

# بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه فردوسی مشهد

گزارش فنی

شبکه های کامپیوتری پیشرفته، بهار ۱۳۹۲

## بررسی چند روش مکانیابی نودهای متحرک

استاد: جناب آقای دکتر سید امین حسینی

گردآورنده: علی آقاخانی

شماره دانشجویی: ۹۰۶۳۸۵۷۱۰۴

بهار ۱۳۹۲

## فهرست مطالب

۳	۱- مقدمه .....
۴	۲- الگوریتم مکانیابی (SALMN(Self Adapting Localization for Mobile Nodes) .....
۵	۲-۱- تنظیمات خود سازگار برای حرکت نودها .....
۶	۳- مکانیابی با استفاده از روش TDOA .....
۶	۳-۱- محاسبات TDOA .....
۱۰	۳-۲- مشکل حل معادله کوچکترین مربعات غیرخطی .....
۱۰	۳-۳- الگوریتم‌ها .....
۱۲	۴- مکانیابی با استفاده از cluster-based MDS .....
۱۳	۴-۱- کلاسترینگ k پرشه برای مکان یابی .....
۱۵	۴-۲- ساختن اتصال inter – cluster با استفاده از نودهای Inter – cluster .....
۱۶	۴-۳- MDS در هر کلاستر .....
۱۹	۵- بررسی یک روش مکانیابی تئوری .....
۱۹	۵-۱- محدودیت فاصله ای بر مبنای تحلیل مکان یابی .....
۲۰	۵-۲- شرایط برای Rigidity .....
۲۱	۵-۳- الگوریتم‌هایی برای شناسایی Rigidity .....
۲۲	۵-۴- ساختار استنتاجی .....
۲۳	۶- فهرست منابع .....

## ۱- مقدمه:

شبکه های WSN، معمولاً از تعداد بسیاری نود حسگر تشکیل شده اند که معمولاً در محیطی به طور تصادفی پخش شده اند، باید توجه داشت که هزینه کم، مصرف انرژی کم، قابلیت جمع آوری داده، پردازش داده و ارسال داده به نود مرکزی از وظایف و خصوصیات این نودها می باشد. در بسیاری از کاربردهایی که شبکه های بی سیم دارند مشخص شدن مکان نودها مهم است، چونکه اینکه ما بدانیم که اطلاعات از چه نقطه ای به دست ما رسیده است می تواند خیلی مهم باشد مثلاً به طور واضح در کاربردهای مانیتورینگ که ما می خواهیم مسیر حرکت و مکان یک شیء را دنبال کنیم باید نود قادر باشد اطلاعات مکانش را به ما مخابره کند، همچنین در پروتکل های مسیریابی یا پروتکل های جمع آوری داده ها نیز مکانیابی نودها حائز اهمیت است. روشهای مختلفی برای مشخص شدن مکان نودها تا کنون بررسی و پیشنهاد شده است که به دو دسته کلی زیر می توان تقسیم کرد:

۱- با استفاده از GPS (Absolute Positioning): که هر نود شامل یک دستگاه GPS باشد تا موقعیت جهانی آن مشخص باشد که هزینه بر است و در ضمن از لحاظ اندازه نودها هم مشکل ساز می شود و همچنین در بعضی مکانها امکانپذیر نیست.

۲- با استفاده از نودهای Anchor و با استفاده از یکسری محاسبات (Relative Positioning): معمولاً از طریق ارتباط نود با سه نود Anchor، مکان نود تخمین زده می شود، البته باید توجه داشت که نودها می توانند متحرک باشند.

در کل می توان الگوریتمهای مکانیابی را از یک منظر به دو دسته تقسیم کرد:

۱- الگوریتمهای استاتیک

۲- الگوریتمهای دینامیک

الگوریتمهای استاتیک تا کنون بیشتر گسترش یافته است و به دو دسته تقسیم شده اند:

۱- Distance-Base: مثل TOA, TDOA, AOA, RSSI

۲- Non Distance-Base: مثل DV-Hop, Convex Programming, MDs-MAP

اکثر الگوریتمهای مکانیابی Distance-Base از تکنیکهای اندازه گیری استفاده می کنند:

۱- اندازه گیری زاویه ورود (AOA)

۲- اندازه گیری فاصله: یکطرفه و دوطرفه، TDOA

۳- اندازه گیری توان سیگنال ورودی (RSS)

تحقیقات مربوط الگوریتمهای دینامیک در حال حاضر بیشتر مورد توجه است، اما تئوریهای موجود کامل نیستند. مکانیابی نودهای متحرک در ردیف الگوریتمهای دینامیک است، این الگوریتمها به اطلاعات Real-Time به دلیل جابجایی نودها نیازمندند

حال به توجه به مطالب بالا به بررسی چند روش مکانیابی و الگوریتمهای مربوطه در نودها می پردازیم.

## ۲- الگوریتم مکانیابی (SALMN(Self Adapting Localization for Mobile Nodes)<sup>[1]</sup>)

همانطور که گفته شد در نودهای متحرک، الگوریتمهای مکانیابی احتیاج به اطلاعات Real-Time دارند. برای بهبود دقت مکانیابی نودهای متحرک پیش بینی و تخمین از حرکت نودها را باید در الگوریتم مکانیابی لحاظ کرد

SALMN بصورت خود سازگار می تواند فرکانس مکان یابی را بر طبق سرعتهای حرکت متفاوت تنظیم کند. پس ما می توانیم یک وقفه به حد کافی کوتاه را برای ایجاد تغییر وضعیت حرکتی نودهای متحرک بیابیم. نود های متحرک تقریباً از حرکت خطی متحد الشکل در دو وقفه زمانی مکان یابی قبل و بعد متابعت می کنند. فرض کنید در زمان  $t(i)$ ، سرعت حرکت نودهای متحرک  $V_{t(i)}$  باشد و زاویه مستقیم  $\theta_{t(i)}$  و مکان  $(x_{t(i)}, y_{t(i)})$  باشد و  $\Delta T_{t(i)}$  فاصله تا مکان اجرایی بعدی  $t(i+1)$  باشد، مکان  $(x_{t(i+1)}, y_{t(i+1)})$  نود متحرک در  $t(i+1)$ ، با مدل حرکتی زیر می تواند پیش بینی شود

$$\begin{cases} x_{t(i+1)} = x_{t(i)} + V_{t(i)} \cdot \Delta T_{t(i)} \cdot \cos \theta_{t(i)} \\ y_{t(i+1)} = y_{t(i)} + V_{t(i)} \cdot \Delta T_{t(i)} \cdot \sin \theta_{t(i)} \end{cases} \quad (1)$$

فرض کنید در زمان  $t(i)$  زمان قبلی  $t(i-1)$  است و مکان  $(x_{t(i-1)}, y_{t(i-1)})$  است. ما می توانیم سرعت حرکت را تخمین بزنیم. فرمول زیر معادله  $V_{t(i)}$  است.

$$V_{t(i)} = \frac{\sqrt{(x_{t(i)} - x_{t(i-1)})^2 + (y_{t(i)} - y_{t(i-1)})^2}}{\Delta T_{t(i-1)}} \quad (2)$$

$$\theta_{t(i)} = \begin{cases} \arctan \frac{y_{t(i)} - y_{t(i-1)}}{x_{t(i)} - x_{t(i-1)}} & x_{t(i)} - x_{t(i-1)} \geq 0 \\ \pi + \arctan \frac{y_{t(i)} - y_{t(i-1)}}{x_{t(i)} - x_{t(i-1)}} & x_{t(i)} - x_{t(i-1)} < 0 \end{cases} \quad (3)$$

مکان یابی نود های متحرک: بر اساس الگوریتم مکانیابی کلاسیک اطلاعات سه نود beacon لازم است برای مکان یابی در دو بعد. در این الگوریتم ها زمانی که حداقل سه نود beacon در شعاع مخابراتی نود مجهول وجود داشته باشد با استفاده از روشهای RSSI یا TOA می شود فاصله  $d_1$  و  $d_2$  و  $d_3$  نود مجهول را تا سه نود beacon اندازه گیری کرد. بر طبق فرمول زیر:

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 = d_1^2 \end{cases} \quad (4)$$

$$\begin{cases} (x-x_2)^2 + (y-y_2)^2 = d_2^2 \end{cases} \quad (5)$$

$$\begin{cases} (x-x_3)^2 + (y-y_3)^2 = d_3^2 \end{cases} \quad (6)$$

بعد از انتقال اعداد معلوم به یک طرف و مجهولات به طرف دیگر معادله و جانشینی نتیجه گیری می کنیم که:

$$AX = B \quad (7)$$

$$A = \begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) \\ 2(x_1 - x_3) & 2(y_2 - y_3) \end{bmatrix} \quad (8)$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix} \quad (9)$$

$$B = \begin{bmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 - d_1^2 + d_2^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 - d_2^2 + d_3^2 \end{bmatrix} \quad (10)$$

که در نهایت معادله زیر بدست می آید:

$$X = (A^T A)^{-1} A^T B \quad (11)$$

هنگامیکه فقط دو نود beacon در شعاع مخابراتی نود مجهول وجود دارد معادله بالا از معادلات ۴ و ۵ ایجاد می شود . و چونکه تعداد معادلات از تعداد مجهولات کمتر است پس غیر قابل حل است. الگوریتمی که در این مقاله به آن اشاره شده محاسبات مدل حرکتی نودهای متحرک را عنوان می کند. ما می توانیم موقعیت جابجایی را در طول زمان استنباط کنیم. اگر محدودیت حرکت اضافه شود راه حلهای اضافه می تواند حذف شود و راه حل واحدی بدست آید. اگر  $(x_a, y_a)$  ,  $(x_b, y_b)$  معادله باینری درجه دو بالا مشتق شوند، تخمین موقعیت نود در این زمان از  $(\hat{x}_{t(i)}, \hat{y}_{t(i)})$  است. فاصله  $d_a$  و  $d_b$  از مکان تخمین زده شده از پاسخ معادله محاسبه می شوند. پاسخ نزدیکتر به مکان تخمین زده شده صحیح است. زمانیکه فقط یک نود beacon در شعاع مخابراتی است الگوریتم مکان یابی کلاسیک جواب نمی دهد. نتیجه مکان یابی می تواند از محیط دایره ای که نود beacon (در موقعیت جاری) بعنوان مرکز آن دایره و شعاع آن از فاصله اندازه گیری شده با استفاده از RSSI یا TOA بدست آمده محاسبه گردد. اما در الگوریتم مذکور در بالا می توان اولاً از معادله معادله حرکت (۱) و معادله دایره (۴) برای بدست آوردن پاسخ استفاده کنیم. ثانیاً ما می توانیم از روش مقایسه بین راه حل و مکان تخمین زده شده استفاده کنیم. سپس از روش مشابهی در الگوریتم دو نود beacon برای کنار گذاشتن یکی از پاسخها استفاده می کنیم.

## ۲-۱- تنظیمات خود سازگار برای حرکت نودها :

می توان موقعیت حال حاضر نود را  $(\hat{x}_{t(i)}, \hat{y}_{t(i)})$  ، بر طبق تخمین از آخرین زمانیکه از مدل حرکتی (که در قسمت قبلی به آن اشاره شده است) استفاده شده و مکان اندازه گیری شده یک نود بر طبق الگوریتم مکان یابی استنتاج نمود. اکنون می توان خطای بین ارزش تخمین زده شده و مکان اندازه گیری شده با فرمول زیر محاسبه نمود:

<sup>1</sup> Self-Adapting

$$E = \sqrt{(x_{t(i+1)} - \hat{x}_{t(i+1)})^2 + (y_{t(i+1)} - \hat{y}_{t(i+1)})^2} \quad (12)$$

آستانه<sup>۲</sup> خطای مکان یابی می تواند  $E_{th}$  فرض شود و حداکثر وقفه<sup>۳</sup> مکان یابی  $T_{max}$  و حداقل فاصله  $T_{min}$  فرض شود. ارزش آنها در در ارتباط است با حدود بالا و پایین سرعت هنگامیکه نودهای موبایل حرکت می کنند. پس وقفه بعدی مکان یابی عبارت است از :

$$\Delta T_{t(i)} = \begin{cases} \omega_1 \cdot \Delta T_{t(i-1)} + \omega_2 T_{min} & E > E_{th} \\ \omega_1 \cdot \Delta T_{t(i-1)} + \omega_2 T_{max} & E \leq E_{th} \end{cases} \quad (13)$$

$$(14)$$

در معادله بالا  $\omega_1$  و  $\omega_2$  ارزش سنجیده شده تعریف می شوند که قابل تنظیم هستند. دامنه ارزش  $[0,1]$  است. هنگامیکه  $\omega_1$  صفر داده شود و  $\omega_2$  یک داده شود و هنگامیکه وقفه از خطا بیشتر باشد، وقفه مکان یابی به  $T_{min}$  برای تنظیم حد وقفه کاهش می یابد. ولی وقتی وقفه کمتر از خطا باشد، وقفه مکان یابی به  $T_{max}$  افزایش می یابد. تعیین ارزش سنجیده شده، وقفه مکان یابی را بسیار سریع تغییر می دهد. هنگامیکه خطا بیش از حد آستانه است، انرژی نود می تواند به خاطر مکان یابی متناوب کم شود. اما هنگامیکه خطا از حد آستانه کمتر است، وقفه مکان یابی  $T_{max}$  فرض می شود و پاسخ نود متحرک سریع نخواهد بود و مدل حرکتی سریعاً نمی تواند به روز شود. این نشان می دهد که  $\omega_1$  و  $\omega_2$  باید این گونه تعریف شوند:

$$\omega_1 = \omega_2 = 0.5 \quad (15)$$

---

<sup>2</sup> Threshold

<sup>3</sup> Interval

هدف، بالانس نگه داشتن همگرایی سرعت و حساسیت مدل حرکت است. و برنامه محاسباتی به شکل زیر می باشد:

```

for (i=1, t(1)=0,  $\omega_1 = \omega_2 = 0.5$ , initial of motion model,  $E_{th}$ ,  $T_{max}$ ,  $T_{min}$  are set,  $t(i)$  is less than simulation time
{
the current location is be estimated on the basis of motion model (1)
If three beacon nodes at least are existed within the range of communication, the nearest three nodes are selected and
triangular localization is executing, according to simultaneous equations (4), (5), (6)
Else if only two beacon nodes are existed within the range of communication, the solution but not unique is derived,
the node nearer to the estimated location should be selected, according to simultaneous equations (4), (5).
Else if only one beacon node is existed within the range of communication, the node nearer to the estimated location
should be selected, according to the solution of simultaneous equations (1), (4).
Else can not be located, the estimated location according to motion model is set the current location of mobile node.
End
End
End
The motion model is updated.  $V_{t(i)}$ ,  $\theta_{t(i)}$  are calculated by simultaneous equations (2), (3).
E between estimated location and measured location is calculated by equation (12)
If  $E \leq E_{th}$ 
 $\Delta T_{t(i)}$  can be calculated by equation (14).
Else
 $\Delta T_{t(i)}$  can be calculated by equation (13).
End
 $t(i+1) = t(i) + \Delta T_{t(i)}$ 
 $i++$ 
}

```

### ۳- مکانیابی با استفاده از روش TDOA [2]

اندازه گیری TDOA بصورت زیر محاسبه می شود:

۱ - فرستنده یک سیگنال  $s(t)$  منتقل می کند که به اندازه  $T_i$  برای دریافت کنند  $i$  مطابق فاصله اش تا هر دریافت کننده تأخیر می کند. این سیگنال می تواند یک راهنما باشد برای یک موبایل (up-link) ، جایی که زمان حرکت ناشناخته است، یا می تواند ناشناخته باشد، همانند وضعیتی که در جنگ الکترونیکی وجود دارد. در سایر وضعیت ها  $T_i$  قابل محاسبه نیست.

۲ - آنالیز همبستگی یک زمان تأخیر  $T_i - T_j$  متناظر اختلاف مسیر برای دریافت کننده های  $i$  و  $j$  به ما می دهد.

در این جا حالت استاتیک را مطالعه می کنیم در جایی که فقط یک سنسجش TDOA برای هر جفت از دریافت کنندگان داریم. ما مقایسه خواهیم کرد فیلتر پارتیکل را بر پایه تخمین زننده با الگوریتم LMS (least mean square) در چهارچوب کوچکترین مربعات غیر خطی.

#### ۳-۱- محاسبات TDOA

سیگنال های دریافت شده عبارتند از :

$$y_i(t) = a_i s(t - \tau_i) + e_i(t), \quad i = 1, 2, \dots, n \quad (1)$$

جایی که دریافت کننده  $i$  در  $x_i, y_i$  قرار دارد و فرستنده در  $x, y$  که مکانی ناشناخته است قرار دارد. با مرجع شناخته شده  $S(t)$  و همزمانی کامل، ما  $T_i(\text{TOA})$  را مستقیماً می‌توانیم تخمین بزنیم، و تخمین بزنیم  $(x, y)$  را با استفاده از چهارچوب مینیمم معادله درجه دو غیر خطی، مشابه با GPS. با مرجع ناشناخته، ساده‌ترین ایده مقایسه کردن سیگنال‌های دریافت شده دو به دو است. فرض کنید یک تابع مرتبط و همبسته بصورت دو به دو تخمین معادله شماره ۲ را محاسبه می‌کند

$$\Delta d_{i,j} = v(\tau_i - \tau_j), \quad 1 \leq i < j \leq n. \quad (2)$$

در جایی که  $v$  سرعت صدا، نور یا نوسان (ارتعاش) آب است. در این جا  $n$  تعداد دریافت کنندگان است و  $(i, j)$  شمارش تمامی  $k$  جفتی دریافت کنندگان، جایی که  $k = \binom{n}{2}$  می‌باشد.

هر  $\Delta d_{i,j}$  متناظر است با موقعیت‌های  $(x, y)$  در یک منحنی هایپربولیک.

در ابتدا فرض کنید که دریافت کننده‌ها هر دو در محور  $x$  در  $x = -D/2$  و  $x = D/2$ ، بترتیب قرار دارند. تابع هایپربولیک می‌تواند بصورت زیر بیان شود.

$$d_2 = \sqrt{y^2 + (x + D/2)^2}, \quad (4a)$$

$$d_1 = -\sqrt{y^2 + (x - D/2)^2}, \quad (4b)$$

$$\Delta d = d_2 - d_1 = h(x, y, D) \quad (4c)$$

$$= \sqrt{y^2 + (x + D/2)^2} - \sqrt{y^2 + (x - D/2)^2}. \quad (4d)$$

بعزاز مختصر سازی‌ها این معادله در یک حالت فشرده‌تر بصورت زیر در می‌آید:

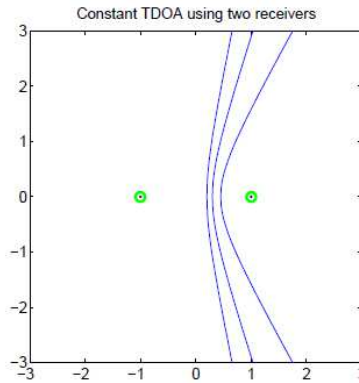
$$\frac{x^2}{a} - \frac{y^2}{b} = \frac{x^2}{\Delta d^2/4} - \frac{y^2}{D^2/4 - \Delta d^2/4} = 1. \quad (5)$$

راه حل برای این معادله خطوط مجانبی دارد در مسیر خطوط با معادله:

$$y = \pm \frac{b}{a} x = \pm \sqrt{\frac{D^2/4 - \Delta d^2/4}{\Delta d^2/4}} x \quad (6)$$

که بعنوان زاویه ورود برای فرستندگان از راه خیلی دور است تعیین می‌شود، شکل (۱) یک تابع هایپربولیک در یک سیستم محلی مختص  $(x, y)$  را نشان می‌دهد.





شکل ۱- تابع هایپربولیک (نشان دهنده ثابت TDOA)

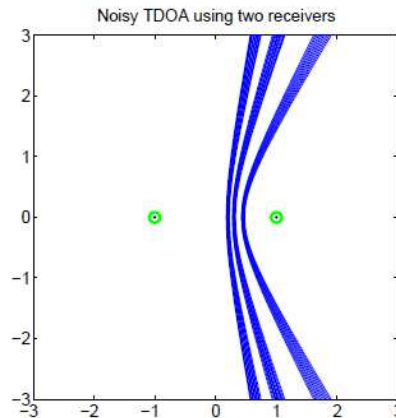
برای یک موقعیت عمومی دریافت کننده، ما به سادگی این تابع هایپربولیک (تابع شماره ۵) را در مختصات محلی  $(x, y)$  به مختصات گلوبال  $(X, Y)$  با استفاده از این معادله (۷)  $\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix} + \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$  برمی گردانیم. جایی که  $Y_0 = \frac{(Y_i Y_j)}{2}$ ,  $X_0 = (x_i + x_j)/2$  نقطه مرکزی جفت دریافت کننده قرار می گیرد. بنابراین تابع هایپربولیک در همبستگی گلوبال با  $\Delta d_{i,j} = h(x, y, D)$  در معادله شماره ۵ داده شده است.

$$D = \sqrt{(Y_i - Y_j)^2 + (X_i - X_j)^2} \quad (8a)$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y - Y_0 \end{pmatrix} \quad (8b)$$

$$\alpha = \arctan\left(\frac{Y_i - Y_j}{X_i - X_j}\right) \quad (8c)$$

با معادلات بالا ما اکنون یک فرم تابعی مناسب برای اندازه گیری در TDOA داریم که یک فضای هایپربولیک را نشان می دهد، که در شکل زیر نمایش داده شده است.



شکل ۲- فضای هایپربولیک

### ۳-۲- مشکل حل معادله کوچکترین مربعات غیرخطی

مشکل اصلی در حل کردن سیستم غیرخطی از K معادله است:

$$\Delta d_{i,j} = h(X, Y; X_i, Y_i, X_j, Y_j), \quad 1 \leq i < j \leq n \quad (9)$$

برای وضعیت فرستنده (X, Y)، موقعیت‌های دریافت کننده داده شده (X<sub>i</sub>, Y<sub>i</sub>) وجود دارد. حالا، تخمین None-linear least squares (X, Y) بوسیله  $(\hat{X}, \hat{Y}) = \arg \min_{(x, y)} \sum_{i>j} (\Delta d_{i,j} - h(x, y, x_i, y_i, x_j, y_j))^2$  داده می‌شود.

برای ساده کردن علائم، از P=(X, Y) استفاده می‌کنیم و پس از جانشینی P=(X, Y) مشکل حداقل سازی<sup>۴</sup> در vector را به فرم زیر بازنویسی می‌کنیم:

$$\hat{P} = \arg \min_P (\Delta d - h(P))^T R^{-1} (\Delta d - h(P)) \quad (10)$$

جایی که  $R = Cov(\Delta d), \Delta d = (\Delta d_{1,2} \dots \Delta d_{n-1,n})^T$  کواریانس ماتریکس برای اندازه‌گیریهای TDOA است. راه حل تخمین حداقل واریانس را تعریف می‌کند. برای فرض گوسی<sup>۵</sup> بر روی نویز TDOA این همزمانی با احتمال ماکزیمم تخمین زده می‌شود.

استفاده از این فرض که  $\Delta d = h(P_0) + e$  در جایی که  $P_0$  یک وضعیت صحیح است و نویز TDOA،  $Cov(e) = R$  را دارد، اولین دسته توسعه Taylor در اطراف ارزش صحیح  $h(p) \approx h(P_0) + h'_p(P_0)(P - P_0)$  را می‌دهد. تئوری least squares هم اکنون به ما معادله شماره ۱۱ را می‌دهد:

$$Cov(\hat{P}) = (h'_p(P_0))^{\dagger} R ((h'_p(P_0))^{\dagger})^T \quad (11)$$

که این امر را میسر می‌کند که  $\hat{p}$  بصورت کافی به موقعیت صحیح (high SNR) نزدیک باشد. برای تابع گوسی نویز e، این بیان همچنین باند پایین تر Cramer-Rao را تعریف می‌کند. یعنی هیچ تخمین زنده‌ای نمی‌تواند بهتر از این باند عمل کند، و این نتیجه را به ما می‌دهد که ما به قدر کافی به جواب صحیح نزدیک شده ایم. از فرمول ۱۱ ما می‌توانیم کمک بگیریم برای این که  $P_0$  مورد نظر کدام است، یا برای جنگ الکترونیک چگونه باید دریافت کنندگان را در بهترین مسیر ممکن قرارداد کمک بگیریم.

### ۳-۳- الگوریتم‌ها

مقایسه سه روش متفاوت:

۱- نقطه اشتراک هر جفت از توابع هایپربولیک را محاسبه کنید. برای هر جفت از توابع هایپربولیک معادله زیر وجود دارد

<sup>4</sup> Minimization

<sup>5</sup> Gaussian

$$\binom{K}{2} = \binom{n}{2} \quad (12)$$

تخمین موقعیت می‌تواند میانگین وزنی این نقاط باشد. از آنجایی که هر جفت هیچ، یک یا دو اشتراک می‌تواند داشته باشد، منطقی که برای پیدا کردن تقاطع صحیح است بدیهی نیست، علاوه بر این پیدا کردن وزن صحیح مقداری مشکل است.

۲- بکار بردن الگوریتم گرادیانت اتفاقی برای مشکل Non-linear least squares .

۳- برای بدست آوردن حدود عددی مشکل مینیمم معادله درجه دو غیر خطی از تکنیکهای مبتنی بر Mont corlo ، استفاده می‌شود. این به عنوان پارسیکل فیلتر (PF) محسوب می‌شود که یک نسخه استاتیک از روشی عمومی است .

اولین مشی نشان داده شده است که نتایج با کیفیت پایین (فرعی و درجه ۲) نسبت به دو روش دیگر به ما می‌دهد. دو روش دیگر در زیر توصیف شده‌اند، الگوریتم گرادیانت نرمال شده به ترتیب زیر می‌تواند نوشته شود:

الگوریتم یک (الگوریتم stochastic gradient):

$$P^{(m+1)} = P^{(m)} - \mu^{(m)} h'_P(P^{(m)}) (\Delta d - h(P^{(m)})) \quad (13)$$

یک سایز بهینه می‌تواند محاسبه شود بوسیله تکنیک‌های ۲ بخشی، line search یا مانند آنچه در شبیه‌سازی است.

$$\mu^{(m)} = \frac{\mu}{(h'_P(P^{(m)}))^T h'_P(P^{(m)})} \quad (14)$$

پارسیکل فیلتر (PF) نسخه استاتیک شناخته شده الگوریتم SIR است.

الگوریتم دو (PF استاتیک):

۱- بصورت اتفاقی در آورید پارسیکهای N را (در این جا موقعیت‌های ممکن) Pi .

۲- ثابت‌های بی‌ثبات CR و CQ را انتخاب کنید و اجازه دهید به موقعیت کواریانس راندوم  $\bar{Q} = C_Q/k^2$  و نویز سنجش بی‌ثبات  $\bar{R} = R + C_R/k^2$  .

۳- برای  $\hat{P}^{(k)}$  تا  $k = 1, 2, \dots$  تکرار کنید تا تقارب خطوط پیش آید و همگرایی دیده شود.

a) محاسبه کنید وزن Particle را برای  $w^i$  با استفاده از احتمال:

$$w^i = w^i / (\sum w^i) \text{ و } w^i = \exp((\Delta d - h(P^i))^T R^{-1} (\Delta d - h(P^i)))$$

$$\hat{P}(k) = \sum_i w^i p^i \quad \text{محاسبه کنید تخمین (b)}$$

(c) با جایگزینی پارتيكلها دوباره نمونه بگيريد ، در جايي كه احتمال بر گزیدن يك پارتيكل با وزنش متناسب است. بعد از نمونه‌گیری مجدد، وزنها در رابطه  $w^i = 1/N$  دوباره محاسبه می شوند.

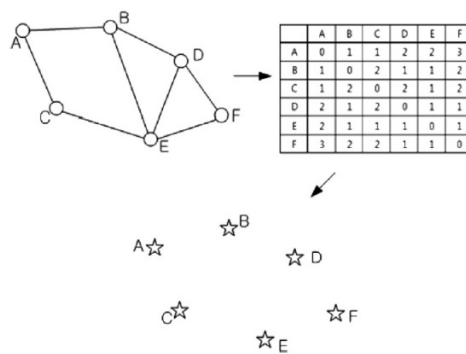
(d) پخش کنید پارتيكلها را بعنوان  $P^i = P^i + w$  در جايي كه  $w \in N(O, Q)$  است.

در پارتيكل فیلتر (PF) استاندارد، K زمان را مشخص می‌کند و یک مرحله به روز کردن زمان در جايي كه پارتيكلها حرکت داده می‌شوند براساس اندازه‌گیری سرعت و یک نويز حرکتی  $w$  ، وجود دارد ، در غير اينصورت الگوريتم ها كاملاً مشابه هستند.

#### ۴- مکانیابی با استفاده از $cluster-based MDS^{[3]}$

RSSI اصلی ترین روش در بین تمهیدات مکان یابی است. این روش، فاصله بین دو سنسور نود را با استفاده کردن از خصوصیت توان سیگنال دریافت شده که با توجه به فاصله تغییر می یابد، اندازه می‌گیرد. اندازه‌گیری فاصله با استفاده از RSSI قابلیت عملی بودن بالایی را دارد. بهر حال الگوریتم قدرت سیگنال متاثر از سنسور نودها و مشخصات محیط اطراف و نويز کانال است. این عیب و چالش بصورت مستقیم خطای مکان یابی را منعکس می‌کند. هدف متریک MDS یافتن پیکربندی نقاط در فضای چند بعدی است. اگر اطلاعات مجاور بدون خطا اندازه‌گیری می‌شد در فضای اقلیدسی، متریک کلاسیک MDS بطور دقیق پیکربندی نقاط را به نمایش می‌گذاشت. MDS کلاسیک MDS-MAP خوانده می‌شود و حاوی مراحل زیر است. اول: محاسبه کوتاهترین مسیرها بین تمام جفت‌های نودها در ناحیه مورد نظر، کوتاهترین مسیرها استفاده می‌شوند برای ایجاد کردن و ساختن ماتریکس فاصله برای MDS، هنگامیکه فاصله یا جفت نودهای همسایه شناخته شده باشد، ارزش لبه<sup>۶</sup> متناظر، فاصله اندازه‌گیری شده است. هنگامیکه ما فقط اطلاعات اتصالی را داریم، تخمین ساده‌ای لازم است برای تعیین کردن ارزش‌ها برای تمام لبه‌ها.

سپس یک الگوریتم کلاسیک برای تمامی جفت‌های با کوتاهترین راه، از جمله الگوریتم Dijkstra's ، می‌تواند بکار برده شود. نتیجه MDS یک نقشه مربوط است که مکان هر نود را می‌دهد. شکل (۳) مراحل بالا را به تصویر کشیده است.



شکل ۳- مراحل تخمین کووردینیت با استفاده از MDS-MAP

<sup>6</sup> Pattern

<sup>7</sup> Edge

با وجودیکه صحت MDS-MAP بالا است، MDS-MAP هنوز با مشکل زیر روبروست. در ابتدا، پیچیدگی زمان بالا است. پهنای باند وسیع و برای تخمین زدن مکان، هنگامیکه تعداد زیادی سنسور وجود دارد محاسبه وسیع لازم است. ثانیاً کووردینیت محاسبه شده برای فضاهای تحت پوشش شبکه با یک شکل non-convex در معماری شبکه ای سلسله مراتبی خطاهای زیادی دارد، یک سنسور نود ممکن است نقش‌هایی cluster head، cluster member یا نود gateway را بازی کند. در HMDS، Cluster head اندازه فاصله را از تمام اعضای cluster جمع‌آوری خواهد کرد و محاسبه MDS محلی را برای تشکیل دادن یک نقشه محلی اجرا می‌نماید. نود gateway مشارکت دارد در حداقل دو cluster مجاور در تغییر دادن اطلاعات بین ۲ کلاستر. هنگامیکه نقشه محلی هر کلاستر بوسیله هر cluster head، محاسبه می‌شود، کلاستری که حاوی عضو است که دارای ID خاصی است نیاز دارد به شروع کردن و راه انداختن فاز ادغامی. فاز ادغامی کالیبره می‌کند کووردینیت های اعضا را در کلاسترهای مختلف. این کلاسترهای ادغام شده بصورت تکرار شونده درخواست ادغام را برای کلاسترهای مجاور پیش‌بینی خواهند کرد و کووردینیت های کلاسترهای مجاور را تا مرتبط شدن تمام کلاسترهای نامتناقض کالیبره می‌کنند و یک نقشه گلوبال محاسبه شود. بهر حال، HDMS دارای مشکل فیکس کردن شماره کلاستر است. اگر خطا وجود داشته باشد در فیلد دریافت، HDMS نتایج نادرستی را می‌دهد.

#### MDS بر پایه کلاستر (cluster – based MDS) برای مکان یابی در یک محیط وسیع:

این تحقیق پیشنهاد می‌کند یک تکنیک مکان یابی را در شبکه های سنسور بی‌سیم در جایی که نودهای سنسور بصورت راندم در یک محیط وسیع پراکنده می‌شوند که به مانیتورینگ نیاز دارد. این مقاله دارای فرضهای زیر است:

۱- هیچ سنسور نودی متحرک نیست.

۲- هر سنسور نودی ID منحصر بفرد، شعاع حسی مشابه و شعاع رادیویی مشابهی را دارد.

۳- حداقل یک راه مسیریابی بین هر جفت سنسور نودها در شبکه وجود دارد، یعنی همه سنسور نودها بهم وصلند.

۴- هر سنسور نودی قابلیت اندازه‌گیری فواصل را با استفاده از RSSI دارد.

تکنیک پیشنهاد شده، CMDS، حاوی کلاسترینگ  $k$  پرشه برای مکان یابی است، MDS در هر کلاستر است، سیستم ادغام کردن کووردینیت کلاستر است. در زیر بطور خلاصه آمده است که چگونه کلاسترها ایجاد می‌شوند و چگونه کووردینیت نسبی تمام نودهای سنسور بوسیله یکی کردن سیستم های کووردینیت نسبی تمام کلاسترها بدست می‌آید.

#### ۴-۱- کلاسترینگ $k$ پرشه برای مکان یابی

این قسمت نشان می‌دهد که چگونه کلاسترهای  $k$  پرشه برای مکان یابی ایجاد می‌شوند. کلاسترها در ۳ مرحله ایجاد می‌شوند:

اول) نودهای cluster head انتخاب می‌شوند.

دوم) نودهای cluster head انتخاب شده کلاسترها را تشکیل می‌دهند.

سوم) اتصال درونی کلاسترها بوسیله به اشتراک گذاری نودهای درون کلاستری (Inter – cluster) که استفاده می‌شوند برای ادغام سیستم های همبسته و مرتبط تمام کلاسترها ایجاد می‌شود .

انتخاب cluster head:

تمام سنسورها بصورت راندوم در یک فیلد وسیع سنسور قراردادده می‌شوند و پیغام Hello را منتشر می‌کنند. سپس هر نود درجه نود خود ، ID تمام نودهای مجاور و فاصله‌های این نودهای مجاور را پیدا می‌کند. یک نود قراردادی (i) که کمترین ID را در بین نودهای مجاورش دارد، نودآغازگر می‌شود برای مرحله ایجاد کلاستر و زمان  $t_1(i)$  را شروع می‌کند . آغازگر، به محض خاتمه، یک نود cluster head می‌شود. بهرحال، اگر آغازگر یک پیغام اعلام cluster head را از سایر نودها، قبل از به پایان رسیدن  $t_1(i)$  دریافت کند، نود تایمر را متوقف می‌کند و یکی از اعضای کلاستر می‌شود. این کار جلوگیری می‌کند از اعلان و شناسایی شدن مکرر نودهای cluster head و فرمول (۱) یک تایمر  $t_1(i)$  است برای شروع کننده:

$$t_1(i) = \alpha_{deg} \max(0, 1 - \frac{deg(i)}{\theta_{deg}(i)}) + t_{BCD} \quad (1)$$

جایی که  $t_{BCD}$  بعنوان یک تأخیر انتشاری راندوم قراردادی، تعیین شود،  $\{i\}$  بعنوان یک درجه اتصال نود  $i$  علامت‌گذاری و تفکیک می‌شود و  $\theta_{deg}$  بزرگترین درجه نودهای مجاور است.  $\propto deg$  یک فاکتور وزنی برای درجه است.

تشکیل کلاستر موقت:

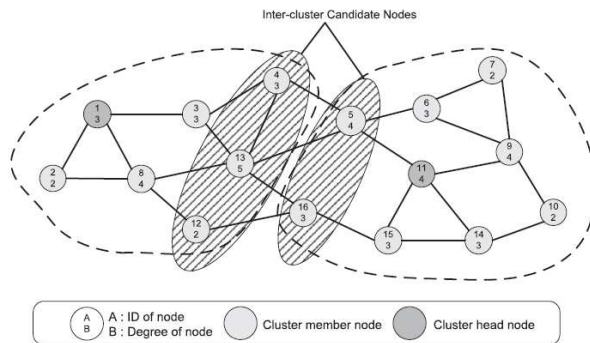
نود cluster head ، پیغامهای شناسایی نود cluster head را به نودهای  $2k$  پرشه می‌فرستد ، این پیغام ۲ معنی دارد:

۱ – هر نود  $k$  پرشه از نودهای cluster head ، یک عضو از کلاستر می‌شود.

۲ – هر نودی که بین  $(k+1)$  پرش و  $2k$  پرش از نود cluster head قرار می‌گیرد یک کاندیدا می‌شود برای یک نود cluster head جدید، این کاندیداها برای یک نود cluster head جدید، تایمر (i)  $t_2$  را شروع می‌کنند.

$$t_2(i) = \alpha_{deg} \max\left(0, 1 - \frac{deg(i)}{\theta_{deg}(i)}\right) + \alpha_{dist} \left(1 - \frac{chdist(i)}{2k}\right) + t_{BCD} \quad (2)$$

جایی که  $chdist(i)$  یک شماره از پرشها بین نود  $i$  و نود  $cluster\ head$  ( $k + 1 \leq chdist(i), chdist \leq 2k$ ) ، و  $\propto dist$  باشد، یک فاکتور مهم را برای مسافت مشخص می‌کند. یک نود با یک تایمر منقضی شده  $t_2i$  یک نود cluster head جدید می‌شود و پیغامهای اعلام cluster head را ، در حالت کامل کننده مراحل ایجاد کلاستر می‌فرستد. اگر یک نود یک پیغام اعلام را از سایر نودها دریافت کند قبل از این که تایمر به آخر برسد و مردود شود، این نود یک عضو از کلاستر دیگری می‌شود. اگر یک نود به یک کلاستر در طول مراحل ایجاد کلاستر قبلاً متعلق بوده است ، این نود هیچ پیغام اعلان نود cluster head جدید را که دریافت می‌کند ارسال نمی‌کند . عملکرد کامل کلاسترینگ ممکن است منجر شود به انتشار تغییر ناگهانی ناشی از یک تعداد زیادی از کاندیداهای cluster head . استفاده کردن از ارزشهای تأخیری راندوم  $t_{BCD}$  از این مسئله جلوگیری می‌کند .

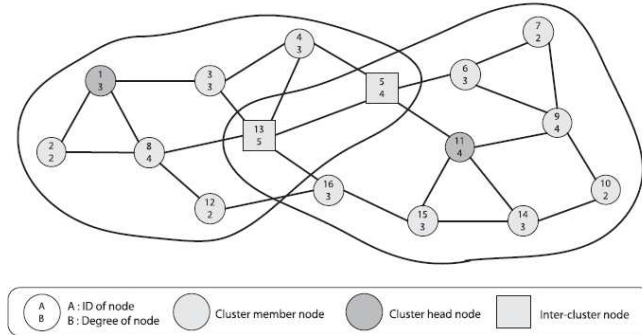


شکل ۴- انتخاب یک cluster head و ایجاد یک کلاستر موقت

در ابتدا هر نود منتشر می‌کند پیغامهای Hello را و درجه‌اش و اطلاعات نود مجاور را دریافت می‌دارد. با داشتن یک ID پایین‌تر نسبت به هر کدام از نودهای مجاور، نود ۱ بعنوان یک شروع کننده به کار می‌رود و تایمر را در  $t_1(i)$  ست می‌کند. اگر تایمر به آخر برسد، نود ۱، یک نود cluster head می‌شود و پیغامها را منتشر می‌کند با مقدار TTL برابر 2K، که ۴ است. نودها داخل ۴ پرش از نود ۱ این پیام را به نودهای مجاورشان با میزان کاهش ارزش TTL تا یک ارسال میکنند. در شکل (۴) نودهای داخل ۴ پرش از نود یک عبارتند از ۲، ۳، ۴، ۵، ۶، ۸، ۱۱، ۱۳، ۱۵ و ۱۶. هر نودی دریافت کننده یک پیام با TTL برابر یک، بعداً پیام را ارسال نمی‌کند، هر نود دریافت کننده یک پیام با TTL برابر ۳ یا بالاتر یک عضو از کلاستر نود یک می‌شود. در جهت معکوس، داشتن پیغامهای اعلان نود cluster head دریافت شده با TTL کمتر از ۳، نودهای ۶، ۵، ۱۵، ۱۱ و ۱۶ کاندیدهای نود cluster head می‌شوند و تایمر را در  $t_2(i)$  ( $i \in \{5,6,11,15,16\}$ ) ست می‌کنند. نودهای ۷، ۹، ۱۰ و ۱۴ از لیست کاندیدها مستثنی می‌شوند از آنجایی که آنها پیامهای اعلان نود cluster head را دریافت نمی‌کنند. در این وضعیت، نود ۶، ۱۱، ۱۵ از نود یک در دورترین محل قرار دارند و درجات نود ۵ و ۱۱ بالاترین است، نتیجتاً در  $t_2(11)$  تایمر در پایین‌ترین ارزش ست می‌شود. هنگامیکه تایمرش به پایان رسید، نود ۱۱، یک نود cluster head جدید می‌شود و پیامهای اعلان نود cluster head را با تکرار این مرحله منتشر می‌کند. یک کلاستر موقت ممکن است مطابق شکل (۴) ایجاد شود.

#### ۴-۲- ساختن اتصال inter – cluster با استفاده از نودهای inter – cluster

بعضی از نودهای مرزی در کلاستر موقت یک پیغام اعلان نود cluster head را دریافت می‌کنند با یک TTL برابر k یا بالاتر از cluster head مجاور. این نودها، نودهای کاندیدی inter – cluster خوانده می‌شوند که تعداد فرد یا چند تایی ازشان ممکن است وجود داشته باشد. این نودها بعداً در مرحله ادغام سیستم کووردینیت میان کلاسترها استفاده می‌شوند. نود cluster head بصورت رانوم یکی از این نودهای کاندید شده Inter-cluster را انتخاب می‌کند و بعداً اطلاعات را برطبق یک نود انتخاب شده inter-cluster به یک head node از کلاستر مجاور تحویل می‌دهد. اطلاعات نود inter-cluster منتقل شده از cluster head به کلاستر مجاور به نود cluster head برای به حساب آوردن نود inter-cluster برای دوباره کانفیگوریشن کردن کلاستر موقت نیاز دارد. با استفاده از این مرحله، هر کلاستری سهمی می‌کند دو تا نود روی هم افتاده inter-cluster را با هر کدام از کلاسترهای مجاورش. این مرحله ۲ نود روی هم افتاده inter-cluster را نتیجه می‌دهد و منظور این است که سیستمهای کووردینیت نسبی را بین دو کلاستر ادغام کند. در شکل (۵)، نود ۴، ۱۲ و ۱۳ نودهای مرزی کلاستر ۱ هستند و TTL پیغام اعلان نود cluster head از نود ۱۱ که head node کلاستر مجاور است، دریافت می‌شود.



شکل ۵- ایجاد کلاسترهای k پرشه با استفاده از نودهای inter-cluster

از آنجایی که نودهای ۵ و ۱۶ نودهای مرزی کلاستر ۱۱ هستند و TTL پیغام اعلان کلاستر از head node یک دریافت میشود، نود ۴، ۱۲ و ۱۳ نودهای کاندیدای inter-cluster کلاسترهای ۱ هستند و نود ۵ و ۱۶ کاندیدای inter-cluster نودهای کلاستر ۱۱ بترتیب هستند. هر نود cluster head بصورت رانوم انتخاب می‌کند یکی از نودهای کاندید شده inter-cluster را بعنوان یک نود inter-cluster برای کلاستر مجاورش. در شکل ۵، کلاسترهای ۱ و ۱۱ انتخاب می‌کنند نودهای ۱۳ و ۵ را بترتیب بعنوان نودهای inter-cluster برای کلاستر مجاور. بعد از آن، کلاستر ۱ دوباره پیکربندی می‌شود با نود ۵، نود inter-cluster برای کلاستر مجاور، در حالی که کلاستر ۱۱ با نود ۱۳ دوباره پیکربندی می‌شود. شکل (۵) نشان می‌دهد که تمام کلاسترهای inter-cluster را با کلاسترهای مجاورشان و ساختمان درون اتصالی بین ۲ کلاستر سهیم می‌کنند.

### MDS-۴-۳ در هر کلاستر

MDS pseudocode in each cluster.

```

Input: role = {ClusterHead, Inter-cluster node, Member} //The role of each node
Output: Array MemberCoordinate[]
1   if role of node is Inter-cluster node or Member
2     then Estimate distance to all neighboring nodes
3     for i ← 1 to n // n is the number of neighbor nodes
4       do NeighborDist[i][0] ← ID of Neighbors
5         NeighborDist[i][1] ← Estimated distance
6     Send NeighborDist and member_id to ClusterHead
7   if role of node is ClusterHead
8     then m ← Number of Cluster Members
9     for i ← 1 to m
10      for j ← 1 to m
11        do Matrix[i][j] ← ∞
12      for i ← 1 to m
13        do Matrix[member_id][NeighborDist[i][0]] ← NeighborDist[i][1]
14      ShortestPath(Matrix)
15      MDS(Matrix)
16      return MemberCoordinate[]

```

جدول ۱- یک شبه برنامه برای بدست آوردن کووردینیت های نسبی در داخل یک کلاستر



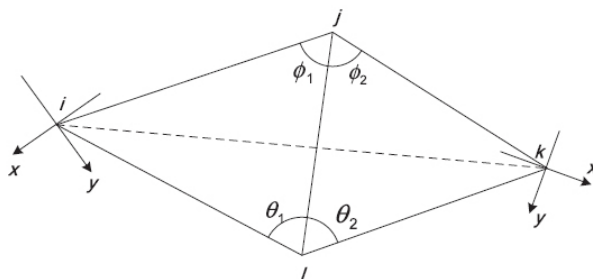
بعد از این که کلاسترینگ  $k$  پر شده برای مرحله مکان یابی کامل شد. هر نود  $cluster\ head$  ساکن یک ماتریکس فاصله می‌شود که از اطلاعات فاصله‌ای بر پایه  $RSSI$  و  $ID$  نودهای مجاور که دریافت شده‌اند از اعضای کلاسترشان استفاده می‌کند. فاصله بین هر جفت از اعضای کلاستر به کمترین فاصله با استفاده از الگوریتم کمترین مسیر بعد از این که نود  $cluster\ head$  ماتریکس فاصله را می‌سازد تغییر داده می‌شود. این اطلاعات استفاده می‌شوند برای اجرا کردن الگوریتم  $MDS$  برای بدست آوردن کووردینیت های نسبی در داخل کلاستر. این مرحله در تمام کلاسترها در یک حالت توزیع شده، تعیین کننده کووردینیت های نسبی تمام کلاسترها اجرا می‌شود. جدول ۱ یک شبه برنامه برای بدست آوردن کووردینیت های نسبی در داخل یک کلاستر استفاده کننده از الگوریتم  $MDS$  را نشان می‌دهد. اگر یک نقش از نود بعنوان نود  $Inter-cluster$  باشد، یکی از اعضایش، نود  $k$  را از خط ۱ تا خط ۶ اجرا می‌کند.  $n$  شماره نودهای مجاور است. نود  $ID$  نود مجاور  $(i)$  را به درون  $Neibor\ Dist[i][0]$  می‌گذارد و ارزش فاصله‌ای تخمین زده شده نود مجاور  $i$  را به  $NeighboDist[i][1]$  می‌گذارد. نود،  $NeighboDist$  و  $member - id$  را به نود  $cluster\ head$  مربوط به خودش انتقال می‌دهد. هر نود  $cluster\ head$  یک ماتریکس فاصله‌ای را برای اعضای کلاسترش با استفاده از  $k$ ، از خط ۷ تا خط ۱۴ از جدول ۱ ایجاد می‌کند. در خط ۱۴، کوتاهترین فاصله بین هر جفت از سنسور نودها بدست می‌آید با استفاده از الگوریتم کوتاهترین راه. سپس در خط ۱۵ الگوریتم  $MDS$  اجرا می‌شود براساس ماتریکس کوتاهترین فاصله. این مرحله در کووردینیت های نسبی اعضای کلاستر نتیجه می‌دهد. کلاستر، بهرحال سیستم کووردینیت نسبی مشابهی را، از آنجا که هر کلاستر بصورت منحصر بفردی کار می‌کند، سهم نمی‌کند. در بخش بعد، چگونگی بدست آمدن سیستم کووردینیت مشابهی برای تمامی کلاسترها را توصیف می‌کنیم.

ادغام سیستم های کووردینیت کلاستر:

این بخش توصیف می‌کند یک تکنیک را برای بدست آوردن یک سیستم همبستگی نسبی منفرد با ادغام کردن سیستم های کووردینیت نسبی از میان کلاسترها. هر نود  $Inter-cluster$  متعلق است به حداقل ۲ کلاستر و تعیین می‌شود بعنوان آدرس های نسبی (وابسته)  $head\ node$  تمامی کلاسترها برای آنهايي که متعلق به آن هستند.

اینکار اجازه می‌دهد که نود  $Inter-cluster$  یک آدرس وابسته برای هر کلاستر داشته باشد که مال خودش است و این آدرس استفاده می‌شود برای ادغام کردن سیستم های کووردینیت کلاسترهای مجاور.

شکل (۶) قسمتی از مرحله ادغام را توصیف می‌کند:



شکل ۶- تخمین  $d_{ik}$  با ساخت مثلث

- تخمین بزنیید فاصله بین ۲ نود cluster head و ادغام کنید سیستم های همبستگی نود cluster head را .
- نودهای  $i$  ،  $j$  ،  $k$  و  $l$  یک گوش تشکیل می دهند. نودهای  $I$  و  $k$  نود cluster head هستند و نودهای  $j$  و  $l$  نودهای inter-cluster مجاور هستند. این نودها  $\theta_1$  ،  $\theta_2$  ،  $\phi_1$  و  $\phi_2$  را می دهند با استفاده از  $\Delta(i,j)$  و  $\Delta(k,j,l)$  از طریق قانون دوّم کسینوسها.

$$\cos \theta_1 = \frac{d_{il}^2 + d_{jl}^2 - d_{ij}^2}{2d_{il}d_{jl}}, \quad \cos \theta_2 = \frac{d_{kl}^2 + d_{jl}^2 - d_{kj}^2}{2d_{kl}d_{jl}}$$

$$\cos \phi_1 = \frac{d_{ij}^2 + d_{jl}^2 - d_{il}^2}{2d_{ij}d_{jl}}, \quad \cos \phi_2 = \frac{d_{kj}^2 + d_{jl}^2 - d_{kl}^2}{2d_{kj}d_{jl}}$$

این حساب می کند  $\theta = \theta_1 + \theta_2$  را با استفاده از فرمول بالا و  $d_{ik}$  فاصله ای است که بین نود cluster head و  $k$  وجود دارد.

$$d_{ik}^2 = d_{il}^2 + d_{lk}^2 - 2d_{il} d_{lk} \cos \theta \quad (3)$$

Merge step pseudocode.

---

```

Input: Array MemberCoordinate[],
Boolean cluster confirm ← false
Output: Array merged MemberCoordinate
1  while entire network is not merged
2  do if role = ClusterHead and the lowest ID among its neighbor
   ClusterHeads
3  then send MPM to neighbor ClusterHeads
   // MPM = Merge Phase Message
4  cluster confirm ← true
5  else if role = ClusterHead and cluster confirm = false
6  then if received MPM
7  then Calculate the rotation angle
8  Calculate the coordination of all members
9  send MemberCoordinate to all Cluster Members
10 cluster confirm ← true
11 send MPM to neighbor ClusterHeads
12 return MemberCoordinate

```

---

جدول ۲- یک شبه برنامه برای مرحله ادغام

این مرحله برای هر نود inter-cluster اجرا می شود. نود  $k$  کووردینیت نود  $i$  را متناظر با سیستم کووردینیت با استفاده از ساخت مثلث ۸ محاسبه می کند. بعد، نود، سیستم کووردینیت را برای همزمان کردن سیستم کووردینیت نود  $k$  می چرخاند. نود  $i$  کووردینیت های جدیدش را به نودهای cluster member خود انتقال می دهد. مرحله ادغام، یکی کردن ۲ سیستم کووردینیت متفاوت را بین ۲ کلاستر، شروع کردن از یک کلاستر با پایین ترین ID نود cluster head بین کلاسترهای مجاور را در بر دارد. این وظیفه اجرا می شود از طریق یک وظیفه توزیع شده از نود cluster head. نود cluster head شروع می کند یک مرحله

<sup>8</sup> Triangulation

ادغامی را بوسیله فرستادن یک پیغام درخواست ادغام به head nod یک کلاستر مجاور. اگر موافقتی باشد با ادغام کووردینیت ها ، cluster head مجاور یک پیغام Ack به cluster head عقبی که یک درخواست ادغام فرستاده بود می فرستد و بنابراین مرحله ادغام شروع می شود. بعد از اینکه مرحله ادغام بین ۲ کلاستر کامل شد، نود cluster head با ID کمتر ، یک نود cluster head جدید می شود و سپس یک درخواست ادغام می فرستد به نود cluster head مجاور. سیستم های کووردینیت وابسته تمامی کلاسترها در یک کلاستر منفرد، با انجام کردن این مراحل بصورت تکراری ، ادغام می شوند . جدول ۲ حاوی شبه برنامه است برای مرحله ادغام. در خط ۱، حالت while استفاده می شود برای تکرار کردن مرحله مگر این که تمامی کلاسترها ادغام شوند. خطوط ۲-۴ شروع کننده مرحله ادغام هستند درجایی که هر نود cluster head با یک کلاستر مجاور منفرد فقط یک درخواست ادغام را با استفاده از پیغام فاز ادغام (MPM) برای نود cluster head مجاور انتقال می دهد. این کلاستر را به حالت صحیحی ست می کند و مختصات نسبی را به اعضای کلاستر انتقال می دهد. هر نود cluster head که دریافت کننده یک MPM است، سیستم های کووردینیت را برای هم تراز کردن آنها بین دو کلاستر روتیت می کند. بعد از ادغام وظیفه تثبیت کلاستر، نود cluster head که MPM را دریافت داشته است ، به طور صحیحی ست می شود. پس از آن یک MPM فرستاده می شود به یک نود cluster head در مجاورت کلاستر ادغام شده. تمامی کلاسترها یک سیستم هماهنگ نسبی ادغام شده با تکرار این مرحله دارند.

#### ۵- بررسی یک روش مکانیابی تئوری [4]

##### ۵-۱- محدودیت فاصله ای بر مبنای تحلیل مکان یابی

نود خود مکان یاب به صورت یک ضرورت برای کاربردهای واقعی در WSN ها الزام آور شده است. اگرچه اغلب الگوریتمهای مکانیابی توزیع یافته ارائه شده است، مهمترین تحلیل نظری از مکان یابی منحصر به فرد هنوز در مراحل اولیه خود از توسعه می باشد. هدف این قسمت، ترکیب کردن تحقیقاتی از مبنای نظری بر روی مشکل مکان یابی WSN می باشد. نشان داده شده است که مشکل مکان یابی در دو بعد به خوبی فهمیده می شود، در هر صورت، فقط نتایج مقیاسهای نسبی در سه بعد قابل دسترسی هستند.

راه حل مشکل برای WSN ها می تواند به صورت ذیل باشد. فرض کنیم یک چارچوب به وجود آمده است که طول لبه مربوط به فاصله inter-sensor محدودیتهای فاصله را تحقق بخشیده است. چارچوب ممکن است به صورت rigid باشد یا نباشد. اگر Rigid باشد، امکان دارد چارچوب دومین و متمایزی شکل بپذیرد. اگر بر حسب تجانس باشد یک چارچوب Rigid منحصر به فرد مطابق با محدودیتهای فاصله وجود دارد، یعنی، چارچوب به صورت جهانی Rigid می باشد، سپس WSN می تواند به عنوان یک موجودیت Rigid از ساختار شناخته شده مورد ملاحظه قرار گیرد و تنها نیاز به دانستن موقعیت اقلیدسی از چندین سنسور در این چارچوب به منظور مکان یابی کردن کل چارچوب در دو یا سه بعد است. اغلب تحقیقات در این موضوع با وجودیکه در مراحل اولیه اش است، بر مبنای متدولوژی محدودیت فاصله است.

تئوری چارچوبهای Rigid جهانی به صورت یک پیش زمینه آماری می باشند. در هر صورت اگر چارچوب ارائه شده به صورت جهانی Rigid باشد، برای تصمیم گیری مشکل وجود دارد. اما مشکل قابل حلتر می شود اگر ما به طور مناسبی بازدهی را به منظور اجتناب از همروی محدود کنیم. خصوصاً، یک روش قدرتمندی از محدود ساختن مشکل این است که فرض کنیم چارچوب به صورت عمومی می باشد، یعنی، مختصات نود های سنسور و نودهای anchor هیچ معادله جبری مهمی را با ضریبهای منطقی مطابقت نمی دهد.

ما می توانیم گراف  $G$  را با خصوصیتی که از  $G$  در فضای  $d$  به صورت سراسری Rigid است را خصوصیت ببخشیم، زمانیکه  $d=1,2$  می باشد. مشکلات خصوصیت بخشی زمانیکه یک چارچوب عمومی  $d$  بعدی Rigid می باشد یا به صورت سراسری Rigid می باشد برای  $d \geq 3$  غیر قابل حل می باشد.

با تعاریف قبلی، ما می توانیم یکی از نتایج اصلی را خلاصه کنیم.

قضیه: اجازه دهیم  $N$  به صورت شبکه ای باشد در  $R^d$  متشکل از  $m$  نود سنسور معمولی که در موقعیتهای  $p_1, p_2, \dots, p_m$  مکان یابی شده اند و  $n-m$  نود anchor در موقعیتهای  $p_{m+1}, \dots, p_n$  مکان یابی شده است. فرض کنیم که حداقل نودهای  $d+1$  در موقعیت عمومی وجود دارند. اجازه دهیم  $G$  به صورت نمودار معین شده از  $N$  باشد و اجازه دهیم  $p=(p_1, p_2, \dots, p_m)$  باشد. سپس شبکه ها به طور منحصر به فردی در یک نقطه جمع می شوند. اگر و تنها اگر  $(G, p)$  به صورت جهانی Rigid باشد.

## ۲-۵- شرایط برای Rigidity

قابلیت حل مشکل مکان یابی WSN اقتضا می کند که  $F_p$  Rigid نباشد و این امر غیر ممکن است، برای اینکه اگر  $F_p$  Rigid نباشد غیر ممکن است که بتوان  $F_p$  را تعیین کرد، تعیین  $F_p$  بر طبق معادل یابی نمی تواند امکان پذیر باشد و به تنهایی اجازه می دهد  $F_p$  به صورت متجانس و منحصر به فردی معین شود. چارچوب اگر دارای تغییر شکل های پیوسته بی اهمیتی نباشد Rigid می باشد. با مفهومی عمومی، ما به Rigidity به عنوان یک خصوصیتی از ارتباط توجه می کنیم اما به هندسه تشکیل توجه نمی کنیم. ما می توانیم از گراف Rigid و چارچوب Rigid بصورت مبادله ای بدون هیچ ابهامی استفاده می کنیم.

چندین لبه برای گرافی از نودهای  $n$  به طور عمومی به صورت Rigid وجود دارد؟ در دو بعد، نود های  $n$  دارای دومین درجه آزادی می باشند هر لبه که ما به گراف اضافه می کنیم می تواند در اغلب درجه یک آزادی حرکت کند. تبدیلها و چرخشهای جهانی همیشه به سوی احتمالات حرکت می کنند بنابراین حداقل  $2n-3$  لبه برای گراف به منظور Rigid لازم است.

در هر صورت، لبه های  $2n-3$  همیشه کافی نمی باشند، به دلیل اینکه تعدادی از لبه ها ممکن است اضافی باشند.

ما نیاز داریم به  $2n-3$  لبه های توزیع شده داریم، برای بیان این موضوع به طور رسمی، اگر زیر گراف دارای لبه های بیش از حد نیازی باشد، تعدادی از این لبه ها اضافی می باشند. لبه های ضروری یا غیر اضافی مستقل می باشند، درحالتی که ردیفهای متشابه آنها از ماتریکس Rigidity به صورت خطی مستقل می باشند. هر لبه مستقل همیشه درجه ای از آزادی را برطرف می سازد. بنابراین لبه های مستقل  $2n-3$  Rigidity را تضمین خواهند کرد.

تعریف ماتریکس Rigidity: اجازه دهیم  $(G, p)$  به صورت ادراک بعد  $d$  از یک نمودار  $G=(V, E)$  باشد. Rigidity ماتریکس از چارچوب  $(G, p)$  به صورت ماتریکس  $M(G, p)$  از اندازه  $|E| \times |V|$  می باشد جاییکه، برای هر لبه  $e=v_i v_j \in E$  می باشد، در ردیف متشابه به  $e$ ، ورودیهایی در دو ستون به رئوس  $i$  متشابه می باشند و  $j$  شامل مختصات  $d$  به ترتیب از  $(p(v_i)-p(v_j))$  و  $(p(v_j)-p(v_i))$  می باشد و به صورت صفرها باقی می ماند.

قضیه: اجازه دهیم  $G=(V, E)$  به صورت یک نمودار باشد. سپس  $G$  در  $R^d$  Rigid می باشد اگر و تنها اگر  $1 \leq |V| \leq d+1$  باشد و  $G$  کامل باشد، یا  $|V| \geq d+2$  و رتک  $M(G, p) = d|V| - (d+1)/2$ .

قضیه: یک نمودار  $G=(V,E)$  با  $|V|=n$  به طور کلی در  $R_2$  Rigid می باشد اگر و تنها اگر که دارای لبه های  $2n-3$  باشد و هیچ زیر گراف  $\hat{E}$  از رئوس  $k$  بیش از لبه های  $2k-3$  نداشته باشد.

تعریف cover: ارائه یک نمودار  $G=(V,E)$ ، یک پوششی از  $G$  به صورت  $X=\{X_1, X_2, \dots, X_t\}$  از زیر مجموعه هایی از  $V$  همانند  $|X| \geq 2$  برای تمامی  $X \in X$

### ۳-۵- الگوریتمهایی برای شناسایی Rigidity

ما مشاهده کردیم که شماری از خصوصیات نمودارهای Laman وجود دارد و اکثریت آنها به یک الگوریتم مرتبط برای شناسایی منجر می شوند. بعد از این، آنچه ما سرانجام مورد توجه قرار می دهیم موقعیت نود ها در WSN می باشد که می توانند در سرتاسر عملکرد الگوریتم یافت شوند.

قضیه: برای یک نمودار  $G=(V,E)$  with  $|V| \geq 2$  در پایین برابر هستند:

(1)  $G$  به طور کلی در  $R_2$  Rigid می باشد

(۲) یک سکوانس Henneberg-2 برای  $G$  وجود دارد

(۳) برای هر جفت از رئوس  $i, j$  در  $V$ ، مولتی گراف  $G^{ij} = (V, E \cup ij)$  با افزودن لبه ما بین رئوس  $i, j$  کسب می شوند که به صورت اتحاد لبه های منفصل از دو زیر گرافی از یک گراف می باشد:

(۴) برای هر لبه  $ij \in E$ ، چند نموداری  $G_{ij}$  توسط لبه دوبله  $ij$  به صورت اتحادیه ای از دو زیر گرافی از یک زیر گراف می باشد

(۵)  $G$  دارای یک قسمت بندی  $3T_2$  می باشد

(۶)  $G$  یک طبقه red-black را پذیرفته است (RBH)

شرایطی برای Rigidity:

اغلب نتایج شناخته شده در نمودارهای Rigidity سراسری با حالت دو بعدی مورد توجه قرار گرفته اند. تنها تعدادی از آنها می توانند به ابعاد بالاتر توسعه یابند، منجر به نتایج نسبی در مکان یابی منحصر به فردی از شبکه های سه بعدی شده است. ابزار اصلی کار با Rigidity به ابعاد بالاتر ماتریکس فشار است

تعریف. ماتریکس فشار: اجازه دهیم  $G=(V,E)$  یک نمودار باشد و  $(G,P)$  یک تحققی از  $G$  در  $R_d$  باشد. برای هر  $i \in V$ ، اجازه دهیم  $E(i)$  به صورت مجموعه ای از لبه های  $G$  باشند که به صورت رویدادی برای  $i$  می باشند. یک تنش پایدار برای  $(G,P)$  به صورت یک نقشه  $\omega R:E$  می باشد. ما یک  $|v|^*|v|$  متقارن یک ماتریکس فشار  $\Omega$  با هر فشار متعادل  $\omega$  برای  $(G,P)$  همانند ذیل را مرتبط کرده ایم. برای هر  $i, j \in V$ ،  $i, j$  مجزا ورودی به ردیف  $j$  و ستون  $i$  از  $\Omega$  به صورت  $-\omega(e)$  می باشد و به عبارت دیگر صفر می باشد. ورودیهای مورب از  $\Omega$  سپس انتخاب می شوند بنابراین ردیف و ستون آن با صفر برابر می باشند.

قضیه: اجازه دهیم  $G=(V,E)$  یک نمودار باشد. سپس  $G$  به صورت استحکام سراسری در  $Rd$  می باشد تنها و تنها اگر  $|v|\leq d+1$  باشد یا  $|v|\geq d+2$  باشد و یک ماتریکس فشار مرتبط  $\Omega$  از یک فشار متعادل  $\omega$  دارای طبقه  $|v|-(d+1)$  باشد.

#### ۴-۵- ساختار استنتاجی

یکی از مفیدترین متدلوژیها برای اختراع یک الگوریتم مکان یابی WSN توزیع شده از طریق گسترش پی در پی<sup>۹</sup> می باشد. بنابراین این یک کمک منحصر به فردی برای ایجاد ساختارها استنتاجی برای نمودارهای Rigid جهانی می باشد که با WSNS ارتباط یافته است. به همین دلیل، با ارائه یک نمودار  $G$ ، ما سعی میکنیم یک ادراک Rigid جهانی  $(G,P)$  در  $Rd$  را به وجود آوریم اگر چنین ادراکی وجود داشته باشد. ساختار Henneberg به صورت ابزار مهم ما در سر تاسر این زیر مجموعه شده است.

تعریف: اجازه دهیم  $G=(V,E)$  یک نمودار باشد. اولین مرحله Henneberg (H1) برای  $G$  به انجام پذیرفته شده است و یک رأس جدید اضافه شده است که با ۲ رأسی که سابقا موجود بوده اند ارتباط یافته اند. دومین مرحله Henneberg برای  $G$  به انجام پذیرفته شده است، یک لبه را توسط یک رأس جدید جایگزین می کند که به نقطه های پایانی خود و به یک رأس اختیاری مرتبط می شود.

قضیه: اجازه دهیم  $H$  یک نمودار محکم جهانی با حداقل ۴ رأس باشد و اجازه دهیم  $G$  از  $H$  توسط HennebergII کسب شده باشد. سپس  $G$  استحکام سراسری می باشد.

ضمنا زمانیکه یک گراف دو بعدی دارای بیش از  $3n-6$  لبه باشد، بصورت Rigid است. توجه شود همانطور که لبه های بیشتری اضافه می شوند قسمت عمده گراف سریعا بهم مرتبط می شود.

---

<sup>9</sup> Sequential extension

## ٦- فهرست منابع :

[1] Wang j., Han T., A Self-adapting Dynamic localization Algorithm for Mobile Nodes in Wireless Sensor Networks,Procedia Environmental Science 11,(2011) ,270-274 .

[2] Gustafsson F.,Gunnarsson F.,POSITIONING USING TIME-DIFFERENCE OF ARRIVAL MEASUREMENTS.Linkoping University, Sweden.

[3] Shon M.,Jo M.,Choo H.,An interactive cluster-based MDS localization scheme for multimedia information in wireless sensor networks,Computer Communications 35,(2012), 1921-1929 .

[4] Zhang Y.,Liu Sh.,Zhao X.,Jia Z.,Theoretic analysis of unique localization for wireless sensor networks,Ad Hoc Networks 10,(2012), 623-634 .