An Artificial Immune System Approach for a Multi-compartment Queuing Model for Improving Medical Resources and Inpatient Bed Occupancy in Pandemics

Smaranda BELCIUG[1], Silviu-Ioan BEJINARIU[3], Hariton COSTIN[2,3],
[1]*University of Craiova, 200585, Romania*
[2]*University of Medicine and Pharmacy Iași, 700115, Romania*
[3]*Institute of Computer Science, Romanian Academy, Iaşi, 700054, Romania*
hcostin@gmail.com

***Abstract*—In the context of the Covid-19 pandemic the pressure that is put on the medical systems is increasing exponentially. Healthcare systems resources are in general scarce, and hence they require policies that ensure the optimal usage of beds and utilization costs. The aim of this study is to explore how artificial immune system approaches for a multi-queuing model may aid the hospital managers improve their resources. The proposed system outlines the route of Covid-19 patients in the intensive care unit (ICU), the compartmental model proposes a reasonable composition of the ICU, considering the queuing parameters, while the artificial immune system optimizes the needed resources (beds plus associated costs). The methodology was demonstrated through a simulation based on real data collected from official sources.**

***Index Terms*—artificial intelligence, evolutionary computation, hospitals, optimization, queueing analysis.**

I. INTRODUCTION

The Covid-19 has rapidly turned into a pandemic. The human race has faced several pandemics throughout history, the greatest one, so far, being the influenza pandemic of 1918. As time passes, societies evolve, but also diseases evolve. People migrate more, independently or in masses, carrying along with them the pathogens. Through this movement, the virus continues to evolve and mutate at a fast rate, making it difficult for scientists to discover antibody drugs [1]. As the virus spreads across the globe, the intensive care units (ICUs) are starting to be overwhelmed. Hospital managers must tackle with an increase in critical bed care occupancy, usage costs, which imply also staff management. On April 27, 2020, the number of Covid-19 infected patients was 2,995,636, with 1,907,055 active cases from which around 3% (57,613 cases) were serious or critical (https://worldofmeters.info/coronavirus - accessed April, 27, 2020). Sadly enough, not all critical cases are admitted to the ICU, due to the fact that the ICU capacity is exceeded in many hospitals [2-4]. It has been shown that the median time between the onset of the symptoms and the clinical recovery is 6-8 weeks [5, 6]. A retrospective case series study involved 1591 critically ill patients from Lombardy, Italy. The patients were admitted between February 20, 2020 and March 28, 2020. On March 25, 2020, the reported median length of stay in the ICU was 9 days, with a (6, 13) 95% confidence interval. In what regards the median length of stay for patients that were discharged from the ICU, the reported number was 8 days, with a (5, 12) 95% confidence interval. As for the people who passed away in the ICU, the median length of stay was 7 days, with a (5, 11) 95% confidence interval, [7].

A hospital manager may be faced with the awful situation when she/he has to turn away patients, due to the fact that all the critical beds from the ICU are occupied. This is the case when the medical system is not sufficiently financed or is the result of a poor resource administration. On the other hand, we might be confronted with the case when we are dealing with an over-provision of hospital beds, leading to a waste of limited resources. In order for these issues to be resolved, we must find a merger between sophisticated analytical approaches and artificial intelligence (AI) techniques, and through this merger to be able to develop intelligent decision support systems that will aid hospital managers make better decision and thus improve patient care, while saving money. Different AI techniques have been deployed in this manner and reported in state-of-the-art literature. For instance, a multi-objective particle swarm optimization together with a binary search representation pattern for the bed allocation issue was proposed in [8]. Patient flow in a cost or capacity constrained medical system was modeled through a non-homogeneous discrete time Markov chain incorporating time-dependent covariates, [9]. Some works tacked with introducing queuing models, which are industry related, to the medical domain. The M/PH/c and M/PH/c/N models were used for optimizing the hospital bed allocation in a loss model, as well as on a modified model which integrated an extra waiting room [10, 11]. Two approaches using evolutionary computation and queueing models mix were developed in [12, 13].

Different from the above-mentioned papers, that use as optimization method genetic algorithms for optimizing the number of beds and costs for the Department of Geriatric Medicine – St. George Hospital, London, in this study we propose for the optimization of the Covid-19 pandemic bed allocation and cost management, an artificial immune system.

The aim of this paper is to propose an intelligent decision support system based on a merger between an artificial

Digital Object Identifier 10.4316/AECE.2020.03003

immune system and a multi-compartmental queuing model. The scope of the study is two-fold: first to propose a bed allocation policy, and secondly to simulate financial resource utilization. The patient flow is modeled using queueing theory, having a Poisson process describing the arrival of patients, the beds in the ICU are thought as servers, and the duration of stay is shaped using a phase-type distribution. The artificial immune system is used for the optimization of the bed allocation policy as well as for the cost assessment.

The paper is organized in 6 sections. A brief review of the nature-inspired metaheuristics is presented in the second section. Section 3 summarizes the design of the intelligent decision support system, Section 4 presents the experimental results, whereas Section 5 deals with the corresponding discussion. Section 6 presents the conclusions.

II. Nature-inspired Metaheuristics

In the last decades, a great number of nature-inspired (NI) optimization metaheuristics were developed and used in artificial systems. They model the intelligence of nature and are use it to solve a huge diversity of classification and optimization problems. It's true that there is no algorithm that can be considered the best for all types of problems. However, some algorithms have superior performances in terms of solution accuracy and convergence speed. Some of these will be briefly described in the following paragraphs. In literature, the NI algorithms are classified according to several criteria [14] such as: trajectory of the search path (trajectory based and population based), interaction of the agents (attraction and non-attraction based), update method (rule and equation based). The most common classification is based on the source of inspiration according to which the NI algorithms can be classified into: evolutionary algorithms; swarm intelligence based bio-inspired algorithms; non-swarm bio-inspired; physics and chemistry phenomena based.

Let consider the case of an optimization problem (OP) in the continuous space. The OP general form is $\min_{x \in S} f(x)$ where f is the *objective function* to be minimized, $S \subset R^d$ is the *domain* of the problem which can include also a list of restrictions and d is the *dimension* of OP given by its number of parameters. In case of population based NI algorithms, a set of agents $X = \{x_i, i = 1, \ldots N\}$ representing the *possible solutions* of the OP are described by d – dimensional vectors $x_i = \left(x_i^1, \ldots x_i^d\right) \in S, i = 1, \ldots N$. The values of the OP parameters are encoded in the possible solutions which are initialized with random values in the problem's domain S before starting the evolutionary process. Depending on the modeled NI process, they represent the position or the status of the agents. During the evolution the vectors x_i change accordingly to the specific *strategy* used in the algorithm. The function to optimize f is modeled by the *fitness function* which is evaluated for all positions reached by x_i. The evolution finishes when the *stop condition* (either the number of iterations or another condition related to the problem's solution) is met. The position in which an agent reached the best value of the fitness function is the *solution* of the problem.

A. Evolutionary algorithms

This category includes algorithms inspired by the biological evolution which is modeled at a reduced scale. Some algorithms in this category are: Genetic algorithm, Differential evolution, and Artificial immune system.

One of the oldest and very efficient in this category is the Genetic algorithm (GA), proposed by Holland in 1975 [15]. It is inspired by the population genetics, and the possible solutions are encoded as chromosome strings. However, GA versions using real encoding have been developed with equally good results. In each generation chromosomes with best fitness values generate offspring that replace chromosomes with lowest values of the fitness function. The *genetic operators* used for new generation creation are: selection, crossover and mutation which is applied with a probability distribution.

B. Bio-inspired algorithms

The bio-inspired algorithms are inspired from biological systems and they model the behavior of some species to find food, to avoid dangers or to perpetuate their species. Depending on how the agents interact and generate a higher level of intelligence than that of each individual, the bio-inspired algorithms are classified as swarm and non-swarm algorithms.

1) Swarm intelligence

The Swarm intelligence category includes: Particle swarm, Multi swarm, Ant colony, Artificial bee colony, Firefly, Bees swarm, Wolf search, Bacterial foraging and many other algorithms.

The **Particle Swarm Optimization** (PSO) is inspired by the bird or fish swarming intelligence and it one of the oldest and most efficient NI algorithm which was proposed by Kennedy and Eberhart in 1995 [16]. During the evolution, the agents (particles) move in directions which are mainly random, but the personal and group's experience is used to direct the particles toward better positions reached before. Let $x_i(t)$ be the position of the i^{th} particle in the t^{th} iteration. The new position is computed as:

$$x_i(t+1) = x_i(t) + w \times s_i(t) + \left(c_1 \times r_1 \times \left(x_i^b - x_i(t)\right)\right) + \left(c_2 \times r_2 \times \left(x^b - x_i(t)\right)\right) \quad (1)$$

w is the movement inertia of all particles, c_1 and c_2 are the personal and social learning coefficients, x_i^b is the best reached position of the i^{th} particle, x_b is the best position reached by any particle and r_1, r_2 are random values.

The **Bacterial Foraging Optimization** (BFO) algorithm proposed by Passino in 2002 [17] model the social behavior of E-coli bacterium for finding nutrients and avoid noxious substances. The goal of bacteria is to maximize the accumulated energy per time unit. During evolution bacteria multiply if their strategy for finding nutrients is good, or die if the accumulated energy is low. The evolution is described by four nested loops: chemotaxis, swarming, reproduction and elimination-dispersal. In the *chemotactic* step, the bacterium performs two types of movement: a tumble in a random direction and then a number of swims while the fitness evaluation in the new position is better. The social behavior is modeled in the *swarming* strategy – the cells can attract or repel each other depending on their status (accumulated energy). The bacteria which accumulated a greater quantity of energy divide into two new bacteria in the *reproduction* step. The new bacteria are created in the same position but they will move in different random directions. The other bacteria, with lower accumulated energy die. To avoid the convergence to local solutions, in the *elimination-dispersal* step, a number of randomly chosen bacteria are eliminated from colony and replaced by new bacteria created in random positions. As in all the other NI algorithms, the solution is given by the position in which a bacterium reached its best healthy status.

2) Non swarm algorithms.

The algorithms included in this category are inspired by biological processes but, in contrast to those in the previous category, the swarming behavior of the agents is not used. Some examples are: Dolphin echolocation, Flower pollination, Fish school search, Cuckoo search.

The **Flower Pollination algorithm** (FPA) is inspired by the reproduction process of flowers through pollination and was proposed by Yang in 2012 [18]. The evolution model is a little more complex in case of FPA. First, the pollination can be done between the flowers of the same plant (*self-pollination*) or between the flowers of different plants (*cross-pollination*). All these strategies are implemented in the model described in [18, 19] by the following four rules:

- Cross-pollination and biotic pollination are considered as *global* pollination, in which pollination vectors move following a Levy distribution,

$$x_i^{t+1} = x_i^t + L\left(\overline{x} - x_i^t\right) \tag{2}$$

where x_i^t, x_i^{t+1} are the positions of the i^{th} agent in the t and $t+1$ iteration, $\overline{x}$ is the global best position and $\boldsymbol{L}$ is the step length computed by the Levy distribution [18],

- Self-pollination and abiotic pollination are considered as "local" pollination:

$$x_i^{t+1} = x_i^t + \varepsilon\left(x_j^t - x_k^t\right) \tag{3}$$

where ε is a random value with uniform distribution and x_j, x_k, $j \neq k$ are two randomly chosen agents.

Developed initially for global optimization in the continuous space, FPA has been shown to be even more efficient [18] than PSO and GA algorithms at least in case of some optimization problems.

Cuckoo Search (CSA) is also one of the most performing NI algorithms, with similar or better performances than those of PSO. It was proposed in 2009 by Yang and Deb and it is inspired by the parasitic behavior of the cuckoo that lays eggs in foreign nests [19, 20]. The possible solutions are encoded as the nests positions - the agents in this case. The movement strategy used in CSA consists of a random Levy flight walk performed by a randomly selected cuckoo:

$$x_i\left(t+1\right) = x_i\left(t\right) + \alpha \oplus Levy\left(\lambda\right) \tag{4}$$

where α is the step size, $\oplus$ is the entrywise multiplication and Levy(λ) is the random step length drawn from a Levy distribution [19]. Then the fitness function is evaluated in the new position and it is compared to the fitness computed in a randomly selected nest. If it is better, the selected nest is replaced by a nest in the new position. In the discovery step, a percent of nests are abandoned and replaced by nests placed in randomly chosen positions. CSA uses long steps which avoid the algorithm to be trapped in local solutions and makes it more efficient than other NI algorithms.

C. Physics / chemistry phenomena based

This category includes algorithms as: Black hole, Gravitational search, Harmony search, Intelligent water drop, Water cycle or Fireworks, which are inspired by other natural or artificial phenomena which does not shape the behavior of living beings.

The **Black Hole Algorithm** (BHA) was proposed in 2013 [21] as a heuristic method for solving data clustering problems. It is inspired by the black hole attraction phenomenon. The star (agent) with the best value of the fitness is considered to be the black hole (BH) which attracts all the other stars. The movement equation is

$$x_i\left(t+1\right) = x_i\left(t\right) + r \times \left(x_{BH} - x_i\left(t\right)\right), i = 1,...N \tag{5}$$

where x_{BH} is the position of the black hole and r is a random value. In fact, the black hole does not change its position and it attracts the other stars until another star became black hole because it reaches a better position. Considering the fact that x_{BH} is in fact the global best position, BHA is considered by some authors [22] to be a simplified version of PSO. In contrast to PSO where the particles continue to swarm in the neighborhood of the best position, in BHA, when a star is too close to the black hole, it is absorbed and replaced by another star created in a random position which increases the chances to discover a better solution.

Another interesting algorithm is **Fireworks** (FWA), proposed by Tan in 2010 [23] and improved in the next years [24]. It is inspired by the explosion of the fireworks and, what makes it different from many other NI algorithms is that a selection strategy is involved in order to keep constant the number of agents (fireworks). During evolution each firework creates a number of descendants (sparks) by explosion. Their number and position depend on some dynamic parameters which are computed every iteration (epoch) depending on the firework's quality: *Explosion strength* – the number of sparks, *Explosion amplitude* – to avoid the convergence to local solutions and displacement. Also, in FWA two types of sparks are created: regular and Gaussian. The new spark's position is computed by altering a single randomly chosen coordinate of the parent firework. The next generation of fireworks is selected and it includes the best, the worst and a number of randomly chosen sparks to keep the population size constant.

Apart from single objective optimization in the continuous space, a great number of versions were developed to solve real optimization problems. A special case is that of the multi-objective optimization (MOO):

$$\arg\min_{x \in S}\left[f_1(x),...f_n\left(x\right)\right], n > 1 \tag{6}$$

Because the objectives cannot be minimized simultaneously, the most convenient solution is to transform the OP in single objective optimization by using convenient weights for each objective. The correct approach is to approximate a set of Pareto optimal solutions (*Pareto front*), from which the most convenient is chosen depending on the specific problem to be solved. This is based on the Pareto dominance relation: a solution x_1 dominates another solution x_2 if the value of all objectives in x_1 are not worse than those in x_2 and for at least one objective the value in x_1 is better than that in x_2. The Pareto front contains all the *strong Pareto optimal* solutions $x^* \in S$ which are not dominated by any other solution:

$$\sim \exists x' \in S \ni \left[f_i(x') \le f_i(x^*) \forall i, 0 < i < n AND \exists i \ni f_i(x') < f_i(x^*) \right] \tag{7}$$

D. Applications

The NI algorithms capabilities were demonstrated for wide range optimization problems in many areas. Some examples are presented below.

The Quantum-inspired Genetic Algorithm is applied in [25] to find the sets of optimal parameters for the wind disturbance alleviation in Flight Control Systems. The Polar Bear optimization is used for intensification of the district heating plant to work with maximum efficiency at the lowest costs in [26]. The efficiency of seven NI algorithms, including PSO, CSA, FPA and GA was evaluated in [27] for the pharmaceutical tableting process optimization. In [28] is presented an Enhanced Oil Recovery technique where NI algorithms and reservoir simulation were used for the design of polymer flooding process in terms of optimal concentration, slug size and initiation time. A PSO based simulation of the rescue process in case of forest fire is proposed in [29]. A comparison of several NI algorithms performances is presented in [30]. As optimization problem, the video motion estimation is considered in the context of streamlining video compression. In [31] PSO is used for handling partial object occlusions in 3D objects tracking procedure. A medical application proposed in [32] uses the Spider monkey optimization algorithm for the parameters tuning in a gradient boosting machines classifier for exudate classification on fundus retinal images. NI algorithms are often used for images registration, fusion, enhancement or segmentation [33]. The resource constrained project scheduling problem is addressed using the Bee algorithm in [34], FPA and PSO in [35] and discrete CSA in [36].

III. Materials and Methods

A. Queuing Model

Previous works have introduced M/PH/c and M/PH/c/N queuing models in order to optimize the use of hospital resources. They enable both the estimation of two queuing parameters, i.e. the probability of lost demands and the mean number of patients in hospital, and the balance between the costs of empty beds against the cost of turning patients away, through a cost function. A simple approach, based on the direct estimation of several values regarding the rejection probability and the associated costs, computed for different parameters settings, has been used for the optimization purpose. To overcome these computation constraints, the current work proposes a flexible strategy for the optimization of hospital bed occupancy, and associated costs. Different from the previous approach, and taking into account the diverse nature of the involved parameters (both discrete and continuous), the proposed model uses artificial immune systems (AIS) in order to optimize both the bed allocation and the associated costs. As before, the mathematical model for the patient flow was developed by combining elements of queuing theory with results from compartmental models in conjunction with phase-type distributions. An associated cost model is set up in order to balance the costs associated with refusing patients and with maintaining the necessary number of beds, using a base-stock policy approach. The AIS based-approach enabled the development of an additional "What-if" analysis, by means of which different scenarios concerning various possible options available for the hospital management were explored. The novelty of this paper consists in providing the medical staff a combined OR/AIS model which can be used to simulate different possible situations, and thus enabling the choice of an optimum decision. This approach coherently and synergistically integrates queuing models, compartmental models, and AIS aiming to support the optimization of the beds and associated costs.

In this study we have used a M/PH/c queuing model, having M as the Poisson arrivals, a phase-type service distribution, and c as the number of servers (beds) [37]. In the case of the COVID-19 pandemic, no queue is possible, hence not allowed. If all the critical beds are occupied, the patients that arrive at that moment in time cannot enter the hospital. We denote λ as the Poisson arrival rate. The probability density function for the phase-type service is:

$$f(t) = \sum_{i=1}^{l} \alpha_i \rho_i e^{-\alpha_i t} \tag{8}$$

together with the corresponding mean $\tau = \sum_{i=1}^{l} \rho_i / \alpha_i$. We have denoted l as the number of phases, α_i as the mixing proportions, and the transition rates are ρ_i, having $\sum_{i=1}^{l} \rho_i = 1$.

Having a given time interval t, the average number of arrivals during t is $\lambda \cdot t$. The average number of arrivals, a, during an average period of length of stay τ is $a = \lambda \cdot \tau$. The likelihood to have j occupied beds is calculated by:

$$P_j = \frac{a_j / j!}{\sum_{k=0}^{c} a^k / k!} \tag{9}$$

The probability of having all c beds occupied is:

$$P_c = B(c,a) = \frac{a^c / c!}{\sum_{k=0}^{c} a^k / k!} \tag{10}$$

The above formula is the Erlang's loss formula, which gives the proportion of patients who cannot be treated by the hospital [38]. We can state this fact if and only if the system is in steady state.

One of the aims of our study was to use an artificial immune system to estimate the above mentioned parameters, c, λ and τ, so that we can determine the appropriate fraction of refused patients, so the system does not crash, $B(c,a)$, the average duration spent in the hospital, $W = \tau \cdot [1 - B(c,a)]$, as well as the average number of patients in the ICU, $L = a \cdot [1 - B(c,a)]$.

Another aim was to estimate best medical care at the minimum possible cost. Technically, we need to keep at a minimum level with minimum costs the loss of potential patients. The search is for the optimal values that can provide a compromise between treating costs and penalty costs (cost of refusing a patient). Thus, we introduce two new parameters: the hospitalization cost h per day per unoccupied bed, and the penalty cost π per turned away patients. Technically, we need to optimize the following cost function:

$$g(c,\lambda,\tau,h,\pi) = \pi \cdot \lambda \cdot B(c,\lambda \cdot \tau) + h \cdot \left\{ c - \lambda \cdot \tau \cdot \left[1 - b(c,\lambda \cdot \tau) \right] \right\} \quad (11)$$

The proposed model is as follows: Covid-19 infected patients are admitted to sub-ICU care at first. Some of them will get better with proper treatment, and will be discharged, others will die before getting to the ICU. Another group of patients will be admitted to the ICU. From here the patients may recover or die there. Figure 1 presents the two-compartment model.

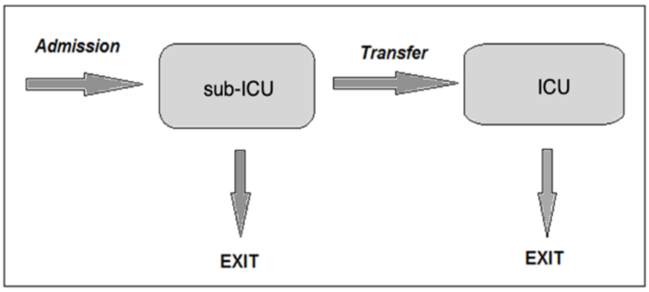

Figure 1. Two-compartment structure for Covid-19 patient treatment

The service time distribution can be discrete time or continuous time stochastic [37].

B. Artificial Immune Systems Optimization

In order to optimize the Covid-19 patient management, we have used artificial immune systems (AIS) to estimate *B(c, a*), and also to minimize the cost function *g(c)*. AIS are inspired by the mechanism of the biological immune system. AIS are divided into three subcategories: clonal selection, negative selection, and immune network algorithms. In this paper, we have used the clonal selection algorithm. Burnet introduced the clonal selection algorithm through his works published in 1957, and 1959 respectively, [39, 40].

The AIS starts with a population of general immune cells which change themselves in response to a certain antigen. The clonal strategy encompasses a population of antibodies, each antibody representing a potential solution to an optimization problem. The antibodies are subjected to selection process that depends on their affinity towards the antigen (problem). After the best antibodies are selected, they undergo duplication and variation processes, which improve their adaptive affinity to the antigen.

So, AIS are bio-inspired algorithms and have the following elements: population of antigens, population of clones, selection process according to their affinity towards the antigen, cloning process of the best antibodies, random mutation of the clones, and finally a replenishing process in order to get out of local optima, [41-43]. Below it is shown the AIS used throughout this study.

AIS algorithm

1. A population of antigens of size n is randomly generated from a certain interval.
2. The best antibodies, or candidate solutions, are chosen based on their affinity towards a certain antigen, or in our case an evaluation of a cost function.
3. The above selected antibodies are cloned proportional to their affinity (e.g. the best antibody has the most clones, etc.)
4. The clones are subjected to hypermutation process, which is inverse proportional to their affinity.
5. The resultant clones, together with the initial antibodies compete once again through a selection process in order to remain in the next generation population.
6. The low-affinity population members are replaced through randomly generated antibodies.
7. The whole cycle is repeated until the termination criterion is reached.

The convergence speed is in concordance with the population size, n. In order to determine the appropriate size, we have meta-heuristically tested population sizes ranging from 50 to 200 antibodies, choosing in the end n = 150 antibodies, number which gave the best performance. In what regards the number of generations, we have tested from 50 generations to 200 generations. The best performance was achieved for a population size of 150 antibodies and 100 generations.

Regarding parameter tuning, we have used the following values: the cloning rate was set taking directly proportional to the antibodies' affinity, whereas for the hypermutation process we have used the non-uniform (normally distributed) mutation, with the mutation percentage of 0.35.

In our approach, an antibody is represented through a vector (c,λ,τ) with each antibody cell belonging to a certain domain that matches real Covid-19 data. The affinity measure is given by *B(c, a)*. In order to resolve the second aim of our study, that is to find the balance between the patient admission and costs for refusing him, we had to minimize the cost function g, using an antibody having the form (c,λ,τ,h,π).

The AIS has been implemented in Python and run on 2.7 GHz Intel Core i5, 8 GB (RAM). The model can be used by healthcare professionals simulate hospital bed allocation and other cost, using different arrival policies, average lengths of stay, bed occupancy, costs, etc.

C. ICU Covid-19 Patients in Italy and Lombardy Data

For the inventory policy we have used the official data made available by the Italian Health Ministry on a daily basis at 6 PM Central Europe Time (https://pselab.chem.polimi.it/bollettino-pandemia-covid-19/ - accessed April 26, 2020) and the European Society of Anesthesiology (https://www.esahq.org/esa-news/dynamics-of-icu-patients-and-deaths-in-italy-and-lombardy-due-to-covid-19-analysis-updated-to-30-march-day-38-evening/ - accessed April 26, 2020). The data time span is from February 22, 2020 to March 31, 2020. The ICU departments had allocated, on average, 650 beds before the onset of Covid-19 that is February 22, 2020. The number of ICU beds increased to 1328, till March 30, 2020. The inpatient management was the two-compartment model (sub-ICU and ICU). The two-compartment model shown in Fig. 1 matches the reality.

For the mean arrival rate of Covid-19 patients in the ICU department, based on the official data, we have divided the time frame into two parts: for March 13 – March 23, 2020, the mean arrival rate equals λ = 200 patients per day, and for March 23 – March 30, 2020, the mean arrival rate equals λ = 63 patients per day. The mean length of stay equaled τ = 8 days, the maximum days spent in the ICU being 15 days.

Considering the general procedure in the health management before the Covid-19 pandemic, we have estimated the cost parameters as it follows [44]. Please take note that these are just suppositions, since we did not have access to the real data.

- the total cost per patient, comprising the bed (350 EUR), treatment and personnel costs (1675 EUR), is 2025 EUR;
- the hospitalization cost is h = 350 EUR per day;
- we have computed the penalty cost as such: the total cost of dispatching a patient multiplied by the average length of stay in the ICU department. We have also considered that the penalty cost is around 30% of the total cost of turning away a patient, giving us the following estimation of π = 1920 EUR for the first time frame (number of beds = 800), and π = 3120 EUR for the first second frame (number of beds = 1300).

IV. Results

In this section, we will present the results that we have obtained using the AIS optimization. A hospital manager can optimize the available beds against the cost of turning away patients. This is imperative, since in every country there are outbreaks of Covid-19 infections, and also places where the number of infected persons is low, thus not needing extra ICU bed acquisition.

ICU Bed Occupancy Optimization in the Context of the Covid-19 Pandemic

The experimental results presented in this subsection are obtained by means of the AIS optimization so that to find the ideal values for the c, λ andτ, i.e. the probability of lost demands $B(c, a)$ is kept at a reasonable level. Since there is no consensus regarding this level, we have considered a threshold of 10% as the maximum rejection reasonable level, implying an ICU bed occupancy ranging from 88 to 95%.

The value domains for the c, λ and τ for the first time frame, were chosen as it follows:

- number of allocated beds $c \in [650, 800]$;
- arrival rate in the ICU $\lambda \in [180, 240]$;
- length of stay $\tau \in [6, 12]$.

The results using the AIS approach are shown in Table I.

We can see from Table I that the AIS approach has computed the minimum rejection probability, 3.4% of patients being refused, and this can be achieved with 798 beds, an arrival rate around 180.2 patients/day, and an average length of stay in the ICU of 6 days. We can see that none of the simulated results was able to achieve a reasonable rejection level. Please take note that this is only an overall simulation, since the dynamics of Covid-19 patients is large, and the simulation regards the highest peak of the outbreak. The number of beds and arrival rate comprises the data from all the hospitals in a supposed country, taking as example Italy. So, each hospital manager can optimize the number of beds and admission rate, according to the given situation, and taking into account whether the hospital is placed in an outbreak region or not.

The value domains for the c, λ and τ for the second time frame were chosen as it follows:

- number of allocated beds $c \in [800, 1300]$;
- arrival rate in the ICU $\lambda \in [50, 75]$;
- length of stay $\tau \in [6, 12]$.

The results using the AIS approach are shown in Table II.

We can see from Table II that the AIS approach has computed the minimum rejection probability, 1.7% of patients being refused can be obtained with 1275 beds, an incoming rate around 52 patients/day, and an average length of stay in the ICU of 6.3 days. We can see, that the experimental results have improved, partly due to the fact that the number of beds increased substantially, and partly because the number of admissions dropped by 2/3. Again, this is an overall simulation. It would be interesting to see the situation per hospital and per region.

TABLE I. B(C,A) VALUES FOR DIFFERENT QUEUING MODEL PARAMETERS

c	*λ*	*τ*	*B(%)*
798	180.2	6	3,4
789	182.6	7.07	3,8
761	185.8	6.01	4,3
750	183.1	6.21	5
723	186.9	6.15	5,9
745	187.2	6.34	6,0

TABLE II. B(C,A) VALUES FOR DIFFERENT QUEUING MODEL PARAMETERS

c	*λ*	*τ*	*B(%)*
1275	52	6.3	1,7
1220	53,2	6,1	2
1218	52,1	6.8	3,1
1200	53,3	6.21	4,8
1212	52.8	6.18	5,2
1199	53,2	6.23	5,7

TABLE III. THE VALUES OF THE COST FUNCTION **g** FOR DIFFERENT MODEL'S PARAMETERS FOR THE FIRST TIME FRAME

c	*λ*	*τ*	*h*	*π*	*g*
799	180	6	350	1900	5969170
795	180.51	6.01	350	1810	6081966
700	181.2	6.03	350	1835	6196883
765	180.3	6.1	350	1890	7572129
650	240	8	350	1800	12081100

TABLE IV. THE VALUES OF THE COST FUNCTION **g** FOR DIFFERENT MODEL'S PARAMETERS FOR THE SECOND TIME FRAME

c	*λ*	*τ*	*h*	*π*	*g*
1275	52	6	350	2800	6862520
1328	50	6	350	3200	8633540
1300	52.1	6	350	3000	8765351
1250	56	6.2	350	3000	16576364
1265	56	6.1	350	2900	82796116

Regarding to the AIS optimization of the healthcare costs, in Tables III and IV, we see the experimental results given by the algorithm for the estimation of the c, λ, τ, h, and π parameters, in order to obtain an optimal balance between the cost function g and the equitable proportion of patients that are turned away. We have used in the simulation the value of the penalty cost which ranged from 1800 EUR to 1920 EUR, for the first time frame, and 2500 EUR to 3120 EUR for the second time frame.

The AIS approach has shown that the cost function is strongly dependent on the number of beds, arrival rate and average time spent in the ICU. The best case hypothetical scenario was found for 799 ICU beds, arrival rate of 180 patients per day, and average time spent in the ICU of 6 days, for the penalty cost of 1900 EUR.

Table IV depicts the experimental results for the second time frame, in terms of the cost function.

The AIS approach has shown that the cost function is strongly dependent on the number of beds, arrival rate and average time spent in the ICU. The best case hypothetical scenario was found for 1275 ICU beds, arrival rate of 52 patients per day, and average time spent in the ICU of 6 days, for the penalty cost of 2800 EUR. Juggling with real data from each hospital, the managers really can optimize the stock policy for the ICU departments.

In both time frames, we need to keep in mind that the arrival rate is obviously characterized by stochasticity, and it takes into account the specific outbreak of each zone. The delay probability as well as the cost is sensitive to the number of beds and arrival rate, and less sensitive to the length of stay in the ICU unit, mainly because the real data shows that no matter the outcome (survival or death), the average length of stay in the ICU for a Covid-19 patient is slightly the same.

Another conclusion that can be drawn is that the huge difference between the holding and the penalty costs determines the weight of the value of the rejection rate to exceed the weight of the number of unoccupied beds.

Regarding these findings, a manager has to consider the arrival rate of patients in her/his geographical area, thus keeping the unoccupied beds at a minimal rate. Please take note, that the costs are hypothetical, since there is no real data available on this matter.

V. Discussion

This present study aimed fulfilling two goals. The first one was to find a merger between a standard queuing model of the form M/PH/c for ICU bed occupancy, and an AIS algorithm. Thus, the hospital managers may assess the system's parameters, in such a way that a reasonable rejection level of Covid-19 patients is obtained, taking into account the corresponding ICU admission rate and number of beds, as well as the average time spent in the ICU.

The second aim was to use again the AIS algorithm to optimize the resource usage in the ICU. For this, we have considered a hypothetical base-stock policy that is often used in inventory systems for expensive and slow-moving items. Taking into account the nature of the problem, the base-stock policy seemed suitable to be used. Through this, the hospital manager can juggle with the number of ICU beds, admission rate, average time spent in the ICU, as well as the penalty and holding costs, and thus make the hospital more cost effective.

Please keep in mind that this study in only indicative, since the actual costs depend on each hospital in particular.

After analyzing the data, we can state that the merger between an artificial intelligence algorithm and a queuing model is fruitful, and it might truly come in handy to every hospital manager, mainly because:

- it can encode in the antibody the whole information that is provided by the cost model, as well as the queuing model;
- the methodology behind the AIS is easy to comprehend, design, and use;
- this new method can be useful for different similar situations.

Other approaches in the literature use genetic algorithms (GA) instead of AIS [45], or Ant Colony Optimization (ACO) [46]. It is possible that GA be implemented with exact the same data used for our simulation. Then a statistical comparison should be made between the two results sets obtained. But this would be the goal of a new study and a research paper.

VI. Conclusion

Queuing models have been used in healthcare systems in order to optimize patient management. In recent years, researchers all over the world have started to merge different artificial intelligence techniques with analytical models, due to their efficiency in optimization problems. These methods are easy-to-understand, and easy-to-use. In this study we have seen that this approach might be used in improving patient management and healthcare cost in the Covid-19 pandemic.

We were interested in finding out whether the results obtained by the current approach are similar to the results obtained by a different optimization technique. Hence, instead of using artificial immune systems, we have applied a genetic algorithm to optimize Erlang's loss formula and found that the fine tuning of the hyperparameters are of huge importance in the optimization process. Since, both algorithms are meta-heuristics, the choice of the values for the population size, clone population, mutation probability, recombination probability etc., has an impact on the outcome. Hence, for the genetic algorithms optimization, if we were to use 150 as population size, the Wrights' heuristic crossover operator, uniform mutation, 0.35 recombination probability, and 0.4 mutation probability, the obtained results would be similar to the artificial immune system approach. It is easy to comprehend that in order to see which model performs better given a real situation, more parameter choices must be explored. This is an interesting direction which we are going to investigate, as more relevant data on Covid-19 becomes available.

Future work should explore an extended queuing system of M/PH/c/N, in which the hospital has a fixed maximum capacity of $N > c$ beds, thus reducing, or even in some cases avoiding all together the patients' denial, when all the ICU beds are occupied. A possible alternative of a sub-ICU unit with (N-c) beds might be a good solution. Also, other optimization methods should be used, leading in the end to a committee of such methods, which work together in finding the optimal solution.

Acknowledgment

The authors would like to thank the European Society of Anaesthesiology and the Italian Health Ministry for providing the data.

REFERENCES

[1] S. Belciug, F. Gorunescu, "How can intelligent decision support systems help the medical research", in S. Belciug, F. Gorunescu: "Intelligent Decision Support Systems – A Journey to Smarter Healthcare", Springer, pp. 71-98, 2020.

[2] J. Phua et al., "Intensive care management of coronavirus disease 2019 (COVID-19): challenges and recommendations", Lancet Respir Med, doi: 10.1016/S2213-2600(20)30161-2.

[3] A. Remuzzi, G. Remuzzi, "COVID-19 and Italy: what next", Lancet, doi: 10.1016/S0140-6736(20)30627-9, 2020.

[4] D. Wang, B. Hu, et al. "Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in Wuhan,China",JAMA,doi:10.1001/jama.2020.1585, 2020.

[5] WHO – China Joint Mission, "Report of the WHO – Chine Joint Mission on Coronavirus Disease (COVID-19)." https://www.who.int/docs/default-source/coronaviruse/who-china-joint-mission-on-covid-19-final-report.pdf (accessed April 27, 2020)

[6] F. Zhou, T. Yu, R. Du, et al., "Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study", Lancer, 1054-1062, 2020

[7] G. Grasselli, A. Zangrillp, A. Zanella, et al., "Baseline characteristics and outcomes of 1591 patients infected with SARS-CoV-2 admitted to ICUs of Lombardy Regiom, Italy", JAMA, doi:10.1001/jama.2020/5394, 2020.

[8] Y-J. Gong, J. Zhang, Z. Fan, "A multi-objective comprehensive learning particle swarm optimization with a binary search-based representation scheme for bed allocation problem in general hospital", Proc IEEE International conference on systems, man, cybernetics, Istanbul, Turkey, 10-13 October, 1083-1088, 2020.

[9] L. Garg, S. McClean, B. Meenan, P. Millard, "A non-homogeneous discrete time Markov model for admission scheduling and resource planning in a cost or capacity constrained healthcare system", Health Care Manage Sci, 13 (2), 155-169, 2010.

[10] F. Gorunescu, S.I. McClean, P.H. Millard, "A queueing model for bed-occupancy management and planning of hospitals", J Oper Res Soc, 53 (1), 19-24, 2002.

[11] F. Gorunescu, S.I. McClean, P.H Millard, "Using a queueing model to help plan bed allocation in a department of geriatric medicine", Health Care Manage Sci, 5, 307-312, 2002.

[12] S. Belciug, F. Gorunescu, "Improving hospital bed occupancy and resource utilization through queueing modeling and evolutionary computation", J Biomed Inf, 53, 261-269, 2014.

[13] S. Belciug, F. Gorunescu, "A hybrid genetic algorithm-queueing multi-compartment model for optimizing inpatient bed occupancy and associated cost", Art Int in Med, 68, 59-69, 2016.

[14] I. Fister Jr., X-S. Yang, I. Fister, J. Brest, D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization", in ELEKTROTEHNIˇSKI VESTNIK 80(3): 1–7, 2013

[15] J.H. Holland, "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence". U Michigan Press, 1975.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization", in Proceedings of the IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948, 1995, doi: 10.1109/ICNN.1995.488968 .

[17] Y. Liu, K. M. Passino, "Biomimicry of Social Foraging Bacteria for Distributed Optimization: Models, Principles, and Emergent Behaviors", in Journal of Optimization Theory and Applications, 115(3), pp. 603–628, 2002, doi: 10.1023/A:1021207331209 .

[18] X.-S. Yang, "Flower Pollination Algorithm for Global Optimization", in J.Durand-Lose, N. Jonoska (eds), "Unconventional Computation and Natural Computation", UCNC 2012. Lecture Notes in Computer Science, 7445, 2012, doi: 10.1007/978-3-642-32894-7_27.

[19] X.-S. Yang, Nature-Inspired Optimization Algorithms, Elsevier, 2014, doi:10.1016/C2013-0-01368-0

[20] X.-S. Yang; S. Deb, "Cuckoo search via Lévy flights". World Congress on Nature & Biologically Inspired Computing (NaBIC 2009). IEEE Publications. 210–214, 2009, doi: 10.1109/NABIC.2009.5393690

[21] A. Hatamlou, "Black hole: a new heuristic optimization approach for data clustering", in Information Sciences, 222, 2013, pp. 175–184, doi: 10.1016/j.ins.2012.08.023.

[22] A.P. Piotrowski, J.J. Napiorkowski, and P.M. Rowinski, "How novel is the "novel" black hole optimization approach?", in Information Sciences, 267, Elsevier, pp. 191–200, 2014, doi: 10.1016/j.ins.2014.01.026.

[23] Y. Tan, Y. Zhu, "Fireworks algorithm for optimization", in Tan Y., Shi Y., and Tan K.C. (eds.), ICSI 2010, Part I, LNCS 6145, pp. 355–364, 2010.

[24] Y. Tan, Fireworks Algorithm. A Novel Swarm Intelligence Optimization Method, Springer-Verlag, 2015.

[25] Q. Bian, B. Nener, X. Wang, "A quantum inspired genetic algorithm for multimodal optimization of wind disturbance alleviation flight control system", in Chinese Journal of Aeronautics, 32(11), 2480-2488, 2019, doi: 10.1016/j.cja.2019.04.013.

[26] M. Woźniak, K. Książek, J. Marciniec, D. Połap, "Heat production optimization using bio-inspired algorithms", in Engineering Applications of Artificial Intelligence, 76, 185-201, 2018, doi: 10.1016/j.engappai.2018.09.003.

[27] H.M. Zawbaa, S. Schiano, L. Perez-Gandarillas, C. Grosan, A. Michrafy, C.-Y. Wu, "Computational intelligence modelling of pharmaceutical tabletting processes using bio-inspired optimization algorithms", in Advanced Powder Technology, 29(12), 2966-2977, 2018, doi: 10.1016/j.apt.2018.11.008.

[28] D. Janiga, R. Czarnota, J. Stopa, P. Wojnarowski, P. Kosowski, "Performance of nature inspired optimization algorithms for polymer Enhanced Oil Recovery process", in Journal of Petroleum Science and Engineering, 154, 354-366, 2017, doi: 10.1016/j.petrol.2017.04.010.

[29] H.G. Zhang, Z.H. Liang, H.J. Liu, R. Wang, Y.A. Liu, "Ensemble framework by using nature inspired algorithms for the early-stage forest fire rescue — A case study of dynamic optimization problems", in Engineering Applications of Artificial Intelligence, 90, 2020, doi: 10.1016/j.engappai.2020.103517.

[30] H.A. Choudhury, N. Sinha, M. Saikia, "Nature inspired algorithms (NIA) for efficient video compression – A brief study", Engineering Science and Technology, an International Journal, 23(3), 507-526, 2020, doi: 10.1016/j.jestch.2019.10.001.

[31] J.G. dos Santos Júnior, J.P.S. do Monte Lima, "Particle swarm optimization for 3D object tracking in RGB-D images", in Computers & Graphics, 76, 167-180, 2018, doi: 10.1016/j.cag.2018.09.011.

[32] R.D. Badgujar, P.J. Deore, "Hybrid Nature Inspired SMO-GBM Classifier for Exudate Classification on Fundus Retinal Images", IRBM, 40(2), 69-77, 2019, doi: 10.1016/j.irbm.2019.02.003.

[33] S.-I. Bejinariu, H. Costin, "A Comparison of Some Nature Inspired Optimization Metaheuristics Applied in Biomedical Image Registration", in Methods of Information in Medicine, 57 (05/06), Georg Thieme Verlag KG Stuttgart – New York, pp. 280-286, 2018, doi:10.1055/s-0038-1673693.

[34] K. Niño, J. Peña, "A Based-Bee Algorithm Approach for the Multi-Mode Project Scheduling Problem", in Procedia Manufacturing, 39, 1864-1871, 2019, doi: 10.1016/j.promfg.2020.01.252.

[35] S.-I.Bejinariu, H. Costin, D. Costin, "Combinatorial versus Priority Based Optimization in Resource Constrained Project Scheduling Problems by Nature Inspired Metaheuristics", in Advances in Electrical and Computer Engineering, 19(1), 17-26, 2019, doi:10.4316/AECE.2019.01003

[36] K. Bibiks, Y-F. Hu, J-P. Li, P. Pillai, A. Smith, "Improved discrete cuckoo search for the resource-constrained project scheduling problem", in Applied Soft Computing, 69, 493-503, 2018, doi: 10.1016/j.asoc.2018.04.047.

[37] M. Faddy, "Examples of fitting structured phase-type distributions", in Appl. Stoch Models Data Anal, 10, 247-255, 1994.

[38] R. Cooper, "Introduction to queueing theory", 2nd Ed. New York, Elsevier, North Holland, 1981.

[39] F.M. Burnet, "A modification of Jerne's theory of antibody production using the concept of clonal selection", Australian Journal of Science, 1957.

[40] F.M. Burnet, "The clonal selection theory of acquired immunity", Vanderbilt University Press, 1959.

[41] J. Brownlee, "Clonal Selection Algorithms", Technical Report 070209A, Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne, University of Technology, 2007.

[42] L.N. De Castro, F.J. von Zuben, "Learning and optimization using the clonal selection principle", IEEE Transactions on Evolutionary Computation, 2002.

[43] L.N. de Castro, J. Timmis, "Artificial immune systems: a new computational intelligence approach", Springer, 2002.

[44] S.S. Tan, et al., "Direct cost analysis of Intensive Care Unit Stay in four European countries: applying a standardized costing methodology", Value of Health, 15, 81-86, 2012.

[45] A. Rahmi, W.F. Mahmudy, M.Z. Sarwani, "Genetic algorithms for optimization of multi-level product distribution", Int. Journal of Artificial Intelligence, Volume 18, Number 1, pp. 135-147, 2020.

[46] A. Naseri, S.M.H. Hasheminejad, "An unsupervised gene selection method based on multiobjective ant colony optimization", Int. Journal of Artificial Intelligence, Vol. 17, Number 2, pp. 1-22, 2019.