Applied Mathematics and Computational Intelligence
Volume 14, No. 2, 2025 [1 – 18]

Improved Tabu Search Method in Solving Overbooking Appointment Scheduling with No-shows Patient

Nazeelda Zukernain[1], Nor Aliza Abd Rahmin[1*], Athirah Nawawi[1]

[1]Universiti Putra Malaysia, Jalan Universiti 1, 43400 Serdang, Selangor Darul Ehsan, Malaysia

* Corresponding author: aliza@upm.edu.my

Received: 19 November 2024
Revised: 6 February 2025
Accepted: 11 April 2025

ABSTRACT

No-shows patient refer to instances where individuals either do not attend their scheduled appointments or cancel at the last minute, resulting in a missed opportunity for the health facility to utilize that time slot. This can lead to both time and financial losses for the facility, disrupting patient care. By having an efficient appointment schedule, these disruption can be overcome by minimizing resource idle time, resource overtime and patient waiting time. This research aims to enhances appointment scheduling by addressing overbooking through a heuristic approach, further refined by the tabu search method. The impact of scheduling multiple patients in the same time slot is examined to determine the optimal number of patients per slot for cost optimization. This problem is addressed using the C programming language. The findings indicate that the tabu search method slightly outperforms the heuristic approach, especially when dealing with larger patient datasets. Other than that, it is proven that the tabu search method functions effectively by having a long-term memory as it executes the programmer faster than previous methods such as genetic algorithm and simulated annealing. Besides, tabu search method is capable in improving the maximum number of patients that can be effectively assigned to the same time slot.

Keywords: Heuristic Procedure, Multiple Assignment, No-Show Patient, Overbooking, Tabu Search Method

1 INTRODUCTION

Upon the advent of each new generation, diverse challenges manifest within the healthcare sector. In [1], it is discovered that Malaysian faces health difficulties where citizens do not have enough access to public health care providers in spite of a good coverage of healthcare services. This arises issues gives negative impact to the demand for healthcare services, either it is from the patient's perspective, doctor's perspective or others.

Due to that, appointment scheduling system are used by hospitals to monitor and manage appointments. Appointment scheduling can enhance the utilization of services and facilities' medical resources while reducing patient wait times. It also aims to build an appointment system that optimizes a specific quality standard in a healthcare application. [2] state that arrival and service time variability, patient and provider preferences, available information technology and the experi-

ence level of the scheduling staff are the factors which can affect the performance of appointment systems.

Within the healthcare sector, the main cause of poor performance in appointment scheduling is the patient no-show behaviour. [3], the overall proportion of patients who did not attend has been found at 30.1 in an Obstetrics and Gynecology clinic. [4] also claimed that a total of 31 no-show rate for MRI screening has been announced. Based on [3] and [4] research, low utilization and productivity of medical resources, a decrease in revenue and an increase in total costs have been some of the negative impacts of patient absence. This behaviour impacts patient waiting times, resource idle time, and resource overtime.

To delve deeper into the issue, the appointment scheduling process involves patients making appointments, with each slot entered the system. On the appointment day, patients check in and wait in designated areas. However, if a patient fails to show up, it disrupts the schedule, causing longer waiting times for those who do arrive. This situation affects patient and doctors experiences. Doctors, with limited time slots for each patient, experience increased idle time when patients do not show up. This idle time translates into unproductive periods during regular working hours, and doctors may find themselves waiting for patients rather than attending to others. Furthermore, missed appointments lead to doctors working beyond regular hours, requiring overtime.

Thus, dealing with patients who miss their appointments leads to longer patient waiting times, idle periods, and increased overtime. Much research has observed this problem by using different methods such as genetic algorithm and simulated annealing. However, the Tabu Search method which has not been applied in this problem is selected in this study to address the issue. TS is known for its long-term memory, enabling it to remember optimal solutions, explore past solutions, and helps in reducing the time needed to solve the problem and finding the best solutions faster. Appointment scheduling effectiveness can be enhanced by minimizing patient waiting times, idle periods, and overtime.

2 LITERATURE REVIEW

It will describe the studies that consider overbooking appointment scheduling as a problem. It comprises two sections. The section 2.1 will discussed on the effects of no-show patient in overbooking appointment scheduling. Meanwhile, section 2.2 explained on the various methods used on solving the overbooking appointment scheduling problem.

2.1 Patient No-show

Patient no-show has been described as the real overbooking appointment scheduling problem. Some patients may miss their appointments or arrive late.

[5] investigated the overbooking scheduling problem and proposed two-stage stochastic mixed-integer linear programming. Patient no-shows, which were assigned to different time slot structures, were considered in finding the optimal solution to overbooking. The effectiveness of three types of time slots in healthcare, which are fixed-length, dome-pattern, and flexible appointment start times, is examined. As a result, it is found that the most effective type of time slot is the fixed-length slot

interval, as it outperforms the others by giving flexible appointment start times.

In the same year, [6] observed overbooking and patient no-shows by focusing on specialty clinics. This research found that missed appointments could result in significant congestion of patients needing to be seen. The aims of this research are to increase the productivity of care providers, enhance patient accessibility to care, and improve clinic appointment scheduling. To achieve these objectives, a novel discrete-time bulk service queue has been developed to model the dynamics of patient accumulation. By examining the trade-off between the expected and actual backlog, it is possible to reduce backlogs with a lower risk of requiring overtime. The proposed model is capable of enhancing operational efficiency, patient outcomes, and patient satisfaction.

In the following year, [7] proposed a new strategy called real-time sequencing. By combining appointment scheduling with real-time sequencing algorithms, this research aims to reduce overall costs, including the weighted sum of patient waiting expenses and provider overtime costs. To achieve these objectives, an optimal real-time sequencing strategy is identified as the "smallest larger of the appointment time and the real arrival time" (LAR) strategy. This strategy aims to minimize provider idleness and reduce patient waiting time. Through the implementation of this model, it is demonstrated that the smallest LAR strategy should be adopted instead of appointment order (AO) or first-in-first-out (FIFO). This is because the performance of AO tends to worsen the problems, while FIFO potentially encourages unpunctuality.

At a cardiology clinic, the overbooking scheduling problem was investigated by [8]. This research proposed a machine learning (ML) algorithm that identifies key characteristics for predicting no-shows and zero consultation length. Additionally, the relationship between ML-based predictions and overbooking was explored. However, due to the limitations of ML algorithms, a CRISP-DM methodology was introduced. This methodology was used to build an ML-based two-part model, consisting of a "stochastic gradient boosted classification tree (SGBCT)" and a "deep neural network regressor (DNNR)." This two-part model has been shown to outperform the clinic's existing approach, achieving a reduction of 56% and 52% in patient waiting time and idle time, respectively, with further improvements assumed in appointment scheduling.

In another study published by [9], they developed a stochastic mixed-integer linear programming (SMILP) framework to address the overbooking scheduling problem. This study concentrates primarily on patient no-shows, requests, and refusals, as well as service duration. In contrast to previous research, this study focuses on the tradeoffs between effective scheduling and expeditious service access. Using the devised framework, an optimization of a simulation model's session appointment schedule is to be attained. The convergence of SMILP, the implications of two-dimensional uncertainty on the appointment scheduling system, and the benefits of optimal appointment scheduling were among the topics examined. Utilizing this methodology, practitioners can determine the optimal level of overbooking at which resource overtime increases and patient wait times are permissible. This study revealed that dynamically adjusting session capacity based on the current appointment lead time may not be advantageous if the patient request rate is fixed. Additionally, it is demonstrated that the dynamic capacity policy performs better in a non-homogeneous request rate environment because the resource level and patient demand are more closely aligned.

2.2 Optimization Methods

In this study, optimization methods are employed to enhance appointment scheduling within the healthcare sector.

[10] investigated the patient no-show condition with various no-show probabilities and different weights, as part of an overbooking model for scheduled arrivals. The aim of this study is to minimize the expected workload for patients' waiting time and a doctor's on duty hours, as well as overtime. A number of considerations have been taken into account, in particular the initial static problem where a group of patients is being scheduled and their traits are known earlier. A new sequence rule was also introduced in which patients were placed according to the same index that is influenced by their characteristics. In such cases where requests for appointment are increasing over time, a heuristic solution has been suggested in which every patient is assigned to on of the remaining slots available within that day's schedule. Consequently, the absence rate and patients' heterogeneity are considered to be important factors in determining an optimal schedule.

[11] aims to identify the maximum number of patients that can be assigned to a time slot by examining the effects of multiple assignments and to construct a near-optimal overbooking appointment scheduling. A heuristics procedure and genetic algorithm are proposed to make a comparison of effectiveness. It is proven that the multiple assignment condition with m = 3 is the best option to reduce healthcare cost. The best result from the heuristics approach will be improved by the genetic algorithm, which has been shown to perform better than the heuristics procedure in solving the problem.

[12] proposed the use of simulated annealing to improve the overbooking appointment scheduling problem, aiming to minimize resource idle time, resource overtime, and patient waiting times. Initially, a heuristic method is applied to establish a starting point, which is then utilized in the proposed simulated annealing approach. The findings indicate that simulated annealing successfully achieves better results and outcomes compared to the heuristic method, particularly when applied to an extensive dataset of patients.

In the year 2023, [13] proposed a two-phase TS algorithm. In view of the new college entrance examination reform, they intended to propose the method to address scheduling issues at Chinese high schools. When the time approaches for students to take the college entrance exam, the schools offer a system of elective classes to help them prepare. However, this has a negative effect on students' time slots and makes it more difficult to create a suitable schedule from the available options. Some constraints which were considered are "teaching plans synchronously", "no idle periods in the timetables of teachers" and others. The characteristics of the graph colouring model has been applied in the model. The motive of the application is to remove unnecessary calculations in the neighbourhood search procedure and to increase computational efficiency. In order to test the efficacy of the algorithm, fifteen practicals with varying scales were selected. Consequently, the proposed algorithm can generate a high-quality available schedule in a brief amount of time and the average satisfaction rate of constraints exceeded 71%.

In the same year, an improved tabu search (ITS) algorithm was introduced by [14]. They intend to resolve heterogeneous fixed fleet open vehicle routing problems with time constraints, one of the most recent transportation issues. ITS uses a modified sweep algorithm to generate some

initial solutions to employ the algorithm. In addition, a variable tabu list and new intensification and diversification mechanisms are employed. Several results demonstrate the efficacy of the ITS, including the fact that it is more effective than the exact algorithm and TS algorithm and is capable of locating suitable solutions. In addition, it is discovered that the proposed ITS algorithm is more efficient and stable than TS, simulated annealing (SA), ITS without intensification, and the 2-opt method for solving medium- and large-scale problems. Due to that, the intensification mechanism is proven to improve the obtained solutions. At the end of this study, exact algorithms can be used for minor problems, while meta-heuristic algorithms such as ITS can be used for more significant problems.

3 MATHEMATICAL MODELING

Each notations, parameters and overbooking model are referred from [11].

3.1 Notations

Table 1 : Sets

N	Set of patients from 1 to n to be scheduled for the session
i	Index of patient
J	Set of time slots in the session
j	Index of time slot
S	Set of scenarios
s	Index of scenario
N_s^1	Set of patients who show up at the healthcare unit under scenario s
N_s^2	Set of patients who do not show up at the healthcare unit under scenario s

3.2 Parameters

Table 2 : Parameters

b_j	Beginning time of time slot j
d_{is}	Service duration for the i^{th} patient under scenario s
w^{ot}	Penalty for each unit of resource overtime
w^{wait}	Penalty for each unit of patient waiting time
w^{idle}	Penalty for each unit of resource idle time
E	Close time for healthcare facility

The beginning time b_j of each slot j is fixed as pre-defined time slots are being used with the overbooking model that is investigated. A fixed time interval has been set for each time slot.

3.3 Decision Variables

$$x_{ij} = \begin{cases} 1 & \text{if the } i^{th} \text{ patient in the schedule is assigned to the } j^{th} \text{ time slot} \\ 0 & \text{otherwise} \end{cases}$$

Table 3 : Decision variables

i	Index of patient
a_i	Appointment time of the i^{th} patient in the schedule
z_{is}^{start}	Start time of the medical service provided to the i^{th} patient under scenario s
z_{is}^{end}	End time of the medical service provided to the i^{th} patient under scenario s
$wait_{is}$	Waiting time for the i^{th} patient for receiving the medical service under scenario s
$idle_{is}$	Idle time of the resource between i^{th} and the $i+1^{th}$ services under scenario s
ot_{is}	Overtime of the resource under scenario s
x_j	Number of patients assigned to the j^{th} time slot
m	Maximum number of patients which can be assigned in one time slot

3.4 Model

$$\min\ w^{ot}\sum_{s\in S} ot_{is} + w^{idle}\sum_{i\in N, s\in S} idle_{is} + w^{wait}\sum_{i\in N_s^1, s\in S}\frac{wait_{is}}{N_s^1}$$

Subject to

$$a_i = \sum_{j\in J} b_j x_{ij},\ \forall i \in N \tag{1}$$

$$\sum_{j\in J} x_{ij} = 1,\ \forall i \in N \tag{2}$$

$$a_{i+1} \geq a_i,\ \forall i \in N\backslash\{n\} \tag{3}$$

$$z_{1s}^{start} = 0,\ \forall s \in S \tag{4}$$

$$z_{is}^{start} = a_i + wait_{is},\ \forall i \in N,\ s \in S \tag{5}$$

$$z_{is}^{start} = z_{(i-1)s}^{end} + idle_{(i-1)s},\ \forall i \in N\backslash\{1\},\ s \in S \tag{6}$$

$$z_{is}^{start} + d_{is} = z_{is}^{end},\ \forall i \in N, s \in S \tag{7}$$

$$z_{ns}^{start} + idle_{ns} - ot_s = E,\ \forall s \in S \tag{8}$$

$$z_{is}^{start}, z_{is}^{end}, wait_{is}, idle_{is}, ot_s \geq 0,\ \forall i \in N,\ s \in S \tag{9}$$

$$x_{ij} \in \{0,1\},\ \forall i \in N, j \in J \tag{10}$$

$$x_j < m,\ \forall j \in J \tag{11}$$

$$2 \leq m \leq 4,\ m \in Z^{+} \tag{12}$$

$$\sum_{j \in J} s_j = n \tag{13}$$

The objective function aims to minimize the resource overtime, resource idle time and patient waiting time. Eq. (1) is added to allocate the appointment time for the i^{th} patient who is set to the j^{th} time slot. Each patient's arrival time is equal to their appointment time, which raises the assumption that the patients are on time. Eq. (2) proves that only one time slot can be allocated to each patient. Eq. (3) ensures that the booked appointments are arranged in the correct order. The beginning time of supplying medical treatment to the first patient is set to zero by Eq. (4). Three components which are patient waiting time, service start time and service end time for each patient are calculated by using Eq. (5) and Eq. (7). Between the $(i-1)^{th}$ and the i^{th} services, Eq. (6) acts as an computation for the resource idle time. Eq. (8) determines the resource overtime meanwhile Eq. (9) executes the non-negativity requirements of the variables. Eq. (10) proves that the patient's assignment is binary.

Eq. (11) will make sure that the number of patients given to the j^{th} time slot is not above the maximum number of patients which can be slotted in a single time. Before an appointment is scheduled, Eq. (12) exhibits the maximum number of patients which can be assigned when m is equals to 2, 3 or 4. As it is shown that the lowest number of patients that can be allocated in one slot is 2, meanwhile the greatest number is 4, the interval of m is set to be [2,4]. The last constraint which is Eq. (13) guarantees that the total number of patients set to every time slots equivalent to the total number of patients allocated for one session which can be shown in mathematical terms, $x_1 + x_2 + ... + x_{12} = n$.

4 DATA

In this research, two different sets of data with different types of distribution are going to be used. The first distribution of the dataset is from the uniform distribution and it is taken from [12]. Meanwhile, the new data is generated by using the exponential distribution.

4.1 Previous Data

From Khairudin et al. (2022), uniform distribution is used to generate the data which focuses on no-show probability, setup time and examination time. However, this is only applicable when the patient show up to their appointment time. For the situation where the patient does not show up, the service duration is set to be zero.

Table 4 : Summary statistics from empirical data

Random variable	Mean	Standard deviation	Distribution
Examination time per test (min)	12.70	8.09	$0.5 + 87 * BETA(2.3, 12.7)$
Setup time for each test (min)	6.40	5.17	$-0.5 + LOGN(7.01, 6.43)$

Therefore, in this research, there are three sets of data which will be used from [12]. It includes the dataset of $D = 1$, $D = 20$ and $D = 100$ where D represents the total number of dataset. These dataset is going to be executed using the heuristic procedure and also the tabu search method.

4.2 New Data

Exponential distribution is used to generate data for no-show probability, setup time and patient examination time. Similar to [12], patient who does not show up to their appointment will have their service duration equal to zero. On the other hand, for those patient who show up, the setup time and examination time will be calculated using the empirical probability distribution.

4.2.1 Notation

Table 5 : Notation for Generate Data

i	Index of patient
d	Index of data set
N	Total number of patients who show up for appointment in each dataset
$setuptime$	Setup time of patient i to receive treatment
$examinationtime_i$	Examination time of patient i to receive treatment
D	Total number of dataset

4.2.2 Generate Data Algorithm

Table 6 : Pseudo code for Generate Data

	Algorithm 1
Step 1	Set a total number of data
Step 2	Generate and assign no-show probability for n patients

Step 3	Generate setup time and examination time for n patients
	(i) if no-show probability of patient $i >$ NSP
	(ii) then assign $setuptime_i$ and $examinationtime_i$ to patient i;
	else
	set $setuptime_i = 0$ and $examinationtime_i = 0$ for patient i
	end if

Step 6	Repeat for next dataset
Step 7	Stop when dataset $> D$

In this study, two distinct datasets, $D = 20$ and $D = 100$ are generated with each dataset comprises information for 14 patients.

5 TABU SEARCH METHOD

For this research, a continuation of research from [11] paper and [12] thesis will be done by proposing another metaheuristic method called the tabu search method to find the near optimal solution for the overbooking appointment. In order to begin the method of TS, the problem will be tackled first using the heuristic procedure by applying the local search algorithm. The notation and algorithm for heuristic procedure is taken from [12] thesis.

TS is one of the metaheuristic problem-solving techniques used to tackle combinatorial optimisation issues. This technique was initially proposed by [15]. A memory can be set by the TS, enabling it to recall existing optimal solutions as well as investigating previous solutions and directing its search. The memory adaptation allow TS to identify better options and discover new potential search areas.

The use of adaptive memory assists TS in learning and it provides with a more flexible and effective search strategies. Furthermore, since this method allows it to avoid the local optimum by selecting a non-improving solution, TS improves performance of heuristic procedure through changing its basic rule.

5.1 Notation

Table 7 : Notation for Tabu Search

i	Index of patient
k	Index of solution
K	Total number of solutions
n	Total number of patients per session
$duration_{ki}$	Duration of patient i to receive treatment for solution k
$endtime_{ki}$	End time of patient i finish received treatment for solution k
$waitingtime_{ki}$	Waiting time of patient i for solution k
$idletime_{ki}$	Idle time of resource between services i and $i+1$ for solution k
WT_k	Total waiting time of solution k
IT_k	Total idle time of solution k
OT_k	Total Overtime of solution k
w_{ot}	Weight measure of overtime
w_{it}	Weight measure of idle time
w_{wt}	Weight measure of waiting time
n_{showup}	Number of patient show up
obj_k	Objective value of solution k
$best_{neighborindex}$	Best neighborhood solution
$best_{obj}$	Best objective value
NUM_{SLOTS}	The arrangement of slots from initial
$tabu_{list}$	Tabu list
$tabu_{size}$	Maximum size of tabu list

Table 8 : Pseudo code for Tabu Search

	Algorithm
Step 1	Set initial value
	$best_{obj} = initial_{obj}$
	$best_solution[NUM_{SLOTS}] = initial_solution[NUM_{SLOTS}]$
Step 2	Define objective value function
	i) Calculate the duration, start time, end time, waiting time and idle time
	a) Set $starttime_{k1} = 0$
	b) Set $duration_{ki} = setuptime_i + examinationtime_i$
	c) Calculate $endtime_{ki}$
	d) If $duration_{ki} == 0$, set $waitingtime_{ki} == 0$;
	Else, Calculate $waitingtime_{ki}$
	End if
	e) If $endtime_{ki} \leq arrival_{k(i+1)}$, set $waitingtime_{k(i+1)} == 0$ and calculate,
	$starttime_k(i+1)$ and $idletime_{ki}$
	Else, if $endtime_{ki} > arrival_{k(i+1)}$, set $idletime_{ki} == 0$ and calculate $starttime_{k(i+1)}$
	f) If $i == n$
	g) If $endtime_{ki} \geq$ end time of a clinic session, set $idletime_{ki} == 0$
	Else, calculate $idletime_{ki}$
	End if
	End if
	ii) Calculate total waiting time, idle time, and overtime.
	$WT[k] = WT[k] + waitingtime[k][i]$
	$IT[k] = IT[k] + idletime[k][i]$
	$OT[k] = endtime[k][n] + waitingtime[k][n] - End$
	iii) Initialize weight measure for each component.
	$w_{ot} = 0.63, w_{it} = 0.30, w_{wt} = 0.07$
	iv) Calculate the objective function.
	$obj_k = w^*_{ot}OT_k + w^*_{it}IT_k + w^*_{wt}(\frac{WT_k}{n_{showup}})$
Step 3	Define neighborhood solutions function
	i) Randomly select the first and second slot to swap
	$index1 = getRandomNumber(1, NUM_{SLOTS} - 4)$
	$index2 = getRandomNumber(1, NUM_{SLOTS} - 4)$
	ii) Use swapping technique to swap
	$temp = neighbor[index1]$
	$neighbor[index1] = neighbor[index2]$
	$neighbor[index2] = temp$
Step 4	Define tabu list function
	i) If $tabuSize < TABU_{SIZE}$
	Add solution to tabu
	Else
	ii) If full, remove the oldest solution and add the new one
Step 5	For every neighborhood solution
	i) Calculate each neighbor[k].
	ii) Compare the objective values between the two neighborhoods and choose the best one.
	$best_{neighborindex} = (average_{obj}[0] < average_{obj}[1])$
Step 6	Update the new best solution and objective value
	if $average_{obj}[best_{neighborindex}] < best_{obj}$
	$best_{obj} = average_{obj}[best_{neighborindex}]$
Step 7	Update tabu list and aspiration criteria
Step 8	End. Stop when termination criteria are met.

The tabu search method starts with the identification of an initial solution and its best objective value, typically derived from a straightforward heuristic approach. The method then advances to the main search phase, where neighboring solutions are generated by applying swapping techniques to this initial solution.

For every neighborhood solution, it will go through the process of calculating the objective value which the function has been defined in the beginning of the procedure. Then, store the objective value for the current neighbor which is needed to be used in the comparison process. Proceed to the comparison process where the objective value between the neighborhoods will be compared and the best solution will be chosen. The best solution will be declared as the current objective value of the best neighborhoods.

In the following step, the tabu list and aspiration criteria will be updated. To update the tabu list, an observation of how long an entry has been in the tabu list is made. If the entry in the tabu list exceeds the tabu list size, which has been set at the beginning, then the entry will be removed. Then, the aspiration criteria are also updated by comparing the current objective value of the best neighborhoods with the best solution. The best solution will be updated if objective value of a current solution of the best neighborhoods is greater than the objective value of the best solution. This procedure will be repeated until the stop criteria are met.

5.2 Computational Result

This comparison is divided into two sections. The first section focuses on comparing the Tabu Search (TS) method with previous methods, including the heuristic procedure, Simulated Annealing (SA), and Genetic Algorithm (GA), using a dataset where $D = 1$. The second section examines larger datasets, specifically $D = 20$ and $D = 100$, but the comparison is limited to the heuristic procedure and the TS method. Each dataset comprises information for 14 patients. For instance, $D = 20$ corresponds to 20 datasets, totaling 280 patients' data. Similarly, $D = 100$ represents data for 1400 patients.

5.2.1 Comparison between previous methods and TS method

Table 9 : Comparison between four methods for $D = 1$

	Heuristic	Simulated Annealing	Genetic Algorithm	Tabu Search
Total objective value	21.9793	21.7460	21.3960	21.8626
Total waiting time	352.920	312.920	252.92	332.92
Total idle time	0.00	0.00	0.00	0.00
Total overtime	31.62	31.62	31.62	31.62
Exact time	0.473 s	0.678 s		0.218 s

Table 10 : Solutions of previous methods and modify TS for dataset $D = 1$

Heuristic												
Slot	1	2	3	4	5	6	7	8	9	10	11	12
No. of Patient	3	1	1	1	1	1	2	1	1	2	0	0
GA												
Slot	1	2	3	4	5	6	7	8	9	10	11	12
No. of Patient	3	1	1	1	0	1	1	1	2	3	0	0
SA												
Slot	1	2	3	4	5	6	7	8	9	10	11	12
No. of Patient	3	1	1	1	1	0	2	1	2	2	0	0
TS												
Slot	1	2	3	4	5	6	7	8	9	10	11	12
No. of Patient	3	1	1	1	1	1	1	1	2	2	0	0

Based on Table 9, GA method shown to give the best performances as it gives the smallest objective value compared to heuristic procedure, SA method and TS method. GA method also has the lowest waiting time compare to others. However, it can be seen that the objective values between the four methods are close to each other, and the difference is not that large. In terms of idle time and overtime, all four methods give the same output meanwhile for the exact time, it can be observed that TS method execute the program faster.

As shown in Table 10, each method identified a different optimal solution for appointment scheduling. However, it is evident that in all these solutions, three patients were assigned to the first slot, leaving the final two slots empty. For the remaining slots, either no patients or only one patient was scheduled, following an overbooking pattern that minimized resource overtime, resource idle time, and patient waiting time.

5.2.2 Comparison between heuristic method and Tabu Search method

Table 11 : Comparison between heuristic procedure and TS method

	$D = 20$		$D = 100$	
	Heuristic	TS	Heuristic	TS
Mean objective value	30.9561	30.9561	27.7188	27.7188
Mean waiting time	42.7981	42.7981	48.742	48.742
Mean idle time	1.1905	1.1905	0.5858	0.5858
Mean overtime	36.8418	36.8418	34.9741	34.9741
CPU time	0.368 s	0.299 s	0.530 s	0.409 s

Table 12 : Solutions of heuristics procedure and modify TS for dataset $D = 20$ and $D = 100$

$D = 20$													
Heuristic	Slot	1	2	3	4	5	6	7	8	9	10	11	12
	No. of Patient	3	1	1	2	1	1	1	1	1	2	0	0
TS	Slot	1	2	3	4	5	6	7	8	9	10	11	12
	No. of Patient	3	1	1	2	1	1	1	1	1	2	0	0
$D = 100$													
Heuristic	Slot	1	2	3	4	5	6	7	8	9	10	11	12
	No. of Patient	4	1	1	1	1	1	1	1	1	2	0	0
TS	Slot	1	2	3	4	5	6	7	8	9	10	11	12
	No. of Patient	4	1	1	1	1	1	1	1	1	2	0	0

5.3 Conclusion

As the dataset size increases, the TS method exhibits limited improvement in both reducing the objective value and enhancing the three performance measures. Therefore, this study endeavors to identify the underlying causes to achieve more favorable outcomes. When employing the TS method with an initial solution consisting of 4 patients in the first slot, an intriguing observation emerges which the neighborhood solutions generated in the TS method closely resemble the outcomes for m = 4 derived from the local search algorithm.

Thus, no matter how many iterations are being run when executing the TS method, the output will always be similar to the results from heuristic procedure. Consequently, there arises a compelling need to refine the TS algorithm, seeking modifications that enable the exploration of a broader spectrum of neighbourhood solutions.

6 MODIFICATION OF TABU SEARCH

To enhance the search for better neighborhood solutions in the TS algorithm by modifying a specific segment identified in Table 13. This segment currently limits exploration of superior solutions within the neighborhood, even after over 1,000 iterations. To diversify the solutions obtained, adjustments are necessary in how neighborhood solutions are generated within this designated section of Table 13.

Table 13 : Pseudo code for General Tabu search

Step 3	Define neighborhood solutions function
	i) Randomly select the first and second slot to swap
	$index1 = getRandomNumber(1, NUM_{SLOTS} - 4)$
	$index2 = getRandomNumber(1, NUM_{SLOTS} - 4)$
	ii) Use swapping technique to swap
	$temp = neighbor[index1]$
	$neighbor[index1] = neighbor[index2]$
	$neighbor[index2] = temp$

Figure 1 proves that the TS method exclusively employs slot swapping from the initial solutions to generate neighborhood solutions, with no alternative methods permitted. This highlights the need to improve the general tabu search approach so that it can explore different solutions and produce better outcomes.

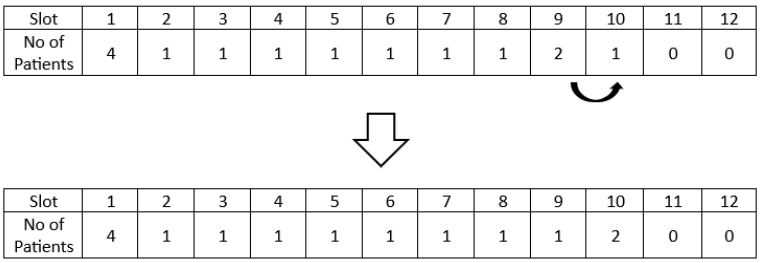

Figure 1 : Swapping technique.

6.1 Modify Tabu Search Pseudo Code

Therefore, for the modification that will be made, TS method will allow the generated neighborhood solution to randomly assign patients to various slots, including cases where certain slots may have zero patients as illustrated in Figure 2.

This technique is deemed acceptable as it adheres to the specified constraints, whereby the total number of patients must always equate to 14. The initial slot follows a multiple assignment approach, where the value of m can be either 2, 3, or 4, while the final two slots maintain a patient count of zero.

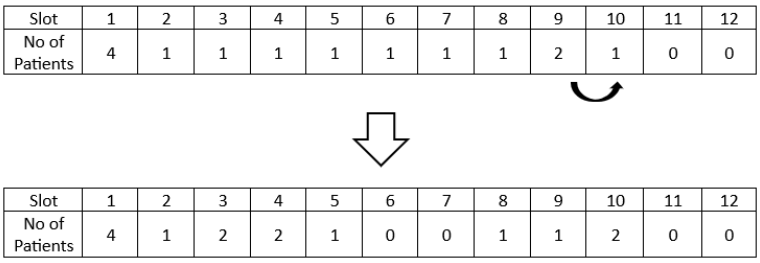

Figure 2 : Swapping and randomly assign technique.

The table below shown the section where the modification has been made in the general tabu search algorithm.

Table 14 : Pseudo code for Modify Tabu Search

Step 3	i) Randomly choose the first slot to be either 2, 3, or 4 $randomFirstSlot = getRandomNumber(2, 4)$ ii) Set the first slot based on the random choice $neighbor[0] = randomFirstSlot$ iii) Randomly select the second slot to swap $index1 = getRandomNumber(1, NUM_{SLOTS} - 4)$ iv) Randomly select the third slot to swap $index2 = getRandomNumber(1, NUM_{SLOTS} - 4)$ v) Calculate the maximum number of patients that can be swapped $maxPatients = min(4 - neighbor[index1], neighbor[index2])$ vii) Randomly determine the number of patients to swap $patientsToSwap = getRandomNumber(0, maxPatients)$ viii) Update the number of patients in the slots $neighbor[index1] += patientsToSwap$ $neighbor[index2] -= patientsToSwap$ ix) Ensure the total number of patients is 14 $totalPatients = calculateTotalPatients(neighbor)$ x) Adjust if needed to ensure a total of 14 patients $adjustment = 14 - totalPatients$ if $adjustment != 0$: a) Randomly select a slot to adjust $adjustIndex = getRandomNumber(1, NUM_{SLOTS} - 2)$ b) Update the number of patients in the selected slot $neighbor[adjustIndex] += adjustment$ xi) Ensure the last two slots remain 0 $neighbor[NUM_{SLOTS} - 2] = 0$ $neighbor[NUM_{SLOTS} - 1] = 0$

7 RESULTS AND DISCUSSION

The results obtained from heuristics procedure and modify tabu search for dataset $D = 20$ and $D = 100$ are shown in Table 15, meanwhile the best solution for both dataset is shown in Table 16. The minimum value which corresponds to the best performance of each measure among heuristics procedure and modify tabu search method is highlighted.

Table 15 : Comparison between heuristic procedure and modify TS method

	$D = 20$		$D = 100$	
	Heuristic	M TS	Heuristic	M TS
Mean objective value	36.2406	36.1352	40.9509	40.9312
Mean waiting time	51.3799	50.7960	61.5669	61.4780
Mean idle time	0.5989	0.5989	0.1973	0.1746
Mean overtime	48.1685	48.1685	57.1843	56.8668
CPU time	0.329 s	0.273 s	0.499 s	0.407 s

Table 16 : Solutions of heuristics procedure and modify TS for dataset $D = 20$ and $D = 100$

$D = 20$													
Heuristic	Slot	1	2	3	4	5	6	7	8	9	10	11	12
	No. of Patient	3	1	1	1	1	1	2	1	1	2	0	0
Modify TS	Slot	1	2	3	4	5	6	7	8	9	10	11	12
	No. of Patient	3	1	1	1	1	1	2	1	0	3	0	0
$D = 100$													
Heuristic	Slot	1	2	3	4	5	6	7	8	9	10	11	12
	No. of Patient	4	1	1	1	1	1	1	1	1	2	0	0
Modify TS	Slot	1	2	3	4	5	6	7	8	9	10	11	12
	No. of Patient	4	1	1	1	2	1	1	0	1	2	0	0

Based on the dataset $D = 20$ in Table 15, from the terms of performances measures for mean idle time and mean overtime, it can be observed that both methods produce the same values. However, in order to decide which method performs better, the method which execute the lowest objective value will be chosen. Thus, in this case, TS method gives a lower mean objective value and also a lower mean waiting time. Although the value of objective value for both methods are closer to each other, but TS method shows a slightly preferable execution.

On the other dataset $D = 100$, both methods has a slight difference in their mean idle time and mean overtime as stated in Table 15. However, TS method gives a better results in terms of mean objective value and also the mean waiting time as it executes a lower values for both measures than the heuristic procedure. Thus, this shows that even for large dataset, TS method still remain as the method that produce a preferable results than heuristic procedure. Thus, this shows that even for large dataset, TS method still remain as the method that produce a preferable results than heuristic procedure.

Pertaining to Table 16, the results indicated that, for smaller datasets, the optimal condition for the maximum number of patients in the same time slot is three. However, with larger datasets, assigning a maximum of four patients to the same time slot proved to be more effective in minimizing the medical center's cost. A consistent trend emerged, with three or four patients allocated to the first slot, leaving the final two slots empty. For other slots, one or occasionally two patients were assigned, following an overbooking scheduling pattern that minimized resource overtime, resource idle time, and patient waiting time.

Therefore, introducing minor adjustments to the tabu search algorithm can result in improved outcomes compared to the heuristic procedure. Consequently, the tabu search general algorithm exhibits high flexibility, allowing for modifications at various stages, including the generated neighborhood part or the tabu list function.

8 CONCLUSION

This research aims to address the issue of overbooking appointment scheduling in the presence of no-show patients. The primary objective is to minimize resource overtime, resource idle time, and patient waiting time. Data from [12] are used to compare tabu search method and assess which approach yields a smaller objective value. However, when no improvement was observed with $D = 100$, a modification of the tabu search method was implemented using new data generated through an exponential distribution. Consequently, the results of the test are as follows:

1. In terms of CPU time, the tabu search procedure exhibited a shorter computation time compared to the three others methods which are the SA, GA and also heuristic procedure.

2. Introducing minor adjustments to the tabu search algorithm can result in improved outcomes compared to the heuristic procedure.
3. The tabu search method is able to solve the overbooking appointment scheduling problem by minimizing the resource overtime, resource idle time and patient waiting time.

ACKNOWLEDGEMENT

All related information has been provided by the Department of Mathematics and Statistics in UPM. We would like to thank Nor Aliza Ab Rahmin and Athirah Nawawi from the Mathematics and Statistics Department in UPM for the fruitful discussion and contribution to this research.

REFERENCES

[1] M. Sohrabi, M. Tumin, and A. F. Osman, "Issues and challenges of public health accessibility among urban poor people: A case study of malaysia, iran and india," *Malaysian Journal of Medical Research (MJMR)*, vol. 2, no. 4, pp. 22–31, 2018.

[2] D. Gupta and B. Denton, "Appointment scheduling in health care: Challenges and opportunities," *IIE transactions*, vol. 40, no. 9, pp. 800–819, 2008.

[3] J. Dreiher, M. Froimovici, Y. Bibi, D. A. Vardy, A. Cicurel, and A. D. Cohen, "Nonattendance in obstetrics and gynecology patients," *Gynecologic and obstetric investigation*, vol. 66, no. 1, pp. 40–43, 2008.

[4] L. V. Green and S. Savin, "Reducing delays for medical appointments: A queueing approach," *Operations Research*, vol. 56, no. 6, pp. 1526–1538, 2008.

[5] Y. Chen, Y.-H. Kuo, P. Fan, and H. Balasubramanian, "Appointment overbooking with different time slot structures," *Computers & Industrial Engineering*, vol. 124, pp. 237–248, 2018.

[6] Z. Fan, X. Xie, R. A. Sanchez, and X. Zhong, "Overbooking for specialty clinics with patient no-shows: A queueing approach," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 396–401.

[7] X. Pan, N. Geng, and X. Xie, "Appointment scheduling and real-time sequencing strategies for patient unpunctuality," *European Journal of Operational Research*, vol. 295, no. 1, pp. 246–260, 2021.

[8] S. Srinivas and H. Salah, "Consultation length and no-show prediction for improving appointment scheduling efficiency at a cardiology clinic: a data analytics approach," *International Journal of Medical Informatics*, vol. 145, p. 104290, 2021.

[9] Y.-H. Kuo, H. Balasubramanian, and Y. Chen, "Medical appointment overbooking and optimal scheduling: tradeoffs between schedule efficiency and accessibility to service," *Flexible Services and Manufacturing Journal*, vol. 32, pp. 72–101, 2020.

[10] C. Zacharias and M. Pinedo, "Appointment scheduling with no-shows and overbooking," *Production and Operations Management*, vol. 23, no. 5, pp. 788–801, 2014.

[11] W. Y. Chua, N. A. Rahmin, and A. Nawawi, "Solving overbooking appointment scheduling problem under patient no-show condition using heuristics procedure and genetic algorithm," *Mathematical Modeling and Computing*, vol. 9, no. 1, pp. 65–73, 2022.

[12] N. I. H. Khairudin, N. A. Rahmin, and A. Nawawi, "*Simulated annealing approach for an overbooking appointment scheduling problem*," Universiti Putra Malaysia, 2022.

[13] Z. Sun and Q. Wu, "Two-phase tabu search algorithm for solving chinese high school timetabling problems under the new college entrance examination reform," *Data Science and Management*, vol. 6, no. 1, pp. 55–63, 2023.

[14] Z. H. Ahmed and M. Yousefikhoshbakht, "An improved tabu search algorithm for solving heterogeneous fixed fleet open vehicle routing problem with time windows," *Alexandria Engineering Journal*, vol. 64, pp. 349–363, 2023.

[15] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & operations research*, vol. 13, no. 5, pp. 533–549, 1986.