The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 6 (2020) 76–82

www.elsevier.com/locate/icte

Switching target communication strategy for optimizing multiple pursuer drones performance in immobilizing Kamikaze multiple evader drones

Ario Yudo Husodo[a,b,*], Grafika Jati[a], Amarulla Octavian[c], Wisnu Jatmiko[a]

[a] *Faculty of Computer Science, Universitas Indonesia, Indonesia*

[b] *Department of Informatics Engineering, Faculty of Engineering, University of Mataram, Indonesia*

[c] *Indonesian Naval Command and Staff College, Indonesia*

Received 30 November 2019; received in revised form 9 March 2020; accepted 31 March 2020

Available online 9 April 2020

Abstract

We propose a method for optimizing multiple-drone pursuers' performance in handling attacks from *Kamikaze* multiple-drone evaders on a battlefield. The central aspect of this problem is to minimize damage produced by evaders towards a defended area guarded by multiple-pursuers. We propose a communication strategy among pursuers where each pursuer can communicate with each other to decide which evaders should be chased and immobilized by each pursuer. We simulate the proposed method in a dynamic 3D environment. The simulation results conclude that our proposed method performs better than the commonly used algorithm for solving this kind of problem.

Keywords: Battlefield; Kamikaze; Multiple-drones; Performance-optimization; Switching-target-communication

1. Introduction

The rapid development of drone technology triggers the increase of possible destructive attacks performed by drones in a battlefield. In nowadays environment, it is possible for multiple drones working together as a swarm (group) to attack an area on a battlefield. Overcoming this thread using conventional technology is seemly useless. A better strategy is needed to tackle this problem efficiently. One of the most promising approaches available for this problem is the usage of multiple drone's technologies that work as a group of pursuers that can automatically catch, destroy, or immobilize any existing evaders, as shown in Fig. 1.

From the computer science field, the problem mentioned earlier is a part of the Multiple-Pursuers Multiple-Evaders (MPME) problem. MPME is a computer science problem in swarm optimization where there are many pursuers trying to capture some evaders trying to infiltrate and attack an area. In this kind of problem, there is a group of evaders attempting to strike a secured area by simultaneously moving towards it as their target. When evaders reach a specific range from the target, evaders can attack the target by any available weapon they own. The main task in this problem is how to coordinate a group of defenders (pursuers) to pursue all evaders then immobilizing them by shooting a weapon upon them. Each pursuer can detect, in a limited range, the existence and location of every evader around the pursuer. However, any pursuer cannot predict the movement of each evader. The goal in MPME problem is to minimize the secured area damage caused by evaders.

There have been many approaches conducted to solve MPME problem variant, for example [1–3]. In MPME research topic, the technical issue in how a pursuer catches, destroys, or immobilizes a target is usually not discussed in detail. This technical issue is strictly related to the implementation environment of the algorithm; for example, a pursuer can shoot an evader using a net weapon. MPME research problem usually simplifies the evaders catching process explanation by automatically destroy an evader when a pursuer is in a specific range of the evader. The common assumption used in MPME research is also that although evader can cause damage to its target area, each evader does not attack any pursuer. This paper will also follow the approaches.

In the conventional approach, MPME problem is solved by locking an evader as a pursuer target. The pursuer will perform

* Corresponding author at: Faculty of Computer Science, Universitas Indonesia, Indonesia.

E-mail address: ario@unram.ac.id (A.Y. Husodo).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2020.03.007

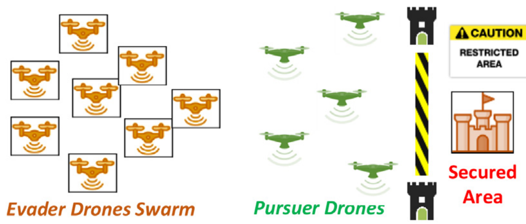

Fig. 1. MPME drones illustration.

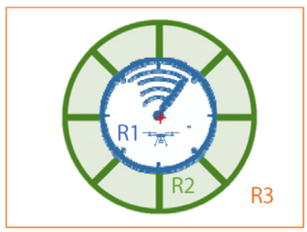

Fig. 2. Regions variety of a pursuer.

some pathfinding and obstacle avoidance algorithms to find the best path to reach the locked target (an evader). In this kind of approach, some siege maneuver is also possible to be conducted. According to the state-of-the-art performance in MPME available solutions, the conventional approaches can practically solve the MPME problem. The approach can be proven mathematically to catch all existing evaders. A research example of this conventional approach can be seen in [1].

The main task in MPME problem is to catch all existing evaders as soon as possible for the sake of damage minimization. Although there have been some solutions available to capture all existing evaders, some improvement is still needed to decrease the damage caused by evaders' attack. This improvement is what this paper offers. We develop an approach based on the development of the Internet of Battlefield Things (IoBT). Here, every pursuer is possible to share data and communicate with each other. As mentioned in [4], in the practical world nowadays problem, one of the open research issues in the anti-drone research field is about data association for tackling multiple drones. This paper tries to propose an approach for solving the open issue.

2. Research scope

This paper provides more detail explanation for multiple drones' communication strategy for the topic described in [5]. As the issue arose on [5], we try to explain how our proposed algorithm can conduct an effective communication strategy among evader drones. Here, we focus our problem on MPME problem variant where evaders act as *kamikaze* troops, where they do not consider the way back home maneuver or even dodging maneuver. Evaders only care about how much damage they can create towards a specific target. The equation about this damage can be seen in [5]. This paper uses the same assumptions and scopes provided by [5].

This research does not explain what kind of path planning algorithm or obstacle avoidance algorithms is conducted by each pursuer to catch an evader. There are many algorithms developed about path planning and obstacle avoidance, for example, [6–12]. This paper will only focus on communication strategy to assign which evader should be chased by each pursuer efficiently. This paper will not either describe what kind of sensor does a pursuer need to detect the location of evaders around the pursuer. Many articles have discussed this, too, for example [7,9,13]. In this paper, each pursuer has a limited detection-range to detect the location of any available surrounding evaders. Each pursuer can obtain the precise location of every evader inside the purser detection-range, but it cannot predict any evader's movement.

This paper does not either describe what technology is used by each pursuer to communicate and share data because it can be varied depending on the technical issue used by the technology of drones. This paper only focuses on what kind of data is shared among the pursuers and how each pursuer should respond to the data received. The focus of this paper is on the pursuer drones' communication algorithm.

3. Proposed method

In our problem scope, each pursuer has 3 different regions. Fig. 2 shows the region variety owned by each pursuer. Please note that regions in Fig. 2 are 2D simplification region of a 3D environment region around a pursuer. In Fig. 2, the center of the circle implies the position of a pursuer. Every pursuer can only detect the location of evaders if they are inside R2 region. Thus, any pursuer cannot detect any evader located in R3 region. When a pursuer chases an evader, there might be a time where the chased evader is so close enough to the pursuer that the evader is inside R1 region. R1 region symbolizes an area where a pursuer can accurately attack evaders by any available weapon. Thus, when an evader is in R1 region, in this paper, the evader will be automatically destroyed for simplicity. This approach is also used in [5]. Many MPME types of research assume that every pursuer has an unlimited amount of weapon/ammunition. Thus, in this research, there is no scenario where a pursuer runs out of ammo. No matter how many evaders enter R1 region of a pursuer, the pursuer can automatically destroy the evaders in R1 region.

Algorithm 1 describes the main algorithm of the proposed method. Each pursuer performs this algorithm. In this algorithm, the variable *Pme* denotes the pursuer running this algorithm. In our proposed method, every pursuer can communicate to all existing pursuers, and there is no master in the pursuers' group. When a pursuer *Px* has locked an evader *lockx* and trying to chase it, other pursuers can know automatically that a pursuer is targeting evader *lockx*. However, the information about which pursuer is chasing the *lockx* can only be known after a pursuer *Pme* call the ***FindPursuerWhoIsChasing****(lockx)* procedure, as shown on Algorithm 1 Line 09. In brief, the procedure of ***FindPursuerWhoIsChasing****(lockx)* is a mechanism where *Pme* asks every pursuer whether they are chasing *lockx* or not. When a pursuer *Px* is chasing *lockx*, it will give its information to *Pme*.

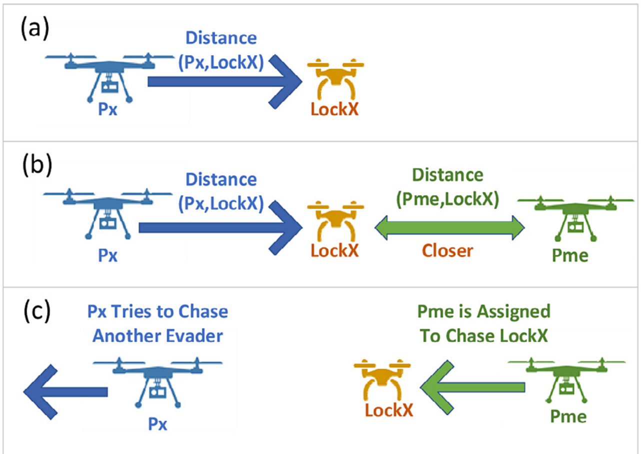

Fig. 3. Simple case of switching target illustration.

Algorithm 1 Proposed Switching Target Strategy

```
01: procedure Chase Evader Performed by a Pursuer Pme
02:   if Pme does not have locked evader to be chased then
03:     nev ← FindNearestEvaderIn_R2
04:     lockx ← FindNearestLockedEvaderIn_R2
05:     if nev = empty then
06:       if lockx = empty then
07:         perform_standBy_maneuver
08:       else
09:         Px ← FindPursuerWhoIsChasing(lockx)
10:         if distance(Px, lockx) > myDistanceTo (lockx) then
11:           Pme ← assignedAsChaserOf(lockx)
12:         else
13:           perform_standBy_maneuver
14:         end if
15:       end if
16:     else
17:       if lockx = empty then
18:         lockEvader(nev)
19:         Pme ← assignedAsChaserOf(nev)
20:       else
21:         if myDistanceTo(lockx) < myDistanceTo(nev) then
22:           Px ← FindPursuerWhoIsChasing(lockx)
23:           if distance(Px, lockx) > myDistanceTo (lockx) then
24:             Pme ← assignedAsChaserOf(lockx)
25:           else
26:             lockEvader(nev)
27:             Pme ← assignedAsChaserOf(nev)
28:           end if
29:         else
30:           lockEvader (nev)
31:           Pme ← assignedAsChaserOf(nev)
32:         end if
33:       end if
34:     end if
35:   else
36:     myLock ← LockedEvaderBeingChasedBy_Pme
37:     Chase(myLock)
38:     if (myLock in R1 Range) then
39:       destroy(myLock)
40:     endif
41:   endif
42: end procedure
```

The primary proposed method is on Algorithm 1 Lines (09–11 and 22–24). The line of codes describes that when a pursuer *Pme* tries to chase a locked evader *lockx*, *Pme* communicates to pursuer *Px,* who is pursuing *lockx*. If the distance of *Pme* is closer to *lockx* compared to the distance of *Px* towards *lockx*, then *Pme* will overtake the chasing of *lockx*. Meanwhile, the *Px* should stop chasing the *lockx* and try to chase another evader.

$$ds = \sqrt{(dx)^2 + (dy)^2 + (dz)^2} \quad (1)$$

Because each drone can move freely in 3D space, (1) calculates the distance between two drones (*ds*), where *dx*, *dy*, and *dz* consecutively refer to the distance unit of two drones on x-axis, y-axis, and z-axis in 3D space. Please note that the term of ***locked evader*** used in Algorithm 1 refers to an evader that has been assigned to be chased by a pursuer. Fig. 3 illustrates the simplest case of switching target strategy proposed by this paper. Fig. 3(a) illustrates the initial position where a pursuer *Px* tries to chase a locked evader *lockx*. When a pursuer *Pme* detects the existence of *lockx,* as shown in Fig. 3(b), *Pme* communicates to *Px* to compare their distance towards *lockx*. If *Pme* is closer to *lockx*, then *lockx* should be chased by *Pme* instead of *Px,* as shown in Fig. 3(c).

By default, when a pursuer does not detect the existence of any evader, it tends to stand by around a secured (protected) area by calling ***perform_standBy_maneuver*** procedure. You may notice that in the proposed method, there is no explicit siege maneuver to chase an evader. We intentionally avoid the siege maneuver to optimize the resource utilization of each pursuer. Instead of performing siege maneuver, we tend to synthesize that damage result of MPME problem can be reduced when each pursuer only chases the nearest available evader that is not being chased by another pursuer.

4. Experiment result

There are 2 main parameters measured as indicators of how good the proposed method performance is; first is the number of iteration (*num_iter*), and second is the number of damage (*num_dmg*). *Num_iter* indicates how many iterations needed by pursuers to catch all evaders trying to attack a secured area. According to [5], *num_iter* is not varied much for any different algorithm; thus, we can ignore this parameter. Meanwhile, *num_dmg* indicates the number of damage received by a secured area caused by evaders' attack. It is calculated as damage accumulation from evaders surrounding the area in a specific range per iteration. In a real-life scenario, evaders can only attack an area when they are in a close distance from the area. Thus, in this experiment, *num_dmg* is increased only when there are evaders in a specific range from a secured area. Please see [5] to know the more detail formula used to measure *num_dmg*. Here, for the performance evaluation parameter, the less *num_dmg* received by an area, the better the performance of the algorithm.

In this research, we test the proposed algorithm in a 3D Unity environment. Here, we use a distance measurement unit as provided by the simulation environment. There is no exact calibration for one distance unit in our simulator into real physical distance units such as a meter. However, we try to

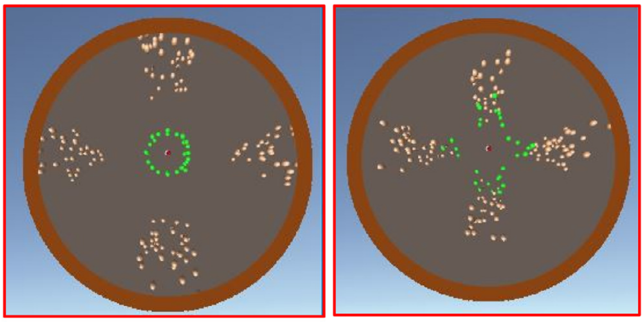

Fig. 4. Mini maps showing position of drones in experiment.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

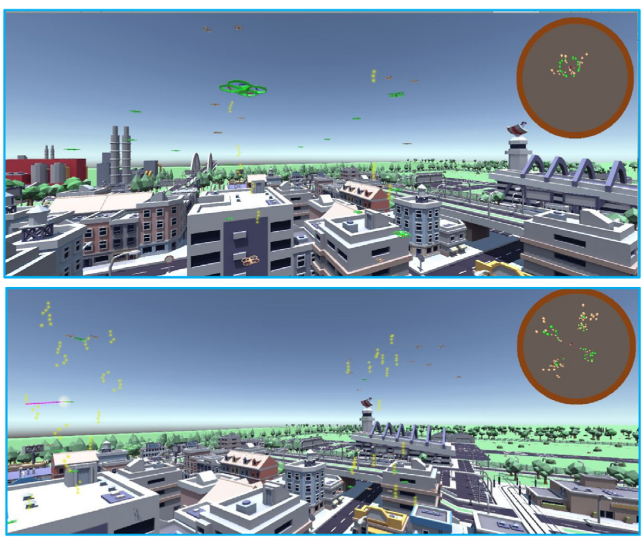

Fig. 5. Simulation environment for performance evaluation.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

manage that one distance unit in our simulator is as far as the maximum distance a pursuer drone can reach in every second. Thus, in our simulator, a pursuer drone can move up to 1 distance unit per iteration. For an analogy, if in a real physical world, a drone has maximum flying speed 8 meters/second, then a distance unit in our simulator is equal to 8 meters in the real world. In our simulator, we use a 3D coordinate system, where the drones are not moving in a strict discrete grid. The coordinate system can be a decimal number, such as 0.123. Each pursuer drone can move to any arbitrary direction.

In our experiment, we do not assume all evaders and pursuers have the same amount of speed. We try to explore the effects of speed variety among drones towards *num_dmg*. We provide Table 1 to explain our parameters in this experiment. Please note that in Table 1, we also provide some different values of pursuer's region R1 and R2. As a gentle reminder, we want to highlight that our proposed algorithm is an optimization algorithm. We want to increase the performance of any available method to handle our discussed problem.

As a consequence, we do not provide random movement of the evaders because this kind of movement can produce some biased results. When an evader drone performs a non-attacking movement such as hiding, this movement will not create any damage to the secured area (*num_dmg*). We can neglect this movement according to our objective. Because our main objective is on decreasing *num_dmg*, we focus our experiment on showing that our proposed algorithm can perform a good result in handling a *kamikaze* drone direct attack. This kind of approach is computationally formal in promoting an optimization algorithm for MPME problem, as used on [5]. In a *kamikaze* drone maneuver, each evader drone movement is intended to make a direct attack towards a target.

Fig. 4(a) shows the initial position of evaders and pursuers in each testing. We set the nearest evader distance from the secured area to 40 units. Please note that Fig. 4 is just a mini-map from the sky-view, which symbolizes every drone with a circle dot. If we take a closer look at the simulation environment, then Fig. 5 shows how we manifest the drones. In Fig. 4, evaders are symbolized with the brown dots, while pursuers are symbolized with the green dots. The red circle in the center of Fig. 4 indicates the central point of the area needed to be secured. Evaders have two main formations from four directions, as shown in Fig. 4(a), which are triangle formation as shown on the left and right side of Fig. 4(a), and rectangle formation as shown on the upside and downside of Fig. 4(a).

Each evader starts from different (random) height, then moving directly but in a constant height to a secured area. Every evader does not perform any hiding or dodging maneuver because it will create a bias on performance evaluation. Here, we follow the evaders' movement equation as used on [5]. When an evader is near enough to its target, it changes its latitude to the height of the secured area. Meanwhile, the pursuers start from the random position from surrounding of a secured area. After some period of iteration, the coordinate of each drone may change, as shown in Fig. 4(b). Please notice that the bigger the size a dot has in Fig. 4, the higher its position from the ground.

Fig. 5 shows that there are two kinds of drones in the simulation environment. In this experiment, the green drones represent pursuer drones; meanwhile, the brown drones represent evader drones. The purple line animates the process of a pursuer capturing an evader. If we take a closer look at Fig. 5, it tends to be complicated for pursuer drones to handle all of the evaders if the pursuers do not have a good strategy to solve the problem. Thus, to efficiently manage the MPME issue discussed here, a switching target communication strategy is proposed.

To gain the objective result performance evaluation of the proposed method, we compare the proposed method with the conventional method usually used for solving MPME problem. There are many variants of conventional methods for solving MPME problem, for example, by using area minimization, as shown on [1]. However, in the general use of the conventional method, each pursuer only chases the nearest evaders detected around pursuer. There is no switching target communication

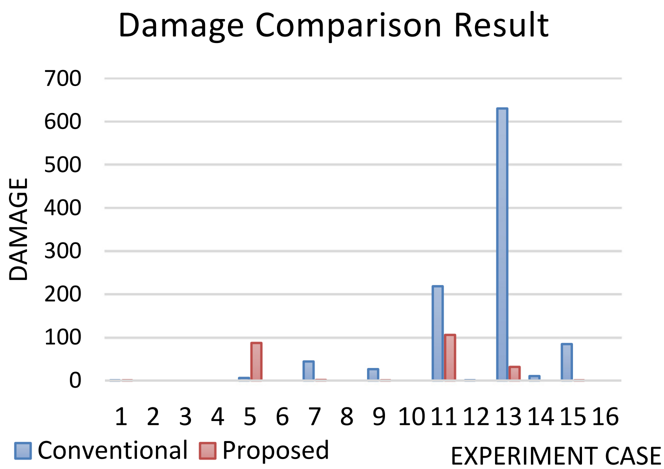

Fig. 6. Number of damage comparison result (Case 1–16).

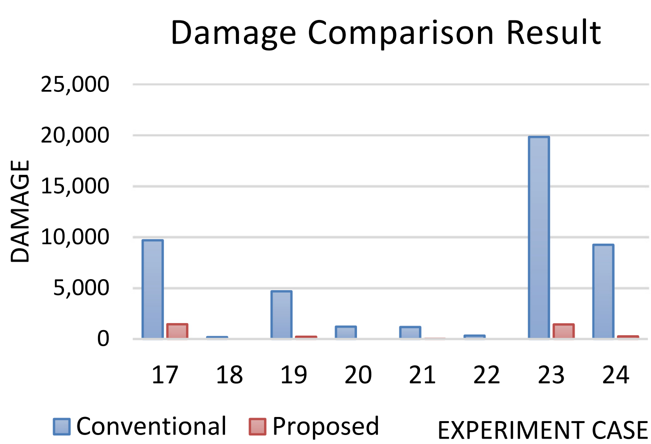

Fig. 7. Number of damage comparison result (Case 17–24).

Table 1
Experiment results for performance evaluation.

Case	Amount		Max Speed		Range		Average Damage (num_dmg)	
	P	E	P	E	R1	R2	Conv.	Prop.
1	40	40	1	1	2	10	0.2	0.2
2	40	40	1	1	4	20	0.0	0.0
3	40	40	1	0.8	2	10	0.0	0.0
4	40	40	1	0.8	4	20	0.0	0.0
5	40	40	1	1.25	2	10	5.9	87.0
6	40	40	1	1.25	4	20	0.0	0.0
7	40	80	1	1	2	10	44.4	1.2
8	40	80	1	1	4	20	0.0	0.0
9	40	80	1	0.8	2	10	26.4	0.2
10	40	80	1	0.8	4	20	0.0	0.0
11	40	80	1	1.25	2	10	218.8	105.8
12	40	80	1	1.25	4	20	0.8	0.0
13	40	160	1	1	2	10	631.0	31.5
14	40	160	1	1	4	20	10.4	0.0
15	40	160	1	0.8	2	10	84.6	0.2
16	40	160	1	0.8	4	20	0.0	0.0
17	40	160	1	1.25	2	10	9,701.4	1,441.2
18	40	160	1	1.25	4	20	167.0	0.0
19	40	320	1	1	2	10	4,671.0	206.2
20	40	320	1	1	4	20	1,210.6	0.0
21	40	320	1	0.8	2	10	1,167.0	6.8
22	40	320	1	0.8	4	20	312.8	0.0
23	40	320	1	1.25	2	10	19,853.6	1,423.2
24	40	320	1	1.25	4	20	9,244.0	243.0

**P = Pursuer, E = Evader, Max Speed is measured in distance unit per iteration, Range is measured as a distance unit, Conv = conventional method, Prop = proposed method*

strategy among them. In our simulation, we implement a conventional method by using this approach: each pursuer only chases the nearest detected evader a pursuer, pursues it until the pursuer can destroy it, then conduct a new search for other evaders.

We have conducted a varied amount of pursuers vs. evaders to evaluate the comprehensive performance evaluation between our proposed method and the conventional method. Because this experiment is related to a dynamic environment, we test each scenario 5 times then write their average num_dmg, as shown in Table 1. Figs. 6 and 7 show the data trend of Table 1. For the sake of readability, we split the chart of Table 1 results in Figs. 6 and 7. Please note that both figures use a different scale on the y-axis. According to those figures, the proposed method performs better than the conventional method commonly used in solving MPME.

5. Discussion and conclusion

According to our experiment results, the proposed method in this paper performs better than the conventional method usually used for solving MPME problem. The proposed method produces less damage compared to the conventional method. In the conventional method approach, each pursuer tends to chase the closest evader detected around the pursuer. Thus, when more than one pursuer surround an evader, all of those pursuers tend to chase the evader then perform siege maneuver. Theoretically speaking, this approach indeed causes the time to catch an evader become more efficient. However, as a drawback, this approach tends to ignore any other evaders that could harm the secured area. Fig. 8 illustrates the situation. In Fig. 8, there are several green pursuers trying to chase red evader ev in the center. Meanwhile, the other brown evaders are ignored.

Although our proposed method mostly outperforms the result of the conventional method, our proposed method produces a worse result compared to the conventional method when we run the experiment case 5 (Table 1). In this case, the conventional method performs better because the number of evaders is equal to the number of pursuers. Comparing this

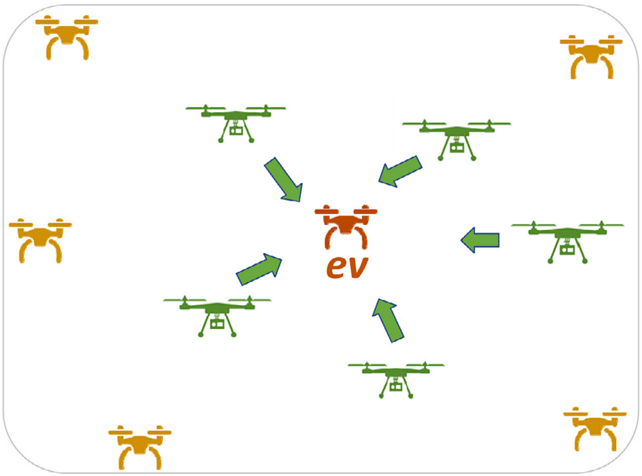

Fig. 8. Drawback example of conventional MPME method.

result with cases 1–4, we analyze that our proposed method is useful to increase the conventional method when the number of evaders is more than the number of pursuers.

According to our experiment, the R1 and R2 areas affect the *num_dmg* significantly. When the coverage of R1 and R2 increase, the *num_dmg* for both conventional and proposed methods decrease. This result indicates that when we try to develop an area defense system using drones, the wider area can be detected by the drones' sensor, the better drones' performance will be. Table 1 results also show that the speed of pursuers' drones is very crucial in defending an area. When the pursuers' speed is less than the evaders' speed, then the damage of the secured area will be more serious. Thus, when we consider developing such a multiple-drone based defense system, we should use the most advanced drone technology that can move at high speed.

In our experiment, we do not analyze the drone's power consumption needed for pursuer drone's communication. This parameter is indeed essential for practical implementation. However, because of the rapid development of battery technology, we analyze that the communication energy usage among drones will not affect much the pursuer drones operation. Besides, according to [14], the communication-related energy used by a UAV in practice is much smaller than the UAV's propulsion energy. The comparison is about a few watts [15] vs. hundreds of watts [16]. Thus, we consider ignoring this parameter.

Different from the conventional method, in the proposed method, each pursuer tries to avoid siege maneuver. Every pursuer communicates to each other to check which pursuer should take care of an evader. By doing this, although the speed for catching a single evader might not be as fast as the performance of conventional method speed, in general, the time required to catch all evaders is faster. Thus, the damage taken from the evaders' attack could be reduced. Because the experiment is conducted in a dynamic environment where the movement of each drone cannot be repeated precisely 100%, it is difficult to state the improvement percentage of the proposed method. It depends on how many pursuers and evaders are involved in the experiment. However, according to our experiment, it is shown that the proposed method could be used to optimize the performance of the conventional MPME method in handling attacks from multiple evaders drone on a battlefield. In this experiment, one of the best improvements of the proposed method is shown in Table 1 Case 15, where damage reduction could perform more than 99%. In conclusion, the usage of switching target communication strategy is experimentally proven that can improve the performance of multiple pursuer drones for catching multiple evaders drones on a battlefield.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ario Yudo Husodo: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing - original draft, Visualization. **Grafika Jati:** Validation, Formal analysis, Writing - review & editing. **Amarulla Octavian:** Conceptualization, Resources. **Wisnu Jatmiko:** Conceptualization, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgment

This work was supported in part by "Penugasan Publikasi Artikel di Jurnal Internasional Kuartil Q1Q2" Grant from Universitas Indonesia with number: NKB0211/UN2.R3.1/HKP.05.00/2019.

References

[1] A. Pierson, Z. Wang, M. Schwager, Intercepting rogue robots: An algorithm for capturing multiple evaders with multiple pursuers, IEEE Robot. Autom. Lett. 2 (2) (2017) 530–537.

[2] L. Gregorin, S.N. Givigi, E. Freire, E. Carvalho, L. Molina, Heuristics for the multi-robot worst-case pursuit-evasion problem, IEEE Access 5 (2017) 17552–17566.

[3] S. Jia, X. Wang, L. Shen, A continuous-time Markov decision process-based method with application in a pursuit-evasion example, IEEE Trans. Syst. Man Cybern. Syst. 46 (9) (2016) 1215–1225.

[4] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, J. Chen, Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges, IEEE Commun. Mag. 56 (4) (2018) 68–74.

[5] A.Y. Husodo, G. Jati, A. Octavian, W. Jatmiko, Enhanced social spider optimization algorithm for increasing performance of multiple pursuer drones in neutralizing attacks from multiple evader drones, IEEE Access 8 (2020) 22145–22161.

[6] A. Mardani, M. Chiaberge, P. Giaccone, Communication-aware UAV path planning, IEEE Access 7 (2019) 52609–52621.

[7] W. Shi, et al., Multi-drone 3-D trajectory planning and scheduling in drone-assisted radio access networks, IEEE Trans. Veh. Technol. 68 (8) (2019) 8145–8158.

[8] M.C.P. Santos, C.D. Rosales, M. Sarcinelli-Filho, R. Carelli, A novel null-space-based UAV trajectory tracking controller with collision avoidance, IEEE/ASME Trans. Mechatronics 22 (6) (2017) 2543–2553.

[9] K. McGuire, G. de Croon, C. De Wagter, K. Tuyls, H. Kappen, Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone, IEEE Robot. Autom. Lett. 2 (2) (2017) 1070–1076.

[10] S.J. Yoo, J.H. Park, S.H. Kim, A. Shrestha, Flying path optimization in UAV-assisted iot sensor networks, ICT Express 2 (3) (2016) 140–144.

[11] A.Y. Husodo, H.A. Wisesa, W. Jatmiko, Dynamic motion planning for conducting obstacle avoidance maneuver of fixed wing autonomous aerial vehicle, in: 2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Nagoya, Japan, 2019, pp. 78–83.

[12] A.Y. Husodo, H.A. Wisesa, W. Jatmiko, The use of direction matrix to determine autonomous quad-copter drone path when facing obstacle, in: 2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Nagoya, Japan, 2019, pp. 89–94.

[13] A.Y. Husodo, G. Jati, N. Alfiany, W. Jatmiko, Intruder drone localization based on 2D image and area expansion principle for supporting military defence system, in :2019 IEEE International Conference on Communication, Networks and Satellite (Comnetsat), Makassar, Indonesia, 2019, pp. 35–40.

[14] Y. Zeng, R. Zhang, Energy-efficient UAV communication with trajectory optimization, IEEE Trans. Wireless Commun. 16 (6) (2017) 3747–3760.

[15] C. Desset, et al., Flexible power modeling of LTE base stations, in :2012 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, 2012, pp. 2858–2862.

[16] C.D. Franco, G. Buttazzo, Energy-aware coverage path planning of UAVs, in: 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, Vila Real, 2015, pp. 111–117.