

Minisum and maximin aerial surveillance over disjoint rectangles

Orhan Karasakal^{1,2}

Received: 7 April 2015 / Accepted: 12 April 2016 / Published online: 27 April 2016
© Sociedad de Estadística e Investigación Operativa 2016

Abstract The aerial surveillance problem (ASP) is finding the shortest path for an aerial surveillance platform that has to visit each rectangular area once and conduct a search in strips to cover the area at an acceptable level of efficiency and turn back to the base from which it starts. In this study, we propose a new formulation for ASP with salient features. The proposed formulation that is based on the travelling salesman problem enables more efficient use of search platforms and solutions to realistic problems in reasonable time. We also present a max–min version of ASP that maximizes the minimum probability of target detection given the maximum flight distance of an aerial platform. We provide computational results that demonstrate features of the proposed models.

Keywords Travelling salesman problem · Integer programming · Planning · Military · Search

Mathematics Subject Classification 90-02

1 Introduction

Surveillance is the systematic observation of aerospace, surface, or subsurface areas, places, persons, or things, by visual, aural, electronic, photographic, or other means (DOD Dictionary of Military Terms 2016). In aerial surveillance, the scope is reduced to an air platform such as an aircraft or an unmanned aerial vehicle (UAV) observes

✉ Orhan Karasakal
okarasakal@bilkent.edu.tr

¹ Industrial Engineering Department, Bilkent University, Ankara 06800, Turkey

² Decision Support Division, Turkish Navy HQ, 06100 Ankara, Turkey

sea or land areas using its sensors. Aerial surveillance has both military and civilian applications, showing the flag, gathering intelligence, monitoring illegal activities or terrorist acts, disaster relief, environmental monitoring and planetary exploration. Aerial surveillance can be costly. Flying a maritime patrol aircraft costs approximately \$ 30,000 per hour, including aircraft depreciation and the cost of surveillance equipment (Ng and Sancho 2009). Although operational cost including maintenance, oil, crew, expandable sensors and equipment is less than the value, it is still very important to minimize the cost of surveillance operations by minimizing total flight time or maximizing mission effectiveness. In this study, we consider the aerial surveillance problem (ASP), finding the shortest path for an aerial surveillance platform that has to visit a series of rectangular areas once and conduct a search in strips to cover each area to an acceptable level of efficiency and return to the base from which it started.

To date, the only work on the ASP is the original contribution by Ng and Sancho (2009) in which the problem was first introduced. They proposed a dynamic programming model for solving the ASP. However, their dynamic programming formulation is subject to the curse of dimensionality for problems larger than 6–7 rectangles.

There are several different studies available on various surveillance problems, but they do not take the search inside a rectangle into consideration. Panton and Elbers (1999) consider the problem of optimal mission planning for an air platform performing synthetic aperture radar (SAR) surveillance. The mission objective is to start from a base, scan a set of land stripes with variable length and orientation but fixed width and end at a possibly different base location. John et al. (2001) extend the earlier work in Panton and Elbers (1999) by addressing the question of how to find an optimal cover of swaths for a region. Ng and Ghanmi (2002) propose an automated surface surveillance system to determine the sequence of sub-regions to maximize the probability of target detection. They utilize probability maps, barrier patrol and develop a heuristic algorithm for searching irregular areas. Grob (2006) discusses maritime surface surveillance operation in a certain sea area and propose a simulation model to quantify the effectiveness of surveillance operations involving surface and air platforms such as frigates, helicopters and maritime patrol aircraft. Jacobson et al. (2006) address the problem of minimizing the overall search route cost for multiple platforms that visit every search location. They model the problem as a multiple travelling salesman problem (m-TSP) and propose simultaneous generalized hill climbing algorithm to solve it. Simonin et al. (2009) deal with optimizing the search effort by a number of sensors so as to maximize the probability of detection of a moving target. A hierarchical search model (i.e. allocation of sensors to search zones at the upper level and placement of search efforts on a continuous plane within given search zone in the lower level) is proposed to solve the problem. Yakıcı and Karasakal (2013) assume that vehicles spent time in the demand nodes according to their service speeds in a min–max vehicle routing problem setting. They address service quality within a demand node implicitly, while we address service quality, i.e. search efficiency within each area, explicitly. The orienteering problem has similar applications on aerial search. In the orienteering problem, the goal is to maximize the total collected fixed score on a path that visits some of the vertices. The length of the path is limited by a given resource. Interested readers are referred to Vansteenwegen et al. (2011) and Chekuri et al. (2012) for the orienteering and related problems. Pietz and Royset (2013)

generalized the orienteering problem with resource-dependent rewards. Instead of a fixed reward, reward level depends on the resource expended at each visited node. Note that not all vertices have to be visited and the resource expenditure within the node (i.e. entry and exit points of rectangular area in our problem) is not modeled explicitly. They applied their approach for smuggler search problem up to 10 targets successfully.

We developed two mixed integer linear programming models for ASP. The first model is for minimizing total distance travelled. It is a new treatment of the dynamic programming model proposed by [Ng and Sancho \(2009\)](#) with several extensions described below. The other is for maximizing minimum probability of target detection. This new model is a max–min version of ASP. Maximin ASP model is used for maximizing the utilization of the capacity of an air platform in surveillance operation.

Advances in technology enable aerial surveillance platforms such as UAV's to fly long hours. Thus, more disjoint rectangles are within reach of those platforms. Additionally, size of rectangles may affect the numbers of disjoint rectangles. There may be a lot more disjoint rectangles occupying approximately same size with several disjoint rectangles. Both the proposed formulations of ASP are solvable up to 40 disjoint rectangles.

We utilize the uniform random search to cover the disjoint rectangles. In uniform random search, an air platform covers a rectangular area by dividing the area along the length or width of the rectangle to equal sub regions, called strips. The air platform flies over the strip along the middle line. The distance between parallel paths of an air platform along two neighboring strips is called the track spacing. We propose a search schema that allows conducting search over exactly the same rectangular area. We allow different track spacing values for each rectangle. However, we force track spacing to have a value, which multiplied with an integer value equals to the width or length of the rectangle. For example, if width of the rectangle is 120 NM, then the track spacing values may be $120/4 = 30$ or $120/5 = 24$ NM. Allowing different track spacing values for each rectangle improves the efficiency of search within rectangles. We define the number of track spacing values depending on the performance requirement (i.e. the minimum probability of target detection). The minimum acceptable probability of target detection may be a standard value for all rectangles or specific to each rectangle. Thus, a planner may place more emphasis on a number of rectangles depending on the need of operation using the minimum acceptable probability of target detection parameter.

We reduce the number of decision variables in the proposed models by aggregating all entry and exit points on the border of each rectangle into eight entry or exit points, two around each corner along length and width of rectangle. Note that there are small differences between exact entry (exit) points for specific track spacing values when length (width) of rectangle is large compared to sweep width. Aggregating entry or exit points into eight points enables us to control the number of decision variables at a tractable level for large number of rectangles to search.

We also develop a problem generation schema to test the performance of the proposed models. Finally, the proposed mixed integer liner programming formulations let using the efficient commercial solvers.

The remainder of the paper is organized as follows: we present the problem description and formulations of ASP in the next section. In Sect. 3, we present a sample case that represents features of the proposed models. Computational results are given in Sect. 4. Remarks and insights are given in Sect. 5. Finally, Sect. 6 discusses benefits of the proposed models and future direction of research.

2 Problem description and model formulation

In ASP, an air platform visits each rectangle once and returns to the base where it starts. That is similar to TSP. The distinguishing characteristic of ASP from that of TSP is the nature of visit. In TSP, we generally consider visit as an instantaneous event takes place without consuming any resource. To mitigate resource consumption, one can add a node's constant event time to each arc entering a node. Here in ASP, visit consumes resource endogenously. We consider both the length of a visit in terms of distance and the efficiency of a visit in terms of the probability of target detection. As we consider a rectangular area instead of a point representing a city in TSP, where to start and to stop a visit also becomes important.

We calculate the probability of target detection, P using the uniform random search formula below (Wagner et al. 1999). The uniform random search formula is a conservative estimate and can be viewed as a lower bound on the probability of target detection in a rectangular area. We also assume that the position of target is uniformly distributed over a rectangular area and target is non-evading. In aerial surveillance operations, the speed of air platforms is large compared to the speed at sea and on land. Thus, we can justify those assumptions. However, the proposed formulations allow other ways of calculating the effectiveness of search in each region.

$$P = 1 - e^{-W/S}, \text{ where } W \text{ is sweep width and } S \text{ is track spacing.}$$

Sweep width is a measure of ability of detection. Sweep width depends on the object of search, capability of the sensor, and environmental conditions prevailing at the time and the place of search. Detection capability of sensors decreases as the distance between the sensor and the object of search increases. The lateral range curve defines the probability of target detection according to the lateral distance between a sensor and an object. The sweep width equals to the area under the lateral range curve. In uniform random search, a searcher wants to cover the area of interest uniformly by dividing the area along the length or width of the rectangle to equal strips. A searcher traverses strips along the middle lines. Thus, the distance between the parallel paths of an air platform along two neighboring strips that is called the track spacing equals to the width of strips. Figure 1 depicts the concept of uniform random search and related parameters. We also assume that target is uniformly distributed in the rectangle, not evading and $S \geq W$. Interested readers are referred to Wagner et al. (1999) for a detailed treatment of area search. Note that we ignore the turning radius of the air platform. This is a pessimistic approach compared to an actual flight path. For example, if an air platform has a turn radius of 0.5 NM, we assume a flight length of 1 NM while the actual flight is 0.785 NM. Turn radius changes depending on the type of air platform, its speed and turn bank angle. However, actual flight distance can be easily integrated in case of a

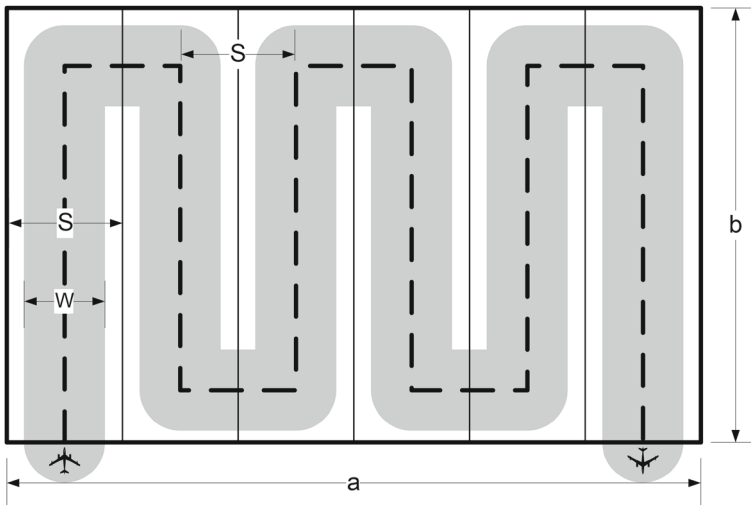


Fig. 1 Uniform random search of a rectangular area. The rectangular area is divided into six search strips along the width of the rectangle. Track spacing $S = a/6$. Distance travelled by the searcher equals to $6b$. We ignore the turning radius of the searcher

real-life application of the proposed models without changing any calculation other than the distance and the respective probability of target detection. These parameters are input for the proposed models and can be treated at the level of details that satisfy the need of the planner.

In Fig. 1, a searcher attempts to search the area using six tracks. Observe that the searcher leaves the area from the side of the rectangle it enters. If it would be five tracks instead of six, the searcher would leave the rectangle from the opposite of the side it entered. We utilize this observation for the development of the model.

Uniform random search assumes that the area of interest is systematically searched starting from a point on the border along the width or length of a rectangle. When an air platform searches an area equal to or larger than the area of interest, the mission is accomplished. From a practical point of view, using predetermined track spacing for all rectangles may be considered. However, the drawback of this approach is searching additional area that the searcher has no interest. Searching an area of no-interest is waste of time and resource. Additionally, boundaries of the area of interest, i.e. rectangles, may already have some slack to take into account the worst case scenario in planning. Up-to-date flight instruments made it easy to follow flight tracks with different track spacing for each rectangular search area. Thus, we allow using different track spacing values for each rectangle in proposed model. Allowing different track spacing for each rectangle enables us to search rectangles exactly and to use time and resources more efficiently.

The maximum probability of target detection for any rectangle is 0.63 using uniform random search formula with the assumption of $S \geq W$. Let \underline{P} be the minimum acceptable probability of target detection in rectangles. Note that \underline{P} is to satisfy the inequality, $\underline{P} \leq 1 - e^{-W/S} \leq 0.63$, for at least one track spacing for all rectangles. Otherwise the rectangle cannot be searched with the minimum acceptable probability

of target detection. The minimum acceptable probability of target detection may be different for each rectangle if required. Thus, the planner may place different levels of search effort for rectangles by modifying \underline{P} to \underline{P}_i and assigning a specific value for each rectangle i . Let t_{\max} and t_{\min} show the maximum and the minimum number of tracks, respectively. Assuming that the sweep width is considerably small compared to the length and the width of the rectangle, $W \ll a$ and $W \ll b$, we define t_{\max} and t_{\min} as follows:

$$t_{\max} = \left\lfloor \frac{a}{W} \right\rfloor$$

$$t_{\min} = \left\lceil \frac{-a \ln(1 - \underline{P})}{W} \right\rceil$$

We replace a with b for the width of the rectangle. Note that, although we assume that the sweep width is much smaller compared to the length and the width of rectangles, the same formulation is valid as long as the length and the width of rectangles are larger than the sweep width.

We have a different pair of entry and exit points for each number of search tracks. Figure 2 depicts sample search tracks emanating around entry point 7 for a rectangle

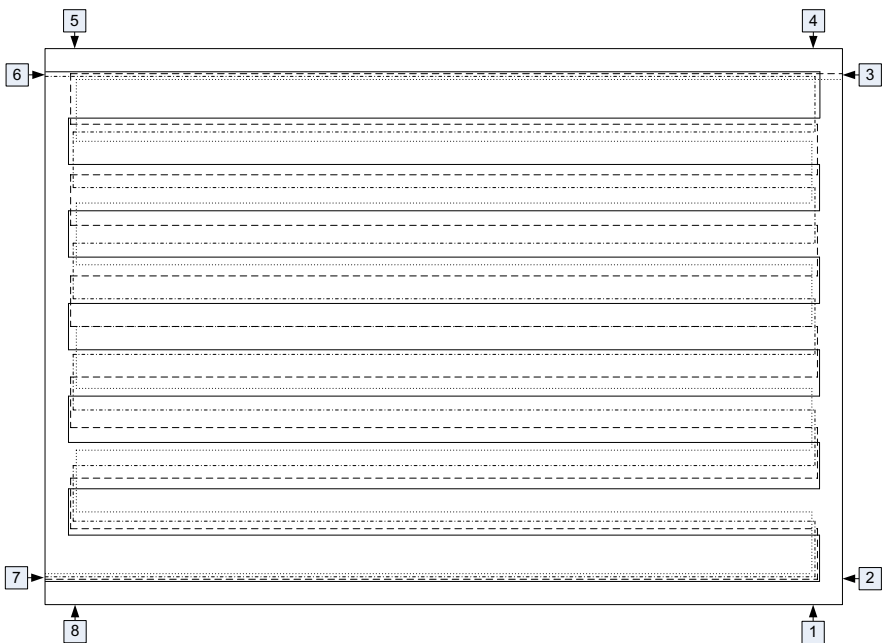


Fig. 2 Sample search tracks emanating from the entry point 7 for a rectangle with $a = 4.3$, $b = 3$, $W = 0.25$ and $\underline{P} = 0.5$. The search track with the continuous line has the maximum number of tracks, 12 and goes to the exit point 6 along the same side. The *dashed line* and the *dashed-dotted line* have 11 and 10 tracks, respectively. The search track with the dotted line has the minimum number of tracks, 9 and goes to the exit point 3 on the opposite side. The *numbers in the boxes* show the aggregated entry or exit points of the rectangle

with $a = 4.3$, $b = 3$, $W = 0.25$ and $\underline{P} = 0.5$. The search track with the continuous line has the maximum number of tracks, 12 and goes to exit point 6 along the same side. The dashed line and the dashed-dotted line have 11 and 10 tracks, respectively. The search track with the dotted line has the minimum number of tracks, nine and goes to exit point three on the opposite side. The entry point for the search pattern with 12 tracks is $b/2t = 3/(2 * 12) = 0.125$ in from the corner point along the width of the rectangle, where t is the number of tracks. Exact entry points for the patterns with 11, 10 and 9 tracks are 0.136, 0.15 and 0.167 in from the corner of the rectangle, respectively. Assuming values for the rectangle are 1/100 of the realistic size (i.e. a rectangle 430 nautical miles by 300 nautical miles), the largest difference is 4.2 nautical miles for 4 different track spacing. Using the exact entry and exit points for each track spacing may easily expand the model size to an intractable level. Thus, we ignore the small error in the distance from one exit point of a rectangle to an entry point of another rectangle. We reduce the number of decision variables in the proposed models by aggregating all entry and exit points on the border of each rectangle into eight entry or exit points, two around each corner along the length and the width of a rectangle. We calculate the place of each aggregated entry or exit point by taking the average of half of the track spacing for all search patterns emanating from the same side of the rectangle around each corner. Representation of aggregated entry or exit points of each rectangle is shown in Fig. 2.

In ASP, we minimize the total distance travelled by an air platform, while satisfying the minimum acceptable probability of target detection. Note that the track spacing values for each rectangle are chosen in advance to satisfy the minimum desired probability of target detection. Thus, this model produces a tour that satisfies the minimum probability of target detection for each rectangle. In maximin ASP, we maximize the minimum probability of target detection for all the rectangles, while satisfying the maximum distance an air platform can fly. We refer ASP as minisum ASP to distinguish it from maximin ASP thereafter.

We present the minisum ASP model (P1) below:

Indices:

- i, j : index of disjoint rectangles, $i = 1$ and $j = 1$ represent the starting point. g and f are aliases of i and j . N denotes the total number of disjoint rectangles plus the starting point.
- l, k : index of entry and exit points of rectangles, $l = 1, \dots, 8$ and $k = 1, \dots, 8$.
- s : index of the search pattern within a rectangle for each feasible combination of entry and exit points of the rectangle.

Data:

- D_{iljk} : distance from point l of rectangle i to point k of rectangle j .
- D'_{ilks} : distance from point l to point k of rectangle i using search pattern s .
- P_{ilks} : probability of detecting target by flying from point l to point k in the rectangle i using search pattern s .
- u_i : node potential of rectangle i that indicates the order of the corresponding rectangle in the tour.

Decision variables:

$$y_{iljk} = \begin{cases} 1, & \text{if the air platform flies from point } l \text{ of rectangle } i \text{ to point } k \text{ of rectangle } j \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ilks} = \begin{cases} 1, & \text{if the air platform flies from point } l \text{ to point } k \text{ of rectangle } i \text{ using} \\ & \text{search pattern } s \\ 0, & \text{otherwise} \end{cases}$$

$$(P1) \quad \text{Min} \quad \sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \quad (1)$$

Subject to

$$\sum_{ilk} y_{iljk} = 1 \quad \forall j = 1, \dots, N \quad (2)$$

$$\sum_{jlk} y_{iljk} = 1 \quad \forall i = 1, \dots, N \quad (3)$$

$$\sum_{lks} z_{ilks} = 1 \quad \forall i = 2, \dots, N \quad (4)$$

$$\sum_{ks} z_{ilks} \leq \sum_{jk} y_{jkil} \quad \forall l = 1, \dots, 8, \quad i = 2, \dots, N \quad (5)$$

$$\sum_{ls} z_{ilks} \leq \sum_{jl} y_{ikjl} \quad \forall k = 1, \dots, 8, \quad i = 2, \dots, N \quad (6)$$

$$u_i - u_j + N \sum_{lk} y_{iljk} \leq N - 1 \quad \forall i = 2, \dots, N \quad j = 2, \dots, N \quad i \neq j \quad (7)$$

$$y_{iljk} \in \{0, 1\} \quad \forall iljk \in \{(i, l, j, k) | i \neq j, i = 1 \wedge l = 1, j = 1 \wedge k = 1\} \quad (8)$$

$$z_{ilks} \in \{0, 1\} \quad \forall ilks \in \{(i, l, k, s) | i \neq 1, l \neq k\} \quad (9)$$

$$u_i \geq 0 \quad \forall i = 1, \dots, N \quad (10)$$

The objective function (1) expresses the total distance travelled. The first part of the equation accounts for the travel between rectangles including the origin. The second part is the total distance travelled within rectangles. Constraint sets (2) and (3) restrict the air platform to enter and to leave each rectangle and the origin only once, respectively. Constraint set (4) allows a single search pattern for each rectangular area. Constraint sets (5) and (6) ensure that a search pattern is chosen only if there is an entry in the starting point and there is an exit in the ending point of the pattern. Constraint set (7) is a slightly modified subtour elimination constraint (i.e. MTZ constraints) due to Miller et al. (1960). Constraints (8) and (9) enforce binary restrictions and constraint (10) enforce nonnegativity restrictions on decision variables. Note that if a search pattern from point l to point k is possible, $D'_{ilks} > 0$, otherwise D'_{ilks} is set to infinity (or a sufficiently large number) and the respective decision variable, z_{ilks} is removed from the model to keep the number of decision variables as small as possible.

Constraint set (5) and (6) can be replaced by the following constraint set:

$$\sum_s z_{ilks} \leq \frac{1}{2} \left(\sum_{jf} y_{ikjf} + \sum_{jg} y_{jgil} \right) \quad \forall ilk, i \neq 1, l \neq k \quad (11)$$

Constraint set (5) and (6) add only 16 constraints for each rectangle, while constraint set (11) adds 32 constraints. However, constraint set (5) and (6) are tighter compared to constraint set (11).

We introduce the following parameter before presenting maximin ASP (P2):

M : maximum distance the air platform can fly.

$$(P2) \quad \text{Max } \alpha \quad (12)$$

Subject to

Constraints (2)–(10)

$$\sum_{iljk} D_{iljk} y_{iljk} + \sum_{ilks} D'_{ilks} z_{ilks} \leq M \quad (13)$$

$$\sum_{lks} P_{ilks} z_{ilks} \geq \alpha \quad \forall i \quad (14)$$

$$\alpha \geq 0 \quad (15)$$

Constraint (12) maximizes the minimum probability of target detection. Constraint set (13) limits the maximum flight distance of the air platform. Constraint set (14) expresses that the decision variable, α , should be less than or equal to the minimum probability of target detection of all rectangular areas. Constraint (15) enforces non-negativity restriction on decision variable α .

3 Sample case

In this section, we present the solution for the sample problem taken from [Ng and Sancho \(2009\)](#). We compare our solution with that of [Ng and Sancho \(2009\)](#), where suitable. The sample problem is depicted in Fig. 3. Dimensions have been scaled to 1/100 of the original problem. In the sample case, there are three disjoint rectangles to be searched. The searcher starts from the origin and returns back. Sweep width is set to 0.25. The largest allowable track spacing in [Ng and Sancho \(2009\)](#), $S = 0.32$, produces a probability of target detection $P = 0.54$. Thus, we use $\underline{P} = 0.54$.

We generated 28 search patterns for the rectangle 1, 20 search patterns for the rectangle 2 and 36 search patterns for the rectangle 3 that have a probability of target detection value larger than 0.55. The solution of the sample problem using the model (P1) is depicted in Fig. 4. The air platform starts its surveillance from the origin and flies to the closest corner of the rectangle 3. It enters the rectangle 3 along the length and follows a pattern parallel to the width of the rectangle with 18 tracks and leaves the rectangle 3 along the same edge. It enters the rectangle 2 along the length and follows a search track parallel to the width of the rectangle with 14 tracks and

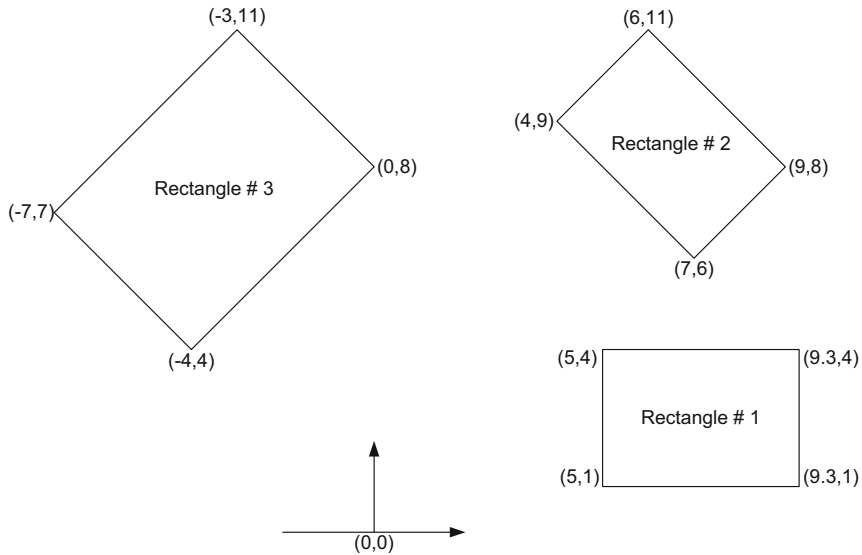


Fig. 3 The sample problem taken from [Ng and Sancho \(2009\)](#). Dimensions rescaled to 1/100 of the original problem. The searcher starts from the origin and returns back

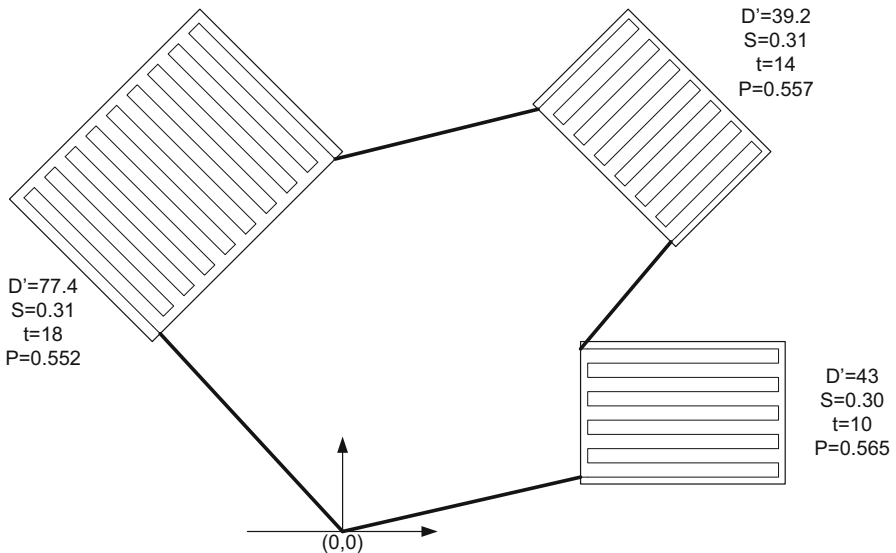


Fig. 4 The solution of the sample problem using model (P1). This solution attains a total cruise length of 177.65 and an overall detection probability of 0.174. The length of the search pattern, the track spacing, the number of tracks and the probability of target detection are given next to each corresponding rectangle. Solution (P1) produces similar patterns with that of [Ng and Sancho \(2009\)](#)

leaves the rectangle 2 from the same edge it enters. In the rectangle 1, it flies a search pattern parallel to the length of the rectangle with 10 tracks, leaves the rectangle 1 from the side it enters and flies back to the origin. The opposite flight direction is

also possible. The total cruise length of this flight is 177.65. Ng and Sancho (2009) uses the product of target detection probabilities of disjoint rectangles as a measure of surveillance efficiency. To be able to compare results, we use the same measure and name it the overall detection probability. Our surveillance plan produces an overall detection probability of 0.174.

Our optimal solution is similar with that of the solution produced by the dynamic programming formulation of ASP proposed by Ng and Sancho (2009). Their solution has a total cruise length of 176.89 and an overall detection probability of 0.1697. We produce a slightly larger overall detection probability. Our solution proposes a track spacing value of 0.31 for the rectangle 3 instead of 0.32 of Ng and Sancho (2009)'s solution. Restricting the aerial surveillance within the rectangles enables us to attain lower track spacing values that produce larger probabilities of target detection.

4 Computational results

In the preceding section, we solved the minisum ASP for a small sample problem taken from Ng and Sancho (2009). In this section, we present a problem generation schema, a detailed solution of a sample problem with 9 disjoint rectangles generated using the proposed problem generation schema and computational results for proposed models. We developed a framework for generating disjoint rectangles randomly and present computational results of randomly generated test problems.

We drew 60 disjoint rectangles on a plane with a length of 29 and a width of 21 randomly. We assume that the rectangles and the plane have a 1/100 scale. Rectangles have different lengths and widths. Without loss of generality, we assume that all rectangles are positioned parallel to both axis for computational simplicity. X and Y coordinates of corners of rectangles are given in Appendix 1. We present the layout of rectangles over the plane in Appendix 2.

We coded models and the test framework in GAMS (general algebraic modeling language) (Brook et al. 1996) and solved models using CPLEX MIP solver version 11.2.1. We choose the specified number of rectangles, 5, 10, 20, 30 or 40 out of 60 randomly. We used two different values, 0.25 and 0.35 for the sweep width, and three different values, 0.50, 0.40 and 0.30 for the minimum acceptable probability of target detection. That setting created $5 \times 2 \times 3 = 30$ different combinations. We solved a total of 150 problems, 5 different problems created by different random number streams for each setting.

We present detailed solution of a problem instance generated using the proposed problem generation schema to give insight on the resulting search patterns with respect to the sample case presented in the previous section. A problem instance with 9 rectangles is generated using the proposed problem generation schema. The sweep width and the minimum probability of target detection are set to 0.25 and 0.30, respectively. The model (P1) with those settings produced a total cruise length of 78.5274 (Fig. 5). If we increase the minimum probability of target detection to 0.40, the total cruise length increases to 105.0135. If we further increase the minimum probability of target detection to 0.50 and 0.55, the total cruise length increases to 125.4282 and 142.0529, respectively. The model (P2) produces a minimum probability of target

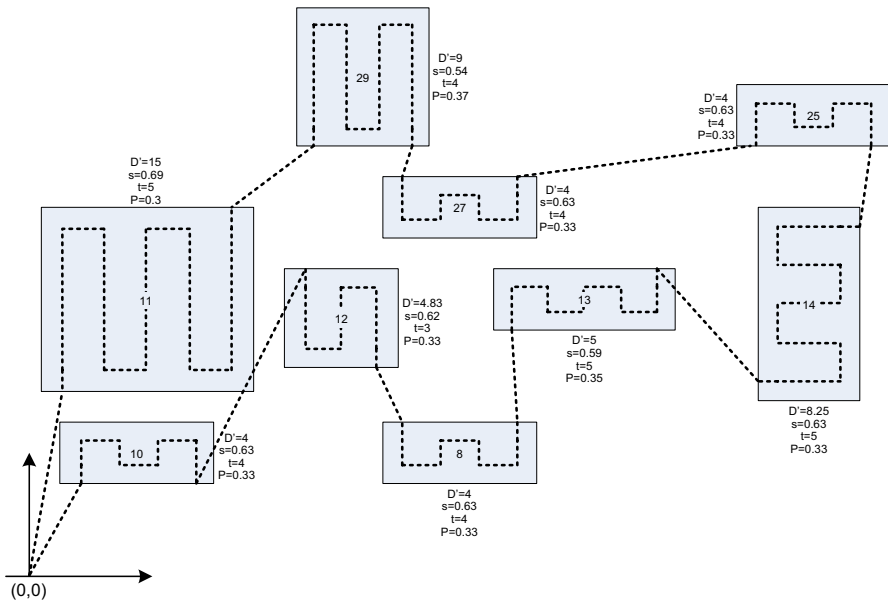


Fig. 5 A problem instance with 9 rectangles generated using the proposed problem generation schema. The sweep width and the minimum probability of target detection are set to 0.25 and 0.30, respectively. The model (P1) with these settings produced a total cruise length of 78.5274. Note that the *numbers within rectangles* represent the identity of the rectangle in the problem generation schema

detection, 0.6117 where the maximum distance an air platform can fly is set to 180. That solution yields a total cruise length of 177.823.

The model (P2) maximizes the minimum probability of target detection while satisfying the maximum distance the air platform can fly. The maximum distance the air platform can fly is the right-hand side value for the constraint (13). The constraint (13) is expected to be binding in an optimal solution. If it is not binding, the solution to the model (P2) becomes trivial. That is choosing the search pattern with the largest probability of target detection for each rectangle and selecting any feasible tour combining the entry and the exit points of the chosen search patterns. Thus, determination of the right-hand side values for different number of rectangles, different position and size of rectangles require trial and error type analysis. Finding suitable numbers for right-hand side values by trial and error adds only superfluous details to our analysis. Here, we want to show that any practical problem size can be solved using the proposed models. Thus, we solved the generated test problems using the minisum version of ASP and reported the results of the model (P2) for five instances of the problem with 40 rectangles, 0.25 sweep width and 0.5 minimum probability of target detection.

Our empirical experimentation with GAMS showed that the problem (P1) with constraint set (11) solves faster than the formulation with constraint set (5) and (6). Thus, we chose to use the modified formulation. We set the absolute and the relative termination criteria for the solver to 0. This enables us to report time to reach the true optimal for each instance. CPLEX solver can utilize parallel processing depending on the hardware. We used 4 parallel threads for solving test problems.

Computational results are reported in Table 1. For each problem combination, we report minimum, maximum and mean computational time in seconds. The largest elapsed time is less than 15 seconds for problems with 10 or less rectangles. The largest instance with 40 rectangles, 0.25 sweep width and 0.30 probability of detection produced results in less than 11 min. The largest solution time is less than 56 min for the case with 40 rectangles, 0.35 sweep width and 0.30 probability of target detection. Since increase in W and P allows less search patterns for each rectangle, it is reasonable to think that solution time on average decreases as the sweep width and the probability of target detection increase. This reasoning did not hold for 30 and 40 rectangles problem sets with $W = 0.25$ and $P = 0.30$.

We solved the model (P2) for 5 instances of the problem with 40 rectangles, 0.25 sweep width and 0.50 probability of detection. The maximum distance an air platform can fly, $M = 1800$, was used. Average CPU time in seconds was 445.24 for 5 test instances. The minimum and the maximum solution times were 30.48 and 918.31 seconds, respectively.

We expanded the computational testing to establish a problem size that can be reasonably solved at different tolerances. We tried to solve problems with 50 rectangles and different parameter settings. We set a time limit of one hour. We report results including the optimality or a non-optimal solution because of the time limit/the memory limit with the optimality gap. The results are depicted in Table 2. Results show that the performance of the solver reduces substantially. The CPLEX solver fails to produce any reasonable solution for many different problem settings. Ng and Sancho (2009) argue that the size of surveillance problem for a manned aerial vehicle is normally restricted to only 5 or 6 rectangles. However, geographical restrictions on the size of rectangles and capabilities of long endurance UAV systems may lead to have more disjoint rectangles to search in realistic scenarios. We can reasonably assume that 40 rectangles will cover a large portion of real problem instances. Thus, development of other solution approaches such as developing bounds for better guidance of the solver's branch and bound procedure or heuristics is not required. However, efficient heuristics developed for the TSP can be a good starting point, if required.

5 Insights

The planner may place more emphasis to certain area of interests by adjusting the minimum acceptable probability of target detection for those rectangles. That can be done by simply modifying the minimum acceptable probability of target detection, P to P_i and assigning a specific value for each rectangle i .

A surveillance operation can be planned in strips over rectangular areas using the proposed approach. This enables us to calculate the effectiveness of search using the uniform random search formula. For an irregular shaped area of interest, we can calculate the smallest circumscribed rectangle (see Zhang et al. 2013) and use the proposed models. The proposed models can be used to search over non-rectangular regions as well. One needs to define the entry and the exit points and probability of target detection for each search pattern connecting an entry and an exit point. However,

Table 1 Computational results for the problem (P1). The proposed model can solve problems up to 40 rectangles in reasonable time

| # of rectangles | Sweep width (W) | \underline{P} | Elapsed time (s) ^a | | |
|-----------------|-----------------|-----------------|-------------------------------|---------|---------|
| | | | Min | Ave | Max |
| 5 | 0.25 | 0.5 | 0.08 | 0.16 | 0.32 |
| 5 | 0.25 | 0.4 | 0.13 | 0.16 | 0.22 |
| 5 | 0.25 | 0.3 | 0.10 | 0.12 | 0.13 |
| 5 | 0.35 | 0.5 | 0.11 | 0.14 | 0.17 |
| 5 | 0.35 | 0.4 | 0.11 | 0.13 | 0.14 |
| 5 | 0.35 | 0.3 | 0.11 | 0.12 | 0.13 |
| 10 | 0.25 | 0.5 | 0.45 | 3.15 | 9.89 |
| 10 | 0.25 | 0.4 | 0.39 | 1.05 | 3.22 |
| 10 | 0.25 | 0.3 | 0.19 | 3.12 | 14.29 |
| 10 | 0.35 | 0.5 | 0.37 | 0.47 | 0.68 |
| 10 | 0.35 | 0.4 | 0.27 | 2.03 | 3.99 |
| 10 | 0.35 | 0.3 | 0.11 | 2.28 | 9.81 |
| 20 | 0.25 | 0.5 | 3.24 | 28.99 | 64.41 |
| 20 | 0.25 | 0.4 | 3.90 | 28.52 | 93.49 |
| 20 | 0.25 | 0.3 | 8.17 | 50.42 | 90.53 |
| 20 | 0.35 | 0.5 | 9.20 | 31.77 | 80.50 |
| 20 | 0.35 | 0.4 | 2.86 | 16.85 | 39.16 |
| 20 | 0.35 | 0.3 | 7.88 | 65.92 | 138.87 |
| 30 | 0.25 | 0.5 | 20.71 | 69.29 | 191.99 |
| 30 | 0.25 | 0.4 | 8.55 | 278.63 | 933.54 |
| 30 | 0.25 | 0.3 | 3.12 | 82.96 | 266.85 |
| 30 | 0.35 | 0.5 | 37.59 | 184.80 | 518.93 |
| 30 | 0.35 | 0.4 | 36.45 | 135.57 | 206.25 |
| 30 | 0.35 | 0.3 | 94.51 | 510.57 | 1515.60 |
| 40 | 0.25 | 0.5 | 410.00 | 691.65 | 1057.76 |
| 40 | 0.25 | 0.4 | 345.25 | 754.98 | 1450.08 |
| 40 | 0.25 | 0.3 | 42.28 | 334.75 | 655.38 |
| 40 | 0.35 | 0.5 | 107.30 | 379.83 | 555.58 |
| 40 | 0.35 | 0.4 | 101.86 | 497.52 | 877.13 |
| 40 | 0.35 | 0.3 | 369.45 | 1788.08 | 3321.46 |

^a Elapsed time for a personal computer with core i5 CPU and 4 GB of RAM

it may be difficult to plan the search pattern inside an irregular area. It is generally a requirement to be able to have a preplanned routing for unmanned aerial vehicles. For a manned surveillance mission, alternative entry and exit points and duration of a non-uniform search over the region may be used to calculate the effectiveness of search for each alternative mission duration, and the optimum flight path can be calculated.

Table 2 Computational results for the problem (P1) with 50 rectangles. The CPLEX solver failed to solve those problems to through optimality because of either time or memory limit. In six out of 12 problems, relative gaps were less than 10 %

| Sweep width (W) | P | Elapsed time (s) ^a | Absolute gap | Relative gap | Out of memory ^b |
|-----------------|-----|-------------------------------|--------------|--------------|----------------------------|
| 0.25 | 0.3 | 2627 | 28.24 | 6.89 | * |
| | | 2574 | 451.77 | 53.48 | * |
| 0.25 | 0.4 | 2648 | 150.52 | 22.35 | * |
| | | 2787 | 1.96 | 0.37 | * |
| 0.25 | 0.5 | 3600 | 119.38 | 15.77 | |
| | | 2630 | 341.39 | 34.15 | * |
| 0.35 | 0.3 | 3600 | 10.99 | 3.35 | |
| | | 3600 | 22.11 | 6.49 | |
| 0.35 | 0.4 | 2866 | 16.43 | 4.00 | * |
| | | 2552 | 383.92 | 48.84 | * |
| 0.35 | 0.5 | 3600 | 1.36 | 0.30 | |
| | | 2939 | 309.66 | 38.40 | * |

^a Elapsed time for a personal computer with core i5 CPU and 4 GB of RAM

^b Solver terminated because of the 4 GB RAM limit of the computer prematurely

We utilize a slightly modified MTZ constraints (Miller et al. 1960) for subtour elimination in both models. MTZ subtour elimination constraints allow assigning a kind of priority to a subset of disjoint rectangles. If it is preferable to search, say, rectangle i early in the operation, one can easily model this by adding a term au_i with $a > 0$ to the objective function of the problem (P1). However, applying a similar approach to the problem (P2) may require detailed fine tuning of parameters in the objective function.

If the minimum acceptable probability level is not carefully set by considering the maximum flight distance of the air platform, the problem (P2) may be infeasible. We may elasticize the model (P2) to handle infeasibility. This may be achieved by modifying the constraint (13) to include a slack variable and penalize the use of slack variable in the objective if the maximum flight distance is exceeded. However, the resulting solution can still have a total distance exceeding the predefined maximum distance. As such, in practice the air platform might have to abort the mission before completion of the total tour. This solution provides insight into how to adjust the minimum detection probability levels to obtain a tour that is feasible with respect to the maximum flight distance of the air platform, and thus can be fully completed in reality.

We implicitly assumed symmetry in our models in terms of the flight distance. Asymmetry can easily be accommodated by suitably assigning values to the parameters D and D' .

Aggregating the entry or the exit points into 8 enables us to control the number of decision variables at a tractable level. Thus, we can solve problems up to 40 rectangles using a standard solver. To understand the effect of aggregation, we can analyse the number of flight paths in three groups, i.e. within rectangles, between rectangles and

between the origin and rectangles. We represent the flight paths within each rectangle using the entry point, the exit point and the index of search pattern. For example, if there are two different search patterns between each entry and exit points, there would be 16 different search patterns in total. If there would be no aggregation, 16 different search patterns would be represented by 16 entry/exit points around a rectangle. All search patterns can be represented both ways, as expected. There would be $(N - 1) * 8 * (N - 2) * 8$ flight paths between rectangles, where N is the total number of rectangles including the origin. There would be $(N - 1) * S * 8 * (N - 2) * S * 8$ flight paths between rectangles without considering aggregation, when there are S search patterns between each entry/exit point combination under aggregation. From the origin to rectangles, there are $2 * (N - 1) * 8$ flight paths, when aggregation is considered. There would be $2 * (N - 1) * S * 8$ flight paths without aggregation. With aggregation, the number of flight paths increases on the order of $O(N^2)$. The number of flight paths increases on the order of $O(S^2 N^2)$, when no aggregation is considered. When the number of search pattern increases, benefit of aggregation increases proportionally with square of the number of search patterns.

6 Conclusion

Time and resource management are always important in any operation, and intelligently managing aerial surveillance operations pays off by substantial cost saving or improved efficiency while consuming a similar level of resources.

This article presents two optimization models for managing the aerial surveillance operation over disjoint rectangles. The first model is for minimizing the total distance travelled by the air platform, while satisfying the minimum acceptable probability of target detection and the other is for maximizing the minimum probability of target detection for all rectangles, while satisfying the maximum distance the air platform can fly.

The proposed solution approach for aerial surveillance can be enhanced along several directions: We solved the proposed models using default settings of CPLEX solver. If one needs to solve a problem with the number of disjoint rectangles larger than 40, one way to go would be customizing the solver for our models. MTZ subtour elimination constraints are used in the proposed models. Subtour elimination constraints introduced by [Dantzig et al. \(1954\)](#) may be used to solve the proposed models. In such a case, the constraint set (5) and (6) can effectively be used instead of the constraint set (11), as they produce a smaller polytope compared to the constraint set (11). Developing heuristics for large size problem instances may be an option for a last resort to use. Optimally assigning more than one air platform for aerial surveillance would be a substantial extension to the present level of modeling for the aerial surveillance.

Appendix 1

See Table 3.

Table 3 X–Y coordinates of corners of rectangles used for problem generation

| Rectangle | Ax | Ay | Bx | By | Cx | Cy | Dx | Dy |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | 28.75 | 1.50 | 28.75 | 2.50 | 26.25 | 2.50 | 26.25 | 1.50 |
| 2 | 25.35 | 0.38 | 25.35 | 3.50 | 24.50 | 3.50 | 24.50 | 0.38 |
| 3 | 23.65 | 1.50 | 23.65 | 2.50 | 21.15 | 2.50 | 21.15 | 1.50 |
| 4 | 21.00 | 0.50 | 21.00 | 1.50 | 19.35 | 1.50 | 19.35 | 0.50 |
| 5 | 18.55 | 1.50 | 18.55 | 2.50 | 16.05 | 2.50 | 16.05 | 1.50 |
| 6 | 15.50 | 0.25 | 15.50 | 2.50 | 14.15 | 2.50 | 14.15 | 0.25 |
| 7 | 12.25 | 0.50 | 12.25 | 1.50 | 9.75 | 1.50 | 9.75 | 0.50 |
| 8 | 8.25 | 1.50 | 8.25 | 2.50 | 5.75 | 2.50 | 5.75 | 1.50 |
| 9 | 5.75 | 0.20 | 5.75 | 1.20 | 3.25 | 1.20 | 3.25 | 0.20 |
| 10 | 3.00 | 1.50 | 3.00 | 2.50 | 0.50 | 2.50 | 0.50 | 1.50 |
| 11 | 3.65 | 3.00 | 3.65 | 6.00 | 0.20 | 6.00 | 0.20 | 3.00 |
| 12 | 6.00 | 3.39 | 6.00 | 5.00 | 4.15 | 5.00 | 4.15 | 3.39 |
| 13 | 10.50 | 4.00 | 10.50 | 5.00 | 7.55 | 5.00 | 7.55 | 4.00 |
| 14 | 13.50 | 2.85 | 13.50 | 6.00 | 11.85 | 6.00 | 11.85 | 2.85 |
| 15 | 17.15 | 3.00 | 17.15 | 5.00 | 14.85 | 5.00 | 14.85 | 3.00 |
| 16 | 20.50 | 3.33 | 20.50 | 5.50 | 18.00 | 5.50 | 18.00 | 3.33 |
| 17 | 22.65 | 4.00 | 22.65 | 5.00 | 21.50 | 5.00 | 21.50 | 4.00 |
| 18 | 28.40 | 4.00 | 28.40 | 6.52 | 25.50 | 6.52 | 25.50 | 4.00 |
| 19 | 28.30 | 7.00 | 28.30 | 9.25 | 27.45 | 9.25 | 27.45 | 7.00 |
| 20 | 26.50 | 7.00 | 26.50 | 8.00 | 24.00 | 8.00 | 24.00 | 7.00 |
| 21 | 23.20 | 7.00 | 23.20 | 9.25 | 22.35 | 9.25 | 22.35 | 7.00 |
| 22 | 21.50 | 6.50 | 21.50 | 9.00 | 19.00 | 9.00 | 19.00 | 6.50 |
| 23 | 18.10 | 7.00 | 18.10 | 9.25 | 16.00 | 9.25 | 16.00 | 7.00 |
| 24 | 15.50 | 6.00 | 15.50 | 9.25 | 14.65 | 9.25 | 14.65 | 6.00 |
| 25 | 14.00 | 7.00 | 14.00 | 8.00 | 11.50 | 8.00 | 11.50 | 7.00 |
| 26 | 10.35 | 7.00 | 10.35 | 10.57 | 9.00 | 10.57 | 9.00 | 7.00 |
| 27 | 8.25 | 5.50 | 8.25 | 6.50 | 5.75 | 6.50 | 5.75 | 5.50 |
| 28 | 7.80 | 7.00 | 7.80 | 13.00 | 6.95 | 13.00 | 6.95 | 7.00 |
| 29 | 6.50 | 7.00 | 6.50 | 9.25 | 4.35 | 9.25 | 4.35 | 7.00 |
| 30 | 4.00 | 7.00 | 4.00 | 9.25 | 0.50 | 9.25 | 0.50 | 7.00 |
| 31 | 2.83 | 10.00 | 2.83 | 13.15 | 1.18 | 13.15 | 1.18 | 10.00 |
| 32 | 6.68 | 11.50 | 6.68 | 12.50 | 3.33 | 12.50 | 3.33 | 11.50 |
| 33 | 11.30 | 11.50 | 11.30 | 12.50 | 8.80 | 12.50 | 8.80 | 11.50 |
| 34 | 14.00 | 9.00 | 14.00 | 11.00 | 11.50 | 11.00 | 11.50 | 9.00 |

Table 3 continued

| Rectangle | Ax | Ay | Bx | By | Cx | Cy | Dx | Dy |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|
| 35 | 16.45 | 11.50 | 16.45 | 12.50 | 13.95 | 12.50 | 13.95 | 11.50 |
| 36 | 19.50 | 10.00 | 19.50 | 11.50 | 17.00 | 11.50 | 17.00 | 10.00 |
| 37 | 23.00 | 10.50 | 23.00 | 11.50 | 20.50 | 11.50 | 20.50 | 10.50 |
| 38 | 27.00 | 10.00 | 27.00 | 11.00 | 24.50 | 11.00 | 24.50 | 10.00 |
| 39 | 29.00 | 11.50 | 29.00 | 14.65 | 27.35 | 14.65 | 27.35 | 11.50 |
| 40 | 26.25 | 12.50 | 26.25 | 13.50 | 23.75 | 13.50 | 23.75 | 12.50 |
| 41 | 23.00 | 13.67 | 23.00 | 14.50 | 22.05 | 14.50 | 22.05 | 13.67 |
| 42 | 21.25 | 12.50 | 21.25 | 13.50 | 18.75 | 13.50 | 18.75 | 12.50 |
| 43 | 15.33 | 13.35 | 15.33 | 16.50 | 13.68 | 16.50 | 13.68 | 13.35 |
| 44 | 12.65 | 12.85 | 12.65 | 16.00 | 11.00 | 16.00 | 11.00 | 12.85 |
| 45 | 10.05 | 13.93 | 10.05 | 18.00 | 8.40 | 18.00 | 8.40 | 13.93 |
| 46 | 7.08 | 13.50 | 7.08 | 14.50 | 5.93 | 14.50 | 5.93 | 13.50 |
| 47 | 4.90 | 13.93 | 4.90 | 18.50 | 3.25 | 18.50 | 3.25 | 13.93 |
| 48 | 2.25 | 13.93 | 2.25 | 17.08 | 0.60 | 17.08 | 0.60 | 13.93 |
| 49 | 3.25 | 19.00 | 3.25 | 20.00 | 0.75 | 20.00 | 0.75 | 19.00 |
| 50 | 7.73 | 15.65 | 7.73 | 20.75 | 5.80 | 20.75 | 5.80 | 15.65 |
| 51 | 11.00 | 18.50 | 11.00 | 20.00 | 8.50 | 20.00 | 8.50 | 18.50 |
| 52 | 13.65 | 17.50 | 13.65 | 20.50 | 12.50 | 20.50 | 12.50 | 17.50 |
| 53 | 17.00 | 17.00 | 17.00 | 20.50 | 15.35 | 20.50 | 15.35 | 17.00 |
| 54 | 20.00 | 13.93 | 20.00 | 18.70 | 17.50 | 18.70 | 17.50 | 13.93 |
| 55 | 24.00 | 15.50 | 24.00 | 19.00 | 20.10 | 19.00 | 20.10 | 15.50 |
| 56 | 27.60 | 16.30 | 27.60 | 17.30 | 24.10 | 17.30 | 24.10 | 16.30 |
| 57 | 29.43 | 15.53 | 29.43 | 18.68 | 27.78 | 18.68 | 27.78 | 15.53 |
| 58 | 29.00 | 19.00 | 29.00 | 20.00 | 26.50 | 20.00 | 26.50 | 19.00 |
| 59 | 25.35 | 19.70 | 25.35 | 20.70 | 21.85 | 20.70 | 21.85 | 19.70 |
| 60 | 20.08 | 19.80 | 20.08 | 20.63 | 19.13 | 20.63 | 19.13 | 19.80 |

Appendix 2

See Fig. 6.

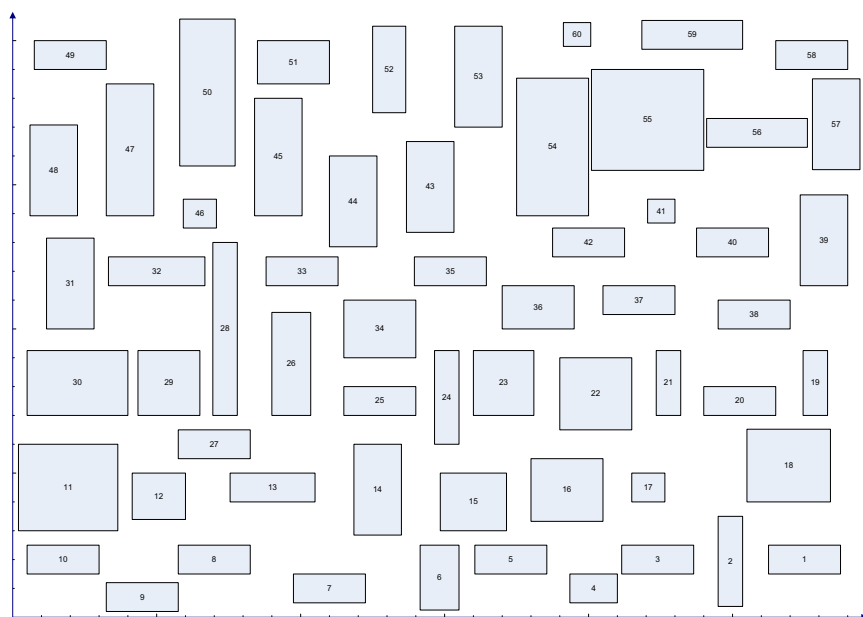


Fig. 6 The layout of 60 disjoint rectangles used to generate sample test problems

References

- Brook A, Kendrik D, Meeraus A (1996) GAMS: a user's guide. Body & Fraser Publishing Company, Massachusetts
- Chekuri C, Korula N, Pal M (2012) Improved algorithms for orienteering and related problems. *ACM Trans Algorithms* 8(3):1–27
- Dantzig GB, Fulkerson DR, Johnson SM (1954) Solution of a large-scale travelling salesman problem. *Oper Res* 2:393–410
- DOD Dictionary of Military Terms (2016) http://dtic.mil/doctrine/dod_dictionary/index.html. Accessed 9 Apr 2016
- Grob MJHB (2006) Routing of platforms in a maritime surface surveillance operation. *Eur J Oper Res* 170:613–628
- Jacobson SH, McLay LA, Hall SN, Henderson D, Vaughan DE (2006) Optimal search strategies using simultaneous generalized hill climbing algorithms. *Math Comput Model* 43:1061–1073
- John M, Panton D, White K (2001) Mission planning for regional surveillance. *Ann Oper Res* 108:157–173
- Miller CE, Tucker AW, Zemlin RA (1960) Integer programming formulation of traveling salesman problems. *J Assoc Comput Mach* 7:326–329
- Ng KYK, Ghanmi A (2002) An automated surface surveillance system. *J Oper Res Soc* 53:697–708
- Ng KYK, Sancho NFG (2009) Regional surveillance of disjoint rectangles: a travelling salesman formulation. *J Oper Res Soc* 60:215–220
- Panton DM, Elbers AW (1999) Mission planning for synthetic aperture radar surveillance. *Interfaces* 29:73–88
- Pietz J, Royset JO (2013) Generalized orienteering problem with resource dependent rewards. *Nav Res Logist* 60(4):294–312
- Simonin C, Le Cadre JP, Dambreville F (2009) A hierarchical approach for planning a multisensor multizone search for a moving target. *Comput Oper Res* 36:2179–2192
- Vansteenkoven P, Souffriau W, Oudheussen DV (2011) The orienteering problem: a survey. *Eur J Oper Res* 209(1):1–10

- Wagner DH, Mylander WC, Sanders TJ (1999) Naval operations analysis, 3rd edn. Naval Institute Press, Maryland
- Yakıcı E, Karasakal O (2013) A min-max vehicle routing problem with split delivery and heterogeneous demand. *Optim Lett* 7:1611–1625
- Zhang F, Wang G, Zeng Q, ye J (2013) An algorithm for minimal circumscribed rectangle of image object based on searching principle axis method. *Res J Appl Sci Eng Technol* 5(11):3083–3086