Multiobjective Optimization of Complete Coverage and Path Planning for Emergency Response by UAVs in Disaster Areas

Krzysztof Trojanowski[1][0000−0001−9009−049X], Artur Mikitiuk[1][0000−0001−5038−1196], Jakub Grzeszczak[1][0000−0001−8679−9076], and Frédéric Guinand[2,1][0000−0002−7915−2781]

[1] Cardinal Stefan Wyszyński University in Warsaw, Poland
{k.trojanowski,a.mikitiuk,jakub.grzeszczak}@uksw.edu.pl
[2] Normandy University of Le Havre, Le Havre, France
frederic.guinand@univ-lehavre.fr

Abstract. Complete Coverage and Path Planning methods operate on many models depending on initial constraints and user demands. In this case, we optimize paths for a set of UAVs in the disaster area divided into rectangular regions of different sizes and priorities representing the expected number of victims. Paths maximize the number of victims localized in the first minutes of the UAVs' operation and minimize the entire operation makespan. The problem belongs to the domain of multiobjective optimization; therefore, we apply the Strength Pareto Evolutionary Algorithm 2, which is equipped with several problem-specific perturbation operators. In the experimental part, we use SPEA2 to four selected test cases from a TCG-CCPP generator powered by actual data on residents in selected regions in Poland published by Statistics Poland.

Keywords: Multiobjective Optimization, Coverage Path Planning, Unmanned Aerial Vehicles, Terrain Coverage, Heuristic Optimization

1 Introduction

In recent times, UAVs have found an increasing number of applications, both civil, like agricultural, scientific, or emergency, and military, like surveillance or battlefield activity. In the presented case, we develop applications for the first reconnaissance in the disaster areas caused by floods or earthquakes. When base transceiver stations (BTSs) are out of order, and ground communication tracks and roads have been flooded or destroyed, victims cannot communicate about their location, state, and needs. In this case, UAVs represent the first, immediate response to the problem of reconnaissance and communication delivery. The UAV application has many aspects that must be subject to analysis, like communication, path planning, or emergency management. UAVs' mobility features and availability or access to additional information to predict the localization of victims in the path-planning process significantly impact the rescue action's plan and effectiveness.

In this research, we develop a path-planning procedure for a team of simultaneously operating UAVs. The paths are optimized to satisfy the two objectives: to get information about the condition and location of all persons in the disaster area and to minimize the time to find the victims demanding immediate help. UAVs are equipped with Mobile Base Stations (MBSs). Turned-on mobile phones of the victims help UAVs discover phones' precise GPS coordinates and owners' identity through phone or IMEI numbers. Additionally, we consider the terrain, which is not uniform in terms of population; selected regions, like villages and cities, have more chances to have people present than others, like farmlands or wastelands. Therefore, the shape of UAVs' paths matters for the average waiting time for help. Time is a critical factor in helping victims. The first minutes following an accident are the most important in emergency management.

We continue our earlier research described in [6] and [13]. In those publications, we formulated a model of the disaster area and proposed a Test Case Generator and a path-planning method for a team of UAVs. The method implements the Local Search approach but operates on a new model-specific solution representation and uses a number of new perturbation operators. We formulated a function expressing the number of victims localized in the first minutes of the UAVs' work and maximized it in the experimental research. In this paper the main contribution is a new two-stage optimization and computer simulations:

1. In the first stage, we generate an initial population of solutions. We make them randomly or employ the earlier proposed Local Search. In this method, a newly generated solution is better than the current one when its overall operation makespan is shorter. When they have equal makespans, the one with fewer victims missed in the first minutes of the UAVs' operation is better. This second criterion is a reformulated function calculating the number of victims localized in the first minutes of the UAVs' work.
2. In the second stage, we apply one of the multiobjective optimization heuristics, The Strength Pareto Evolutionary Algorithm 2 (SPEA2), using the initial population from the previous stage. Two criteria are minimized: the overall operation makespan and the number of victims missed in the first minutes of the UAVs' operation. Both stages use the same previously proposed representation of solutions, but in the latter one, we propose new perturbation operators.
3. In the experimental part, we used an earlier proposed benchmark from the Test Case Generator. The nondominated sets from SPEA2 vary depending on the initial population and settings of evolutionary parameters. We find their most beneficial settings and indicate the pros and cons of multi-criteria optimization compared to the single-criteria approach.

The paper consists of seven sections. Section 2 presents the model of the disaster area. Section 3 discusses the representation of the set of paths for a team of UAVs. Section 4 describes two solution evaluation criteria defining the objective space. Section 5 presents SPEA2 and problem-specific operators. The experimental part of the research is described in Section 6. Finally, Section 7 concludes the paper.

2 Representation of A Disaster Area Map

A team of UAVs operates over a rectangle disaster area. We aim to generate UAV paths that cover the entire area and can cross but should not overlap. The precise location of the disaster victims is unknown. However, some estimations about their density in selected places exist. Therefore, we divide the area into a set of non-overlapping regions covering the entire area and having priorities representing chances for the presence of victims. The regions are rectangles of different sizes; every place within a region has the same population density. In the presented research, the region's priority depends on the population density in the region; however, in general, the priority may also originate from natural disaster circumstances like the epicenter's location and the strength of tremors in the case of quakes or hydrographical conditions in the case of floods. Every region must belong to one of the paths because we have to localize all victims.

The sub-areas with a similar population density are approximated by rectangle regions since path generation algorithms satisfying UAV mobility features exist for such types of regions [2,4,8]. Models dividing the area into regions were proposed in numerous approaches to the Coverage Path Planning problem [1,7,9,10,11,12]. However, none have regions with assigned priorities for the coverage order. For the experimental research, we used the Test Case Generator for Problems of Complete Coverage and Path Planning (TCG-CCPP) [6], which proceeds data on residents in a 1-kilometer grid from the "2021 Population and Housing Census in Poland" published by Statistics Poland[3]. The generator transforms an input grid of size n by n squares with population density estimated from the census into an area divided into rectangular regions with different population density levels.

A region has two attributes: a surface area and an expected number of citizens/victims (ENV). Still, when we divide the region's ENV by the area, we derive the third attribute — population density. ENV for a path equals the sum of ENVs for the regions in this path, and ENV for the disaster area equals the sum of ENVs for all paths.

As mentioned earlier, regions from TCG-CCPP are rectangular. Hence, they are easy to cover by UAV flight trajectories. Therefore, we do not build precise paths within these regions. We rather optimize the order of visiting regions, where neighbor regions in the path do not necessarily have to be adjacent on the terrain map. Thus, the given path-planning problem is a combinatorial one.

3 The Problem Solution — UAVs' Paths

When a disaster occurs, all paths for u UAVs must be fully defined before the UAVs start operating over the disaster area. Hence, the operational area model must already exist and be available for computations. Briefly, the area must be divided into a set of convex regions concerning their priorities, and we need

[3] Geostatistics Portal, `https://geo.stat.gov.pl`, the date of access: Dec 28, 2022

to know the number of available UAVs, their mobility features, and starting locations. A central computational unit generates paths when the rescue teams with UAVs have already been dispatched and are on their way, or even earlier, during periodic safety inspections and tests.

We need paths for all UAVs, so a single solution consists of u paths containing lists of regions' IDs. In the sum of paths, no ID can be omitted to ensure the finding of all victims. Moreover, the sum must contain no duplicated IDs to avoid redundant workloads during the paths' execution. The solution execution is called a round. Managing distances between UAVs is unnecessary since the paths are generated offline, so no interactive navigation occurs.

A region's coverage time for a UAV equals a surface unit's coverage time multiplied by the surface area. The traversing time equals the distance to traverse divided by the UAV's cruising velocity. The surface unit's coverage time and the UAV's cruising velocity are the problem's parameters that are constant over the entire round. Each path execution time considers the sum of the overall time of regions' coverage and the time necessary to traverse between them. For any two consecutive regions in the path being adjacent to each other in the terrain, traverse time equals zero. Otherwise, the traverse time equals the execution time for a UAV's direct flight in a straight line between the two closest points of the region's boundaries. It is important to stress that the location of the region's entry and exit points does not change the number of found victims over time.

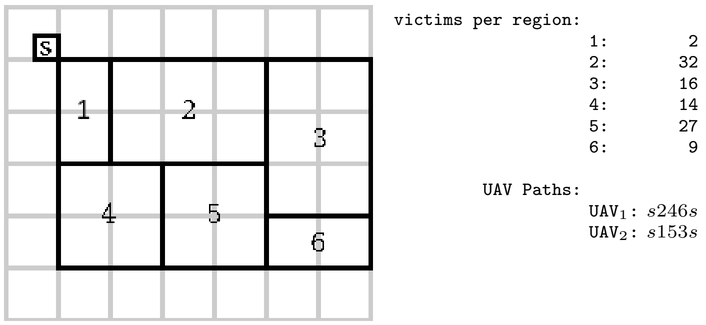
4 Evaluation of UAVs' Paths

We use a model where the area has the size of $n \times n$ one-kilometer base units and is fully covered by rectangle regions whose side lengths are also expressed in base units. UAV paths consist of two types of tasks: (1) a region covering and (2) traversing from one region to another. Execution of the region coverage takes time proportional to the region's area in square kilometers multiplied by a unit coverage time T_{sc}. During the region's coverage, the number of localized victims grows linearly. When UAV traverses from one region to another, no victims are found. The traverse path is a straight line, and its execution takes time inversely proportional to the UAV's cruising velocity V_{cr}.

A model of an example problem and its example solution S are presented in Figure 1. The left part of the figure shows the disaster area divided into regions, whereas the right part shows example paths for a team of two UAVs. Both UAVs start from the position s and execute paths $s246s$ and $s153s$, respectively, where digits represent the region's IDs. Figure 2 shows charts of the number of already found victims over time for the two parts of S, that is S_1 executing $s246s$ and S_2 — $s153s$, and example values of $T_{\text{sc}} = 2$ and $V_{\text{cr}} = 1$. The chart of the number of already localized victims over time for S is the sum of charts calculated for each UAV. Detailed discussion on the charts can be found in [13].

We use two criteria to evaluate solutions. The first one equals the makespan of the round, that is, the path execution time of the UAV that finished last:

$$f_1(\mathbf{x}) = \max(T_i) \quad \text{where } i \in \{1, \dots, u\} \tag{1}$$



victims per region:

1:	2
2:	32
3:	16
4:	14
5:	27
6:	9

UAV Paths:
UAV_1: $s246s$
UAV_2: $s153s$

Fig. 1. An example test case and its solution: layout of an area divided into regions (on the left) and paths for two working UAVs having a starting point s in the corner of the area (on the right); extracted from [13]

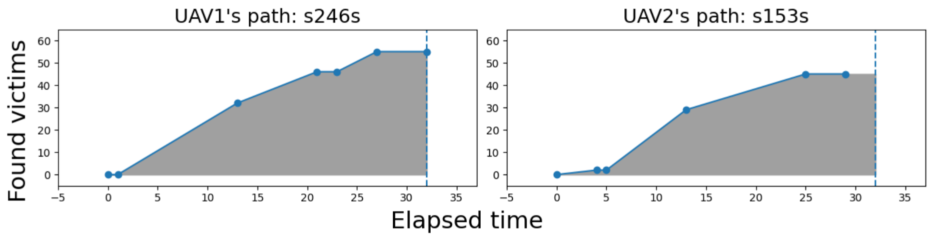


Fig. 2. Graphs of the number of detected victims over time; extracted from [13]

The second one is the sum of the areas over the curves describing the number of victims found over time, called S_{AOC}. Due to the different execution times of the UAVs' paths, the curves of the UAVs that finished earlier than the last one are extended to its length, called $T_{\max}$.

In Figure 2, $S_{\text{AOC}}(i)$ equals the sum of differences between the curve's bounding rectangle area of size $T_{\max} \cdot ENV(S_i)$ and the gray area under the curve. For an ideal solution, S_{AOC} equals 0. However, it is an unrealistic scenario. The second optimization function f_2 looks as follows:

$$f_2(\mathbf{x}) = \frac{S_{\text{AOC}}}{f_1(\mathbf{x}) \times v_{\text{exp}}} \qquad \text{where } S_{\text{AOC}} = \sum_{i=1}^{u} ext(S_{\text{AOC}}(i), T_{\max}) \tag{2}$$

$S_{\text{AOC}}(i)$ is an area over the curve defined by the i-th UAV's path. The parameter v_{exp} defines the ENV of the entire disaster area, and $ext(S_{\text{AOC}}(i), T_{\max})$ is a function that returns the area over the i-th curve extended to $T_{\max}$. The area of the sum of rectangle boundings for S_{AOC} equals $f_1(\mathbf{x}) \cdot v_{\text{exp}}$.

Eventually, the optimization goal is: $\min[f_1, f_2]$.

5 The Optimization Method

In the experimental part, we use one of Pareto-based evolutionary multiobjective optimization algorithms [3], namely Strength Pareto Evolutionary Algorithm Version 2 (SPEA2) [14]. Main steps of SPEA2 are presented in Algorithm 1.

Algorithm 1 SPEA2

```
 1: Initialization $P$                          ▷ generate an initial population of solutions
 2: Evaluation($P$)                             ▷ evaluate two objective values for each solution
 3: Evaluation$_{SF}$($P$)                      ▷ evaluate SPEA2 fitness
 4: $E \leftarrow$ Update($P$)                  ▷ initialize archive (external set)
 5: repeat
 6:     Evaluation$_{SF}$($E$)                  ▷ evaluate SPEA2 fitness
 7:     $P \leftarrow$ Tournament($E$)          ▷ perform binary tournament selection
 8:     $P' \leftarrow$ Variation($P$)          ▷ apply recombination and mutation
 9:     Evaluation($P'$)                        ▷ evaluate two objective values for each solution
10:     Evaluation$_{SF}$($P' \cup E$)          ▷ evaluate SPEA2 fitness
11:     $E \leftarrow$ Update($P' \cup E$)
12: until termination condition met
```

The algorithm manages two populations of solutions: a reference set P and an archive E. In the first step of the algorithm, we generate an initial population of solutions for P. For each of them, we evaluate the two objective functions (Step #2). Henceforth, the position of solutions in the objective space is known, and one can assess the SPEA2 fitness of solutions (Step #3). Eventually, we copied the entire reference set to the archive since we assumed that the archive size equals the reference set size (Step #4). Then, the main loop starts.

In the first step of the main loop (Step #6), we evaluate SPEA2 fitnesses of solutions in the archive E, which is redundant in the first execution of the main loop but not in the next ones. Then, the binary tournament selection on the archive's population generates clones to save to the current reference set P (Step #7). Variation operators perturb solutions in P and make its new representation, P'. There are two types of variation operators: binary (crossover) and unary (mutation). Solutions represent a set of u paths for all UAVs. Therefore, the operators have to consider the complex structure of the solution representation.

For pairs of solutions randomly selected from P, we perform two variation operators: crossover and mutation (Step #8). Crossover implements the PMX operator [5]. PMX operates on two paths: permutations of elements from n unique IDs. In our case, we have a team of UAVs, and a solution consists of a set of their paths being sequences of regions' IDs. For the aim of the representation adaptation to PMX, we concatenate the UAVs' paths into one list of regions' IDs and remember the paths' lengths, that is, concatenation points. This way, we always obtain lists of the same constant length regardless of the UAVs' path lengths and know how to reverse the process of merging paths into one list. The lists' lengths equal the total number of regions. Then, the PMX operator is executed in a typical manner. First, for the two parents randomly selected from P, it selects uniformly two cut points along the list at random. Then, the substrings between the two cut points are copied into offsprings. Next, the remaining values are copied from respective parents, following the rules defined in PMX, to guarantee the feasibility of newly created paths. In the last step, we restore the division into UAVs' paths concerning the remembered concatenation points.

Then, we perform the mutation operator. Four mutation operators adjusted to the current problem-specific representation have been proposed in [13]. In this research, two new operators are proposed. Hence, we have six operators in hand.

All the operators apply three modification procedures, each based on two evaluation criteria. Modification procedures are as follows:

1. **M**ove — in a randomly selected UAV's path, we insert a randomly selected sector in front of another randomly selected sector,
2. **E**xchange — any two randomly selected sectors in two randomly selected UAV's paths are inserted in front of randomly selected sectors in the opposite paths,
3. **H**op — we insert a randomly selected sector from a randomly selected UAV's path in front of another randomly selected sector of another randomly selected UAV's path.

In the three procedures, neither the UAVs' paths nor the new locations for sectors in these paths are selected uniformly at random. Chances for a path to be selected may depend on one of two criteria: its total execution time (the longer, the more chances — criterion pcT) or the total expected number of victims in all sectors in the path (the more victims found, the more chances — criterion pcP). Chances for selecting a location in a path for a sector s may also depend on one of two criteria. One of them is the traverse time (criterion scT). That is, for each pair of sectors s_i and s_j, we calculate traverse times from s_i to s and s to s_j, and the chances are inversely proportional to the sum of the two (the shorter traverse time, the more chances). Chances for selecting a location in a path for a sector s may also depend on the difference between the population density in s and in sector s_i in front of which s will be located (criterion scD). In this case, for each sector s_i in the path, we calculate the difference between its population density and the population density in s, namely $\mathrm{diff}(s_i, s)$. When all the differences are greater than zero, the chances are proportional to them. Otherwise, the chances are proportional to: $1+\mathrm{diff}(s_i, s)-\min(\mathrm{diff}(s_j, s))\ j \in \{1, \ldots, \text{the path's length}\}$.

Eventually, we obtain six mutation operators labeled with the first letter of the modification procedure and the selection criteria symbols: M[pcT, scT], M[pcP, scD], E[pcT, scT], E[pcP, scD], H[pcT, scT], H[pcP, scD]. The operators originate from the four problem-specific perturbation operators presented in [13]. These four operators have symbols LD:GD, LD:GT, LT:GD, and LT:GT. They can be mapped on the current six ones as follows: operator LD:GD works the same as application of the two: M[pcP, scD] and E[pcP, scD], LD:GT — M[pcP, scD] and E[pcT, scT], LT:GD — M[pcT, scT] and E[pcP, scD], LT:GT — M[pcT, scT] and E[pcT, scT].

Since we did not know which mutation would be the most beneficial, we applied a simple reward system. We use a table T_{mut} containing six cells where initial values in all cells equal $1/6$. They represent the probability of the mutation operator selection for the six operators. The mutation step begins with a random selection of the operator. Probability selection depends on the factors in T_{mut}, which add up to one. The operator receives a prize when the mutated offspring is better than its parent, that is, the respective cell in T_{mut} grows by δ_{rew}. Then, values in cells are normalized to sum up to one again.

The remaining steps of the main loop represent generic procedures of SPEA2. In Step #9, we evaluate the two objective functions for all new solutions stored in P' and in Step #10 — the SPEA2 fitness of solutions. Then, a new archive E content is selected from the sum: P' and E (Step #11). It is a ranking selection based on the solution's SPEA2 fitness values. The main loop is executed for a limited number of the SPEA2 fitness evaluations, and this number is one of the algorithm's parameters.

The following parameters control the algorithm: the reference set size $|P|$, the archive size $|E|$, crossover probability p_{cross}, mutation probability p_{mut}, the table T_{mut} storing coefficients representing mutation operators' chances for application, prize δ_{rew} for mutation operators for improving the solution, and a stopping condition which is the maximum number of SPEA2 fitness function calls max_{nffc}.

We have five ways of the initial population generation: random initialization and using the outcome of four versions of Local Search with four problem-specific perturbation operators: LD:GD, LD:GT, LT:GD, and LT:GT [13]. LS optimizes just the first objective function, f_1. In the case of a tie, the second objective function, f_2 defined in eq.(2), decides. SPEA2 is executed five times for every test case, once using a random population and four times — populations generated by LS.

6 Experimental research

6.1 Plan of experiments

In the experimental part, we did tests with the SPEA2 algorithm. A solution represents a set of paths for $u = 10$ UAVs, and the total expected number of victims $v_{\text{exp}} = 100\%$ of the population in the test area, hence each region must appear just once in one and only one path. Therefore, the sum of the paths' lengths in all solutions is the same and equals the total number of regions. The algorithm parameters are as follows: population size equals 30, $p_{\text{cross}} \in \{0.25, 0.5, 0.75, 1\}$, $p_{\text{mut}} \in \{0.25, 0.5, 0.75, 1\}$ and $\delta_{\text{rew}} = \{0.001, 0.005, 0.01, 0.05\}$.

The size of the disaster area's side $n = 30$, the unit coverage time $T_{\text{sc}} = 10$, the UAV's cruise velocity during traverse from one region to another nonadjacent one $V_{\text{cr}} = 1$, and stopping condition, that is, the maximum number of SPEA2 fitness function calls $max_{\text{nffc}} = 500000$. We have four testing areas from TCG-CCPP covering: Gliwice, Lodz, Warsaw (SW), and Gdansk Bay.

We did five groups of experiments. They differ in the origin of the SPEA2 initial population. In four groups, the initial population is the outcome of the Local Search application using one of the four versions of perturbation: LD:GD, LD:GT, LT:GD, or LT:GT [13]. Each version of LS was called 30 times to generate a population of 30 solutions. In the fifth group, the initial population was generated randomly.

Objective space consists of two criteria: f_1 as defined in Eq. (1) and f_2 as defined in Eq. (2). Both criteria are minimized. Experiments aim at verification of the ability of multiobjective optimization algorithms to find valuable sets of nondominated solutions in the given objective space. Additionally, we investigate the role of initial populations for SPEA2 in searching for the most prominent sets and the influence of the initial population on the tuning of SPEA2 evolutionary parameters.

6.2 Results of experiments

Each of the Figures 3-6 presents the five most promising sets of solutions found by SPEA2 using initial populations generated by four LS operators for the four testing areas. Figure 7 shows the five most promising sets of results obtained using random initial populations for the four areas.

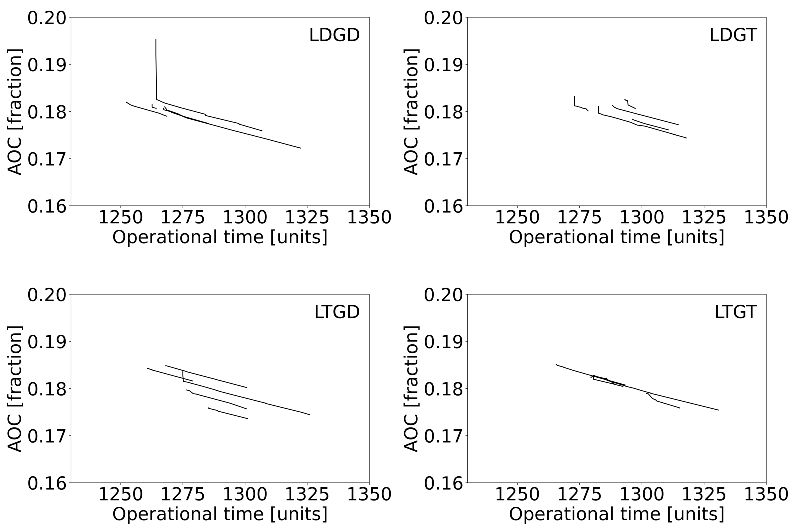


Fig. 3. The five most promising sets of solutions found by SPEA2 using initial populations generated by four LS operators for the Gliwice area

On most diagrams, there is no set containing both the best solution in terms of makespan and the best solution in terms of the area over the curve. Only the results obtained from the initial population generated by LT:GT for the Warsaw

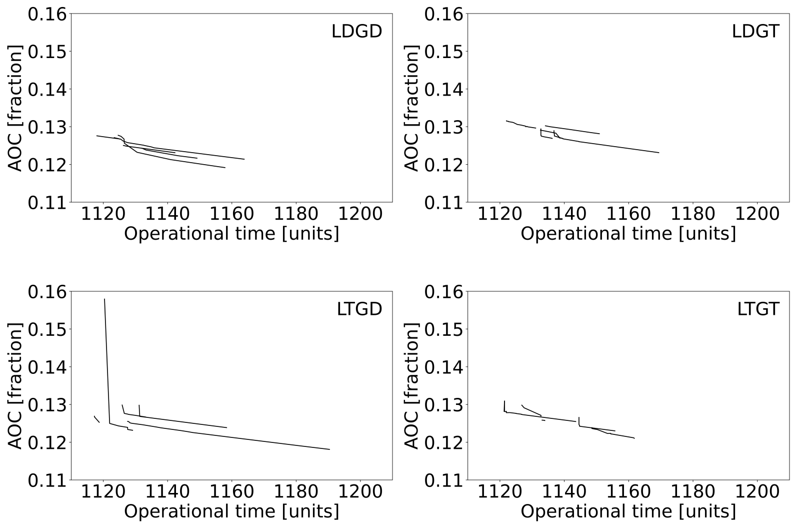


Fig. 4. The five most promising sets of solutions found by SPEA2 using initial populations generated by four LS operators for the Lodz area

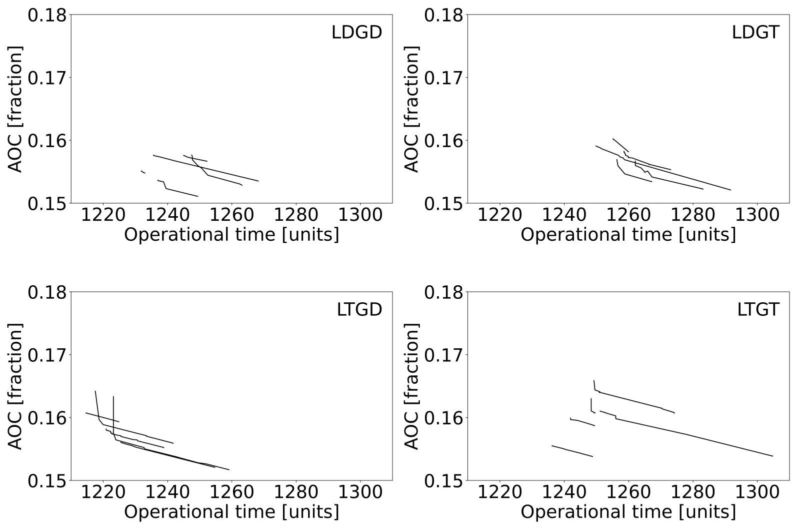


Fig. 5. The five most promising sets of solutions found by SPEA2 using initial populations generated by four LS operators for the Warsaw (SW) area

area and from the random initial populations for the Warsaw and the Gdansk Bay areas contain a set with the best solutions in both terms. On the other

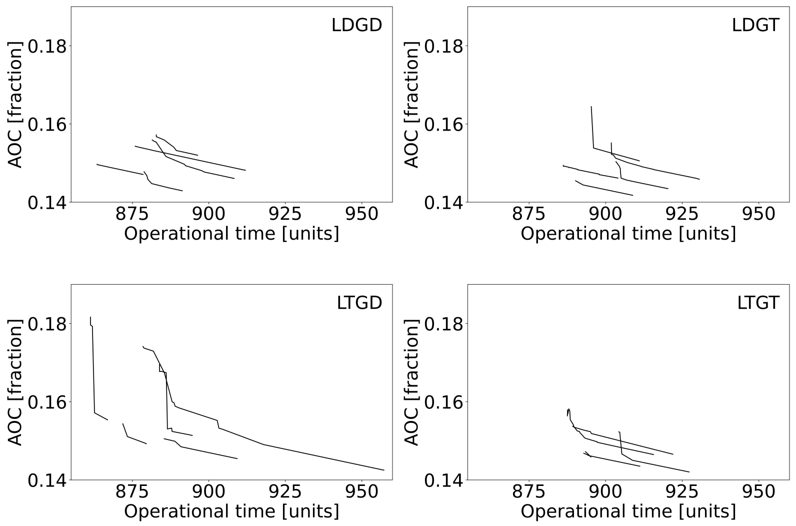


Fig. 6. The five most promising sets of solutions found by SPEA2 using initial populations generated by four LS operators for the Gdansk Bay area

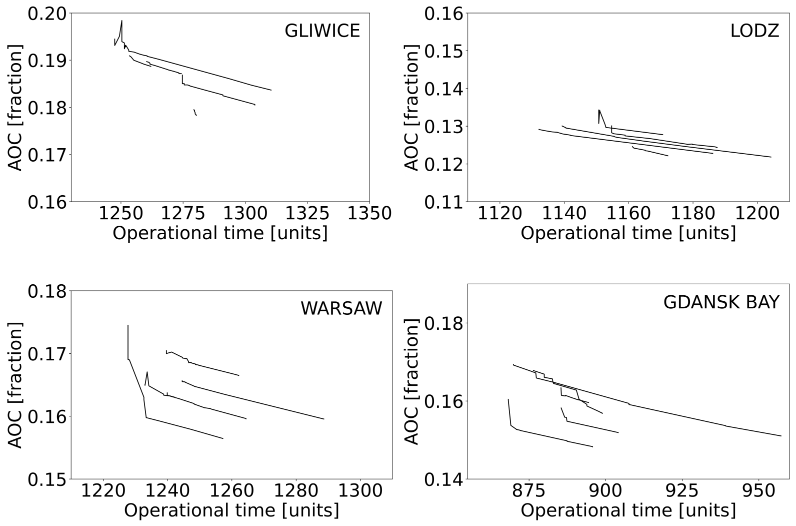


Fig. 7. The five most promising sets of solutions found by SPEA2 using random initial populations for the four areas

hand, there are diagrams with some sets that contain only solutions worse than those of the remaining sets.

Table 1. Configurations of SPEA2 parameters: p_{cross}, p_{mut} and δ_{rew} for the experiments, which results are presented in Figure 3 (the Gliwice area), Figure 4 (the Lodz area) and Figure 7

init.	Gliwice area			Lodz area		
pop.	p_{cross}	p_{mut}	δ_{rew}	p_{cross}	p_{mut}	δ_{rew}
LS[LD:GD]	0.50	0.75	0.050	0.25	0.50	0.050
	0.75	1.00	0.010	0.25	0.75	0.001
	1.00	0.50	0.005	0.50	0.50	0.001
	1.00	0.50	0.010	0.50	0.50	0.010
	1.00	0.50	0.050	1.00	0.50	0.010
LS[LD:GT]	0.75	0.75	0.010	0.25	0.50	0.005
	1.00	0.50	0.005	0.25	1.00	0.005
	1.00	0.75	0.005	0.75	0.25	0.001
	1.00	0.75	0.010	0.75	0.50	0.005
	1.00	1.00	0.005	0.75	1.00	0.005
LS[LT:GD]	0.25	0.75	0.005	0.50	0.75	0.005
	0.25	0.75	0.010	0.50	0.75	0.010
	0.50	0.75	0.010	0.75	0.50	0.005
	1.00	0.75	0.010	1.00	0.50	0.001
	1.00	1.00	0.005	1.00	0.50	0.010
LS[LT:GT]	0.50	0.75	0.005	0.25	0.50	0.005
	0.50	0.75	0.010	0.25	1.00	0.001
	0.75	0.75	0.010	0.50	0.75	0.005
	0.75	1.00	0.005	0.75	0.75	0.005
	1.00	0.75	0.050	1.00	0.50	0.001
randomly	0.75	0.75	0.050	0.50	1.00	0.001
	0.75	1.00	0.005	0.50	1.00	0.005
	1.00	0.50	0.005	0.75	0.75	0.001
	1.00	0.75	0.010	0.75	0.75	0.005
	1.00	1.00	0.005	1.00	0.75	0.005

Experiments described in [13] show that for the Lodz and Warsaw areas, the perturbation operator LT:GT produced the best LS results. In our current experiments, for the Lodz area, both the best solutions were produced from the initial population generated by LT:GD. For the Warsaw area, the best makespan solution was in a set obtained from the initial population generated by LT:GD, while the best AOC solution was contained in a set obtained from the initial population generated by LD:GD. According to [13], for the Gdansk Bay area, operator LT:GD produced the best LS results. In our current experiments, the best makespan solution was in a set obtained from the initial population generated by LT:GD, while the best AOC solution was contained in a set obtained from the initial population generated by LD:GT. LD:GT produced the best LS results for the Gliwice area. In the current research, we got the best makespan solution using a random initial population. The best AOC solution was in a set

Table 2. Configurations of SPEA2 parameters: p_{cross}, p_{mut} and δ_{rew} for the experiments, which results are presented in Figure 5 (the Warsaw (SW) area), Figure 6 (the Gdansk Bay area) and Figure 7

init.	Warsaw (SW) area			Gdansk Bay area		
pop.	p_{cross}	p_{mut}	δ_{rew}	p_{cross}	p_{mut}	δ_{rew}
LS[LD:GD]	0.75	0.50	0.001	0.25	0.50	0.005
	0.75	0.50	0.005	0.25	0.75	0.001
	0.75	0.75	0.050	0.50	0.75	0.001
	1.00	0.75	0.001	0.50	1.00	0.001
	1.00	0.75	0.010	0.75	0.75	0.001
LS[LD:GT]	0.50	0.75	0.010	0.25	0.50	0.001
	0.75	0.50	0.005	0.25	1.00	0.005
	0.75	1.00	0.005	0.50	0.75	0.001
	1.00	0.25	0.005	0.75	0.50	0.001
	1.00	0.50	0.001	0.75	0.75	0.001
LS[LT:GD]	0.25	0.75	0.010	0.25	0.75	0.001
	0.50	0.50	0.050	0.50	0.50	0.001
	0.75	0.75	0.010	0.50	1.00	0.001
	0.75	1.00	0.050	0.75	0.50	0.001
	1.00	0.75	0.010	0.75	0.75	0.001
LS[LT:GT]	0.25	1.00	0.050	0.25	0.50	0.001
	0.50	0.75	0.005	0.50	0.50	0.001
	0.50	0.75	0.010	0.50	0.75	0.005
	0.50	0.75	0.050	0.50	1.00	0.001
	0.75	0.75	0.005	0.75	0.50	0.001
randomly	0.50	0.75	0.010	0.25	1.00	0.001
	0.75	1.00	0.010	0.50	0.75	0.001
	1.00	0.50	0.010	1.00	0.50	0.001
	1.00	0.75	0.005	1.00	0.50	0.005
	1.00	1.00	0.010	1.00	0.75	0.001

obtained from the initial population generated by LD:GD. Thus, all five ways of generating an initial population for SPEA2 were useful for some data sets.

Tables 1 and 2 show configurations of SPEA2 parameters for experiments which results are presented in Fig 3- 7. All used values of p_{cross}, p_{mut}, and δ_{rew} appear in these tables. Moreover, for each way of generating the initial solution and for each area, the five most promising results were obtained using five different triples of values (p_{cross}, p_{mut}, δ_{rew}). Among 100 values of p_{cross}, 1.0 appeared 29 times, 0.75 28 times, 0.5 26 times, and 0.25 only 17 times. Among 100 values of p_{mut}, the most common was 0.75 (45 times), while 0.25 was the least common (only twice), 0.5 appeared 32 times, and 1.0 – 21 times. Values of δ_{rew} 0.001, 0.005, 0.01, and 0.05 appeared 32, 34, 24, and 10 times respectively.

We can conclude from the aforementioned observations that almost all values of SPEA2 parameters can provide promising results for some data sets. Only the probability of mutation 0.25 should be avoided in future experiments.

Figure 8 shows the improvement of solutions returned by tested algorithms. While Local Search offered quite a lot of improvement, further optimization with SPEA2 managed to shorten these solutions by up to 20%

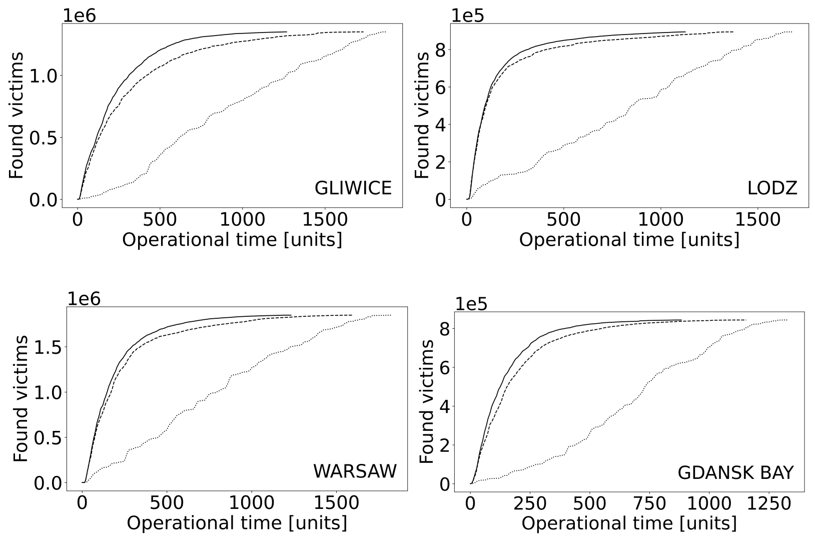


Fig. 8. Cumulative number of victims found over time for a randomly generated solution (dotted), a solution optimized by Local Search (dashed), and an example solution returned by SPEA2 (solid)

7 Conclusions

We applied the SPEA2 multiobjective optimization algorithm to the problem of complete coverage and path planning for emergency response by UAVs in disaster areas. We aimed to maximize the number of victims localized in the first minutes of the UAVs' operation and minimize the entire operation makespan.

For experiments, we used four test cases from TCG-CCPP based on the data from the Geostatistics Portal by Statistics Poland. Initial solutions for SPEA2 were created randomly or using LS with one of the four pairs of perturbation operators we described in our earlier paper. Our experiments generated promising sets of problem solutions.

We repeated the experiments with several parameters' values that determined the probability of crossover and mutation and the reward for a successful mutation. The results showed that almost all values used for these parameters could provide the most promising results for some data sets. The only restriction is that the mutation probability should be at least 0.5. The SPEA2 algorithm was able to significantly improve the entire operation makespan.

References

1. Basilico, N., Carpin, S.: Deploying teams of heterogeneous UAVs in cooperative two-level surveillance missions. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 610–615. IEEE (Sep 2015). https://doi.org/10.1109/iros.2015.7353435
2. Cabreira, T., Brisolara, L., Ferreira Jr., P.R.: Survey on coverage path planning with unmanned aerial vehicles. Drones **3**(1), 4 (Jan 2019). https://doi.org/10.3390/drones3010004
3. Emmerich, M.T.M., Deutz, A.H.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Natural Computing **17**(3), 585–609 (May 2018). https://doi.org/10.1007/s11047-018-9685-y
4. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robotics and Autonomous Systems **61**(12), 1258–1276 (Dec 2013). https://doi.org/10.1016/j.robot.2013.09.004
5. Goldberg, D.E., Lingle Jr., R.: Alleles, loci, and the traveling salesman problem. In: Grefenstette, J.J. (ed.) the First Int. Conf. on Genetic Algorithms and their Applications. pp. 154–159. L. Erlbaum Associates Inc., Pittsburgh, PA, USA (1985)
6. Grzeszczak, J., Trojanowski, K., Mikitiuk, A.: Test case generator for problems of complete coverage and path planning for emergency response by UAVs. In: Artificial Intelligence and Soft Computing. LNCS, vol. 14125, pp. 497–509. Springer (2023). https://doi.org/10.1007/978-3-031-42505-9_42
7. Kapanoglu, M., Alikalfa, M., Ozkan, M., Yazici, A., Parlaktuna, O.: A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time. Journal of Intelligent Manufacturing **23**(4), 1035–1045 (May 2010). https://doi.org/10.1007/s10845-010-0404-5
8. Khan, A., Noreen, I., Habib, Z.: On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges. Journal of Information Science and Engineering **33**(1), 101–121 (2017), `https://jise.iis.sinica.edu.tw/JISESearch/pages/View/PaperView.jsf?keyId=154_1997`
9. Li, L., Shi, D., Jin, S., Kang, Y., Xue, C., et al.: Complete coverage problem of multiple robots with different velocities. International Journal of Advanced Robotic Systems **19**(2) (Mar 2022). https://doi.org/10.1177/17298806221091685
10. Lin, H.Y., Huang, Y.C.: Collaborative complete coverage and path planning for multi-robot exploration. Sensors **21**(11), 3709 (May 2021). https://doi.org/10.3390/s21113709
11. Nasirian, B., Mehrandezh, M., Janabi-Sharifi, F.: Efficient coverage path planning for mobile disinfecting robots using graph-based representation of environment. Frontiers in Robotics and AI **8**, 1–19 (Mar 2021). https://doi.org/10.3389/frobt.2021.624333
12. Tan, C.S., Mohd-Mokhtar, R., Arshad, M.R.: A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. IEEE Access **9**, 119310–119342 (2021). https://doi.org/10.1109/access.2021.3108177
13. Trojanowski, K., Mikitiuk, A., Grzeszczak, J., Guinand, F.: Complete coverage and path planning for emergency response by UAVs in disaster areas. In: Computational Collective Intelligence. LNCS, vol. 14162, pp. 647–659. Springer (2023). https://doi.org/10.1007/978-3-031-41456-5_49
14. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. TIK Report 103, ETH Zurich, Computer Engineering and Networks Laboratory (May 2001). https://doi.org/10.3929/ETHZ-A-004284029