Chapter 3
A Generalized Single-Period Home Healthcare Delivery Problem

3.1. Introduction

In this chapter, we present a highly generic MIP model for HHC routing and scheduling problem capable of managing a wide variety of patient, caregiver, and procedure attributes. A major criticism of existing literature on HHC delivery is that models are highly specific and are not easily adaptable to be used in practice. We bridge this gap by presenting a highly generalized model that can easily be reduced to serve a wide variety of cases. As mentioned, our work does not restrict any key procedure attributes like staff and visit requirements or limit the number of requests a patient can make, simultaneously incorporating most of the commonly used constraints presented in the literature in a single model. Hence making it relatively easy to modify the model by changing parameters only. An example of this is provided in Section 3.7 by using the proposed model on instances provided by Fikar et al. (2016) with minimal modification. Further, the chapter includes the first-ever implementation of the inconvenient time slot (example of multiple time windows), which has not yet been considered in HHC literature. In the model, we also incorporate the constraints to limit contact between caregivers and patients to adhere to the COVID-19 guidelines. The use of such a set of constraints is novel in the literature related to HHC. Additionally, while most of the studies on the HHC problem are based in Europe or Northern America, the problem setting in our work evolves from the Indian scenario and only considers languages and gender for the compatibility of assignment in addition to the procedure capability.

Due to the generalized nature of the model, the computational requirement to solve the problem exceeds the commonly available capability quite early. In the current work, multiple unconventional ‘hardware first’ techniques are developed to overcome the issue. Two MIP-based decomposition algorithms supported by a local search algorithm are designed first to divide the problems into hardware-manageable smaller subproblems and then solve the resulting subproblems. The decomposition technique generates independent clusters known as districts or groups. However, instead of generating clusters on the geographical plane using only the location-based information, we incorporate staff and patient requirements, capabilities, attributes, and preferences to form suitable groups. Additionally, a modified Genetic Algorithm (p-GA) with a novel 3-phase fitness evaluation function is developed to accommodate the complexity of constraints. Inherent parallelism in the evolutionary process is used to utilize the multi-core architecture of commonly available computational hardware.

The rest of the chapter is organized as follows. Section 3.2 presents the details of the problem being considered. The resulting mathematical model is presented in Section 3.3. and is used to solve a small problem instance in Section 3.4 to demonstrate the model’s capability. Section 3.5 includes the details of solution techniques, while Section 3.6 focuses on their comparative computational efficiency and tries to find a suitable niche for their application. Section 3.7 presents the effectiveness of our work on the benchmark instance from Fikar et al. (2016) and highlights the required modifications. Experiments on the ‘request accommodation policies’ are included in Section 3.8. Finally, Section 3.9 concludes the chapter.

3.2. Problem description

The current section explains a generalized problem of scheduling and routing of caregivers faced by an HHC service provider. At any given instance, a set of patients dispersed over a given geographic area (a city) is considered. There can be a variety of healthcare procedures offered by the service provider, such as nebulization, catheterization, and tracheotomy suction, to name a few. Each patient may demand a different procedure to be delivered to their home, while some patients may require multiple procedures. Further, certain procedures need to be repeated multiple times a day for a patient as per the requirement. Some of the procedures are interrelated, and precedence needs to be maintained (e.g., procedure A can only be performed before or 1 hour after procedure B is completed). For the recurring procedures, a certain gap is to be maintained (e.g., a minimum gap of 5 hours is to be maintained during two consecutive visits of a procedure). Patients may also have strong preferences for the gender of the caregivers or the language they speak. In addition, a patient may specify a time window that is not convenient to them which must be taken into account while planning the services.

To provide healthcare services at home, the service provider has a set of caregivers with specific technical qualifications to perform a set of procedures. Our model assumes that a caregiver is either qualified to perform a particular procedure or not. The skill level of the caregivers is not considered explicitly at this stage. In addition to the technical qualification, the language and gender preferences between the caregiver and patient are considered to decide the compatibility. A caregiver is said to be compatible with a specific demand request from a patient if they have the technical capability to carry out the required procedure, can speak at least one of the languages specified by the patient, and meets the gender preference of the patient. The challenge is then to assign caregiver(s) to a patient to perform the requested procedures based on compatibility. Apart from the compatibility, there are several other aspects that need to be taken into account while assigning caregivers to the patients. Typically, caregivers operate in shifts with specific start and end timing. Each shift is also characterized by a time window reserved for the mandatory break for the caregivers. Therefore, their availability at a specific time also needs to be considered while assigning them to a patient and procedure. Like the patients, caregivers are also dispersed over the given geographic area, which may be termed depots/hubs. They must start and end their duty at the respective depots. During the day, a caregiver can serve multiple patients with varied procedures. Similarly, a patient may get served by multiple caregivers for the procedures ordered. Some of the procedure demands the involvement of more than one caregiver (e.g., to lift the patient). However, motivated by the current pandemic situation, an additional restriction of 'limited-contacts' is also imposed on the patients as well as caregivers that limit the number of people they come in contact with during the day. The involvement of multiple caregivers further complicates their scheduling and routing due to the waiting and disjunction involved.

There is revenue associated with each procedure requested by patients. With the limited number of qualified caregivers, the objective is to maximize the total revenue generated by scheduling as many visits as possible by respecting the restrictions on patient-procedure and caregiver assignments and different temporal constraints. Note that, in this chapter, we do not consider travel costs or daily wages as components of the objective functions. The workforce size of caregivers is assumed to be fixed. We also assume that travel costs are constant since caregivers are usually provided with monthly bus passes or fixed travel allowances. This makes travel costs a fixed expense that can be excluded from the model. A similar assumption was made by Braekers et al. (2016). We realize that the resulting model is only usable in specific scenarios. Therefore, a broader set of factors is considered when formulating the objective function for the multi-objective HHC delivery problem presented in Chapter 4. The current problem is modelled as an extension of the multi-depot heterogeneous fleet vehicle routing problem with time windows (MD-HVRPTW) with additional side constraints. In this context, patients are treated as delivery nodes and healthcare centers are taken as hubs. Caregivers are the agents who need to visit these patients to perform the requested procedures with the condition that their routes must start and end at the hub. Due to the objective of maximizing revenue, the problem can also be seen as a team-orienteering problem (Gunawan et al., 2016).

From the strategic perspective, the objective of maximizing revenue can be achieved either by maximizing the number of patients that have to be completely served (for all their requested procedures) or by maximizing the fulfilled requests made by all the patients. **To identify the best operational strategy for HHC service providers, four independent objectives are formulated**. The first two are modelled under the policy of 'partial accommodation,' where serving all the procedure requests made by a selected patient is not mandatory. The latter two objectives are modelled under the policy of 'complete accommodation,' where it is mandatory to serve all procedures requested by a patient if the patient is selected for service. As hinted above, the present model is not a multi-objective optimization model, and these objectives are to be implemented independently of each other. Constraint definitions for both policies are slightly different and are explained in subsequent text.

3.3. Mathematical model

Model [M0]

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 procedures, $P = P_0 \cup P_1$
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

Input parameters:

Patient-related input parameters:

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

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.

Caregiver-related input parameters:

C_{kp} $\begin{cases} 1, \text{ if procedure p can be performed by caregiver } k. \\ 0, \text{ otherwise.} \end{cases}$

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 allowed working time for staff *k* (in minutes).

Procedure-related input parameters:

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

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

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

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

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

D_{pq} $\begin{cases} 1, \text{ if there is precedence to be held between procedure } p \text{ and procedure } q. \\ 0, \text{ otherwise.} \end{cases}$

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).

$\bar{T}_s$ 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 $\begin{cases} 1, \text{ if patient } i \text{ is selected for home healthcare delivery.} \\ 0, \text{ otherwise.} \end{cases}$

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

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

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

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.

$\overset{\frown}{\pi}_{ipuk}$ 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.

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

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

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

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

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

Objective functions:

$$max\ Z_1 = \sum_{p \in P} \sum_{i \in M} q_{ip} \tag{3.1}$$

$$max\ Z_2 = \sum_{i \in M} \sum_{p \in P} \left(q_{ip} * F_{ip} * \Omega_p \right) \tag{3.2}$$

$$max\ Z_3 = \sum_{i \in M} p_i \tag{3.3}$$

$$max\ Z_4 = \sum_{i \in M} \sum_{p \in P} \left(p_i * P'_{ip} * F_{ip} * \Omega_p \right) \tag{3.4}$$

Subject to

Assignment constraints:

$$q_{ip} \le P'_{ip} \qquad \forall\ i \in M, p \in P \tag{3.5}$$

$$\sum_{u \in V} v_{ipu} = q_{ip} * F_{ip} \qquad \forall\ i \in M,\ p \in P \tag{3.6}$$

$$v_{ip0} = 0 \qquad \forall\ i \in M, p \in P \tag{3.7}$$

$$v_{ipu} - v_{ipu-1} \leq 0 \qquad \forall\, i \in M, p \in P, u \in V\backslash\{0,1\} \qquad (3.8)$$

$$\sum_{k\in K} y_{ipuk} = v_{ipu} * R_{ip} \qquad \forall\, i \in M, \forall\, p \in P, u \in V \qquad (3.9)$$

$$y_{ipuk} \leq a_k * C_{kp} \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (3.10)$$

$$y_{ipuk} \leq \sum_{e\in L} (L'_{ie} * L''_{ke}) \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (3.11)$$

$$y_{ipuk} \leq \sum_{g\in G} \left(G'_{ig} * G''_{kg}\right) \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (3.12)$$

Routing constraints:

$$\sum_{p=0} \sum_{u=0} \sum_{j\in M} \sum_{q\in P} \sum_{v\in V} x_{hpujqvk} = a_k * H''_{kh} \qquad \forall\, k \in K, h \in H \qquad (3.13)$$

$$\sum_{i\in M} \sum_{p\in P} \sum_{u\in V} \sum_{q=0} \sum_{v=0} x_{ipuhqvk} = a_k * H''_{kh} \qquad \forall\, k \in K,\, h \in H \qquad (3.14)$$

$$\sum_{j\in N} \sum_{q\in P} \sum_{v\in V} x_{ipujqvk} = y_{ipuk} \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (3.15)$$

$$\sum_{j\in N} \sum_{q\in P} \sum_{v\in V} x_{jqvipuk} = y_{ipuk} \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (3.16)$$

$$x_{hpu\hat{h}qvk} = 0 \qquad \forall (h, \hat{h}) \in H, (p,q) \in P, (u,v) \in V, k \in K \qquad (3.17)$$

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

$$\forall i \in N, j \in M, (p,q) \in P, (u,v) \in V, k \in K \qquad (3.18)$$

Break and workload constraints:

$$\bar{T}_s - \pi_{h00k} \leq \mathrm{B} * (1 - a_k * H''_{kh} * S''_{ks}) \qquad \forall\, k \in K, h \in H, s \in S \qquad (3.19)$$

$$\xi_{ipuiqvk} = 0 \qquad \forall i \in M, (p,q) \in P, (u,v) \in V, k \in K \qquad (3.20)$$

$$\xi_{ipujqvk} \leq x_{ipujqvk} \qquad \forall (i,j) \in N, (p,q) \in P, (u,v) \in V, k \in K \qquad (3.21)$$

$$\sum_{i\in N}\sum_{p\in P}\sum_{u\in V}\sum_{j\in N}\sum_{q\in P}\sum_{v\in V} \xi_{ipujqvk} = a_k \qquad \forall\, k \in K \qquad (3.22)$$

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

$$\forall\, (i,j) \in N, (p,q) \in P, (u,v) \in V, k \in K \qquad (3.23)$$

$$\hat{\pi}_k - \bar{\bar{\vartheta}}_s * S''_{ks} \leq 0 \qquad \forall\, k \in K, s \in S \qquad (3.24)$$

$$\bar{\vartheta}_s * S''_{ks} - \hat{\pi}_k \leq 0 \qquad \forall\, k \in K, s \in S \qquad (3.25)$$

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

$$\leq \sum_{s\in S}\left(\bar{\bar{T}}_s - \bar{T}_s\right) * S''_{ks} \qquad \forall\, k \in K \qquad (3.26)$$

Scheduling constraints:

$$\left(\pi_{ipuk} - \tau_{ipu}\right) * y_{ipuk} = 0 \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (3.27)$$

$$\left(\ddot{\pi}_{ipuk} + \dddot{\pi}_{ipuk} - \pi_{ipuk}\right) * y_{ipuk} = 0 \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (3.28)$$

$$\tau_{ipu} + T_p - \bar{T}'_i \leq \mathrm{B} * \left(1 - \overleftarrow{\delta}_{ipu}\right) \qquad \forall\, i \in M, p \in P, u \in V \qquad (3.29)$$

$$\bar{\bar{T}}'_i - \tau_{ipu} \leq \mathrm{B} * \left(1 - \vec{\delta}_{ipu}\right) \qquad \forall\, i \in M, p \in P, u \in V \qquad (3.30)$$

$$\overleftarrow{\delta}_{ipu} + \vec{\delta}_{ipu} = v_{ipu} \qquad \forall\, i \in M, p \in P, u \in V \qquad (3.31)$$

$$\tau_{ipu} \leq \mathrm{B} * v_{ipu} \qquad \forall\, i \in M, p \in P, u \in V \qquad (3.32)$$

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

$$\forall\, i \in M, p \in P, u \in V \backslash \{0,1\} \qquad (3.33)$$

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

$$\forall\ i \in M\ , (p,q)\ \in P, (u,v) \in V \qquad (3.34)$$

$$\tau_{iqv} + T_q - \tau_{ipu} \leq \mathrm{B} * \left(1 - \vec{\sigma}_{ipuqv}\right) \quad \forall\ i \in M\ , (p,q)\ \in P, (u,v) \in V \qquad (3.35)$$

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

$$\forall\ i \in M\ , (p,q)\ \in P, (u,v) \in V\ , (p \neq q\ or\ u \neq v) \qquad (3.36)$$

Limited person-to-person contact constraints:

$$\sum_{p \in P} \sum_{u \in V} y_{ipuk} \leq \mathrm{B} * \eta'_{ik} \qquad \forall\ i \in M\ , k \in K \qquad (3.37)$$

$$\left(y_{ipuk} + y_{ipul}\right) - z_{ipukl} \leq 1 \quad i \in M\ , p \in P,\ u \in V, (\mathrm{k}, l) \in K\ , k < l \qquad (3.38)$$

$$\sum_{i \in M} \sum_{p \in P} \sum_{u \in V} z_{ipukl} \leq \mathrm{B} * \eta''_{kl} \qquad \forall\ k, l \in K\ \ , k < l \qquad (3.39)$$

$$\sum_{k \in K} \eta'_{ik} \leq \lambda' \qquad \forall\ i \in M \qquad (3.40)$$

$$\sum_{i \in M} \eta'_{ik} + \sum_{l \in K} \eta''_{kl} \leq \lambda'' \qquad \forall\ \mathrm{k} \in K \qquad (3.41)$$

$$p_i \in \{0,1\} \qquad \forall\ i \in\ M \qquad (3.42)$$

$$q_{ip} \in \{0,1\} \qquad \forall\ i \in\ M, p \in P \qquad (3.43)$$

$$v_{ipu} \in \{0,1\} \qquad \forall\ i \in\ M, p \in P, u \in V \qquad (3.44)$$

$$a_k \in \{0,1\} \qquad \forall\ k \in K \qquad (3.45)$$

$$y_{ipuk} \in \{0,1\} \qquad \forall\ i \in M, p\ \in P, u\ \in V, k \in K \qquad (3.46)$$

$x_{ipujqvk} \in \{0,1\} \qquad \forall\ (i,j) \in N, (p,q) \in P, (u,v) \in V, k \in K \qquad (3.47)$

$z_{ipukl} \in \{0,1\} \qquad \forall\ i \in M\ , p \in P, u \in V, (k,l) \in K \qquad (3.48)$

$\eta'_{ik} \in \{0,1\} \qquad \forall\ i \in M, \mathrm{k} \in K \qquad (3.49)$

$\eta''_{kl} \in \{0,1\} \qquad \forall\ (\mathrm{k}, l) \in K \qquad (3.50)$

$\tau_{ipu} \geq 0 \qquad \forall\ i\ \in M, p \in P, u\ \in V \qquad (3.51)$

$\pi_{ipuk} \geq 0 \qquad \forall\ i\ \in N, p \in P, u \in V, k \in K \qquad (3.52)$

$\ddot{\pi}_{ipuk} \geq 0 \qquad \forall\ i\ \in M\ , p \in P, u \in V, k \in K \qquad (3.53)$

$\dddot{\pi}_{ipuk} \geq 0 \qquad \forall\ i\ \in M\ , p \in P, u \in V, k \in K \qquad (3.54)$

$\hat{\pi}_k \geq 0 \qquad \forall k \in K \qquad (3.55)$

$\xi_{ipujqvk} \in \{0,1\} \qquad \forall\ (i,j) \in N, (p,q) \in P, (u,v) \in V, k \in K \qquad (3.56)$

$\overleftarrow{\delta}_{ipu} \in \{0,1\} \qquad \forall\ i \in M, p \in P, u\ \in V \qquad (3.57)$

$\vec{\delta}_{ipu} \in \{0,1\} \qquad \forall\ i \in M, p \in P, u\ \in V \qquad (3.58)$

$\overleftarrow{\sigma}_{ipuqv} \in \{0,1\} \qquad \forall\ i \in M, (p,q) \in P, (u,v) \in V \qquad (3.59)$

$\vec{\sigma}_{ipuqv} \in \{0,1\} \qquad \forall\ i \in M, (p,q) \in P, (u,v) \in V \qquad (3.60)$

While Objective (3.1) maximizes the total procedure requests served under the partial accommodation condition, Objective (3.2) maximizes the generated revenue under the same. Both Objective (3.1) and (3.2) are to be optimized under Constraints (3.5) to (3.60). Constraint (3.5) ensures that a procedure can be chosen for home healthcare delivery only if it is requested by the patient. It is imposed on individual procedures. Therefore, the procedures selected for service for a patient are independent of each other; thus, the policy of partial accommodation is attained. Constraint (3.6) to (3.8) initiates the exact number of visits needed to perform the selected procedures. Similarly, Constraints (3.9) guarantees the assignment of the required number of staff for each scheduled visit. Constraints (3.10) ensure that only on-duty staff capable of performing the procedure is assigned to a given visit. Constraints (3.11) and (3.12) hold the language and gender preferences of the patients.

Constraints (3.13) and (3.14) ensure that the trip of each staff starts and ends at their respective hubs. Constraints (3.15) and (3.16) maintain the continuity of the route along with Constraint (3.18), which eliminates the possibility of sub-tour formation. Similarly, Constraint (3.17) eliminates the possibility of movement between the hubs. Constraint (3.19) will start the route for the healthcare workers at the start time of their assigned shift. Constraints (3.20) to (3.25) are used to schedule a lunch break for the staff in between their assignments. Constraints (3.21) to (3.23) schedule exactly one lunch break for an on-duty staff between assigned visits. Additionally, Constraints (3.24) and (3.25) are used to restrict lunch break starting time for the scheduled break between given time windows. Constraint (3.26) restricts the total work allotted to staff within their assigned shift. Constraints (3.27) and (3.28) generate the arrival, waiting, and procedure starting time for every visit. Constraints (3.29) to (3.31) are used to avoid the ITW, mentioned by the patients beforehand, by either scheduling the visit before the time window or after. Constraints (3.32) and (3.33) maintain the proper gap between multiple visits of the same procedure in a day for a single patient. Similarly, Constraints (3.34) to (3.36) maintain disjunction between two procedures (if any) and restrict the possibility of scheduling two visits at the same time for a patient.

Constraints (3.37) to (3.41) ensure that the prescribed limit for person-to-person contact is not violated. Constraint (3.37) keeps track of patient-to-caregiver contacts. Interaction between a specific caregiver and a specific patient is only counted once, regardless of the number of interactions during the day. Similarly, Constraints (3.38) and (3.39) count the number of distinct contacts between caregivers by keeping track of instances when two different staff came in contact with each other (team-up during visits requiring multiple staff). Similar to caregiver-patient interaction, interaction between the same set of caregivers is only counted once, regardless of their repeated interaction throughout the day. Constraints (3.40) and (3.41) are used to keep the number of contacts under the allowed limits. Finally, Constraints (3.42) to (3.60) restrict all the decision variables within their appropriate value range. The alternate policy of complete accommodation of all the requested procedures for the selected patients has been implemented using Constraints (3.6) to (3.60). Here, Constraint (3.5) is replaced by Constraint (3.61), which is written as follows.

$$q_{ip} = p_i * P'_{ip} \qquad \forall\, i \in M, p \in P \qquad (3.61)$$

Constraint (3.61) ensures that every procedure requested by the selected patients is fulfilled. Relevant objectives for the policy of complete accommodation are expressed by Objective (3.3) and Objective (3.4). Objective (3.3) maximizes the number of selected patients under the restriction that all the procedure requests made by these patients will be fulfilled. Alternatively, Objective (3.4) maximizes the total generated revenue under the same condition. HHC problem, as modeled above, can be classified as a Clustered Team Orienteering Problem as described by Angelelli et al. (2014), where a score is associated with each cluster and can only be gained if all the nodes in that cluster are served. In our case, clusters are either made at the procedure level for the policy of partial accommodation (Objectives 3.1 and 3.2) or at the patient level for the policy of complete accommodation (Objectives 3.3 and 3.4).

3.4. An Illustrative Example

Table 3.1: Hub information.

Hub ID	Hub location	
	Latitude	Longitude
Hub 1	30.7004	76.8295
Hub 2	30.72935	76.78053
Hub 3	30.71166	76.83585

Table 3.2: Caregiver information.

Caregiver ID	Caregiver attributes			
	Hub	Shift	Language	Gender
HCW1	Hub 1	Morning	English, Hindi, Regional	Male
HCW2	Hub 2	Morning	English, Hindi	Female
HCW3	Hub 3	Evening	English, Hindi, Regional	Female

Table 3.3: Patient information.

Patient's ID	Patient's Attributes					
	Location		Language Preference	Gender Preference	Inconvenient Time	
	Latitude	Longitude			Start Time	End Time
Patient1	30.7703	76.79536	English, Hindi, Regional	Female	11:05	11:30
Patient2	30.72208	76.81581	Hindi, Regional	Male	None	
Patient3	30.67779	76.77103	English,	None	09:00	09:30
Patient4	30.73695	76.739	Regional	None	14:10	15:20
Patient5	30.74778	76.81607	Hindi, Regional	Male	15:15	16:20
Patient6	30.69344	76.77445	Hindi, Regional	Male	19:10	21:00

Table 3.4: Procedure information.

Procedures ID	Service Times (minutes)	Staff Required	Visits Needed Per Day	Revenue Per Visit (Rupees)	Procedure Required by	Procedure Capability
P7	20	1	1	400	Patient6	HCW1, HCW2, HCW3
P10	10	1	1	500	Patient1, Patient6	HCW1, HCW2, HCW3
P12	15	1	1	100	Patient2	HCW1, HCW2, HCW3
P16	20	1	1	350	Patient4	HCW1, HCW2, HCW3
P18	10	1	1	450	Patient1	HCW1, HCW2, HCW3
P22	20	1	1	350	Patient3	HCW1, HCW2, HCW3
P23	30	2	1	500	Patient4	HCW1, HCW2, HCW3
P24	30	1	2	350	Patient3	HCW1, HCW2, HCW3
P28	25	1	1	400	Patient5	HCW2
P29	35	1	3	200	Patient5	HCW2
P30	35	1	1	250	Patient4	HCW2
P35	10	1	1	100	Patient3	HCW2
P36	10	1	1	200	Patient5	HCW2
P38	45	1	1	500	Patient1, Patient4	None

To explain the proposed model, a small problem instance is set up that consists of 3 caregivers, 6 patients, and 14 procedures. The relevant information regarding the hubs, caregivers, patients, and procedures is presented in tabular format. Additionally, a rudimentary analysis of the solution is presented following the 3-D representation of the solution. While **Table 3.1** consists of the list of hubs with their geographical location, **Table 3.2** contains relevant information (like their

ID, assigned hub, shift, gender, and language capability) regarding each healthcare staff. As an example, HCW1 is a male who can speak and understand all three languages and is assigned to Hub 1 for morning duty. The morning shift for our model starts at 8 a.m. and ends at 4 p.m. (similarly, the evening shift begins at 1 p.m. and ends at 9 p.m.). Attributes of other healthcare staff can be read likewise. Similarly, **Table 3.3** elaborates on the various patient's preferences. For example, Patient2 cannot speak or understand English and must only be assigned to male staff (based on his expressed gender preference) who can converse in Hindi or regional language. Additionally, Patient2 resides at the mentioned location and does not have any inconvenient time slot that needs to be taken into consideration. Similarly, the procedure's service time, staff requirement, visit frequency, and associated revenue are mentioned in **Table 3.4**. Travel time between the nodes (patient locations and hubs) is provided in **Table 3.5**. Movement between hubs is irrelevant to our model; hence, a sufficiently high value is put as a placeholder.

Table 3.5: Travel distances between nodes.

Distance Matrix (minutes)									
	Hub 1	Hub 2	Hub 3	Patient1	Patient2	Patient3	Patient4	Patient5	Patient6
Hub 1	1000	1000	1000	21.5	11.0	24.0	26.1	20.9	16.8
Hub 2	1000	1000	1000	12.7	9.0	18.3	12.8	12.1	11.0
Hub 3	1000	1000	1000	17.9	7.4	20.5	22.5	17.3	13.2
Patient1	21.5	12.7	17.9	0	14.7	26.6	13.7	11.9	19.7
Patient2	11.0	9.0	7.4	15.2	0	16.4	19.6	13.3	9.5
Patient3	24.0	18.3	20.5	28.3	18.3	0	18.4	27.2	9.9
Patient4	26.1	12.8	22.5	13.5	19.4	18.6	0	20.2	14.9
Patient5	20.9	12.1	17.3	12.9	13.2	26.0	21.4	0	19.1
Patient6	16.8	11.0	13.2	21.0	11.0	6.9	12.8	20.0	0

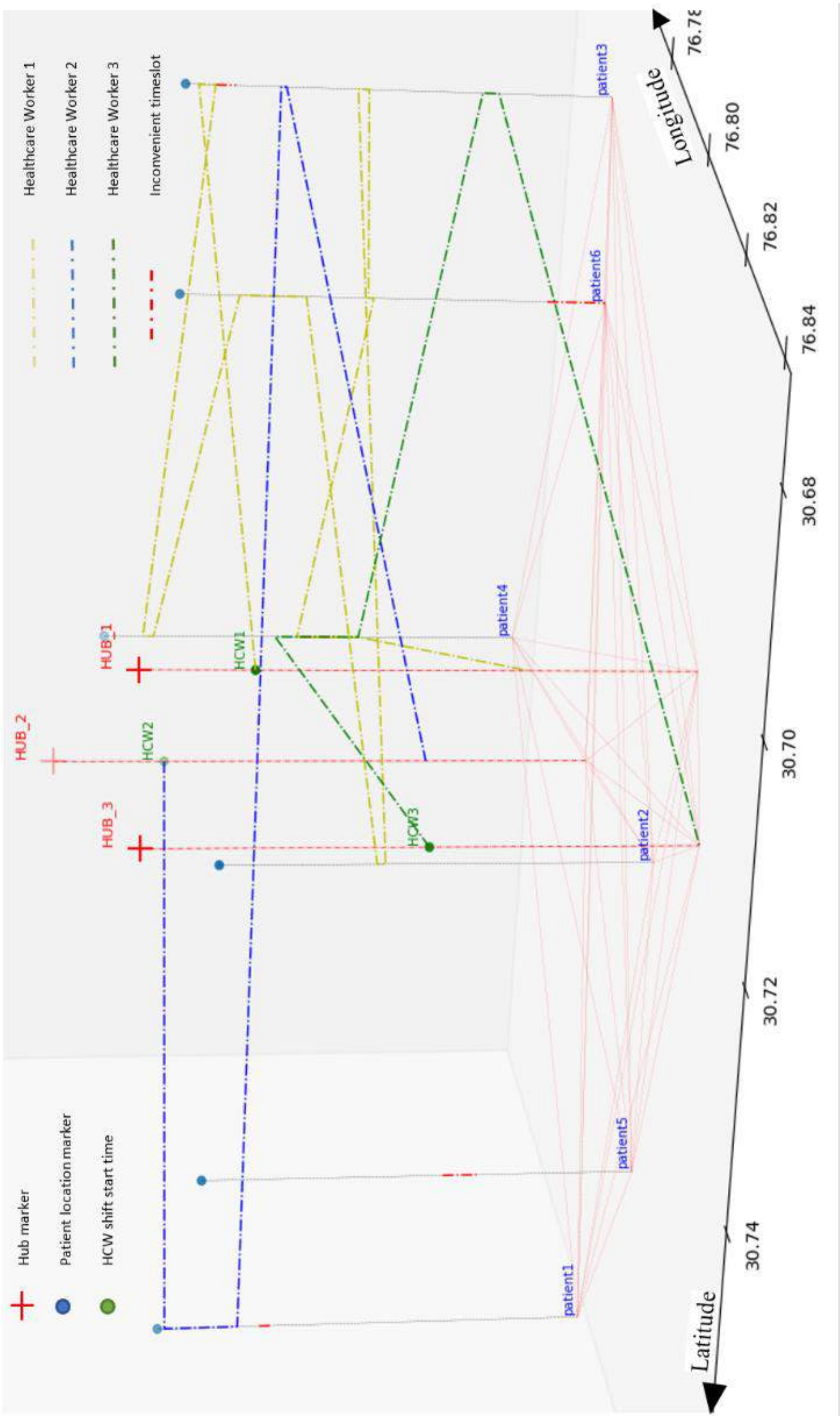


Figure 3.1: 3-D representation of feasible solution.

A 3-D representation of results obtained from running the MIP model (partial accommodation) for the illustrative example is displayed in **Figure 3.1**. Hub locations are represented by a red '+' sign, while patient locations are represented by a blue dot. The passage of time for the decision-making period is represented by vertical lines, "time-line" from top to bottom. Inconvenient time slots are marked on the respective patient's timeline by the red dot and dash line. Routes of all three caregivers are presented by different colored lines: yellow for HCW1 (healthcare worker 1), blue for HCW2, and green for HCW3. The route of a particular staff starts from the topmost point of their corresponding lines and moves downward. The movement of a line in the horizontal direction indicates movement in space (from one node to another), while movement in the vertical axis relates to the movement in time. For example, the route of the second healthcare worker (HCW2) starts from HUB_2 and goes to the nodes corresponding to patient1 and patient3, finally ending at HUB_2. HCW2 avoids the inconvenient time window for patient1 by finishing the procedure before the said time window.

Table 3.6 and **Table 3.7** separately show the selected procedures of each patient under different policy considerations. Shaded cells in both tables represent the procedures requested by the corresponding patients. Cells containing the '✓' mark show the successfully scheduled procedures in the optimal solution corresponding to the accommodation policy. By observing **Table 3.6** and **Table 3.7**, It can be seen that most of the requests have been accommodated under the partial accommodation policy, while the opposite is true for the complete accommodation case. A comparison of the optimal solution under the mentioned policies is presented in **Error! Reference source not found.** It can be observed t hat, under the 'complete accommodation' policy, 4 out of 6 patients were rejected entirely, while only one patient was denied under the alternative policy. Similarly, the amount of revenue generated is much higher in the partial accommodation case.

Table 3.6: Selected procedures under partial accommodation policy (Objective-3.1).

	Prescribed Procedures													
	P7	P10	P12	P16	P18	P22	P23	P24	P28	P29	P30	P35	P36	P38
Patient1		✓			✓									☐
Patient2			✓											
Patient3						✓		✓				✓		
Patient4				✓			✓				☐			☐
Patient5									☐	☐			☐	
Patient6	✓	✓												

Table 3.7: Selected procedures under complete accommodation policy (Objective-3.3).

	Prescribed Procedures													
	P7	P10	P12	P16	P18	P22	P23	P24	P28	P29	P30	P35	P36	P38
Patient1		☐			☐									☐
Patient2			✓											
Patient3						☐		☐				☐		
Patient4				☐			☐				☐			☐
Patient5									☐	☐			☐	
Patient6	✓	✓												

Table 3.8: Result comparison between the two accommodation policies.

Policy description	Number of Rejected Patients	Number of Requests	Number of visits	Revenue (rupees)	Solution Time (seconds)
Complete accommodation	4	3	3	1000	111.8
Partial accommodation	1	10	11	3950	117.4

3.5. Solution approach

The HHC routing and scheduling model presented in Section 3.3 can be solved to optimality with commercial solvers such as GUROBI. However, this is only true for a limited problem size. These solvers are not able to give even a feasible solution to the large and practical size problems within a reasonable computational time. This calls for the development of a specialized solution approach for the HHC routing and scheduling model. For this purpose, we initially developed a pair of MIP-based decomposition heuristics, namely single-stage decomposition (SSD) and multi-step decomposition (MSD). These heuristics aim to reduce the large problem instances into smaller and more manageable subproblems that can be solved in a given computational budget with the off-the-shelf solvers without moving too far from the optimality. The solution obtained by these heuristics is further improved using local search (LS). These heuristics are found to be outperforming the state-of-the-art solver GUROBI for small and medium-sized problem instances. However, their inability to competently handle very large-size problem instances has led us to develop a metaheuristics-based solution approach that is computationally efficient. In particular, we propose a variant of genetic algorithm (GA), namely p-GA (Parallelly-processed Genetic Algorithm), that utilizes the inherent parallelism during the formation of next-generation individuals in GA. The current section expands upon the logic and workings of these algorithms. The description of these algorithms presented below as well as the comparison done in Section 3.6, is specific to Objective (3.1) only. The algorithms are suitably modified while implementing the remaining objectives.

3.5.1. MIP-based Decomposition Heuristic

The proposed MIP-based decomposition methods divide patients and staff into numbers of subgroups where patients of a subgroup can only be served by the staff assigned to that subgroup. Such subgroups can be formed by determining the feasible patient-procedure-caregiver assignments. However, due to the interaction between multiple patients and caregivers and scheduling and routing requirements on caregivers, finding the independent sub-groups may be as complex as solving the original problem. Therefore, an efficient approach for the formation of independent subgroups of patients and caregivers is developed by solving the Model **[M0]** while relaxing the scheduling and routing constraints. The resulting problem is a relaxed version of Model [**M0**] and is denoted as Model [**M1**], which is written as follows.

Model [M1]

Objective function (1)

Subject to

$$\sum_{i \in M} \sum_{p \in P} \sum_{u \in V} \left(\bar{t} + T_p\right) * y_{ipuk} + \theta \leq (\bar{\bar{T}}_s - \bar{T}_s) * S''_{ks} \qquad \forall\, k \in K, s \in S \qquad (3.62)$$

and Constraints (3.5 - 3.13), (3.37 - 3.46), (3.48 - 3.50) from Model **[M0]**.

The objective function for Model [**M1**] is the same as that of Objective (3.1) in the model [**M0**]. When solved, Model **[M1]** gives an upper bound for Model [**M0**], and the objective function value is termed as Quality Score (QS) for further discussion.

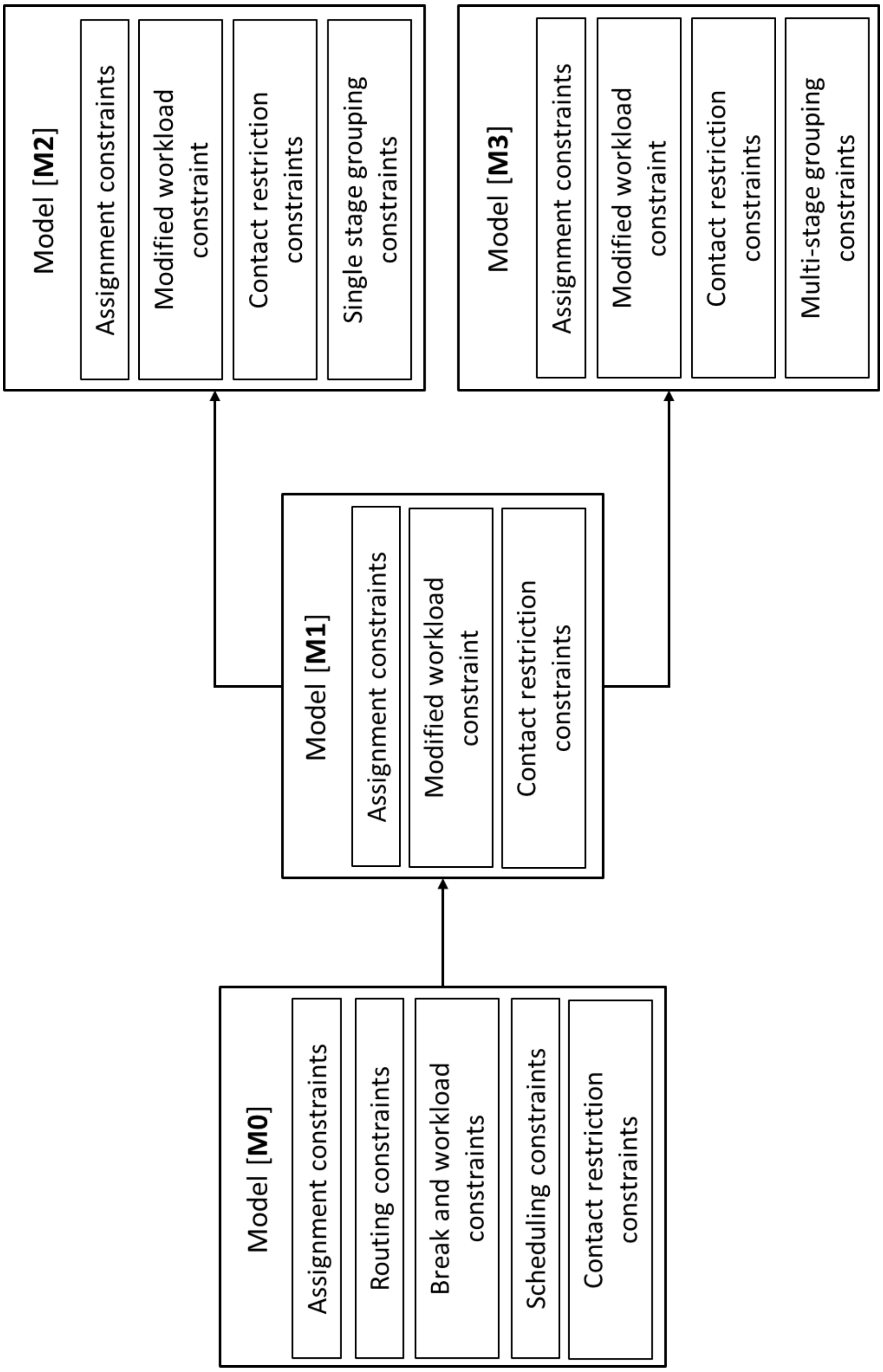


Figure 3.2: Framework for MIP-based decomposition strategies.

The constraint set for Model [**M1**] contains Assignment Constraints (3.5) to (3.13), Contact limiting constraints (3.37) to (3.41), and the nature of variables is the same as defined for Model [**M0**]. The work-load constraint (3.26) in Model [**M0**] has been suitably modified as a Constraint (3.62) that restricts the total number of assignments for particular healthcare staff to a predefined limit (number of visits that can be made while considering service time for visits and a generous travel time estimate). It also acts as a placeholder for scheduling and routing constraints. Model [**M1**] has been used for the MIP-based heuristic with appropriate decomposition strategies. In particular, we propose two decomposition strategies – Single Stage Decomposition (SSD) and Multi-Stage Decomposition (MSD). In Single Stage Decomposition, as implemented by Model [**M2**], the number of subgroups to be formed is exogenously determined based on the available computational budget and the problem size. In the case of Multi-Stage Decomposition, implemented by Model [**M3**], the subproblems are created by repeatedly dividing the problem into two parts till the subproblems created can be solved by the given computational budget. The schematic diagram of the decomposition schemes is represented in **Figure 3.2**. The solution quality obtained by single or multi-stage decomposition is further improved using local search.

3.5.1.1 Single-Stage Decomposition

The working of SSD is explained in **Figure 3.3**. The procedure starts by taking the inputs as the original problem modelled by **[M0],** the problem size of the original problem, and the maximum allowable size of the problem that can be solved by the solver at hand. The next step is to relax the original problem according to Model **[M1],** as explained in the previous subsection. The objective function value of **[M1]** will be considered as a quality score (QS) that is used as a target to be achieved by all the subgroups cumulatively. The number of subgroups formed by SSD depends upon the computational capability and available time. For example, if the available computational facility can solve up to 10 patient-procedure requests problem for Model **[M0]** to optimality, then the problem of 100 patient-procedure requests is divided into ten subgroups. In case of restriction on the overall time limit for the entire procedure, the SSD method can either be used by increasing the number of subgroups or placing a suitable time cap for solving the resulting subproblems. The latter strategy has been adopted in our case. It should be noted that the number of sub-groups formed is based on the total number of requested procedures. However, it neglects the maximum number of procedures that can be served with the given number and the compatibility of the staff, which is indicated by the QS. Therefore, in order to divide the number of achievable procedures equally among the subgroups, we define a Maximum Assignment Limit (MAL) for each subproblem. The MAL is the ratio of QS to the number of subgroups formed. The number of subgroups to be formed and MAL are used as inputs to Model **[M2],** which is solved in the next step to divide the original problem into the required number of subgroups. The model **[M2]** is defined using the following additional notation and constraints.

Model [M2]:

Additional sets, indices, and input parameters:

F	Set of subgroups indexed by f, F = {1, 2, …, f}
φ	Maximum assignment limit

Additional decision variables:

$$g_{if} \quad \begin{cases} 1, \text{ if patient } i \text{ is assigned to group } f. \\ 0, \text{ otherwise.} \end{cases}$$

g_{kf} $\begin{cases} 1, \text{ if healthcare staff } k \text{ is assigned to group } f. \\ 0, \text{ otherwise.} \end{cases}$

Objective function (1)

Subject to

$$\sum_{f \in F} g_{if} = p_i \qquad \forall\, i \in M \qquad (3.63)$$

$$\sum_{f \in F} g_{kf} = 1 \qquad \forall\, k \in K \qquad (3.64)$$

$$y_{ipuk} \leq \sum_{f \in F} \left(g_{if} * g_{kf} \right) \qquad \forall\, i \in M, p \in P, u \in V, k \in K \qquad (3.65)$$

$$\sum_{i \in M} \left(\left(\sum_{p \in P} q_{ip} \right) * g_{if} \right) \leq \varphi \qquad \forall\, f \in F \qquad (3.66)$$

Constraints (3.5 - 3.13), (3.37 - 3.46), (3.48 - 3.50) from Model **[M0],** and Constraint (3.62).

In addition to constraints from Model **[M1]**, **[M2]** also contains Constraints (3.63) to (3.66). Constraint (3.63) assigns each patient selected for service to exactly one subgroup. Constraint (3.64) does the same for each healthcare staff. Constraint (3.65) imposed a restriction that a patient belonging to a subgroup can only be served by staff belonging to the same subgroup. Finally, Constraint (3.66) is used for fairly distributing procedure-caregivers' assignments across the subgroups.

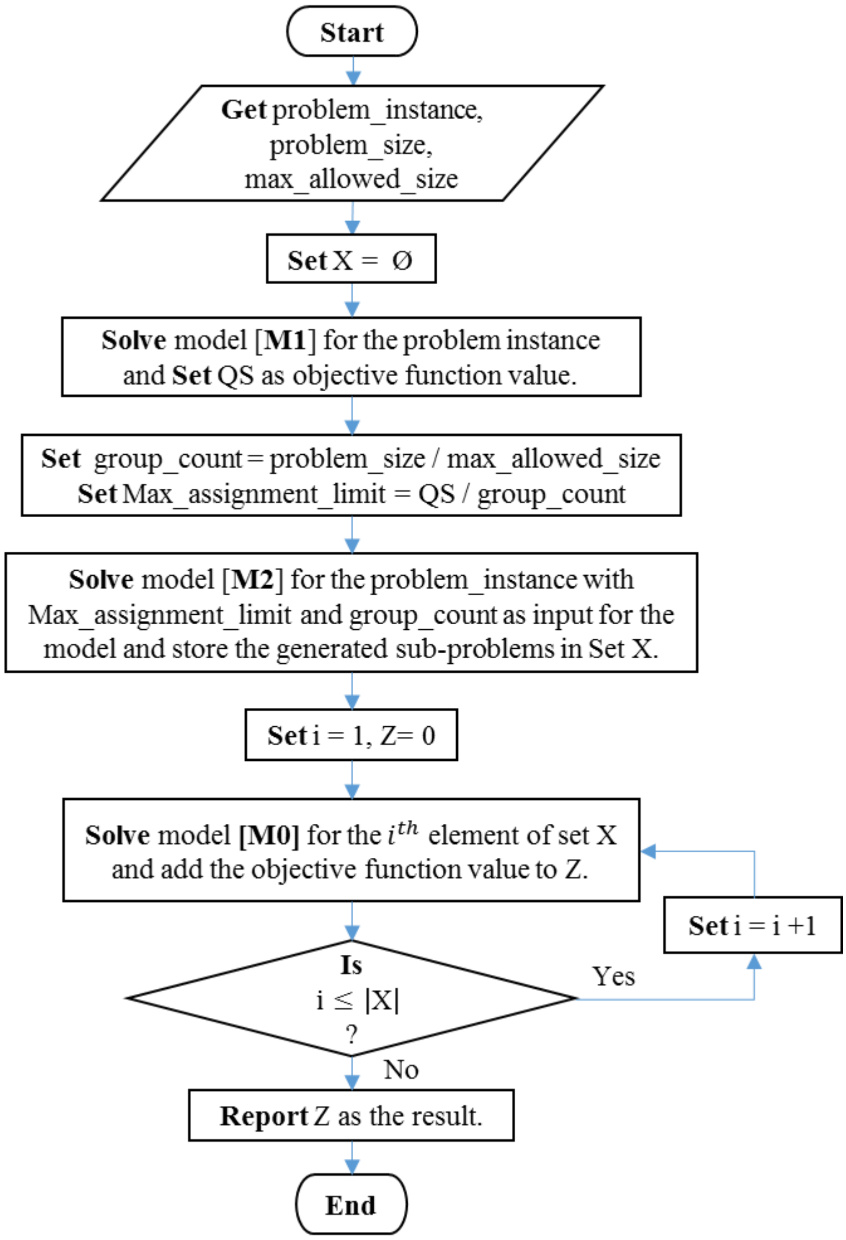


Figure 3.3: Flowchart – Single Stage decomposition.

MIP model [**M2**] is used to accomplish the decomposition of the problem in a single stage. Solution of **[M2]** gives the required number of subgroups or subproblems. The original problem **[M0]** is then individually solved for each of the subproblems obtained from **[M2]** to generate the assignment, scheduling, and routing decisions. The objective function values of the solved subproblems are added together and reported at the end of the procedure along with the complete solution.

Algorithm 1. Single Stage Decomposition

1: **Obtain** problem P_0.

2: Q0 ← **Solve** P0 for MIP model M1.

3: **Obtain** problem_size, max_allowed_size.

4: group_count = problem_size / max_allowed_size.

5: Max_assignment_limit = Q0 / group_count.

6: [X] ← **Solve** P0 for MIP model M2 (Max_assignment_limit, group_count).

7: **For** subproblems in [X]:

8:     **Solve** Xi for unified Model M0.

9:     **Compile** solution in {Z}

10: **End.**

3.5.1.2. Multi-Stage Decomposition

The working of MSD is presented in **Figure 3.4**. It can be observed that it shares the core concept with SSD. However, it differs from SSD in the procedure for creating the subgroups. Knowing the maximum allowable size of a problem that can be solved by the solver at hand, the MSD continues dividing the main problem into two parts in each iteration and thus reducing the problem size by half until the latest subproblem size is sufficiently small to handle by the solver. The mathematical model used to bifurcate the given problem instance in each iteration is given by Model **[M3]**.

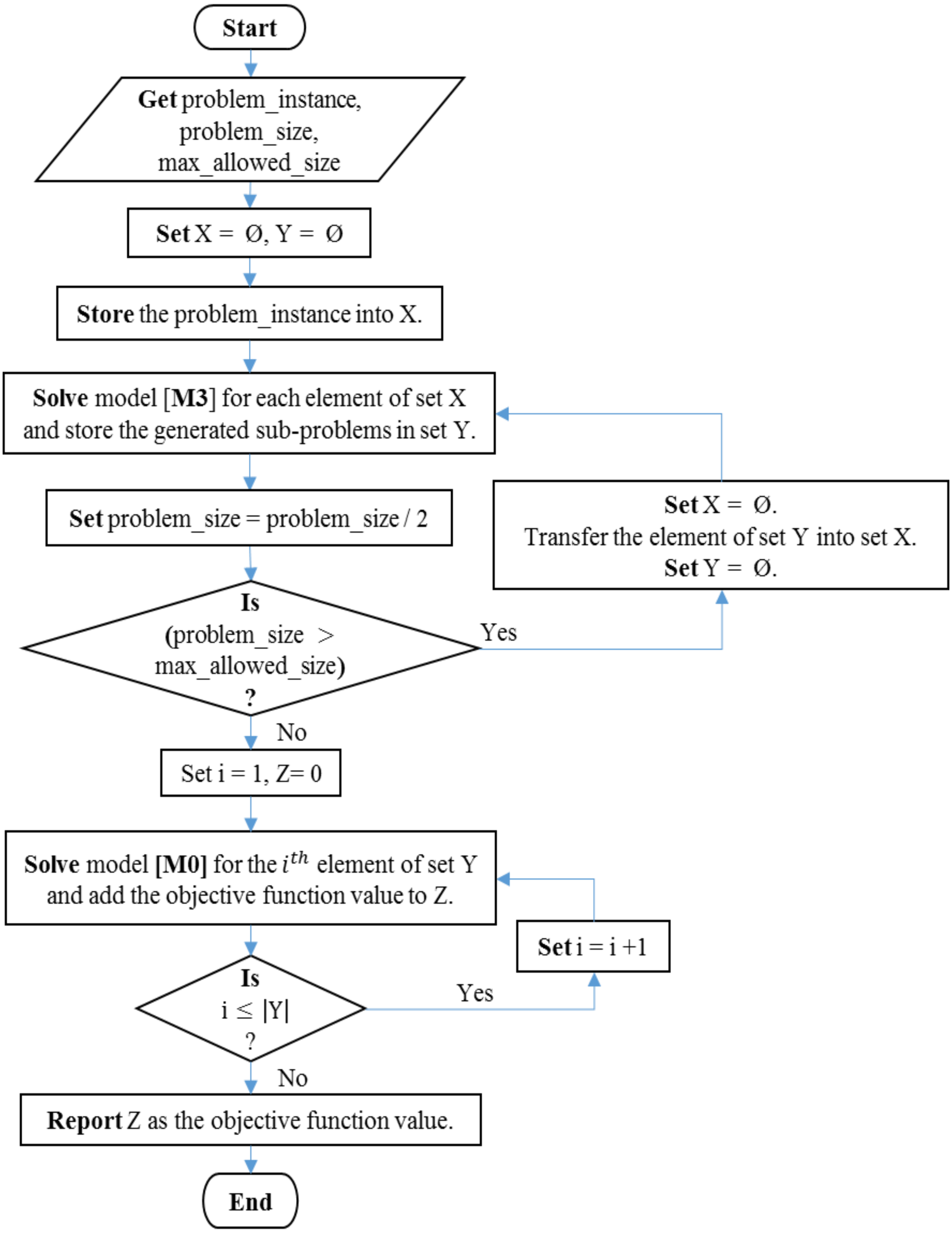


Figure 3.4: Flowchart – Multi-Stage decomposition.

Model [M3]:

Objective function (1)

Subject to

$$\sum_{i\in M}\left(\left(\sum_{p\in P} q_{ip}\right) * g_{i0}\right) - \sum_{i\in M}\left(\left(\sum_{p\in P} q_{ip}\right) * g_{i1}\right) \leq 1 \qquad (3.67)$$

$$\sum_{i\in M}\left(\left(\sum_{p\in P} q_{ip}\right) * g_{i1}\right) - \sum_{i\in M}\left(\left(\sum_{p\in P} q_{ip}\right) * g_{i0}\right) \leq 1 \qquad (3.68)$$

Constraints (3.5 - 3.13), (3.37 - 3.46), (3.48 - 3.50) from model **[M0]** and Constraint (3.62 - 3.65).

Model [**M3**] is a suitable modification of Model [**M2**] to achieve the multi-stage decomposition. Constraint (3.66) in Model **[M2]** is replaced with the Constraint (3.67) and (3.68) that ensure the equal distribution of the number of assigned procedures between the groups. These constraints limit the difference between the total number of assigned procedures in each group to one.

For MSD, Model [**M3**] is initially solved for the original problem instance that is stored in a set X. The solution produces two subproblems that are stored in a set Y. The division of the original problem into two halves also reduces the problem size by half. It is then checked whether the subproblem size is sufficiently small to be solved by solver. The procedure stops if the subproblem can be solved by the solver. Otherwise, set X is emptied, and subproblems stored in set Y are transferred to set X. The Model [**M3**] is solved for each subproblem in set X and produces further subproblems, and the loop continues. The Model [**M0**] is then individually solved for each of the subproblems, and the solutions are stored in set Y to generate the assignment, scheduling, and routing decisions. This will develop a complete solution to the original problem.

Algorithm 2. Multi-Stage Decomposition

Obtain problem P_0.
Obtain problem_size, max_allowed_size, I=0.
Repeat
    problem_size = problem_size / 2.
    I = I + 1.
Until (problem_size > max_allowed_size**)**
[X] ← **Solve** P0 for MIP model M3.
Repeat
    For element in [X]:
        [Y] ← **Solve** Xi for MIP model M3.
    Empty [X]
    [X] ← [Y]
    Empty [Y]
    I = I - 1.
Until (I > 0)
For subproblems in [X]:
    Solve Xi for unified model M0.
    Compile solution in {Z}
End

3.5.1.3. Local Search

Both SSD and MSD are heuristics procedures and cannot guarantee the optimality of the solution. Further, due to the formation of sub-groups of the patients and caregivers, there is a possibility that the solution will get stuck in a local optimum. To overcome this difficulty, we propose a local search procedure to further improve the solution obtained by the MIP decomposition procedures [**Figure 3.5**]. The local search is an iterative procedure that searches for opportunities to improve the solution in its neighborhood.

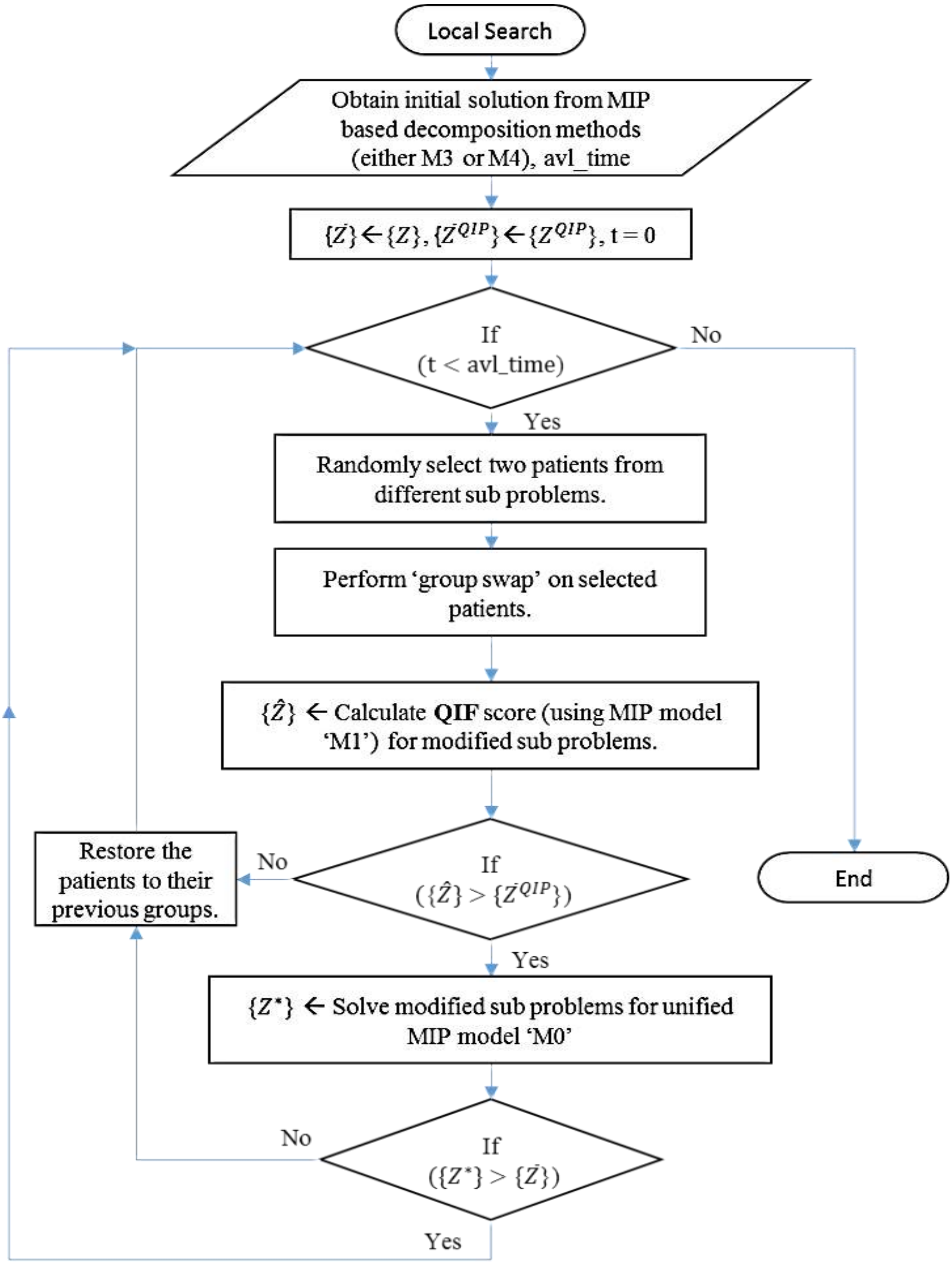


Figure 3.5: Flowchart – Local search algorithm.

In our case, the neighborhood for local search is defined by a "patient swap" operation that randomly selects two patients from different subgroups for swapping in order to move around the solution space. Such swapping changes the composition of the sub-groups from where the patients are selected for swapping while other sub-groups remain intact. The change in the objective function value, if any, can be found by solving the Model [**M0**] for the groups affected due to swapping. However, it might be time-consuming to run the model [**M0**] in case of every swap operation. Therefore, an efficient heuristic test is proposed to tackle this. The relaxed version of [**M0**], that is, Model [**M1**] for such sub-groups, is solved instead of Model [**M0**]. Model [**M1**] can be solved very quickly and provides an upper bound on the objective. If the cumulative objective function value of [**M1**] for the subproblems after the swap is better than before, the Model **[M0]** is solved for the sub-groups, and its objective function value is checked. The swap is only exercised if it improves the overall objective function value. Otherwise, the original sub-groups are kept intact, and the local search procedure continues until a stopping criterion (overall time limit or improvement limit) is met.

3.5.2. Rank-Based Decomposition Heuristic

The rank-based decomposition heuristic (RDH) utilizes the procedure compatibility, language, and gender preference between patients and healthcare staff to decompose large-size problems. The logic behind this method is to satisfy the patients with more common requirements (procedures that can be administered by most of the staff) by assigning them to the less capable staff, given that the staff is capable of performing the procedure required. By following this logic, more capable staff can be kept free for patients with highly specialized procedure requirements (procedures requiring fewer caregivers to be completed).

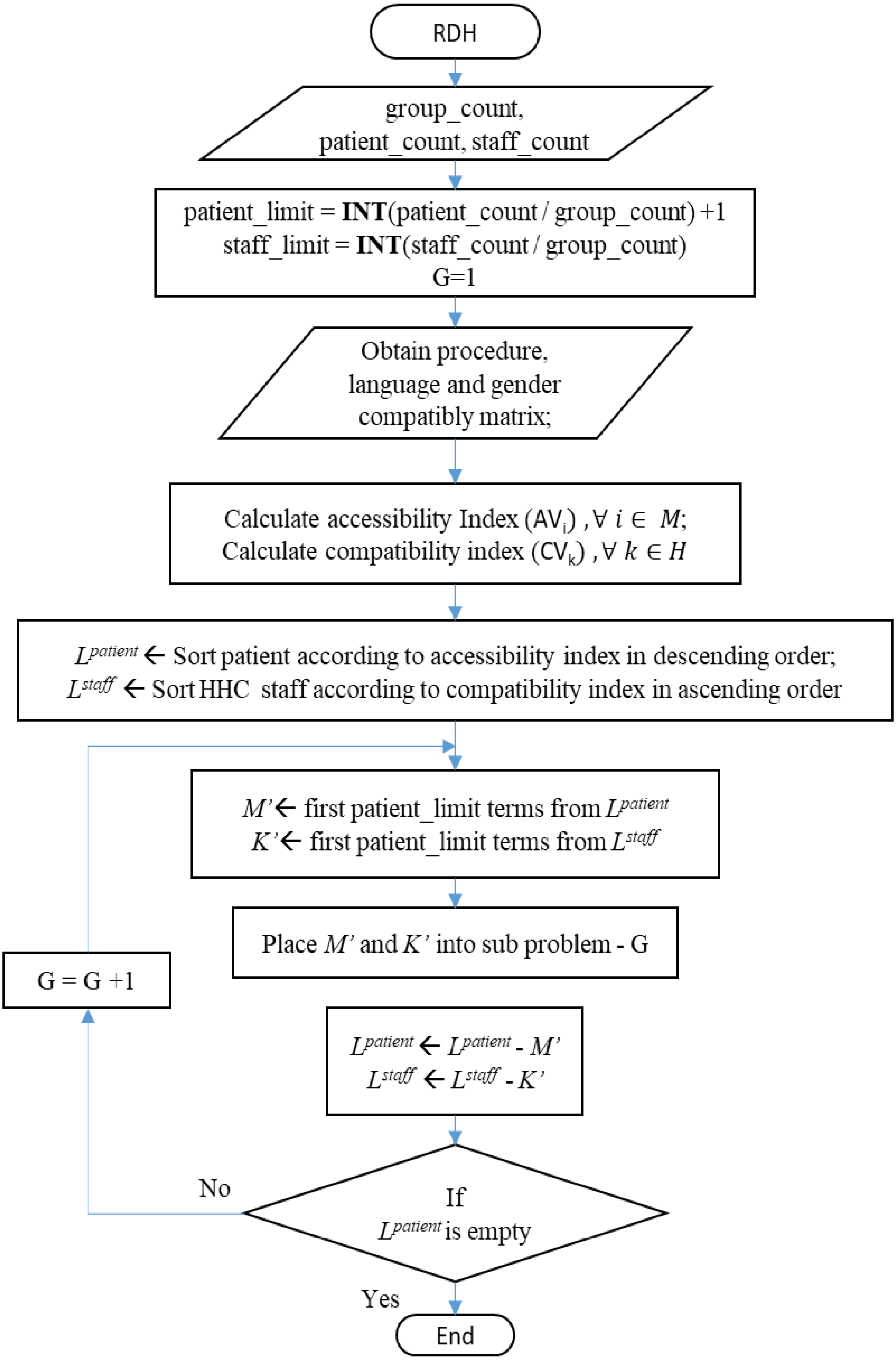


Figure 3.6: Framework – Ranked-based decomposition.

Steps relating to the RDH are provided in **Figure 3.6**. First, staff's Capability values (CV) are calculated based on how many given procedures they can perform while adhering to respective language and gender preferences. According to these CV values, the staff is sorted from least capable to most capable. Similarly, the patient's Accessibility value (AV) is calculated by adding up all the potential assignments that can be made for that particular patient. The average accessibility value for a patient is obtained by dividing the accessibility value by the total number of visits to be scheduled for that specific patient. Otherwise, patients with a higher number of visit requirements will show higher Accessibility value than patients with fewer visit requirements. Sub problems are generated by placing highly accessible patients with less capable staff, in turn leaving highly capable staff to be grouped with less accessible patients and improving the possibility of obtaining a better solution for Model **[M0]**.

In principle, more parameters (other than procedure compatibility and language and gender preferences) that either qualify or disqualify a staff to be assigned to a visit can be further implemented to improve the effectiveness of the algorithm. Distance from the hub can be used to disqualify some patients for the staff assigned to that hub based on the maximum travel distance criteria (if exist). Patients are most likely to be served if assigned to the hub closer to them rather than the hub far from them. Similarly, time overlap between a patient's preferred time window and staff shift can also be used to get better CV and AV estimates. Patients with a larger inconvenient time window in the morning shift will most likely be dropped from the assignments if assigned to a group mainly containing the morning shift staff.

Algorithm 3. Rank-Based Decomposition Heuristic

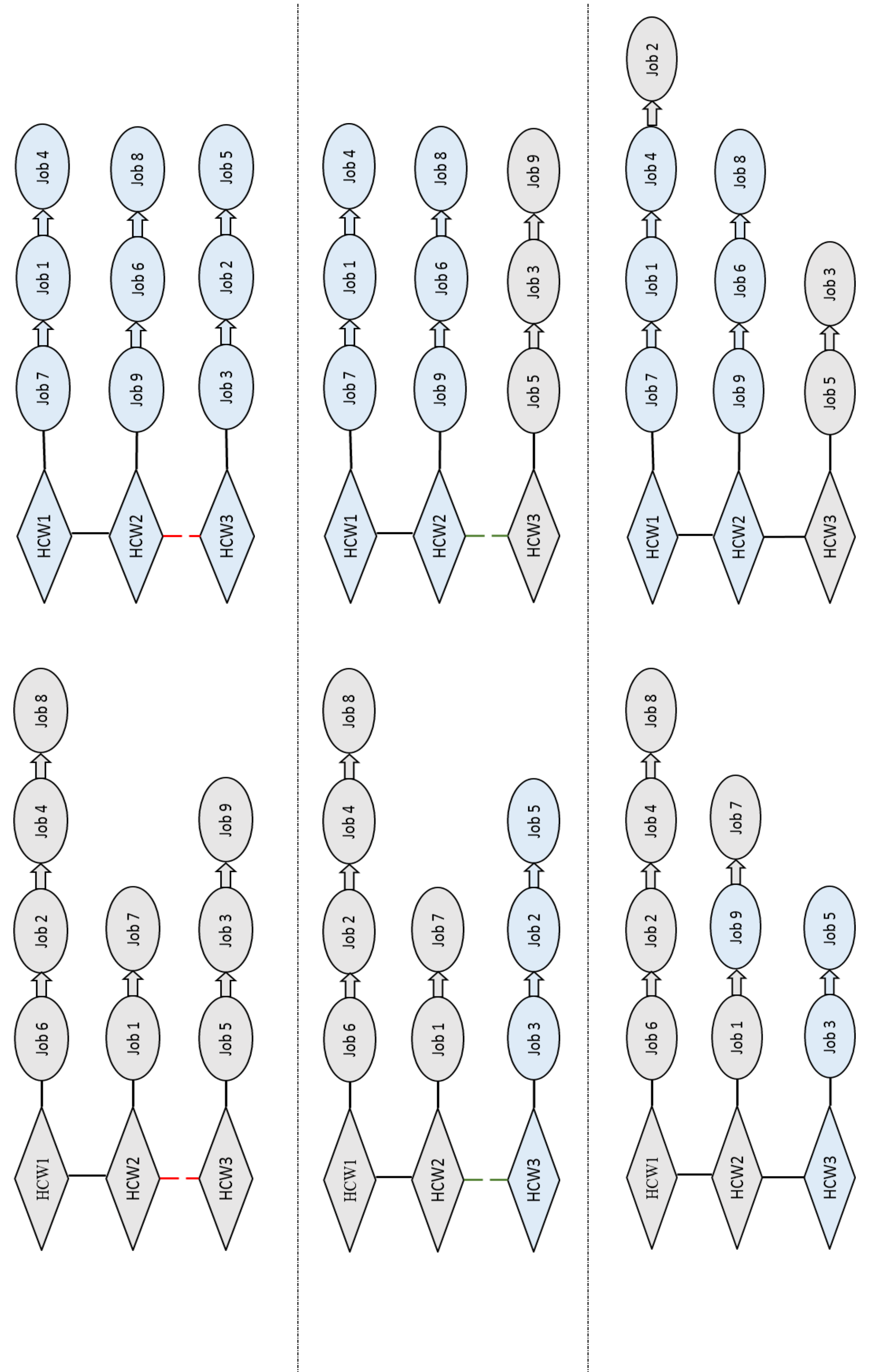
```
Obtain group_count, patient_count, & staff_count.
Calculate patient_limit & staff_limit.
Obtain procedure, language & gender capability matrix.
Calculate compatibility and accessibility Index.
L^patient ← Sort patients according to accessibility index in descending order.
L^staff ← Sort HHC staff according to compatibility index in ascending order.
Repeat
        Initiate new subproblem.
        Add first 'patient_limit' terms from L^patient to subproblem.
        Add first 'staff_limit' terms from L^staff to subproblem.
        Update L^patient & L^staff
Until (len (L^patient) = = 0 ))
```

3.5.3. *p*-Genetic Algorithm (*p*-GA)

Genetic algorithm (GA) utilizes the concept of evolution by natural selection to move through the solution space, evolving better and better solutions in every upcoming generation through selection, crossover, and mutation operators (Holland, 1992). A remarkably wide range of applications of GA explains its popularity in scientific literature and demonstrates its capability to find a good quality solution without getting entrapped in local optima. In the majority of the cases, the application of GA (i.e., the design of its operators) is straightforward. However, in the case of the HHC routing and scheduling problem, major concerns are the designing and interpretation of the chromosomes, evaluation of fitness function, and constraint handling with the minimum use of the computational resource. In our implementation, a caregiver-centric job-sequencing type of design is used to build the chromosomes, and a novel 3-step fitness evaluation technique that keeps track of constraint violation at every step is developed. Additionally, a parallel processing framework was also developed to help p-GA fully utilize the available computational resources. The steps of the proposed p-GA are explained in the following subsection.

3.5.3.1. Chromosome Design

A chromosome represents $|K|$ tours of caregivers, one for each caregiver, where each tour represents the sequence in which requested procedures by patients are carried out. The requested procedures are characterized by the patient, procedure type, and visit frequency and uniquely identified by 'Job' as shown in **Figure 3.7**. The genes of the chromosome are represented by the jobs, and their allele values are the corresponding job ID. The total length of the chromosome is given out by the summation of the product of each procedure, its visit frequency, and the number of caregivers required to perform it. A population consists of a finite number of chromosomes. In each generation, roulette wheel selection is performed to generate a mating pool, and then a single-point crossover is performed to generate the offspring in the next generation. A repair operator is used to modify the chromosomes in case of misplaced or repetitive genes. The Selection, Crossover, and Repair are performed as represented in **Figure 3.7**. We refer the readers to Deb (1999) for the technical details of the GA operators. In our case, the most important aspect of the GA is the constraint handling and the fitness function evaluation.



(a) **(b)** **(c)**

Figure 3.7: GA operators. (a) Representation of parent solutions with crossover point. (b) Crossover operator. (c) Repair operator.

3.5.3.2. Fitness Evaluation

To incorporate all the constraints in such a simple chromosome design and calculate the fitness score, a highly elaborated three-step fitness evaluation method is developed. In the first step, the tours of each caregiver are simulated with a virtual agent that tries to complete all the jobs assigned to it in the sequence as shown in the chromosome. The virtual agent starts its tour from its assigned hub and moves from one job to the next, adding travel time and service time for each consecutive stop and time-stamping every job. For jobs that require more than one staff, the staff that reaches the location first waits for the other staff to join. 'Start of service' is only time-stamped once every assigned staff is present at the location. After the 'end of service' is time-stamped, agents move to their next assigned visit. Before leaving for the next visit in the list, agents check whether a break or 'end of shift' is to be scheduled based on the time left in the agent's shift. Jobs left unvisited after the end of the shift are marked 'unvisited'. This step of fitness evaluation only takes care of the workload constraint, shift, lunch-break time, and the number of staff requirements. At this stage, the fitness function is not concerned with the violation of any other constraints.

In the next step, the fitness evaluation function chronologically goes through every stamped event and tracks the interaction between every person (staff and patient alike). As soon as the contact limit for an agent is reached, the remaining jobs from the tour of the agent are marked unvisited. A similar method is also used for patients to limit their contacts. Thus, the second stage imposes limited-contact restrictions. The remaining constraints are checked in the third and final phase. The third step considers only the jobs marked 'visited' at the end of the second stage and assigns a score of 10 points to each of them. The jobs in the tour of each caregiver are then checked for potential constraint violation and penalized if the violation is found. A ‘feasible’ tour for each caregiver will be represented by the jobs that have a perfect 10 score. If an intermediate job in a tour of a caregiver turns out to be infeasible, its corresponding time (service and travel) is to be treated as ‘waiting’ time in between the adjacent feasible jobs. The final fitness score is calculated based on the objective function and accommodation constraints. In the case of complete accommodation, the fitness score for a patient is only added to the overall fitness score if all the visits concerning that patient have a perfect 10 score.

3.5.3.3. Parallelism

Genetic algorithms, by design, have a strong potential to be implemented in parallel. Inspiration can be derived from nature, where the development of creatures (manifestation of genetic information) does not take place in a sequential manner but happens parallel to and independent of each other. While commercial MIP solvers are heavily optimized to utilize the commonly available multicore hardware, traditional GA works in a sequential manner and fails to wield the available processing power effectively. Some effort has already been made in the literature to modify GA to utilize the multicore architecture. Multiple instances of GA can be run in parallel on the different cores to reduce the random variation. In this approach, interaction between the populations of different runs is not allowed, and the best solution from the best solutions of the different runs is reported as the final solution. The method reduces the dependency of the quality of the solution on the randomly selected initial population without affecting the efficiency. Island-GA has also been used, where interaction between populations of isolated islands is only allowed after predefined generations or fitness values (Cantu-Paz & Goldberg, 2000). In our proposed parallel GA implementation (named p-GA), we do not run multiple instances of GA on different cores in parallel. Instead, we try to effectively utilize the available number of cores for the creation and fitness evaluation of different chromosomes in the same generation parallel. For example, if 60 cores are available for processing, the proposed p-GA approach develops and evaluates 60 chromosomes in a generation simultaneously and thereby reduces the computational time required. In our implementation of GA, selection, crossover, repair, and fitness evaluation for every individual of next-generation happens parallelly on a different core. The major focus of our multicore implementation of GA is on reducing the overall computational time requirement for each generation and thereby trying to increase the number of generations that can be evolved in a given time cap. This type of implementation of p-GA is novel in the literature. The concept of parallelism can also be used for the other steps in GA, such as initial population generation and mutation. However, the computational benefit of parallelism needs to be traded off with its implementation cost. For example, if the time required to distribute the chromosomes on different cores is more than carrying out mutation operation sequentially in a generation, there is no point in going ahead with the p-GA. Therefore, p-GA is beneficial for specific operators that demand high computational time and occur frequently. By using the multiprocessing library in python and high core count CPU, we were able to reduce the time requirement ten folds for large size problems, bringing p-GA on par with MIP-based methods.

3.6. Computational Experiment

The aim of our computational experiments is to compare the alternative models for HHC routing and scheduling and to evaluate the performance of different solution approaches proposed in this work. For this purpose, extensive experimentation is carried out by generating hybrid data sets and on the benchmark instances from the literature, as explained in the following subsections.

3.6.1. Experimental Settings

The dataset used for experimentation is termed as hybrid dataset for the reason that it is a combination of real and artificial data. Information regarding healthcare staff's location (hub), procedure capabilities, and list of available procedures are acquired from a home healthcare delivery start-up that was launched in 2015 in Mohali, India. In the first four years, the startup has clocked in over 7.5 million hours of caregiving and is now eyeing expansion across 20+ Indian cities. Due to confidentiality concerns, personal information regarding both patient and healthcare staff, such as patient's location, procedure requirements, and patient's and staff's shift, gender, and language preferences, are not shared with us; hence, they are generated randomly. Initially, 400 points are selected randomly on the map of Chandigarh city in India to represent the location of patients. The actual traveling distance between these locations is calculated using the BING Map by Microsoft. A list of 364 anonymous caregivers distributed over 52 distinct hubs with their procedure capability for the 40 different procedures has been made available to us. Based on the problem configuration, the required number of caregivers is randomly selected from this list with their procedure capability and hub location. The input parameters that are considered for the given data or generated randomly are summarized in **Table 3.9**.

To evaluate and fine-tune the capability of the solution approaches, we have designed an extensive testbed of problem instances. Twenty-one problem configurations with varying numbers of patients and caregivers were generated, as explained in the following. • Number of caregivers: 10, 15, 20, and 25 for medium-sized instances; 50, 75, and 100 for large instances. • Number of patients: 2 times, thrice, and four times the number of healthcare staff.

Table 3.9: Description of the range and values of various input parameters.

Parameters	Description	Values / Limits
Patient's preferences	Language	Hindi, English, and Panjabi
Inconvenient time window	Start and End time	6 am - 7 pm
Caregiver shifts	Morning shift	6 am - 2 pm
	Evening shift	11 am - 7 pm
Caregiver's characteristics	Language	Hindi, English, and Panjabi
Procedure's characteristics	Visit frequency	1 - 3
	Staff required	1 - 3
	Service time	5 - 45 minutes (in increments of 5)
	Revenue	100 - 500 rupees (in increments of 50)
Contact limits	Patient	5
	Caregiver	7

From all possible combinations of the number of caregivers and patients, we get 7 × 3 = 21 problem configurations. For each problem configuration, three problem instances (referred as 'sets') are generated randomly, resulting in 63 problem instances. Each instance signifies randomly selected values for locations of caregivers and patients, number of requested procedures, capability of caregivers, and patient preferences. Other parameters of the model, such as the total number of procedures offered, revenue for each procedure, and contact limits, remain the same across the problem instances. The problem instances are named as HHI A_B_C_D, where

A: Number of HHC staff available.

B: Number of patients available.

C: Size of procedure set; procedures available to be chosen from.

D: Set designation (1, 2 and 3).

Table 3.10: Size of the problem instances from Set_1.

Instance Id	Hub	Caregivers	Patients	Procedures	Variables		Constraints
					Continuous	Binary	
HHI 10_20_40_1	8	10	19	30	5.30E+04	1.19E+08	1.67E+08
HHI 10_30_40_1	6	10	29	33	8.90E+04	2.41E+08	3.47E+08
HHI 10_40_40_1	8	10	35	34	1.11E+05	3.86E+08	5.53E+08
HHI 15_30_40_1	12	15	27	37	1.38E+05	5.63E+08	7.90E+08
HHI 15_45_40_1	13	15	38	38	1.99E+05	1.02E+09	1.43E+09
HHI 15_60_40_1	12	15	58	40	3.20E+05	2.12E+09	3.04E+09
HHI 20_40_40_1	12	20	35	32	2.05E+05	8.16E+08	1.15E+09
HHI 20_60_40_1	15	20	52	40	3.81E+05	2.59E+09	3.67E+09
HHI 20_80_40_1	13	20	73	40	5.34E+05	4.27E+09	6.14E+09
HHI 25_50_40_1	15	25	48	37	4.05E+05	2.45E+09	3.46E+09
HHI 25_75_40_1	14	25	75	40	6.84E+05	5.71E+09	8.21E+09
HHI 25_100_40_1	15	25	91	40	8.30E+05	8.10E+09	1.17E+10
HHI 50_100_40_1	22	50	96	40	1.74E+06	2.01E+10	2.86E+10
HHI 50_150_40_1	26	50	133	40	2.41E+06	3.64E+10	5.22E+10
HHI 50_200_40_1	21	50	182	40	3.30E+06	5.94E+10	8.64E+10
HHI 75_150_40_1	34	75	141	40	3.82E+06	6.62E+10	9.43E+10
HHI 75_225_40_1	30	75	205	40	5.56E+06	1.19E+11	1.73E+11
HHI 75_300_40_1	29	75	281	40	7.62E+06	2.08E+11	3.03E+11
HHI 100_200_40_1	33	100	185	40	6.68E+06	1.37E+11	1.97E+11
HHI 100_300_40_1	36	100	278	40	1.00E+07	2.84E+11	4.12E+11
HHI 100_400_40_1	35	100	373	40	1.35E+07	4.80E+11	7.01E+11

Table 3.10 presents the complexity of the generated problem instances for Set_1. It shows the size of an instance based on the number of hubs, patients, caregivers, and procedures considered in that instance. Furthermore, it includes the number of variables (binary and continuous) and constraints needed to model the problem instance. All the Problem instances are made available online for further experimentation at the following link-
https://drive.google.com/drive/folders/1NGqY4PAzboPlpJd7Bxv7hFHSeVb8s6qS?usp=sharing.

Hardware/Software specifications:

All experiments are performed on a single CPU node (2* Intel Xeon SKL G-6148, 40 Cores, 2.4 GHz) of the PARAM SHIVAY supercomputing facility available at our institute. The total computational capability available on a single CPU node is approx. 3.8 TFLOPS. Despite its average capability, the high core count of this particular hardware gives us the ability to utilize p-GA's full potential. To solve various MIP models, a well-known state-of-the-art mathematical optimizer, GUROBI Optimizer, is used, which is also highly capable of utilizing the multiprocessing capability of our chosen hardware. Every experiment is run for 7200 seconds (2 hours). After the time limit is exhausted, the best solution found is reported as the result. For p-GA, an additional restriction of 2000 generation is also imposed; the algorithm will terminate if either the time or generation limit is reached. The results of the computational experiments are explained in the coming sections.

3.6.2. Algorithm selection

Initially, pilot experiments were carried out to determine the ability of the state-of-the-art MIP solver GUROBI 9.0.3 to solve model **[M0]** to optimality. For this purpose, tiny problem instances comprised of 2-7 caregivers were generated following the same pattern of variations as given above. The problem instances were solved with a solution time limit of 300 seconds. The result in terms of problem load time and actual solution time in seconds is shown in **Table 3.11**. It is clear from **Table 3.11** that the available computational capacity (both hardware and software) is vastly incapable of solving even smaller instances with five caregivers within the imposed time limit and cannot be relied upon for solving the larger and practical-sized instances. This demonstrates the need for specialized solution approaches. In what follows, we discuss the performance of the proposed matheuristics and the p-GA for the 63 problem instances generated, as explained in subsection 3.6.1. A time limit of 7200 seconds is imposed on all runs of the problem instances. The results of the three solutions approaches are summarized in **Table 3.12** with a sample result for one instance in each problem category.

Table 3.11: Limitations of commercial solver.

Staff count	Instance load time (second)			Solution time (second)		
	Patient count			Patient count		
	2×staff count	3×staff count	4×staff count	2×staff count	3×staff count	4×staff count
2	11	82	167	140	7	300+
3	66	412	1172	300+	300+	300+
4	167	1492	**7397**	300+	300+	300+
5	1908	**12911**	**-**	300+	300+	**-**
6	5949	**-**	-	300+	**-**	**-**
7	-	**-**	**-**	**-**	**-**	**-**

Table 3.12: Summary of the result obtained from different algorithms for set_1 instances.

Instance Id	Upper bound	SSD			MSD			p-GA					
								Best			Average*		
		Objective score	Gap (%)	Time (sec)	Objective score	Gap (%)	Time (sec)	**Objective score**	**Gap (%)**	**Time (sec)**	Objective score	Gap (%)	Time (sec)
HHI 10_20_40_1	59	41	43.90	4933	32	84.38	5177	**36**	**63.89**	**412**	35	68.57	414
HHI 10_30_40_1	67	56	19.64	2941	52	28.85	1969	**49**	**36.73**	**423**	48	39.58	424
HHI 10_40_40_1	86	55	56.36	2720	56	53.57	2657	**52**	**65.38**	**471**	51	68.63	480
HHI 15_30_40_1	68	53	28.30	5629	49	38.78	2903	**40**	**70.00**	**449**	39	74.36	456
HHI 15_45_40_1	79	66	19.70	4396	63	25.40	3830	**56**	**41.07**	**494**	52	51.92	510
HHI 15_60_40_1	123	86	43.02	5144	84	46.43	3455	**73**	**68.49**	**627**	68	80.88	665
HHI 20_40_40_1	56	43	30.23	3772	43	30.23	3255	**38**	**47.37**	**463**	35	60.00	472
HHI 20_60_40_1	105	76	38.16	5284	72	45.83	4196	**60**	**75.00**	**643**	59	77.97	671
HHI 20_80_40_1	171	120	42.50	7466	118	44.92	5762	**99**	**72.73**	**974**	93	83.87	1040
HHI 25_50_40_1	107	51	109.8	7342	83	28.92	2900	**66**	**62.12**	**759**	65	64.62	815
HHI 25_75_40_1	185	97	90.72	7448	129	43.41	8003	**105**	**76.19**	**1022**	103	79.61	1091
HHI 25_100_40_1	205	145	41.38	5252	119	72.27	6261	**119**	**72.27**	**1152**	115	78.26	1214
HHI 50_100_40_1	244	_	_	3344	144	69.44	7219	**115**	**112.17**	**1962**	114	114.04	2134
HHI 50_150_40_1	326	_	_	4329	156	108.97	7575	**158**	**106.33**	**2555**	154	111.69	2778
HHI 50_200_40_1	420	_	_	_	_	_	_	**180**	**133.33**	**4025**	178	135.96	4371
HHI 75_150_40_1	_	_	_	_	_	_	_	**188**	_	**3401**	181	_	3700
HHI 75_225_40_1	_	_	_	_	_	_	_	**207**	_	**6235**	196	_	6747
HHI 75_300_40_1	_	_	_	_	_	_	_	**231**	_	**7201**	220	_	7202

The identifiers of the problem instances are given in Column 1 of **Table 3.12**. Column 2 lists the upper bounds for the problem instances obtained by solving the relaxation of Model **[M0]** as explained in Section 3.5. For each solution approach, the best result obtained, the percentage gap (the difference in upper bound and the best solution obtained), and the CPU time in seconds is reported. It is observed that in the case of SSD, GUROBI Optimizer was able to solve problems with up to 20 healthcare staff and 80 patients within 7200 seconds, with a time limit of 500 seconds for the decomposition step and **min** {**500, time-remaining/group-count}** seconds limit for solving each group obtained. No significant improvement was observed by increasing the time allotted for SSD's decomposition step from 500 to 1000 seconds or removing time limits for solving each obtained group. In the case of MSD, GUROBI was able to solve significantly larger problem instances that up to 50 caregivers and 100 patients with the limit of 500 seconds for each decomposition stage of MSD and keeping **min {500, time-remaining/group-count}** seconds limit for solving each group obtained from the decomposition step. Like the SSD, MSD did not show any improvement with the increase in the time limits.

For p-GA, five independent runs of the algorithm were used to account for the inherent randomness in the algorithm. Therefore, both the average and the best values of the objective function and CPU time obtained from 5 runs of p-GA are reported in **Table 3.12**. The p-GA was able to solve (generate a feasible solution) for all the instances within the time limit. While the algorithm was able to converge for smaller instances (up to 25 healthcare staff and 100 patients) under the time limit, it failed to do so for larger instances. In some cases (instances with 75 staff and 300 patients and larger), p-GA did not manage to complete 2000 generations and was forced to terminate prematurely. Nevertheless, the p-GA was able to generate feasible solutions for every problem instance.

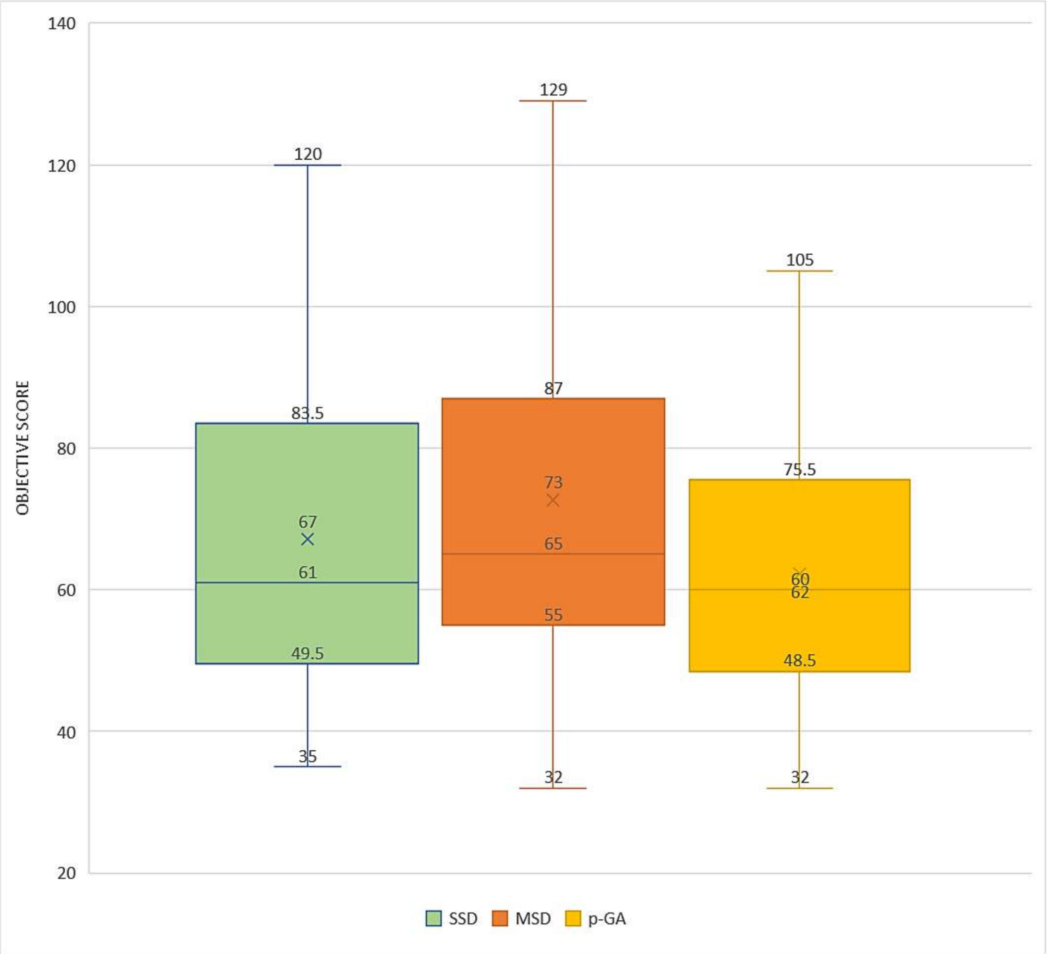


Figure 3.8: Box and whisker plot comparing the solution quality of the algorithms.

The results of all 63 problem instances with different solution approaches are given in **Appendix B.** To compare their performance, a box and whisker plot is generated. **Figure 3.8** is drawn considering only the instances that are solved by all the solution methods within the imposed time limit. **Figure 3.8** indicates that the median value from MSD is slightly better than that from SSD and p-GA. The second and third-quartile comparison of the plots hint at the superiority of MSD over the other approaches. However, no conclusive remark can be made regarding the superiority of one algorithm over another from the comparison of the individual box plots. Therefore, an additional statistical analysis is conducted to determine whether there is a significant difference in the results obtained from different algorithms. First, the assumption of normality was established by successfully performing the Kolmogorov-Smirnov test on data obtained by each method individually. With all the assumptions met, a positive result was obtained (*p*-value < 0.0001) by one-way repeated measures ANOVA at the significant level of 0.05. Finally, paired samples *t*-test (post-hoc test for one-way repeated measures ANOVA) was conducted with $\alpha = 0.05$, the results of which are presented in **Table 3.13.** The statistical results show that the MIP-based decomposition methods are significantly different as compared to p-GA. This is intuitive due to the different philosophies and design of the algorithms. From the t-values reported in **Table 3.13**, it can be noted that the SSD and MSD perform better than p-GA. Further, no significant difference was observed in the quality of solutions obtained by SSD and MSD algorithms. Here, it is necessary to understand that these conclusions are only valid for small-size problem instances (up to 25 healthcare staff and 75 patient problems) for which all algorithms were at least able to produce feasible solutions in all three sets.

Table 3.13: Result of paired samples *t*-test.

	p-value			*t*-value		
	SSD	MSD	p-GA	SSD	MSD	p-GA
SSD	-	0.050	0.006	-	2.033	-2.947
MSD	0.050	-	< .00001	-2.033	-	-8.272
p-GA	0.006	< .00001	-	2.947	8.272	-

SSD: Single-stage decomposition; MSD: Multi-stage decomposition.

With "reliability" defined as the capability of an algorithm to produce feasible solutions for a greater number of problem instances, p-GA comes out to be the most reliable algorithm. Although the MIP-based matheuristics outperform p-GA in terms of solution quality for small problem instances, p-GA was able to solve all the instances within the considered time limit. Additionally, p-GA is computationally efficient and requires significantly less time than the MIP-based decomposition methods. Finally, it was the only algorithm that could solve larger and practical size instances in the given time limit. Hence, the p-GA is considered to be the only viable algorithm for all further experimentations.

3.7. Experimentation on benchmark problem instances

One of the objectives of this work is to develop a *generic* mathematical model for the integrated problem of assignment, routing, and scheduling for home healthcare delivery. The benefit of the generalized model is that it can work in a wide variety of problem settings and specific contexts with only the appropriate parameter tuning without any structural changes in the model. Therefore, the model and the proposed solution approach presented in this work can be applied in different contexts. To demonstrate this, we test our mathematical model on the benchmark problem instances used by Fikar & Hirsch (2014) and Fikar, Juan, Martinez, & Hirsch (2016). The problem set, as described in Fikar & Hirsch (2014), consists of three variations on the number of patients and the number of caregivers. The number of patients considered are 75, 100, and 125, whereas the number of caregivers considered are 24, 32, and 40, respectively. The caregivers are further divided according to three different levels of qualifications: 1, 2, and 3, where 3 indicates the most qualified caregiver. Five instances for each problem size are provided for the two distinct geographical areas, urban and suburban. The distinction is primarily based on the travel distances between the patient's locations. A total of 30 problem instances ($3 \times 2 \times 5$) are generated from the above-mentioned variations. Additionally, the qualification level of the caregiver, preferred time window, and the service time for each patient are provided for the problem instances separately.

Table 3.14: Modification needed to adapt our work to a benchmark instance set from the existing literature.

Problem aspect	Requirements	Modifications
Agent-related constraints	1. 3-level of skill qualifications. 2. Only one level of downgrading is allowed. 3. No gender preference was specified.	a) Modify L' and L'' according to **Table *3.15*** to accommodate skill level requirements. b) Set G'= [1, 1, 1] for every patient to make Constraint (3.12) inactive.
Break and workload	1. Maximum workload =10 hours. 2. Break needed after 6 hours.	a) W_K = 600 (minutes). b) $\bar{\vartheta}_s$ = 240, $\bar{\bar{\vartheta}}_s$=360
Time window	1. Visits should be scheduled between T^{start} and T^{end}.	a) Constraints (3.26) to (3.28) have to be implemented twice. b) Once with $\bar{T}_i'$= 0 and $\bar{\bar{T}}_i'$= T^{start}. c) Again with $\bar{T}_i'$= T^{end} and $\bar{\bar{T}}_i'$= 600.
Precedence constraints	Not required	D_{pq} = 0, *for all p, q.*
Contact restriction constraints	Not required	λ'=100, λ''=100.
Travel time	Travel time by car or by walking is provided.	Travel time by car is used.

Table 3.15: Parameter values to incorporate the skill level in presented mathematical models.

Patient related		Caregiver related	
Required qualification	L'	Possessed qualification	L''
1	[1, 0, 0]	1	[1, 0, 0]
2	[0, 1, 0]	2	[1, 1, 0]
3	[0, 0, 1]	3	[0, 1, 1]

It should be noted that we do not intend to solve the models proposed by Fikar & Hirsch (2014). Instead, we only use the dataset used by these researchers to test the performance of our proposed model. The necessary modifications required to adopt our mathematical model for the chosen problem instances are presented in **Table 3.14** and **Table 3.15**. The model with the new parameters is solved using the p-GA algorithm for 3600 seconds. The p-GA is run five times for every instance, and the best and average values of the number of patients served, number of staff used, and unproductive time are reported in **Table 3.16.**

It can be seen from **Table 3.16** that the p-GA performs quite well with respect to the objective of maximizing the number of patients served. In the case of 26 out of 30 problem instances, it serves 100% of the patient's request with the available resources. A service level of 99% and above has been reached for the remaining four instances. As evident from **Figure 3.9**, the model presented by Fikar & Hirsch (2014) did perform better in regards to resource utilization and used fewer caregivers to achieve the same result. On the other hand, our model seems to perform better in regards to 'unproductive work', but this is mainly due to the sharing aspect of the problem, which is considered in Fikar & Hirsch (2014) work but not included in our model. Caregiver waiting time will definitely increase if caregivers have to wait for a common vehicle to travel from one place to another, increasing the total unproductive work. Even with these drawbacks, this experiment proves the suggested generality and the ease of adoption of our work for a wide variety of problem cases.

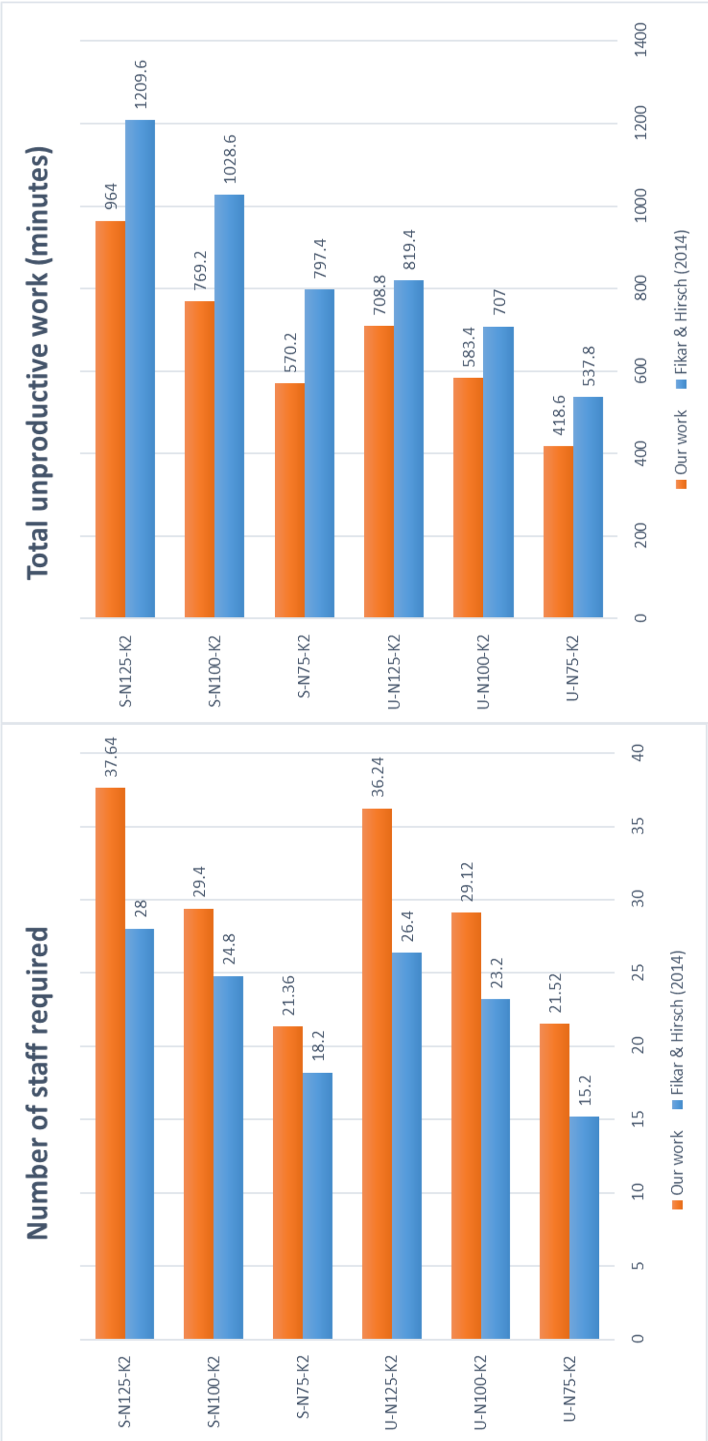


Figure 3.9: Comparison with benchmark instances.

Table 3.16: Result obtained for the benchmark instances using p-GA.

Instance ID	Patient served		Staff used		Unproductive work (minutes)	
	Best	Average	Best	Average	Best	Average
U1-n75-k2-m12-8-4	75	74.4	19	20.6	427	463.4
U2-n75-k2-m12-8-4	75	75	19	21.6	409	417.2
U3-n75-k2-m12-8-4	75	74.6	19	22	432	461.8
U4-n75-k2-m12-8-4	75	74.4	20	21.8	423	444.2
U5-n75-k2-m12-8-4	75	74.4	20	21.6	402	432.6
U1-n100-k2-m16-10-6	100	99.4	29	30.4	557	617.6
U2-n100-k2-m16-10-6	100	99.2	28	29.2	567	648.4
U3-n100-k2-m16-10-6	100	99	26	28.4	562	608.4
U4-n100-k2-m16-10-6	**99**	98.2	28	29.4	610	664.4
U5-n100-k2-m16-10-6	**99**	98.4	26	28.2	621	671.4
U1-n125-k2-m20-12-8	125	123.2	34	36.4	703	831
U2-n125-k2-m20-12-8	125	123.6	33	35.4	692	817.8
U3-n125-k2-m20-12-8	**124**	121.6	35	36.2	747	913.8
U4-n125-k2-m20-12-8	125	123.2	35	36.6	708	840.6
U5-n125-k2-m20-12-8	125	124.2	35	36.6	694	747.2
S1-n75-k2-m12-8-4	75	73	20	20.6	582	659.8
S2-n75-k2-m12-8-4	75	74.4	19	20.8	564	626.8
S3-n75-k2-m12-8-4	75	74.6	20	22.4	567	611.4
S4-n75-k2-m12-8-4	75	74.2	20	21.6	565	641.2
S5-n75-k2-m12-8-4	75	74.2	18	21.4	573	637.8
S1-n100-k2-m16-10-6	**99**	98.4	29	29.8	815	888.4
S2-n100-k2-m16-10-6	100	98.8	28	29.4	755	844.2
S3-n100-k2-m16-10-6	100	99.2	25	27.6	755	816.2
S4-n100-k2-m16-10-6	100	99.4	30	31.2	753	784.8
S5-n100-k2-m16-10-6	100	98.6	28	29	768	879.2
S1-n125-k2-m20-12-8	125	123.6	37	38	952	1038.6
S2-n125-k2-m20-12-8	125	122.6	37	37.8	963	1089.6
S3-n125-k2-m20-12-8	125	124.2	37	38	948	1003
S4-n125-k2-m20-12-8	125	123.6	36	37.6	990	1073.6
S5-n125-k2-m20-12-8	125	123.8	36	36.8	967	1034.4

3.8. Comparison of patient accommodation policies

As discussed in Section 3.2, there are two major policies for serving the patients: complete accommodation and partial accommodation. In the case of a complete accommodation policy, all the requested procedures of a patient need to be completed if the patient is selected for HHC delivery. On the other hand, in case of partial accommodation, a subset of requested procedures can be performed for a patient. Under these two policies, the objective can be either maximizing the number of patients served or maximizing the total revenue generated. Thus, the four objective functions are formulated as Equations 3.1 to 3.4 in the mathematical model presented in Section 3.3. To get the insights from these objectives, the problem instances were run independently on each objective. The results obtained are summarized in **Table 3.17** and **Table 3.18** for partial and complete accommodation policies, respectively.

For each objective, total revenue, number of fully served patients, total number of requests served, and processing time are reported in **Table 3.17** and **Table 3.18**. It should be noted that the results presented are the best-known values out of 5 runs of p-GA on each instance. There is no significant difference in CPU time reported for different objectives. However, the results show a considerable variation in the case of the other three attributes, namely revenue, number of patients, and procedures completed. The following approach is adopted for each problem instance for a clearer understanding of the results. Firstly, the value obtained for each attribute across all four objectives is normalized following a min-max normalization method. This eliminates any scaling issue within different attribute values and places all the scores between 0 and 1. A utility score is then calculated for each objective using an additive utility function that assigns an equal weightage to the attributes. A graph of these utility scores for all problem instances is shown in **Figure 3.10**.

Table 3.17: Summary of results under complete accommodation restriction for set_1 instance.

Complete accommodation policy								
Instance ID	**Objective – 3.3**				**Objective -3. 4**			
	Total revenue	Patient count	Request served	Time taken	Total revenue	Patient count	Request served	Time taken
HHI 10_20_40_1	1850	4	10	411	2550	5	14	410
HHI 10_30_40_1	5050	8	17	421	4400	8	19	419
HHI 10_40_40_1	5100	10	22	479	5950	11	22	466
HHI 15_30_40_1	3450	4	12	451	2350	3	10	444
HHI 15_45_40_1	3550	8	17	508	5400	9	26	489
HHI 15_60_40_1	7550	13	33	619	7200	14	29	602
HHI 20_40_40_1	5450	14	21	461	5600	15	24	449
HHI 20_60_40_1	6350	13	25	610	7300	15	33	611
HHI 20_80_40_1	6850	14	31	1023	6200	14	34	971
HHI 25_50_40_1	6850	12	29	644	6950	12	32	689
HHI 25_75_40_1	8200	18	37	1058	9500	19	43	1006
HHI 25_100_40_1	11400	28	53	1172	15850	31	57	1106
HHI 50_100_40_1	7300	16	40	1947	7150	17	37	1857
HHI 50_150_40_1	10400	25	57	2613	13950	30	66	2379
HHI 50_200_40_1	12850	30	64	4194	14600	31	72	3838
HHI 75_150_40_1	12500	27	66	3666	13150	28	71	3369
HHI 75_225_40_1	13250	30	83	6692	14000	30	73	6055
HHI 75_300_40_1	17500	40	96	7201	17250	38	87	7201
HHI 100_200_40_1	14900	37	84	5527	16700	36	84	5016
HHI 100_300_40_1	17750	37	99	7201	19750	39	102	7200
HHI 100_400_40_1	20800	53	125	7201	17900	45	108	7202

Table 3.18: Summary of the results under partial accommodation relaxation for set_1 instance.

Instance ID	Partial accommodation policy							
	Objective -3.1				Objective -3.2			
	Total revenue	Patient count*	Request served	Time taken	Total revenue	Patient count*	Request served	Time taken
HHI 10_20_40_1	10950	3	33	413	12000	4	33	413
HHI 10_30_40_1	13400	7	44	425	14800	7	47	425
HHI 10_40_40_1	17250	9	52	481	18850	9	54	481
HHI 15_30_40_1	13450	3	37	455	14700	4	40	455
HHI 15_45_40_1	18550	8	51	511	20350	9	54	512
HHI 15_60_40_1	22950	12	69	632	25550	11	72	641
HHI 20_40_40_1	13400	14	37	462	12750	14	35	462
HHI 20_60_40_1	17950	11	56	622	19500	14	59	616
HHI 20_80_40_1	30150	13	93	1032	31000	13	92	1031
HHI 25_50_40_1	22550	13	67	711	22500	13	66	756
HHI 25_75_40_1	28400	18	99	1066	30350	22	99	1063
HHI 25_100_40_1	36600	27	117	1171	37050	29	119	1172
HHI 50_100_40_1	34400	15	112	2086	36800	17	113	2070
HHI 50_150_40_1	46950	27	155	2748	50800	26	155	2657
HHI 50_200_40_1	55850	28	176	4332	60200	28	183	4339
HHI 75_150_40_1	56600	26	179	3726	59050	26	176	3691
HHI 75_225_40_1	65350	24	203	6733	69450	24	209	6711
HHI 75_300_40_1	67250	28	219	7201	73600	37	219	7200
HHI 100_200_40_1	66800	39	211	5631	73150	40	221	5703
HHI 100_300_40_1	67100	27	213	7202	71800	26	216	7201
HHI 100_400_40_1	71200	31	227	7201	79550	35	240	7201

*Number of patients with all requests served in spite of the partial accommodation policy.

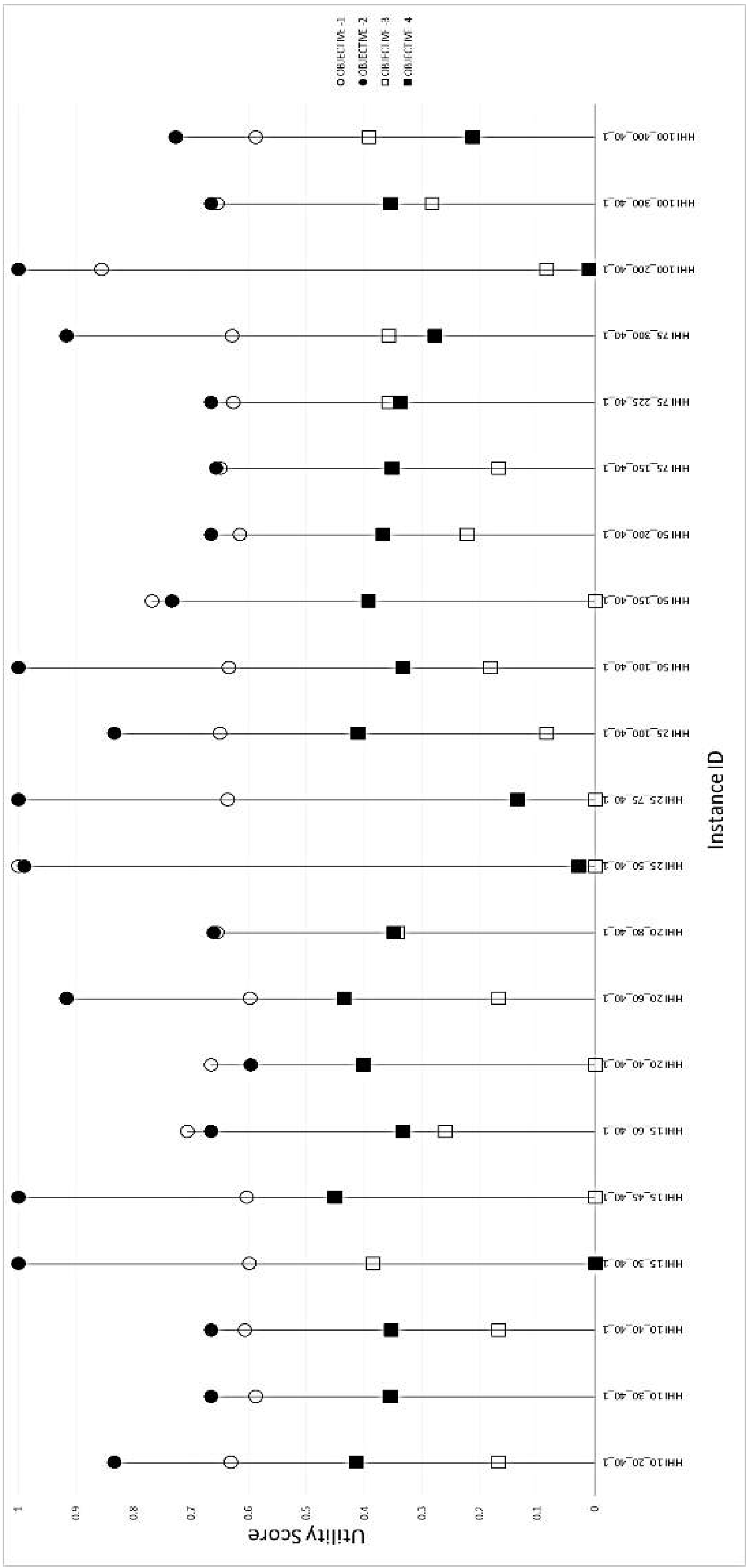


Figure 3.10: Comparison of utility score under different objective functions.

Figure 3.10 gives an interesting interpretation of the results obtained. It shows that the partial accommodation policy outperforms the complete accommodation policy based on the utility value. For example, the number of requests served and the total revenue generated in the partial accommodation strategy are multifold compared to their respective values under complete accommodation restrictions. On the other hand, no significant difference was observed in the total number of fully served patients. As a result, the utility scores are better for Objectives 3.1 and 3.2 (partial accommodation policy) as compared to Objectives 3.3 and 3.4 (complete accommodation policy) in all the cases [**Figure 3.10**]. Further, it can be seen that the revenue maximization objectives give better utility score value as compared to procedure/patient maximization objectives, irrespective of the policy. In fact, for some of the problem instances, it is found that the number of patients fully served under the patient maximization objective is less than the number of patients served under the revenue maximization objective. This is also found to be true under the partial accommodation policy. This is a counter-intuitive observation that can possibly be justified by the following reasons. As the patients/procedures are unweighted in the case of patient/procedure maximization (Objective 3.1 and 3.3), it is difficult to find a substantial improvement in the fitness value of the genetic algorithm with an incremental increase in the number of patients/procedures.

In contrast, when the actual revenue associated with the patients/procedures is considered, the p-GA performs better due to significant difference in the fitness values of the chromosomes. Further, a revenue maximization objective gives preference to a more profitable selection of the procedures, which can be seen in some of the problem instances. Once again, it should be noted that these observations are based on the best values reported for 21 problem instances. For a broader analysis, we considered the average value of the attributes taking into account all 63 problem instances. The results are shown in **Figure 3.11**.

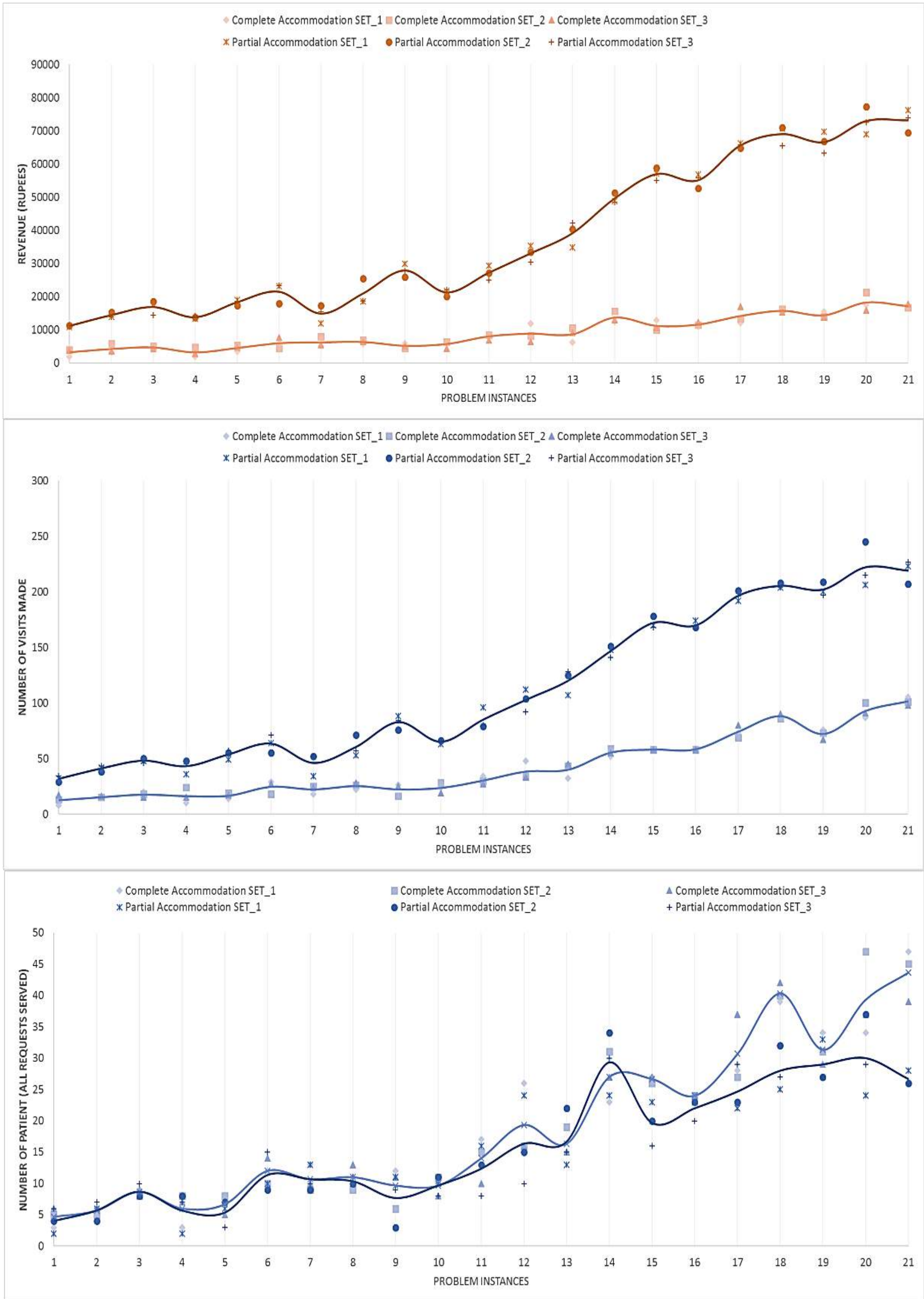


Figure 3.11: Comparison of objective function values under different accommodation policies.

By compiling the scatter plots for different sets into one graph, as shown in **Figure 3.11**, and plotting the average values for each policy separately, it can be observed that when optimized for revenue, the partial accommodation strategy gives the most favorable results. The partial accommodation policy results in around 300% improvement in revenue generation at a cost of a 12% decrease in the number of patients that are fully served. Similar results can also be observed when maximizing resource utilization under the allotted visit criteria. This marks a 173 % increase in the number of allotted visits, with only a 12% decrease in completely accommodated patients. In comparison, priority to the full accommodation of patients does not significantly impact the results, as shown in part 3 of **Figure 3.11**. The complete results for all 63 problem instances are provided in **Tables B.7** to **B.12**.

Additional experiments are carried out to estimate the total capacity needed to ensure the full demand satisfaction for all the patients. A pool of 800 part-time staff is generated in the same manner described in Subsection 3.6.1 to be used in addition to the available staff for the current problem instances. A relaxation in workload restriction is also made for this part of the experiment; all the caregivers (regular as well as part-time) can now be scheduled for four additional hours over the regular 8-hour shift. With complete demand satisfaction of all the requests made as a strict requirement, the objective of the problem changes to minimization of the total regular and overtime staff used. As a result of this, the availability of the staff is no longer a parameter and calls for an additional decision variable. This brings a change in the objective function as well as some of the constraints in the Model **[M0]**. Similarly, the proposed p-GA cannot be directly applied to solve the problem.

We have proposed an ad-hoc heuristic to efficiently solve the modified problem of minimizing the staff used.

To keep the thesis concise, the necessary modifications to the mathematical model and the ad-hoc heuristic are not presented in the main chapters and are provided in **Appendix A**. With the modified problem setting, computational experiments are carried out, and the results containing the number of regular and part-time staff employed, the number of staff scheduled for overtime, and additional capacity used (in hours) have been reported in **Table 3.19**. It should be noted that **Table 3.19** only includes the results of the first set of problem instances in each configuration. Similarly to the modified model and ad-hoc heuristic, the complete results for all three sets of instances are presented in Table **B.13** to **B.18** as a part of the Appendix.

From **Table 3.19**, it can be seen that the total number of staff required to fulfill all the requests made is at least three times more than the existing staff. This requirement is even more for the larger instances. With the goal of reducing the total staff needed to fulfill all the requests, the model has chosen more capable and versatile part-time staff over the regular staff that may be less capable and restricted by their shifts. Similarly, part-time staff are overwhelmingly chosen for the overtime work. Finally, by observing the last columns, it can be seen that, on average, three times more work hours are needed to maintain the complete request satisfaction requirement. There is no wonder that the policy of full demand satisfaction would also indicate a maximum revenue that can be earned. However, this policy needs to be reviewed from the perspective of resulting profits.

Table 3.19: Analysis of staff utilization.

Instance ID	Staff used			Staff On overtime		Work hours		
	Total	Regular	Part-time	Regular	Part-time	Initially available	Additional	Total
HHI 10_20_40_1	32	9	23	1	12	80	236	316
HHI 10_30_40_1	34	10	24	2	11	80	244	324
HHI 10_40_40_1	37	9	28	2	14	80	288	368
HHI 15_30_40_1	34	11	23	4	15	120	260	380
HHI 15_45_40_1	43	11	32	3	15	120	328	448
HHI 15_60_40_1	65	11	54	0	17	120	500	620
HHI 20_40_40_1	39	19	20	5	9	160	216	376
HHI 20_60_40_1	58	17	41	2	28	160	448	608
HHI 20_80_40_1	80	18	62	4	29	160	628	788
HHI 25_50_40_1	55	19	36	1	19	200	368	568
HHI 25_75_40_1	82	22	60	1	37	200	632	832
HHI 25_100_40_1	81	22	59	7	27	200	608	808
HHI 50_100_40_1	118	39	79	11	46	400	860	1260
HHI 50_150_40_1	146	44	102	7	58	400	1076	1476
HHI 50_200_40_1	213	45	168	7	85	400	1712	2112
HHI 75_150_40_1	198	69	129	10	55	600	1292	1892
HHI 75_225_40_1	272	65	207	11	112	600	2148	2748
HHI 75_300_40_1	340	67	273	20	123	600	2756	3356
HHI 100_200_40_1	279	93	186	11	63	800	1784	2584
HHI 100_300_40_1	369	88	281	14	126	800	2808	3608
HHI 100_400_40_1	436	95	341	20	165	800	3468	4268

Table 3.20: Financial analysis.

INSTANCE ID	CASE – A[1]				CASE – B[2]				
	Revenue collected	Regular staff wage	Profit[3]	Profit/ Wage	Revenue collected	Additional wage[4]	Total wage	Profit[3]	Profit/ Wage
HHI 10_20_40_1	12000	4363	7637	1.75	28150	8592	12955	15195	1.17
HHI 10_30_40_1	14800	4266	10534	2.47	31350	9574	13840	17510	1.26
HHI 10_40_40_1	18850	3830	15020	3.92	43650	11414	15244	28406	1.86
HHI 15_30_40_1	14700	5932	8768	1.48	33600	9995	15927	17673	1.10
HHI 15_45_40_1	20350	5665	14685	2.59	45400	12814	18479	26921	1.45
HHI 15_60_40_1	25550	5640	19910	3.53	64450	18917	24557	39893	1.62
HHI 20_40_40_1	12750	7887	4863	0.62	29750	7789	15676	14074	0.89
HHI 20_60_40_1	19500	7892	11608	1.47	53800	16810	24702	29098	1.17
HHI 20_80_40_1	31000	7942	23058	2.90	84700	25145	33087	51613	1.55
HHI 25_50_40_1	22500	9764	12736	1.30	54600	13896	23660	30940	1.30
HHI 25_75_40_1	30350	10055	20295	2.02	79100	24708	34763	44337	1.27
HHI 25_100_40_1	37050	9794	27256	2.78	92350	23031	32825	59525	1.81
HHI 50_100_40_1	36800	19757	17043	0.86	118550	34347	54104	64446	1.19
HHI 50_150_40_1	50800	19530	31270	1.60	152350	43584	63114	89236	1.41
HHI 50_200_40_1	60200	20187	40013	1.98	210300	69643	89830	120470	1.34
HHI 75_150_40_1	59050	29697	29353	0.99	162900	55311	85008	77892	0.91
HHI 75_225_40_1	69450	29389	40061	1.36	255150	89732	119121	136029	1.14
HHI 75_300_40_1	73600	29142	44458	1.53	311750	114948	144090	167660	1.16
HHI 100_200_40_1	73150	39691	33459	0.84	204300	75482	115173	89127	0.77
HHI 100_300_40_1	71800	39425	32375	0.82	325350	118818	158243	167107	1.05
HHI 100_400_40_1	79550	39687	39863	1.00	408050	146505	186192	221858	1.19

[1] Revenue maximization under partial accommodation (Objective 2).
[2] Full demand satisfaction by employing the part-time staff as well and allowing overtime for all the employees.
[3] profit = revenue collected – wage (travel allowance for staff are not included).
[4] Additional wage = Hiring cost for part-time employees + overtime cost.

For this purpose, a financial analysis of the results obtained is performed and presented in **Table 3.20**. Staff wages have been taken in proportion to their procedure capability over the procedure set with the rate for full-time and part-time in the ratio of 5:4; that is, part-time staff with the same capability as regular staff will only be paid 80% of salary for the same work, regular as well as overtime work. Additionally, overtime compensation for all caregiver has been set at twice the rate compared to their wages for regular work shifts. Total generated revenue, regular staff wages, net profit, and the ratio of profit to total wages are presented in **Table 3.20** for two cases – revenue maximization under the partial accommodation policy (Objective 3.2) and the full demand satisfaction policy.

It can be seen in **Table 3.20** that profit does necessarily increase with the introduction of part-time staff and overtime. But this requires additional investment in the form of staff wages, which might not be possible in every situation. Additionally, the excess staff required on a full-time/part-time basis might not be available whenever required. Further, some of the expenses, such as travel passes for the part-time workers, can be part of the expenses and not accounted for in these calculations. The profit might even go negative under the higher fixed costs and overheads related to hiring the employees. Finally, from the ratio of profit to expenses (total wages), it is clear that the instances have a lower profit-to-expenses ratio under the full demand satisfaction policy, indicating its lower potential to turn a dollar spend into profit. So, a rule of thumb can be extracted from this analysis that the healthcare service provider should only hire additional capacity in terms of part-time staff or overtime work in case of excess idle capital.

3.9. Conclusions

In this chapter, we have developed a generalized MIP model for home healthcare delivery capable of routing heterogeneously skilled staff from multiple hubs in multiple shifts while considering several patient-staff compatibility criteria. The model is also capable of scheduling and routing jobs that may require more than one visit from more than one staff. The "inconvenient time window" and "limited contact restriction" considered in the present work are novel contributions to the existing literature. Further, the generality of the proposed model was successfully tested by adopting it for the benchmark instance set from the literature. Four different objective functions were tested to find the most suitable policy regarding the demand fulfillment of a selected patient. With revenue, total number of requests served, and number of completely served patients as the necessary aspect of the solution quality, "partial accommodation" was found to be the best strategy for an HHC delivery problem with limited staff availability. Additional experiments were also conducted to assess the financial viability of acquiring additional resources in terms of part-time staff and/or overtime. Maximum use of the idle capital was found to be the best strategy if a sufficient number of caregivers were available.

On the solution side, two matheuristics are developed to decompose the large problems to be solved by the MIP solver. A novel version of GA named p-GA is proposed. A clear framework was also presented and implemented for the use of parallel processing to reduce the solution time in the implementation of p-GA. While MIP-based decomposition methods were able to find better solutions for many small and medium problem instances, p-GA was the only method that managed to produce feasible solutions for all the instances. The result was validated by performing a statistical test on the quality of the solution obtained by the three algorithms for the smaller problem instances. It has been concluded that while SSD should be preferred for small-medium instances, p-GA will be most effective while dealing with large-size problems.