MULTI-START LOCAL SEARCH FOR ONLINE SCHEDULING IN PARALLEL OPERATING THEATRE

NUR NEESHA ALIMIN, NOR ALIZA ABD RAHMIN[1], GAFURJAN IBRAGIMOV,
AND NAZIHAH MOHAMED ALI

ABSTRACT. The present work is considered online operating theatre scheduling for regular patients and also it is involving emergency patients on a day. The emergency patients urge to schedule the earliest as their level of priority is higher than the regular patients and it is the importance of online scheduling problem. The local search (LS) method was tested for scheduling the parallel operating theatre with an objective to minimize the delay and overtime cost of operating theatre. A modification of the local search algorithm was performed to improve the solutions and it is called multi-start local search (MSLS). This research found that, MSLS method performed better than LS method since it has reduced the total cost of the operating theatre significantly.

1. INTRODUCTION

Operating theatres (OT) scheduling is an essential operational problem in the health care industry and the OT facilities had a large consumption in a hospital. Most of the health care organizations are having difficulties in providing the best services with lower costs. Researchers that study on combinatorial optimisation problem on scheduling are able to improve the efficiency of the OT management in the health care area. There are numerous research has been done in scheduling

OT such as [2], [4] and others. In 2018, Mateus et al. [8] explored local search (LS) for elective surgeries scheduling problem and they compared the results obtained with mixed integer linear programming model. They aimed to minimize the days in the waiting list and minimize the penalty for unscheduled surgery. The method showed a good result with a short amount of computational time. However, emergency patients are not being considered in their research for the scheduling problem.

Marti et al. [6] studied on the multi-start methods that applied to the optimization problem. The methods consist of two phase where the solution is generated first and then it proceeds with improving the solution. This paper is motivated by the methods where we could improve the scheduling problem in order to achieve a satisfactory result for the management of operating theatre and also patients. There is limited literature on the optimization problems that are considering multi-start local search and the works can be found in [1] and [5].

In the emergency surgery cases, not many researchers are considering on scheduling the emergency patients as scheduling for the emergency patients is regarded as a challenging task due to their uncertain times of arrival. Guerriero and Guido [3] stated that most of the literature on planning and scheduling of OT are considering regular patients only. A few works regarding online scheduling are found in [7], [9] and [10]. Bouguerra et al. [7] introduced online assignment strategies for emergency patients scheduling. They considered three patient classes which are emergent, critical and work in cases. The strategies and decision support tool are built for each class to handle the uncertain flow of patients. In 2018, Rahmin [9] studied on online scheduling problem involving emergency patients with uncertain arrival of times. A zero-one programming model is developed by the researcher to schedule multiple OT in a day. Other than that, the model is tested with generated data and an algorithm that compatible with the model is developed to solve the scheduling problem. The results showed an improvement in minimizing the cost and the schedule are generated in a short computational time even with a large data set.

Jung et al. [10] construct a model for designating the OT capacity for elective and also emergency patients that arrived randomly to prevent an excessive delay occurs. They also developed frameworks for weekly aggregate schedules, daily schedules and rescheduling when emergency patients arrived with an objective to minimize the total costs in terms of expected operating time, idle time and

overtime. In addition, they presented heuristic-based using the Longest Processing Time (LPT) - Shortest Processing Time (SPT) rule for the daily schedules and they scheduled the patients for a maximum of three OT. However, in terms of the daily schedules, the rules to schedule the patients is differ from our study where we scheduled the highest weighted (priority) patients the earliest in the OT and a maximum of ten OT is considered. In addition, we have a different objective function where the cost minimization involved overtime and also delay costs.

This paper is an extension on the problem studied by Rahmin [9] where emergency patients is considered together with regular patients to create an efficient schedule for solving the online scheduling problem in parallel OT. It is an online scheduling problem for the reason that the emergency patients will arrived suddenly on the day and they need to be schedule as soon as possible into the existed schedule. Their priority level is higher than regular patients and because of that we have to schedule them the earliest when the OT is available. This paper presents the modification of LS method which is multi-start local search (MSLS) method to apply in the OT scheduling with an objective to minimize the cost of the delay patients and also the cost of overtime usage of OT. We compare the LS method with MSLS method regarding the costs to show the best schedules obtained.

The paper is organized as follows. Section 2 introduces the model, method and algorithms used in this research. In Section 3, we present the results of scheduling using generated data, followed by discussion to compare the best method for the problem. Finally, the conclusion and future works are given in Section 4.

2. Materials and Methods

In this section, we presented a zero-one programming (ZOP) model and a new modified algorithm based on the local search. A new algorithm of MSLS is developed for solving the online scheduling problem in parallel OT. In this paper, scheduling operating theatre is a computationally difficult problem especially when it involved online scheduling which cannot be easily solved. Therefore, we applied a heuristic method to achieve a near optimal solution for the problem. Before that, a basic heuristic procedure is also applied for obtaining the best initial schedule.

2.1. **Zero-One Programming Model.** In this section, a zero-one programming model is presented. The model is used for the online scheduling problem. All the

notations, decision and binary variables used in the model are introduced in Table 1.

TABLE 1. Notations used in the ZOP model

i	Operating Theatre (OT)
j	Patient
m	Total number of OT
n	Total number of patients
t_{ij}	Planning duration time for patient j in OT i
T	Time horizon
T_1	Upper bound of theatre time for OT i after time horizon T
T_2	Upper bound of theatre time for OT i after time horizon T_1
ω_j	Weight of penalty patient j if the patient untreated on the day
β_{i1}	Penalty of completion time for OT i after time horizon in time section 1 $(T, T_1]$
β_{i2}	Penalty of completion time for OT i after time horizon in time section 2 $(T_1, T_2]$

The following is the zero-one programming model that used in this paper.

$$(2.1) \quad min(P) = \sum_{j=1}^{n} \omega_j v_j + \sum_{i=1}^{m} (\beta_{i1} z_{i1} + \beta_{i2} z_{i2})$$

$$(2.2) \quad s.t. \quad \sum_{i=1}^{m} x_{ij} + v_j = 1, \quad \forall j = 1, 2, \ldots, n$$

$$(2.3) \quad \sum_{j=1}^{n} t_{ij} x_{ij} \leq T + (T_1 - T) z_{i1} + (T_2 - T) z_{i2}, \quad \forall i = 1, 2, \ldots, m$$

$$(2.4) \quad z_{i1} + z_{i2} \leq 1, \quad \forall i = 1, 2, \ldots, m$$

$$(2.5) \quad x_{ij} \in \{0, 1\}, \quad \forall i = 1, 2, \ldots, m, \quad \forall j = 1, 2, \ldots, n$$

$$(2.6) \quad z_{i1}, z_{i2} \in \{0, 1\}, \quad \forall i = 1, 2, \ldots, m$$

$$(2.7) \quad v_j \in \{0, 1\}, \quad \forall j = 1, 2, \ldots, n$$

The objective function (2.1) is to minimize the cost if the patients are untreated on the day, $\sum_{j=1}^{n} \omega_j v_j$ and the cost of overtime usage of the OT, $\sum_{i=1}^{m} (\beta_{i1} z_{i1} + \beta_{i2} z_{i2})$. Constraint (2.2) is to ensure the patient j will be either assigned to one

of the OT or delayed into the next. Constraint (2.3) is to ensure the sum of the planning durations in each OT is less than or equal to the time horizon T and also the overtime period if there is any. Constraint (2.4) is to ensure that only one overtime band will be counted since the cost for overtime band 1 is included in overtime band 2. The decision variable (2.5) and binary variables (2.6) and (2.7) will always take value of either 1 or 0.

This model is applied in the process of online scheduling when the emergency patients arrived on the day the operations going through for the regular patients. The two elements in the objective function (2.1) can only be implemented for the online scheduling because it involves the decision to delay the patient or extend the time duration of OT that occurs on that day. It is essential to make the best decision immediately as soon as the emergency patients arrived.

2.2. **Local Search Method.** LS method is used in solving the online scheduling problem since it involved in searching throughout the neighbourhood for obtaining the best solution. The algorithm will move from one solution in the neighbourhood to the next solution until an optimal solution is found. After that, the previous solution will be replaced with a better solution that has been found, then the neighbourhood executed again for searching new solution and none will be left out in the searching procedure. The search will stop when there is no improved solution found in the neighbourhood. The algorithm of the LS is modified to suit with our problem by adding a condition based on the ZOP model in order to further improve and obtain a better solution. A more detailed searching throughout the neighborhood also will be conducted by modifying the algorithm of LS. The details of the algorithm are further discussed in the section of algorithms.

2.3. **Algorithms.** This section presents two algorithms which are initial schedule algorithm and MSLS algorithm. The notations used in the algorithms are introduced in Table 2.

TABLE 2. Notations in the algorithms

i	Operating Theatre (OT)
i_0	OT i_0
j	Patient
j_e	Emergency patient
$j(n)$	Last patient

m	Total number of OT
n	Total number of patients
z_{i1}	Overtime band 1 in OT i
z_{i2}	Overtime band 2 in OT i
t_{ij}^p	Planning duration time for patient j in OT i
t_{ij}	Actual duration time for patient j in OT i
$\pi_i(j)$	Sequence of patient j for OT i
ω_j	Weight of penalty patient j
T	Time horizon in a day
T^e	Arrival time of emergency patient
T_{ij}^c	Completion time of patient j in OT i
L	List of patients
$pos(j)$	Position of patient
$L(n)$	Total number of patients in the list
x_{ij}	1 if patient j is assigned to OT i; 0 otherwise
v_{j_e}	1 if emergency patient j_e is untreated on the day; 0 otherwise
$v_j(i_0)$	1 if patient j is untreated in OT i_0; 0 otherwise

2.3.1. *Initial Schedule Algorithm.* A basic heuristic procedure is used to obtain the initial schedule. The initial schedule algorithm is introduced in Table 3. In step 1, we input the data for patients where the number of patients in each OT will be decided according to the time horizon in a day. Then, the OT will set to one. In step 2, we assign the patients into OT based on planning duration of the patient. The step will repeat until the total number of OT is reached. Step 3 is where we swap the patient in one of the OT with the patient in another OT if it gives a better result than before. We sort the patients in decreasing order based on the weight of patients in the final step. The patients with a higher number of weights show the priority of the patient will be scheduled earlier. Therefore, the initial schedule is obtained.

TABLE 3. Initial Schedule

1: Input t_{ij}^p and ω_j for $j = 1, \ldots, n$ and $i = 1, \ldots, m$ according to T. Set $i = 1$.
2: Apply the following for $j = 1, \ldots, n$.

If $t_{ij}^{p_a(i)} < \ldots < t_{ij}^{p_a(m)}$
    Assign patient j to OT i
else
    Assign patient j to OT $i + 1$
Set $i = i + 1$. Repeat Step 2 until $i = m - 1$.
3: Apply the following for $i_0 = 1, \ldots, m$.
    If $v_j(i_0) = 0$ and $v_{j+1}(i+1) = 1$
        Set $x_{ij} = 1$ and $x_{ij+1} = 0$
        Calculate objective function $P = \pi_{i_0}(j) \cdot v_j(i_0) + \pi_i(j+1) \cdot v_{j+1}(i)$
        Swap patients $v_j(i_0) = 1$ and $v_{j+1}(i+1) = 0$
        Calculate new objective function P'
        If $P' < P$
            $x'_{ij} = x_{ij}$
            $x'_{ij+1} = x_{ij+1}$
        else
            $x'_{ij} = x_{ij} - 1$
            $x'_{ij+1} = x_{ij+1} + 1$
    else if $v_j(i_0) = 0, v_{j+1}(i+1) = 0$ or $v_j(i_0) = 1, v_{j+1}(i+1) = 1$
        continue
    Repeat Step 3 for $j = 1, \ldots, n$ and $i = 1, \ldots, m$.
4: Apply the following for $i = 1, \ldots, m$.
    Form a sequence $\pi_i = (\pi_i(1), \pi_i(2), \ldots, \pi_i(n))$ for the patients such that
    $\omega_{\pi_i(1)} \geq \omega_{\pi_i(2)} \geq \ldots \geq \omega_{\pi_i(n)}$.

2.3.2. *Multi-Start Local Search Algorithm.* After obtaining the initial schedule, MSLS will be applied when emergency patient arrive in the system. The MSLS algorithm is introduced in Table 4. In step 1, we set the completion time of patient as zero for the starting. In step 2, we input the data for emergency patients into the list of patients. Then, we check for the condition of objective function and also we count the total planning duration of regular patients with emergency patients to check for the cost if there will be included any overtime band. After that, the position of emergency patient is set to last patient in the list for each OT and overtime cost is calculated.

In step 3, we assign the emergency patient into a position in the OT that depends

on the time arrival of the emergency patient. The step will repeat until last patient reached. In step 4, we calculate the delay cost for all patients in each OT. The other patients already scheduled on the day may be delay to another day or the usage of the OT on the day may be extend depends on the minimum costs. In step 5, we will repeat step 2 for next emergency patient until there is no emergency patient available. Then, we will compute the total cost for each OT.

TABLE 4. Multi-Start Local Search

1: Set $j = 0$ and $T_{ij}^c = 0$ for all $i = 1, \ldots, m$.

2: Input $t_{ij_e}^p, \omega_{j_e}$ and T^e for j_e into L. Set $\upsilon_{j_e} = 1, z_{i1} = 0$ and $z_{i2} = 0$

     If $\omega_{j_e} + \beta_{i1} z_{i1} + \beta_{i2} z_{i2} > 0$ and $t_{ij}^p + t_{ij_e}^p \leq T$

        $z_{i1}' = 0$ and $z_{i2}' = 0$

     else if $\omega_{j_e} + \beta_{i1} z_{i1} + \beta_{i2} z_{i2} > 0$ and $t_{ij}^p + t_{ij_e}^p \leq T_1$

        $z_{i1}' = 1$ and $z_{i2}' = 0$

     else if $\omega_{j_e} + \beta_{i1} z_{i1} + \beta_{i2} z_{i2} > 0$ and $t_{ij}^p + t_{ij_e}^p \leq T_2$

        $z_{i1}' = 0$ and $z_{i2}' = 1$

     Set new value $\upsilon_{j_e}' = 0$ and new value $x_{ij_e}' = 1$.

     Set $pos(j_e) = L(n)$ for all $i = 1, \ldots, m$.

     Calculate $OvertimeCost(j + j_e)$ for $t_{ij}^p + t_{ij_e}^p$

3: Apply the following for $i = 1, \ldots, m$.

     Set $T_{ij}^c = T_{ij}^c + t_{ij+1}^p$.

     If $T^e \leq T_{ij}^c$

        Move position j_e such that $pos(j_e) = pos(j + 1)$

     Set $j = j + 1$. Repeat Step 3 until $j = n$.

4: Calculate $DelayCost(j + j_e - j(n))$ for $t_{ij}^p + t_{ij_e}^p - t_{ij(n)}^p$

     If $OvertimeCost(j + j_e) < DelayCost(j + j_e - j(n))$

        Accept j_e and continue the surgery.

     else

        Accept j_e and delay $j(n)$ in i.

5: If no j_e, stop.

     Compute $TotalCost$ for $i = 1, \ldots, m$.

     else go to Step 2.

---

3. Results and Discussion

We used the same generated data for LS and MSLS method in solving the online scheduling problem. Three types of range are generated for the planning duration time of patients. The data are generated using uniform distribution, $U[30, 120]$ for planning duration time of patients, t_{ij}^p. Then, we generate a random number $r = 1, 2, 3$ for a set of data in a day. For the first type of range, we set $t_{ij}^p = t^p$ if $r = 1$. The second type of range is set as $t_{ij}^p = t^p + t^e$ where t^e is generated from $U[10, 40]$ if $r = 2$. Lastly, we set $t_{ij}^p = 1000$ if $r = 3$.

The three types of range are generated for the planning duration time to test the compatibility of the model with the algorithm. For the actual duration time for patients, it is generated from the uniform distribution, $U[t_{ij}^p - 20, t_{ij}^p + 20]$ which means the actual duration may be finished twenty minutes early or twenty minutes late. Other than that, the data of emergency patients are generated with the same planning and actual duration time distribution as regular patients. The arrival times for emergency patients are generated from $U[1, 480]$. For the weight of patients, ω_j is generated from the uniform distribution defined on $U[20, 80]$.

We have generated fifty different data for several numbers of OT and the data have been tested with LS and MSLS methods. Both of the results are compared and showed in the average number. Figure 1-4 presented the average number of the results in four criteria based on the number of OT. In Figure 1, it is clearly showed the average number of patients treated is increased when the number of OT increased for both methods. There is a small difference (zero percent to two percent) between both methods where the MSLS has more patients treated than LS.

Figure 2 below shows the average number of patient untreated based on number of OT. The total actual duration of the OT in a day is involved in the scheduling process, thus the number of OT does not show a constant increment or decrement. The patients untreated in a day will be lesser or none if the OT does not have overtime and vice versa. By comparing MSLS and LS methods, MSLS achieved an improvement result where lesser patients untreated. There is a large gap between both methods where the higher number of OT has reduced the number of patients untreated except for the result of four OT (increased by 18 percent). The patients untreated are higher for four OT as the schedule focused on achieving a better result in terms of the total cost of the OT. The graph (Figure 2) also shows that
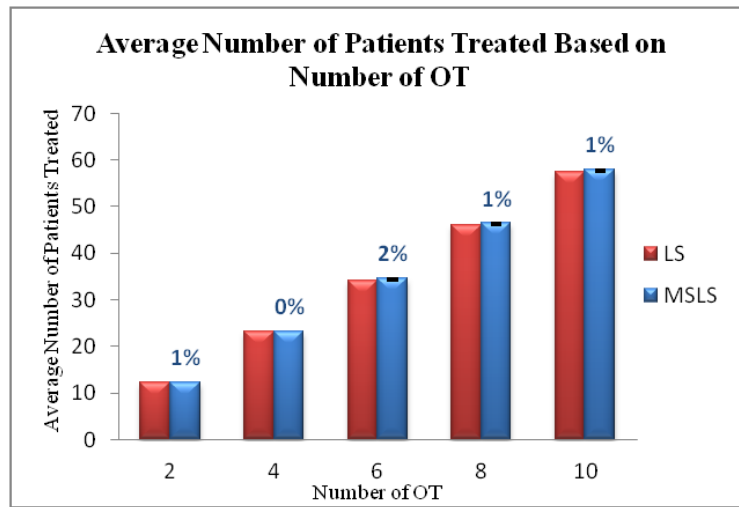
FIGURE 1. The number of OT determined the average number of patients
treated. The number of percentage shows the gap difference between LS
and MSLS method.

the patients untreated is reduced by 100 percent in ten OT where there is zero
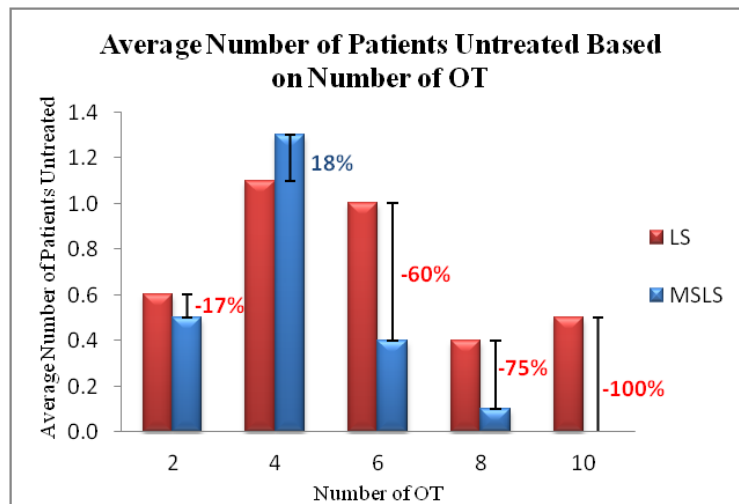patients untreated in the OT.

FIGURE 2. The graph shows the average number of patients untreated for
each number of OT.

In Figure 3, the graph indicates the average total cost based on number of OT
and it shows the graph of the total cost varied because it depends on the total

actual duration of the OT. MSLS method has greatly minimized the total cost than LS method especially for a higher number of OT. The highest reduction of the total cost is by 72 percent for eight OT and the lowest reduction of the total cost is by six percent for two OT as shown in Figure 3.
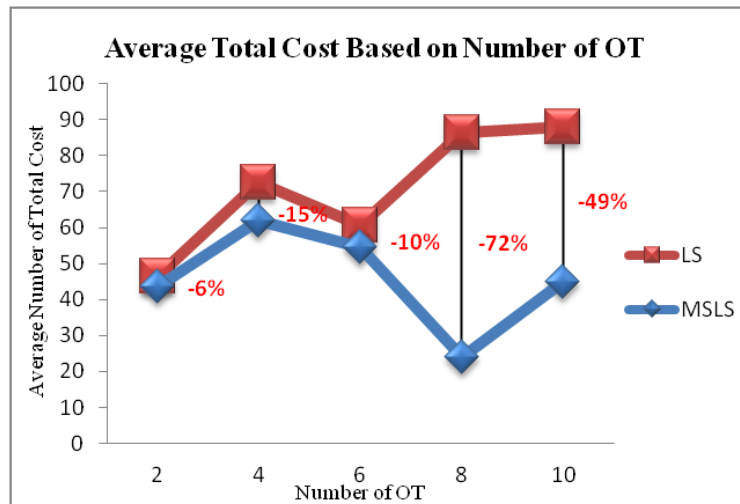

FIGURE 3. The average total cost is illustrated for each number of OT.

Figure 4 illustrates the average CPU time to generate the schedule for MSLS and LS methods. The average CPU time is in the range of zero to 3.50 seconds even though there are slightly differences between both methods. The figure shows an improvement for the CPU time of scheduling for the MSLS method as the number of OT increase.

4. CONCLUSION

This research aimed to minimize the cost of scheduling regular and emergency patients in the OT. We have tested fifty different data and the results are conveyed in the average value. MSLS method used the same generated data as LS method for the comparison purpose. Based on the results obtained in Figure 3, it can be concluded that the MSLS method has significantly reduced the total cost by 30.4 percent in average over all the number of OT. Other than that, MSLS method shows that most of the OT obtained a lower number of patients untreated and there is none of the patient untreated for ten OT as seen in Figure 2. Overall, the results of using MSLS method are efficient and able to perform in a short amount
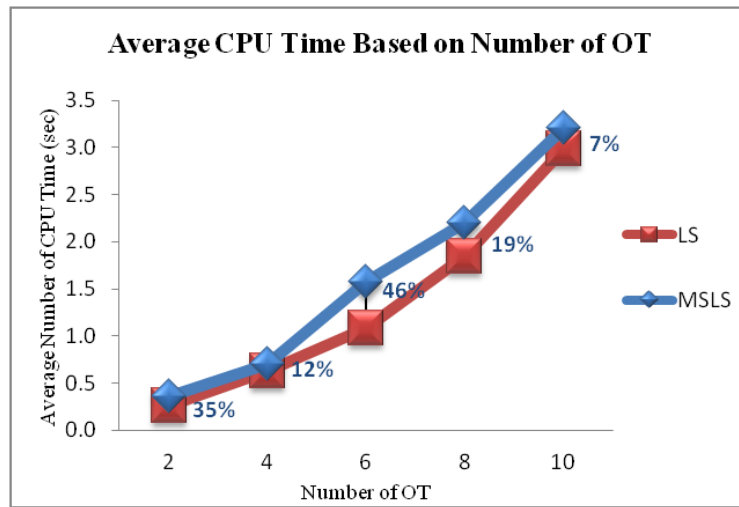
FIGURE 4. The number of OT effects the average CPU time.

of time even for the schedule of a high number of OT. Thus, it is a good contribution for hospital that has more OT for the scheduling. In the future, meta-heuristic method such as genetic algorithm can be considered for solving the online scheduling problem.

ACKNOWLEDGMENT

REFERENCES

[1] G. BRONMO, M. CHRISTIANSEN, K. FAGERHOLT, B. NYGREEN: *A multi-start local search heuristic for ship scheduling–a computational study*, Computers & Operations Research, **34** (2007), 900-917.

[2] H. FEI, N. MESKENS, C. CHU: *A planning and scheduling problem for an operating theatre using an open scheduling strategy*, Computers and Industrial Engineering, **58**(2) (2010), 221–230.

[3] F. GUERRIERO, R. GUIDO: *Operational research in the management of the operating theatre: A survey*, Health Care Management Science, **14**(1) (2011), 89–114.

[4] Y. LIU, C. CHU, K. WANG: *A new heuristic algorithm for the operating room scheduling problem*, Computers & Industrial Engineering, **61** (2011), 865–871.

[5] A. GYORGY, L. KOCSIS: *Efficient Multi-Start Strategies for Local Search Algorithms*, Journal of Artificial Intelligence Research, **41** (2011), 407–444.

[6] R. MARTI, M.G.C. RESENDE, C. C. RIBEIRO: *Multi-start methods for combinatorial optimization*, European Journal of Operational Research, **226** (2013), 1–8.

[7] A. BOUGUERRA, C. SAUVEY, N. SAUER: *Online assignment strategies for emergent, urgent and work-in-cases surgeries in an operating theatre*, In Proceedings of the International Conference on Control, Decision and Information Technologies (CoDIT), (2016), 438–443.

[8] C. MATEUS, I. MARQUES, M.E. CAPTIVO: *Local search heuristics for a surgical case assignment problem*, Operations Research for Health Care, **17** (2018), 71–81.

[9] N.A.A. RAHMIN: *Online scheduling in operating theatre scheduling*, M. Phil. Thesis, University of Southampton, 2018.

[10] K.S. JUNG, M. PINEDO, C. SRISKANDARAJAH, V. TIWARI.: *Scheduling elective surgeries with emergency patients at shared operating rooms,* Production and Operations Management Society, **28**(6) (2019), 1407–1430.

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, UNIVERSITY PUTRA MALAYSIA
43400 SERDANG, SELANGOR, MALAYSIA.
Email address: neeshaalimin@gmail.com

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, UNIVERSITY PUTRA MALAYSIA
43400 SERDANG, SELANGOR, MALAYSIA.
Email address: aliza@upm.edu.my

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, UNIVERSITY PUTRA MALAYSIA
43400 SERDANG, SELANGOR, MALAYSIA.
Email address: ibragimov@upm.edu.my

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, UNIVERSITY PUTRA MALAYSIA
43400 SERDANG, SELANGOR, MALAYSIA.
Email address: nazihah@upm.edu.my