Tailoring a Red Deer Algorithm for Solving an Integrated Surgery Planning and Scheduling Problem

Asma Ouled Bedhief [1], Amira Brahmi [2], Najla Aissaoui [3], Safa Bhar Layeb [4]

Abstract— **This paper tackles a challenging Integrated Surgery Planning and Scheduling Problem that simultaneously considers operating rooms and recovery beds. The objective is to minimize the maximum daily closing time of the operating theatre, a key metric for optimizing hospital resource utilization and improving the quality of patient care. To address this NP-hard problem, we propose using the Red Deer Algorithm (RDA), a recently developed evolutionary metaheuristic inspired by the mating behavior of Scottish Red Deer during their breeding season. To evaluate the effectiveness of the proposed approach, computational experiments were carried out on a series of benchmark instances. The results highlight the effectiveness of the RDA in generating high-quality surgical schedules, contributing to improved operational efficiency and enhanced patient outcomes..**

I. Introduction

Healthcare systems have been increasingly challenged in recent years to deliver high-quality care with limited resources. Among the various departments within a hospital, the operating theatre stands out as one of the most critical and expensive, as it comprises several key resources including operating and recovery rooms and plays a pivotal role in hospital operations.

The operating rooms (OR) department, in particular, holds immense significance, accounting for 60% to 70% of all hospital admissions, primarily due to surgical interventions. Moreover, it is estimated that the OR accounts for more than 40% of a hospital's total expenses [1]. These statistics highlight the necessity of efficient management of the surgical process, which involves careful planning, scheduling, and resource coordination. In this context, the integration of surgery planning and scheduling tasks has attracted increased attention in recent literature. Surgery planning focuses on assigning a surgery date to each patient, while scheduling involves allocating patients to resources and determining the order of surgeries on the scheduled day. Integrating these tasks aims to improve coordination and optimize the performance of the surgical unit, ultimately enhancing overall hospital efficiency.

Several studies have explicitly addressed the integration of planning and scheduling tasks. For instance, Guido et al. [2] developed a multi-objective integer linear programming model aimed at optimizing hospital operating room planning and scheduling. They also proposed a hybrid genetic algorithm approach to improve convergence towards Pareto-optimal solutions. Similarly, Aringhieri et al. [3] introduced a tabu search algorithm to solve an integrated OR planning and scheduling problem that considers post-surgery bed availability. Their approach begins with a greedy constructive method to generate an initial solution, which is then refined through a search strategy aimed at optimizing patient and hospital utilities. Extensive computational experiments on real data sets demonstrated the effectiveness of the proposed metaheuristic. Akbarzadeh et al. [4] tackled the integration of surgical case planning and scheduling by incorporating nurse re-rostering and patient-nurse assignments, with the goal of maximizing operating room utilization and profitability. They developed a heuristic approach based on linear programming and column generation to produce high-quality feasible solutions. Additionally, Doulabi et al. [5] proposed a branch-and-price-and-cut algorithm for integrated OR planning and scheduling, focusing on the allocation and sequencing of surgeries over a planning horizon. Their results indicated an average optimality gap of 2.81%. Wang et al. [6] addressed planning and scheduling for laminar-flow operating rooms, which involves assigning surgeries to specific dates while accounting for various resource constraints and minimizing overtime and undertime costs. To solve their problem, they developed a particle swarm optimization algorithm combined with heuristic rules. Other studies have proposed alternative solution methods for integrated surgery planning and scheduling problems [7, 8, 9, 10]. For a detailed literature review, the interested readers can refer to [11, 12].

A thorough review of the literature reveals that, despite a growing number of studies addressing the integrated surgery planning and scheduling problem, this research direction

1: LR-OASIS, Department of Industrial Engineering, National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia Department of Industrial Engineering, National Engineering School of Bizerte, University of Carthage, Tunis, Tunisia
Email: asma.ouledbedhief@enib.ucar.tn
2: LR-OASIS, Department of Industrial Engineering, National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia
Email: amira.brahmi@enit.utm.tn
3: LR-OASIS, Department of Industrial Engineering, National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia Department of Industrial Engineering, National Engineering School of Carthage, University of Carthage, Tunis, Tunisia
Email: najla.aissaoui@enicar.ucar.tn
4: LR-OASIS, Department of Industrial Engineering, National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia Centre Génie Industriel, Université Toulouse, IMT Mines Albi, Albi, France
Email: safa.layeb@enit.utm.tn

remains relatively underexplored compared to the more common approaches that treat planning and scheduling as separate problems. Most existing works tend to focus either on surgery planning—i.e., assigning surgery dates—or on surgery scheduling, which involves sequencing and allocating patients to resources such as operating rooms. Fully integrated approaches that combine both aspects into a unified model are significantly fewer. Moreover, when the integration of both operating rooms and recovery beds is considered, the number of relevant studies becomes even more limited. This indicates a notable gap in current research efforts. In addition, to the best of our knowledge, the Red Deer Algorithm, known as RDA [13], despite its promising performance in other combinatorial optimization problems [14], has never been applied to this integrated context. This opens a promising research avenue for exploring the effectiveness of RDA in addressing the complex requirements of joint surgery planning and scheduling.

The remainder of this paper is structured as follows: Section II provides a detailed description of the integrated surgery planning and scheduling problem under this study. Section III introduces the Red Deer Algorithm (RDA) and outlines its adaptation to effectively tackle this NP-hard problem. Section IV presents and discusses the computational results obtained from benchmark instances to evaluate the performance of the proposed approach. Finally, Section V concludes the paper by summarizing the key findings and suggesting promising directions for future research.

II. Problem Description

In this paper, we address an NP-hard integrated elective surgery planning and scheduling problem that simultaneously considers both operating rooms and recovery beds. Due to its inherent complexity, this problem has been largely overlooked in previous research [4]. By incorporating both operating rooms and recovery beds, our goal is to develop a more comprehensive approach to surgical scheduling and resource allocation.

Given a set of elective surgeries (patients) to be performed and a limited number of available operating rooms and recovery beds, the challenge is to determine, within a given planning horizon: (1) the surgery date for each patient; (2) the assignment of operating rooms and recovery beds to each surgery; and (3) the sequence of surgeries in each operating room, recovery bed, and operating day. The objective is to create an optimal schedule that minimizes the maximum daily closing time of the operating theatre, essentially minimizing the makespan across all surgeries, operating rooms, and recovery beds.

In the elective surgery planning and scheduling problem under consideration, we make the following assumptions:

- A set of identical, multifunctional operating rooms is available during designated opening and closing hours.
- Human and equipment resources are readily accessible whenever required.
- The duration of each surgery encompasses both the time needed to prepare the patient in the operating room (e.g., anesthesia) and the actual surgical procedure.
- We adopt an open scheduling strategy, meaning that no restrictions based on medical specialties are imposed.

To address this complex optimization problem, we aim to develop and tailor the Red Deer Algorithm, a nature-inspired metaheuristic, to the specific requirements of integrated surgery planning and scheduling. The goal is to leverage the strengths of RDA in navigating large and complex solution spaces to produce high-quality surgical schedules that optimize resource utilization and improve overall operational efficiency. In the following section, we provide a detailed description of the Red Deer Algorithm and explain how it is adapted to effectively solve the integrated problem under consideration.

III. Proposed Red Deer Algorithm

The Red Deer Algorithm (RDA), recently introduced by Fathollahi-Fard et al. [14], is a population-based evolutionary metaheuristic inspired by the unique mating rituals of Scottish red deer during their breeding season. This natural event, known as the rut, occurs annually between late September and late November. During this period, male red deer, or stags, return to territories inhabited by female groups (hinds) and engage in intense competition with rival males. These contests involve a combination of behaviors such as loud vocalizations (roaring), side-by-side walking, and direct physical combat. A stag's reproductive success is closely linked to both its dominance behavior and physical attributes, including body size, strength, antler development, and vocal prowess. The outcome of these encounters can be severe, with some stags sustaining serious injuries or even dying. Victorious males assert dominance by expelling weaker rivals and securing exclusive mating rights with the hinds, embodying a form of natural selection.

In the RDA framework, this biological phenomenon is translated into an algorithmic structure where the initial population comprises male and female red deer candidates. The process begins with a selection phase, representing the **intensification** stage, during which male individuals compete for dominance. Based on their fitness, males are classified into two subgroups: *commanders* (the strongest individuals) and *stags* (those who are outperformed during the competition). Each commander forms a *harem*, a group of hinds (females), whose size is proportional to the commander's strength and dominance level.

Next, the **diversification phase** simulates the mating behavior. Each commander mates with a proportion (α) of hinds within his own harem and a proportion (β) from nearby harems. These mating interactions generate offspring, which serve as new candidate solutions, contributing to the formation of the next generation.

By alternating between intensification and diversification, the RDA achieves a balance between exploration (searching new areas of the solution space) and exploitation (refining promising solutions). This dynamic enables the algorithm to efficiently explore the search space and produce high-quality solutions for complex optimization problems.

Figure 1 illustrates the flowchart of the proposed RDA [14], highlighting the sequence of steps involved.

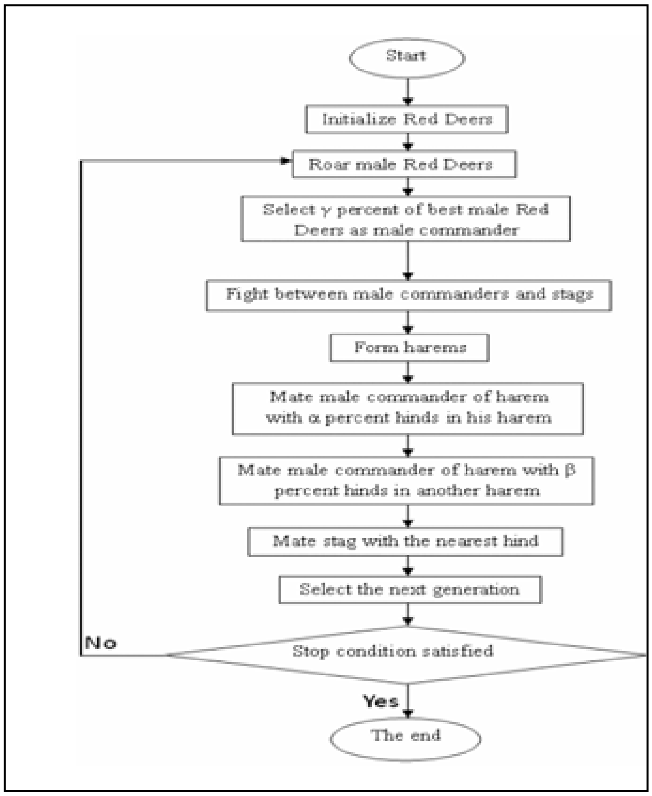

Figure 1. Flowchart of the proposed Red Deer Algorithm

Algorithm 1 also provides the pseudo-code of the proposed Red Deer Algorithm metaheuristic, which we have adapted in this study.

Algorithm 1: *Red Deer Algorithm for Elective Surgery Planning and Scheduling Problem*

1. **Input**:
   Set parameters: MaxIterations, PopSize, MaleNumber, α, β, μ
2. **Initialization**:
   Generate an initial population of PopSize random solutions (Red Deers)
   Evaluate the fitness (makespan in our study) of each solution
   Select the best solution as BestSolution
   Set Iteration = 0
3. **Main Loop**:
   While Iteration ≤ MaxIterations **do**
   For each male Red Deer in the population:
   Compute fitness (makespan)
   End For
   Form commanders and stags based on fitness

   For each commander:
   Compete with stags (fight)
   Update positions of commanders and stags
   End For

   For each commander:
   Mate with α% of hinds from his own harem
   Mate with β% of hinds from neighboring harems
   End For

   For each stag:
   Mate with randomly selected hinds from any harem
   End For

   Generate offspring and evaluate their fitness
   Select the best individuals to form the next generation
   Update BestSolution if a better one is found
   Increment Iteration: Iteration = Iteration +1
   End While
4. **Output**:
   Return BestSolution

It is important to highlight that RD Algorithm relies on six key input parameters that require tuning. These include:

- ***MaxIterations***, which defines the maximum number of iterations;
- ***PopSize***, representing the total population size of Red Deer candidates;
- ***MaleNumber*,** the number of male individuals within the population;
- ***α***, the proportion of mating occurring within each harem;
- ***β***, the proportion of mating involving hinds from other harems; and
- ***μ***, the ratio of male commanders among the male population.

A description of the main components and operational phases of our RDA algorithm is provided below.

A. Red Deer Representation

In this study, we represent each Red Deer (a solution to the surgery planning and scheduling problem, which includes operating rooms and recovery beds) using a four-row array.

- **The first row** consists of a randomly generated sequence of integers from 1 to N (number of patients), determining the order in which patients will be scheduled within the same operating room.

- **The second row** assigns each patient to a specific surgery date within a predefined planning horizon, ensuring an even distribution of procedures.
- **The third row** specifies the operating room (OR) assigned to each patient, ensuring that surgeries within the same room follow the order defined in the first row.
- **The fourth row** determines the recovery bed (RB) assigned to each patient, following a First-In, First-Out (FIFO) principle to optimize bed utilization.

For instance, let's consider a scheduling and sequencing problem with 10 surgical procedures for 10 patients. We assume that only three operating rooms and three recovery beds are available within a three-day planning horizon. A feasible solution (Red Deer) to the problem could be illustrated in Table I below.

TABLE I. AN EXAMPLE OF A RED DEER REPRESENTATION

Patients	5	1	3	9	10	8	7	6	2	4
Day	3	1	2	1	1	1	3	3	3	2
OR	2	2	1	3	3	1	2	2	2	3
RB	3	1	2	1	2	2	2	3	3	2

B. Generation of an Initial Population

The initial Red Deer population is composed of randomly generated yet feasible candidate solutions. Once the population is initialized, all individuals are evaluated and sorted in descending order based on their fitness values. The top MaleNumber candidates, representing the best-performing solutions, are designated as male red deer, while the remaining individuals form the group of female red deer, referred to as hinds.

C. Fitness Determination

Similar to genetic algorithms, the fitness function in the Red Deer Algorithm evaluates an individual (feasible solution) based on characteristics such as grace, roaring power, and fighting strength. In our case, the fitness value of a solution (Red Deer) represents the maximum closing time of the operating theatre corresponding to that solution, which is effectively the makespan value calculated across surgeries, operating rooms, recovery beds, and planning horizon.

D. Selection of Commanders Among the Male Population

The most successful males are those who attract females and form harems, earning the title of "commanders", while the less dominant males are referred to as "Stags". The number of commanders is determined by the following calculation:

$$CommandersNumber = Round\,(\mu \times MaleNumber) \qquad (1)$$

where μ is the proportion of commanders among the male population, randomly generated in the interval [0, 1]. It is preferable for μ to be greater than 0.6, as this increases the number of commanders who mate with a large number of females. The remaining males, who are less successful in attracting females, are classified as "Stags." Their number is then calculated as the complement of the commanders:

$$StagsNumber = MaleNumber - CommandersNumber \qquad (2)$$

G. Males' Fighting and Mating

Similar to the approach described in [14], the Red Deer males compete against each other based on their fitness scores. In each encounter, one male emerges as the winner, and the positions of the males are updated following a bubble sorting method. The strongest male assumes control of the harem, while the loser is expelled. The selection of hinds for mating within the harems is conducted as outlined by Fathollahi-Fard et al. [14].

H. Selection of the New Population

As shown by Yadav et al. [15], tournament selection has shown superior performance compared to rank-based selection, particularly in terms of execution speed during population generation. It is also found to be more effective than roulette wheel selection when it comes to convergence, as it typically requires fewer generations to reach optimal fitness values.

Given its efficiency in identifying high-quality solutions, we adopt the elitism selection strategy for constructing the next generation, where the best-performing individuals, based on their fitness values, are retained.

IV. NUMERICAL RESULTS

In this section, we present some preliminary results regarding the performance of the proposed RDA algorithm and the quality of solutions it generates. Our metaheuristic algorithm is implemented in Python language and ran with an Intel Core i5-8265U Processor, 1.60 GHz computer with 8GB of RAM.

A. Dataset Description

To evaluate the performance of the proposed algorithm, we used a dataset of six problem instances extracted from [16]. Each instance is characterized by the following attributes:

- Instance identifier (I1, I2, ..., I6),
- Number of patients (10, 15, 20, 25),
- Number of available operating rooms (2, 3, 4, 5),
- Number of available recovery beds (2, 3, 4),
- Planning horizon in number of operating days (3 or 5).

Both surgical and recovery durations were generated randomly using a discrete uniform distribution, as suggested in [17].

Table II summarizes the key characteristics of the six problem instances considered in this study.

TABLE II. THE INSTANCES CHARACTERISTICS

	#Patients	#Days	#Operating Rooms	#Recovery Beds
I1-10-5-2-2	10	5	2	2
I2-10-3-3-2	10	3	3	2
I3-15-3-2-2	15	3	2	2
I4-15-5-3-3	15	5	3	3
I5-20-5-4-4	20	5	4	4
I6-25-5-4-3	25	5	4	3

It is well established that proper parameter tuning significantly impacts the performance of metaheuristic algorithms. A balance must be struck between the solution quality and the required computational time. To this end, we conducted several empirical tests to identify suitable parameters for RDA. The final parameter configuration is as follows:

- **MaxIterations** = 100
- **PopSize** = 100
- **MaleNumber** = 10
- **α** = 70%
- **β** = 30%
- **μ** = randomly generated in [1%, 10%]

These settings were found to provide high-quality solutions within a reasonable computational time.

B. *Primary Results Analysis*

The performance of the proposed RDA is compared to that of the Tabu Search (TS) algorithm, using the same problem instances. Table III summarizes the results in terms of percentage deviation from the optimal makespan (PD%) and CPU time (in seconds).

The percentage deviation is calculated as:

$$PD\% = (C_{max} - C_{max}^*) / C_{max}^* \tag{3}$$

Where C_{max} refers to the maximum operating theatre closing time—or makespan—generated by either the proposed Red Deer Algorithm or the Tabu Search for a given problem instance. In contrast, C_{max}^* denotes the optimal solution derived from the Mixed-Integer Programming (MIP) model developed for this problem, as described in [17].

TABLE III. COMPUTATIONAL RESULTS

	Tabu Search		**RDA**	
	PD (%)	***CPU (s)***	***PD (%)***	***CPU (s)***
I1-10-5-2-2	1.4	0.0	0.0	3.9s
I2-10-3-3-2	1.8	0.0	0.0	3.7
I3-15-3-2-2	3.7	1.0	0.0	4.5
I4-15-5-3-3	3.9	1.5	0.2	5.1
I5-20-5-4-4	5.7	2.5	0.0	6.6
I6-25-5-4-3	7.6	5.0	1.0	12.2

The results clearly show that the Red Deer Algorithm consistently outperforms the Tabu Search in terms of solution quality, while maintaining high computational efficiency.

- ***Optimality***: RDA achieves optimal solutions (PD% = 0.0) for four out of six instances (I1, I2, I3, I5), whereas TS fails to reach optimality in any instance.
- ***Deviation from Optimality***: Even in the most complex scenario (I6), involving 25 patients and tighter resource constraints, RDA limits the deviation from optimality to just 1.0%, compared to 7.6% for TS.
- ***Average Performance***: The average percentage deviation (PD%) across all instances is only 0.2% for RDA, significantly outperforming TS, which has an average PD% of 4.0%.

From a computational times perspective, TS is slightly faster, particularly for smaller instances where it completes execution almost instantly. However, RDA also exhibits excellent computational efficiency, solving all instances in less than 13 seconds. This minor increase in CPU time is justified by the considerable improvement in solution quality, making RDA a compelling choice for real-world hospital scheduling where accuracy is crucial.

Moreover, RDA demonstrates greater scalability. As the number of patients increases—from 10 to 25—a moderate rise in CPU time is observed (from 3.7 seconds to 12.2 seconds), while the quality of solutions remains consistently high. In contrast, TS shows a growing performance gap as problem size and complexity increase, confirming the robustness of RDA in handling larger and more demanding instances.

In summary, RDA not only provides near-optimal or optimal solutions with minimal deviation but also maintains high responsiveness. These features make it a strong candidate for real-time or large-scale integrated surgery scheduling applications, especially when both operating rooms and recovery beds must be considered simultaneously.

V. Conclusion

This paper introduced a tailored Red Deer Algorithm for the integrated surgery planning and scheduling problem, considering both operating rooms and recovery beds. The RDA demonstrated excellent performance, achieving near-optimal solutions for most instances and maintaining impressive computational efficiency, with all instances solved in under 13 seconds. These results highlight the algorithm's potential for real-time applications in hospital environments.

Future research could focus on extending the current model to incorporate additional real-world constraints, such as surgeon availability, emergency surgeries, equipment limitations, and patient priority levels. Furthermore, comparing the Red Deer Algorithm with other effective metaheuristics, as well as hybridizing RDA with other metaheuristic techniques or machine learning approaches, could further enhance its performance and robustness in dynamic and uncertain hospital environments.

References

[1] Dhupar, R., Evankovich, J., Klune, J. R., Vargas, L. G., & Hughes, S. J. (2011). Delayed operating room availability significantly impacts the total hospital costs of an urgent surgical procedure. *Surgery*, *150*(2), 299-305.

[2] Guido, R., & Conforti, D. (2017). A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. *Computers & Operations Research*, *87*, 270-282.

[3] Aringhieri, R., Landa, P., Soriano, P., Tanfani, E., & Testi, A. (2015). A two level metaheuristic for the operating room scheduling and assignment problem. *Computers & Operations Research*, *54*, 21-34.

[4] Akbarzadeh, B., Moslehi, G., Reisi-Nafchi, M., & Maenhout, B. (2020). A diving heuristic for planning and scheduling surgical cases in the operating room department with nurse re-rostering. *Journal of scheduling*, *23*, 265-288.

[5] 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 on Computing*, *28*(3), 432-448.

[6] Wang, Y., Tang, J., Pan, Z., & Yan, C. (2015). Particle swarm optimization-based planning and scheduling for a laminar-flow operating room with downstream resources. *Soft Computing*, *19*, 2913-2926.

[7] Nyman, J., & Ripon, K. S. N. (2018, July). Metaheuristics for the multiobjective surgery admission planning problem. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE.

[8] Di Martinelly, C., Baptiste, P., & Maknoon, M. Y. (2014). An assessment of the integration of nurse timetable changes with operating room planning and scheduling. *International Journal of Production Research*, *52*(24), 7239-7250.

[9] Marques, I., Captivo, M. E., & Vaz Pato, M. (2015). A bicriteria heuristic for an elective surgery scheduling problem. *Health care management science*, *18*, 251-266.

[10] Riise, A., & Burke, E. K. (2011). Local search for the surgery admission planning problem. *Journal of Heuristics*, *17*, 389-414.

[11] Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. European journal of operational research, 201(3), 921-932.

[12] Rachuba, S., Reuter-Oppermann, M., & Thielen, C. (2024). Integrated planning in hospitals: a review. *OR Spectrum*, 1-54.

[13] Zitar, R. A., Abualigah, L., & Al-Dmour, N. A. (2023). Review and analysis for the Red Deer Algorithm. Journal of Ambient Intelligence and Humanized Computing, 1-11.

[14] Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2020). Red deer algorithm (RDA): a new nature-inspired meta-heuristic. *Soft computing*, *24*, 14637-14665.

[15] Yadav, S. L., &Sohal, A. (2017). Comparative study of different selection techniques in genetic algorithm. *International Journal of Engineering, Science and Mathematics,6*(3), 174-180.

[16] Tayyab, A., & Saif, U. (2022). A two-stage genetic artificial bee colony algorithm for solving integrated operating room planning and scheduling problem with capacity constraints of downstream wards. *IEEE Access*, *10*, 131109-131127.

[17] Landa, P., Aringhieri, R., Soriano, P., Tànfani, E., & Testi, A. (2016). A hybrid optimization algorithm for surgeries scheduling. *Operations Research for Health Care*, *8*, 103-114.