Cooperative Target Observation Moving Over a Planar Graph using a Modified Hill Climbing Search

Levi P. Figueiredo, Jose E. B. Maia
Universidade Estadual do Ceara UECE
60714-903 - Fortaleza - Ceara - Brasil

ABSTRACT

Consider an urban scenario in which N aerial UAVs, each with limited radius of observation R, must patrol $M > N$ targets moving on the surface. The movement of the UAVs is free while the movement of targets is restricted to certain paths, such as urban roads. Targets are friends who can, for example, be attacked by enemies. In this scenario, it can be assumed that the positions of the targets and the observers, obtained from GPS, are transmitted to a central command, and that the targets are collaborative, not avoiding the presence of the observers. This scenario is a new instance of the Cooperative Target Observation (CTO) problem. This work investigates a centralized algorithm, modified hill climbing, to command the UAVs with a view to maximize the average number of targets observed by at least one observer. The average performance of the proposed algorithm is superior to that of similar algorithms in this new problem setting.

Keywords

Cooperative target observation, Centralized command of UAVs, Modified hill climbing, Moving over a planar graph

1. INTRODUCTION

Unmanned vehicles (UAVs), whether terrestrial, aquatic or aerial, such as drones, already accumulate a variety in civil applications or military defense and attack. Civil applications include environmental monitoring [8], medical assistance [23], transport of goods [15], electronic surveillance [24], and aerial data surveys using photogrammetry techniques or LIDAR [20] sensors. Military applications of UAVs already reported include mission of attack [4], defense against attacks by other UAVs [3, 10, 16], reconnaissance [6] and border surveillance [22].

UAVs are a type of agents suitable for use as observers. The Cooperative Target Observation (CTO) problem domain is one in which a team of moving surveillance robots, for example, drones, must maintain the observation of another target robot team in motion, in order to maximize the Average Number of Observed Targets ($ANOT$) in the period.

Figure 1 illustrates the Cooperative Targets Observation (CTO) problem. In it, the observers are drones that fly over the environment and the targets, which are of two types, soldiers and vehicles, move on the ground. As illustrated, each observer has limited visibility. Drones are guided by a central command to maintain observation of targets on the ground. In this configuration of the problem, the positions of the targets are informed, for example, by GPS (Global Positioning System), and the targets do not escape from the observers. Targets and observers can work as a team. The CTO problem domain has a variety of instances depending on the type of movement of the targets, resource constraints, the interaction between targets and observers, and the stated specific objective. In [11] a general classification of this class of problems is described. The reader interested in a broader view should consult this work.

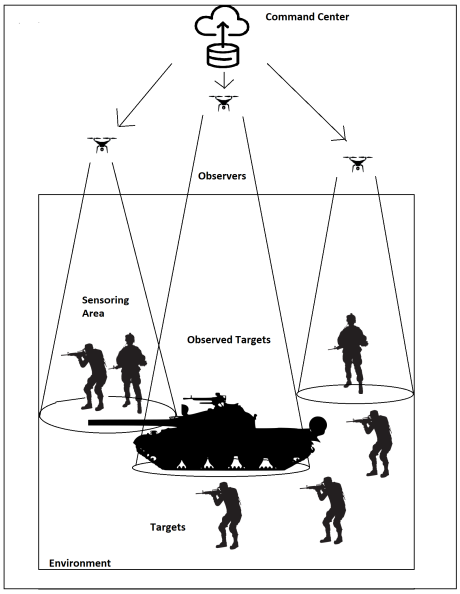

Fig. 1. CTO problem illustration. Observers and targets work as a team. Observers are drones guided by a central command in order to maintain the observation of targets which do not escape from observers.

In this paper, a new setting and algorithm for the cooperative targets observation problem is presented. In the configuration faced in this work the targets move on a planar graph. For a concrete example, consider an urban scenario in which N aerial UAVs, each with limited observation radius R, must patrol $M > N$ targets moving on land. The movement of the UAVs is free while the movement of targets is restricted to certain paths, such as urban roads. Targets are friends who can, for example, be attacked by enemies. In this scenario, it can be assumed that the positions of the targets and the observers, obtained from GPS, are transmitted to a central command, and that the targets are collaborative, not avoiding the presence of the observers.

In this work an algorithm for this problem is developed which adds knowledge of the domain through heuristics to improve the performance of the basic hill climbing algorithm and another already present in the literature.The work is organized as follows: Section 2 is a brief review of related work, Section 3 presents the methods used in order to solve the problem, Section 4 shows the results obtained and Section 5 concludes.

2. RELATED WORKS

To define the variation of the CTO problem studied in this work, be it the tuple (S, O, X) [21] where S is a limited, non-toroidal and closed two-dimensional spatial region, O is a set of N observing robots and X is a set of M targets. Observers and targets work together as a team. Denote by $\boldsymbol{\mu}_i = [x_i, y_i]$ an observer position in S and by $\mathbf{x}_i = [x_i, y_i]$ a target position in S. The constant speed of the observers is taken as a unitary reference to specify the relative speed of the targets. Each observer is equipped with a limited range sensor with radius R (denoted by *sensor_range*). In simulations, space and time are discretized to approximate continuous movements so that the observers walk 10 steps for each unit distance and the targets walk 10 steps for each step of an observer.

Although this work establishes a new setting for the CTO problem, this section briefly reviews some related works that put the proposal in perspective. The reader can also expand his view of this problem by reading the works in [18, 17, 5, 2, 19, 9].

In [21], Parker formally defines a more general version of this problem, called CMOMMT, for Cooperative Multi-Robot Observation of Multiple Moving Targets, in which the targets are not collaborative and the environment is partially observable. The objective function to be maximized for the observer team is the Average Number of Observed Targets ($ANOT$) in the simulation period, defined by the expression:

$$ANOT = \frac{1}{T}\sum_{t=0}^{T}\sum_{j=1}^{M}\bigvee_{k=1}^{N} a_{kj}, \qquad (1)$$

where $A = \{a_{ij}\}$ is an $N \times M$ matrix with the $a_{ij} = 1$ if target j is in the sensor range of the observer i and and 0 when not, and T is the number of time-steps of the simulation. The operator $\bigvee_{k=1}^{N}$ on a column of the A matrix causes each observed target to be counted only once. Since in this work the number of targets is constant in each simulation, $M = 24$, the percentage index, normalized by the number of targets in each execution, was used.

$$\rho = \frac{ANOT}{M}. \qquad (2)$$

Luke et al [14] defines the CTO problem as a simplified version of the CMOMMT problem in [21] in which the targets are collaborative and the environment is fully observable. For this problem he proposes centralized and decentralized algorithms, based on k-means and hill climbing, to calculate the trajectory of the observers.

The CTO problem has been studied in more recent works as in [2, 19, 1, 7]. However, these publications do not address the target mobility configuration introduced in the present work. The present work is based on the definition in Luke et al. and the centralized algorithms proposed there. The k-means and hill climbing versions in [14] are taken as baselines for performance comparison with the proposed variant front of the new environment configuration and target movement.

3. METHODS

This section describes the proposed algorithm to command observers and those used for comparison. The notation will follow that introduced in Section 2.

3.1 K-means based Command

An initial idea to command the observers' movement is to use a clustering algorithm with K-means being the widely used algorithm. When the number of K groups is predetermined, K-means seeks to minimize intra-group variance and maximize separation between groups. These properties favor the improvement of the ANOT index.

To map the CTO command problem to K-means the target positions are taken as points to group and the number of observers is made N = K. The basic K-means, however, has two problems: it can get stuck in a local minimum and it does not take into account the range of the observer's vision. In addition, being trapped in a local minimum can also result in the phenomenon of an empty cluster. Even so, this simple algorithm gives good results and is often used in performance comparisons. The procedure is described in Algorithm 1.

Algorithm 1: K-Means based command

```
Input: actual positions of targets and observers (μ_i, x).
Output: new positions of observers μ_{i+1}.
μ ← actual_positions_observers
x ← actual_positions_targets
t ← 1
while t ≤ max_it do
    for i in range(targets) do
        c_ik = argmin_k ||x_i − μ_k||^2
    end for
    for j in range(observers) do
        μ_j = (1/N) Σ_{i=1}^{N} c_ij
    end for
    t ← t + 1
end while
return μ_{i+1} = (1 − β)μ_i + βμ
```

In Algorithm 1, the beta parameter determines the observer's next destination to an intermediate point between the current position and that determined by k-means.

3.2 Hill-climbing based Command

An intuition about the behavior of solutions to the CTO problem, already present in works [21, 14], is that if two state-space configurations of observer and target positions result in the same value of the index ρ defined in the equation (2) then the one with larger the total area covered by the sensors, ie, the one in which the average distance between the observers is higher, is preferable. This notion is used in [21, 14] to construct heuristics to solve the problem. In [14], Luke et al. use this notion to create a variant of the hill climbing (HC) algorithm and compare its performance against that of k-means.

In the present work, a variant of the notion of that the higher average distance between the observers is preferable is used to add a heuristic step to a Hill climbing algorithm and its performance is compared to the k-means and Hill Climbing described in Luke et al. [14]. This algorithm will be denoted by HC+heuristic or HC+h and its pseudo-code is shown in Algorithm 2. The HC+h algorithm is described here. The reader is referred to article [14] for a description of the HC algorithm used in performance comparison.

Algorithm 2: Hill Climbing based command

Input: actual positions of observers and targets ($\boldsymbol{\mu}'$).
Output: new positions of observers ($\boldsymbol{\mu}$).
$\boldsymbol{\mu}' \leftarrow actual\_positions\_observers$
$eval(\boldsymbol{\mu}')$
$t \leftarrow 1$
while $t \leq max\_it$ **do**
  $\boldsymbol{\mu} \leftarrow perturb(\boldsymbol{\mu}')$
  $\theta = heuristic(\boldsymbol{\mu}, \boldsymbol{\mu}')$
  if $eval(\boldsymbol{\mu}, \theta) \geq eval(\boldsymbol{\mu}', \theta)$ **then**
    $\boldsymbol{\mu}' \leftarrow \boldsymbol{\mu}$
  end if
  $t \leftarrow t + 1$
end while
return $\boldsymbol{\mu}$

The structure and the $WHILE$ loop of Algorithm 2 is that of a classic hill climbing algorithm. The command line $\theta = heuristic(\boldsymbol{\mu}, \boldsymbol{\mu}')$ is what distinguishes the algorithms. Call the $\boldsymbol{\mu}$ solution parent and child the $\boldsymbol{\mu}'$ perturbed solution obtained from $\boldsymbol{\mu}$. The proposed heuristic for this new setting of the CTO problem consists of the following steps:

(1) Accept the child if he observes more targets;
(2) If you observe the same number of targets, compute

$$H = \sum_{o \epsilon O} \sum_{t \epsilon T} \begin{cases} dist(o,t), & \text{if R/2 ¡ dist(o,t) ¡ R;} \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where dist (o, t) is the Euclidean distance between the observer $o \epsilon O$ and the target $t \epsilon T$. Accepts the child solution if $H_{child} < H_{parent}$. This motivates observers to meet targets that are closer;
(3) If $H_{child} = H_{parent}$ and the child and parent solution observe the same number of targets, then compute:

$G = \sum_{o \epsilon O'}^{t \epsilon T'} min(dist(o,t))$

where O' is the set of observers that observe no target and T' is the set of targets that are not observed by any observer. Accepts the child solution if $G_{child} < G_{parent}$. This motivates observers who do not observe any target to be approaching targets that are not observed by any observer;
(4) Otherwise, reject the child position and move on to the next observer.

With that the description of the proposed algorithm is complete.

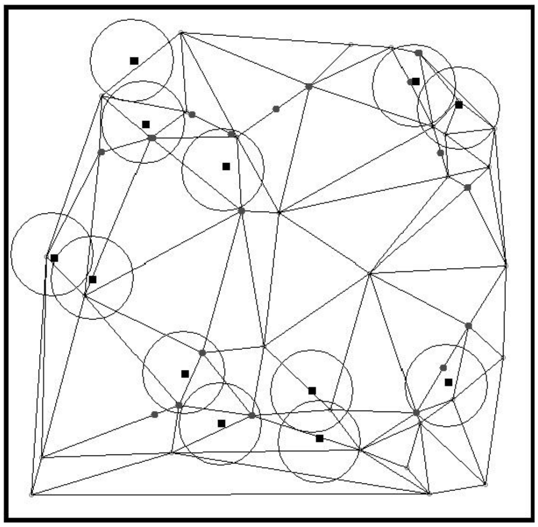

Fig. 2. Simulation snapshot in the MASON environment for the hill climbing algorithm. In this execution, max it iterations was achieved with the algorithm remaining in a local minimum where each observer is observing some target, but there is targets not observed by any of the observers.

4. EXPERIMENTS AND RESULTS

The performance was analyzed by simulation in the MASON [13] environment. Observers walk in the direction indicated by the last received command. The central command does not need to update the trajectory of observers at each step of time. The observer trajectory command (UR) update rate is a relevant parameter because each update places a communication load on the system. The sensitivity to these parameters also measures the robustness of the system, in latu sensu, to losses of messages and other updating faults. In algorithm 3, this command calculation frequency is given by α: an update of the observer's destination at each α step.

To generate planar graphs, a two-step procedure was used: first, the vertices were generated in random positions and then a Delaunay triangulation algorithm [12] was applied to construct the edges in order to result in a planar graph. All experiments were performed with graphs of 40 vertices.

Figure 2 shows a simulation snapshot in the MASON environment. It shows a generated random graph, circles representing the range of the observers' sensors, and the points on the edges of the graph are moving targets. Note that in this figure the algorithm is clearly at a local minimum where there are observers not observing any target and targets not observed by any of the observers.

In the tests, the simulation used the same parameters adopted in [14]: a rectangular 2D space, with 150x150 units dimension, where targets and observers are inserted; each experiment has a limited time of 1500 time-steps; observers move at 1 unit per time-step, while the targets can move at various speeds RV = $\{0.1, 0.25, 0.5, 0.75, 0.9\}$ unit per time-step; the sensor range in each observer can be SR = $\{15, 18, 20, 22, 25\}$ units; and the rate of updating of the trajectory of the observers was set in $\alpha = 10$; $M = 24$ and $N = 12$. To collect results, each configuration was simulated 20 runs, each with initial random configuration and independent random number generator seeds. For abbreviation, the algorithms will

Algorithm 3: General simulation pseudo-code

Input: Vector of the initial positions of targets and observers and simulation parameters.
Output: Visualization of the simulation and calculation of index $ANOT$.
for t **in range**(T) **do**
  Move targets and observers towards current destinations
  Calculate and store $g(B(t)_{.j})$ for $j = 1...n$
  Update target destinations
  if $t \ mod \ \alpha = 0$ **then**
    Run the observers command algorithm (KM, HC or HC+h)
    Update observer destinations
  end if
end for
Calculate index $ANOT$

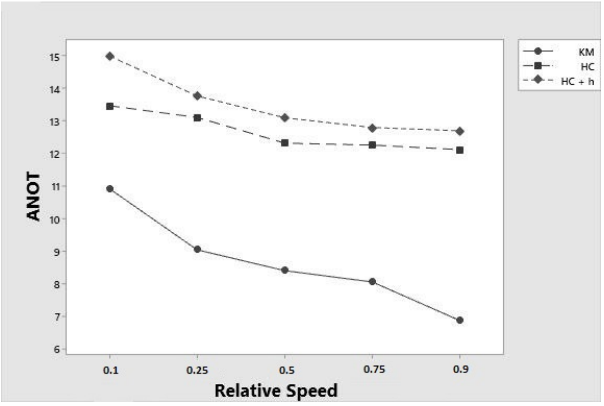

Fig. 3. Comparison of ANOT performance when varying the relative speed (RS) for the sensor range set at 20.

be referred to as k-means[14], HC[14] and HC+heuristic or HC+h - for this work. The results are shown in Table 1.

Table 1 is divided into two parts, where, in each part is varying one of the parameters RV or SR, keeping the others fixed at the median of their set of values. For a better view of these results, the data in Table 1 were plotted in Figures 3 and 4. An analysis of this table and figures is as follows:

—The HC+h algorithm presents superior performance for all scenarios.

—When the relative speed (RS) is low or the sensor range (SR) is high or both simultaneously, the difference in algorithm performance is lower. These are less challenging scenarios.

—In the high relative speed (RS) or low range sensor (SR) challenging situations or both simultaneously, it is when the performance of HC+h presents greater resilience while those of the others degrade.

—The standard deviations between batches of 20 runs did not differ significantly for the algorithms.

The total processing time for Table 1 was 93 hours. For real-time execution the maximum duration of one cycle of the HC + h algorithm was 2 s. These numbers are referred to a CORE I7 processor.

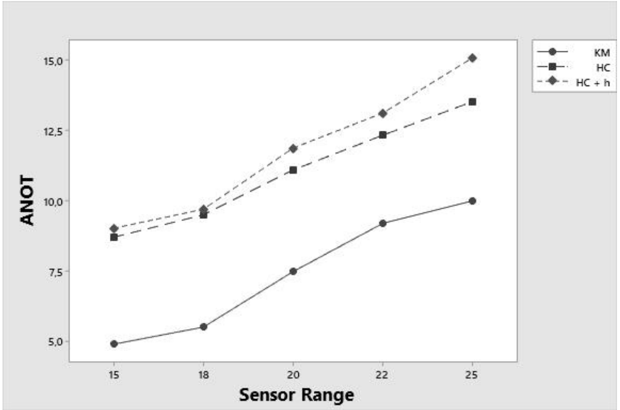

Fig. 4. Comparison of ANOT performance when varying the sensor range (SR) for the relative speed set at 0.5.

5. CONCLUSION

A new configuration of the CTO problem, with the movement of the targets on a planar graph, and a new variant of a centralized algorithm to control the trajectory of the observers are presented and evaluated in this work. A hypothetical motivational example of application was presented in which the edges of the graph are the traffic lanes of an urban region. Performance was analyzed by simulation in MASON [13].

Comparative performance table by varying the critical parameters of the problem, ie the range of the sensors (SR) and the relative speed between observers and targets (RV) show that the proposed algorithm HC+h improved the average performance and slightly reduced the variance against the baseline versions published in [14].

Some work limitations are immediately identified and point to future work. The first is that the proposed algorithm, just like the original, does not construct target motion model. It is believed that incorporating this feature and then applying reinforcement learning to the trajectory of observers will improve the effectiveness of the algorithm. The second is to obtain and execute with real scenario data to counter performance with that obtained in simulation. [14]

6. REFERENCES

[1] Joao P B Andrade, Robson Oliveira, Thayanne F da Silva, José E Bessa Maia, and Gustavo AL de Campos. Organization/fuzzy approach to the cto problem. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 444–449. IEEE, 2018.

[2] Rashi Aswani, Sai Krishna Munnangi, Praveen Paruchuri, and Hilton San Francisco. Improving surveillance using cooperative target observation. In *AAAI*, pages 2985–2991, 2017.

[3] Christoph Briese, Andreas Seel, and Franz Andert. Vision-based detection of non-cooperative uavs using frame differencing and temporal filter. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 606–613. IEEE, 2018.

[4] Andrew Callam. Drone wars: Armed unmanned aerial vehicles. *International Affairs Review*, 18, 2015.

[5] Jesús Capitan, Luis Merino, and Aníbal Ollero. Cooperative decision-making under uncertainties for multi-target surveil-

Table 1. The average performance ρ of the HC + heuristic (HC+h) algorithm proposed against K-means + heuristic and Hill Climbing evaluated in [14], when it varies: sensor range (SR), relative speed between observers and targets (RV). As the case, RV=0.50 or SR=15 when the other parameter is varying, N=12 and M=24.

Sensor range	-	15	18	20	22	25
K-means[14]	m	4.89	5.50	7.48	9.20	10.00
	sd	1.11	0.97	1.97	1.5	2.10
HC [14]	m	8.70	9.50	11.10	12.34	13.54
	sd	1.18	0.91	0.84	0.88	1.11
HC+h	m	9.02	9.51	11.88	13.13	15.10
	sd	1.47	0.57	1.16	0.54	1.19
Rel. speed	-	0.90	0.75	0.50	0.25	0.10
K-means[14]	m	10.92	9.04	8.40	8.05	6.86
	sd	1.40	2.02	1.49	1.70	0.89
HC [14]	m	13.47	13.11	12.32	12.26	12.12
	sd	0.83	0.96	0.82	0.76	0.94
HC+h	m	15.00	13.77	13.10	12.80	12.70
	sd	1.04	0.85	1.08	1.18	1.17

lance with multiples uavs. *Journal of Intelligent & Robotic Systems*, 84(1-4):371–386, 2016.

[6] Victor Catano and Jeffery Gauger. Information fusion: Intelligence centers and intelligence analysis. In *Information Sharing in Military Operations*, pages 17–34. Springer, 2017.

[7] Levi P Figueiredo and José E B Maia. Busca hill-climbing modificada aplicada á observao de alvos cooperativos (cto) movendo-se sobre um grafo planar [in portuguese]. In *2020 11o COMPUTER ON THE BEACH (COTB2́0)*, pages –. Universidade do Vale do Itajaí, 2020.

[8] Patricia K Freeman and Robert S Freeland. Agricultural uavs in the us: potential, policy, and hype. *Remote Sensing Applications: Society and Environment*, 2:35–43, 2015.

[9] Jinwen Hu, Lihua Xie, Kai-Yew Lum, and Jun Xu. Multiagent information fusion and cooperative control in target search. *IEEE Transactions on Control Systems Technology*, 4(21):1223–1235, 2013.

[10] Sungsik Huh, Sungwook Cho, Yeondeuk Jung, and David Hyunchul Shim. Vision-based sense-and-avoid framework for unmanned aerial vehicles. *IEEE Transactions on Aerospace and Electronic Systems*, 51(4):3427–3439, 2015.

[11] Asif Khan, Bernhard Rinner, and Andrea Cavallaro. Cooperative robots to observe moving targets. *IEEE Transactions on Cybernetics*, 48(1):187–198, 2018.

[12] Der-Tsai Lee and Arthur K Lin. Generalized delaunay triangulation for planar graphs. *Discrete & Computational Geometry*, 1(3):201–217, 1986.

[13] Sean Luke, Claudio Cioffi-Revilla, Keith Sullivan Liviu Panait, and Gabriel Balan. Mason: A multi-agent simulation environment. *Simulation: Transactions of the society for Modeling and Simulation International*, 82(7):517–527, 2005.

[14] Sean Luke, Keith Sullivan, Liviu Panait, and Gabriel Balan. Tunably decentralized algorithms for cooperative target observation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 911–917. ACM, 2005.

[15] Hamid Menouar, Ismail Guvenc, Kemal Akkaya, A Selcuk Uluagac, Abdullah Kadri, and Adem Tuncer. Uav-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 55(3):22–28, 2017.

[16] Timothy L Molloy, Jason J Ford, and Luis Mejias. Detection of aircraft below the horizon for vision-based detect and avoid in unmanned aircraft systems. *Journal of Field Robotics*, 34(7):1378–1391, 2017.

[17] Beulah Moses. Convergence of annealing algorithms for cooperative uavs. *Recent Adv. Appl. Theor. Math.*, pages 109–116, 2013.

[18] Beulah Moses and Lakhmi C Jain. Cooperative target observation of uavs using simulated annealing. *International Journal of Intelligent Defence Support Systems*, 1(2):116–129, 2008.

[19] Sai Krishna Munnangi and Praveen Paruchuri. Improving wildlife monitoring using a multi-criteria cooperative target observation approach. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

[20] Francesco Nex and Fabio Remondino. Uav for 3d mapping applications: a review. *Applied geomatics*, 6(1):1–15, 2014.

[21] Lynne E Parker. Cooperative robotics for multi-target observation. *Intelligent Automation & Soft Computing*, 5(1):5–19, 1999.

[22] Carlos Paucar, Lilia Morales, Katherine Pinto, Marcos Sánchez, Rosalba Rodríguez, Marisol Gutierrez, and Luis Palacios. Use of drones for surveillance and reconnaissance of military areas. In *International Conference of Research Applied to Defense and Security*, pages 119–132. Springer, 2018.

[23] James C Rosser Jr, Vudatha Vignesh, Brent A Terwilliger, and Brett C Parker. Surgical and medical applications of drones: A comprehensive review. *JSLS: Journal of the Society of Laparoendoscopic Surgeons*, 22(3), 2018.

[24] Yunhui Wang, Zhijuan Zhan, and Bing Xue. Operation method of electronic warfare uav. In *Global Intelligence Industry Conference (GIIC 2018)*, volume 10835, page 108351L. International Society for Optics and Photonics, 2018.