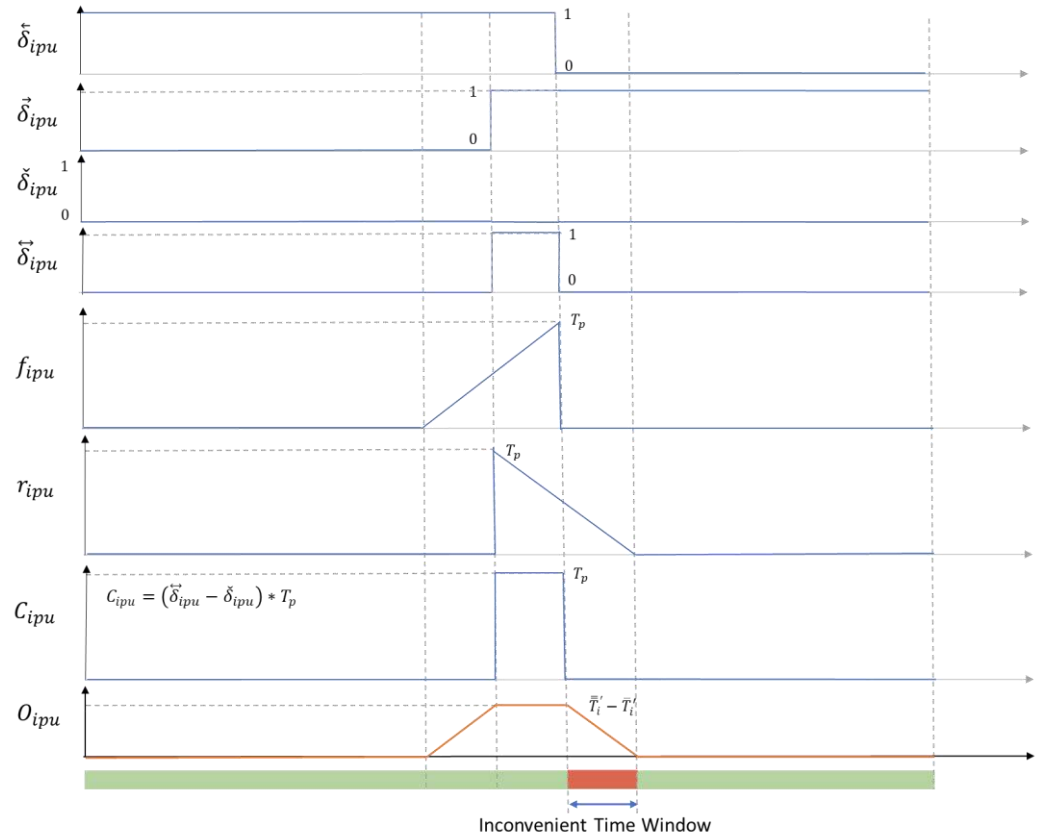


Figure 4.4: Classification of time segments in relation to the ITW.



(a) Case I – service time less than inconvenient time slot



(b) Case II– service time greater than or equal to than inconvenient time slot

Figure 4.5: Values of decision variables related to the ITW constraints.

In order to accurately find the individual overlap of each scheduled visit with the inconvenient time slot, the model needs to consider the position of the start time of the visit in relation to the inconvenient time window. Intuitively, when a visit starts in zone A or B, as shown in **Figure 4.4**, there will be no overlap with the inconvenient time window. The overlap occurs only when the start time of the visits is scheduled in zone C. In addition to this, the amount of the overlap will also depend upon two cases – whether the service time for the visit is shorter or greater than the length of the time window. The values of all the relevant decision variables for the two cases are presented in **Figure 4.5(a)** and **Figure 4.5(b),** respectively. The proposed mathematical model identifies the appropriate case and calculates the correct overlap for each scheduled visit individually. While Constraint (4.54) judges the position of the start time of a particular visit relative to the start time of the inconvenient time window, Constraint (4.55) does the same for the finish time of the visit in relation to the finish time of the inconvenient time window. Constraints (4.56) and (4.57) produce the appropriate values for four binary variables that are represented in **Figure 4.5**. Constraints (4.58) and (4.59) calculate the forward and/or rear overlap. Finally, constraint (4.60) applies the appropriate correction to calculate the overall overlap of a visit with the patient's inconvenient time window. Values of binary decision variables along with forward overlap, real overlap, overlap correction, and the net overlap have been graphed in **Figure 4.5** in relation to the scheduled start time of the visit.

4.4. An illustrative example

Table 4.1: Hub information.

Hub ID	Hub location	
	Latitude	Longitude
Hub 1	30.71166	76.83585
Hub 2	30.71166	76.83585
Hub 3	30.71166	76.83585

Table 4.2: Healthcare staff information.

Caregiver ID	Caregiver attributes				
	Hub	Shift	Language	Gender	Daily wage (Rupees)
HCW1	Hub 1	Evening	Hindi, Regional	Female	375
HCW2	Hub 2	Morning	English, Regional	Male	375
HCW3	Hub 3	Morning	English, Hindi, Regional	Male	350

Table 4.3: Patient information.

Patient's ID	Patient's Attributes					
	Location		Language Preference	Gender Preference	Inconvenient Time	
	Latitude	Longitude			Start Time	End Time
Patient1	30.7381	76.77893	English, Regional	None	None	
Patient2	30.75447	76.77188	Hindi, Regional	None	12:30	15:15
Patient3	30.7285	76.81258	English, Hindi	Male	16:00	17:45
Patient4	30.73208	76.74261	English, Hindi, Regional	Female	10:15	10:30
Patient5	30.76001	76.78355	English, Hindi	Female	None	
Patient6	30.67413	76.7938	English, Regional	None	06:15	08:00

Table 4.4: Procedure information.

Procedures ID	Service Times (minutes)	Staff Required	Revenue Per Visit (Rupees)	Procedure Requirement [Visits needed, Skill-level]	Procedure Capability [Skill-level]
P1	30	1	300	Patient4 [1, 5]	HCW1 [7], HCW2 [5], HCW3 [7]
P6	5	1	200	Patient2 [2, 4]	HCW1 [8], HCW2 [9], HCW3 [5]
P7	20	1	400	Patient2 [2, 5]	HCW1 [10], HCW2 [5], HCW3 [9]
P9	15	1	300	Patient3 [2, 3]	HCW1 [7], HCW2 [10], HCW3 [7]
P15	15	3	350	Patient1 [1, 2]	HCW1 [10], HCW2 [9], HCW3 [10]
P16	20	1	350	Patient3 [3, 3]	HCW1 [6], HCW2 [7], HCW3 [6]
P20	20	1	200	Patient6 [2, 7]	HCW1 [9], HCW2 [7], HCW3 [5]
P23	30	2	500	Patient5 [2, 1]	HCW1 [8], HCW2 [7], HCW3 [7]
P30	35	1	250	Patient6 [1, 4]	HCW1 [9]

Table 4.5: Travel distances between nodes.

Distance Matrix (minutes)									
	Hub 1	Hub 2	Hub 3	Patient1	Patient2	Patient3	Patient4	Patient5	Patient6
Hub 1	1000	1000	1000	14.3	15.7	8.3	22.4	15.0	19.5
Hub 2	1000	1000	1000	14.3	15.7	8.3	22.4	15.0	19.5
Hub 3	1000	1000	1000	14.3	15.7	8.3	22.4	15.0	19.5
Patient1	14.3	14.3	14.3	0	8.3	9.4	11.6	7.5	19.0
Patient2	15.7	15.7	15.7	6.9	0	10.9	10.5	4.6	24.1
Patient3	8.3	8.3	8.3	9.6	11.4	0	16.5	10.6	17.1
Patient4	22.4	22.4	22.4	10.3	9.1	17.0	0	11.9	23.1
Patient5	15.0	15.0	15.0	6.8	5.3	10.1	11.2	0	23.7
Patient6	19.5	19.5	19.5	21.0	23.7	17.0	23.0	23.8	0

This section presents an illustrative example using a small problem instance with three healthcare staff, six patients, and nine procedures as an example for the proposed model. GUROBI optimizer, a state-of-the-art MIP solver, is used to solve the problem instance to optimality. The optimizer is capable of handling complex mathematical models containing equality constraints, quadratic objective function, and quadratic constraints and utilizes several inbuilt methods to efficiently solve the given model. Tables containing the necessary information about the healthcare staff, patients, procedures, and hubs are also provided for easier understanding. From the start, **Table 4.1** contains the list of geographical locations for the hub nodes. **Table 4.2** provides information about the healthcare staff, such as their ID, hub, shift, language capability, gender, and daily wages. For example, HCW2 is a female staff assigned to Hub 2 for morning duty, and she can speak and understand only English and regional language. For the current example, a regular shift is 8 hours, where the first (morning) and second (evening) shifts start at 6 a.m. and 11 a.m., respectively. An additional 4 hours of overtime can also be allotted to the selected staff at the end of their shift. Specifics of other caregivers can also be read in a similar manner.

Table 4.3 contains information regarding patient attributes and preferences. For example, Patient 3 is incapable of speaking/understanding the regional language and can only be served by a male caregiver capable of speaking either English or Hindi (gender preference provided by the patient). The location for Patient 3, along with the inconvenient time slot (from 4:00 pm to 5:45 pm), is also provided. Similarly, the procedure's specifications are mentioned in **Table 4.4**. Patient's requests regarding a particular procedure are also provided with the required number of visits and minimum skill requirement for the…

Solution 1:

Table 4.6: Assignment decision for Solution 1.

Job ID	Patient-Procedure-Visit				Caregiver ID		
					HCW1	HCW2	HCW3
1	Patient1	P15	✓	Visit1	✓	✓	✓
2	Patient2	P6	✓	Visit1			✓
3				Visit2	✓		
4		P7	✓	Visit1			✓
5				Visit2	✓		
6	Patient3	P9	✓	Visit1			✓
7				Visit2			✓
8		P16	✓	Visit1			✓
9				Visit2		✓	
10				Visit3			✓
11	Patient4	P1	✓	Visit1	✓		
12	Patient5	P23		Visit1			
13				Visit2			
14	Patient6	P20	✓	Visit1		✓	
15				Visit2	✓		
16		P30	✓	Visit1	✓		

Table 4.7: Job sequence and schedule for Solution 1.

HCW1								Job 1		Job 11	Job 3	Job 5	Job 16	Job 15
HCW2				Job 14			Job 9	Job 1						
HCW3		Job 8	Job 6		Job 2	Job 4		Job 1	Job 7	Job 10				

Start time		**06:10**	**06:30**	**08:45**	**08:50**	**08:55**	**11:10**	**14:10**	**15:55**	**16:10**	**16:50**	**16:55**	**18:30**	**19:00**

Solution 2:

Table 4.8: Assignment decision for Solution 2.

Job ID	Patient-Procedure-Visit				Caregiver ID		
					HCW1	HCW2	HCW3
1	Patient1	P15		Visit1			
2	Patient2	P6	✓	Visit1			✓
3				Visit2	✓		
4		P7	✓	Visit1			✓
5				Visit2	✓		
6	Patient3	P9	✓	Visit1			✓
7				Visit2			✓
8		P16	✓	Visit1			✓
9				Visit2			✓
10				Visit3			✓
11	Patient4	P1	✓	Visit1	✓		
12	Patient5	P23		Visit1			
13				Visit2			
14	Patient6	P20	✓	Visit1	✓		
15				Visit2	✓		
16		P30	✓	Visit1	✓		

Table 4.9: Job sequence and schedule for Solution 2.

HCW1							Job 14	Job 16	Job 11			Job 3	Job 5	Job 15
HCW2														
HCW3		Job 8	Job 6	Job 4	Job 2	Job 9				Job 7	Job 10			

Start time		**06:10**	**06:30**	**08:45**	**09:05**	**11:10**	**11:20**	**11:40**	**13:40**	**15:45**	**16:10**	**17:05**	**17:10**	**19:20**

…assigned staff. Similarly, a list of healthcare staff capable of performing that procedure is also provided with their skill level. Finally, **Table 4.5** provides the time required to travel between all the nodes (hubs and patient locations). However, for the links connecting two hubs, a sufficiently high travel time value is used to avoid the movement using these links.

For comparison, we present a set of non-dominated feasible solutions to the selected problem instance. Solution 2 produces a higher profit and generally performs better or on par with Solution 1 for the majority of objectives. In contrast, Solution 1 was able to serve more requests and similarly increased the count of patients with full demand satisfaction. Hence, none of the solutions dominated the other in all the objective function values and cannot be preferred without additional input from the decision-maker. Details of both solutions are provided using suitable tables. In addition to the selection decision (whether a request is selected or not), **Table 4.6** provides information regarding the assignments of jobs to the selected caregivers for Solution 1. Visiting sequences for the assigned jobs are presented in **Table 4.7** for each healthcare staff with the scheduled start time. Similar information relating to Solution 2 is presented in **Table 4.8** and **Table 4.9**. By comparing both solutions using **Table 4.10**, the following observations can be easily made.

1. Solution 1 uses an additional staff to achieve a higher request fulfillment, which, in turn, incurs an additional operating cost, resulting in a lower profit.
2. Jobs relating to the same patient have been scheduled close to each other. Example- Job 2 and Job 4 for Patient 2, Job 6 and Job 8 for Patient 3, and Job 16 with Job 14 or Job 15 for Patient 6.

3. Under the gender preference (for female staff) provided by Patient 5, procedure P23 cannot be performed by the available staff. Because the procedure requires two healthcare staff, while only one female staff is available for assignments.

Table 4.10: Comparison of the objective function values.

Solution Number	Objective Function Value				
	Net profit (Rupees)	Loss of employed labor (Rupees)	Fully served patients	Balanced workload (Minutes)	Patient inconvenience (Minutes)
1	1963.92	1069.80	**5**	116.634	288.317
2	**2205.29**	**719.578**	4	116.634	**273.317**

4.5. Solution methodology

The proposed MIP model can only be solved optimally for very small instances by commercially available solvers, such as the GUROBI optimizer, even if only one of the objective functions is considered. For tackling a realistic example of the proposed problem, a more efficient method is required. Additionally, due to the multi-objective setting, the challenge is to generate a pool of non-dominated solutions. For this purpose, we have used a multi-objective metaheuristic (NSGA-III) which is ideal for generating multiple efficient solutions due to its population-based structure. We also have employed extensive parameter tuning based on commonly accepted metrics for assessing the quality of the Pareto front. The implementation details of these metaheuristics and the quality indicators of the Pareto front are given in the following subsections.

4.5.1. NSGA-III

Where the genetic algorithm (GA) has been very effective in the field of large-scale single-objective combinatorial optimization problems, the non-dominated sorting genetic algorithm (NSGA) has done the same for its multi-objective counterpart. As a variation of GA, NSGA utilizes some of the same operators like selection, crossover, and mutation to improve a set of randomly generated initial solutions. However, to incorporate the multi-objective nature of the selected problems, specific operators such as non-dominated fronts and sharing function was developed (Srinivas & Deb, 1994). Consistent modification to the various operators and improvement to the capability of NSGA has been observed over the years. Non-dominated Sorting Genetic Algorithm II (NSGA-II) introduces the crowding distance to replace the sharing function method to ensure the diversity in the obtained Pareto front (Deb et al., 2002). Similarly, 'fast non-dominated sorting' is used to reduce the time complexity of obtaining non-dominated fronts. Further modifications to the algorithm were proposed in NSGA-III by introducing the reference directions to ensure diversity in the eventual Pareto front. For the current work, we use the NSGA-III version with the fast non-dominated sorting and reference directions to find the best Pareto front approximation. Details of the other operators are the same as described by Deb and Jain (2014)**.**

4.5.1.1. Chromosome design and fitness evaluation

The chromosome design for the proposed NSGA-III consists of the $|K|$ routes, one for each healthcare staff. All the requested jobs (patient-procedure-visit combination) are randomly assigned to these routes. Based on the number of staff required, jobs may appear more than once. One such chromosome is exemplified

in **Figure 4.6**. An equivalent binary version for such a chromosome can be described by a two-dimensional matrix of size $|K|\times|J|$ and $|K|$ two-dimensional matrix of size $|J|\times|J|$, where J represents the set of all requested jobs. In binary representation, the first 2D matrix contains the assignment information from **Figure 4.6**, while the second set of matrices contains the information regarding their sequence. Both representations of the chromosome are identical to each other with respect to the contained information. Hence, the total length for any such chromosome can be calculated as $|K|\times|J| + |K|\times|J|\times|J|$. Crossover and repair operators, as presented in **Figure 4.6**, are used to evolve the initial set of random solutions with each iteration of NSGA-III. In addition to the Single-point crossover operator [**Figure 4.6(a)**], the performance of the NSGA-III is also tested with the Uniform crossover operator. In contrast with the Single-point operator, where offspring is created by combining the routes from the two parent solutions using a single crossover point, Uniform crossover randomly chooses one of the parent solutions for each healthcare worker's route. For both alternatives, the most appropriate mutation rate is obtained during the parameter tuning stage.

To estimate the quality of a solution, a three-stage simulation-based fitness evaluation function is proposed. Virtual agents representing the healthcare staff are used to follow the route as presented in the chromosome. Agents log the arrival, procedure start-time, and departure time for each job based on the travel and service time of the job scheduled before it. At this stage, the agent only keeps track of the caregiver's shift, break time, and additional staff requirements (if any) before logging the procedure start time. It should be noted that the start of a procedure is only logged after all the assigned staff have reached the patient's location. Following this, any job that has a finish time inside the decision-making period after

this stage can be termed as the ‘visited job’. In the second step, every interaction due to ‘visited jobs’ for each patient and caregiver is chronologically counted. For an individual, the status of the ‘visited’ jobs falling beyond the prescribed limit for allowed contacts is toggled back to ‘unvisited’. In the third stage, every job with ‘visited’ status is assigned a 10-point score and checked for the violation of remaining constraints. The benefit of this approach is that the violation of constraints remained to be considered only depending upon the individual job rather than its interaction with others. After the penalty for each violation is subtracted from the score, jobs having a perfect ten score are only considered for actual visits. Travel and service time allotted during step 1 for an ‘unvisited’ job is taken as waiting time for the next job in the sequence. The necessity of hiring a particular caregiver, total travel, waiting, and workload are calculated based on the jobs considered for actual visits. Similarly, fitness values for Objectives (4.1), (4.2), and (4.4) can be easily calculated from available information. The fitness value for Objective (4.3) represents the number of patients who have a perfect ten score for all their procedure requests. Finally, violation of the patient’s inconvenient time window is calculated using the procedure start time and service time for each scheduled job for that patient. Based on the maximum visit frequency of the assigned procedures, the largest gaps between two consecutive jobs are subtracted from the overall makespan to calculate the makespan for the appropriate number of sessions. The value of cumulative time window violation and the sum of all the session’s makespan is used as fitness for the Objective (4.5).

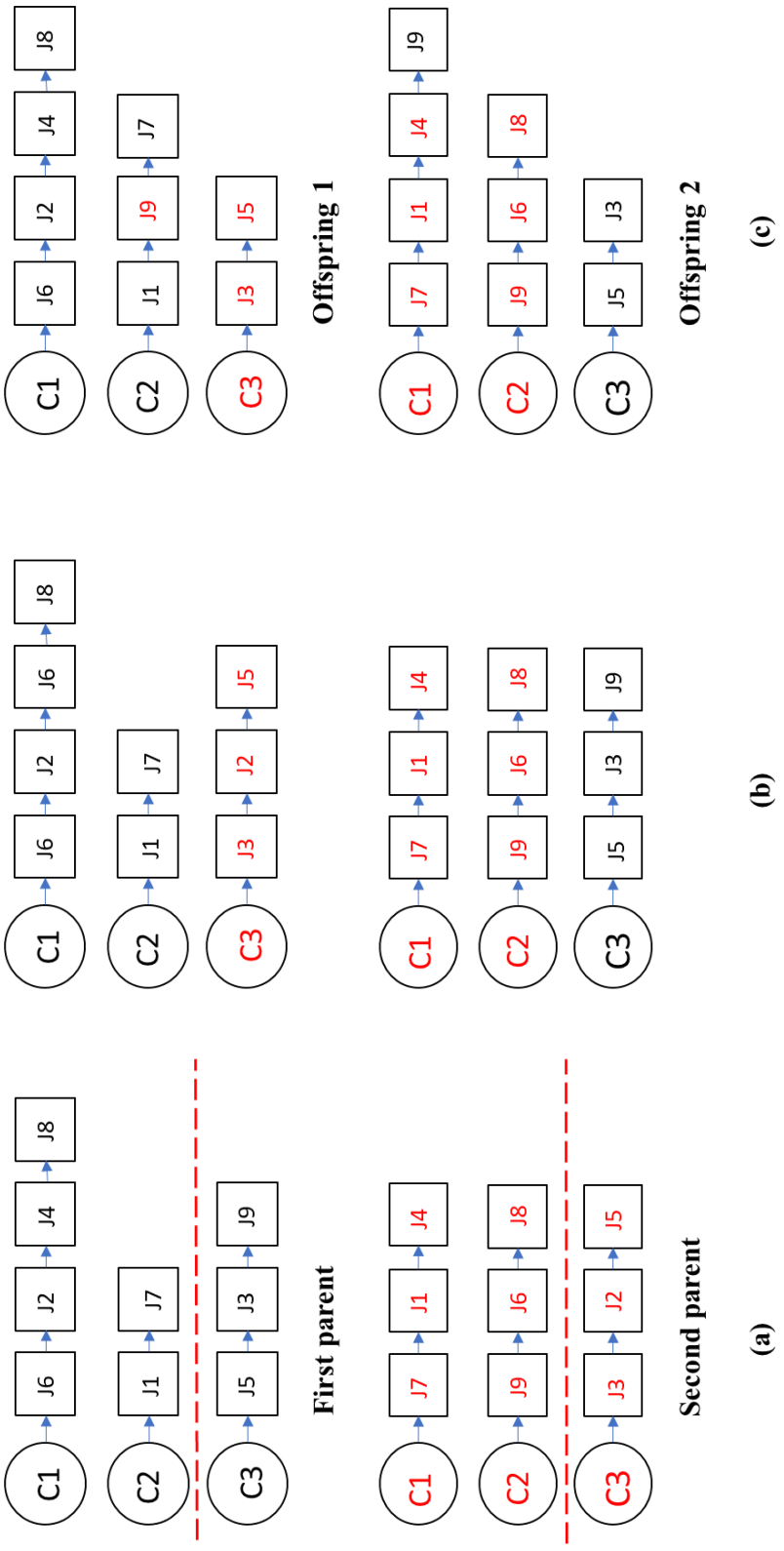


Figure 4.6: NSGA operators. (a) An example of the selection and crossover operation (b) Recombination step (c) repair operator

4.5.1.2. NSGA operations and selection of the reference directions

In addition to the common GA operators, specific NSGA operators are required to incorporate the multi-objective aspect of the problem. In NSGA-III, individuals from the current population are randomly selected to generate the offspring solutions. The algorithm employs a passive elite selection method by combining the current population with offspring. Fast non-dominated sorting (Deb et al., 2002) is used to separate the combined population in non-dominated fronts of increasing ranks. While fast non-dominated sorting ensures the movement of the population toward the Pareto front, NSGA-III uses reference directions to preserve the diversity. A set of well-spread-out points (directions) is required to find a good approximation of the Pareto front.

Das-Dennis method (Das & Dennis, 1998) is used to generate such points as it generates the points in a very structured manner. With five objectives and three partitions in the direction of each objective, a total of 35 reference points are generated as suggested by (Deb & Jain, 2014). Using the scatter-plot matrix method, a graphical representation of the reference direction in normalized objective function space is shown in **Table 4.11.** NSGA-III utilizes these directions to give preference to the solutions from a less dense region and avoid picking multiple solutions in any one direction. The algorithm is terminated at 7200 seconds (2 hours), and the 35 members from the last generation, one belonging to each reference direction, are taken as the best Pareto front approximation.

Table 4.11: Coordinates of the reference points in the normalized objective space.

Reference point	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
1	0.000	0.000	0.000	0.000	1.000
2	0.000	0.000	0.000	0.333	0.667
3	0.000	0.000	0.000	0.667	0.333
4	0.000	0.000	0.000	1.000	0.000
5	0.000	0.000	0.333	0.000	0.667
6	0.000	0.000	0.333	0.333	0.333
7	0.000	0.000	0.333	0.667	0.000
8	0.000	0.000	0.667	0.000	0.333
9	0.000	0.000	0.667	0.333	0.000
10	0.000	0.000	1.000	0.000	0.000
11	0.000	0.333	0.000	0.000	0.667
12	0.000	0.333	0.000	0.333	0.333
13	0.000	0.333	0.000	0.667	0.000
14	0.000	0.333	0.333	0.000	0.333
15	0.000	0.333	0.333	0.333	0.000
16	0.000	0.333	0.667	0.000	0.000
17	0.000	0.667	0.000	0.000	0.333
18	0.000	0.667	0.000	0.333	0.000
19	0.000	0.667	0.333	0.000	0.000
20	0.000	1.000	0.000	0.000	0.000
21	0.333	0.000	0.000	0.000	0.667
22	0.333	0.000	0.000	0.333	0.333
23	0.333	0.000	0.000	0.667	0.000
24	0.333	0.000	0.333	0.000	0.333
25	0.333	0.000	0.333	0.333	0.000
26	0.333	0.000	0.667	0.000	0.000
27	0.333	0.333	0.000	0.000	0.333
28	0.333	0.333	0.000	0.333	0.000
29	0.333	0.333	0.333	0.000	0.000
30	0.333	0.667	0.000	0.000	0.000
31	0.667	0.000	0.000	0.000	0.333
32	0.667	0.000	0.000	0.333	0.000
33	0.667	0.000	0.333	0.000	0.000
34	0.667	0.333	0.000	0.000	0.000
35	1.000	0.000	0.000	0.000	0.000

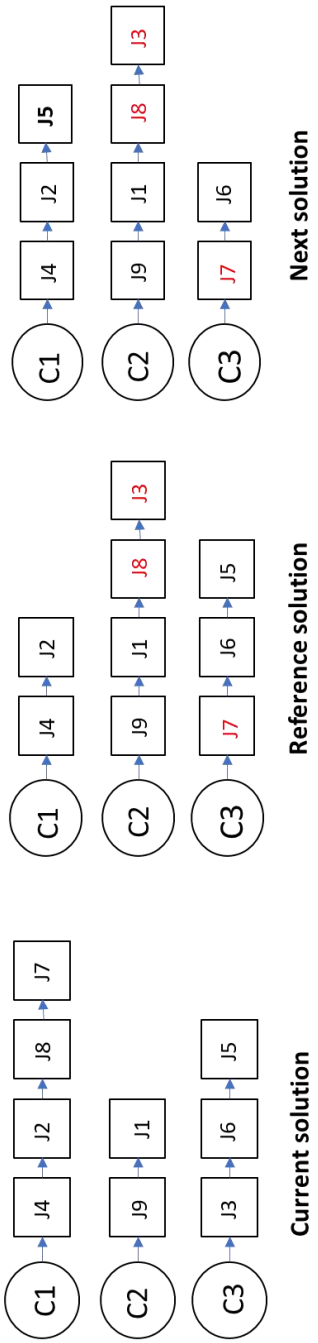
4.5.2 Multi-objective Particle Swarm Optimization

Particle Swarm Optimization (PSO), as described by Kennedy and Eberhart (1995), simulates the foraging behavior of a large bird flock to find a good quality solution in a continuous search space. The location and velocity of swarm particles are regularly updated with each iteration based on cognitive as well as social influence. The best location for a specific particle (cognitive influence) and the best global location discovered by the swarm (social influence) are tracked throughout the search procedure. The paper also emphasizes the value of an inertial component for velocity calculation. To tackle a highly discrete search space, modifications to the algorithm were proposed by Kennedy and Eberhart (1997). The Hamming distance was used to calculate the difference between two solutions, and trajectories and velocities were defined as a change in the probability that a bit of a binary string representing the solution would be one or zero. Further, Coello and Lechuga (2002) proposed the use of an explicit finite repository of non-dominated solutions (locations) for a multi-objective problem. The algorithm randomly picks a location from the repository to be used as a social influence for the calculation of the particle's velocities. While improvements to the PSO, for single as well as multi-objective cases, have been consistently proposed, we utilize the basics proposed in the above-mentioned works to shape our implementation of the metaheuristic.

4.5.2.1 Distance and Velocity

We utilize the same solution representation and fitness evaluation methods, as described for NSGA-III **[Figure 4.6],** for the MOPSO. Because our solution is not represented as a binary string, Hamming distance cannot be used to calculate the difference between two solutions. A method based on the position of the job in

a solution representation is developed to overcome this. First, the position of each job on the basis of the assigned caregiver and the job's place in the caregiver's allotted job sequence is calculated. Then, a point-of-difference (POD) matrix to represent the differences between the current solution (current position of the particle) and reference solution (personal or global best position) is generated. **Figure 4.7** presents an example of calculating the POD matrix for a set of solutions.



Job	J1	J2	J3	J4	J5	J6	J7	J8	J9
Current solution	[2,2]	[1,2]	[3,1]	[1,1]	[3,3]	[3,2]	[1,4]	[1,3]	[2,1]
Reference solution	[2,2]	[1,2]	[2,4]	[1,1]	[3,3]	[3,2]	[3,1]	[2,3]	[2,1]
Point of difference	0	0	1	0	0	0	1	1	0

Figure 4.7: Calculating the point-of-difference matrix.

The matrix is populated with zeros for jobs with the same position in both solutions and ones for otherwise. The sum of the elements of this matrix is treated as the distance between two solutions, as an exact number of changes can convert the current solution into the reference solution. This definition of the distance between two solutions is used to achieve the particle's movement in the search space.

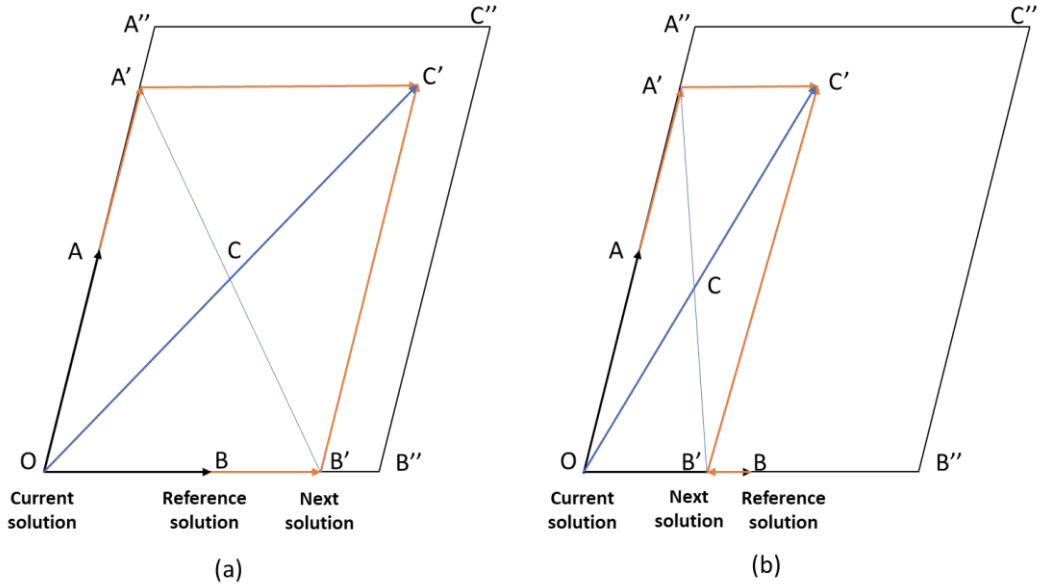


Figure 4.8: Calculating the next position of a particle using the vector sum method.

In order to find the solutions, a specific distance away from the current and the reference position of the particle, some ‘movement’ operators are developed. The movement operators utilize the information from the POD matrix to facilitate the movement of the swarm in the search space. Details of these operators are explained below in relation to the example presented in **Figure 4.8**. Based on the current solution (*O*) and the reference solutions (*A* and *B*), **Figure 4.8** shows the conventionally used vector sum approach for PSO to traverse a continuous search space to find the next location of the swarm particle. The steps needed to achieve

the same result using the developed movement operators and the method of their implementation are as follows.

Type I: Solution beyond the reference point

To find a solution *B'* on the line segment *BB'',* as shown in **Figure 4.8(a)**, the positions of |*BB'*| amount of randomly selected jobs will need to change. Additionally, solution B' should also be | *OB'*| distance away from the current solution *O*. This can be achieved by carefully changing the position of the | *BB'*| randomly selected jobs with zero POD value in the reference solution. If a solution can be found where the action of changing the position of the selected jobs does not affect the position of others, such a solution will be required distance away from the *O* and *B* and will lay on the line segment *BB''*.

Type II: Solution inside the reference point

To find the Solution |*BB'*| distance away from the reference solution and on the line segment OB [**Figure 4.8(b)**], positions of |*BB'*| randomly selected jobs, with POD value one, are exchanged by their positions in the current solution. A solution obtained by this will have exactly |*BB'*| one's changed into the zeros and will be |*BB'*| unit distance closer to the current solution.

An appropriate movement operator must be employed multiple times to find the solution on C', the next position of the particle for PSO operating in a continuous search space. First, the solution at *A'* is calculated using the correct movement operator based on the position of point *A'* in relation to the points *O* and *A*. Similarly, the solution at *C* can be found using the type-II movement operator on the solutions corresponding to points *A'* and *B'*. Finally, the Type-I movement operator is employed to find the solution at *C'* using the solution at point *C* as the

new reference solution. It is important to note here that special attention needs to be given to minimally affect the existing sequence of the jobs during the insertion of jobs in the new positions. Jobs are either placed at the vacant positions of the sequence or attached to the end of the job sequence. Additionally, there may be cases where no solution exists at the desired points, and the solution nearest the desired point is selected.

4.5.2.2 Elite repository

An explicit repository of 35 non-dominated solutions is maintained to be used as the social influence for the particle velocities. After every iteration, solutions from the elite repository and new solutions (current position of the particles) are merged, and fast non-dominated sorting is performed. Thirty-five non-dominated solutions from the best non-dominated fronts are used to replace the old solutions from the repository. The same reference directions that are used in the case of NSGA III (explained in Subsection 4.5.1.2) are used to ensure diversity in the elite repository. While calculating the next location of a particle, a solution from the repository is randomly selected to be utilized as the global best solution (social influence). Roulette wheel selection is used to ensure the exploration of the region that is sparsely represented by the solutions in the elite repository. The average distance of all the solutions from a particular solution is used to calculate the probability of selection. After the termination criteria of 7200 seconds are met, the elite repository is presented as the best Pareto front approximation. Finally, the Taguchi method is used to find the best parameter values for the MOPSO. Following the steps presented in Subsection 4.5.2, 105 particles were selected for the MOPSO with a 2.5 weightage for both social and cognitive influence over the inertia.

4.5.3 Multi-objective Grey Wolf Optimizer

Grey wolf optimizer (GWO), as presented in Mirjalili et al. (2014), simulates the hunting behavior of the grey wolf pack to find near-optimal solutions for highly combinatorial problems. From an iteration, the three best solutions are chosen to represent the location of the pack leaders. The location of Alpha, Beta, and Delta wolves (pack leaders) is then used to estimate the location of prey. The remaining wolves from the pack, named Omega, are then used to encircle and kill the prey based on their estimated position. The algorithm balances the exploration and exploitation characteristics of the search pattern by reducing the encircling radius with every iteration. Multi-objective implementation for GWO was first proposed by Mirjalili et al. (2015) by utilizing an explicit elite repository of non-dominated solutions, similar to the MOPSO metaheuristic. Leaders for the wolf pack were stochastically selected from the less dense region of the elite repository. While our specific implementation of MOGWO borrows the basics from Mirjalili et al. (2015), the method for calculating the next location for the omega wolves and selecting the location of leader wolves is novel.

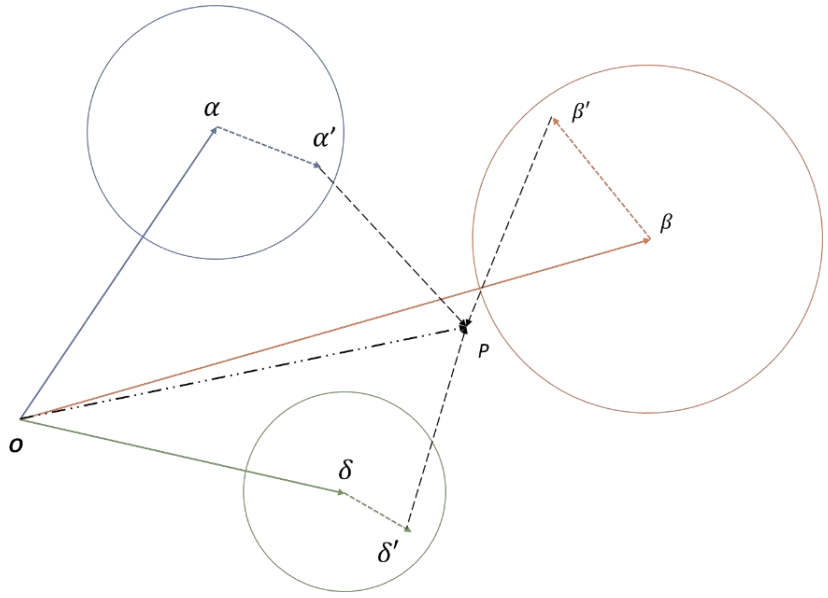


Figure 4.9: Method for estimating the location of the prey in GWO.

As explained earlier, omega wolves are used to encircle the prey (a good quality solution). Because the location of prey is not known with any certainty, the locations of leader wolves are used to estimate it. Three different estimations acquired from alpha, beta, and delta are then combined to find the most probable location of the prey. First, the distance between the omega and a leader (say alpha) is obtained by calculating the POD matrix. The distance between the omega and leader wolf, alongside the number of current iterations, is used to calculate the encircling radius. A new solution (α') inside the encircling radius is generated by altering the solution representing the leader wolf. As the distance between α and α' is the sum of the element of the POD matrix for these solutions, solution α' is generated by changing the positions of the $|\alpha\alpha'|$ jobs in the solution representing the position of the alpha wolf. Similar position estimation for omega is also generated with the beta and delta wolves. Three-position estimations are then combined to find the next position of an omega wolf **[Figure 4.9]**. Solutions representing the new wolves' positions are then combined with the elite repository, as explained in Subsection 4.5.2.2 for MOPSO. Fast non-dominated sorting and the reference directions (Subsection 4.5.1.2) are used to find new candidates for the elite repository while maintaining diversity. Leader wolves for the next iteration are stochastically picked from the less populated areas of the updated repository. Finally, As MOGWO only has one parameter that can be tuned, five runs for all three values of the parameter are carried out. The average of quality indicator values from the five run is used to find the most suitable wolf-pack size**.** With a larger value of the overall quality indicator (Y) preferable, a pack size of 70 wolves is picked for the MOGWO.

4.6. Computational Experiments and Results

Computational experiments are designed and carried out to validate the proposed metaheuristic with the established performance indicators. First, a method of experimental design is used to find the best-performing combination of various parameters for the NSGA-III. The performance of the best-performing version is then tested on instances of varying sizes. Finally, the solutions obtained for the same are also used for further analysis.

4.6.1. Experimental Settings

Problem instances developed to test the proposed mixed-integer programming model and the performance of metaheuristics is based on a real-world example. Data used to generate these problem instances are grounded in information gathered from a home healthcare service provider from Mohali, India. In its eighth year of operation, the company boasts more than 7.5 million hours of caregiving and is expecting to expand across 20+ Indian cities. Being in various stages of development in different cities, data gathered for the HHC provider offers a good basis for generating a diverse set of problem instances. Firstly, the company has provided us with a list of 40 procedures that they offer as a service. They have also made available a list of 364 different caregivers operating in Chandigarh, a city in India. The list also includes some essential attributes like hub locations and procedure capabilities for each caregiver. Then, 400 different locations are randomly picked inside Chandigarh city as a potential location for the patients. Bing Map, a service provided by Microsoft, is then used to calculate the travel distance between all the potential nodes (hub and patient's location). To generate the problem instances, the appropriate number of healthcare workers and patients are

randomly selected from their respective lists along with the mentioned attributes. Additional information like the caregiver's gender, language capability, and shift are randomly generated. With the obvious privacy concerns, the same method is also used for patients' preferences regarding staff's gender, language capability, and inconvenient time windows. With real-life core data and randomly selected attributes, the obtained problem instances can be characterized as a hybrid in nature. Additional information related to the patient, caregiver, and procedure is provided in **Table 4.12.**

Table 4.12: Description of the range and values of various input parameters.

Parameters	Description	Values / Limits
Patient's attributes	Language	English, Hindi, and Panjabi
Inconvenient time window	Length	0 - 3 hour (in an increment of 15 minutes)
Caregiver shifts	First shift	6:00 – 14:00
	Second shift	11:00 – 19:00
	Overtime	4 hours at the shift end
Caregiver's characteristics	Language	English, Hindi, and Panjabi
	Skill-level	1-10
Procedure's characteristics	Visit frequency	1, 2 or 3
	Caregivers required	1, 2 or 3
	Service time	5 - 45 minutes (in increments of 5)
	Revenue per visit	100 - 500 rupees (in increments of 50)
Number of allowed contacts	For patients	Up to 5
	For caregivers	Up to 7

Following this procedure, a set of 24 problem instances (examples) with multiple combinations for the number of staff and patients is produced. Problem sizes are systematically chosen to represent a wide variety of cases, details of which are as follows.

- Number of caregivers: 5, 10, 15, 20, and 25 for small to medium-sized examples; 50, 70, and 100 for large instances.
- Number of patients: 2, 3, and 4 times of the caregivers.

Table 4.13: Characteristics of the generated problem instances.

Configuration Number	Instance ID	Features of problem instance				
		Staff Count	Patient Count	Procedures Requested	Visits Needed	Jobs
1	HHI 5_10_40	5	10	25	42	48
2	HHI 5_15_40	5	15	47	79	102
3	HHI 5_20_40	5	18	51	87	105
4	HHI 10_20_40	10	19	53	85	115
5	HHI 10_30_40	10	29	73	125	163
6	HHI 10_40_40	10	38	96	152	218
7	HHI 15_30_40	15	26	67	107	151
8	HHI 15_45_40	15	41	118	197	257
9	HHI 15_60_40	15	54	144	229	288
10	HHI 20_40_40	20	39	102	172	232
11	HHI 20_60_40	20	56	148	238	315
12	HHI 20_80_40	20	73	201	336	452
13	**HHI 25_50_40**	**25**	**47**	**132**	**216**	**279**
14	HHI 25_75_40	25	72	196	323	449
15	HHI 25_100_40	25	91	219	339	454
16	HHI 50_100_40	50	90	245	414	550
17	HHI 50_150_40	50	139	385	629	860
18	HHI 50_200_40	50	187	506	828	1077
19	HHI 75_150_40	75	136	362	584	744
20	HHI 75_225_40	75	213	554	927	1263
21	HHI 75_300_40	75	284	759	1247	1632
22	HHI 100_200_40	100	185	478	809	1092
23	HHI 100_300_40	100	281	749	1197	1606
24	HHI 100_400_40	100	354	955	1541	2068

As seen in **Table 4.13**, the problem instances are named HHI A_B_C, where A: Number of available caregivers. B: Number of patients. C: Number of the procedures available to be chosen from. The problem instances are made available online to provide a fair opportunity to reproduce or examine our work. Access to them can be gained by following the link- https://drive.google.com/drive/folders/1P7A6fGXtGKPDmC0RPgJrFxcL3EJCcgo9?usp=sharing .

A single node consisting of 2∗ Intel Xeon SKL G-6148 CPUs (40 Cores, 2.4 GHz) is used for all the experiments. The net available computational capability for such a CPU node is only around 3.8 TFLOPS. It should be noted that despite the average capability, the specific hardware was selected for its high core counts as we follow the parallel implementation for the solution evaluation phase for all three algorithms.

4.6.2. Quality Indicators

As the proposed metaheuristics do not guarantee the exhaustive search of the solution space, the final set of non-dominated solutions only approximates the actual Pareto front. Variations in the Pareto front approximation (PFA) are to be expected with different algorithms, parameter values, and random seeds for the same algorithm. Hence, a need to establish the relative quality of PFA with each other is observed. Deb (2001) points out three aspects that need to be considered to get an accurate assessment of Pareto front approximation: closeness to the Pareto-optimal front, diversity of PFA set, and spread of PFA. Zitzler et al. (2008) present a compressive list of mathematical functions as a quality indicator for the above-mentioned aspect of PFA. It should be noted that for some quality indicators, it is assumed that the Pareto-optimal front is known. As this assumption cannot be met,

we use modified indicators where an ideal solution is used instead of the Pareto-optimal set.

4.6.2.1. Mean Ideal Distance

Mean Ideal Distance (MID) calculates the average value of the distance from the ideal point to the solutions in PFA. It behaves as the quality indicator for the 'closeness to Pareto-optimal set' and is expressed by Equations (4.96) and (4.97). An algorithm with a lower value of MID is preferred.

$$C_i = \sqrt{\left(\frac{x_i^1 - x_{ideal}^1}{x_{worst}^1 - x_{best}^1}\right)^2 + \left(\frac{x_i^2 - x_{ideal}^2}{x_{worst}^2 - x_{best}^2}\right)^2 + \dots} \qquad \forall\, i \in n \qquad (4.96)$$

$$MID = \frac{\sum_{i=1}^{n} C_i}{n} \qquad (4.97)$$

4.6.2.2. Spread of Non-Dominated Solution

As evident by the name, the Spread of Non-dominated solution (SNS) calculated the degree of spread in the PFA. First, the distance of each solution from the ideal point is calculated. The standard deviation of the distances with the mean ideal distance (MID) is taken as the indicator for the spread of the solutions. A solution set having a higher value of SNS is preferred as it uniformly represents the solution space covered by the non-dominated set.

$$\boldsymbol{SNS = \frac{\sum_{i=1}^{n} |MID - C_i|}{n-1}} \qquad (\mathbf{4.98})$$

4.6.2.3. Max-Spread

While SNS quantifies the uniform spread of the solution inside the region covered by the PFA, Max-spread (MS) is used to calculate the extent of the covered

region itself. Distances between extreme solutions, as shown in Equation (4.99), are used to calculate the MS value for a PFA. Similar to SNS, a solution set with a higher value of MS is preferred.

$$\boldsymbol{MS} = \sqrt{\left(\boldsymbol{x}^{\boldsymbol{1}}_{\boldsymbol{max}} - \boldsymbol{x}^{\boldsymbol{1}}_{\boldsymbol{min}}\right)^{\boldsymbol{2}} + \left(\boldsymbol{x}^{\boldsymbol{2}}_{\boldsymbol{max}} - \boldsymbol{x}^{\boldsymbol{2}}_{\boldsymbol{min}}\right)^{\boldsymbol{2}} + \ldots} \qquad (\mathbf{4.99})$$

Some of the studies have also used the number of solutions in PFA (Tirkolaee et al., 2022) and CPU time (Nikzamir & Baradaran, 2020) as additional indicators. We find that these are not necessary for our methodology as both values are kept the same for all three algorithms. Similarly, they are also kept constant during the parameter tuning of the metaheuristics, which is described in the following subsection.

4.6.3. Parameters tuning using the Taguchi method

To achieve the best possible implementation of proposed algorithms, several parameters are needed to be fine-tuned. Three parameters for NSGA-III and MOPSO and one for MOGWO are identified for this process. A description of these parameters with three levels of possible values for each is presented in **Table 4.14**. For NSGA-III and MOPSO, with three parameters and three levels of values, a total of 27 trials are needed to exhaust all the possible combinations. Additionally, to overcome the random variations, multiple runs for each trail are to be considered. With only five runs for each combination, 540 hours ($2 \times 27 \times 5 \times 2$) of computation resources are required for the first two algorithms alone. In contrast, MOGWO with only one parameter needs 30 hours ($3 \times 5 \times 2$).

Table 4.14: Considered parameter values for the three metaheuristics.

Algorithm	Parameters	Value for three parameter levels		
		1	2	3
NSGA-III	Population size (P_s)	70	105	140
	Crossover rate (C_r)	90%	95%	100%
	Mutation rate (M_r)	1%	2%	3%
MOPSO	Number of particles (N_p)	70	105	140
	Weight for cognitive influence (W_{ci})	1.5	2.0	2.5
	Weight for social influence (W_{si})	1.5	2.0	2.5
MOGWO	Size of the wolf-pack	70	105	140

To overcome this difficulty, a design of experiment (DOE) approach in the form of the Taguchi method is employed for the first two algorithms. The method, as proposed by Taguchi (1986), is a simple, efficient, and systematic approach to reducing the total number of trails during experimental design. By utilizing carefully developed orthogonal arrays, it reduces the required number of experiments significantly. The selection of the right orthogonal array depends on the number of parameters and the value levels for each parameter. For NSGA-III, where three levels of values are to be considered for three parameters, the L9 orthogonal array is selected. The method only requires the trials to be carried out for this combination of values only. Signal-to-noise (S/N) ratio is then calculated using Equations (4.100) and (1.101). The description of the L9 orthogonal array alongside the resulting values of quality indicators is presented in **Table 4.15** and **Table 4.16** for each Taguchi trial. A medium-size problem instance (HHI 25_50_40) is used for this part of the experiment. Quality indicator values were calculated by comparing the nine PFAs obtained under each operator separately.

On the basis of the Y-values, the Minitab software is used to calculate the S/N ratios, the means of which are presented in **Figure 4.10.**

$$Y = (-MID + SNS + MS) \quad (4.100)$$

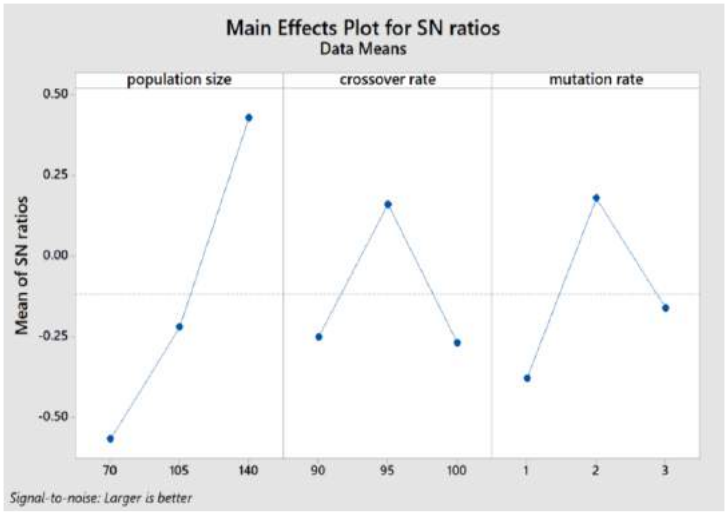
$$S/N = -10 \times log\left(\frac{\sum(1/Y^2)}{n}\right) \quad (4.101)$$

Table 4.15: L9 orthogonal array for the Taguchi trials.

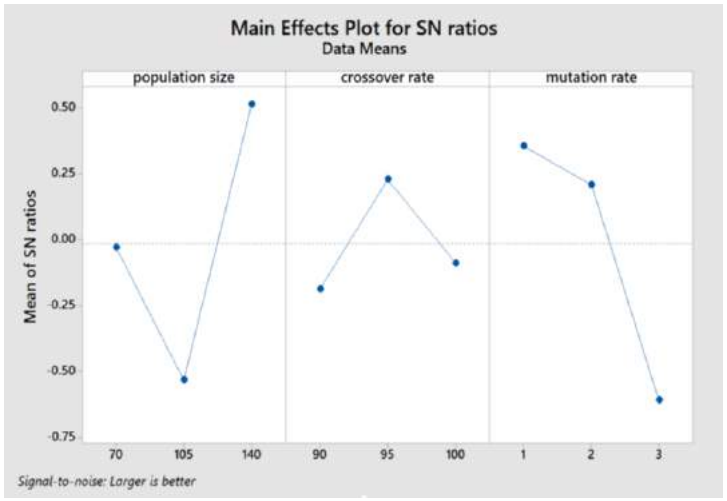
Trial number	Value of levels		
	Parameter 1	Parameter 2	Parameter 3
1	1	1	1
2	1	2	2
3	1	3	3
4	2	1	2
5	2	2	3
6	2	3	1
7	3	1	3
8	3	2	1
9	3	3	2

Table 4.16: Quality indicator values obtained using NSGA-III variants for the Taguchi trials.

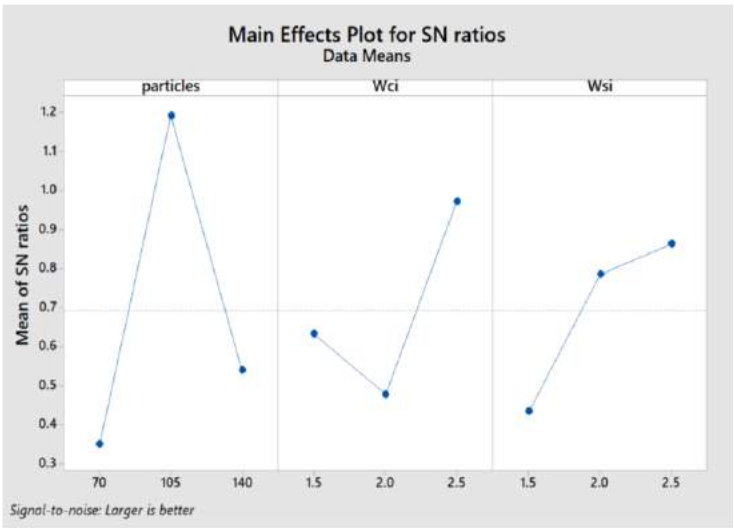
Trial number	Quality Indicators							
	Single-point crossover				Uniform crossover			
	MID	SNS	MS	Y	MID	SNS	MS	Y
1	1.028	0.186	1.702	0.860	1.022	0.195	1.812	0.985
2	1.015	**0.213**	1.861	1.059	**0.954**	0.210	1.799	1.055
3	1.008	0.209	1.702	0.903	1.016	0.173	1.795	0.953
4	0.971	0.206	1.745	0.979	1.027	0.178	1.825	0.976
5	0.965	0.174	1.752	0.962	1.009	0.179	1.703	0.873
6	**0.958**	0.194	1.748	0.984	1.028	0.196	1.808	0.977
7	0.981	0.188	**1.882**	**1.089**	0.979	0.186	1.768	0.975
8	0.987	0.197	1.827	1.038	1.011	**0.219**	**1.967**	**1.175**
9	0.966	0.212	1.780	1.026	1.007	0.177	1.872	1.043



(a)



(b)



(c)

Figure 4.10: S/N ratio plots (a) NSGA-III with Single-point crossover (b) NSGA-III with Uniform crossover (c) MOPSO

Table 4.17: Quality indicator values of the Pareto fronts obtained by MOPSO.

Trail number	Quality Indicators			
	MID	SNS	MS	Y
1	0.949	0.226	1.710	0.986
2	0.960	0.245	1.806	1.091
3	0.974	**0.269**	1.754	1.049
4	**0.953**	0.218	1.836	1.102
5	0.967	0.247	1.841	1.121
6	0.954	0.260	**1.916**	**1.222**
7	0.954	0.214	1.885	1.145
8	0.979	0.198	1.744	0.964
9	0.992	0.231	1.852	1.091

Table 4.18: Quality indicator values of the Pareto fronts obtained by MOGWO.

Trail number	Size of the wolf-pack	Quality Indicators			
		MID	SNS	MS	Y
1	70	1.106	0.186	1.977	**1.058**
2	105	0.976	0.204	1.538	0.766
3	140	1.033	0.226	1.498	0.691

By observing the graph, the best values for the population size, crossover rate, and mutation rate for the NSGA-III with a Single-point crossover operator were found to be 140, 95%, and 2%, respectively. Similarly, for the case of NSGA-III with the Uniform crossover, the best values for the three parameters were observed to be 140, 95%, and 1%, respectively. The best combination of parameter values with the resulting values of quality indicators for MOPSO are presented in **Table 4.17**. For the MOPSO, 105 particles were selected with a 2.5 weightage for both social and cognitive influence over the inertia. As MOGWO only has one parameter that can be tuned, five runs for each value of the parameter are carried out. The average of the result is presented in **Table 4.18.** With a larger value of the overall quality indicator (Y) preferable, the corresponding value for wolf-pack size is picked for the MOGWO.

Table 4.19: Comparison of the NSGA-III crossover operators using the Pareto front quality indicators.

Configuration Number	MID		SNS		MS		Y-value	
	Single-point crossover	Uniform crossover	Single-point crossover	Uniform crossover	Single-point crossover	Uniform crossover	Single-point crossover	Uniform crossover
1	1.046	1.070	0.196	0.184	1.930	1.849	1.079	0.964
2	1.072	1.064	0.177	0.171	2.010	2.056	1.114	1.163
3	1.018	0.992	0.195	0.205	1.731	1.861	0.907	1.073
4	0.892	0.995	0.188	0.147	1.481	1.460	0.777	0.612
5	0.989	0.982	0.141	0.136	1.669	1.475	0.821	0.630
6	0.813	0.905	0.192	0.193	1.657	1.638	1.035	0.926
7	0.905	0.925	0.164	0.225	1.599	1.651	0.857	0.951
8	0.994	0.997	0.132	0.155	1.652	1.718	0.791	0.876
9	0.961	0.929	0.155	0.172	1.608	1.740	0.803	0.983
10	0.903	0.932	0.181	0.177	1.649	1.767	0.927	1.013
11	0.901	0.919	0.181	0.201	1.760	1.811	1.040	1.092
12	0.962	0.950	0.179	0.156	1.737	1.670	0.954	0.876
13	0.887	0.868	0.180	0.197	1.659	1.549	0.952	0.877
14	0.953	0.924	0.162	0.190	1.711	1.701	0.920	0.967
15	0.915	0.925	0.148	0.134	1.526	1.575	0.760	0.785
16	0.894	0.950	0.168	0.145	1.552	1.725	0.825	0.919
17	0.979	1.048	0.173	0.162	1.627	1.666	0.821	0.781
18	1.157	1.233	0.148	0.167	1.784	1.954	0.775	0.888
19	0.926	1.005	0.134	0.163	1.447	1.686	0.655	0.844
20	0.975	1.029	0.169	0.135	1.629	1.577	0.823	0.683
21	1.071	1.170	0.127	0.130	1.479	1.716	0.535	0.675
22	0.823	0.914	0.114	0.118	1.384	1.556	0.675	0.761
23	0.941	0.993	0.103	0.103	1.475	1.570	0.638	0.681
24	1.045	1.100	0.124	0.109	1.359	1.596	0.438	0.604

Table 4.20: Result of paired sample *t*-test for the two crossover operators of NSGA-III.

Quality indicators	*t*-value	*p*-value	$\alpha = 0.05$	Winner
MID	3.692	**0.001**	**Yes**	Single-point crossover
SNS	0.395	0.697	No	None
MS	2.564	**0.017**	**Yes**	Uniform crossover
Y-value	1.239	0.227	No	None

Further, a detailed comparison of the two crossover operators is then carried out using their best respective parameter values. The average quality indicator values for the PFAs obtained for various problem instances using these versions of NSGA-III are presented in **Table 4.19.** Both versions of the NSGA-III are also compared using the paired sample t-test, and the result is presented in **Table 4.20**. Results show that for the MID indicator, with $\alpha = 0.05$, there is a significant difference between the performance of the two versions of NSGA-III. A similar conclusion can also be made for the MS indicator. However, by observing the *t*-values, no clear winner can be declared between these crossover operators. When only considering the MID indicator where a lower score is preferred, the positive *t*-value concludes the superiority of the Single-point crossover operator over the Uniform crossover. In contrast, for the MS indicator where a higher score is preferred, positive *t*-value points toward the Uniform crossover as a more efficient technique. Additionally, no significant difference between the two operators was observed for the SNS indicator. Finally, for the cumulative indicator Y-value (Equation- 4.100), the test also concludes that there is no significant difference between the performance of two competing crossover operators. As the *t*-value was unable to declare an overall winner, a simpler implementation of NSGA-III (namely with the Single-point crossover) is selected for further experiments.

4.6.4. Performance comparison

To find the best-performing metaheuristic, the quality indicator values were obtained for the approximation to the Pareto front. Each problem from the generated instance set was solved using the metaheuristics under the same computational budget. To remove any discrepancies due to the random variations in the working of the algorithms, five independent runs were performed for each algorithm. Additionally, to ensure the correct assessment of performance indicators, ideal and extreme solutions for a specific problem instance were obtained using all the solutions from the final Pareto front approximation generated by all 15 runs (five runs for the three algorithms). Objective function values for 525 (35×5×3) solutions for a single problem instance are then normalized using these extreme values of objective functions. After normalization, the best and worst values of the objective function should be zero and one, respectively, while others will vary linearly between the two. Finally, values for the performance indicators are calculated according to the equations from Subsection 4.6.2. Average values for the quality indicators from five runs of each algorithm are presented separately in **Table 4.21**. The table also contains the cumulative value (Y) of performance indicators alongside the average number of iterations each algorithm was able to perform under the fixed computational budget. It can be easily observed that the number of iterations drops drastically for the larger problem under the allotted computation budget for all three algorithms. As larger problems have larger computational requirements per iteration, it is an expected result. Similarly, for a specific problem instance, it can be seen the MOGWO always iterates more frequently than the MOPSO and NSGA-III. This can be attributed to the number of agents that each algorithm needs to manage in each iteration. MOGWO, with the least number of

agents (70 wolves), will have to perform fewer calculations per iteration than the NSGA-III, with a population size of 140. As we are working with the best combinations of parameters of each algorithm, it is interesting to see how each algorithm has compromised the number of agents (population/particles/wolves) and the number of iterations to give its best performance. Finally, to compare the performance of the metaheuristics, the box and whisker plots for each quality indicator are presented in **Figure 4.11**.

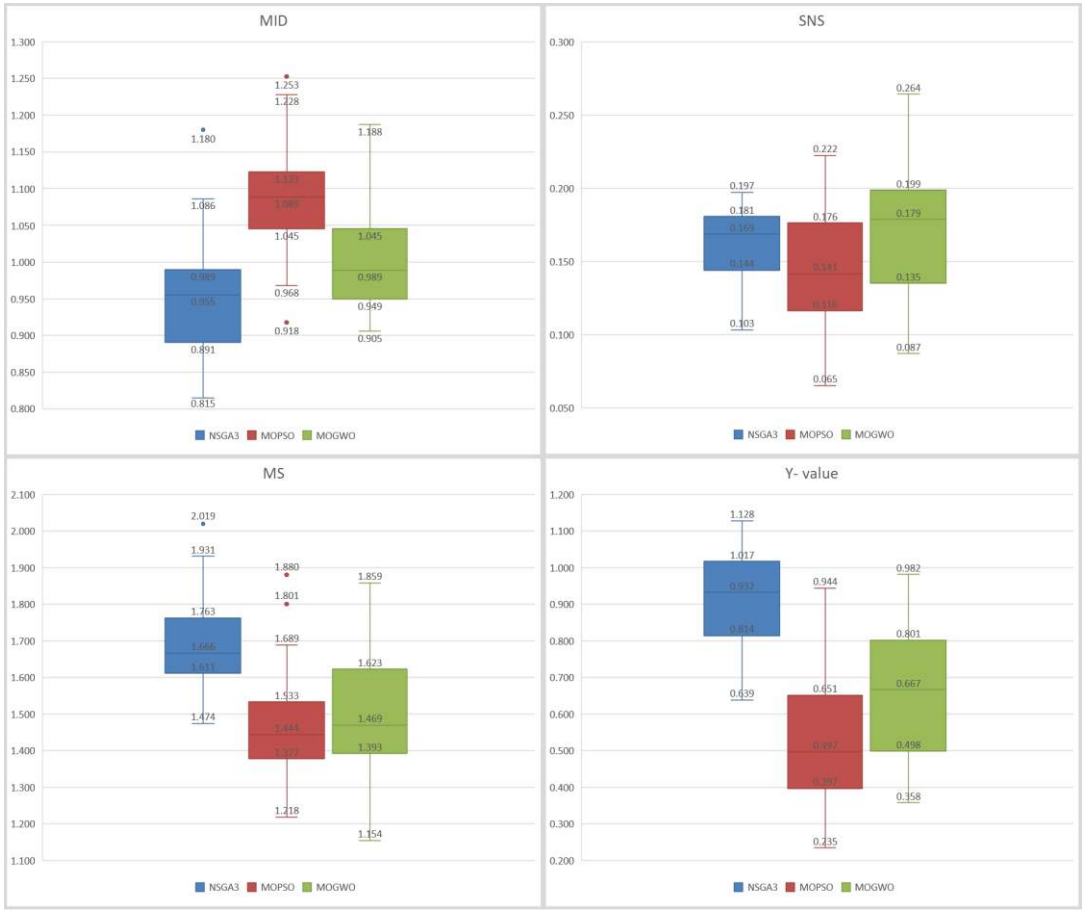


Figure 4.11: Performance comparison based on the quality indicators.

Some conclusions regarding the relative performances of the proposed metaheuristics can be drawn by analyzing the charts in **Figure 4.11** for the box sizes and their overlap with each other, along with the spread of the whiskers. For the

MID indicator, where a smaller value is preferable and no overlap is observed between boxes of MOPSO and MOGWO, it can be concluded that the latter performs better than the former. In contrast, while comparing the MID values for NSGA-III and MOGWO, a clear overlap for the boxes can be observed. Additionally, with the median value for NSGA-III just falling within the overlapped region, the conclusion that "*there is likely to be a difference between the two groups*" cannot be made. The opposite of this can be observed for the MS indicator, where the median for NSGA-III avoids falling inside the overlap of the middle two quartiles (box region) for NSGA-III and MOGWO. Hence, it can be said that the performance of the NSGA-III regarding maximum spread is likely better than the MOGWO. For the SNS indicator, where clear overlap between the box regions can be seen for any two metaheuristics and at least one median falls within this region, no inferences can be made using box and whisker plots alone. Finally, by reading the plot for net quality indicator (Y), the Lower limit of the third quartile for NSGA-III is greater than the lower limit of the first quartile for other algorithms. Hence, there is no overlap with the boxed region of NSGA-III, and it can be conclusively declared the best algorithm among the three. When comparing MOPSO and MOGWO using the Y-value, the median of MOPSO is just within the overlapping boxed region of the two algorithms. To establish further differentiation between these algorithms, some additional statistical tests can be performed. However, as NSGA-III was found to be producing the best quality approximations of the Pareto front, we have opted not to conduct those tests and suggest it as the future scope of the work. Similarly, we also suggest that a statistical analysis should be carried out for any modification made to the various operators of NSGA-III.

Table 4.21: Average values of the results obtained by five independent runs.

Configuration Number	Quality Indicators									Y-value			Number of Iterations		
	MID			SNS			MS								
	NSGA-III	MOPSO	MOGWO	NSGA-III	MOPSO	MOGWO	NSGA-III	MOPSO	MOGWO	NSGA-III	MOPSO	MOGWO	NSGA-III	MOPSO	MOGWO
1	1.046	1.106	1.07	0.196	0.17	0.193	1.931	1.88	1.859	1.081	0.944	0.982	23245	35045	37061
2	1.069	1.12	1.112	0.178	0.199	0.136	2.019	1.689	1.816	1.128	0.768	0.84	20958	34019	35484
3	0.976	1.099	0.97	0.197	0.185	0.237	1.843	1.801	1.651	1.064	0.886	0.918	20088	32798	33985
4	0.875	1.085	0.924	0.19	0.142	0.184	1.497	1.39	1.154	0.812	0.447	0.415	19438	32253	33620
5	0.989	1.123	1.047	0.141	0.119	0.154	1.669	1.43	1.406	0.821	0.426	0.513	17175	32114	33102
6	0.815	0.918	0.905	0.193	0.208	0.231	1.663	1.459	1.639	1.04	0.75	0.965	15309	31573	32281
7	0.871	1.044	0.949	0.168	0.178	0.183	1.651	1.539	1.454	0.948	0.674	0.688	17499	31653	32718
8	0.949	0.968	0.951	0.159	0.168	0.21	1.823	1.261	1.425	1.033	0.462	0.684	12028	25768	30803
9	0.989	1.03	1.012	0.16	0.157	0.214	1.763	1.456	1.484	0.934	0.582	0.687	9987	21192	30097
10	0.902	1.051	1.026	0.185	0.125	0.179	1.677	1.323	1.662	0.96	0.397	0.815	12238	24347	30288
11	0.902	0.995	0.934	0.181	0.222	0.264	1.761	1.45	1.548	1.041	0.677	0.878	9232	18500	29226
12	0.962	1.053	0.98	0.179	0.173	0.175	1.737	1.43	1.302	0.955	0.549	0.497	6420	17636	23111
13	0.887	1.061	0.965	0.18	0.209	0.179	1.659	1.287	1.491	0.952	0.434	0.705	9610	18395	29809
14	0.975	1.062	0.965	0.158	0.124	0.201	1.749	1.42	1.422	0.931	0.482	0.658	6389	17015	21712
15	0.882	1.041	0.918	0.154	0.141	0.18	1.581	1.218	1.387	0.854	0.318	0.65	5954	15613	20265
16	0.94	1.093	0.979	0.171	0.143	0.187	1.735	1.512	1.551	0.966	0.562	0.759	4063	13560	17265
17	0.984	1.178	1.011	0.173	0.138	0.119	1.636	1.39	1.25	0.825	0.35	0.358	2267	9011	12087
18	1.18	1.253	1.133	0.159	0.132	0.156	1.924	1.633	1.653	0.903	0.512	0.676	1502	6964	8936
19	0.918	1.137	1.04	0.138	0.115	0.158	1.495	1.57	1.531	0.715	0.548	0.649	2310	8588	11047
20	0.975	1.118	1.039	0.169	0.107	0.135	1.629	1.251	1.391	0.824	0.24	0.488	1123	5234	6732
21	1.06	1.217	1.188	0.14	0.11	0.117	1.636	1.504	1.573	0.716	0.398	0.503	695	3510	4677
22	0.862	1.051	0.934	0.112	0.111	0.12	1.474	1.466	1.397	0.724	0.527	0.582	1286	5469	6901
23	0.943	1.121	0.997	0.103	0.065	0.093	1.478	1.439	1.359	0.639	0.383	0.454	692	3361	4528
24	1.086	1.228	1.128	0.122	0.089	0.087	1.605	1.373	1.421	0.641	0.235	0.381	443	2345	3188

4.6.5. Discussion on Pareto front and extreme solutions

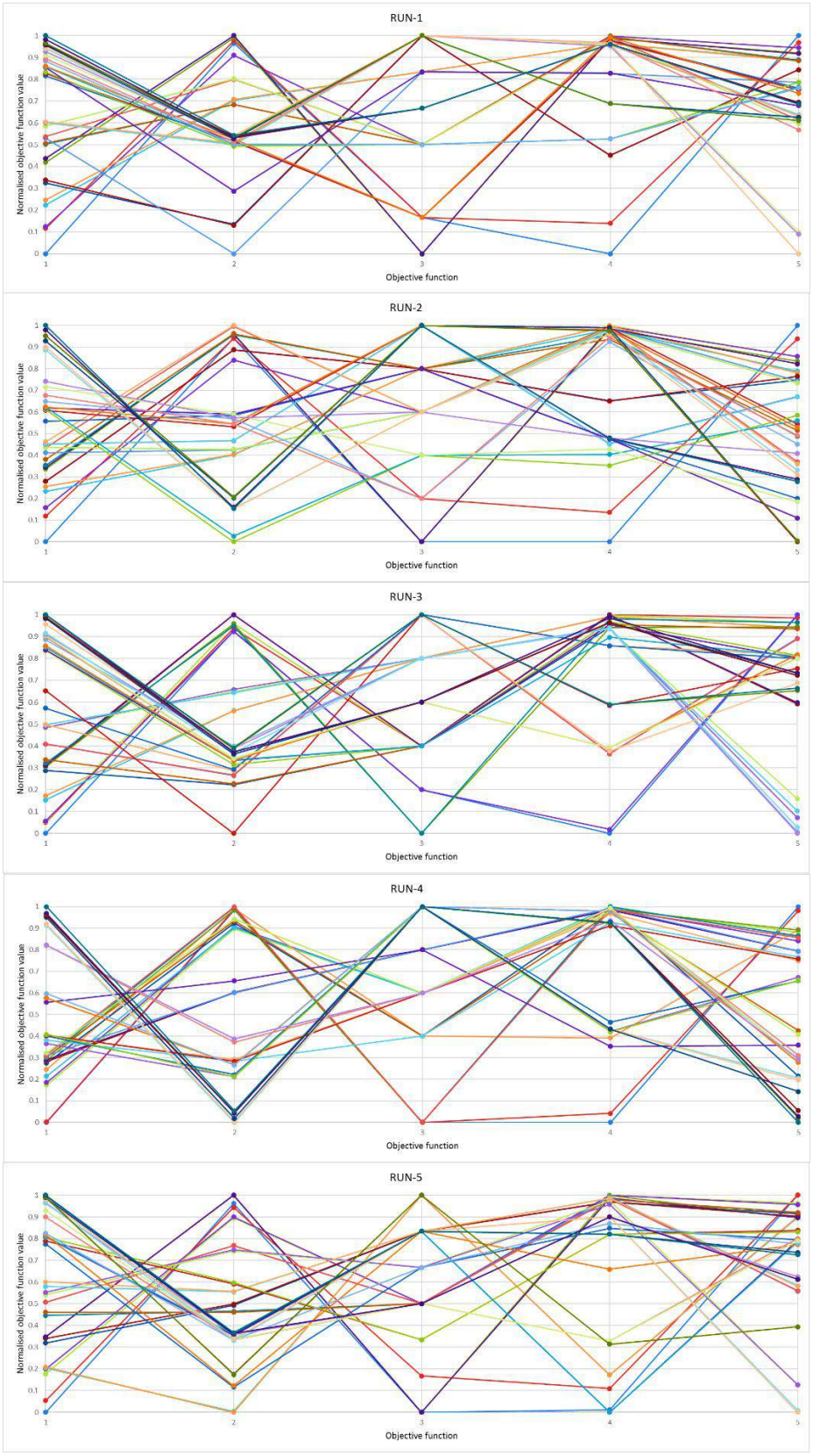


Figure 4.12: Pareto front obtained by NSGA-III for HHI 25_50_40.

In this section, an analysis of the Pareto front approximations found by metaheuristic NSGA-III is presented. Again, the average-size problem instance of 25 healthcare staff and 50 patients (problem configuration-13 from **Table 4.13**) is selected for the said analysis. **Figure 4.12** presents the solutions from the PFA obtained by the independent runs of the NSGA-III using the value path method. A single solution from the PFA is represented by the line joining the normalized objective function values on the vertical axis. It is clear from **Figure 4.12** that the metaheuristic was able to find a sufficiently diverse set of non-dominated solutions in each run. While the spread of the objective function values is found to vary in different runs, a good spread for each objective function was found in at least one of the runs. Similarly, by observing the change in the slopes of the lines, judgment regarding the quality of a solution in relation to the good trade-off between the two consecutive objectives can be made. For example, run-2 and run-5 find a good trade-off between Objectives 1 and 2 in comparison to the run-4. Additionally, the zig-zag nature of the lines hints toward the inherent competitiveness among the selected objectives, where improvement in one can only be made at the expense of the other.

While **Figure 4.12** conveys the information about the spread of the solutions in the PFA, it fails to report on its extent. **Table 4.22** presents the extreme values of the objective functions for each run. The consistency of the algorithm in finding these extreme solutions can be judged by observing **Table 4.22**. For Objectives 3 and 4, the algorithm was consistently able to find the extreme values in each run. In contrast, NSGA-III falls short in finding the non-dominated solutions with higher values of Objective 5 in run-2 and run-4. Similar observations can also be made for Objectives 1 and 2. The ideal solution and the nadir solution obtained by collecting

the best and the worst values for each objective from all runs have also been mentioned in **Table 4.22**. These theoretical solutions are used to calculate the values for the quality indicators for PFA.

Table 4.22: Extreme solutions obtained by NSGA-III for problem instance HHI 25_50_40.

		Run 1	Run 2	Run 3	Run 4	Run 5	Ideal solution	Nadir solution
Objective 1	Min	36.17	-287.00	536.59	-339.00	-294.17	6912.09	-339.00
	Max	5886.25	6270.75	6912.09	6724.60	5891.51		
Objective 2	Min	321.08	307.65	273.45	335.98	274.96	273.45	2048.84
	Max	1851.72	1168.73	1487.55	1542.18	2048.84		
Objective 3	Min	0	0	0	0	0	6	0
	Max	6	5	5	5	6		
Objective 4	Min	1.05	0.10	0.08	0.58	0.35	0.08	465.58
	Max	440.50	461.30	407.47	461.10	465.58		
Objective 5	Min	15.00	5.00	40.00	10.00	15.00	5.00	1898.92
	Max	1294.83	569.10	1898.92	566.30	1886.92		

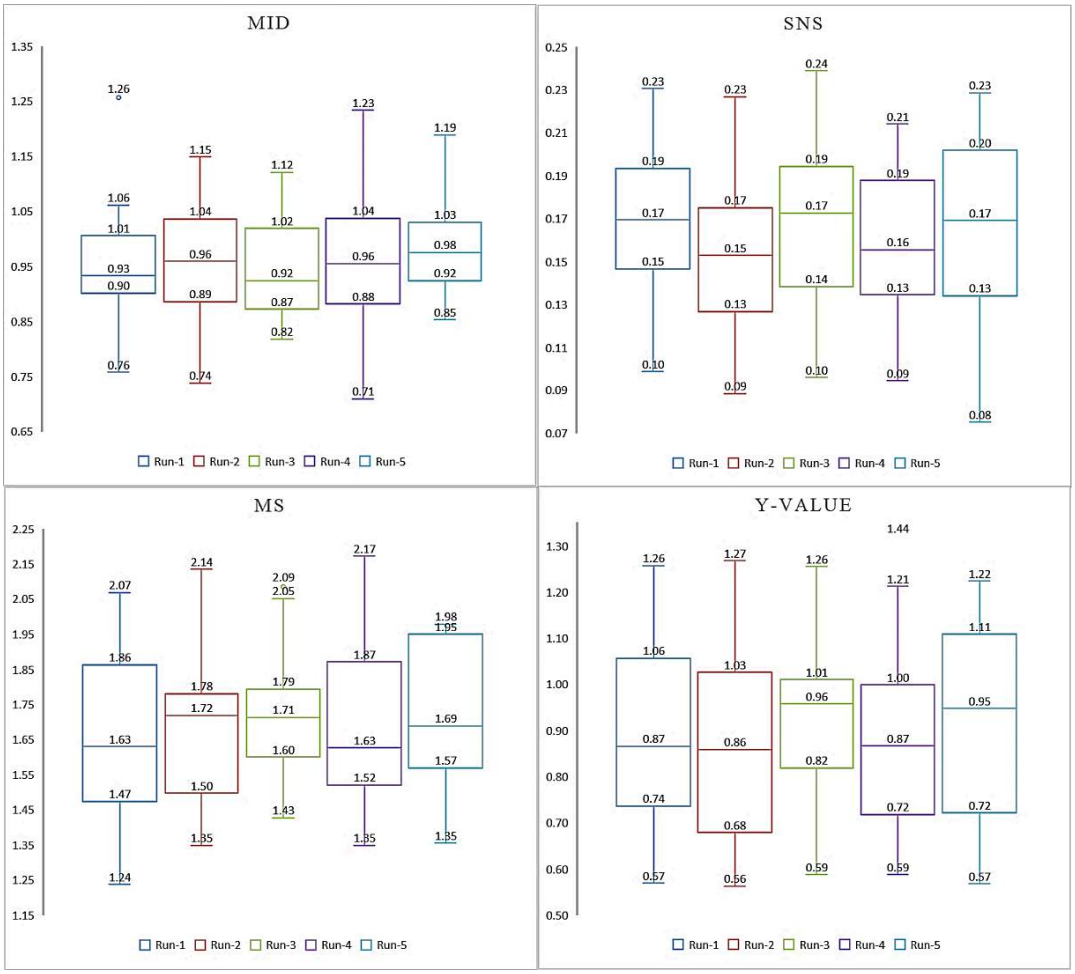


Figure 4.13: Box and whisker plot comparing the runs of NSGA-III.

Some statistical tests are also carried out to establish the capability of NSGA-III in producing consistent quality Pareto solutions. Firstly, separate box and whisker plots are drawn to compare the five independent runs of NSGA-III for each quality indicator **[Figure 4.13**]. On the surface, the metaheuristic seems to produce the same quality of solution in each run as all the boxes of an individual Box and Whisker plot overlap with each other. To confirm the result, One-way repeated measures ANOVA is selected because the data set is obtained by the repeated runs of the NSGA-III on the same problem instances. First, the Kolmogorov-Smirnov test for normality is performed to ensure the requirements of the one-way repeated measures ANOVA. With all requirements met, ANOVA is used to find any

significant difference among independent runs of NSGA-III for each quality indicator separately. The result of the test is presented in **Table 4.23**.

Table 4.23: Result of one-way repeated measures ANOVA for 5 independent runs of NSGA-III.

	Mean	Standard Deviation	*F*-ratio	*p*-value	α = 0.05	α = 0.01
MID	0.960	0.099	2.480	**0.049**	**Yes**	No
SNS	0.163	0.038	1.014	0.404	No	No
MS	1.693	0.199	0.739	0.568	No	No
Y-value	0.896	0.195	0.514	0.726	No	No

Results from **Table 4.23** show that, with α = 0.01, there is no significant difference between the approximate Pareto fronts obtained by the independent runs for any quality indicator. However, with α = 0.05, at least one of the runs of the NSGA-III was found to be significantly different from the others on the basis of the MID indicator. To investigate further, a paired sample t-test (post-hoc test for one-way repeated measures ANOVA), with α = 0.05, was conducted on all possible pairs of independent runs for only the MID values **[Table 4.24].** The test shows that Run-5 performed worse than Run-1 and Run-3 on the MID indicator. The same result can also be substantiated by observing the Box and Whisker plot for the MID indicator. Medians of neither Run-1 nor Run-3 fall inside the box of Run-5, which hints toward the likely difference between the compared groups. Hence, it can be concluded that our implementation of NSGA-III can produce a consistent quality Pareto front for SNS, MS, and Y indicators and only falls marginally short in the case of the MID indicator. **It should be noted that this result does not show that Pareto fronts obtained during Run-5 will be consistently worse for all instances in comparison to other runs.** To pick the best Pareto front for further analysis,

quality indicators for the test problem HHI 25_50_40 are independently compared. **Table 4.25** presents the value of the quality indicators with the cumulative performance indicator (Y) for the PFA obtained for problem instance HHI 25_50_40. With the best Y-value, approximated Pareto front found in run-5 is selected for further discussion and is presented in **Table 4.26**.

Table 4.24: Result of paired sample *t*-test for independent runs of NSGA-III using MID indicator.

	***p*-value**				
	Run 1	Run 2	Run 3	Run 4	Run 5
Run 1	-	0.179	0.858	0.442	**0.012**
Run 2	0.179	-	0.350	0.748	0.068
Run 3	0.858	0.350	-	0.616	**0.013**
Run 4	0.442	0.748	0.616	-	0.103
Run 5	**0.012**	0.068	**0.013**	0.103	-
	***t*-value**				
	Run 1	Run 2	Run 3	Run 4	Run 5
Run 1	-	1.384	0.181	0.783	**2.731**
Run 2	-1.384	-	-0.954	-0.325	1.918
Run 3	-0.181	0.954	-	0.508	**2.680**
Run 4	-0.783	0.325	-0.508	-	1.698
Run 5	**-2.731**	-1.918	**-2.680**	-1.698	-

Table 4.25: Quality indicator values for the five independent runs of NSGA-III using HHI 25_50_40.

Run	Quality indicators			Y-value
	MID	SNS	MS	
1	**0.80480**	0.17868	1.66782	1.04170
2	0.85333	0.17377	1.47889	0.79934
3	0.91357	0.18221	1.71621	0.98485
4	0.88952	0.18173	1.54506	0.83726
5	0.97307	**0.18236**	**1.88746**	**1.09675**

Table 4.26: Pareto front obtained by run-5 of NSGA-III for problem instance HHI 25_50_40.

Solution Number	Objective function values					Utility Score
	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5	
1	-294.17	343.85	0	460.58	15.00	1.972
2	36.17	376.08	1	415.07	15.00	2.272
3	795.84	462.10	3	0.35	86.13	3.533
4	935.42	453.17	3	0.63	96.13	3.554
5	966.09	2042.51	6	385.50	206.13	2.277
6	989.00	2048.84	6	385.50	201.13	2.280
7	1681.18	1176.47	5	15.00	176.13	3.527
8	1811.68	1165.65	5	15.00	196.82	3.543
9	1851.52	275.91	0	1.30	208.78	3.241
10	1853.42	274.96	0	6.70	208.78	3.230
11	2463.92	1224.21	3	9.52	164.77	3.311
12	2549.92	1232.02	3	9.52	164.77	3.321
13	2837.13	687.25	3	4.23	842.67	3.323
14	2837.13	687.25	3	8.23	838.85	3.317
15	3049.00	733.29	4	11.83	1651.20	3.050
16	3119.59	722.21	4	19.92	1651.20	3.050
17	3289.26	1065.50	5	7.00	795.28	3.536
18	3420.42	1062.90	5	7.00	800.28	3.556
19	4495.36	1843.13	4	71.30	399.62	3.199
20	4589.26	1001.42	2	84.07	318.30	3.371
21	4689.26	991.37	2	84.07	334.32	3.385
22	4701.76	1420.52	5	465.58	407.83	2.785
23	4715.34	1411.59	5	465.58	417.83	2.787
24	4761.15	1832.53	5	159.38	443.25	3.202
25	4800.43	1461.32	4	60.92	435.63	3.467
26	5268.10	1457.31	3	312.97	389.77	2.861
27	5455.10	1452.10	3	312.97	409.77	2.883
28	5661.51	1414.60	3	47.15	720.35	3.343
29	5681.43	1458.35	5	46.57	1870.90	3.041
30	5781.43	1448.31	5	46.57	1886.92	3.055
31	5819.26	1410.84	5	84.07	508.75	3.738
32	5827.68	1738.98	6	319.80	1150.88	2.871
33	5839.84	1743.10	6	319.80	1150.88	2.871
34	5881.68	1403.66	3	47.15	740.35	3.374
35	**5891.51**	**1399.76**	**5**	**84.07**	**528.75**	**3.745**
Minimum Value	-294.17	274.96	0	0.35	15.00	1.972
Maximum Value	5891.51	2048.84	6	465.58	1886.91	3.745

Table 4.26 presents the utility values for all 35 non-dominated feasible solutions along with their objective function values for problem instance HHI 25_50_40. To obtain the utility value of a solution, a normalized score between zero and one for each objective was first calculated using the minimum and maximum values. A weighted sum of these normalized scores is presented as the utility value of the solution in **Table 4.26**. With equal weightage allotted to each objective function, Solution 35 provides the best utility score. The benefit of dedicated multi-objective metaheuristics like NSGA-III can be observed in this stage. Best solutions under varying weights can easily be calculated by only a single run of the algorithm, making it easier for a decision-maker to evaluate and select the most appropriate solution for the day.

4.6.6. Calculating the cost of patient convenience

As stated earlier, one of the aims of this study is to establish the relationship between patient convenience and operational profit. While the extreme values for both the objectives (Objective 1 and Objective 5) can be seen in **Table 4.26**, to find the strict relation between the two objectives, the dominance between solutions should be checked strictly based on these objectives. For example – when considering all five objectives, Solutions 30 and 31 are not dominated by each other. While Solution-31 provides better value for Objectives 1, 2, and 5, Solution-30 performs better for Objective-4. However, when considering these two solutions for only Objective 1 and Objective 5, Solution 31 dominates Solution 30. This issue needs to be eliminated while establishing the relationship between patient convenience and profit, and hence, the dominance relationship among the solutions is reexamined. First, the PFA from each run is considered separately, and all the

dominated solutions in relation to Objectives 1 and 5 are removed. The remaining solutions from the Pareto fronts are charted in **Figure 4.14**.

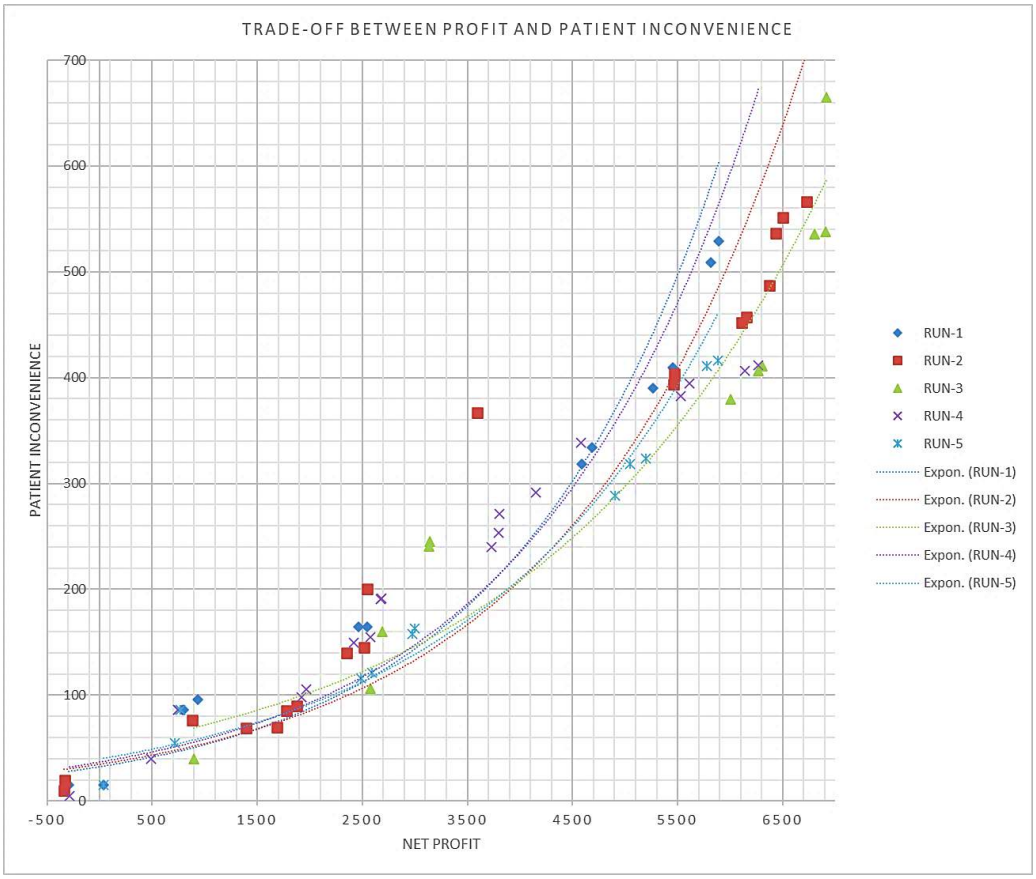


Figure 4.14: Non-dominated solutions while only considering Objective-1 and Objective-5.

It can be easily observed that while extreme values for Objective-1 remain unchanged, the maximum value for Objective 4.5 (patient inconvenience) has come down from 1886.91 to 528.75 for Run 1. Similar changes can also be observed for all the runs. Additionally, a competitive relation between the two objectives can also be seen. Improvement in patient convenience (decrease in the value of Objective 4.5) was achieved by sacrificing the profit. Further, a crescent shape formed by the solutions hinds the exponential relation between the objectives where profit reduces slower than the improvement in patient convenience. To further

improve the findings, all the newly non-dominated solutions from five runs (12, 19, 11, 19, and 13 from Runs 1, 2, 3, 4, and 5 respectively) are pooled together. A dominance check for these 74 solutions is performed, and the non-dominated solutions (27 in number) are finally used to establish the effect of patient convenience over the net profit.

Table 4.27: Trade-off analysis between Patients' inconvenience and Net profit.

Solution Number	Net profit (Rupees)	Patient inconvenience	D1 (%)	D2 (%)	D2/D1
1	6912	664.97	0	0	_
2	**6911.43**	**537.95**	**0**	**19.4**	**2130.59**
3	6800.68	535.6	1.5	19.8	12.85
4	6375.76	487	7.4	27.2	3.67
5	**6307.93**	**411.22**	**8.3**	**38.7**	**4.65**
6	6270.75	411.63	8.8	38.7	4.37
7	6266.94	406.48	8.9	39.5	4.44
8	6138.92	406.63	10.7	39.4	3.7
9	6004.93	379.7	12.5	43.6	3.48
10	**5200.58**	**323.53**	**23.6**	**52.1**	**2.21**
11	5046.75	318.53	25.7	52.9	2.06
12	4904.49	288.53	27.7	57.5	2.08
13	**3803.59**	**271.18**	**42.9**	**60.1**	**1.4**
14	3793.42	253.33	43	62.8	1.46
15	3726.83	239.65	43.9	64.9	1.48
16	2996	162.77	54	76.7	1.42
17	2974.58	157.77	54.3	77.4	1.43
18	2592.42	121.28	59.6	83	1.39
19	2580.01	106.08	59.7	85.3	1.43
20	1966.67	105.15	68.2	85.5	1.25
21	1918.34	98.15	68.9	86.5	1.26
22	1874.51	90	69.5	87.8	1.26
23	1781.84	85	70.8	88.5	1.25
24	1687.67	70	72.1	90.8	1.26
25	894.58	40	83	95.4	1.15
26	36.17	15	94.8	99.2	1.05
27	-339	10	100	100	1

Table 4.27 presents all 27 non-dominated solutions (in relation to only Objective 4.1 and 4.5) obtained by all five NSGA-III runs. Percentage reduction with respect to Solution 1 for both Objectives 1 and 5 are presented under D1 and D2 columns, respectively. The last column (D2/D1) calculates the extent of gain in patient convenience with respect to the corresponding sacrifice in the profit. The same is also graphed for each non-dominated solution in **Figure 4.15** in the order of decreasing profit. **Figure 4.15** confirms the exponential relation between a reduction in profit and an improvement in patient convenience, where said gains plateau beyond a point. In particular, a significant improvement (~20%) in patient convenience was achieved with virtually no reduction in profits (Solution 2). In the same manner, approximately 40% improvement in patient convenience (38.7% reduction in the value of Objective 4.5) can be achieved by sacrificing only 8.3% in profits. It can also be seen that the improvement in patient convenience relative to the sacrificed profit diminishes rapidly. To achieve a minimum 50% improvement in patient convenience, almost 25% profit reduction is expected (Solution 10). Similarly, 60% reduction in Objective 4.5 was only achieved by sacrificing more than 40% of the profit.

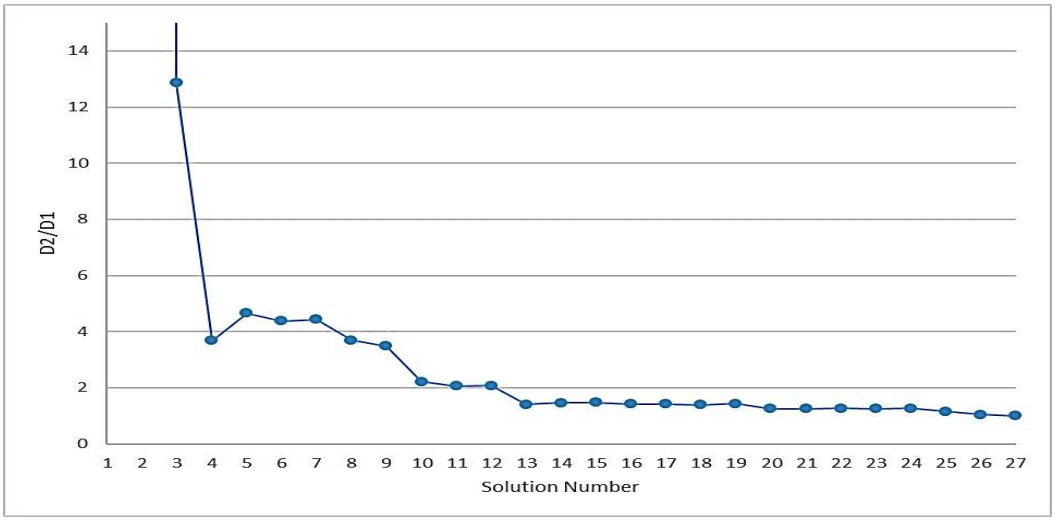


Figure 4.15: Relative gain in patient convenience with a reduction in profit.

Additionally, **Figure 4.15** can be used by a decision-maker to evaluate two adjacent solutions for better gains. For example- with a more lenient goal of achieving an approximate (not minimum) 50% reduction in patient inconvenience, a decision-maker might prefer Solution-9 over Solution-10. Whereas Solution-9 only sacrifices half the profit in comparison to Solution-10 for nearly similar improvements.

4.6.7. Robustness testing

In this section, the robustness of the Pareto front under the incremental increase in service and travel time is tested. Once again, the medium-sized problem instance HHI 25_50_40 is selected for the experiment. To establish the robustness of the obtained Pareto front, 50 variations to the selected problem instance are considered. To generate these 50 variations, in addition to the original problem instance, an increasingly high value (in an increment of 2%) for both the service as well as travel time is separately considered. First, the original problem is solved in order to obtain a well-diverse Pareto front while adhering to the experimental setting explained in Section 4.6.1. The experiment is to establish the robustness of this Pareto front under the increase in service and travel time. To achieve this, the chromosomes of the solutions belonging to the approximated Pareto front are saved and evaluated for the varying service and travel time. New objective scores for all 50 variations are generated using the simulation-based 3-step fitness evaluation method from NSGA-III. Changes in objective function values are used to check the dominance of the solutions from the original Pareto front. The number of the non-dominated solutions under each variation of the original problem is tracked. In addition to this, quality indicator values for the new Pareto fronts consisting of only the non-dominated solutions for each problem variation are then obtained. 10

independent runs are performed to calculate the average percentage degradation in the quality indicator values and the number of non-dominated solutions for the robustness analysis.

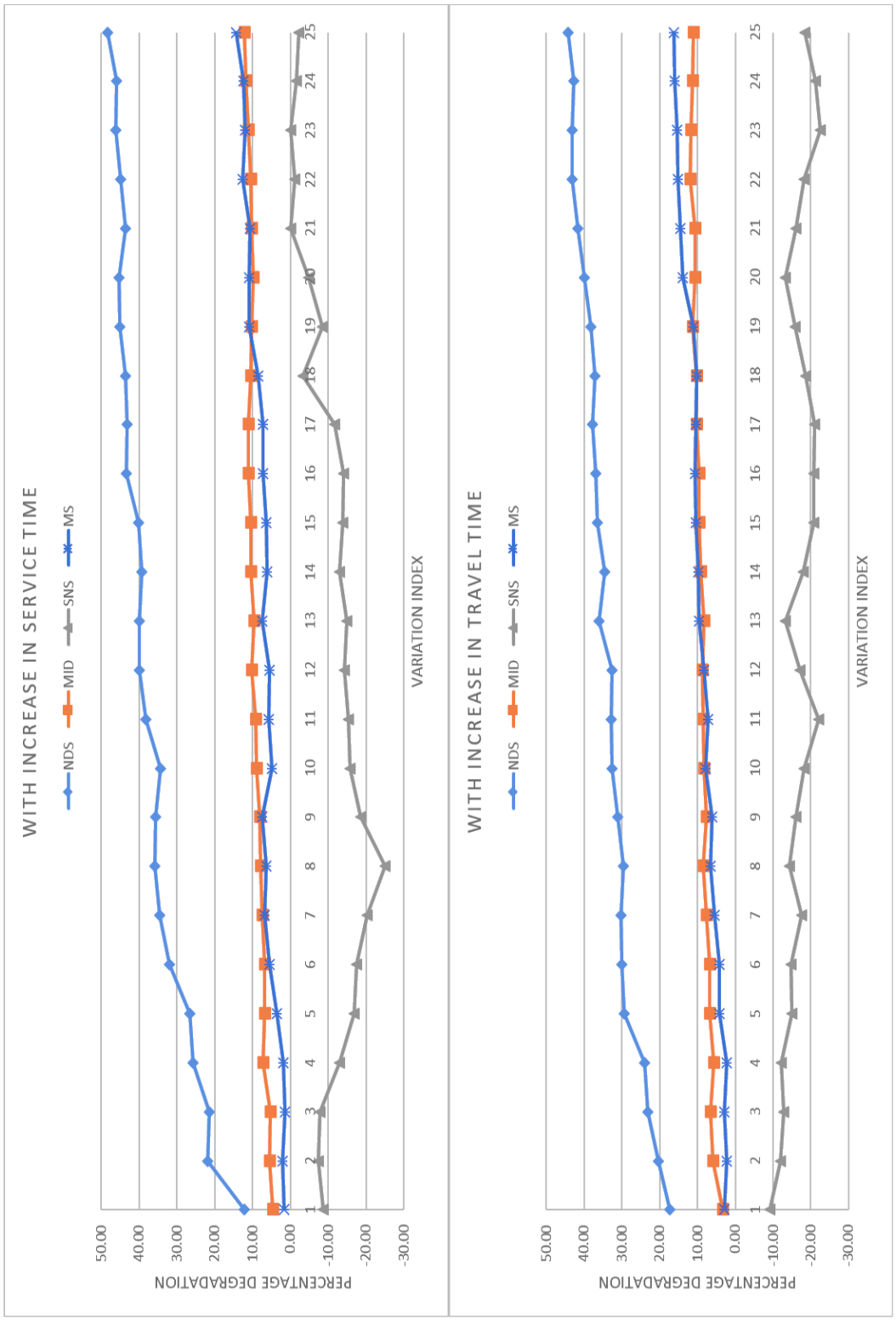


Figure 4.16: Changes in the quality indicator values during robustness analysis.

For a clearer understanding, only the average percentage degradation in the quality indicator values and the number of non-dominated solutions are graphed in **Figure 4.16**. As expected, a study decrease in the non-dominated solutions is observed with an increase in both the service and travel time. However, even for the maximum increase considered for the parameters, more than half of the non-dominated solutions for the original problem instance remain non-dominated. An even better result was observed for the MID and MS indicators. Under service time variations, a maximum of 12.2% and 14.5% reduction is seen for the MID indicator and MS indicator, respectively. Similarly, for travel time variations, a maximum of 11.2% and 16.4% degradation is recorded. In contrast, with an improvement in the value of the SNS indicator (negative degradation), the new Pareto front seems to have a better spread of the non-dominated solutions. This, while counter-intuitive, can be explained by the gaps left in the Pareto front after discarding the dominated solutions. It can be clearly observed that while the number of non-dominated solutions decreases steadily with the increase in service time and travel time, the quality of the Pareto front remains stable. With only a minor degradation in the quality indicators, we conclude that the Pareto front remains useful for small variations in the service or travel time.

4.7. Conclusions

In this work, we have proposed a highly generalized multi-objective home healthcare delivery model. Model is proven to be capable of handling a wide variety of requests in relation to the number of staff requirements, number of visit requirements, procedural dependencies, etc. Most of the restrictions that are commonly encountered in the home healthcare field were successfully incorporated into the model. Modifications to the implementation of ‘workload balance’

restrictions as well as ‘patient’s time window’ were also proposed as a novelty to the existing literature. Complications in implementing the latter, inconvenient time windows as the soft restriction, are successfully taken care. The model can identify all the possible cases and calculate the overlap of the scheduled visits with ITW accordingly. Finally, the definition and method to calculate patient inconvenience due to unnecessarily scattered visits are presented and incorporated into the mathematical model. The model identifies the correct number of sessions required to generate a ‘proper schedule’ for the selected visits and minimizes the total inconvenience due to improper schedules. On the solution approach front, an efficient implementation for a dedicated multi-objective metaheuristic NSGA-III was successfully developed and tested to solve a healthy variation of problem instances for the proposed HHC problem. Widely accepted quality indicators are first used to find the best parameter values for NSGA-III by following the Taguchi method for the experimental design. An approximated Pareto front for a medium-size problem was then generated and analyzed. The proposed implementation of NSGA-III was found to be capable of finding a diverse set of non-dominated solutions for a generalized home healthcare routing and scheduling problem. Further experiments are also conducted to establish the effect of patient convenience on the net profit. Experiment finds that a significant improvement in patient convenience can be achieved with only a minuscule reduction in profits. Finally, we present our recommendations to the decision-maker for the specific implementation of the study.