IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/45AI/2025/4527925/
Vol. 2025 (20), Article ID 4527925

Generating a Region Map for The Problem of Complete Coverage and Path Planning*

Krzysztof TROJANOWSKI, Jakub GRZESZCZAK and Artur MIKITUK

Cardinal Stefan Wyszynski University In Warsaw, Poland

Correspondence should be addressed to: Krzysztof TROJANOWSKI, k.trojanowski@uksw.edu.pl

* Presented at the 45th IBIMA International Conference, 25-26 June 2025, Cordoba, Spain

Abstract

Emergency response procedures for the victims of natural disasters require information about the number and location of victims and their needs or injuries. Unmanned Aerial Vehicles (UAVs) can efficiently reconnaissance the disaster area. The sum of UAVs' routes has to cover the entire disaster area, which is divided into sectors with priorities. The sector identification is based on merging areas with a similar expected number of victims. We prioritize sectors with more victims; UAVs should visit them as early as possible. This paper proposes a new method of building sectors for real-world data from GIS (Geographic Information System) data sets. The process is based on merging cells into regions using a flood-fill search approach, which can be implemented directly on the input due to the rasterized structure of these data sets. We verify the usefulness of the method by computer simulations.We perform experiments optimizing UAV paths for four areas in Poland. We divide each area in five ways into collections of regions and compare the usefulness of the division for optimizing UAV routes regarding two objectives to be minimized: the execution time of the operation and the size of the area over the curve of the cumulative distribution function of detected victims over time. A good division of the area into regions can increase the quality of the results.

Keywords: UAV path planning · evolutionary multiobjective optimization · GIS data sets.

Introduction

When a natural disaster happens, the first step of emergency response is identifying people in need, their number and locations, and establishing communications to determine their needs and injuries. A team of UAVs can be the first choice in this case. UAVs are never limited by ground damage or flooding in the disaster area. In our research, we plan UAV team routes covering the entire area, prioritizing contact with the largest possible number of victims in the shortest possible time. UAVs carry Mobile Base Stations (MBSs) to connect with the turned-on mobile phones of the victims and receive phones' precise GPS coordinates and owners' identity through phone or IMEI numbers.

In this paper, we propose a new method of building disaster area representations compatible with a model of the problem proposed by Grzeszczak et al. (2023) and based on GIS data sets about the number of residents in Poland. The area is represented as a set of convex regions, and the regions' structure, size, and number impact the effectiveness of UAVs' routes generated by optimization algorithms. The earlier method proposed by Trojanowski et al. (2024) of region construction arranges cells into regions using arbitrarily defined thresholds of resident numbers originating from their distribution over the entire area. The new method considers the

Cite this Article as: Krzysztof TROJANOWSKI, Jakub GRZESZCZAK and Artur MIKITUK, Vol. 2025 (20) "Generating a Region Map for The Problem of Complete Coverage and Path Planning " Communications of International Proceedings, Vol. 2025 (20), Article ID 4527925, https://doi.org/10.5171/2025.4527925

similarities in the number of residents in the connected cells. We evaluate the usefulness of these two methods in practice. We select four areas in Poland and generate test cases for each of them using the two methods, the old and the new ones. For a given area, the earlier proposed method generates one benchmark class consisting of one test case. The new method defines two benchmark classes, called **FibD** and **LinD**, each consisting of two versions of the test case generation. Hence, for a given area, we generate five test cases in total. Then we perform UAVs' route optimization and compare the results obtained for each area. Better methods generate test cases that bring faster contact with the largest possible number of victims. Thus, the results of the UAVs' route optimization algorithm may indicate the winner of the method comparison.

The paper consists of six sections. Section 2 presents the methods of generating regions for the disaster area, including the one newly proposed. Section 3 briefly describes the optimization algorithm used to evaluate the region construction methods. Section 4 presents a plan of experiments. The experimental part of the research is described in Section 5. Finally, Section 6 concludes the paper.

Generation of Maps of Regions for The Disaster Area

Information on population density is often provided in the form of GIS maps where each cell has assigned its number of residents. Such data structure of the raw information imposes discrete modeling of disaster areas.

Our goal is to find UAV team routes that establish contact with the largest possible number of victims while covering the entire region in the shortest possible time. Hence, the solution structure fitting the model space represents ordered lists of cells for the team members to visit. We want to reduce the complexity of this optimization problem, so raw GIS maps undergo preprocessing where cells with similar populations are merged into regions. Final maps should consist of non-overlapping rectangular regions of different sizes. Hence, the map generation process has two main stages: in the first one, we select cells to merge and turn them into larger regions, and in the second one, we split concave regions into sets of rectangular ones. The model enforces the rectangular shape of all regions to ensure that the full-coverage routes can be generated using well-known algorithms proposed by Basilico and Carpin (2015), Kapanoglu et al. (2010), Li et al. (2022), Lin and Huang (2021), Nasirian et al.. (2021) and Tan et al.. (2021).

Grzeszczak et al. (2023) proposed a simple yet effective method of region generation from GIS data sources. The method is based on assigning cells to several groups w.r.t. their population sizes. We generated test cases with this method to evaluate the optimization algorithm applied to the problem of complete coverage and path planning for emergency response by UAVs in disaster areas, as proposed by Trojanowski et al. (2023). Subsection 2.1 briefly describes this method. Now, we propose another method that emphasizes the role of population-level similarities between neighboring cells to be merged. We describe it in Subsection 2.2. Subsection 2.3 shows the second stage of the generation process aimed at turning concave regions into convex ones, following the process proposed by Grzeszczak et al.. (2023), common for both methods from Subsections 2.1 and 2.2. For experimental comparisons described in this paper, we applied the two methods to generate new test cases from the GIS data sources described in Subsection 2.4.

Method of Regions Generation Based on Uniform Distribution

Using the population histogram of all considered cells, we divided the histogram into six ranges of roughly equal population totals and used the thresholds between them as region classifiers. Next, we merged neighboring cells with the same classifiers into islands, considering each island as a single region. Finally, we used the convex division algorithm to divide all concave regions into smaller convex ones.

Method of Regions Generation Based on Simmilar Cells Connection

The division of the area into regions based on similarity constraints (Algorithm 1) works with the reference set of all cells in the area and two sequences of values.

The first sequence defines rising upper limits for cell population values considered by the algorithm. The second defines fractions of a current limit, which are used as an acceptable threshold of similarity between cells.

The algorithm starts with a complete set of cells in the area. Cells above the current limit are temporarily excluded from calculations for each step of the main loop.

Then, the algorithm uses a flood-fill search approach for each acceptable threshold to determine groups of similar cells around each considered cell. This step is defined by the RCP procedure, presented in Algorithm 2. For any cell c, an area representation A, and a threshold t, the RCP procedure returns a collection of cells reachable from c by moving between cells with an absolute difference in population from the starting cell smaller than t.

Algorithm 1 A Procedure For Dividing the Area Into Regions

Input:
A — the input set of all cells (the area to be divided into regions),
Σ — max. population sizes of the cells to be considered,
Δ — acceptable thresholds between the neighboring cells' population sizes

procedure PDAR(A, Σ, Δ)
  $U_R \leftarrow \emptyset$ ▷ the set of regions is empty
  for all $\sigma \in \Sigma$ **do**
    for all $\delta \in \Delta$ **do**
      $t \leftarrow \sigma * \delta$ ▷ calculate the acceptable difference between cells
      $A' \leftarrow \{c \in A | _{\mathcal{P}(c)<\sigma}\}$ ▷ $\mathcal{P}(c)$ denotes the population of the cell c
      for all $c \in A'$ **do**
        $R(c) \leftarrow \mathrm{RCP}(c, c, A', t)$ ▷ generate a candidate region
        if $R(c) \in U_R \vee |R(c)| = 1$ **then** ▷ ignore duplicate or small regions
          continue
        if $\forall c' \in R(c),\ R(c) = \mathrm{RCP}(c', c', A', t)$ **then**
          $U_R \leftarrow U_R \cup R(c)$ ▷ add a region to the list
          $A \leftarrow A \setminus R(c)$ ▷ ignore the cells in future calculations
    if $\sigma > max(A)$ **then** ▷ stop the limit-generation procedure
      break
  for all $c \in A$ **do** ▷ transform remaining cells into regions
    $U_R \leftarrow U_R \cup R(c)$
  return U_R

Algorithm 2 A Region Construction Procedure

Input:
c — current cell,
c_{ref} — starting cell for a new region,
A — the input set of considered cells,
δ — acceptable population difference between cells

procedure RCP(c, c_{ref}, A, δ)
  $R \leftarrow \{c\}$ ▷ R consists of just one cell, c
  for all $c_n \in \mathcal{N}_A(c)$ **do** ▷ for all neighbors of c from A
    if $|\mathcal{P}(c_n) - \mathcal{P}(c_{\mathrm{ref}})| \leq \delta$ **then** ▷ if the difference between pop. sizes is $\leq \delta$
      $R \leftarrow R \cup \mathrm{RCP}(c_n, c_{\mathrm{ref}}, A, \delta)$ ▷ RCP is recursively run for c_n
  return R ▷ the newly generated region R is returned

When the size of a generated region is greater than one, and its shape can be recreated by the procedure starting from each of its consisting cells, the cells are permanently excluded from further testing, and the region is saved. This procedure is repeated with different similarity constraints and ends after the first limit value higher than the grid's maximum is tested.

Finally, all cells not permanently excluded from the search despite reaching the termination condition are transformed into single-cell regions to ensure the complete representation of the area.

Convex Division Algorithm

We decided to break concave regions into multiple smaller rectangles to ensure easy coverage path creation for any considered region. For a concave set of cells, we consider them in a random order. For each cell that has not already been excluded, we define a new convex region with its bounds and try to expand it in random directions for as long as possible. Each time a direction is selected, we check if all the neighboring cells are parts of the

original island and were not already excluded. If the check is true, the region is expanded, and those cells are excluded from further search. Otherwise, the region will be prevented from expanding in this direction until completion. The procedure is repeated until all growth directions are locked. This approach allows us to find regions that were more likely to show up in the set of all possible divisions of the area.

GIS Data Sources

For experimental comparisons, we selected four testing areas, called frames, representing the compilation of information from two GIS Data Sources.

The first data source is a Map of Areas at High Risk of Flooding from Polish Geological Institute – National Research Institute (2007) prepared between 2003 and 2006 by the Polish Geological Institute as one of the tasks of The Polish Hydrogeological Survey. The designated areas show the maximum possible range of overflow presence (i.e., the position of the groundwater table close to the land level, which results in wetness) in the vicinity of a river valley.

The second data source is the map ”The JRC-GEOSTAT population grid, Version 2018” from Batista e Silva et al.. (2021) published by GEOSTAT on April 13, 2021. It was necessary to use the second source since the map from the Polish Geological Institute identifies only the hazardous areas and has no information about their residents.

Evaluation of the Benchmarks – Multiobjective Heuristic Optimization

Before the benchmarks' evaluation is described, it is worth recalling that in our research, we optimize contact with the largest possible number of victims in the shortest possible time. Arranging the order of visited regions on UAV paths is not the only way to achieve this goal. How the disaster area is divided into regions also matters, and this can be done in many ways, as described above. The results of these two actions, which are (1) the area division into regions and (2) the regions' arrangement in paths, are not independent. On the contrary, a good division of the area into regions can increase the arrangement's efficiency.

Trojanowski et al. (2023) evaluated paths with the following two objectives: the longest execution time among the paths in the solution, called f1, and the share of the area over the plotted curve depicting the number of all discovered victims over time for this solution, called f2. The share is calculated for the rectangle formed by the makespan of the solution with the shortest execution time in the current set of solutions and the number of all discovered victims over time (see the publication by Trojanowski et al. (2023) for a detailed description of f2). We minimize both objectives.

To evaluate the quality of an area's division into regions, we optimize the paths in this area using a heuristic multiobjective optimization algorithm with the two objectives mentioned above. We assume that the final result of this algorithm reflects the quality of the area division. Hence, the algorithm's outcome enables the comparison of different divisions.

We used one of the Pareto-based multiobjective evolutionary algorithms for the experimental evaluation of area division methods, namely the SPEA2 algorithm. We implemented the algorithm as Zitzler, Eckart et al. (2001) described. However, our representation of the solution required problem-specific operators of crossover and mutation. A solution contains paths for a team of UAVs. A path consists of a starting point, an ordered subset of IDs of non-empty regions from the set of all regions, and an ending point. Empty regions, like lakes or deserts, have no residents assigned. Thus, there is no benefit in visiting them. The sum of region IDs from the paths equals the set of IDs of all non-empty regions, and duplicates are not allowed.

Crossover implements the PMX operator proposed by Goldberg and Lingle Jr. (1985). Before the operator is executed, we create new representations of solutions, concatenating all the UAV paths in a solution into a list of unique regions. The conversion of the set of paths into one sequence is necessary to apply PMX. The resulting permutation is reverted to the original form by cutting the sequence into lists of the same lengths as before and inserting them between each path's starting and ending points.

In the mutation operator, we insert a randomly selected region s from one path at a randomly chosen position in another path. Chances for a path to be selected as a donor depend on its total execution time; the longer, the more chances. In the case of the recipient, chances for its selection also depend on its total execution time, but in the opposite way, the shorter, the more chances. When the recipient is chosen, we select the position, that is, the region si in front of which s will be located. Chances for si depend on the difference between the population

density in the region s and si, namely diff(si, s). Briefly, when all the differences are greater than zero, the chances are proportional to them; otherwise, the chances are proportional to: 1 + diff(si, s) − min(diff(sj, s)), j ∈ {1, … , the path's length}.

Plan of Experiments

We did experiments with three benchmark classes, UniB, FibD, and LinD. Both FibD and LinD have two instances that differ in some of the control parameters, while there is one instance of UniB. Raw data for the benchmarks represent frames on the map of Poland with the side size 31×30 units. The selected frames are: (1) the area near Brzeszcze, (2) the area near Tarnow, (3) Generating a Region Map 7 the delta of the Vistula River, and (4) the Warsaw-North area, and we refer to them as frames B, T, V, and W. Eventually, we get five sets of test cases: UniB, LinD1, LinD2, FibD1 and FibD2, each consisting of four test cases.

The first set, UniB, was generated using the earlier proposed method, which divides the cells into six ranges of roughly equal population totals and merges neighboring cells only from the same range. The remaining ones were generated using the newly proposed method based on connecting the most similar cells.

Test cases from LinD are generated with semi-linear limits for max. population sizes of the cells to be considered (the set Σ in Algorithm 1, PDAR). The algorithm loops values of 2×10n, 5×10n, and 10×10n, incrementing n from 1 until the first limit higher than the test case's population is used.

For the test cases from FibD, we define limits in the set Σ using a Fibonacci sequence starting from numbers 30 and 60. The termination condition works the same as for LinD cases.

We use two sets of acceptable thresholds between the neighboring cells' population sizes (the set Δ in Algorithm 1, PDAR) for different test cases. For LinD1 and FibD1, the values are 0.1, 0.15, and 0.2, and for LinD2 and FibD2, they are 0.1, 0.2 and 0.3.

For each of the twenty test cases, we did a set of experiments with the SPEA2 algorithm described in Section 3. Each set was performed using the crossover probability 75% and two values of the mutation probability: 50% and 75%. The outcome of a single execution of SPEA2 is a set of non-dominated solutions. The solutions form a Pareto set in the two-objective space. The performance of SPEA2 is presented using a hypervolume indicator proposed by Zitzler and Thiele (1999). The Pareto set's hypervolume measures the size of the space enclosed by all solutions of the set and a pre-defined reference point.We apply the straightforward reference point estimation method given by Ishibuchi et al. (2017) and Jiang and Yang (2017), which takes the worst, i.e., the maximum value of each objective in the set of non-dominated solutions. For fair comparisons, we used four reference points calculated for groups of results for the test cases obtained from the frames B, T, V, and W, respectively. Eventually, we present distributions of the hypervolumes (minimum, lower quartile, median, upper quartile, and maximum) in boxplots and means of the hypervolume.

Results of Experiments

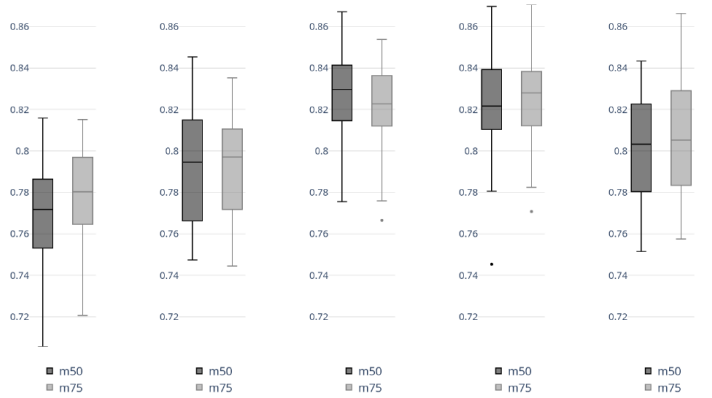
For test cases from the frames B, T, V, and W build with different region construction methods, Figures 1–4 present hypervolumes for the results of SPEA2 using different mutation probabilities. The figures depict the distribution of the hypervolumes (minimum, lower quartile, median, upper quartile, and maximum) in respective boxplots. The mutation probability identifies the SPEA2 versions: m75 denotes probability 75% while m50 — 50%.

One can see from Fig. 1–4 and Table 1 that for three out of four frames, mean hypervolume results for the benchmark class UniB are lower than the ones for the other benchmarks classes. Especially for frame V, the results obtained using UniB are significantly lower than the ones obtained from the other benchmarks. Only for frame T, we got the lowest mean hypervolumes using LinD1. The highest mean hypervolumes were almost always obtained using the benchmark FibD2. Only for frame B, using mutation probability 75% makes LinD1 the winner.

Thus, the proposed new method of generating a map of regions gives better hypervolume results than our old method.

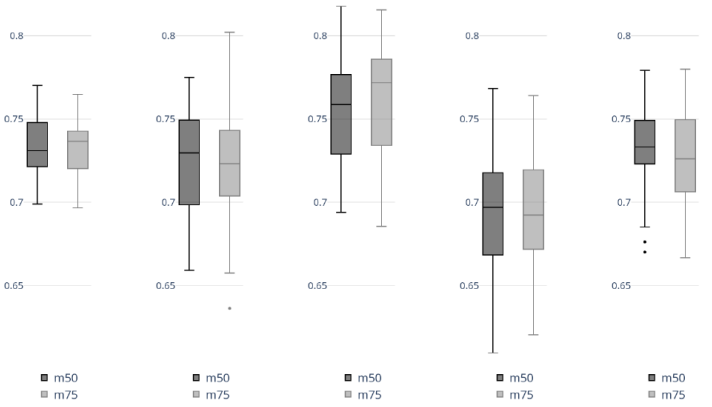
Table 1. Mean hypervolumes; in each column, the least means are printed in bold, the highest — in italics

	Frame **B**		Frame **T**		Frame **V**		Frame **W**	
div.	m50	m75	m50	m75	m50	m75	m50	m75
UniB	**0.768**	**0.778**	0.734	0.733	**0.719**	**0.724**	**0.783**	**0.787**
FibD1	0.793	0.793	0.724	0.724	0.844	0.835	0.815	0.814
FibD2	*0.825*	0.822	*0.755*	*0.762*	*0.885*	*0.896*	*0.840*	*0.836*
LinD1	0.822	*0.824*	**0.692**	**0.694**	0.870	0.877	0.813	0.804
LinD2	0.801	0.807	0.733	0.727	0.863	0.865	0.802	0.806



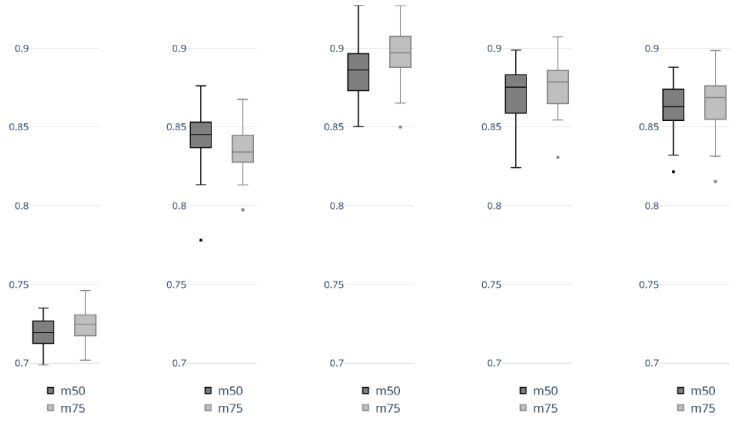
(a) **UniB** (b) **FibD1** (c) **FibD2** (d) **LinD1** (e) **LinD2**

Fig. 1. Hypervolumes for test cases from the frame **B** for benchmarks **UniB**, **FibD1**, **FibD2**, **LinD1** and **LinD2**



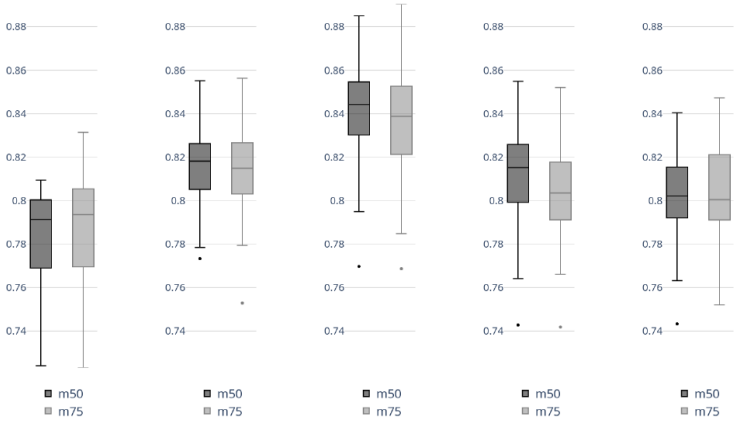
(a) **UniB** (b) **FibD1** (c) **FibD2** (d) **LinD1** (e) **LinD2**

Fig. 2. Hypervolumes for test cases from the frame **T** for benchmarks **UniB**, **FibD1**, **FibD2**, **LinD1** and **LinD2**



(a) **UniB** (b) **FibD1** (c) **FibD2** (d) **LinD1** (e) **LinD2**

Fig. 3. Hypervolumes for test cases from the frame **V** for benchmarks **UniB**, **FibD1**, **FibD2**, **LinD1** and **LinD2**



(a) **UniB** (b) **FibD1** (c) **FibD2** (d) **LinD1** (e) **LinD2**

Fig. 4. Hypervolumes for test cases from the frame **W** for benchmarks **UniB**, **FibD1**, **FibD2**, **LinD1** and **LinD2**

Conclusions

In this paper, we present a new method of generating a map of regions in the disaster area. This method is based on similar cell connections. The resulting map can be used by algorithms optimizing a set of paths for a team of UAVs to be dispatched over the area in question. This reconnaissance mission aims to locate victims in the area in the shortest possible time while trying to find as many of them as possible during the first minutes of the operation.

We conducted experiments to compare the results of the heuristic optimization algorithm obtained for two groups of test cases based on the same origins: the ones created with the new method and those with the old one used in our previous studies. For this purpose, we created new test cases using GIS data sources. We evaluated both methods using the SPEA2 algorithm for multiobjective heuristic optimization. Their relative performance is presented by a hypervolume indicator. Our research shows that the proposed new method of generating a map of regions gives better hypervolume results than our old method.

Acknowledgements

The authors thank Yevhenii Palamarchuk for his assistance in computer simulations.

References

- Basilico and Carpin (2015). Basilico, N. and Carpin, S. (2015). Deploying teams of heterogeneous UAVs in cooperative two-level surveillance missions. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 610–615. IEEE.
- Batista e Silva et al. (2021), Batista e Silva, F., Dijkstra, L., and Poelman, H. (2021). The JRC-GEOSTAT population grid, Version 2018 (13 April 2021). https://ec.europa.eu/eurostat/web/gisco/geodata/population-distribution/geostat. Data retrieved on 7.08.2024.
- Goldberg and Lingle Jr. (1985), Goldberg, D. E. and Lingle Jr., R. (1985). Alleles, loci, and the traveling salesman problem. In Grefenstette, J. J., editor, Proceedings of the First International Conference on Genetic Algorithms and their Applications, pages 154–159, Pittsburgh, PA, USA. L. Erlbaum Associates Inc.
- Grzeszczak et al. (2023), Grzeszczak, J., Trojanowski, K., and Mikitiuk, A. (2023). Test case generator for problems of complete coverage and path planning for emergency response by UAVs. In Artificial Intelligence and Soft Computing, volume 14125 of LNCS, pages 497–509. Springer.
- Ishibuchi et al. (2017), Ishibuchi, H., Imada, R., Setoguchi, Y., and Nojima, Y. (2017). Reference point specification in hypervolume calculation for fair comparison and efficient search. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17, page 585–592. ACM.
- Jiang and Yang (2017), Jiang, S. and Yang, S. (2017). A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. IEEE Transactions on Evolutionary Computation, 21(3):329–346.
- Kapanoglu et al. (2010), Kapanoglu, M., Alikalfa, M., Ozkan, M., Yazici, A., and Parlaktuna, O. (2010). A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time. Journal of Intelligent Manufacturing, 23(4):1035–1045.
- Li et al. (2022), Li, L., Shi, D., Jin, S., Kang, Y., Xue, C., Zhou, X., Liu, H., and Yu, X. (2022). Complete coverage problem of multiple robots with different velocities. International Journal of Advanced Robotic Systems, 19(2):172988062210916.
- Lin and Huang (2021). Lin, H.-Y. and Huang, Y.-C. (2021). Collaborative complete coverage and path planning for multi-robot exploration. Sensors, 21(11):3709.
- Nasirian et al. (2021). Nasirian, B., Mehrandezh, M., and Janabi-Sharifi, F. (2021). Efficient coverage path planning for mobile disinfecting robots using graph-based representation of environment. Frontiers in Robotics and AI, 8:1–19.
- Polish Geological Institute – National Research Institute (2007). Polish Geological Institute – National Research Institute (2007). Maps of Areas at High Risk of Flooding. https://www.pgi.gov.pl/en/phs/data/9047-maps-of-areas-at-high-risk-of-flooding. html. data retrieved on 7.08.2024.
- Tan et al. (2021), Tan, C. S., Mohd-Mokhtar, R., and Arshad, M. R. (2021). A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. IEEE Access, 9:119310–119342.
- Trojanowski et al. (2023). Trojanowski, K., Mikitiuk, A., Grzeszczak, J., and Guinand, F. (2023). Complete coverage and path planning for emergency response by UAVs in disaster areas. In Computational Collective Intelligence, volume 14162 of LNCS, pages 647–659. Springer.
- Trojanowski et al. (2024), Trojanowski, K., Mikitiuk, A., Grzeszczak, J., and Guinand, F. (2024). Multiobjective optimization of complete coverage and path planning for emergency response by uavs in disaster areas. In ICCS, Part V, volume 14836 of LNCS, pages 150–165. Springer.

- Zitzler and Thiele (1999), Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation, 3(4):257–271.
- Zitzler, Eckart et al. (2001). Zitzler, Eckart, Laumanns, Marco, and Thiele, Lothar (2001). SPEA2: Improving the strength pareto evolutionary algorithm. TIK Report 103, ETH Zurich, Computer Engineering and Networks Laboratory.