Lecturer Reza BEHMANESH, PhD
E-mail: rezaehs@yahoo.com
Department of Industrial Engineering, Science and Research Branch
Islamic Azad University, Tehran, Iran
Professor Mostafa ZANDIEH, PhD
E-mail: m_zandieh@sbu.ac.ir (Corresponding author)
Department of Industrial Management, Management and
Accounting Faculty, ShahidBeheshti University, G.C., Tehran, Iran
Assistant Professor Seyyed Mohammad HADJI MOLANA, PhD
E-mail: molana@srbiau.ac.ir
Department of Industrial Engineering, Science and Research Branch
Islamic Azad University, Tehran, Iran

MULTIPLE RESOURCE SURGICAL CASE SCHEDULING PROBLEM: ANT COLONY SYSTEM APPROACH

Abstract. *In this paper, we address the multiple-resource surgical case scheduling(MRSCS) problem in multi-operating theatre to minimize makespan. The constraints of no-wait multiple resource flexible job shop problem (FJSP) are considered for formulating MRSCS problem because the environment of the operating room is similar to the FJSP. Minimization of makespan forMRSCS is NP-hard combinatorial optimization problem; hence we employ the ant colony optimization algorithm so as to tackle this problem. These proposed approaches are illustrated by three test cases include small, medium, and large dataset. Consequently, the results of the experiments indicate that ant colony system and elitism ant system outperform traditional ant system since, the mean, the standard deviation, the best, and the worst solutions of proposed algorithms are better than the results of the traditional algorithm.*

Keywords*: MRSCS, ant colony system, FJSP, operating room, Makespan.*

JEL Classification: C61

1. Introduction

During the recent decade, the healthcare industries have been growing and thereby, the costs of this industry are increasing. Based on statistics, healthcare expenditure of US will reach 19.5% of US GDP by 2017(Zhao and Li,2014). On the other hand, operating rooms(ORs) are considered as the engine of the hospital and more than 40% of costs come from various resources of surgery and ORs (Denton et al., 2007). Therefore, improving OR management and patient flow

DOI: 10.24818/18423264/54.1.20.16

seems essential. As a result, planning and scheduling play a crucial role in OR management and hence, some researchers and practitioners are attracted to study operating room planning. In the healthcare, a “surgical schedule” is operated by determining the sequence of the surgical cases as well as assigning them to the ORs, surgeons, nurses in order to optimize objectives such as utilization, idle time, overtime, etc. Operating room scheduling generally deals with strategic, tactical, and operational problems(Behmanesh and Zandieh, 2019). Magerlein and Martin (1978) defined a problem namely, “surgical process scheduling (SPS) problem” that is divided into two sub-problems:(i) advance scheduling; and (ii) allocation scheduling. The first sub-problem is a tactical problem in which a future date is determined for each surgical case. However, the second part is an operational problem in which the start time and resource allocation of the cases are determined. This sub-problem is called “surgical case scheduling (SCS) problem”. In literature, the patient is divided into the elective and non-elective case. An elective case is a patient that is scheduled in advance, while a non-elective case may arrive in the hospital randomly.

In a surgical case processing, the patient is transported from either wards or ambulatory surgical unit (ASU) to the pre-holding unit (PHU). The nurse checks the patient’s documents while he/she is being held in the PHU. The patient occupies both nurse and PHU bed. Then, the patient is moved to the operating room and in this stage, other resources such as the nurse, OR, anesthetist, and surgeon are allocated to the surgical case. At the end of the surgical procedure, the patient is transported to the pre-anesthesia care unit (PACU) where he/she is recovered from residual effects of anesthesia under the care of the nurse in the PACU. In the third stage, the nurse and the PACU bed are allocated to the patient. We focus on the MRSCS problem in which pre-surgery, surgery, and post-surgery durations are deterministic. The resources of the first stage include of PHU beds and nurses. The resources of the second stage include surgeon groups, anesthetists, ORs, and nurses. The resources of the third stage consist of PACU beds and nurses. We consider the makespan as criteria to assess the addressed MRSCS problem. In this paper, the following contributions to the literature are offered:

- We study the MRSCS considering multiple resources and multiple stages as an important topic in the operating theatre planning and then we extend a mixed integer linear programming(MLIP) based on concepts of FJSP to characterize the problem.
- We develop ant colony system (ACS) and elitism ant system (EAS) to solve the large instances of the addressed problem.

Next sections are described as follows: In section 2, several studies regarding the scope of our problem are reviewed. In section 3, the mathematical model for MRSCS is built. In section 4, we propose an algorithm for solving MRSCS.

DOI: 10.24818/18423264/54.1.20.16

In section 5, we provide illustrative examples and computational experiments. Lastly, we conclude and present our suggestions for future research in section 6.

2. Background and related work

Roshanaei et al. (2017) addressed the SCS problem considering only OR and surgeon as resources. The authors have applied exact algorithms to solve the SCS problem in their study however, using exact methods to solve NP-hard combinatorial problems consumes more times against to swarm or evolutionary approaches. Therefore, the meta-heuristics are suitable methods to tackle the SCS with associated savings in time. Al-Refaie et al. (2018) proposed the optimization models for multiple-period scheduling of the patients in ORs and intensive care units (ICUs). Required resources in their work include ORs and surgeons. However, the authors don't consider other resources of the real case. Moreover, the pre-surgery stage is not considered in their model. These gaps can be filled by modeling the MRSCS in the three-stage operating theatre. Liu et al. (2018) studied the SCS problem considering OR and surgeons as resources. The authors constructed a two-step MIP model to maximize the utilization and minimize the cost of the operating theatre and to improve surgeons' satisfaction under uncertainty. The strong point of their work is to use uncertain data in the model but this model can be extended by considering all resources of real cases in MRSCS.

In the literature of the surgical suite management, most researches focus only on the second stage. Also, the optimization of the utilization in OR has received more attention among other resources. There are few studies in the literature that takes into account both OR and surgeon utilization (Roshanaei et al., 2017, HashemiDoulabi et al., 2016, Fei et al., 2009, Jebali et al., 2006). To the best of our knowledge, there is no research in the literature that models the SCS in order to optimize all resources in the OR for the surgery as we research. Also, there are few studies that considered the upstream units (pre-operative) (Pham and Klinkert, 2008). To the best of our knowledge, there is no literature on mathematical modeling of multi-stage and multiple-resource operating theatre scheduling as we construct. Various structures of the shops are taken into account to model and solve the SCS problem. Some researchers observed similarities between the operating room environment and the job shop environment. Pham and Klinkert (2008) developed novel multi-mode blocking job shop scheduling to model the SCS problem, however,(Xiang et al., 2015) considered generalization of job shop and then formulated a multiple-resource FJSP. We assume the sequence of three stages must be followed completely in the operating theatre, and so this assumption needs to the constraint that follows the rules of no-wait flow shop. In the no-wait situation, the orders are not allowed to wait between two successive machines.

DOI: 10.24818/18423264/54.1.20.16

To the best of our knowledge, the model of MRSCS has not been studied according to the constraints of no-wait FJSP.

NP-hard problems in large-scale are tackled by meta-heuristic and swarm intelligence algorithms (Dekhici and Belkadi, 2015, Arun and Kumar, 2017). In many researches in the field of the SCS problem, some heuristic or meta-heuristic procedures such as genetic algorithm (Marques et al., 2014), simulated annealing (Beliën and Demeulemeester, 2007, Beliën et al., 2009), tabu search (Lamiri et al., 2009, Saremi et al., 2013), and ant colony optimization (Xiang et al., 2015, Behmanesh et al., 2019) were developed to achieve near-optimal solutions, because this problem is NP-hard combinatorial optimization problem (Marques et al., 2014). The ACO is compatible with the MRSCS problem since this is classified into the constructive algorithms and these always generate a feasible solution in a short time,while improvement approaches may generate infeasible solutions for the MRSCS after applying their operators and hence more time may be needed to repair the infeasible solutions. Therefore, these reasons motivated us to employ several versions of the ACO algorithm for solving the MRSCS in this study. The first ACO algorithm was introduced by (Dorigo et al., 1991, Dorigo, 1992).In meta-heuristic approaches, a tradeoff between the exploration and the exploitation mechanisms is needed to make an efficient optimization algorithm(Behmanesh, 2016). Therefore, other pheromone updating strategies are considered to improve the exploration of the ACO algorithm. To the best of our knowledge among the area of SCS in literature, no research is available that employs several versions of the ACO algorithm such as EAS, and ACS to solve the problem.

3. Mathematical programming for MRSCS

Since the MRSCS problem is an NP-hard, mathematical programming models cannot be considered as an effective method to solve large-scale problems, but these can provide a basic structure to make an effective heuristic. We develop a model for the MRSCS problem according to FJSP model. There are n elective cases, t resource type and m resource sets for each type. The patient, stage, resource type, and resource of each type are denoted by i, j, r, k, respectively. The parameters and variables of the model are described by the following symbols:

Sets	
I	Set of all the elective patients
SG	Set of specializations for surgeries
I_s	The subset of patients based on specializations
J_i	Set of operations of patient$i \in I$
R	Set of all resource types
O_{ij}	Patient$i \in I$ in stage $j \in J$
R_{ij}	Set of the eligible resource type for operation O_{ij}
$K_{r_{ij}}$	Set of resource type r(exception of surgeon group) $r \in R -$

DOI: 10.24818/18423264/54.1.20.16

	$\{3\}, r_{ij} \in R_{ij}$
K_{r_s}	The subset of all surgeons based on specialization $s \in SG$ in resource type $r = 3$
Parameters	
P_{ijrk}:	The processing time of operation O_{ij} if performed on resource k of type r
M:	A large positive number
n:	Total number of patients
m_r:	Total number of resources for each resource type (8 types)
Decision variables	
ST_{ijrk}:	The start time of operation O_{ij} by resource k of type r
ET_{ijrk}:	The end time of operation O_{ij} by resource k of type r
ET_i :	The completion time of patient i
C_{max}:	Makespan
v_{ijrk}:	Equals to 1 if operation O_{ij} performed on resource k of type r, equals 0 otherwise
z_{ijhgrk}:	Equals to 1 if operation O_{ij} precedes operation O_{hg} on resource k of type r, equals 0 otherwise
g_{ijrk}:	Equals to 1 if operation O_{ij} performed by surgeon k of special group r, equals 0 otherwise. This variable is used for all involved multiple-resources in the process of the surgery.

A general model of MILP is constructed for the MRSCS problem. In the following model, Equation (1) states minimum makespan. Constraint(2) reflects makespan according to the completion time of the patients. Equation (3)determines the end times of the patients at the end of the last stage.

$$\min C_{max} \tag{1}$$

$s.t.$

$$ET_i \leq C_{max} \quad \forall\, i \in I \tag{2}$$

$$ET_i \geq \sum_{k \in K_{r_{ij}}} ET_{ijrk}\ \forall\, i \in I, j = 3, r = 8 \tag{3}$$

Constraints (4) and (5) guarantee that the difference between the start time and the end time of the operation for the patients during all stages (only for surgeons in the second stage) is equal to the processing time of these stages on eligible resources. Constraints (6) and (7) make sure same requirements of equations (4) and (5) but for other involved resources in the second stage(resource types #4-#6).

$$ST_{ijrk} + ET_{ijrk} \leq M v_{ijrk} \forall\, i \in I / I_s, j \in \{1,2,3\}, r \in R_{ij}\{1,2,3,7,8\}, k \in K_{r_{ij}} \tag{4}$$

$$ST_{ijrk} + P_{ijrk} - M\left(1 - v_{ijrk}\right) \leq ET_{ijrk} \forall\, i \in I / I_s, j \in \{1,2,3\}, r \in R_{ij}\{1,2,3,7,8\}, k \in K_{r_{ij}} \tag{5}$$

DOI: 10.24818/18423264/54.1.20.16

$$ST_{ijrk} + ET_{ijrk} \leq Mv_{ijrk} \quad \forall\, i \in I, j = 2, r \in R_{ij}\{4,5,6\}, k \in K_{r_{ij}} \tag{6}$$

$$ST_{ijrk} + \sum_{k \in K_{r_s}} P_{ijrk} g_{ijrk} - M(1 - v_{ijrk}) \leq ET_{ijrk} \quad \forall\, i \in I, j = 2, r \in R_{ij}\{4,5,6\}, k \in K_{r_{ij}} \tag{7}$$

Constraints (8) and (9) specify that two different operations of O_{ij} and O_{hg} cannot be processed at the same time on any resource in set $R_{ij} \cap R_{hg}$. Equation (10) ensures that *j*th operation of the patient must be exactly started after the end time of *(j-1)*th of the operation of the same patient.

$$ET_{hgrk} - Mz_{ijhgrk} \leq ST_{ijrk} \quad \forall\, i, h \in I / I_s, i \ll h, j, g \in J, r \in R_{ij} \cap R_{hg}, k \in K_{r_{ij}} \cap K_{r_{hg}} \tag{8}$$

$$ET_{ijrk} - M(1 - z_{ijhgrk}) \leq ST_{hgrk} \quad \forall\, i, h \in I / I_s, i \ll h, j, g \in J, r \in R_{ij} \cap R_{hg}, k \in K_{r_{ij}} \cap K_{r_{hg}} \tag{9}$$

$$\sum_{k \in K_{r_{ij}}} ST_{ijrk} = \sum_{k \in K_{r_{i(j-1)}}} ET_{i(j-1)rk} \; \forall\, i \in I / I_s, j \in \{2,3\}, r \in R_{ij} \tag{10}$$

Constraint (11) and (12) make sure that all required resources for each stage must have the same start and end time, respectively.

$$\sum_{k \in K_{r_{ij}}} ST_{ijrk} = \sum_{k' \in K_{r_{ij}}} ST_{ijr'k'} \; \forall\, i \in I / I_s, j \in J, r, r' \in R_{ij} \tag{11}$$

$$\sum_{k \in K_{r_{ij}}} ET_{ijrk} = \sum_{k' \in K_{r_{ij}}} ET_{ijr'k'} \; \forall\, i \in I / I_s, j \in J, r, r' \in R_{ij} \tag{12}$$

Equation (13) enforces that one and only one resource from each resource type must be assigned to an operation of the patient. Finally, constraint (14) demands that one and only one surgeon from each group can operate. Constraints (15-20) denote positive and binary decision variables.

$$\sum_{k \in K_{r_{ij}}} v_{ijrk} = 1 \quad \forall\, i \in I / I_s, j \in J, r \in R_{ij} \tag{13}$$

$$\sum_{k \in K_{r_s}} g_{ijrk} = 1 \quad \forall\, i \in I_s, j = 2, r = 3 \tag{14}$$

$$ST_{ijrk}, \quad ET_{ijrk} \geq 0 \quad \forall\, i \in I, j \in J, r \in R_{ij}, k \in K_{r_{ij}} \tag{15}$$

$$ET_i \geq 0 \quad \forall\, i \in I \tag{16}$$

$$C_{max} \geq 0 \tag{17}$$

$$v_{ijrk} \in \{0,1\} \forall\, i \in I, j \in J, r \in R_{ij}, k \in K_{r_{ij}} \tag{18}$$

$$g_{ijrk} \in \{0,1\} \forall\, i \in I_s, j = 2, r = 3, k \in K_{r_s} \tag{19}$$

$$z_{ijhgrk} \in \{0,1\} \forall\, i, h \in I / I_s, i \ll h, j, g \in J, r \in R_{ij} \cap R_{hg}, k \in K_{r_{ij}} \cap K_{r_{hg}} \tag{20}$$

DOI: 10.24818/18423264/54.1.20.16

4. Ant colony optimization (ACO) as a method

A bi-level ACS/EAS algorithm is proposed that in its first level, surgical cases are taken into account as cities of the tour and thereby, the sequence of the cases is determined in this level. Then in the second level, required multiple-resources of every stage are assigned to patients. Therefore, two graphs are generated by the ACS/EAS algorithm that sequence of patients are determined in the first (outer) graph and resource allocation is done inside the second (inner) graph. The pseudo code of the algorithm is presented as follow.

Algorithm1. Bi-level ACS
1. Input: an instance MRSCS of a combinatorial problem Pm 2. Initialize Pheromone Values and other parameters (*it*,*m*), β, α, ρ **3. While** stop criteria not met (i<it) **do** 4. Put *m* ants on a random node (surgical case) 5. Construct an ant solution with the *resources start time (0)* 6. **While** all ant shave not been assigned to nodes, $k < m$**do** 7. Initialize tabu:= φ; surgical cases (SC):= I 8. Construct an ant solution by visiting a node *i* in the outer graph according to the transition rule Eqs(26,27,30) 9. $tabu = tabu \cup \{I_i\}$ and $I = I \backslash \{I_i\}$ 10. Determine *start time (ST)* and *end time* (*ET*) of each *SC* according to the no-wait 11. Determine available resources for surgical case 12. Ant enters into the inner graph and constructs available resource set G 13. Construct a resource allocation for ant solution 14. **For** each resource type *t* **do** 15. Construct an ant solution by visiting a node *tm* in the inner graph according to the transition rule Eqs(28,29) 16. Local update inner pheromone trial based on Eq(31) 17. **End for** 18. Update time window of occupied resources **19. End while** 20. Calculate single objective (*makespan*) for an ant solution 21. Compare iteration based best ant solution, record its tabu as the global best solution 22. Global update in both outer and inner pheromone trial based on Eqs(21-25) **23. End while** **24. End while** **25. Plot related graphs (**Gantt chart, convergence**)** **26. Return** the best solution found

DOI: 10.24818/18423264/54.1.20.16

According to the pseudo code, the construction of the outer graph is commenced by putting m ants on surgical cases randomly and then initial pheromone is provided. After that, multiple resources are allocated to the patient that chosen by an ant and hence, the inner graph is made by choosing each resource from each type of resource. Then, the time window of selected resources, as well as local pheromone, are updated and ant exits from the inner graph and goes to the outer graph in order to visit other surgical cases for probable choosing. These stages are repeated until an ant implement outer graph so that all surgical cases are sequenced and required resources are assigned to all patients. Moreover, this loop must be repeated for *(m)* ants. For example, ant #1 goes to visit surgical case #1 and then goes to visit required resources like bed #2 from PHU as resource type, anesthetist #5 from anesthetist group as resource type, etc. Then, the ant goes to visit another surgical case according to transition rules, after assigning the resources to the case #1 and this process is repeated until scheduling all patients for all ants. Finally, makespan obtained by all ants are compared and the best is recorded and the pheromones are updated for the next iteration of the algorithm. The algorithm is iterated until stopping conditions i.e. maximum iteration.

Pheromone updating procedure related to proposed algorithms and transition rules are described in this section. EAS was introduced by (Dorigo et al., 1996), and ACS was introduced by (Dorigo and Gambardella, 1997).Rules of ACS are formulated based on the following equations. The Strategy of pheromone updating in the outer graph is according to the following equation:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}^{global-best} \quad (21)$$

$$\Delta\tau_{ij}^{global-best} = \begin{cases} \frac{Q}{C_{max}} if\ ant\ k\ goes\ through\ (i,j)\ in\ this\ iteration \\ 0, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad o.w. \end{cases} \quad (22)$$

and $\Delta\tau_{ij}^{global-best}$ is increasing the value of the pheromone from case *i* to case *j* in the iteration of the best route and C_{max} is makespan of the best agent. And in local update or exploration strategy, the ant *#k* decreases pheromone when it adds a component c_{ij}(the route of patient *i* to patient *j*) to its partial solution in accordance with the following:

$$s_k \cup \{c_{ij}\} \Rightarrow \tau_{ij}(t) = (1-\aleph).\tau_{ij}(t) + \aleph.\tau_0 \quad (23)$$

where, parameter ℵ controls the exploration factor, and initial pheromone (small constant value) is notated by τ_0.Besides, the strategy of the pheromone updating in the inner graph is according to the following equation:

$$in(\tau_{tm}^{i}(t+1)) = (1-\rho).in(\tau_{tm}^{i}(t)) + \Delta in(\tau_{tm}^{ibest}) \quad (24)$$

DOI: 10.24818/18423264/54.1.20.16

$$\Delta in(\tau_{tm}^{ibest}) = \begin{cases} \frac{Q}{C_{max}} if\ ant\ k\ goes\ through\ surgery(\ i)\ with\ resource\ graph\ (t,m) \\ 0, \qquad o.w. \end{cases} \quad (25)$$

and $\Delta in(\tau_{tm}^{ibest})$ is increasing the value of the pheromone for case *i* on resource *m* from type *t* in the iteration of the best route and C_{max} is makespan of the best agent. Then, transition rules in the graphs are shown as the following formulations. In the outer graph, the probability of the choice of case *j* after case *i* is shown as follows:

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{l\in I_i^k}[\tau_{il}(t)]^{\alpha}.[\eta_{il}]^{\beta}} \qquad if\ j \in I_i^k \quad (26)$$

where, the pheromone value in the current iteration for case *i* to *j* is denoted by $\tau_{ij}(t)$, and the pheromone, and heuristic factors are denoted by α, β, and also η_{ij} denotes the heuristic information of problem for the case *i* to *j* that is shown according to the following equation:

$$\eta_{ij} = (T_{j1} + T_{j3} + \max(T_{j2}^{SGm}))/(T_{j1} + T_{j3} + \max\left(T_{j2}^{SGm}\right) + A) \quad (27)$$

where, T_{j1}, T_{j2}^{SGm}, T_{j3} are processing time pre-surgery, surgery, and post-surgery, also parameter A is a constant value. In the inner graph, the probability of the choice of resource *m* from type *t* is shown as follows:

$$P_{tm}^{ki}(t) = \frac{\left[in(\tau_{tm}^{i}(t)).in(\lambda_m)\right]^{\alpha}.[in(\eta_{tm})]^{\beta}}{\sum_{g\in G_i^k}\left[in(\tau_{tg}^{i}(t)).in(\lambda_g)\right]^{\alpha}.[in(\eta_{tg})]^{\beta}} \qquad if\ j \in G_i^k \quad (28)$$

where, $in(\tau_{tm}^{i}(t))$ is pheromone value in the current iteration for resource *m* from type *t*, and the heuristic information for resource *m* from type *t* is denoted by $in(\eta_{tm})$that is shown as follows:

$$in(\eta_{tm}) = B/(ES_{il}^{tm} + T_{il}^{tm}) \quad (29)$$

where, ES_{il}^{tm} is the earliest time of resource *m* from type *t* for case *i* in stage *l* and $T_{il}^{tm}$is operating time of case *i* in stage *l* when resource *m* from type *t* is applied. Also, parameter B is a constant value. ACS works according to pseudo-random proportional choice rule using a controller parameter namely qm_0. Agent *#k* chooses patient *j* after patient *i* with probability less than or equal to qm_0 based on the following equation (greedy walking):

$$j = \arg \quad \max\{[\tau_{il}(t)]^{\alpha}.[\eta_{il}]^{\beta}\} \qquad if\ l \in I_i^k \quad (30)$$

DOI: 10.24818/18423264/54.1.20.16

And on the other hand, agent *#k* chooses patient *j* after patient *i* with probability greater than qm_0 based on the probabilistic choice rule of ant system (equation 26).Also, there is a new strategy of the pheromone updating that is denoted by $in(\lambda_{tm})$. This strategy is very effective for resource utilization and is described in the below equation:

$$in(\lambda_{tm}) = in(\lambda_{tm}) - q_0 \quad (31)$$

where, q_0 states decremented pheromone value. It must be noted that all formulations exception of equations (23,30) are considered for EAS algorithm.

5. Results and statistical analysis

5.1. Examples and data

To assess the proposed algorithms, we considered three cases (small, medium, and large) those are different in operating time, the number of patients, and assigned resources. Each case consists of three various examples. Cases category and their specifications are shown in Table 1. Surgeries are classified into five types based on their duration (Table 2). The surgery's duration is according to simulation model constructed by (Xiang et al., 2015) and each problem was generated based on different durations of surgery type.

Table 1.Surgical cases and the resources

P	Surgical case	PHU bed	Nurse	Srg	ORs	PACU bed	Anesthesia	Surgery type (S:M:L:EL:S)
1	8	1	5	5	2	2	5	2:4:1:1:0
2	10	2	8	6	4	4	6	2:6:1:1:0
3	10	2	8	6	4	4	6	2:5:2:1:0
1	15	3	10	6	4	3	8	3:9:2:1:0
2	20	3	15	10	5	4	8	4:12:3:1:0
3	20	3	15	10	5	4	8	4:10:3:3:0
1	30	4	19	10	6	5	9	7:16:3:2:2
2	30	4	22	12	6	5	11	5:15:3:4:3
3	30	5	22	12	6	6	12	3:15:3:4:5

As it is observed, three problems of each case are different in size of surgeries (column 2), size of resources (column 3-8), and surgery type structure (column 9).

Table 2. The classification of the surgeries

Desc	Pre-surgery	Surgery Case					Post-surgery
		S	M	L	EL	Special	
Duration	R.N	R.N	R.N	R.N	R.N	R.N	R.N
(min)	(8,2)	(33,15)	(86,17)	(153,17)	(213,17)	(316,62)	(28,17)

DOI: 10.24818/18423264/54.1.20.16

5.2. Parameter setting of proposed algorithms

Various parameters in ACS and EAS algorithms are effective in their performance especially solution quality and computational time. For instance, some parameters are considered in proposed algorithms according to Table 3.

Table 3. Parameters for algorithms

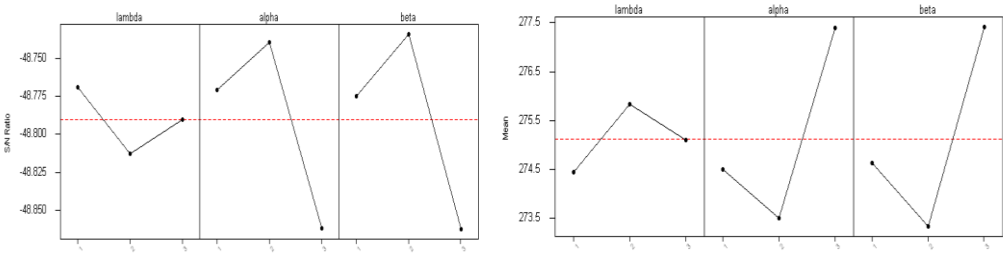
Parameter	Description	Parameter	Description
(m, Max-It)	(The No. of ants, iterations)	β	heuristic factor
ρ	evaporate rate	q_0	decremented pheromone
α	pheromone factor	λ_0	resource-related pheromone

We designed experiments according to the Taguchi design of experiment in order to set parameters for algorithms on test cases. Therefore, three factors namely, λ_0, α, and β were set to three levels as shown in Table 4. In the first place, the parameter setting was done by considering the signal to noise ratio (S/N) to build a robust algorithm, then the mean of the makespan was taken to set parameters.

Table 4.Factors and their levels for optimizing ACS

Factor	Level 1	Level 2	Level 3
λ_0 (Factor	1	5	9
α (Factor B)	0.1	0.9	3
β (Factor C)	0.1	2	9

The effect of these factors on S/N and mean of makespan is shown in Figure 1.As it is shown, the best set of α, β, and $\lambda_0$are determined in mediate level e.g. 0.9 for α and 2 for β and 5 for $\lambda_0$in order to build the robust and efficient algorithm.



(a) signal to noise ratios (b) average of makespan

Figure 1.Main effect plot of three factors (λ_0, α, and β)

DOI: 10.24818/18423264/54.1.20.16

MATLAB and GAMS were applied for coding the algorithms. For this aim, a computer with Core (TM) Duo CPU T2450, 2.00 GHz, and 1 GB of RAM was used. Final setting parameters of each algorithm for three cases are displayed in Table 5.These experiments were done for all algorithms on all test cases.

Table 5.Final setting parameters for all algorithms

Case no.	Algorithms	(max-it, m)	q_0	λ_0	α	β	ρ
1	ACS	60,50	0.4	5	0.9	2	0.2
	EAS	60,50	0.4	5	0.9	5	0.2
	AS	60,50	0.1	4	0.9	5	0.1
2	ACS	80,50	0.4	5	0.9	2	0.2
	EAS	80,50	0.4	5	0.9	5	0.2
	AS	80,50	0.1	4	0.9	5	0.1
3	ACS	100,60	45	5	0.9	2	0.2
	EAS	100,60	47	5	0.9	5	0.2
	AS	100,60	0.1	4	0.9	5	0.1

Also, Figure 2 displays the convergence value of three algorithms (best solution) versus different populations from 10 ants to 50 ants. As it is indicated, more population impacts on the performance of algorithms to achieve a better solution.

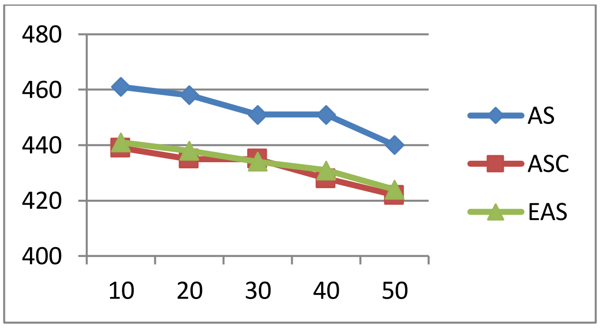


Figure2. Convergence diagram with different populations of the ant colony

5.3. Comparison between algorithms

Firstly, we ran three algorithms and MILP model on a very small case as presented in Table 6along with small data of Table 1to validate our approach. All algorithms were repeated 30 times and then the mean of the makespan found by ACS and EAS was compared to results of MILP. Algorithms are validated in comparison with MILP model. On the other hand, global solutions found by proposed algorithms outperform basic algorithm as shown in Table 7.

DOI: 10.24818/18423264/54.1.20.16

Table 6. Test cases for comparing MILP and ACO algorithms

P	Surgical case	PHU bed	Nurse	Srg	ORs	PACU bed	Anesthesia	Surgery type (S:M:L:EL:S)
1	3	2	6	6	2	2	2	0:2:1:0:0
2	5	1	9	6	6	2	6	2:2:1:0:0

Table 7.The validation of the AS, ACS, and EAS algorithms

Sample	**GAMS**	AS	ACS	EAS	$(\frac{ACS-GAMS}{GAMS}\%)$	CT* (ACS/GAMS)
1(very small)	**181**	184.20	**181.00**	**181.00**	0.00%	100%
2(very small)	**234**	238.80	**234.00**	**234.00**	0.00%	6.66%
3(small)	**402**	427.56	**409.63**	409.66	1.90%	3.5%
4(small)	**253**	279.80	**265.13**	267.10	4.79%	2.2%
5(small)	**252**	281.26	**264.80**	270.70	5.08%	2.2%
6(medium)	**---**	362.00	**348.60**	353.83	---	---

*. Computational Time

All algorithms were repeated 30 times for each instance in order to compare three algorithms. RPD index was applied to homogenize all data because obtained makespan values of problems are heterogeneous. RPD value of each makespan is obtained according to the following equation:

$$RPD_{ij} = \frac{makespan_{ij} - min_j(makespan_{ij})}{min_j(makespan_{ij})} \tag{32}$$

where the index of the problem is notated by i and j is introduced as the index of the algorithm. In order to compare three algorithms for solving large-scale instances, we ran these using data of Table 1.The results of the normalized experiments based on RPD are indicated in Table 8 and the ANOVA test was applied to verify whether convergence values of algorithms are different significantly. The result of the ANOVA for comparison of three algorithms is presented in Table 9, and it is indicatedthatconvergence valuesof algorithms are different.

Table 8. Results of RPD for algorithms

Case.No.	Mean of RPD (Make span)		
	ACS	EAS	AS
Small-P1	0.003355	0.003432	0.047281
Small-P2	0.003149	0.010643	0.058774
Small-P3	0.001129	0.023555	0.063493

DOI: 10.24818/18423264/54.1.20.16

Medium-P1	0.002012	0.017094	0.040626
Medium -P2	0.008583	0.009560	0.050891
Medium -P3	0.007000	0.004822	0.046769
Large-P1	0.008727	0.001460	0.036642
Large -P2	0.005579	0.002776	0.038574
Large -P3	0.003529	0.002240	0.039452
MEAN	**0.004785**	**0.008398**	**0.046945**

Table 9.ANOVA test for comparison the ACO algorithms

***Hypothesis*:**	H_0: $\mu_{AS} = \mu_{ACS} = \mu_{EAS}$ H_1: Otherwise				
Source of variation	DF	SS	MS	F	*P*-value
Algorithm	2	0.294872	0.147436	839.982	0.000
Error	807	0.141647	0.000176		
Total	809	0.436519			
Result:	Reject H_0				

Furthermore, both Scheffe's comparison and Tukey's comparison were applied in ANOVA to determine the relationship between algorithms for finding the algorithm with a qualified and efficient optimal solution. The results of the comparison tests are presented in Table 10 and as it is shown, ACS and EAS outperform the AS significantly and moreover, the ACS outperforms the EAS. As a consequence, we infer that our proposed approaches are promising meta-heuristic algorithms to provide good solutions for solving MRSCS problem because of their better exploration in comparison with AS.

Table 10. Comparison Tests for three algorithms in the convergence value

Alg. (A)	Alg. (B)	Mean Difference (A-B)	*P*-value		Result
			Scheffe	Tukey	
ACS	EAS	-.00361318757*	0.007	0.005	ACS < EAS
ACS	AS	-.04215989027*	0.000	0.000	ACS < AS
EAS	AS	-.03854670269*	0.000	0.000	EAS < AS
Result:	$\mu_{ACS} < \mu_{EAS} < \mu_{AS}$				

*. The mean difference is significant at the 0.05 level

Besides, Figure 3 presents median and inter-quartile range (IQR) values of the algorithms on the test problems. The size of each rectangle displays the IQR. The short line at the end of each rectangle indicates the maximum and minimum values and the median is represented by the short line in each rectangle. As it is indicated,

DOI: 10.24818/18423264/54.1.20.16

ACS occupies the lowest position in the graph as compared to the EAS and AS while AS occupies the highest position. On the other hand, the ACS rectangle occupies the smallest area, and this indicates that the ACS has the smallest degree of variance.

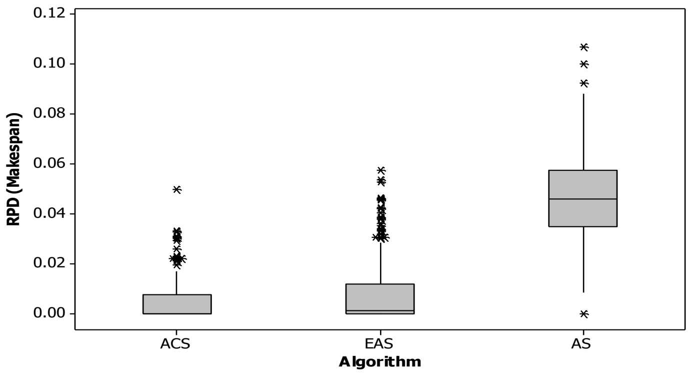


Figure3.Box plot of the mean of RPD (makespan) value on all test problems to compare the AS, ACS, and EAS

The average, standard deviation, the best and the worst solution by running algorithms on each test problem in the dataset are presented in Table 11.In each test case, the results of the ACS and EAS are more efficient than those of AS.

Table 11.Comparison between AS, ACS, and EAS

Cases	Problems	Algorithms	Average	St.Dev	Min	Max
Small	**P1**	ACS	**409.6333**	**2.008316**	**407**	**415**
		EAS	409.6667	2.309401	407	415
		AS	427.5667	4.492778	420	438
	P2	ACS	**265.1333**	**4.174911**	**259**	277
		EAS	267.1	4.27785	260	276
		AS	279.8	3.845284	272	286
	P3	ACS	**264.8**	**3.336321**	**259**	**273**
		EAS	270.7	3.724569	262	276
		AS	281.2667	4.184811	274	290
Medium	**P1**	ACS	**348.6**	**3.747183**	**340**	**358**
		EAS	353.8333	5.47775	344	363
		AS	362	5.30452	350	372
	P2	ACS	**366.6333**	**4.64226**	**356**	379
		EAS	367	5.489802	356	377
		AS	382	3.859605	374	389

DOI: 10.24818/18423264/54.1.20.16

	P3	ACS	433.4333	**3.94517**	**422**	**439**
		EAS	**432.5**	4.297152	424	442
		AS	450.5333	5.624658	440	461
Large	**P1**	ACS	529.6333	**3.189242**	524	535
		EAS	**525.8333**	3.705386	**518**	**533**
		AS	544.3	6.798326	531	558
	P2	ACS	628.1333	**3.104317**	620	**633**
		EAS	**626.4**	4.79655	**615**	634
		AS	648.7333	6.180801	635	665
	P3	ACS	732.8333	**4.705707**	722	744
		EAS	**731.9**	4.929503	**722**	**742**
		AS	759.0333	8.965387	746	778

As it is indicated, both proposed algorithms i.e. ACS and EAS outperform traditional AS according to the results of the average, standard deviation, best, and worst. Although the ACS outperforms the EAS significantly based RPD, we can discuss each test problem, separately. For instance, in both small and medium test case problems, the average of the solution obtained by ACS is better than those of EAS whereas, results of EAS are better than those of ACS in large test problems. Besides, like these results are seen in the best solution obtained so that ACS obtains the better solution for small and medium tests while the EAS obtains the efficient solutions for large tests. On the other hand, it is observed that range of solutions obtained by EAS is larger than those of ACS according to standard deviation and therefore, applying the ACS in all cases is more robust than EAS. As a consequence, we can point out that ACS outperforms the EAS exactly and accurately, although the EAS presents solutions better than ACS sometimes in large cases.

6. Conclusion

In this paper, we proposed a new approach so as to tackle MRSCS problem. In this paper, new meta-heuristic approaches with high robustness and high quality are introduced to solve the MRSCS problem. Our methodology is based on ACO algorithms so that we developed bi-level ACS and EAS. To illustrate our proposed algorithms, we generated three cases with different sizes. In accordance with results and discussions, it can be concluded that both ACS and EAS outperform traditional AS. Additionally, the ACS outperforms the EAS and therefore our proposed bi-level ACS obtains solutions more exact and accurate than EAS although the EAS gives us the better mean of the makespan on large test problems in comparison to the ACS. At last, we suggest some directions for extending the ACO for future research on this topic. A developed max-min ant system (MMAS) can be a good method to improve the proposed algorithms of this paper. Also, emergency cases can be considered in MRSCS problem and a new

DOI: 10.24818/18423264/54.1.20.16

algorithm can be developed to tack lean online SCS in the real world. On the other hand, the development of the ACO algorithm for multi-objective MRSCS problem can be considered as novel research.

REFERENCES

[1] **Al-Refaie, A., Judeh, M., Chen, T. (2018)** ***,Optimal Multiple-Period Scheduling and Sequencing of Operating Room and Intensive Care Unit***;*operational Research*,18(3), pp. 645-670;

[2] **Arun, B., Kumar, T.V. (2017)*, Materialized View Selection Using Bumble Bee Mating Optimization***;*International Journal of Decision Support System Technology*, 9(3), pp. 1-27;

[3] **Behmanesh, R. (2016)*, Nephron Algorithm Optimization: Inspired of the Biologic Nephron Performance;***International Journal of Applied Metaheuristic Computing*, Vol. 7(1), pp. 38-64;

[4] **Behmanesh, R., Zandieh, M., Hadji Molana, S. (2019)*, The Surgical Case Scheduling Problem with Fuzzy Duration Time: An Ant System Algorithm;***Scientia Iranica*, 26(3), pp. 1824-1841;

[5] **Behmanesh, R.& Zandieh, M. (2019)*, Surgical Case Scheduling Problem with Fuzzy Surgery Time: An Advanced Bi-Objective Ant System Approach;*** *Knowledge-Based Systems*, 104913;

[6] **Beliën, J., Demeulemeester, E. (2007)** ***, Building Cyclic Master Surgery Schedules with Leveled Resulting Bed Occupancy;*** *European Journal of Operational Research*, Vol. 176, pp. 1185–1204;

[7] **Beliën, J., Demeulemeester, E., Cardoen, B. (2009)** ***, A Decision Support System for Cyclic Master Surgery Scheduling with Multiple Objectives;*** *J sched*, 12, pp. 147-161;

[8] **Dekhici, L., Belkadi, K. (2015)** ***, A Bat Algorithm with Generalized Walk for the Two-Stage Hybrid Flow Shop Problem;*** *International Journal of Decision Support System Technology*, 7(3), pp. 1-16;

[9] **Denton, B., Viapiano, J., Vogl, A. (2007)*, Optimization of Surgery Sequencing and Scheduling Decisions under Uncertainty;***Health Care Management Science*, 10(1), pp. 13–24;

[10] **Dorigo, M., Maniezzo, V., Colorni, A. (1991)*, Positive Feedback as a Search Strategy;***Technical Report 91-016, Dip. Elettronica*, Politecnico di Milano, Italy;

[11] **Dorigo, M. (1992)*, Optimization, Learning, and Natural Algorithms;*** *PhD Thesis*, Dip. Elettronica, Politecnico di Milano, Italy, (in Italian);

DOI: 10.24818/18423264/54.1.20.16

[12] **Dorigo, M., Maniesso, V., Colorni, A. (1996),** ***The Ant System: Optimization by a Colony of Cooperating Agents;****IEEE Trans. Systems Man Cybernet*. B 26(1), pp. 29–41;

[13] **Dorigo, M., Gambardella, L.M. (1997) ,** ***Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem;*** *IEEE Trans. Evolut. Comput*. 1, pp. 53–66;

[14] **Fei, H., C. Chu, N. Meskens. (2009),** ***Solving a Tactical Operating Room Planning Problem by a Column-Generation-Based Heuristic Procedure with Four Criteria;****Annals of Operations Research*, 166(1). pp. 91–108;

[15] **Hashemi Doulabi, S. H., Rousseau, L. M., Pesant. G. (2016) ,** ***A Constraint-Programming-Based Branch-and-Price-and-Cut Approach for Operating Room Planning and Scheduling,****informs Journal onComputing*, 28(3), pp. 432–448;

[16] **Jebali, A., Hadjalouane, A., & Ladet, P. (2006),** ***Operating Rooms Scheduling;****International Journal of Production Economics*, 99(1–2), pp. 52–62;

[17] **Lamiri M., Grimaud F., Xie X. (2009) ,** ***Optimization Methods for a Stochastic Surgery Planning Problem;****Int J Prod Econ*, 120(2), pp. 400–410;

[18] **Liu, Hongwei; Zhang, Tianyi; Luo, Shuai (2018),** ***Operating Room Scheduling and Surgeon Assignment Problem under Surgery Durations Uncertainty;****Technology and Health Care*,26(2), pp. 297-304;

[19] **Magerlein, J.M., Martin, J.B. (1978) ,** ***Surgical Demand Scheduling: A Review;****Health Service Research*, 13, pp. 418–433;

[20] **Marques, I., Captivo, M.E., Vaz Pato, M. (2014),** ***Scheduling Elective Surgeries in a Portuguese Hospital Using a Genetic Heuristic;****Operations Research for Health Care*, Vol. 3(2), pp. 59-72;

[21] **Pham, Dinh-Nguyen, Klinkert, Andreas (2008) ,** ***Surgical Case Scheduling as a Generalized Job Shop Scheduling Problem;*** *European Journal of Operational Research*, Vol. 185, pp. 1011–1025;

[22] **Roshanaei, V., Luong, C., Aleman, D.M., Urbach, D.R. (2017),*****Collaborative Operating Room Planning and Scheduling;****Informs journal on computing*, Vol. 29(3), pp. 558-580;

[23] **Saremi, A., Jula, P., ElMekkawy, T., Wang, G.G. (2013),** ***Appointment Scheduling of Outpatient Surgical Services in a Multistage Operating Room Department;*** *Int. J. Production Economics*, Vol. 141, pp. 646–658;

[24] **Xiang, W., Yin, J., Lim, G. (2015) ,** ***An Ant Colony Optimization Approach for Solving an Operating Room Surgery Scheduling Problem;****Computers & Industrial Engineering*, Vol. 85, pp. 335–345;

[25] **Zhao, Z., Li, X. (2014) ,** ***Scheduling Elective Surgeries with Sequence-Dependent Setup Times to Multiple Operating Rooms Using Constraint Programming;****Operations Research for Health Care*, Vol. 3(3), pp. 160-167.

DOI: 10.24818/18423264/54.1.20.16