IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 6, 2018, accepted December 25, 2018, date of publication January 8, 2019, date of current version January 29, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2891424*

# The Minimization of Public Facilities With Enhanced Genetic Algorithms Using War Elimination

**PAVEL SEDA[1], MICHAEL MARK[2], KUAN-WU SU[3], MILOS SEDA[4], JIRI HOSEK[1], AND JENQ-SHIOU LEU[3], (Senior Member, IEEE)**

[1]Department of Telecommunications, Brno University of Technology, 616 00 Brno, Czech Republic
[2]Chair of Operations, Economics and Strategy, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland
[3]Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan
[4]Institute of Automation and Computer Science, Brno University of Technology, 616 69 Brno, Czech Republic

Corresponding author: Pavel Seda (xsedap01@vutbr.cz)

This work was supported in part by the U.S. Department of Commerce under Grant BS123456.

**ABSTRACT** In this paper, we focus on the problem of minimizing a network of state facilities that provide essential public services (schools, offices, and hospitals). The goal is to reduce the size of the network in order to minimize the costs associated with it. However, it is essential that every customer should be able to access an appropriate service center within a reachable distance. This problem can arise in various scenarios, such as a government cutting back on public service spending in remote areas or as a reaction to changing demographics (population increase/decrease). In general, this task is NP-hard which makes the problem particularly hard to scale. Therefore, for larger problems, heuristic methods must be employed to find an approximation of the optimum. To solve this problem with satisfactory results, we have presented an enhanced version of the genetic algorithm based on war elimination and migration operations. This modification overcomes the well-known shortcoming of GAs when the population becomes gradually more and more similar, these results in a diversity decrease which in turn leads to a sub-optimal local minimum. We test the performance of the novel algorithm against the standard heuristic benchmarks on the widely accepted Beasley OR-library dataset for optimization problems. Finally, we provide a case study based on real data, where a municipality tries to minimize the number of schools in a region while satisfying accessibility and other region-specific constraints.

**INDEX TERMS** Genetic algorithms, minimisation, public facilities, set covering problem, war elimination.

## I. INTRODUCTION

A set covering problem (SCP) is a notoriously difficult problem with many practical applications. The main challenge is its computational cost as its complexity is of $\mathcal{O}(2^n)$. Therefore, for $n$ sufficiently large, it is impossible to compute the exact solutions in a reasonable time [1], [2]. For this reason, instead of attempting to solve the problem using exact algorithms, it is often preferable to consider employing heuristic based simplifications. Although these do not guarantee finding the global optimum, it allows at least for a good approximation [3], [4].

The class of stochastic heuristics is widely applicable (e.g., to solving combinatorial optimization problems, in medicine to support the clinical decisions or to learn robot behavior [4], [5]) and creates an essential part of artificial intelligence methods. It contains genetic algorithms, simulated annealing, tabu-search, ant colony optimization, particle swarm optimization, firefly algorithm and others, but as the ''No Free Lunch Theorem'' states, for a general problem, there is no preference among heuristics [6]. It depends on a given problem, its instance, parameter settings, number of executions, etc. In spite of the fact that these methods are considered as methods of global optimization, we cannot avoid cases when they converge to a local optimum and find a solution far from the global optimum [2], [3], [5], [7].

Here, we focus on a genetic algorithm and propose its enhancement to overcome the problems mentioned above. The proposed algorithm is verified against a widely debated topic of the minimization of public facilities network, which provides services for their users in a reachable distance.

### A. COVERING SERVICES

As Europe is constantly demographically evolving, it is required to dynamically respond to its needs. Since the number of newly born children is constantly decreasing, it is evident that the number of public facilities such as schools, children's hospitals and libraries could be reduced, which would bring financial savings [8]–[10].

However, as these institutions are essential for its region, they cannot be removed in an arbitrary manner and a certain level of service has to be maintained. Any reduction plan has to adhere to these three points:

- Services have to be available within a specific range.
- Services have weights which express their value for the population (depends on the operational size of the service, prestige, and additional attributes).
- A selected subset of services must not be removed completely. These represent essential or deep-rooted facilities indispensable for the community that have to remain in the solution.

This problem has a complexity of $\mathcal{O}(2^n)$, which expresses the number of possible combinations of $n$ service centres in the final solution. Currently, there are few practical applications of set covering problems, e.g., the problem of Wireless Ad Hoc Network (WANET) to select nodes to forward a message to the next nodes/consumers [11] or designing the transit route network [12] which expresses the cost of the route network deviation from the shortest path. Although these papers are dealing with the same theoretical problem, the real-world use case brings different challenges that need to be addressed. In our case, we choose the reduction of the number of schools in a particular region in Europe.

### B. MAIN CONTRIBUTION

In this paper, we focus on a novel modification of a Genetic Algorithm (GA) that addresses a well-known issue of gradually decreasing the diversity in the population that leads to a local minimum. We show that this convergence can be substantially mitigated or fully avoided by employing the modification described in this paper. Our contribution is twofold. First, we propose an enhanced version of the GA that overcomes the problem of local minima based on introducing a so-called *war* operator. Furthermore, compared to other metaheuristic alternatives, our formulation allows for a straightforward and concise implementation. Second, we showcase the viability of the solution on a practical case study minimizing the network of public schools in a region in Central Europe.

The remaining part of this paper is structured as follows. In Section II, we provide an overview of genetic algorithms and introduce our *war elimination* modification. Section II-B details the newly created model, its implementation and showcase its ability to outperform traditional GA's in terms of individual fitness. The analysis and implementation of the case study on covering services are detailed in Section III. Finally, IV and VI present the results obtained and concluding remarks.

## II. PROPOSED ENHANCEMENT OF THE GENETIC ALGORITHM

### A. GENETIC ALGORITHM

*Genetic algorithm* is a stochastic adaptive algorithm containing the following operators and parameters:

$$GA = (N, P, f, \Theta, \Omega, \Psi, \tau) \tag{1}$$

where $P$ is a population of $N$ (*individuals*): $P = \{S_1, S_2, \ldots, S_N\}$. Each individual $S_i$, $i = 1, \ldots, N$ is a string (or set) of a fixed length with $n$ numbers (in the case of the SCP given by binary values) representing a solution to the problem.

$f$ refers to the so-called *fitness function*, which assigns a positive real number to each individual:

$$f : S_i \to \mathbb{R}^+, \quad i = 1, \ldots, N. \tag{2}$$

$\Theta$ is the *selection operator* that selects $u$ individuals from $P$:

$$\Theta : P \to \{P_1, \ldots, P_u\}. \tag{3}$$

$\Omega$ is a set of genetic operators, including a *crossover operator* $\Omega_c$, *mutation operator* $\Omega_m$ and possible other problem-specific operators that all together generate *offspring* (*children*) from parents

$$\Omega = \{\Omega_c, \Omega_m, \ldots\} : \{P_1, \ldots, P_u\} \to \{O_1, \ldots, O_v\}. \tag{4}$$

$\Psi$ is the *reproduction* operator that deletes $v$ selected individuals (typically given by randomly selected ones among the individuals with fitness function worse than the average) from actual population $P$. Subsequently, $v$ children are added to the new population $P(t+1)$.

$$P(t+1) = P(t) - \Psi(P(t)) + \{O_1, \ldots, O_v\}. \tag{5}$$

$\tau$ is the criterion for breaking the run of the genetic algorithm:

$$\tau : P(t) \to \{\texttt{True}, \texttt{False}\}. \tag{6}$$

The operator $\Theta$, genetic operators $\Omega$, and operators of reproduction $\Psi$ are stochastic [13]–[20].

The whole model is shown in Alg. 1. Detailed information about GA and GA operations, e.g., the selection types as roulette wheel selection or rank selection can be found in [15], [21]–[23], [23]–[25].

### B. PROPOSED ENHANCEMENT OF GENETIC ALGORITHM

There are several alternative formulations of the GA that aim at the problem of homogeneous population and avoiding local minima such as Tarpeian method [26], [27], which randomly kills individuals not adhering to given standards or Island model [28], [29] which partitions the population into sub-populations, where only local interactions are allowed (migrants are moved between sub-populations periodically). The disadvantage of these modifications over our solution is mainly their implementation difficulty as in the case of the island model (working with parallel populations is burdensome to implement) or a general lack of robustness and stability (Tarpeian method) [26]–[29].

**Algorithm 1** Standard Version of the Genetic Algorithm

**Input:** *numberOfIndividuals* – number of individuals in the population; *mutation* – mutation type/rate; *crossoverType* – type of crossover; *selection* – selection type; $\tau$ terminal condition, e.g., number of iterations

**Output:** the best individual $N$ (solution of the problem);

```
1: function geneticAlgorithm
2:     generate population (P);
3:     evaluate individuals (f);
4:     check if τ = true;        ▷ if true return the best N
5:     while τ != true do
6:         select individuals (Θ);
7:         create new individuals by mutation and crossover
       using selected (Ω_c, Ω_m);
8:         evaluate and insert newly created individuals (Ψ);
9:     end while
10: end function
```

To address these problems we propose a modification based on a novel operator called *war* inspired by the patterns in the population evolution observed during wartime periods. The operator unravels in three stages:

(i) *beforeWar* The first stage happens $x$ generations before the actual war and only archives the predetermined percentage $p_a\%$ of the population. The idea behind is that during the wartime, the social perception of the world radically changes. The archived population will serve as a base of significantly different individuals bringing this change in form of trends from the past.

(ii) *duringWar* During the war, population is sorted by its fitness value and divided into $n$ "social" classes with increasing sizes. Each class then suffers a population reduction due to two factors, $p_d\%$ is removed as "death toll" and $p_e\%$ is archived and removed as emigration. Emigrants are individuals who successfully managed to flee saving themselves and, thus, being completely removed from the population. The elimination ratio is increasing across the "social classes" reflecting the scenario from World War I and II where people with contacts and money were able to save themselves and emigrate. At the same time, archived individuals from *beforeWar* enter the population, bringing significantly different trends mirroring the change in the social perception of the world that the war brings.

(iii) *afterWar* After the war the emigrants from the previous phase come back into the population and the process ends with the same number of individuals as it started.

The flow of the population during the whole process can be viewed in Fig. 1 or as Alg. 2. It is important to point out that whenever individuals are removed either due to emigration or as a result of "death toll", the best individual has to remain in the population.

From the implementation perspective, sorting ($P$) is usually performed using MergeSort rather than QuickSort, even though the complexity is the same [30]–[32]. QuickSort has two major deficiencies compared to MergeSort:

1) It is not stable.
2) It does not guarantee $\mathcal{O}(n \log n)$ performance, it could easily degrade to quadratic performance on specific inputs.

The benefit of QuickSort compared to MergeSort is that it consumes less memory. MergeSort consumes $\mathcal{O}(n)$ external space while QuickSort only $\mathcal{O}(\log n)$ (to maintain the call stack) [33]. For memory constrained devices (e.g., embedded devices) it could be beneficial to replace the MergeSort algorithm with QuickSort for the sake of saving the memory [34], [35].

The example of parameter settings for dividing and reducing the population is shown in Tab. 1. These parameters could be arbitrarily modified to test the best case.

In essence, the war operator worsens the average fitness value of the population while bringing back new variants of the solution. As such, the ideal timing of the operator is when the fitness value starts to plateau – approaching the local minimum. This can be measured as a percentage change in the fitness value between consecutive generations falling under a given threshold (i.e. new generations do not bring enough improvement).

## III. MINIMISATION OF NETWORK COVERING SERVICES AND IMPLEMENTATION

We base our research on discussions presented in [9] and [10] on optimisation of public facilities (offices, schools and hospitals) that provide essential services for the population so

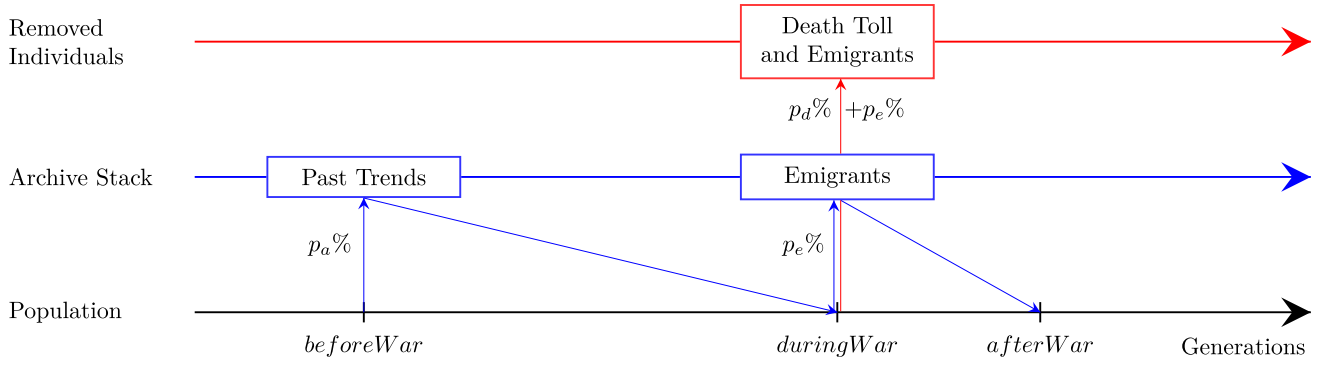



**FIGURE 1.** Process and population flow of the *war* operator.

**Algorithm 2** Proposed Enhancement of Genetic Algorithm

**Input:** *mutation* – mutation type/rate; *crossoverType* – type of crossover; *selection* – selection type; $\tau$ terminal condition, e.g., number of iterations

**Output:** the best individual (approximate solution);

```
1: function geneticAlgorithm()
2:     generate population (P);
3:     evaluate individuals (f);
4:     check if τ == true;        ▷ if true return the best
5:     while τ != true do
6:         switch war do
7:             case beforeWar
8:                 archive p_a% randomly selected individuals;
9:             case duringWar
10:                duringWar();
11:            case afterWar
12:                emigrants return back;
13:        select individuals (Θ);
14:        create new individuals with crossover (Ω_c);
15:        mutate them (Ω_m);
16:        evaluate and insert newly created individuals (Ψ);
17:    end while
18: end function

19: function duringWar()
20:    sort P descending;
21:    divide P into four groups;
22:    remove part of P from each group unevenly p_d% + p_e%;
23:    individuals moved out (emigrants) p_e%;
24:    migrants incoming at the end of the war p_a%;
25: end function
```

**TABLE 1.** The example of *duringWar* operation used on the population with 100 individuals.

| groups | number of ind. | elimination ratio [%] | reduced ind. |
|---|---|---|---|
| group 1 | 5 | 20 | 1 |
| group 2 | 15 | 40 | 6 |
| group 3 | 30 | 50 | 15 |
| group 4 | 50 | 56 | 28 |

that the cost is as low as possible and each inhabitant has a specific public facility in a reachable distance. Because the reachable distance is subjective and depends on the type of service, e.g., for schools it would be other than for hospitals and also it depends on political aspects and agreements between political parties, we omit these aspects and address a formal mathematical approach to solve these tasks.

In general, the set covering problem does not have any threshold of availability. It is given by the matrix of binary values and the covering depends on a suitable choice of rows. This task is NP-hard and, for larger instances $n > 50$ (see Tab. 2), could not be solved exactly in a reasonable time. For this reason, numerous heuristics have been developed [36]–[42].

**TABLE 2.** Computing time based on $n$ input value for Intel Core i5-3330 Procesor Quad-Core, 3 GHz processor.

| $n$ | $2^n$ | computing time |
|---|---|---|
| 10 | 1024 | $\approx$ 0,341 $\mu$s |
| 20 | 1048576 | $\approx$ 0,349 ms |
| 30 | 1073741824 | $\approx$ 0,357 s |
| 40 | $\approx 1{,}09951 \cdot 10^{12}$ | $\approx$ 366,5 s |
| 50 | $\approx 1{,}12590 \cdot 10^{15}$ | $\approx$ 4,34 days |
| 60 | $\approx 1{,}15292 \cdot 10^{18}$ | $\approx$ 12,18 years |
| 70 | $\approx 1{,}18059 \cdot 10^{21}$ | $\approx$ 12478,77 years |
| 80 | $\approx 1{,}20893 \cdot 10^{24}$ | $\approx$ 12778262 years |

We convert the minimization of public facilities to the standard set covering problem by means of a threshold, the distance matrix is transformed into the binary reachability matrix.

To validate the results, we present modified data structures and models to guarantee that the results contain all the important service centres. We propose additional columns (or additional constraints in the model) in the reachability matrix instead of traditional weights, which only increase the probability that the particular service would not be omitted.

### A. PROBLEM FORMULATION

Assume that the transport network contains $m$ vertices, that can be used as operating service centres, and $n$ vertices to be served and, for each pair of vertices $v_i$ (considered as service centres) and $v_j$ (serviced vertex), their distance $d_{ij}$ is given and $D_{max}$ is the maximum distance which will be accepted for operation between the service centres and serviced vertices [43].

The aim is to determine which vertices must be used as service centres for each vertex to be covered by at least one of the centres and for the total number of operating centres to be minimal.

*Remark 1:*

1) A condition necessary to solve the task is that all of the serviced vertices are reachable from at least one place where an operating service centre is considered.
2) Serviced vertex $v_j$ is reachable from vertex $v_i$, which is designated as an operating service centre if $d_{ij} \leq D_{max}$. If this inequality is not satisfied, vertex $v_j$ is unreachable from $v_i$.

Here, $a_{ij} = 1$ means that vertex $v_j$ is in a reachable distance to service centre $v_i$, and $a_{ij} = 0$ means that it does not satisfy it. Similarly, $x_i = 1$ means that service centre $i$ is selected, while $x_i = 0$ means that it is not selected.

Then, the set covering problem can be described by the following mathematical model:

Minimise

$$z = \sum_{i=1}^{m} x_i \tag{7}$$

subject to

$$\sum_{i=1}^{m} a_{ij} x_i \geq 1, j = 1, \ldots, n \tag{8}$$

$$x_i \in \{0, 1\}, \quad i = 1, \ldots, m \tag{9}$$

The objective function (7) represents the number of operating centres, constraint (8) means that each serviced vertex is assigned at least one operating service centre. The parameter $D_{max}$ represents a threshold of service reachability.

If the weights of service centres are expressed, then we add coefficient $c_i$ to the Eq. 7. Due to minimization, the greater the weights are, the smaller the coefficients must be. The corresponding mathematical model would change as follows [36], [38], [43]–[45]:

Minimise

$$z = \sum_{i=1}^{m} c_i x_i \tag{10}$$

subject to

$$\sum_{i=1}^{m} a_{ij} x_i \geq 1, \quad j = 1, \ldots, n \tag{11}$$

$$x_i \in \{0, 1\}, \quad i = 1, \ldots, m \tag{12}$$

$$\sum_{i=1}^{m} c_i = 1 \tag{13}$$

*Example 1:* Consider the following *distance matrix* which expresses service centres and customer locations and $D_{max} = 35$. Rows are service centres and columns are customer locations.

serviced vertices (customer locations)

| centres | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 28 | 47 | 124 | 8 | 14 | 74 | 23 | 128 |
| 2 | 91 | 22 | 21 | 36 | 26 | 28 | 32 | 24 |
| 3 | 24 | 58 | 5 | 41 | 139 | 76 | 44 | 78 |
| 4 | 36 | 47 | 168 | 34 | 46 | 36 | 82 | 5 |
| 5 | 11 | 98 | 46 | 103 | 144 | 24 | 21 | 7 |

Depending on $D_{max} = 35$ the reachability matrix [46] would be as follows:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 2 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 3 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 5 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Δ

### B. CONSIDERATION OF SPECIAL CASES

In this section, we will focus on special cases which have to be considered during the implementation process of set covering problem.

These cases show the need to consider specific aspects and modification of input data. Individual cases are displayed using the reachability matrix.

- Unreachable and unnecessary service centre:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 2 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

In the previous matrix, it is shown that the 3rd row contains only 0 values, which means that this row (service centre) could be omitted, thus, it does not cover any customer location.

In addition, we see that the 5th column contains only zeros. That means the threshold distance is too low because that customer is not able to reach any service centre in the selected threshold distance.

Basically, there are three solutions to that problem:

- Removing this customer location from the calculation (if this customer location is not significant)
- Extend threshold distance
- Add another service centre

Adding another service centre would be good to consider in cases of too many customers that cannot reach any service centre. To choose a good location to place a new service centre, the Voronoi diagram decomposition approach and its largest empty circle property may be applied [47]–[50].

- Necessary service centres

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 5 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

In this scenario, we can see that service centre 5 (row 5) cannot be omitted because this is the only service centre available for customer location 3 (column 3).

We could have more than this necessary centre which could not be omitted, e.g., if the service centre has a high weight.

- Service centre as an immediate solution

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

From the previous matrix, we can see that service centre 3 is an immediate solution because it covers all the customer locations. It could be a good hint for the government that location of this service centre could be the best one for a specific service to increase the number of services here, e.g., if the operational capacities of the service centres are not sufficient.

- Service centre included in the solution by a political aspect

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

In a real-world scenario, it may be required that some service centres are included in the final solution even if

the weight of the service is not high. For that reason, we are presenting an additional modification for input data. This is done by adding dummy columns where only a particular service centre is covering this dummy customer location. After this modification, this leads to the special case mentioned in the Necessary service centres item. The example of services which were predetermined as necessary by the political motivation is shown in the matrix above (service centres 2 and 5). We can see that these service centres are the only service centres covering a specific dummy column. However, instead of adding dummy columns, the same effect may be achieved by assigning a values of 1 to the corresponding decision variables directly in the model, here this means $x_2 = 1$ and $x_5 = 1$.

### C. TESTING DATA

In this paper, data from the OR-Library were used. Originally described by Beasley [51], this library is a collection of testing data for combinatorial problems of operations research. For testing matrices with dimensions 200 × 2000 (=> $n = 200$) were used [52].

These data are not in the form of standard matrices but only specify how matrix should be created. They have the below format:

- Number of rows ($m$), number of columns ($n$).
- Weights of columns $c(j), j = 1, \ldots, n$.
- For each row $i$ ($i = 1, \ldots, m$), the number of columns that are covered by this row, and specification of these columns in the ascending order [52].

Below is an example snippet from test data input format.

```
50 500
1 1 1 1 1 1 1 1 1 1 2
2 2 2 2 2 2 2 3 3 3 3 3
3 3 3 3 3 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 6 6 6 6
...
81
4 8 10 11 12 14 15 16 18 23 26 28
45 52 53 54 56 57 60 67 74 80 81 86
106 108 112 118 121 125 126 133 137
176 177 183 190 195 199 201 204 206
250 252 253 261 262 263 267 272 277
320 324 325 327 330 334 335 345 347
376 377 380 386 396 410 415 425 430
469 473 477 478 479 483 496
73
1 2 4 5 13 19 21 28 36 44 45 49 50
59 61 62 67 68 71 73 74 77 82 83 92
102 106 108 118 130 131 134 140 147
159 169 170 172 175 177 178 179 186
229 232 235 239 240 254 258 262 281
317 329 332 333 341 343 346 348 359
404 409 410 412 424 453 472 481 485
...
```

**Algorithm 3** Data Preprocessing

**Input:** inputData from OR-Library;

```
function DataPreprocessing()
    numberOfRows ← read from first line;
    numberOfColumns ← read from first line;
    costs ← readCostOfServices();
    parsedData ← initialise map with (service centre id
  associated with a list of customer locations which this
  service covers)
    while !endOfFile do
        coveringArray = line split by (" ");
        if coveringArray.length == 1 then
            continue                    ▷ Skip informative line
        end if
        for each cover in coveringArray do
            key ← line id (service centre);
            parsedData ← put (key, cover);
        end for
    end while
end function
```

**Output:** *costsCollection* – The list of costs for each service

```
function ReadCostOfServices
    costsCollection ← initialise collection;
    while line ← readLine != null do
        costsArray ← line split by (" ");
        if costsArray.length == 1 then
            break;                      ▷ List of costs ends
        end if
        for each cost in costsArray do
            costsCollection ← add new cost;
        end for
    end while
    return costsCollection;
end function
```

These test data were preprocessed to obtain the necessary data format. For that reason, we create the model introduced in Alg. 3.

### D. CONSTRUCTION OF INDIVIDUAL AND POPULATION

After data preprocessing in Alg. 3, fields *parsedData* and *costs* contain information about service centre coverage and cost, respectively.

With this data we are able to create *Individual* objects which contain the following attributes:

- *chromosome* (represented as array of Boolean values)
- *fitnessValue*
- *isCustomerLocCovered*
- *chromosomeCovering*
- *necessaryRows*

The chromosome bit is 0 if the corresponding service centre is not included, and 1 if it is included in the solution. The chromosome must be instantiated with services which are necessary, from the other perspective

**Algorithm 4** Repair Operator in Minimization of Covering Services

**Input:** $SC = \{1, \ldots, m\}$ = the set of all service centres (rows); $CL = \{1, \ldots, n\}$ = the set of all customer locations (columns); $SCS$ = the set of service centres in a solution; $UCL$ = the set of uncovered customer locations; $w_i$ = the number of service centres that cover customer location $j, j \in CL$ in $S$; $\alpha_j\{i \in SC \ |a_{ij} = 1\}$ = the set of service centres that cover customer location $j, j \in CL$; $\beta_i = \{j \in CL|a_{ij} = 1\}$ = the set of customer locations that are covered by service centre $i, i \in SC$;

1: **function** repairOperator()
2: initialise $w_j = |S \cap \alpha_j|, \forall j \in CL$;
3: initialise $UCL = \{j|w_j = 0, \forall j \in CL\}$;
4: **for all** customer locations $j$ in $UCL$ **do**
5: find the first service centre $i$ in $\alpha_j$ that minimizes
6: $1 / |UCL \cap \beta_i|$;
7: $SCS \leftarrow SCS + i$;
8: $w_j \leftarrow w_j + 1, \forall j \in \beta_i$;
9: $UCL \leftarrow UCL - \beta_i$;
10: **end for**
11: **for all** service centre $i$ in $SCS$ **do**
12: **if** $w_j \geq 2, \forall_j \in \beta_i$ **then**
13: $SCS \leftarrow SCS - i$;
14: $w_j \leftarrow w_j - 1, \forall_j \in \beta_i$;
15: **end if**
16: **end for**
17: **end function** ▷ {The set of service centres ($SCS$) is now a feasible solution without redundant service centres}

unnecessary services could be omitted (see special cases in Section III-B). The fitness value is computed based on Eq. 10. Field *isCustomerLocCovered* is computed based on the *chromosomeCovering* list indicating which services are covered by this list of service centres. The easiest way to check it is to test if *chromosomeCovering* list size is the same as the number of columns in the matrix. If the presented service centres in *chromosome* field do not cover all the customer locations, we have to apply a repair operator. The algorithm for the repair operator is defined in Alg. 4. This algorithm has to be used after each generation of new individuals due to the fact that operations like crossover could generate a non-feasible solution. Firstly, the algorithm identifies uncovered customer locations. Since the minimization of covering services is a minimization problem in the first *for* cycle, service centres with low-cost ratio are considered in the first place and for the second *for* cycle, service centres with high cost are not included whenever possible. And finally, the field *necessaryRows* contains the list of service centres that will be appended to the final solution.

The population object contains the following fields:

- *populationSize*
- *individuals*
- *bestFitness*
- *matrix*

**Algorithm 5** Generation of Population

**Input:** *pop_size* ← individuals in the population;

1: **function** InitializePopulation()
2: *bestFitness* ← real number max value;
3: *individuals* ← initialise the set of all individuals;
4: **for** i ← 0 **to** *pop_size* **do**
5: *ind* ← new Individual;
6: *ind.chromosome* ← gen. random bool array;
7: check chromosome covering;
8: **if** *isCustomerLocCovered* == **false then**
9: apply repair operator defined in Alg. 4;
10: **end if**
11: evaluate *ind*; ▷ Assign fitness value
12: *individuals* ← add *ind*;
13: **end for**
14: *bestFitness* ← compute individual with the best fitness;
15: **end function**

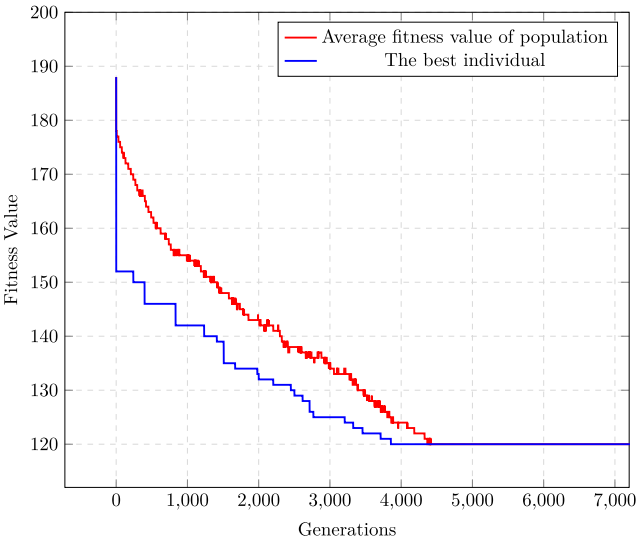



**FIGURE 2.** Population size 200, uniform crossover, roulette wheel selection.

Population object contains a set of *individuals* and information about service centres and customer locations represented by the given matrix. The set of individuals is initially created with Algorithm 5.

## IV. EXPERIMENTS AND RESULTS

To prove the advantage of the proposed enhancement, we provide a set of tests to verify that this algorithm is a viable alternative to standard methods. For each of these tests, 1000 measurements were performed.

Fig. 2 depicts the best run using the standard GA implementation, we see that, after 4000 generations, the algorithm got stuck in a local minimum with a fitness value of 120.

When the local minimum is reached, i.e. additional generations do not produce improvements in fitness value, we are able to use our proposed *war* operator. In this particular case, the operator was used after 5000 generations and is

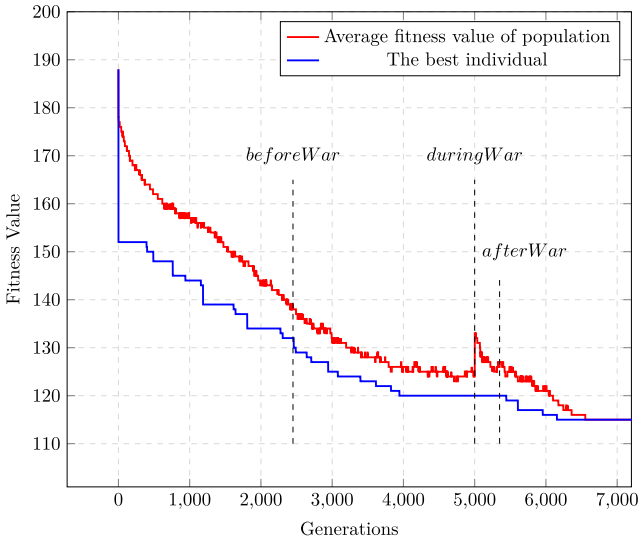



**FIGURE 3.** Population size 200, uniform crossover, roulette wheel selection, *war* operator applied.

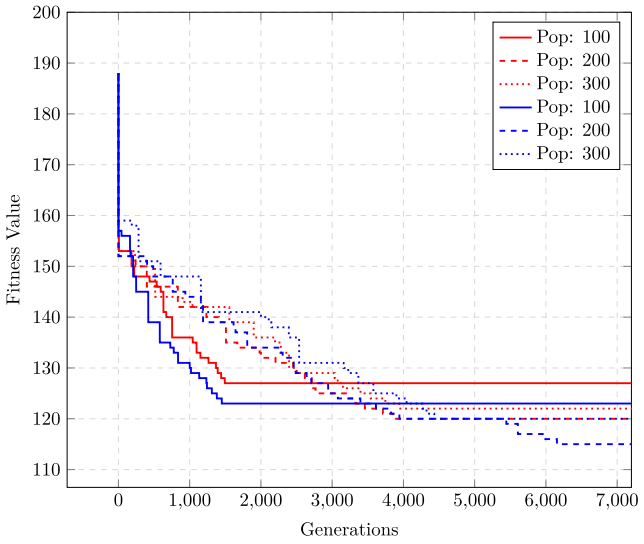



**FIGURE 4.** Testing the stability of standard GA (red lines) and enhanced GA (blue lines) using different population size.

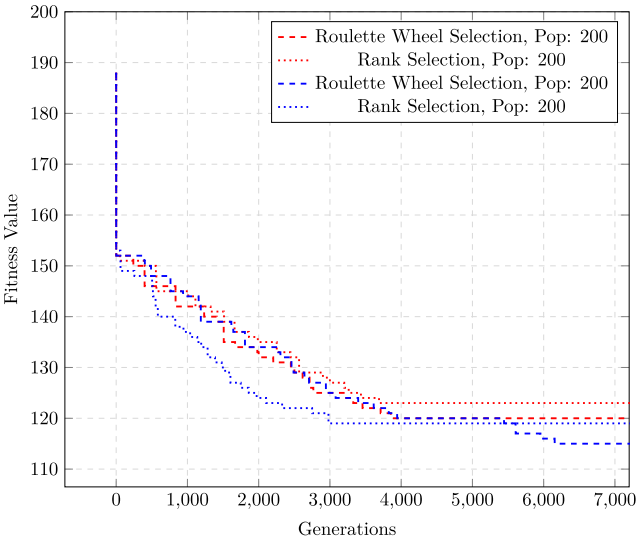



**FIGURE 5.** Testing stability of standard GA (red lines) and enhanced GA (blue lines) using different selection operators.

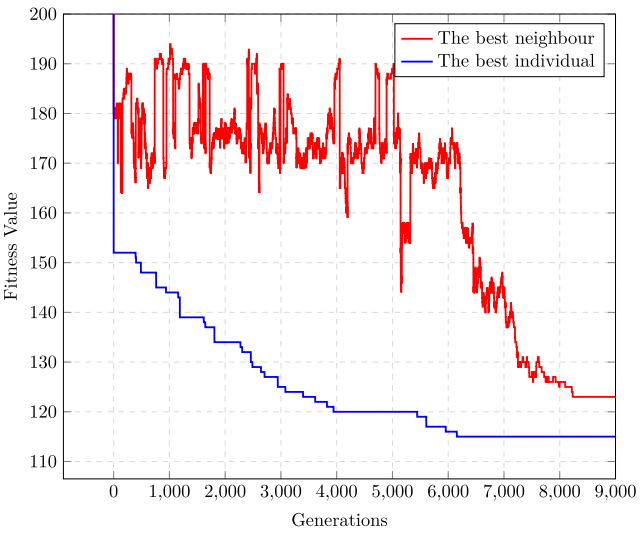



**FIGURE 6.** Results obtained by the simulated annealing algorithm.

**TABLE 3.** Summarized results on OR-Beasley data.

| | without *war* operator | using *war* operator |
|---|---|---|
| rank selection | 123 | 119 |
| roulette wheel selection | 120 | 115 |
| pop 100 | 127 | 123 |
| pop 200 | 120 | 115 |
| pop 300 | 122 | 120 |

responsible for a decrease in the average fitness level for subsequent generations (see Fig. 3). Since the fitness value of the average population has been increased (worsened), it generates better solutions in the next generations (this is the reason why also the average fitness value of the population is shown in Fig. 2 and Fig. 3). Using *war* operator, the best result reaches the fitness value 115.

To test the robustness of the algorithm, we provide comparisons for different parameter settings. The proposed enhancement versus standard GA is shown in Fig. 4 in cases of the population size being changed to 100, 200, and 300. The red lines are the results with standard GA implementation and the blue lines show the proposed GA enhancement.

For further comparisons, two selection types are chosen: (i) rank selection and (ii) tournament selection (implemented based on literature [15], [22], [23], [23]–[25], [53]). The results are shown in Fig. 5. We see that the enhanced version yields better results for both selection types.

For reference comparisons, we decided to implement another meta-heuristic algorithm, namely the simulated annealing algorithm, which, as with genetic algorithms, is widely used at present [54]–[58]. To implement simulated annealing, we only replace the algorithm part because the representation of Neighbor is the same as the representation of the Individual in GA. The result is shown in Fig. 6.

Tab. 3 summarizes the results of the provided experiments.

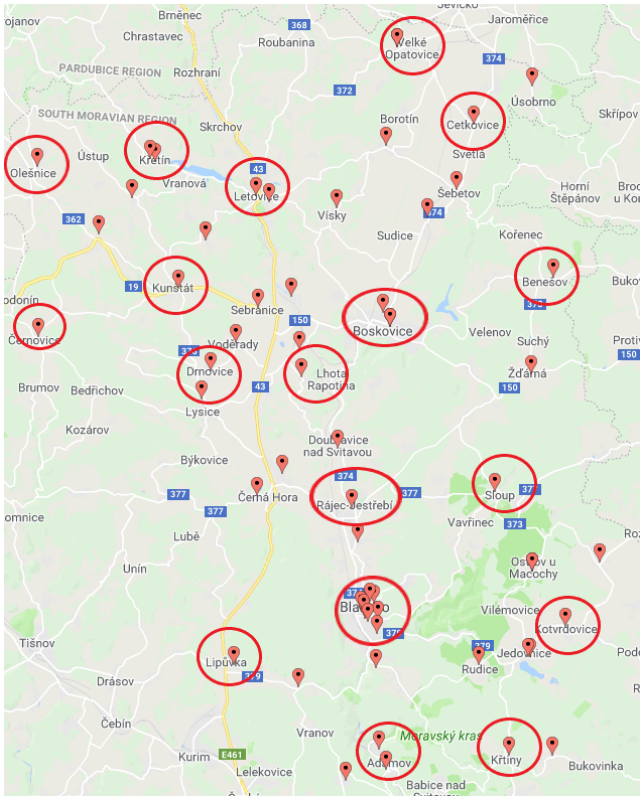



**FIGURE 7.** Elementary Schools in Blansko district.

## V. REAL WORLD SCENARIO

The first big issue of the real-world scenario is how to retrieve the data. We need to download the addresses for services and customer locations from particular office databases or from the sources available on the Internet. When filling in these matrix headers for rows (service centres) and columns (customer locations), the best method to find the distance between them is to create a parser for web pages (Open Street Maps) to retrieve the distances between each service centre and customer location in the matrix. For that reason, we created a script in Java language with Selenium framework [59] to parse information about the route distance between service centres and customer locations assigned to each service centre. For our scenario, it is the matrix with dimensions $54 \times 112$ and obviously, filling that matrix manually would be very tedious.

The trained proposed algorithm was used in cooperation with Blansko District to reduce the number of schools. The political requirement was that we could not omit schools which are located in the cities having some strategic importance (Adamov, Blansko, Boskovice, Kunštát, Letovice, Olešnice, Rájec-Jestřebí, Velké Opatovice), thus, these schools are preselected as necessary services (part of the solution) similar to Sec. III-B on special cases.

The selected distance for each customer location to the service centres have to be within a 15 km route-distance. For that requirement, the number of schools was reduced from 54 to 30. The schools which were not removed are highlighted by red circles in Fig. 7. It is clear that these service centres are effectively spread all over the district. If the operation capacity of selected schools would be exceeded, it will be necessary to extend the capacity of the schools. If it is not possible to extend the capacity of the school, we expect additional schools to be added in a reachable distance. Additional schools would be selected by their service weight (parameter $c$ in Eq. 10). The proposed solution is used as a skeleton for the government, they still need to include additional political aspects, which are out of the scope of this paper.

**TABLE 4. Summarized results from performed measurements on real-world data.**

| | number of schools | |
|---|---|---|
| | without $war$ operator | using $war$ operator |
| rank selection | 34 | 31 |
| roulette wheel selection | 33 | 30 |
| pop 100 | 35 | 31 |
| pop 200 | 33 | 30 |
| pop 300 | 34 | 31 |

We test the performance of the GA on real-world data for the same initial model configurations as considered in the Beasley OR-library case (Tab. 3). The results are shown in Tab. 4, and we see that the best parameterization is the same as with the Beasley OR-library benchmark data.

## VI. CONCLUSION

In this paper, we introduced a model for the minimization of public facilities. This problem is $\mathcal{NP}$-hard with a complexity of $\mathcal{O}(2^n)$ and thus, in order to solve larger problems, arbitrary meta-heuristics have to be used. We propose a modified GA algorithm based on a novel *war* operator. The model is validated on a standard OR-library benchmark dataset against vanilla GA, and one of other metaheuristics from the family of naturally inspired algorithms, that is, simulated annealing. The results prove that our extension manages to outperform traditional techniques in terms of the fitness value while allowing for relatively simple and effective implementation compared to other metaheuristic-based alternatives. Furthermore, we perceive its potential for other $\mathcal{NP}$-hard problems for the case of the previously used model of the genetic algorithm not yielding satisfactory results.

Lastly, the viability of the proposed algorithm is showcased on a real case-study, minimization of school network in a Central European region (see Fig. 7).

## ACKNOWLEDGMENTS

Research described in this paper was financed by the National Sustainability Program under Grant LO1401. For the research, infrastructure of the SIX Center was used.

## REFERENCES

[1] S. G. Devi, K. Selvam, and S. P. Rajagopalan, "An abstract to calculate big o factors of time and space complexity of machine code," in *Proc. Int. Conf. Sustain. Energy Intell. Syst. (SEISCON)*, Jul. 2011, pp. 844–847.

[2] A. N. Rybalov, "On the P vs NP problem over reals with integer oracle," in *Proc. Dyn. Syst., Mech. Machines*, Nov. 2016, pp. 1–4.

[3] S. Mandal, P. Mallick, D. Mandal, R. Kar, and S. P. Ghoshal, "Optimal FIR band pass filter design using novel particle swarm optimization algorithm," in *Proc. IEEE Symp. Humanities, Sci. Eng. Res. (SHUSER)*, Jun. 2012, pp. 141–146.

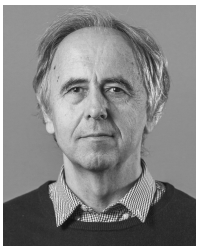
[4] J. Chen and S. Chong, "A research for scheduling method of rehabilitation medical resource for smart traditional Chinese medicine based on genetic algorithm and its application on the treatment for dysphagia," *China Med. Equip.*, vol. 14, no. 3, pp. 6–12, 2017.
[5] A. H. Karami and M. Hasanzadeh, "An adaptive genetic algorithm for robot motion planning in 2D complex environments," *Comput. Elect. Eng.*, vol. 43, pp. 317–329, Apr. 2015.
[6] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[7] S. Wang *et al.*, "A multi-approaches-guided genetic algorithm with application to operon prediction," *Artif. Intell. Med.*, vol. 41, no. 2, pp. 151–159, 2007.
[8] Eurostat. (2018). *Being Young in Europe Today—Demographic Trends*. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php/Being_young_in_Europe_today_-_demographic_trends
[9] W. H. Organization. (2018). *Number of Hospitals*. [Online]. Available: https://gateway.euro.who.int/en/indicators/hfa_471-5011-number-of-hospitals
[10] G. Pascal and I. Notarangelo. (2018). *Hospitals in Europe: Healthcare Data*. [Online]. Available: http://www.hospitalhealthcare.com/hope/hospitals-europe-healthcare-data
[11] S. Agathos and E. Papapetrou, "On the set cover problem for broadcasting in wireless ad hoc networks," *IEEE Commun. Lett*, vol. 17, no. 11, pp. 2192–2195, Nov. 2013.
[12] M. Owais, M. K. Osman, and G. Moussa, "Multi-objective transit route network design as set covering problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 670–679, Mar. 2016.
[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
[14] D. Simon, *Evolutionary Optimization Algorithms*. Hoboken, NJ, USA: Wiley, 2013.
[15] D. E. Goldberg, *Genetic Algorithms*. London, U.K.: Dorling Kindersley, 2008.
[16] L. Cedeño-Rodríguez, J. P. Ayala, S. Guerra-Jiménez, and A. Fernandez-Correa, "Optimizing parameters for a dynamic model of high-frequency hid lamps using genetic algorithms," *DYNA*, vol. 82, no. 189, pp. 182–188, 2015.
[17] H.-C. Chang, Y.-P. Chen, T.-K. Liu, and J.-H. Chou, "Solving the flexible job shop scheduling problem with makespan optimization by using a hybrid taguchi-genetic algorithm," *IEEE Access*, vol. 3, pp. 1740–1754, 2015.
[18] Z. Wang, J. Li, K. Fan, W. Ma, and H. Lei, "Prediction method for low speed characteristics of compressor based on modified similarity theory with genetic algorithm," *IEEE Access*, vol. 6, pp. 36834–36839, 2018.
[19] N. Hou, F. He, Y. Zhou, Y. Chen, and X. Yan, "A parallel genetic algorithm with dispersion correction for HW/SW partitioning on multi-core CPU and many-core GPU," *IEEE Access*, vol. 6, pp. 883–898, 2018.
[20] P. Bezák, "Using motion planning and genetic algorithms in movement optimization of industrial robots," Univ. Ilmenau, Ilmenau, Germany, Tech. Rep. 5, 2012.
[21] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
[22] C. Echegoyen, A. Mendiburu, R. Santana, and J. A. Lozano, "On the taxonomy of optimization problems under estimation of distribution algorithms," *Evol. Comput.*, vol. 21, no. 3, pp. 471–495, 2012.
[23] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.
[24] W. Song and C. Huang, "Mining high utility itemsets using bio-inspired algorithms: A diverse optimal value framework," *IEEE Access*, vol. 6, pp. 19568–19582, 2018.
[25] S. Karakatič and V. Podgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem," *Appl. Soft Comput.*, vol. 27, pp. 519–532, Feb. 2015.
[26] R. Poli, M. Salvaris, and C. Cinel, "Evolution of an effective brain-computer interface mouse via genetic programming with adaptive tarpeian bloat control," in *Genetic Programming Theory and Practice IX*. New York, NY, USA: Springer, 2011, pp. 77–95.
[27] M. Kommenda, G. Kronberger, M. Affenzeller, S. M. Winkler, and B. Burlacu, "Evolving simple symbolic regression models by multi-objective genetic programming," in *Genetic Programming Theory and Practice XIII*. Cham, Switzerland: Springer, 2016, pp. 1–19.
[28] M. Kurdi, "A new hybrid island model genetic algorithm for job shop scheduling problem," *Comput. Ind. Eng.*, vol. 88, pp. 273–283, Oct. 2015.
[29] M. Kurdi, "An effective new island model genetic algorithm for job shop scheduling problem," *Comput. Oper. Res.*, vol. 67, pp. 132–142, Mar. 2016.
[30] A. Davidson, D. Tarjan, M. Garland, and J. D. Owens, "Efficient parallel merge sort for fixed and variable length keys," in *Proc. Innov. Parallel Comput. (InPar)*, May 2012, pp. 1–9.
[31] N. Faujdar and S. P. Ghrera, "Performance evaluation of merge and quick sort using GPU computing with CUDA," *Int. J. Appl. Eng. Res.*, vol. 10, no. 18, pp. 39315–39319, 2015.
[32] F. Hu and G.-Y. Wang, "Analysis of the complexity of quick sort for two dimension table," *Chin. J. Comput.-Chin. Ed.*, vol. 30, no. 6, p. 963, 2007.
[33] J. Bloch, *Effective Java*. Chennai, India: Pearson, 2016.
[34] M. Avraham, D. Inbar, and Z. Paz, "SDRAM memory device with an embedded NAND flash controller," U.S. Patent 7 752 380 B2, Jul. 6, 2010.
[35] J. Aslund, "Systems and methods for configuring an electronic device," U.S. Patent 9 380 059 B2, Jun. 28, 2016.
[36] B. Crawford, R. Soto, R. Cuesta, and F. Paredes, "Application of the artificial bee colony algorithm for solving the set covering problem," *Sci. World J.*, vol. 2014, Apr. 2014, Art. no. 189164.
[37] J. E. Beasley, "An algorithm for set covering problem," *Eur. J. Oper. Res.*, vol. 31, no. 1, pp. 85–93, 1987.
[38] A. A. Coco, A. C. Santos, and T. F. Noronha, "Senario-based heuristics with path-relinking for the robust set covering problem," in *Proc. 11th Metaheuristics Int. Conf. (MIC)*, 2015, pp. 1–8.
[39] R. Soto *et al.*, "Solving the non-unicost set covering problem by using cuckoo search and black hole optimization," *Natural Comput.*, vol. 16, no. 2, pp. 213–229, 2017.
[40] B. Klocker and G. R. Raidl, "Solving a weighted set covering problem for improving algorithms for cutting stock problems with setup costs by solution merging," in *Proc. Int. Conf. Comput. Aided Syst. Theory*. Cham, Switzerland: Springer, 2017, pp. 355–363.
[41] B. Crawford, R. Soto, M. Olivares-Suárez, and F. Paredes, "A binary firefly algorithm for the set covering problem," in *Modern Trends and Techniques in Computer Science*. Cham, Switzerland: Springer, 2014, pp. 65–73.
[42] J. Huang *et al.*, "Site selection of nature reserve based on the self-learning tabu search algorithm with space-ecology set covering problem: An example from daiyun mountain, southeast china," *Chin. J. Appl. Ecology*, vol. 28, no. 1, pp. 219–230, 2017.
[43] M. Šeda and P. Šeda, "A minimisation of network covering services in a threshold distance," in *Mendel* (Advances in Intelligent Systems and Computing). Heidelberg, Germany: Springer, 2015.
[44] J. E. Beasley and P. C. Chu, "A genetic algorithm for the set covering problem," *Eur. J. Oper. Res.*, vol. 94, no. 2, pp. 392–404, Oct. 1996.
[45] J. M. Lanza-Gutierrez, B. Crawford, R. Soto, N. Berrios, J. A. Gomez-Pulido, and F. Paredes, "Analyzing the effects of binarization techniques when solving the set covering problem through swarm optimization," *Expert Syst. Appl.*, vol. 70, pp. 67–82, Mar. 2017.
[46] W. Ji, L. Wang, A. Haghighi, M. Givehchi, and X. Liu, "A reachability based approach for machining feature sequencing," *J. Manuf. Syst.*, vol. 40, pp. 96–104, Jul. 2016.
[47] H. Edelsbrunner, A. Glazyrin, O. R. Musin, and A. Nikitenko, "The Voronoi functional is maximized by the Delaunay triangulation in the plane," *Combinatorica*, vol. 37, no. 5, pp. 887–910, 2017.
[48] J. D. Hyman, C. W. Gable, S. L. Painter, and N. Makedonska, "Conforming delaunay triangulation of stochastically generated three dimensional discrete fracture networks: A feature rejection algorithm for meshing strategy," *SIAM J. Sci. Comput.*, vol. 36, no. 4, pp. A1871–A1894, 2014.
[49] H. Si, "TetGen, a delaunay-based quality tetrahedral mesh generator," *ACM Trans. Math. Softw.*, vol. 41, no. 2, p. 11, 2015.
[50] T. Su, W. Wang, Z. Lv, W. Wu, and X. Li, "Rapid Delaunay triangulation for randomly distributed point cloud data using adaptive Hilbert curve," *Comput. Graph.*, vol. 54, pp. 65–74, Feb. 2016.
[51] J. E. Beasley, "OR-library: Distributing test problems by electronic mail," *J. Oper. Res. Soc.*, vol. 41, no. 11, pp. 1069–1072, Nov. 1990.
[52] J. Beasley. (2018). *Set Covering*. [Online]. Available: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/scpinfo.html
[53] J. Cohoon, J. Kairo, and J. Lienig, *Evolutionary Algorithms for the Physical Design of VLSI Circuits*. Berlin, Germany: Springer, 2003.
[54] S. Deng, C. Yuan, J. Yang, and A. Zhou, "Distributed mining for content filtering function based on simulated annealing and gene expression programming in active distribution network.," *IEEE Access*, vol. 5, pp. 2319–2328, 2017.
[55] K. G. Krishnakumar and E. T. Fredrik, "Ant colony optimization (ACO) with simulated annealing approach (SAA) for optimized multipath operation in vanet," Karpagam Acad. Higher Educ., Tamil Nadu, India, Tech. Rep. 10, 2018.
[56] M. Mann, O. P. Sangwan, P. Tomar, and S. Singh, "Automatic goal-oriented test data generation using a genetic algorithm and simulated annealing," in *Proc. 6th Int. Conf.-Cloud Syst. Big Data Eng. (Confluence)*, Jan. 2016, pp. 83–87.

[57] C. Wang, D. Mu, F. Zhao, and J. W. Sutherland, "A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows," *Comput. Ind. Eng.*, vol. 83, pp. 111–122, May 2015.
[58] F. Y. Vincent and S.-Y. Lin, "A simulated annealing heuristic for the open location-routing problem," *Comput. Oper. Res.*, vol. 62, pp. 184–196, Oct. 2015.
[59] A. M. F. V. de Castro, G. A. Macedo, E. F. Collins, and A. C. Dias-Neto, "Extension of selenium RC tool to perform automated testing with databases in Web applications," in *Proc. 8th Int. Workshop Automat. Softw. Test*, May 2013, pp. 125–131.

**PAVEL SEDA** received the M.Sc. degree in electrical engineering from the Faculty of Electrical Engineering and Communication, Brno University of Technology (BUT), Czech Republic, in 2017, and the M.Sc. degree in applied informatics from Masaryk University, in 2018. He is currently pursuing the Ph.D. degree with BUT. From 2014 to 2018, he was a Java Developer with IBM Czech Republic, in one of its developer divisions. His research interests include different areas in artificial intelligence especially with the focus on heuristic algorithms and in the area of software architectures including microservices architecture.

**MICHAEL MARK** received the bachelor's degree in mathematics from Université Pierre et Marie Curie, Paris, and the M.Sc. degree in computational finance from the University of Leicester. He is currently pursuing the Ph.D. degree with the Chair of Operations, Economics and Strategy. From 2015 to 2016, he was a Quantitative Risk Analyst with EY, London, in one of its advisory divisions. His current research interest includes credit collections. His other research interests include stochastic systems and decisions under uncertainty.

**KUAN-WU SU** received the B.S. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, and the M.S. degree from the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, where he is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering. He was with Trustable Ltd., Taiwan, as a System Engineer, from 2003 to 2005, and held a position as a Special Assistant to the President with eMorse Ltd., Taiwan, from 2005 to 2009, with experiences in telecommunication industry and financial and intellectual property management. He had been a Visiting Junior Fellow with the Tokyo Institute of Technology, in 2017. His research interests include human-centric computing, mobile UI/UX, sensor networks, cloud/edge computing, music information retrieval, social networking, and machine learning along with neural networks and evolutionary algorithm.



**MILOS SEDA** received the M.Sc. and Ph.D. degrees in cybernetics from the Brno University of Technology and the M.Sc. degree in mathematical computer science from Masaryk University. From 1976 to 1986, he was a System Designer with První brněnská strojírna. He is currently a Professor and a Senior Researcher with the Institute of Automation and Computer Science, BUT. His current research interests include meta-heuristic algorithms, stochastic systems, and graph algorithms. He has (co)-authored over 150 research works on those topics.

**JIRI HOSEK** received the M.S. and Ph.D. degrees in electrical engineering from the Faculty of Electrical Engineering and Communication, Brno University of Technology (BUT), Czech Republic, in 2007 and 2011, respectively, where he is currently an Associate Professor and the Deputy Vice Head for research and development and international relations with the Department of Telecommunications, BUT. He has co-authored over 80 research works on networking technologies, wireless communications, quality of service, user experience, and the Internet of Things (IoT) applications. His research work has concentrated on industry-oriented research and development projects in the area of future mobile networks, the IoT, and home automation services.

**JENQ-SHIOU LEU** received the B.S. degree in mathematics and the M.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1991 and 1993, respectively, and the Ph.D. degree on a part-time basis in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2006. He was an R&D Engineer with Rising Star Technology, Taiwan, from 1995 to 1997. He was an Assistant Manager with Mobitai Communications and Taiwan Mobile, from 1997 to 2007. In 2007, he joined the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei, as an Assistant Professor, where he was an Associate Professor, from 2011 to 2014, and has been a Professor since 2014. He is currently the Chairperson with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, and the Chair of the Graduate Institute of Electro-Optical Engineering. His research interests include heterogeneous networks and mobile services over heterogeneous networks. He has been a Senior Member of the IEEE, since 2017.