STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXX**, Number 1-2, 2025
DOI: 10.24193/subbi.2025.01

NETWORK OPTIMIZATION PROBLEM APPLICABLE FOR BREAST CANCER SCREENING COST MINIMIZATION

ATTILA MESTER AND ANCA ANDREICA

Abstract. We investigate the problem of breast cancer screening optimization, using various techniques applicable in domains where the data format is not defined in advance. The aim is to minimize the cost related to the screening of patients while maximizing the beneficial effect of the process regarding some key breast cancer indices. Our model can be easily adjusted to other similar network optimization tasks where a goal function has to be minimized across a geographical surface. We present the problem's key similarities to the Travelling Salesman Problem and underline the fact why we choose a deterministic algorithm compared to a Simulated Annealing-based solution. Furthermore, we present the usefulness of the Elastic Stack regarding this application and offer a concrete solution to the problem defined by our generated dataset, respecting the European data distributions in this domain.

Introduction

To optimize the breast cancer screening process, it is necessary to analyze the trends of various domain indicators. These can be simple indices like incidence or mortality rate, or other ones such as participation rate at screening process / prevention campaigns, detection rate, malign/benign percentage, period between detection and first medical treatment, etc.

There are several works in the literature related to optimizing the breast cancer screening [7, 6, 1]. The advantage of the proposed method is that we offer a deterministic algorithm for finding the optimal number of mammography machines needed for a nationwide screening, we determine the exact location of the medical units based on the density of the population and the

Received by the editors: 1 March 2023.

2010 *Mathematics Subject Classification.* 91-04, 91-08.

1998 *CR Categories and Descriptors.* G.1.6 [**Optimization**]: Subtopic – *Simulated annealing* G.1.6 [**Optimization**]: Subtopic – *Constrained optimization.*

Key words and phrases. optimization, network, graph traversal, DBSCAN.

distribution of medical personnel, and offer a detailed technical description of the process, including where and when each patient needs to be examined, and how many days the national screening will last according to the algorithm.

In the first part of this work, we present existing literature for the analysis of indicator trends. The second part of the paper presents the proposed framework used to simulate a nationwide screening process and offers a deterministic solution regarding the optimal cost–screening time ratio.

1. Joinpoint regression analysis

Joinpoint regression analysis is in fact a linear regression, built up of multiple segments – the number of which is undefined and has to be determined by the algorithm. This regression approximates a set of points, in our case, on the format $(year, indexvalue)$. In the literature, this problem is known as joinpoint regression with k points. Formally, one can state the problem as follows. Let us note the points $(x_1, y_1), (x_2, y_2), .., (x_n, y_n)$, find the regression model function M, so that the least square error (LSSQ) of the model is minimized – as stated in Equation 1 [5].

$$\begin{aligned} M(x) = ax + b + s_1(x - t_1)^+ + .. + s_k(x - t_k)^+ \\ LSSQ = \sum (y_i - M(x_i))^2, \end{aligned} \tag{1}$$

where t represents the positions of the joining points, and $e^+ = e$ if $e > 0$ else 0.

The linear regression problem is a classic one, having an analytical but also an estimative solution, obtained by optimization methods. In our case, the problem has two components: finding the optimal number of joining points k, and determining the positions (x, y) of these points. These two issues define the directions and approaches in the literature. We will present in turn these two problems. Fig. 1 shows a data set (index), on which we will build the regression in several sequences. This dataset was used in [5], and contains incidence rates for prostate cancer in the US. between 1975 and 1995.

1.1. **Dataset.** In order for the experiments to be as relevant as possible, our goal was to use real data from the studied field. The European Cancer Information System[1] offers a REST API, which exposes a huge amount of historical statistics. By sending 60 million requests to this server, we were able to populate a local Elasticsearch database. These requests are answered by a JSON

[1] https://ecis.jrc.ec.europa.eu/

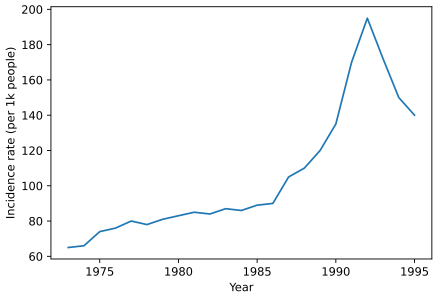


Figure 1. Incidence rate of US prostate cancer, 1975 – 1995.

dictionary in the following format: which region reported the indicator (city/-country), age range and sex of the people for whom the index belongs, year, and type of cancer.

The advantage of this system is that having the data locally, we can analyze the data set by applying custom visualizations, not only through some plots provided by the official site. Thus, we created a Kibana dashboard containing several views that provide us with valuable information on the trend for several types of cancer, age categories, year of onset, etc., shown on a timeline in which the index was recorded. This timeline gives us a kind of histogram of documents, showing how many records were reported in the selected period in Kibana.

1.2. **Scientific approach - Literature review.**

It is easy to prove that the higher the k, the smaller the LSSQ error, so the more accurate the regression. However, taking into account the fact that in the field of medicine, we are looking for a more general trend, extended over a longer period of time, we are looking for k as small as possible - so that the regression error is still as small as possible. To determine the optimal number k, there are several variants in the literature. In a recent paper [3], Gkioulekas and Papageorgiou applied two aspects from information theory to penalize a more complex model, that is, to keep k as small as possible. These are the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Both assume that there is no perfect model, and therefore the optimal model must be found by measuring the relative distance between the model and the ground truth. By penalizing a high k value, we can find the model with as few joining points as possible. Another approach found in the literature is described in [5], which favors a small k by applying the hypothesis testing method. In the initial hypothesis *H0*, they assume that the optimal model uses k_0 points, and the alternative hypothesis *H1* says k_1 points ($k_0 < k_1$). Thus, a higher number of joining points k_1 is accepted only if it results in a statistically relevant improvement to the model, applying permutation testing, as follows. To decide whether the initial hypothesis (using k_0) points can be rejected, we compare fitting models several times using k_0 and k_1 points. But, each time, on a dataset slightly altered from the initial one, introducing noise, by permuting the error vector from the initial model. There are cases in which the relative quality between the two models is at least as good as at the initial fitting, on the initial dataset. If this percentage number (p value) is high enough (above a 5% threshold), then we do not reject the initial hypothesis, we accept it because the changes assumed by H1 do not bring the relevant improvements. For a defined k, we must determine the position of the joining points. In their paper [5], Kim et al. mention an exhaustive search method called the grid-search method, meaning that any combination of positions for points in the range of x_{min} and x_{max} is considered. In practice, this method is too expensive regarding execution time and allows only positions with integer value for abscissa. A demonstration of this method is shown in Fig. 2. Here we can see how the grid-search method works in finding the optimal position for the $k = 2$ joining points. Each subplot represents a unique position for these two points, shown as their title. The blue lines represent the input data – US prostate cancer incidence rate, as shown in Fig. 1. The X and Y axis represent the year and the incidence rate, respectively. The red lines represent the output of the joinpoint regression algorithm using the respective two joining points.

The approach described in [3] is to test of each k, until the AIC or BIC indices improve (i.e. do not decrease). This method is built on the idea described in [8], called Optimal Piecewise Linear Regression Analysis (OPLRA), which can determine the positions of the joining points - the disadvantage is that k must be fixed by the user, and in [3] it is determined by the algorithm. To summarize the necessary steps, we will need the following functions.

i. A function that has as input the points over which we apply the regression (X, Y), and k. Having k, we can estimate the initial positions of the joining points by dividing the sequence $[x_{xmin}, x_{max}]$ into k equal parts. Then we use the simplex method and the LSSQ error function to determine each of the k points.

ii. A function that has as input k_0 and k_1, and determines the optimal model, by direct comparison, either using indices for model complexity as AIC, or hypothesis testing.

After applying the regression for each of the grid-search cases, we obtain the optimal model, having $k = 2$ joining points at the abscissas 1988 and 1992, demonstrated in Fig.3.

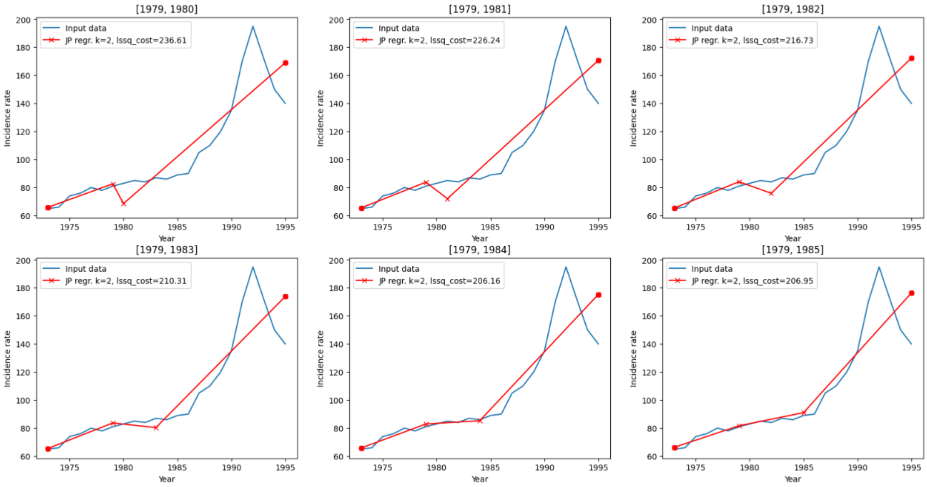


Figure 2. Extract from the grid-search method to determine the optimal position for the $k = 2$ joining points.

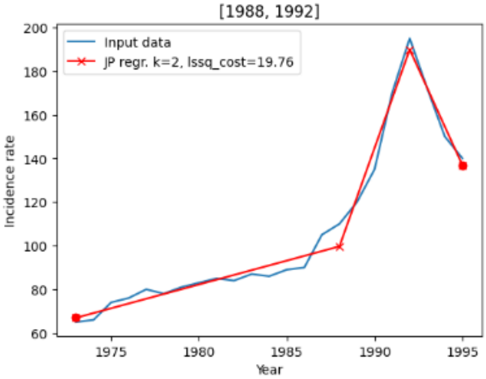


Figure 3. The optimal regression model with $k = 2$ joining points in the abscissas 1988 and 1992.

2. Optimizing a national screening process

In this section, we describe our methodology used for optimizing the process of breast cancer screening. Since the task lacks essential specifications, our focus was to determine the dataset, and the goal functions that we will optimize (e.g. minimize the logistical cost – the transport of equipment, maximize the amount of help the application offers to the population).

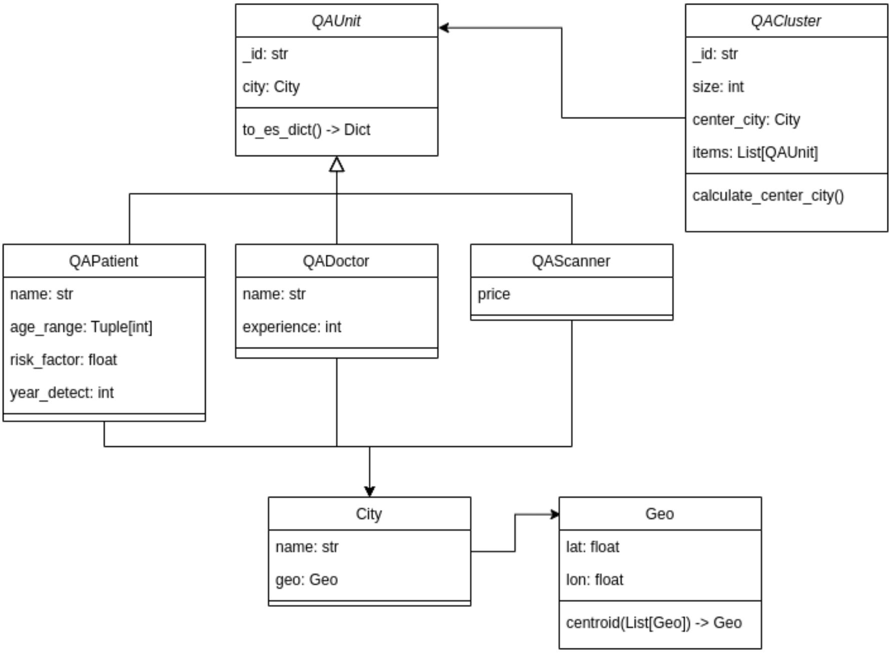


FIGURE 4. Entities of the generated dataset.

2.1. **Defining the data structure.** We need to generate a population with a given age, risk factor, and geographical distribution, generate an appropriate number of healthcare specialists, and also some mammography machines. This project aims the study of Romania's population, so the number of the generated healthcare people in the current experiment is influenced by national statistics. Similarly, the geographical position of doctors and patients is determined by national data. European statistics also take a crucial part in determining the age and risk factor distribution of the generated patients[2]. The generated data is stored in a local Elasticsearch[3] database in order to be able to query the data for subsequent analysis, and a live Kibana dashboard is also created, in order to visualize the nature of the generated data – and possible, to track the runtime of the future algorithm. The dashboard reflecting the current dataset is shown in Fig. 5. The data generated by us follows OOP principles, as shown in Fig. 4.

2.2. **Data storage.** We use the Elastic Stack in order to offer three separate containers. *Logstash* corresponds to an ETL process, used in the real-time

[2] `https://ecis.jrc.ec.europa.eu/`

[3] `https://www.elastic.co/elastic-stack/`

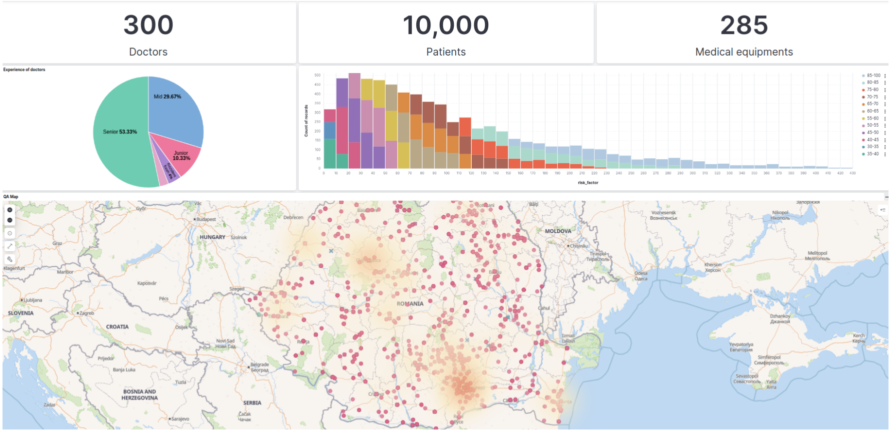


Figure 5. Kibana dashboard of the generated healthcare people, medical equipment and patients.

insertion of data. Our generated data will be stored in *Elasticsearch*, a NoSQL database, which will be visualized using *Kibana*. In our Logstash pipeline configuration, we use the *mutate:convert* filter plugin to extract the latitude and longitude values from the input file.

2.3. **Generating the data.** In order to develop a realistic algorithm, we needed to generate a population of entities representing patients with a certain age, risk factor, and geographic distribution, furthermore, also health professionals and mammography machines (scanners). The project is specific to the population of Romania, the data generation module respects the national statistics regarding the number of medical professionals but also their geographical position, as well as of the patients. European statistics determine the age and risk factor distribution of the generated patients. The dashboard created to visualize the generated data is shown in Fig. 5. The first row depicts the number of different entities: 300 doctors with different experiences in the field, 285 scanners, and $10,000$ patients with age distribution and breast cancer risk factors according to European data. The second row shows a pie chart regarding the seniority level of the medical professionals: more than half of the personnel are generated with senior experience level, while there are 10% juniors i.e. fresh graduates – these percentages can also be modified within the data generation module. We can also see a distribution plot of the risk factor values separated by age groups, and a heatmap over the generated entities.

2.4. **Simulated annealing (SA).** The optimization of a screening process refers to minimizing the expenses related to the overall operation – possibly both governmental and private costs, and also the time required for the process, measured in days. This optimization resembles the classic Traveling Salesman Problem (TSP). In TSP, we have an agent that traverses a list of cities, each one exactly once, starting from a city and returning to the same location. The goal is to minimize the length of the road – the distance travelled by the agent. TSP is a classic NP-hard problem, which means that there is no polynomial algorithm that finds the optimal solution, and we have to apply different heuristics and optimization methods to find an optimal solution – a classic approach being evolutionary algorithms (GA), or SA. A comprehensive study on the TSP problem is offered in [2]. In this section we will present the similarities of our screening problem with that of TSP, we will present our technical approach, but also the reasons why this SA method could not be used in this project.

In the first step, we assume that we have a single doctor and a modular scanner that can be carried by this doctor. Then, the national screening process consists of the path travelled by this doctor, who has to reach every city and arrive back home, from where they started the traversal. The function to be minimized is the path travelled by the doctor — i.e. an optimal order of cities must be found. We now add the first element of complexity.

* We have multiple agents (doctors), and the union of their route must contain every city (not necessarily just once). The goal is to reduce the length of routes travelled by doctors. In order to simulate the screening as described in this project, we add one more element of complexity.

** The doctor does not necessarily have to visit the city, it is possible that the "the city visits the doctor", in the sense that the patients choose the nearest center of screening, and doctors have a fixed location – a medical center, or healthcare unit.

In classical TSP, a possible solution is to use an SA-based algorithm, which provides a random, initial order of cities, and through iterative changes applied to this list (e.g. mutation), we obtain an optimal order of the cities, according to a certain function, known as the fitness function, e.g. the length of the route. Given the fact that in this project we have the detail *, we cannot use SA with a single list of cities, respectively a single cost function. We need to use one list of cities for each doctor, and simultaneously apply SA on these lists. This approach is possible, but we still need the cost function applied on the combination of all lists, which evaluates if the existing configuration has errors such as cities visited by multiple doctors. This means that those SAs cannot be run independently, and any configuration of different SAs must be

```
1. day:
[ unit1-city1-patient1, unit2-city2-patient2,  u3-c3-p3 ...]
2. day:
[ unit5-city3-patient5, unit3-city1-patient8,  u2-c4-p9 ...]
```

Figure 6. Simulated annealing model: the aim is to reach an optimal order of scanning operations.

checked on each route (per doctor), which results in an unacceptable runtime, not applicable in practice (having 300 routes according to 300 doctors, and 537 cities). Besides these clear limitations, the information on required days is completely ignored by this model.

Given that we have one more complexity **, even if we manage to get a result with the method described above, the solution would not reflect a real-life screening process. We need to use another SA model. We can use another list of fixed length, which represents a day in the screening process, where one element represents a scanning operation, i.e. a union of the medical center, doctor, and patient – the goal is to get an optimal order, as shown in Fig. 2.4.

This model solves the problem of the previous model, we will also determine the required days. After SA is completed, we get the list of patients that will need to be scanned on the first day. After which, we repeat the process on the next day, etc. Unfortunately, this method is not practical either, for several reasons:

- runtime – with the final algorithm proposed in this report, the minimum required time is 180 days, which means 180 SA simulations according to this model
- the list mentioned in this model is extremely long, we have a Cartesian product between patients ($10,000$), cities (537), and centers (see arg. 3)
- the number of centers is unknown. For this model, we need the number of medical centers, which should be determined by the algorithm.

In addition to all of the above, SA is non-deterministic, a property that would not be preferred in an application aimed to optimize a process that impacts the health of thousands of patients. The final algorithm proposed in this paper was formed upon analysis of all of the key points listed above, which we present in the following.

2.5. **DBSCAN clustering.** Optimizing the national screening process requires an algorithm to find the required number of mammography machines (scanners) and medical centers, but also their location. Applying an optimization method to obtain the scanners (their number and location), due to reasons described in Section 2.4, would result in an unacceptable runtime, furthermore, no guarantee would be offered on the optimality of the solution. Considering that in practice, mammography machines will be placed in already existing medical centers, probably in larger cities, resulted in the basic concept of the proposed solution: fixing the number of scanners. If we assume that we have n scanners, we can determine the cities in which they should be placed – according to the geographical distribution of doctors and patients. For this step, we need to obtain this distribution, i.e. to cluster the doctors, respectively the patients, according to their location - eventually, we obtain a map with *QACluster* documents (Fig. 4) which we can validate as a debugging method by comparing with the heatmap provided by Kibana, depicted in Fig. 5. DBSCAN is an unsupervised learning method, based on the concept of spatial density (according to a distance function), according to which it groups the elements into tight clusters. The advantage over K-means is that the number of clusters does not need to be specified beforehand, which is very useful in this application regarding spatial data [4]. The algorithm works with the following important concepts:

- Eps – parameter that defines the radius of a neighborhood
- minPoints – the minimum number of nodes that must be in the vicinity of the Eps radius of a node n for n to be a base node in the cluster.

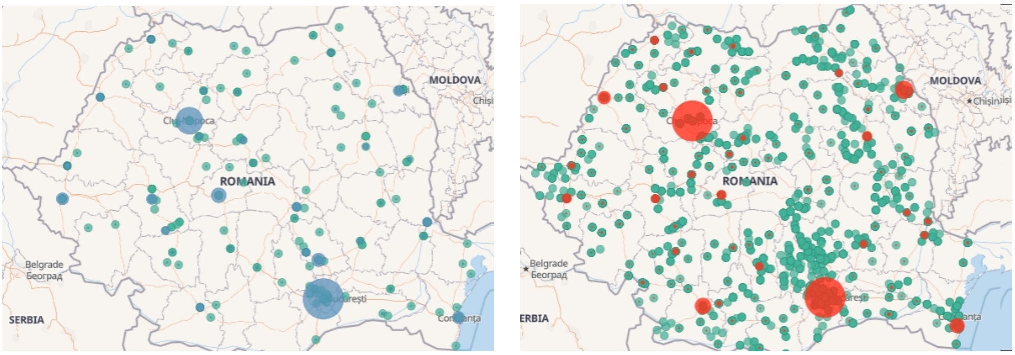
The benefits of using DBSCAN clustering in a TSP problem are detailed in a recent work [1], where the authors argue that this method facilitates a more efficient computation time of the simulation.

For the application of DBSCAN clustering, we use the *sklearn* library[4].

The result of this clustering applied on $10,000$ patients and 300 doctors is shown in Fig. 5. Having obtained these clusters, we can continue the description of the proposed algorithm by distributing the scanners in the largest clusters.

2.6. **Deterministic grid-search algorithm.** The essence of the proposed algorithm consists of trying each n for the number of scanners, and simulating the screening process having n scanners. First, we distribute the machines in the largest n cities in each possible way, evaluating the cost related to this processes. Then, we select the optimal number n for which the cost functions are optimal, according to certain hybrid criteria. Although the proposed method is an exhaustive search algorithm, the runtime is absolutely acceptable, running several thousand simulations in under 5 minutes, and it has the

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html`



Doctors Patients

FIGURE 7. DBSCAN clustering on the generated dataset of $10,000$ patients and 300 doctors.

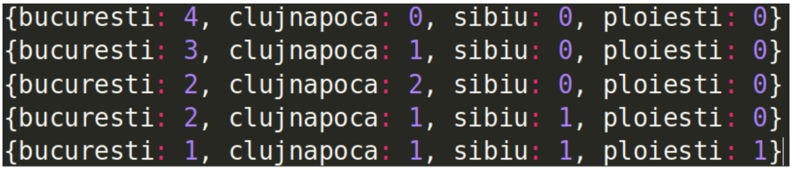
advantage that the obtained result is deterministic – offering a guarantee that for a certain set of generated data (doctors, patients), the optimal simulation chosen by the algorithm is constant. This is a clear advantage over an SA-type method or other metaheuristics.

2.7. **Simulation with n mammographic machines.** The proposed algorithm is based on the assumption that if we determine the examination centers, then the screening process is well defined, we have no other aspects to optimize – the centers determine the patients' decision, and each patient will choose the one that is closest. In other words, all we need to optimize through this proposed algorithm is the choice of the number of centers, their location, and the distribution of n scanner machines in these centers, so that the cost related to this model is minimal.

The algorithm starts with distributing the scanners in examination centers (Section 2.8). For each distribution, the screening process is simulated: until we still have patients to examine in the queue, we start a new day, we select a batch of patients who can be scanned based on the number of machines, we determine the nearest city for each patient, and we send them to be examined, removing that patient from the queue. In this process, we take into account the maximum capacity of a center, but also the number of possible scans with a single machine, per day – detailed in Section 2.9. If a center if filled according to its maximum capacity, the model is penalized due to the associated cost of an overcrowded medical center. If – for some reason – the patient cannot be scanned on the current day (e.g. overcrowded center), then we postpone them to the next day. These are decisive factors to select the optimal model, because if a medical center in a very dense area of patients does not have enough scanners, then the number of days needed for screening increases, precisely because patients are continuously rescheduled. The algorithm can determine the optimal number of machines to avoid this, also considering the initial investment cost of the machines.

2.8. **Distribution of scanner machines in centers – backtracking.** The simulation function receives as input the number of scanners, the list of doctors and patients, but also the list of clusters of doctors and patients, determined by DBSCAN. The basic concept is that the scanners will be distributed in the largest cities, thus, for n machines, we must distribute them to the largest n clusters of doctors. A hybrid selection might be applied that also takes into account patient clusters (Section 3), for the time being it was assumed that the natural distribution of doctors and patients is not significantly different. Since we want to provide a deterministic solution, we will try each combination of the distribution of the scanner machines in the largest cities, with one constraint: a smaller city cannot have more scanners than a larger city. Algorithmically, the problem translates to finding all non-increasing sequences of fixed length, having the sum equal to a number n. For this, we apply the backtracking method. An example for the distribution of cars in cities, having $n = 4$ cars is depicted in Fig. 8.

2.9. **Fitness functions & Constants.** Fitness functions, in a general metaheuristic context, measure how close a single unit of solution is to the desired achievements according to some fixed criteria. Once we have several simulations, we need to sort them according to certain criteria (i.e. the output of our fitness function) so that we can choose the top k of them. For a certain number of n scanners, we will select the top 3 runs, according to a hybrid criterion, and for all simulations, we select the top 5 – it is important to mention this aspect because although the algorithm selects the top k solutions, also marked on the final plot, the user may choose any other run, regardless of the algorithm's decision. The proposed algorithm optimizes two aspects: the cost related to national screening, and the days needed to examine all patients. These two aspects are not comparable, their importance relies on some



```
{bucuresti: 4, clujnapoca: 0, sibiu: 0, ploiesti: 0}
{bucuresti: 3, clujnapoca: 1, sibiu: 0, ploiesti: 0}
{bucuresti: 2, clujnapoca: 2, sibiu: 0, ploiesti: 0}
{bucuresti: 2, clujnapoca: 1, sibiu: 1, ploiesti: 0}
{bucuresti: 1, clujnapoca: 1, sibiu: 1, ploiesti: 1}
```

FIGURE 8. Distribution of $n = 4$ mammography machines in the largest n clusters of doctors.

outer factors which we add by weight constants to express one single score of weighted sum of these measures. All constants used in these cost functions are shown in Fig. 9. We present the cost functions in the following.

(1) Governmental cost. Naturally, if we have 537 scanners, we can create an examination center in each city, so no patient will have to travel to another city, and the screening will be extremely fast. However, this way the government cost increases a lot because many scanners have to be bought and more centers are needed to be prepared. The algorithm increases the government cost with each scanner purchased, and each test center has a daily maintenance cost (set to a small value, because centers exist in larger cities anyway), so a model with multiple scanners and centers is penalized.

(2) Private cost (Patient cost). That is, the cost paid by patients. Optimally, a patient should not travel a lot for a screening - therefore, the algorithm selects the nearest medical center, and penalizes the model with the cost of travel. If the distance is long enough that the patient will need accommodation, then the model is also penalized with an accommodation cost.

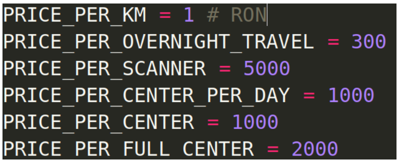
(3) Total cost. The sum of the above two. Ideally, both costs should be minimized.

(4) Days required. A national screening should not take years – as is the result of running with 3 scanners (560 days). Thus, the model is rewarded for offering a solution which finishes the screening of all patients in as few days as possible.

(5) Hybrid criterion. Each of the mentioned aspects must be taken into account, thus, for the selection of the best runs, we take into account the necessary days, and the total cost (with 70% weight on the government cost).

3. Improvement ideas

The distribution of mammography machines is based on the location of doctors, assuming that a dense region of doctors probably means a dense



```
PRICE_PER_KM = 1 # RON
PRICE_PER_OVERNIGHT_TRAVEL = 300
PRICE_PER_SCANNER = 5000
PRICE_PER_CENTER_PER_DAY = 1000
PRICE_PER_CENTER = 1000
PRICE_PER_FULL_CENTER = 2000
```

Figure 9. Constants used in the screening simulation.

region of patients as well. The algorithm is easily extensible with another weighting parameter that would mean taking into account the geographical distribution of patients as well.

Machines are placed into the largest clusters of doctors (i.e. medical centers), based on a backtracking algorithm, presented in Section 2.8. Unfortunately, this solution grows exponentially, for $n = 10$ scanners, we have 42 possibilities of placing them in 10 cities, and for $n = 20$ scanners, we have 627 possibilities, each resulting in a new simulation. An optimization method could be applied that considers only those configurations that have a chance that during the simulation, the model will have a minimum cost.

At the moment, a series of metadata-level information is not used by the proposed algorithm, precisely to simplify the problem and generalize the proposed solution. Information such as the price of a scanner, the age and risk category of the patient, as well as the experience of doctors, are available at the level of data generation and their storage in Elasticsearch (searchable in Kibana), but at the level of the optimization algorithm, they are not used. These constants can be adjusted according to global standards [6]. The scanner has a fixed price and patients are queued in a randomized order, not taking into consideration their risk factor – which is a must in an ideal screening programme [7]. The algorithm is easily extensible to take this information into account as well.

4. Results and discussion – validating the health economics tool

In this section, we present different simulations for the dataset shown in Fig. 5. The algorithm started with $n = 1$ mammography machines and stopped at 20 because the cost function no longer improved. On the generated plots shown in Fig. 10, 11, 12, 13, one dot represents one screening simulation, the x and y axes will show the days needed for screening, respectively the total cost, and the size of the points represents the percentage of the government cost in the total cost – it will be important at higher n values. Furthermore, the header contains all the technical details related to the result of the particular simulation. We can observe that these images also show the three best runs according to our hybrid criteria: taking into account both the cost and the days required to run the screening.

We analyze $n = 2$ scanners in Fig. 10. The selected centers will be in Bucharest and Cluj-Napoca, and the most optimal run regarding the required days is when we have both machines placed in Bucharest - the density of patients is higher there.

We analyze $n = 5$ scanners in Fig. 11. We can see that the algorithm selects the most optimal simulations with 2, 3 and 4 medical centers, marked with

X. The cost associated with them is interesting: the total cost decreases as there are more examination centers due to the patient cost (lower travel costs). From the size of the points we can see that the percentage of government cost in the total cost has increased from 2 centers to 4 centers, which means that in total, the final cost has become lower.

We analyze $n = 14$ scanners in Fig.12. We can see that the total price of the screening process has decreased, although we now have more scanners and centers to maintain, which means an increased percentage of government cost in the total cost -– reflected by the point size, compared to previous simulations. According to the government cost criterion, the best model has only 2 centers, and according to the private cost we have 14 centers. The selected optimal model has 12 medical centers, with 2 machines each in Bucharest and Cluj-Napoca, and 1 in every other center.

We analyze all of the simulations in Fig. 13. We have nearly 2000 simulations, for scanners between $n = 1$ and $n = 20$, the execution time being only a few minutes (3 minutes). We can see that as we add mammography machines, the government cost increases 0000and the private cost decreases. But since more centers and scanners result in fewer rescheduled patients and overcrowded centers (both penalized by the model), the government cost does not increase as much as the private cost decreases. This optimization trend is observed up to $n = 19$, where it is not worth adding more scanners – the selected optimal models have 15, 16, 17, 18, and 19 mammography machines, respectively, in $14 - 15$ centers, and they require ≈ 180 days for a complete screening of the patients.

5. Conclusions and future work

This paper presents a novel framework for optimizing a nationwide breast cancer screening process. The key aspects which are offered by this algorithm are the deterministic nature of the solution it provides, and the precise description of the simulation details, offering guidelines for each patient on which day and which examination center they should visit in order for the screening to be carried out in the shortest timespan while minimizing governmental and private expenses. Future work includes and is not limited to implementation details specified in Section 3.

6. Acknowledgments

This work was supported by a grant of the Romanian Ministry of Education and Research, CCCDI - UEFISCDI, project number PN-III-P2-2.1-PED-2019-2607, within PNCDI III.

```
        == Simulating 2 scanners, in 2 ways across the cities
====== Best configurations: ======
  1. BALANCED #0     :: 2 centers |  756 days | all cost  2611269.1 ;
  gov. cost   163200.0 (6.2%) ; civil cost  2448069.1
  1. BALANCED #1     :: 1 centers |  625 days | all cost  3919550.0 ;
  gov. cost    73500.0 (1.9%) ; civil cost  3846050.0
  2. GOV. COST       :: 1 centers |  625 days | all cost  3919550.0 ;
  gov. cost    73500.0 (1.9%) ; civil cost  3846050.0
  3. CIVIL COST      :: 2 centers |  756 days | all cost  2611269.1 ;
  gov. cost   163200.0 (6.2%) ; civil cost  2448069.1
  4. ALL COST        :: 2 centers |  756 days | all cost  2611269.1 ;
  gov. cost   163200.0 (6.2%) ; civil cost  2448069.1
  5. FEWEST DAYS     :: 1 centers |  625 days | all cost  3919550.0 ;
  gov. cost    73500.0 (1.9%) ; civil cost  3846050.0
1.0 {bucuresti: 1, clujnapoca: 1}
1.1 {bucuresti: 2}
2. {bucuresti: 2}
3. {bucuresti: 1, clujnapoca: 1}
4. {bucuresti: 1, clujnapoca: 1}
5. {bucuresti: 2}
```

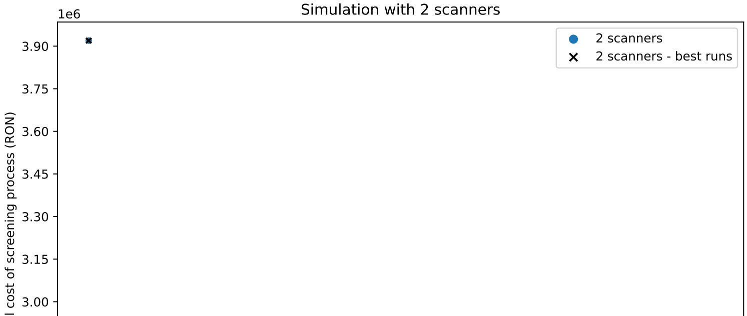


FIGURE 10. Simulation with $n = 2$ scanner machines: 2 centers, 625 days needed.

```
    == Simulating 5 scanners, in 7 ways across the cities
====== Best configurations: ======
  1. BALANCED #0     :: 4 centers |  360 days | all cost  2210048.4 ;
  gov. cost   173000.0 (7.8%) ; civil cost  2037048.4
  1. BALANCED #1     :: 3 centers |  355 days | all cost  2457990.6 ;
  gov. cost   134500.0 (5.5%) ; civil cost  2323490.6
  1. BALANCED #2     :: 2 centers |  314 days | all cost  2887869.1 ;
  gov. cost   439800.0 (15.2%) ; civil cost  2448069.1
  2. GOV. COST       :: 3 centers |  355 days | all cost  2457990.6 ;
  gov. cost   134500.0 (5.5%) ; civil cost  2323490.6
  3. CIVIL COST      :: 5 centers |  581 days | all cost  2124036.2 ;
  gov. cost   320500.0 (15.1%) ; civil cost  1803536.2
  4. ALL COST        :: 5 centers |  581 days | all cost  2124036.2 ;
  gov. cost   320500.0 (15.1%) ; civil cost  1803536.2
  5. FEWEST DAYS     :: 2 centers |  314 days | all cost  2887869.1 ;
  gov. cost   439800.0 (15.2%) ; civil cost  2448069.1
1.0 {bucuresti: 2, clujnapoca: 1, oradea: 1, sibiu: 1}
1.1 {bucuresti: 2, clujnapoca: 2, oradea: 1}
1.2 {bucuresti: 3, clujnapoca: 2}
2. {bucuresti: 2, clujnapoca: 2, oradea: 1}
3. {bucuresti: 1, clujnapoca: 1, oradea: 1, sibiu: 1, craiova: 1}
4. {bucuresti: 1, clujnapoca: 1, oradea: 1, sibiu: 1, craiova: 1}
5. {bucuresti: 3, clujnapoca: 2}
```

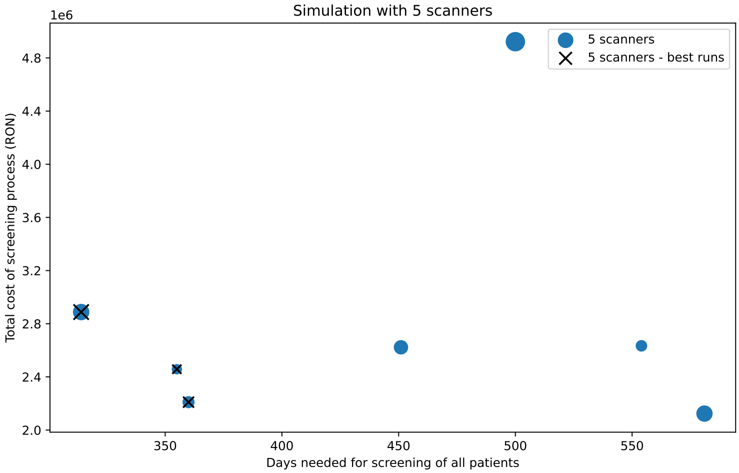


FIGURE 11. Simulation with $n = 5$ scanner machines: $2 - 4$ centers, $314 - 360$ days needed.

```
    == Simulating 14 scanners, in 135 ways across the cities
====== Best configurations: ======
  1. BALANCED #0     :: 12 centers |  207 days | all cost   935075.8 ;
  gov. cost   330400.0 (35.3%) ; civil cost   604675.8
  1. BALANCED #1     :: 12 centers |  211 days | all cost   985875.8 ;
  gov. cost   381200.0 (38.7%) ; civil cost   604675.8
  1. BALANCED #2     :: 10 centers |  206 days | all cost  1012609.7 ;
  gov. cost   286000.0 (28.2%) ; civil cost   726609.7
  2. GOV. COST       :: 2 centers |  554 days | all cost  2678869.1 ;
  gov. cost   230800.0 (8.6%) ; civil cost  2448069.1
  3. CIVIL COST      :: 14 centers |  332 days | all cost  1087803.5 ;
  gov. cost   548800.0 (50.5%) ; civil cost   539003.5
  4. ALL COST        :: 12 centers |  207 days | all cost   935075.8 ;
  gov. cost   330400.0 (35.3%) ; civil cost   604675.8
  5. FEWEST DAYS     :: 6 centers |  192 days | all cost  2095934.3 ;
  gov. cost   567200.0 (27.1%) ; civil cost  1528734.3
1.0 {bucuresti: 2, clujnapoca: 2, oradea: 1, sibiu: 1, craiova: 1, constanta: 1,
albaiulia: 1, iasi: 1, ploiesti: 1, arad: 1, calan: 1, baiamare: 1}
1.1 {bucuresti: 3, clujnapoca: 1, oradea: 1, sibiu: 1, craiova: 1, constanta: 1,
albaiulia: 1, iasi: 1, ploiesti: 1, arad: 1, calan: 1, baiamare: 1}
1.2 {bucuresti: 2, clujnapoca: 2, oradea: 2, sibiu: 2, craiova: 1, constanta: 1,
albaiulia: 1, iasi: 1, ploiesti: 1, arad: 1}
2. {bucuresti: 13, clujnapoca: 1}
3. {bucuresti: 1, clujnapoca: 1, oradea: 1, sibiu: 1, craiova: 1, constanta: 1,
albaiulia: 1, iasi: 1, ploiesti: 1, arad: 1, calan: 1, baiamare: 1, timisoara: 1,
brasov: 1}
4. {bucuresti: 2, clujnapoca: 2, oradea: 1, sibiu: 1, craiova: 1, constanta: 1,
albaiulia: 1, iasi: 1, ploiesti: 1, arad: 1, calan: 1, baiamare: 1}
5. {bucuresti: 3, clujnapoca: 3, oradea: 2, sibiu: 2, craiova: 2, constanta: 2}
```

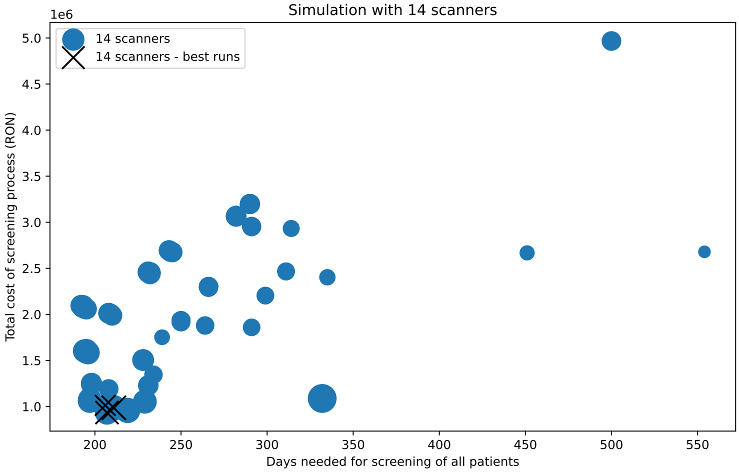


Figure 12. Simulation with $n = 14$ scanner machines: 12 centers, 207 days needed.

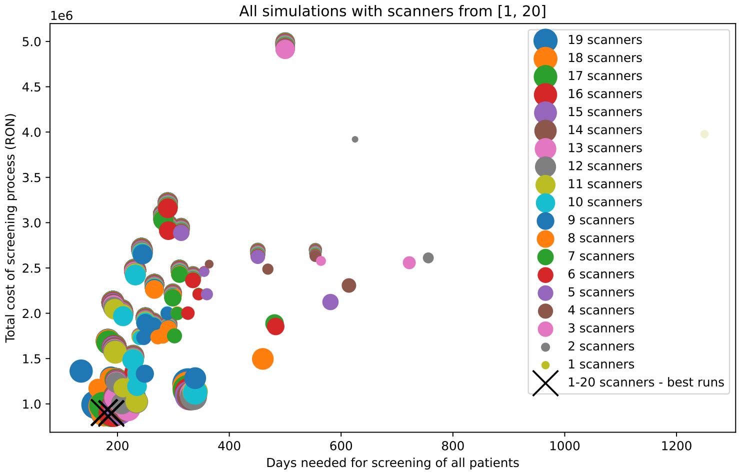


FIGURE 13. Simulation with $n = 1-20$ scanner machines; best result yield by $16 - 18$ machines, ≈ 180 days needed.

References

[1] Matthew Agostinelli. Density-based clustering heuristics for the traveling salesman problem. 2017.

[2] Omar Cheikhrouhou and Ines Khoufi. A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy. *Computer Science Review*, 40:100369, 2021.

[3] Ioannis Gkioulekas and Lazaros G Papageorgiou. Piecewise regression analysis through information criteria using mathematical programming. *Expert Systems with Applications*, 121:362–372, 2019.

[4] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014.

[5] Hyune-Ju Kim, Michael P Fay, Eric J Feuer, and Douglas N Midthune. Permutation tests for joinpoint regression with applications to cancer rates. *Statistics in medicine*, 19(3):335–351, 2000.

[6] Wenhui Ren, Mingyang Chen, Youlin Qiao, and Fanghui Zhao. Global guidelines for breast cancer screening: a systematic review. *The Breast*, 64:85–99, 2022.

[7] Adam Yala, Peter G Mikhael, Constance Lehman, Gigin Lin, Fredrik Strand, Yung-Liang Wan, Kevin Hughes, Siddharth Satuluru, Thomas Kim, Imon Banerjee, et al. Optimizing risk-based breast cancer screening policies with reinforcement learning. *Nature medicine*, 28(1):136–143, 2022.

[8] Lingjian Yang, Songsong Liu, Sophia Tsoka, and Lazaros G Papageorgiou. Mathematical programming for piecewise linear regression analysis. *Expert systems with applications*, 44:156–167, 2016.

Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca

Email address: {attila.mester, anca.andreica}@ubbcluj.ro