International Journal of Future Generation Communication and Networking
Vol. 9, No. 2 (2016), pp. 83-94
http://dx.doi.org/10.14257/ijfgcn.2016.9.2.09

Achieving Maximal NJRL Method for Jamming UAVs Network

Zhang Yu[1], Peng Xiaodong[2] and Liu Feng[1]

[1]School of Computer and Information Science, Southwest University, Chongqing, China
[2]Neijiang Normal University
zhangyu@swu.edu.cn, 348281716@qq.com, liuf@swu.edu.cn

Abstract

Different from others, we focus on designing jamming methods for disrupting UAVs network efficiently. The jamming problem is formulated. Then we introduce two new methods (AMN-U2C and AMN-U2U) for jamming UAVs network. Both of them adopt the idea intending to achieve maximal number of jammed receiving links. AMN-U2C method searches locations for jammer by moving jammers along the path from the point of a UAV to the center point of others. AMN-U2U method searches by shifting jammers from the point of a UAV to another UAV. Finally we simulate the jamming attack on UAVs network using AMN-U2C, AMN-U2U, TP, GA and Random methods. The result shows AMN-U2U does best at most cases.

Keywords*: Jamming, UAVs network, Achieving maximal NJRL, AMN-U2C, AMN-U2U*

1. Introduction

UAVs are used for a wide range of missions[1]. UAVs network is built upon a shared communication medium. It is possible to launch jamming attacks in wireless sensor networks[2]. The various types of jamming is provided in [3]. Ref. [2, 4] do work for detecting the jamming attacks. Ref. [5-8] try to avoid jamming attack and protect the communications. In [5, 6] the UAVs jamming problem is formulated as a zero-sum pursuit-evasion game, and Isaacs' approach is used to obtain motion strategies for a pair of UAVs to evade the jamming attack. Ref. [7] presents some methods for jamming prevention. Ref. [8] introduces mechanisms which attempt to protect the network from jamming attacks.

Some techniques of jamming attack are presented in [3,9]. Jamming attack and network defense problems are studied for wireless sensor networks in [10]. The authors introduce a heuristic algorithm for an efficient jamming strategy. Ref. [6] provide motion strategies for an jammer to jam the communication between a pair of UAVs. Our previous work[11] studies the problem of jamming UAVs network. Triangle method and GA (Genetic Algorithm) based method are introduced. Here we provide our new jamming methods for achieving better jamming performance. Our contributions to the UAVs jamming attack research are as follows: 1) we extend Triangle method from [11] to TP(Three Points) method. The latter one takes a weaker condition than the former one; 2) we introduce AMN-U2C (Achieving maximal NJRL – from a UAV to the center point of other unhandled UAVs) jamming methods for attacking the UAVs networks. The idea of AMN-U2C is to achieve maximal number of jammed receiving links; 3) We also present AMN-U2U(Achieving maximal NJRL – from a UAV to other UAVs) jamming method which uses the same ideal of AMN-U2C, but uses improved greedy searching strategy and performs better; 4) We simulate 5 jamming methods (3 from this paper and 2 from [11]), compare and analyze simulation results, and give out our recommendation for how to choose the UAVs jamming method.

The rest of the paper is organized as follows. The UAVs jamming problem is introduced and analyzed in Section 2. Triangle method is extended and TP is presented in Section 3. New methods AMN-U2C and AMN-U2U are introduced and described in detail respectively in Section 4 and 5. The simulation is carried out and the result is analyzed in Section 6. Finally, Section 7 concludes the paper.

2. Jamming Problem Statement

Here we assume jammers have speed as fast as expected. At this case, the jamming is not sensitive about UAV network topology. The Nicholson JSR model [12] $\xi=(P_{JT}G_{JR}G_{RJ}/P_TG_{TR}G_{RT})10^{4\log_{10}(D_{TR}/D_{JR})}$ is used in this paper, where P_{JT} is the power of the jammer's transmitting antenna, P_T is the power of the transmitter, G_{TR} is the antenna gain from transmitter to receiver, G_{RT} is the antenna gain from receiver to transmitter, G_{JR} is the antenna gain from jammer to receiver, G_{RJ} is the antenna gain from receiver to jammer, h_J is the height of the jammer antenna above the ground, h_T is the height of the transmitter antenna above the ground, D_{TR} is the Euclidean distance between transmitter and receiver, and D_{JR} is the Euclidean distance between jammer and transmitter.

$U=\{1,2,...,n\}$ denotes the set of UAVs, n is an integer and $n\geq 1$; $J=\{1,2,...,m\}$ is the set of Jammers, m is an integer and $m\geq 1$; Sometimes u_i represents UAV i; l_i and l_j are the location of u_i and Jammer j. The location of UAV is assumed to be restricted into its task area L. UAVs have bidirectional communication links. UAV u_i uses $Link_{i\to k}$ to send packets to u_k. UAV u_k uses $Link_{i\to k}$ to receive packets from u_i. $Link_{i\to *}$ / $Link_{*\to i}$ are used to denote UAV i 's all sending links and all receiving links respectively.

The jamming effect is denoted as JE_{jik}. Let $c=(P_{JT}G_{JR}G_{RJ}/(\xi P_TG_{TR}G_{RT}))^{1/4}$, if $D_{JR}\leq cD_{TR}$, then $JE_{jik}=1$; else $JE_{jik}=0$ .When $JE_{jik}=1$, UAV u_k will not get packets from i, i.e. $Link_{i\to k}$ is jammed by jammer j. Otherwise, $Link_{i\to k}$ is not jammed by jammer j. The jamming problem can be modeled as $Max(z=\sum_{j\in J;i,k\in U}JE_{jik})$, subject to $l_i\in L, i\in U$.

3. Three Point (TP) Method

3.1. Problem Analysis

Proposition 1: when jammer j is placed at the same location of UAV i, $Link_{*\to i}$ will be disrupted.

Proof: When jammer j is placed at the same location of UAV i, D_{JR} will be zero. $D_{JR}\leq cD_{TR}$ always holds. Therefore, $Link_{*\to i}$ will be disrupted by jammer j. □

Proposition 2: if $c\geq 1$, when jammer j is placed at the same location of UAV i, $Link_{*\to i}$ and $Link_{i\to *}$ can be disrupted at the same time.

Proof: It is possible to disrupt $Link_{*\to i}$ (Prop.1). When jammer j is placed at the same location of UAV i, D_{JR} equals to zero. a) For any other UAV k which sends packets to UAV i, $D_{JR}\leq D_{TR}$ is true, $Link_{k\to i}$ is disrupted. Therefore $Link_{*\to i}$ will be disrupted; b) For any other UAV k which receives packets from UAV i, $D_{JR}=D_{TR}$ is true, $Link_{i\to k}$ is disrupted. Therefore $Link_{i\to *}$ will be disrupted.□

Proposition 3: for jamming two UAVs (i and k) using one jammer j , if condition $c \geq 1$ holds, $Link_{i\to k}$ and $Link_{k\to i}$ can be disrupted at the same time.

Proof: When using one jammer to jam two UAVs, there exist locations for a jammer to make $D_{JR} \leq cD_{TR}$ is true.

1) When UAV i and k locate at a same location. We can find a location for jammer j to disrupt $Link_{*\to i}$ (Prop.1). For UAV i and k have same location, $Link_{*\to k}$ will be disrupted. Therefore $Link_{i\to k}$ and $Link_{k\to i}$ can be disrupted at the same time.

2) When UAV i and k locate at different locations. It is easy to find locations for a jammer, through a) Drawing a line to connect UAV i and k ; b) Selecting a point on the line. When jammer j locates on one of these locations, $D_{JR} \leq D_{TR}$ holds for both UAV i and k . As shown in Figure 1 a), jammer j locates at any point on the line between UAV i and k , the distance between the transmitter and the receiver is D_{ik} ($D_{ki} = D_{ik}$); the distance between the receiver and the jammer is D_{ij} and D_{kj} . For $D_{ij} \leq D_{ik}$ and $D_{kj} \leq D_{ik}$, so $D_{JR} \leq cD_{TR}$ holds.□

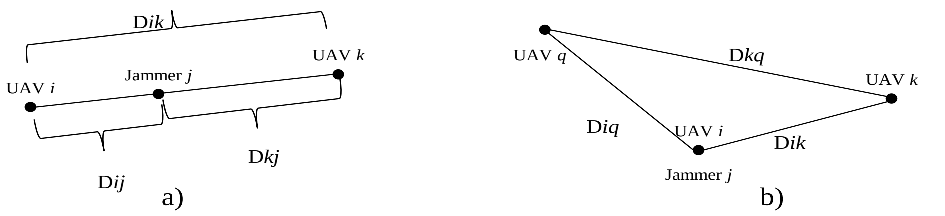

Figure 1. Jamming Two UAVS with One Jammer

Theorem 1: for jamming any 3 UAVs (UAV i , k and q) with one jammer j , if condition $c \geq 1$ hold, all links among these UAVs can be disrupted at the same time.

Proof:

1) When UAV i , k and q locate at the same location. Proposition 1 has shown that $Link_{*\to i}$, $Link_{*\to k}$, $Link_{*\to q}$ can be disrupted at the same time.

2) When two UAVs locate at same one location and the third one locates at another different location. Without loss of generality, let UAV i and k locate at location loc_1 , UAV q locates at location loc_2 . If jammer j is placed at loc_1 , a) $Link_{i\to k}$ and $Link_{k\to i}$ will be disrupted (Prop.1); b) $Link_{i\to q}$, $Link_{q\to i}$, $Link_{k\to q}$ and $Link_{k\to i}$ can also be disrupted (Prop.3).

3) When UAV i , k and q locate at different location loc_i , loc_k and loc_q respectively. The distance between two UAVs is D_{ik} , D_{iq} and D_{kq} . Without loss of generality, let $D_{ik} \leq D_{iq} \leq D_{kq}$ as shown in Figure 1 b). Considering such a case that jammer j is placed at loc_i . a) $Link_{i\to k}$, $Link_{i\to q}$, $Link_{k\to i}$ and $Link_{q\to i}$ will be disrupted (Prop.2); b) $Link_{k\to q}$ is disrupted because of $D_{iq} \leq D_{kq}$; c) $Link_{q\to k}$ is disrupted because of $D_{ik} \leq D_{kq}$.□

3.2. TP Jamming Algorithm

We present TP jamming method for the UAV network. It is based on Theorem 1. The detail of TP method is shown in Algo 1. Step T1-3 initializes the variables. All UAVs are put into a temporary set S_{UAV} in T1. Variable LOC is a set used to store the locations found for jammers. It is initialized as an empty set in T2. Variable u_{last} save the UAV searched last time. It is used to avoid searching for a UAV twice. Next searching process will start with a UAV different from u_{last}.

Step T4-21 is used to find appropriate locations for jammers. T4 starts a loop. The loop stops till all UAVs are removed from S_{UAV} .Variable $Foundq$ denotes whether a location is found. At the begging of iteration, it is initialized to $false$. T6 computes the number of UAVs in S_{UAV}. If there is only one UAV, then the location of the UAV will be added to set LOC in T7. T8 checks whether the number of elements in S_{UAV} is 2. If so, the midpoint of the two UAVs is added to set LOC. Function $\mathrm{GetMidLoc}$ in T9 returns the midpoint of two UAVs.

If there are 3 or more UAVs in S_{UAV}, we use a third point searching method to find the locations for jammers. Function $\mathrm{RandomUAV}(UAVSet,u)$ in T11 and T12 returns a UAV in $UAVSet$, which is different from u. The search begins with two random UAVs u_i and u_k ($u_i \neq u_k$, $u_i \neq u_{last}$ and $u_k \neq u_{last}$). A temporary set S_{UAV2} is used to save the UAVs in S_{UAV} except u_i and u_k. Then T14-15 starts a for loop to search the third UAV u_q which meets the condition $D_{ik} \leq D_{kq}$ and $D_{iq} \leq D_{kq}$. If a UAV is found, $Foundq$ is set to $true$ in T16, and u_q is removed from S_{UAV} in T17. The for iteration continues till all UAVs in S_{UAV2} are visited. When the for loop stops, T18 checks whether one or more UAVs are found. If found, the location of u_i is added to set LOC in T19, u_i and u_k are removed from S_{UAV} in T20.

Variable u_{last} is replaced with u_i in T20, and a new while iteration will begin at T4. When the while loop (T4) stops, all locations are found and saved in set LOC. Function $\mathrm{AssignLoc}(LOC, J)$ in T22 assigns the found locations to jammers. If $|LOC|$ (the number of found locations) is equal to or bigger than $|J|$ (the number of jammers), then first $|J|$ locations in LOC are assigned to the jammers, as shown in step AL1-3. If $|LOC| \leq |J|$, all locations are assigned to $|LOC|$ jammers, and the rest of other jammers are assigned with random values in AL4-8.

3.3. Computational Complexity

In Algorithm 1, the while loop (step 4) will run N times at most, N is the number of UAVs. The for loop (step 14) will run $(\mathrm{N}-2)$ times at most. So the code from step 1 to 21 has computational complexity of $\mathrm{O}(\mathrm{N}^2)$. The code of step A.1 to A.3 has computational complexity of $\mathrm{O}(\mathrm{N})$. The code of step A.4 to A.8 has computational complexity of $\mathrm{O}(\mathrm{M})$, where M is the number of jammers. The computational complexity of function $\mathrm{AssignLoc}$ is $\mathrm{O}(\max(\mathrm{N},\mathrm{M}))$. Therefore TP algorithm has $\mathrm{O}(\mathrm{N}^2)$ computational complexity.

4. AMN-U2C Jamming Method

4.1. Analysis

If a jammer moves towards a UAV, the distance between them (D_{JR}) will decrease. If a jammer get close enough to a UAV, $D_{JR} \leq cD_{TR}$ may hold. We define $AJRL$ to represent an area in which the receiving link is disrupted by a jammer. $AJRL_{i \to k}$ is an area in which if a jammer located, the receiving link from UAV i to k will be disrupted. The shape of $AJRL_{i \to k}$ is a sphere, and the center of the sphere is the point that UAV k locates. A UAV has $n-1$ $AJRLs$, n is the number of UAVs,. As shown in Figure 2, $AJRL_{1 \to 2}$ represents the jamming area in which if a jammer locates, UAV 2 is not able to successfully receive packets from UAV 1. The radius of $AJRL_{i\to k}$ depends on the distance between UAV i and k. The whole area, which is a union $\sum_{i\in U} AJRL_{i\to k}$, is referred as $AJRL_{*\to k}$. When a jammer is placed at a certain location, it may locate in zero, one, two or more $AJRLs$. $NJRL$ represents the number of receiving links jammed by a jammer in a specific location. $NJRL$ will be different if jammer is placed at different points, as shown in Figure 2.

4.2. AMN-U2C Algorithm

AMN-U2C method is based on such an idea that 1) estimating the NJRL according to UAVs' $AJRLs$; 2) searching locations on the line from a UAV to the center of other UAVs, as shown in Figure 3. The location with maximal NJRLs is selected; 3) avoiding placing a jammer to locations already covered.

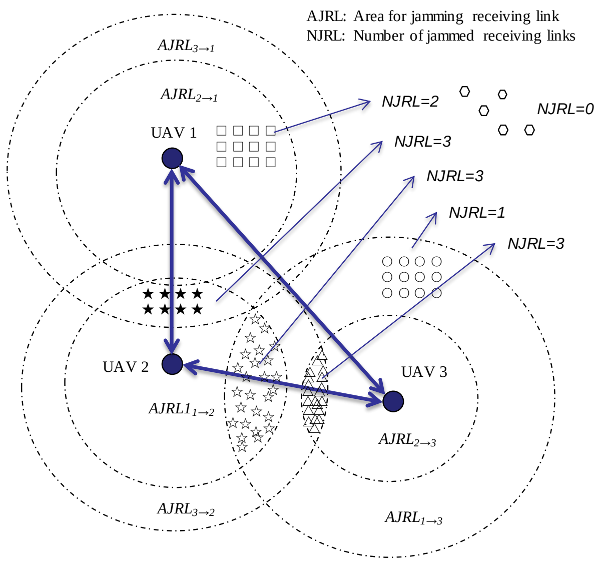

Figure 2. AJRL and NJRL

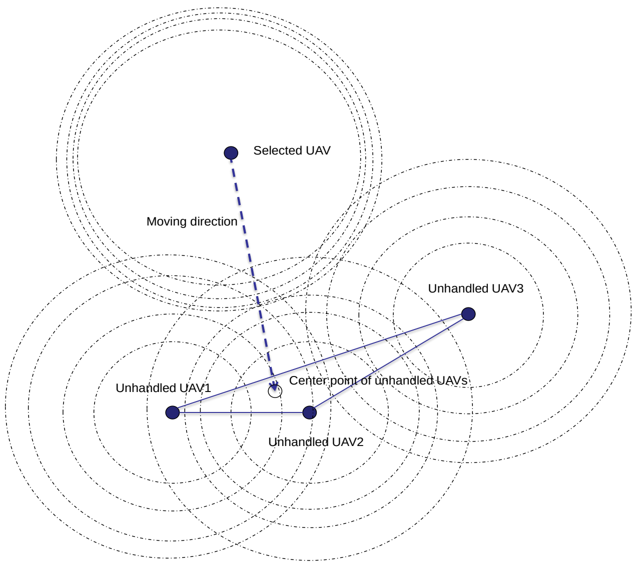

Figure 3. Moving Direction

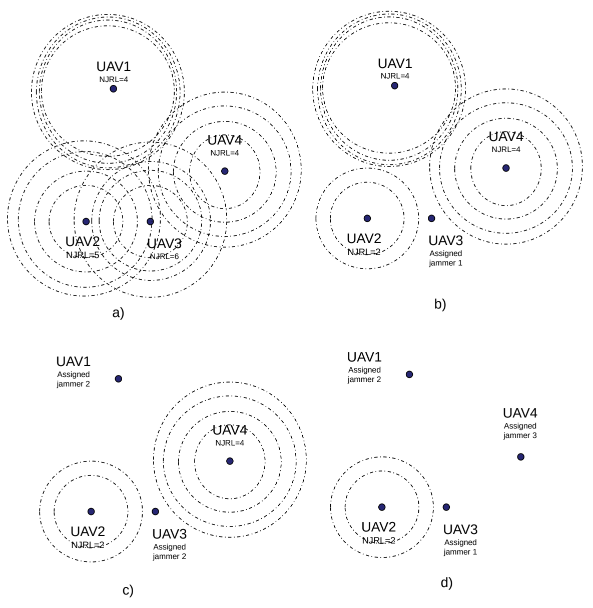

Figure 4. AMN-U2U

AMN-U2C algorithm (Alog.2) firstly calls function $\mathrm{AJRL}()$ to generate all $AJRL$ s in step M1. Then it begins to search locations for jammers from M2 to M14. When a jammer is assigned a location, it will be marked as assigned. One jammer j is selected in M3, which has not been assigned a location. A UAV u unhandled is selected in M4, and its location will be set as the starting point p_s of jammer j in M6. If such an unhandled UAV does not exist, then the while loop (M2) will be broken, and the algorithm go to M15. In M7, the center point of all unhandled UAVs will be computed and save to variable p_c. Point p_c is set as the destination of the searching path. AMN-U2C algorithm uses L steps/iterations to move jammer j from p_s to p_c (M8). For each iteration (from M9 to M11), function $\mathrm{Fitness}()$ is used to compute jammer j 's jamming effect, and the returned result is saved to variable f . Fitness f is then compared with the temporary best fitness value. If f is greater, then the temporary best fitness value is updated to f , and the jammer j 's current location is used to replace the best point p_b. After achieving the best position, it is assigned to jammer j in M12. UAV u is marked as handled and jammer j is marked as assigned in M15-16, to avoid searching them again.

For the while loop (from M2 to M14) may be broken, when an unhandled UAV does not exist, there may exist jammers which are unassigned locations. Therefor step M15 and M16 are used to assign random locations to all unassigned jammers.

Function $\mathrm{AJRL}()$ calculates all $AJRL$ s and saves them in a two-dimensional array $arrAJRL$ (A1 to A6). Function $\mathrm{Pos}()$ returns the position of a UAV or jammer. Function $\mathrm{Dis}()$ returns the distance between two points. The center of $AJRL$ is set to the position of the UAV in A4. Step A4-5 compute d_{ik} (the distance between two UAVs), and the radius of $AJRL$ is set to cd_{ik} .

The function $\mathrm{Fitness}(point)$ (from Figure 1-7) is used to compute the fitness value of a point. The total jammed links is saved to variable $njrl$ which is initialized to 0 in Figure 1. Firstly, it checks whether the point locates in a UAV's $AJRL$ .The distance between the point and a specific UAV is computed and saved in variable *dis* in Figure 3. If *dis* is less than the UAV *AJRL* 's radius, variable *njrl* (initialized as 0 in Figure 1) is increased by 1 in Figure 6. That means if a jammer placed at the point, the UAV u_i cannot get packets from u_k through the receiving link. The process continues after all UAV's *AJRLs* are checked. Finally, the *njrl* of the point is returned as the fitness value.

Algorithm 1: TP algorithm

```
TP()
T1.  $S_{\text{UAV}} = U$
T2.  $LOC = \varnothing$
T3.  $u_{last} = nil$
T4.  while $S_{\text{UAV}} \neq \varnothing$
T5.    $Foundq=false$
T6.    if $|S_{\text{UAV}}| = 1$
T7.      $LOC = LOC + S_{\text{UAV}}[1].Location$
T8.    if $|S_{\text{UAV}}| = 2$
T9.      $LOC = LOC + \text{GetMidLoc}(S_{\text{UAV}}[1], S_{\text{UAV}}[2])$
T10.   if $|S_{\text{UAV}}| \geq 3$
T11.     $u_i = \text{RandomUAV}(S_{\text{UAV}}, u_{last})$
T12.     $u_k = \text{RandomUAV}(S_{\text{UAV}} - u_i, u_{last})$
T13.     $S_{\text{UAV2}} = S_{\text{UAV}} - u_i - u_k$
T14.     for $q$=Low($S_{\text{UAV2}}$) to High($S_{\text{UAV2}}$)
T15.       if $(D_{ik} \leq D_{kq})$ and $(D_{iq} \leq D_{kq})$
T16.         $Foundq=true$
T17.         $S_{\text{UAV}} = S_{\text{UAV}} - u_q$
T18.   if $Foundq$
T19.     $LOC = LOC + u_i.Location$
T20.     $S_{\text{UAV}} = S_{\text{UAV}} - u_i - u_k$
T21.   $u_{last} = u_i$
T22. AssignLoc($LOC, J$)
Function AssignLoc($LOC, J$)
AL1  if $|LOC| \geq |J|$
AL2    for $a$=Low($J$) to High($J$)
AL3      $J[a].Location = LOC[a].Location$
AL4  if $|LOC| \leq |J|$
AL5    for $a$=Low($LOC$) to High($LOC$)
AL6      $J[a].Location = LOC[a].Location$
AL7    for $b$=High($LOC$)+1 to High($J$)
AL8      $J[b].Location = RandomLocation$
FUNCTION AJRL()
A1.  for each $u_i \in U$
A2.    for each $u_k \in U$
A3.      $d_{ki} = \text{Dis}(\text{Pos}(u_i), \text{Pos}(u_k))$
A4.      $arrAJRL_{k\to i}.\text{Center} = \text{Pos}(u_i)$
A5.      $arrAJRL_{k\to i}.\text{Radius=cd}_{ik}$
A6.  return $arrAJRL$
```

Algorithm 2: AMN-U2C algorithm

```
AMN-U2C()
M1.  AJRL()
M2.  while not all jammers are assigned locations
M3.    select one unassinged jammer $j$
M4.    search one unhandled UAV $u$
M5.      if $u$ doese not exist then break
M6.    set the point of $u$ as the start point $p_s$
M7.    compute $p_c$ of other unhandled UAVs
M8.    move $p_s$ to $p_c$ using L steps
M9.    for each step compute $f = \text{Fitness}()$
M10.     if $f$ is the best one then
M11.       save current point as $p_b$
M12.   assign $p_b$ to jammer $j$
M13.   mark UAV $u$ is handled
M14.   mark jammer $j$ is assigned
M15. for each unassigned jammer $j$
M16.   assign random location to $j$
FUNCTION Fitness($point$)
F1.  $njrl = 0$
F2.  for each $u_i \in U$
F3.    $dis = \text{Dis}(\text{Pos}(u_i), point)$
F4.    for each $u_k \in U$
F5.      if ($dis \leq arrAJRL_{ki}.Radius$)
           and ($arrAJRL_{ki}$ is not coverd)
F6.        $njrl++$
F7.  return $njrl$
```

Algorithm 3: : AMN-PSO algorithm

```
AMN-PSO()
N1.  AJRL()
N2.  for each jammer $j \in J$
N3.    $f_{\text{best}} = 0$
N4.    for $u \in U$
N5.      $f$=Fitness(Pos($u$))
N6.      if $f > f_{\text{best}}$
N7.        $f_{\text{best}} = f$
N8.        $u_{\text{best}} = u$
N9.    assign Pos($u_{\text{best}}$) to jammer $j$
N10.   remove ARJLs coverd by $j$
```

4.3. Computational Complexity

In Algorithm 2, it is obvious that both function AJRL() and function Fitness() have computational complex of $\text{O}(\text{N}^2)$, where N is the number of UAVs. The while loop (from step M2 to M16) has M iterations at most. The computational complex of step M3 is $O(M)$ and $O(N)$ respectively. When computing the center point of unhandled UAVs (in step M7), at most N iterations are needed. The for loop (in step M9) runs L times. During each iteration, the fitness of the jammer is computed by calling function $\text{Fitness}()$. So the computational complex from step M8 to M11 is $O(LN^2)$. The step from M15 to M16 has computational complex of $O(M)$. Therefore, the overview computational complex of AMN-U2C algorithm is $O(N^2)+O(M(M+N+N+LN^2))+O(M)=O(M(M+LN^2))=O(M^2+LMN^2)$.

5. AMN-U2U (Achieving Maximal NJRL-From A UAV to Another UAV)

5.1. Analysis

AMN-U2U method is based on such an idea that 1) estimating the NJRL according to UAVs' $AJRLs$; 2) searching locations by moving jammers from one UAV to another. The location with maximal NJRLs is selected; 3) avoiding placing a jammer to locations already covered.

The AJRLs are computed firstly. AMN-U2U computes the NJRLs of each UAV's location. Secondly, it selects the location with maximal NJRLs and assigns the location to a jammer. Thirdly, the AJRLs which are covered by the jammer are removed. By doing this, it can avoid placing two or more jammers in a same location or its neighborhood. Again, the AMN-U2U computes the NJRLs of each UAV, selects the maximal one for the next jammer and removes the related AJRLs. The process goes on till all jammers are assigned locations.

For example, the AJRLs of four UAVs and the NJRLs of each UAV's location are computed and shown in Figure 4 a). UAV 3 has maximal NJRL of 6. It is selected firstly. Its location is assigned to jammer 1 and all AJRL covered by jammer 1 are removed, as shown in Figure 4 b). Secondly, as shown in Figure 4 c), UAV 1 is selected; its location is assigned to jammer 2; and all AJRL covered by jammer 2 are removed. Thirdly, UAV 4 is selected and the next operation is shown in Figure 4 d). The process goes on till all jammers are assigned locations.

5.2. AMN-U2U Algorithm

AMN-U2U algorithm (Algorithm 3) firstly calls function $\text{AJRL}()$ to generate all $AJRLs$ in step N1. It looks for locations for each jammer. Variable f_{best} saves the best fitness value and is initialized to 0 in step N3. A for loop (from N4 to N8) is used to find the UAV which has best fitness value. During each iteration, it computes the fitness value of UAV u and saves to variable f in step N5. From N6 to N8, if f is greater than f_{best}, then f_{best} is updated by f and UAV u is saved as the best UAV variable u_{best}. After the best UAV is found, its location is assigned to a jammer in step N9. Then all ARJLs covered by this jammer will be removed in step N10. The process goes on till all jammers get assigned.

5.3. Computational Complexity

In Algorithm 3, function $\text{AJRL}()$ and function $\text{Fitness}()$ have computational complex of $O(N^2)$, where N is the number of UAVs. So the for loop (from N4 to N8) has computational complex of $O(N^3)$.The main loop (from step N2 to N10) has M iterations at most. The overall computational complex of AMN-U2U is $O(MN^3)$.

6. Simulation

For evaluating the jamming performance, we define a metric RJT which represents the ratio of jammed links to total links. It is computed by $\sum_{i,k \in U, i \neq k} \text{JammedLink}_{i \to k} / \sum_{i,k \in U, i \neq k} \text{Link}_{i \to k}$. The simulation scenario is set as the same in [11]. The moving radius and height are set to 20km and 9km respectively. The simulation uses $\{1,2,..,20\}$ jammers to jam the UAVs communication network with $\{20,30,..,120\}$ UAV nodes respectively. The Random method, GA method and Triangle method (extended as TP method here) from [11] are also used for comparing.

From Figure 5 6, it is obvious that for any kind of jamming method, more jammers yield higher RPL. It is easy to understand, many hands make light work. Normally, a jammer blocks some communication links in specific UAVs networks. Suppose, $m\text{-}1(m>1)$ jammers are used to jamming a UAVs network, when the $m\text{th}$ jammer is added, the number of total blocked links may increase. For there may exist links $\Delta_{links}(\geq 0)$ which can be blocked by the $m\text{th}$ jammer, but are not blocked by other $m\text{-}1$ jammers. When the $m\text{th}$ jammer is added, the number of total blocked links will increase by Δ_{links} . Therefore, the jamming performance may be better when using more jammers, regardless of the jamming methods adopted.

The Random jamming method performs worst at most cases. That is because it randomly places the jammers without any optimizing operations. The GA jamming method uses genetic algorithm to search locations for jammers. It performs better than the Random jamming method. TP jamming method is the extension of Triangle method[11]. It does similarly as Triangle method does. If JSR is set to 0.2, when jamming the UAVs network with nodes number range from 20, 30 to 120 respectively, TP method does better than both GA method and Random method.

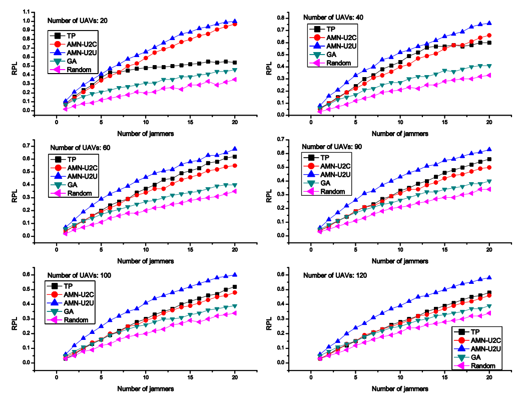

Figure 5. Simulation Result (JSR C=0.2

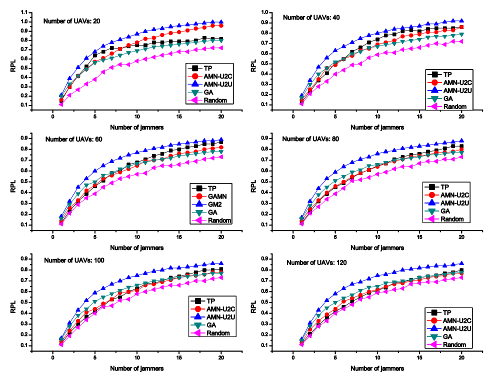

Figure 6. Simulation Result (JSR C=0.4)

It is easy to get that AMN-U2C does better than GA and Random method. AMN-U2C method performs better than TP method at following cases. 1) When the UAVs network has 20 nodes, using 7 or more jammers; 2) When the UAVs network has 40 nodes, using 16 or more jammers. As JSR c is set to 0.2, the condition of $c \geq 1$ does not hold, so TP method may not assure all the links of the three nodes can be blocked by one jammer. But it can block some of them. With the number of jammers increasing, the performance increases, and TP method may output better result than AMN-U2C does. It is obvious in Figure 5 and Figure 6 AMN-U2U jamming method is the best one among these methods. It outputs best jamming result at all cases.

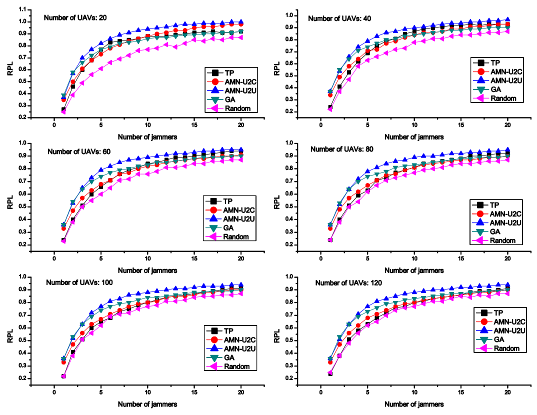

Figure 7. Simulation Result (JSR C=0.6)

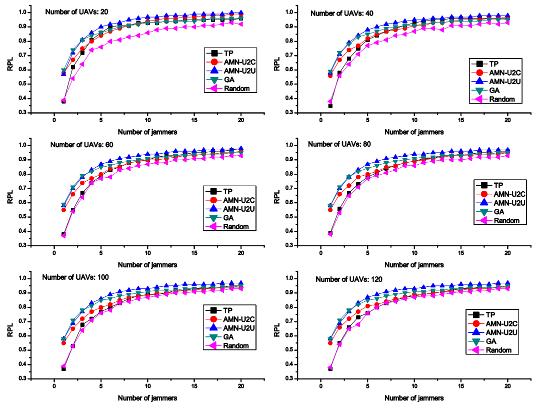

Figure 8. Simulation Result (JSR C=0.8)

Figure 7 and 8 show the similar result as Figure 5 and 6: 1) For all methods, more jammers yield higher jamming performance; 2) Random method output the worst result. GA method does better than Random method; 3) AMN-U2C and TP method do better than GA does. At most cases, TP method can output higher RPLs than TP method. However, sometimes AMN-U2C performs better; 4) AMN-U2U outputs best jamming result at all cases among the five jamming methods.

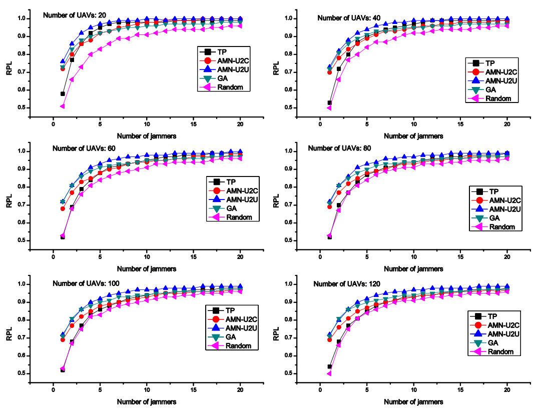

Figure 9. Simulation Result (JSR C=1.0)

Figure 9 and Figure 10 show the results which simulated with variable $c = 1.0$ and $c = 1.2$ respectively. The same conclusion can also be drawn as previously described. Furthermore, with a higher variable $c\ (\geq 1)$, the jamming performances of these methods get closer. The difference among Random, GA, TP and AMN-U2C methods is not significant anymore. However, AMN-U2U is clearly the best one.

7. Conclusion

UAVs network are wildly used, especially in military fields. Unlike researches on avoiding jamming attacks, we focus on designing methods for efficiently disrupting the communication network. Firstly, the UAVs jamming problem is formulated and analyzed. Secondly, Triangle method is extended to TP method. Triangle method is derived from condition. In TP method, the condition becomes which is weaker. Thirdly and most importantly, we introduce two new methods. Both AMN-U2C and AMN-U2U methods intend to achieve maximal number of jammed receiving links. AMN-U2C method searches locations for jammer by moving jammers along the path from the point of a UAV to the center point of others. However, AMN-U2U method searches locations by shifting jammers from the point of a UAV to another UAV. Finally, AMN-U2C, AMN-U2U, TP methods, including GA, Random methods from [11], are simulated. The simulation result shows: 1) for all methods, more jammers lead to higher jamming performance; 2) Random method output the worst result. GA method does better; 3) AMN-U2C and TP method do better than GA does. At most cases, TP method can output higher RPLs than TP method. However, sometimes AMN-U2C performs better; 4) with a higher variable , the difference among Random, GA, TP and AMN-U2C methods is not significant anymore; 5)AMN-U2U outputs best jamming result at all cases among the five jamming methods.

The computational complex of AMN-U2U, AMN-U2C, TP, GA, Random are, , , and , where is the number of UAVs, is the number of jammers, is the steps from one point to the center point of UAVs, and are the number of solutions and the number of evolution iterations in GA method. Therefore, when jamming UAVs network, we suggest 1) if the number of UAVs is not great, AMN-U2U should be selected; 2) if fast computing is required, TP jamming method is the first choice; 3) if variable is big enough, AMN-U2U, AMN-U2C, TP and GA methods are all acceptable.

Acknowledgements

This research is supported by Chongqing Intelligent Instrument and Control Equipment Engineering Technology Research Center.

References

[1] K. Hartmann and C. Steup, "The vulnerability of UAVs to cyber attacks-An approach to the risk assessment", 5th International Conference in Cyber Conflict (CyCon), **(2013)**, pp. 1-23.
[2] W. Xu, W. Trappe, Y. Zhang and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks", Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, **(2005)**, pp. 46-57.
[3] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs", Communications Surveys & Tutorials, IEEE, vol. 11, **(2009)**, pp. 42-56.
[4] M. Singh, K. Babbar and K.L. Jain, "A Survey on Intrusion Detection System in Wireless Sensor Networks", International Journal, vol. 3, **(2014)**.
[5] S. Bhattacharya and T. Basar, "Game-theoretic analysis of an aerial jamming attack on a UAV communication network", American Control Conference (ACC), **(2010)**, pp. 818-823.
[6] S. Bhattacharya and T. Başar, "Differential game-theoretic approach to a spatial jamming problem", Advances in Dynamic Games, ed: Springer, **(2013)**, pp. 245-268.
[7] P. Chaturvedi and K. Gupta, "Detection and Prevention of various types of Jamming Attacks in Wireless Networks", ed: IJCNWC, **(2013)**.
[8] K. Pelechrinis, M. Iliofotou and S.V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers, IEEE Communications Surveys and Tutorials", vol. 13, **(2011)**, pp. 245-257.
[9] W.Y. Xu, K. Ma, W. Trappe and Y.Y. Zhang, "Jamming sensor networks: Attack and defense strategies", IEEE Network, vol. 20, **(2006)** May-June, pp. 41-47.
[10] M.Y. Li, I. Koutsopoulos and R. Poovendran, "Optimal Jamming Attack Strategies and Network Defense Policies in Wireless Sensor Networks", IEEE Transactions on Mobile Computing, vol. 9, **(2010)** August, pp. 1119-1133.
[11] Y. Zhang and L.S. Yang, "Triangle and GA Methods for UAVs Jamming", Mathematical Problems in Engineering, vol. 2014, **(2014)**, p. 8.
[12] R. Poisel, "Modern communications jamming principles and techniques", Artech House Publishers, **(2011)**.

Authors

Zhang Yu, He is an associate professor in Southwest University of China. His main research interests include wireless network jamming, industrial wireless communication, etc.