

توسعه یک الگوریتم بهترین مسیر در شرایط نایقینی و کمبود داده مبتنی بر نظریه فازی شهودی

علی سبزه‌علی یمقانی^۱، محمدرضا ملک^{۲*}

^۱ دانشجوی کارشناسی ارشد سیستم‌های اطلاعات مکانی - دانشکده مهندسی نقشه‌برداری - دانشگاه صنعتی خواجه

نصیرالدین طوسی

sabzali.ali@gmail.com

^۲ دانشیار گروه سیستم‌های اطلاعات مکانی - دانشکده مهندسی نقشه‌برداری - دانشگاه صنعتی خواجه نصیرالدین طوسی

mrmalek@kntu.ac.ir

(تاریخ دریافت مهر ۱۳۹۳، تاریخ تصویب اردیبهشت ۱۳۹۴)

چکیده

نظریه مجموعه‌های فازی شهودی، تعمیمی از نظریه مجموعه‌های فازی می‌باشد که در آن علاوه بر تابع عضویت از تابع عدم عضویت هم استفاده می‌شود. این مورد موجب شده تا بعضی از محدودیت‌های نظریه فازی معمول مثل پشتیبانی از شک و تردید را برطرف سازد. از طرفی با توجه به اینکه یکی از مسائل موجود در گراف، یافتن کوتاه‌ترین مسیر در شرایط عدم قطعیت و نبود اطلاع کافی از فاصله گره‌ها می‌باشد. با توجه به نکات ذکر شده در این مقاله الگوریتم کوتاه‌ترین مسیر دایجسترا برای گراف با یال‌های فازی شهودی در شرایط کمبود اطلاعات تعمیم داده شده است. در روش ارائه شده در مقاله برای مقایسه مسیرها از روش انتگرال‌گیری استفاده شده است. دلیل استفاده از این روش امکان دخیل‌سازی حالات مختلف تصمیم‌گیرنده یعنی بدبین، معمولی و خوشبین می‌باشد. این موضوع باعث می‌شود تا بتوان نتایج ورودی‌های یکسان را منطبق بر شرایط مختلف، تغییر داد. در مرحله بعد الگوریتم روی یک شبکه با ابعاد مناسب آزموده شده و با حالت فازی معمولی مقایسه شده است. در نهایت با ارائه شکل و جدول به مقایسه نتایج الگوریتم و ارائه تفاوت مسیرها در دو حالت فازی و فازی شهودی پرداخته شده است.

واژگان کلیدی: گراف فازی شهودی، مسئله کوتاه‌ترین مسیر، الگوریتم دایجسترا، اعداد فازی شهودی، نایقینی

* نویسنده رابط

۱- مقدمه

نیاز به پیش‌بینی شرایط مختلف احساس می‌شود. چراکه تغییر شرایط باعث تغییر طول مسیر می‌شود. از اینرو استفاده از مدل مبتنی بر فازی شهودی می‌تواند منطقی‌تر و با پشتیبانی نظری بیشتر باشد. از طرفی با توجه به اینکه در این مقاله از روش انتگرال‌گیری برای محاسبه مقدار امتیاز هر یال استفاده شده است، می‌توان حالت خوش‌بینی یا بدبینی را در محاسبه مسیر لحاظ نمود که این موضوع تاثیر شایان توجهی در تعیین نتایج بهتر و واقعی‌تر خواهد داشت.

در زمینه الگوریتم کوتاه‌ترین مسیر فازی کم و بیش تحقیقاتی صورت گرفته است. Klein [۶] در تحقیقی تعدادی راهکار جدید جهت مسئله کوتاه‌ترین مسیر فازی ارائه داده است. Lin و همکاران [۷] به بررسی مسئله کوتاه‌ترین مسیر فازی پرداختند، آنها به کمک روش برنامه‌ریزی خطی فازی به ایجاد توابع عضویت برای یال‌های شبکه راه همت گماشتند. Okada و همکاران [۸] به بررسی کاربرد الگوریتم کوتاه‌ترین مسیر فازی در مسئله مسیریابی در شبکه‌های راه پرداخته‌اند. Ji و همکاران [۹] به کمک استفاده از α -cut و الگوریتم ژنتیک یک روش ترکیبی توسعه دادند. سپس به کمک ارائه یک شبکه فرضی به بررسی کارایی الگوریتم خود پرداختند. Hernandez و همکاران [۱۰] در تحقیق خود الگوریتمی را توسعه دادند که به کمک آن می‌توان به تصمیم‌گیرنده در اتخاذ تصمیم نهایی کمک کرد. به عبارتی نتایج این الگوریتم به صورت فازی نمی‌باشد و تصمیم‌گیرنده راحت‌تر نتیجه نهایی را اتخاذ می‌کند. ضمناً باید توجه داشت که در بسیاری از تحقیقات قبل از این تحقیق نتایج به صورت فازی ارائه می‌شد و به تصمیم‌گیرنده کمکی در زمینه تصمیم‌گیری نهایی ارائه نمی‌شد. Deng و همکاران [۱۱] الگوریتم دایجسترا معمولی را برای حالت عدم اطمینان توسعه داده‌اند. روش انجام تحقیق به این صورت بوده است که از یال‌های با مقادیر فازی به جای مقادیر قطعی استفاده شده است. ایده استفاده از یال‌های فازی شهودی در مسئله کوتاه‌ترین مسیر توسط Karunambigai و همکاران [۱۲] ارائه شده است. Gani [۱۳] و Mukherjee [۱۴] به بررسی مسئله کوتاه‌ترین مسیر در گراف با اعداد فازی شهودی پرداخته‌اند. همچنین Rangasamy و همکاران [۱۵] نیز بر روی مسئله ارائه فرامسیر^۱ بر روی یک شبکه فرضی تحقیق کرده‌اند. نکته

مسئله یافتن کوتاه‌ترین مسیر از نقطه مبدا به نقطه مقصد در زمینه‌های گوناگونی کاربرد دارد. در یک شبکه طول یال‌ها می‌تواند نشان‌دهنده طول مسیر، زمان مسیر و یا هر پارامتر دیگری باشد. گراف جهت‌دار مانند $G(N,E)$ در نظر گرفته شود. مجموعه $N=\{1, 2, 3, \dots\}$ نشان‌دهنده گره‌های موجود در گراف می‌باشد. مجموعه E نیز حاوی تعدادی یال به صورت زوج مرتب‌های (i, j) می‌باشد [۱]. تا به حال کاربردهای مختلفی از مسئله کوتاه‌ترین مسیر فازی در مراجع مختلف بررسی شده است؛ برای مثال Dubois و همکاران [۲] به بررسی ابعاد مختلف مجموعه‌های فازی و کاربردهای آن در مسائلی مانند هوش مصنوعی، پردازش تصاویر، تشخیص مکالمه، پزشکی، روانشناسی، جامعه‌شناسی، اقتصاد و غیره پرداختند. Kristianto و همکاران [۳] به طراحی زنجیره تامین به کمک منطق فازی پرداخته‌اند. حال فرض شود یال‌هایی وجود دارد که دارای طول مشخصی نباشند یا اینکه طول آنها مشخص بوده ولی در اثر تاثیر پارامترهای دیگر مانند ترافیک، حادثه و غیره تغییر کنند. همچنین در مواردی مانند شبکه‌ی راه‌های مورد استفاده برای قاچاق، تردیدهای مختلفی در تعیین وزن یال‌ها، برای نیروهای امنیتی وجود دارد. حال آنکه در نظریه فازی تفاوتی بین عدم وجود اطلاعات و یا وجود دلایلی بر له و علیه یک موضوع داده نمی‌شود. به عبارت دیگر اگر تابع عضویت عنصری از مجموعه فازی نیم باشد؛ نمی‌توان استنباط کرد که اطلاعات اندک بوده و یا دلایل مثبت و منفی هر دو به یک اندازه موجودند [۴]. از طرفی زمانی که کمبود اطلاعات وجود دارد، در این حالت روش‌های متکی بر مجموعه‌های فازی به خوبی عمل نکرده ولی در مقابل مدل‌های فازی شهودی بهتر عمل می‌کنند [۴]. با توجه به نکاتی که اشاره شد، در این شرایط استفاده از مسئله کوتاه‌ترین مسیر با یال‌های قطعی یا فازی [۵] می‌تواند نتایج مناسبی نداشته باشد. در نتیجه، این مسائل انگیزه‌ای برای استفاده از گراف با یال‌های فازی شهودی به جای فازی معمولی شده است.

همانطور که اشاره شد در بعضی مسائل مانند مسیریابی خودروهای حمل قاچاق مشکل نبود اطلاعات کافی وجود دارد. از طرفی حتی در صورت وجود اطلاعات از مسیر اشرار

^۱ hyperpath

$$0 \leq \mu_A(x) + v_A(x) \leq 1, \forall x \in X \quad (2)$$

می‌باشد. اعداد $\mu_A(x)$ و $v_A(x)$ به ترتیب مقدار تابع عضویت و عدم عضویت المان x از مجموعه A را نمایش می‌دهند. اعداد فازی شهودی توسعه‌ای از مجموعه فازی شهودی می‌باشد [۲۴].

تعریف ۲. عدد فازی شهودی مثلثی A به صورت:

$$A = ([\langle a, b, c \rangle; \mu_A], [\langle e, f, g \rangle; v_A]) \quad (3)$$

می‌باشد که در آن:

$$0 \leq \mu_A \leq 1, 0 \leq v_A \leq 1, 0 \leq \mu_A + v_A \leq 1 \quad (4)$$

همچنین $v_A \leq \mu_A^c(x)$ می‌باشد، بنابراین:

$$\langle e, f, g \rangle \leq \langle a, b, c \rangle^c \quad (5)$$

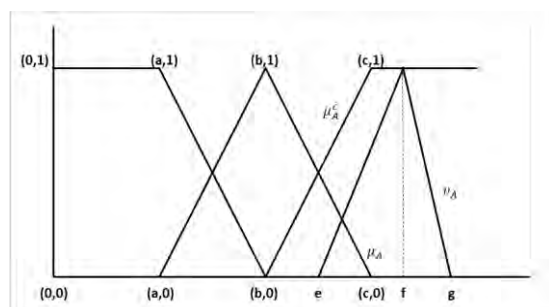
می‌باشد. همچنین باید روابط:

$$f \leq a \ \& \ g \leq b \quad \text{یا} \quad e \geq b \ \& \ f \geq c \quad (6)$$

برقرار باشد. در شکل ۱ نمایش عدد فازی شهودی مثلثی:

$$A = (\langle a, b, c \rangle, \langle e, f, g \rangle) \quad (7)$$

با شرط $e \geq b \ \& \ f \geq c$ آورده شده است [۲۴، ۱۵].



شکل ۱- عدد فازی شهودی مثلثی $A = (\langle a, b, c \rangle, \langle e, f, g \rangle)$

۲-۲- الگوریتم دایجسترا

الگوریتم دایجسترا توسط دانشمند هلندی ادزجر دایجسترا معرفی شد. این الگوریتم برای گره مبدا در یک گراف، مسیر با کمترین هزینه (مانند کوتاه‌ترین مسیر از لحاظ طولی) از آن گره به هر گره دیگر را پیدا می‌کند.

قابل توجه در این تحقیقات عدم دخیل‌سازی حالات مختلف تصمیم‌گیرنده می‌باشد. به عبارتی دیگر بین حالت خوشبینانه، معمولی و بدبینانه تفاوتی قائل نشده است. این نکته موجب شد تا نیاز به ایجاد یک الگوریتم مسیریابی ساده و کارا بر پایه اعداد فازی شهودی و با امکان دخیل‌سازی حالات مختلف تصمیم‌گیرنده احساس شود. در این مقاله الگوریتم دایجسترا معمولی به حالت فازی شهودی توسعه داده شده است. به این صورت که به جای استفاده از یال‌هایی با مقادیر کریسپ از یال‌های با مقادیر فازی شهودی استفاده شده است.

ساختار این مقاله به این صورت است که در بخش دوم مقاله، دو مشکل مطرح و راه‌حل آنها ارائه می‌شود. اولی نحوه ترکیب یال‌ها می‌باشد که به کمک استفاده از اعداد فازی شهودی مثلثی این مشکل حل شده است؛ دومی نحوه مقایسه یال‌ها می‌باشد. برای مقایسه یال‌ها با یکدیگر روش‌های متفاوتی وجود دارد. از جمله آنها مقایسه بر اساس مرکز ثقل [۱۶، ۱۷]، مجموعه‌های بیشینه و کمینه [۱۸]، انتگرال گیری [۱۹، ۲۰] و روش‌های دیگر [۲۱، ۲۲، ۲۳، ۲۴] می‌باشد. در نهایت روش انتگرال گیری برای این تحقیق استفاده شده است، در بخش سوم به نحوه مدلسازی الگوریتم دایجسترا در گراف فازی شهودی و نحوه اجرا آن پرداخته می‌شود. بخش چهارم به ارائه یک مثال عددی از یک گراف با ابعاد مناسب، برای نمایش کارایی الگوریتم و مقایسه آن با حالت فازی معمولی اختصاص داده شده است. در نهایت بخش پنجم به نتیجه‌گیری مقاله اختصاص داده شده است.

۲- مبانی

در این بخش در مورد اعداد فازی شهودی مثلثی، نحوه امتیازدهی و رتبه بندی آنها بحث می‌شود. در مرحله بعد از آنها برای ارائه الگوریتم دایجسترا در گراف با یال‌های فازی شهودی استفاده می‌شود [۲۵، ۲۶، ۲۷].

۲-۱- مبانی مجموعه فازی شهودی

تعریف ۱. یک مجموعه فازی شهودی توسط:

$$A = \{ \langle x, \mu_A(x), v_A(x) \rangle \mid x \in X \} \quad (1)$$

نمایش داده می‌شود، به طوریکه $\mu_A: X \rightarrow [0,1]$ و $v_A: X \rightarrow [0,1]$ ، با شرط:

مشکل اصلی الگوریتم دایجسترا در شبکه با یال‌های فازی شهودی، نحوه ترکیب یال‌ها و همچنین تعیین امتیاز برای گره‌ها می‌باشد. در ادامه راه‌حلی برای مرتفع‌سازی مشکلات ذکر شده ارائه می‌شود.

فرض شود:

$$A = (< a_1, b_1, c_1 >, < e_1, f_1, g_1 >) \quad (8)$$

و:

$$B = (< a_2, b_2, c_2 >, < e_2, f_2, g_2 >) \quad (9)$$

دو عدد فازی شهودی مثلثی باشند. در نتیجه مجموع این دو عدد به صورت:

$$A + B = (< a_1 + a_2, b_1 + b_2, c_1 + c_2 >, < e_1 + e_2, f_1 + f_2, g_1 + g_2 >) \quad (10)$$

بیان می‌شود [۱۵]، با توجه به سهولت ترکیب اعداد فازی شهودی مثلثی در این مقاله بدون از دست دادن کلیت از این اعداد استفاده شده است. پس تا به حال مشکل ترکیب یال‌ها مرتفع شد. فرض شود:

$$A = (< a, b, c >, < e, f, g >) \quad (11)$$

یک عدد فازی شهودی مثلثی باشد. برای محاسبه میزان امتیاز عضویت و عدم عضویت این عدد می‌توان از روش انتگرال‌گیری به صورت:

$$S_{A-M}^{\alpha} = \left(\frac{1}{2}\right) [ac + b + (1 - \alpha)a] \quad (12)$$

$$S_{A-N}^{\alpha} = \left(\frac{1}{2}\right) [ag + f + (1 - \alpha)e] \quad (13)$$

استفاده کرد. در اینجا $\alpha \in [0,1]$ می‌باشد که برای تصمیم‌گیرنده خوشبین مقدار ۱ و برای تصمیم‌گیرنده بدبین مقدار ۰ را اخذ می‌کند. در این تحقیق مقدار این ضریب ۰,۵ انتخاب شده است. البته بسته به شرایط مختلف می‌توان مقادیر متفاوتی را اتخاذ کند. حال فرض شود:

$$A = (< a, b, c >, < e, f, g >) \quad (14)$$

همچنین می‌توان از این الگوریتم برای تعیین کوتاه‌ترین مسیر بین دو گره مشخص استفاده کرد؛ دلیل آن نیز این است که نتیجه الگوریتم در هر مرحله یک درخت می‌باشد. پس با خاتمه الگوریتم در نقطه مورد نظر نتیجه دلخواه حاصل می‌شود. حال برای مثال فرض شود، گره‌های موجود در گراف نمایانگر مقاصد سفر و یال‌های موجود در گراف نمایانگر طول شبکه باشند. در این صورت می‌توان از طریق این الگوریتم کوتاه‌ترین مسیر بین مقاصد سفر را پیدا کرد [۱۱].

فرض شود گره آغازین، نقطه مبدا نامیده شود. در نتیجه طول مسیر گره X برابر فاصله نقطه مبدا تا X می‌باشد. الگوریتم دایجسترا مجموعه‌ای از مقادیر اولیه برای طول مسیر در نظر می‌گیرد و آنها را مرحله به مرحله بهبود می‌بخشد. در ادامه خلاصه‌ای از روش کار الگوریتم ارائه می‌شود:

- ۱- نسبت دادن مقدار بی‌نهایت به عنوان طول اولیه به همه گره‌ها به جز نقطه مبدا (مقدار صفر می‌گیرد).
- ۲- ایجاد یک بردار با طول تعداد کل گره‌ها برای ذخیره‌سازی گره قبلی هر گره: مقدار null برای تمامی گره‌ها به جز نقطه مبدا قرار داده می‌شود. نقطه مبدا نیز مقداری ندارد و خالی می‌ماند.
- ۳- ابتدا همه‌ی نقاط در مجموعه نقاط بازدید نشده قرار می‌گیرند.

- ۴- حال برای تمامی گره‌هایی که با گره فعلی یال مشترک دارند، پروسه رلکسیشن انجام می‌شود. برای مثال اگر فاصله گره B از A مقدار ۳ باشد و مقدار طول گره A برابر ۴ باشد در نتیجه مقدار جدید طول گره B برابر ۷ می‌شود. اگر این مقدار از مقدار قبلی طول گره B کمتر باشد جایگزین آن می‌شود. سپس نقطه قبلی گره B گره A ثبت می‌شود.

- ۵- بعد از اینکه نزدیک‌ترین گره از گره فعلی تعیین شد. سپس گره فعلی از مجموعه گره‌های بازدید نشده خارج می‌شود.

- ۶- اگر تمامی گره‌ها در مجموعه گره‌های بازدید شده باشند، الگوریتم خاتمه می‌یابد. در غیر این صورت از بین گره‌های بازدید نشده، گره با کمترین طول به عنوان نقطه بعدی انتخاب می‌شود و سپس الگوریتم به مرحله ۴ باز می‌گردد.

۳- مدل‌سازی

۲- حال برای هر گره v (متصل به u با یک یال مشترک) مقدار $relax = Dist[u] + Dist_between(u, v)$ محاسبه می‌شود. اگر رابطه $relax < Dist[v]$ برقرار باشد. در این صورت مقدار $Dist[v] = relax$ و $previous[v] = u$ قرار داده می‌شود.

۳- حال در میان گره‌هایی که به عنوان v انتخاب شده‌اند، گره با کمترین مقدار تابع $Dist[]$ انتخاب می‌شود و به عنوان u جایگزین می‌شود. سپس به مرحله اول رجوع می‌شود.

در نهایت توجه شود که در تمامی مراحل الگوریتم برای ترکیب، محاسبه امتیاز و مقایسه یال‌ها از فرضیات ارائه شده استفاده شده است.

```

1 function intuitionistic fuzzy Dijkstra(Graph, source):
2   for each vertex v in Graph:
3     Dist[v] := infinity;
4     Previous[v] := null;
5   end for;
6   Dist[source] := 0;
7   S := the set of all nodes in Graph
8   while S is not empty :
9     u := vertex in S with smallest Dist[];
10    if Dist[u] = infinity :
11      break;
12    end if;
13    remove u from S;
14    for each neighbor v of u :
15      relax = Dist[u] + Dist_between(u, v);
16      //the intuitionistic triangular number like A={<a, b, c>, <e, f, g>} is used
17      if relax < Dist[v] :
18        Dist[v] := relax;
19        Previous[v] := u;
20        decrease-key v in S;          //Reorder v in the Queue
21    end if;
22  end for;
23  end while;
24  return Dist[];
25 end Dijkstra.

```

شکل ۲- شبه کد الگوریتم دایجسترا فازی شهودی ارائه شده

۴- پیاده‌سازی

در این بخش یک مثال عددی از یک شبکه راه ارائه می‌شود. همانطور که در شکل ۳ مشاهده می‌شود این شبکه شامل ۲۵ گره و ۹۶ یال می‌باشد. همچنین فرض شده مقادیر یال‌ها اعداد فازی شهودی مثلثی باشد که در جدول ۱ نمایش داده شده است. همانطور که در جدول‌های ۲ و ۳ مشاهده می‌شود، به ترتیب حل مسئله کوتاه‌ترین مسیر با یال‌های فازی و فازی شهودی انجام شده است. در ادامه به ارائه توضیح در مورد نحوه حل گراف با یال‌های فازی شهودی پرداخته می‌شود. ضمناً نحوه حل گراف با یال‌های فازی هم به همان نحو صورت می‌پذیرد. فقط باید توجه داشت که در حالت فازی معمولی فقط از امتیاز محاسبه شده از تابع عضویت استفاده می‌شود.

یک عدد فازی شهودی مثلثی باشد. مقدار امتیاز برای شاخص دقت به صورت :

$$S_{A-AC}^{\alpha} = \frac{S_{A-M}^{\alpha} + S_{A-N}^{\alpha}}{2} \quad (15)$$

تعریف می‌شود [۲۸]، در نهایت فرض شود :

$$A = (< a_1, b_1, c_1 >, < e_1, f_1, g_1 >) \quad (16)$$

و:

$$B = (< a_2, b_2, c_2 >, < e_2, f_2, g_2 >) \quad (17)$$

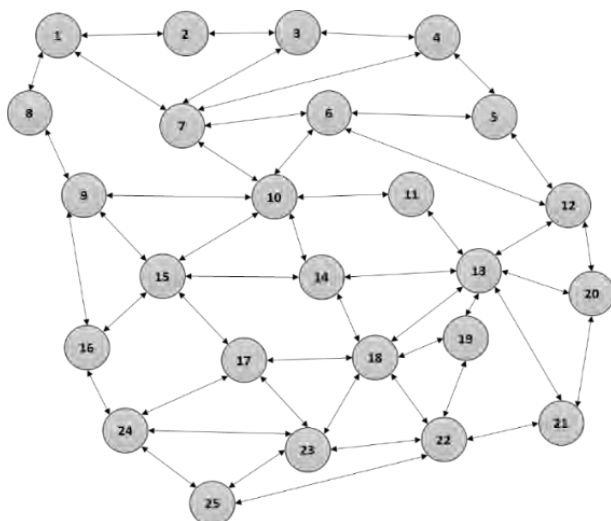
دو عدد فازی شهودی مثلثی باشند. اگر $S_{A-AC}^{\alpha} < S_{B-AC}^{\alpha}$ باشد، در نتیجه $A < B$ می‌باشد.

همانطور که مشاهده شد، مشکل محاسبه امتیاز مسیرهای مختلف و مقایسه آنها به کمک استفاده از روش انتگرال‌گیری و محاسبه مقدار امتیاز شاخص دقت مرتفع شد. حال می‌توان به کمک این فرضیات الگوریتم دایجسترا عادی را به محیط فازی شهودی بسط داد.

در ادامه شبه کد الگوریتم ارائه شده توضیح داده می‌شود. مجموعه S شامل مجموعه گره‌های بازدید نشده می‌باشد. در هر مرحله که به گره با کمترین فاصله از گره فعلی رسیده می‌شود؛ یک گره از این مجموعه خارج می‌شود. مقدار تابع $Dist[v]$ نمایش‌دهنده کوتاه‌ترین فاصله فعلی نقطه v از نقطه مبدا می‌باشد. مقدار تابع $Dist_between(u, v)$ نشان دهنده طول یال بین u و v می‌باشد. مقدار تابع $previous[v]$ برابر نزدیک‌ترین نقطه قبل از نقطه v ، در طول کوتاه‌ترین مسیر از نقطه مبدا تا v می‌باشد. از متغیر $relax$ نیز برای نگه‌داری مقدار طول جدید نقطه v استفاده شده است.

الگوریتم ارائه شده در شکل ۲ به صورت زیر عمل می‌کند:

۱- ابتدا چک می‌شود تا S خالی نباشد؛ در صورت خالی نبودن S ، u در S پیدا می‌شود. اگر مقدار $Dist[u]$ برابر بی‌نهایت نباشد، u از مجموعه S خارج می‌شود (توجه شود u در هر مرحله گره با کمترین مقدار $Dist[]$ می‌باشد).

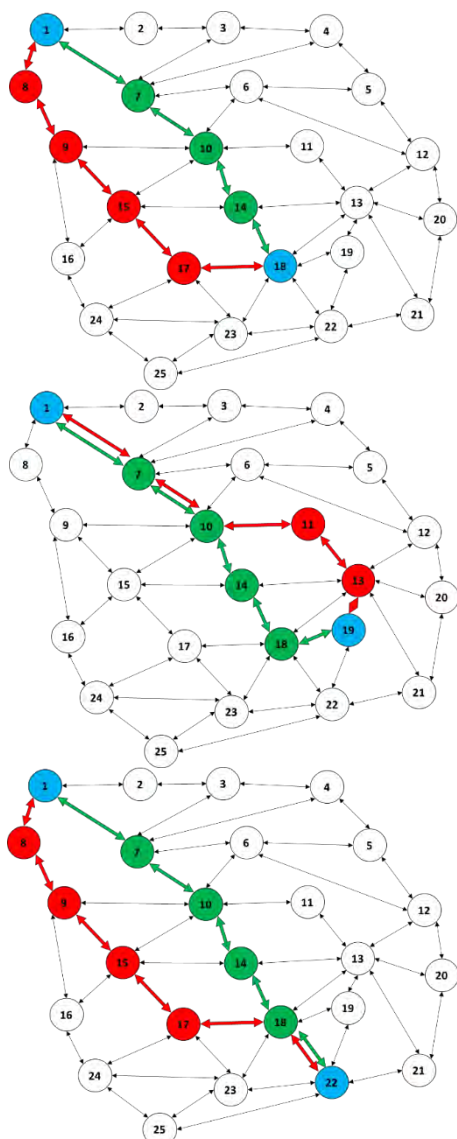


شکل ۳- شبکه فازی شهودی

جدول ۱- وزن یال‌های شبکه به صورت اعداد فازی شهودی مثلثی به همراه امتیاز محاسبه شده برای آنها

یال	طول به صورت فازی شهودی	امتیاز محاسبه شده	یال	طول به صورت فازی شهودی	امتیاز محاسبه شده
(1,2)	<2,3,5> <3,5,5,7>	<3.25,5.25,4.25>	(13,14)	<4,5,7> <5.5,8,9>	<5.25,7.625,6.4375>
(1,7)	<4,6,7> <6.5,8,9>	<5.75,7.875,6.8125>	(13,18)	<4,5,6> <5.5,7,9>	<5,7.125,6.0625>
(1,8)	<1.5,3,4> <4,5,7>	<2.875,5.25,4.0625>	(13,19)	<1.5,2,3> <2,3,5,5>	<2.125,3.5,2.8125>
(2,3)	<2,3,4> <3,5,6>	<3,4.75,3.875>	(13,20)	<4,5,6,5> <5.5,7,8>	<5.125,6.875,6>
(3,4)	<3,4,6> <4,6,5,8>	<4.25,6.25,5.25>	(13,21)	<4,6,7> <6,7,5,8,5>	<5.75,7.375,6.5625>
(3,7)	<3.5,4,5> <4.5,5,7>	<4.125,5.375,4.75>	(14,15)	<4.5,6,7> <6.5,7,9>	<5.875,7.375,6.625>
(4,5)	<2.5,3,5,5> <4,6,7>	<3.625,5.75,4.6875>	(14,18)	<3,4,5> <9,12,13,5>	<4,11.625,7.8125>
(4,7)	<6,7,9> <7,10,12>	<7.25,9.75,8.5>	(15,16)	<2.5,4,5> <4,5,5,7>	<3.875,5.5,4.6875>
(5,6)	<4,5,7> <5.5,7,8>	<5.25,6.875,6.0625>	(15,17)	<3,4,5,5> <4,6,7,5>	<4.125,5.875,5>
(5,12)	<3,4,6> <4,7,8>	<4.25,6.5,5.375>	(16,24)	<3,4,5> <4,6,6,5>	<4,5.625,4.8125>
(6,7)	<3,4,6> <4,7,9>	<4.25,6.75,5.5>	(17,18)	<4,5,6> <5.5,7,7,5>	<5,6.75,5.875>
(6,10)	<2.5,3,5> <3,6,7>	<3.375,5.5,4.4375>	(17,23)	<3,4,5> <4,5,5,6,5>	<4,5.375,4.6875>
(6,12)	<5.5,6,5,8> <7,9,10>	<6.625,8.75,7.6875>	(17, 24)	<3.5,5,6> <5.5,7,9>	<4.875,7.125,6>
(7,10)	<2.5,3,5> <3.5,5,6>	<3.375,4.875,4.125>	(18,19)	<2,3,3,5> <3,4,5>	<2.875,4,3.4375>
(8,9)	<2,3,4> <4,5,7>	<3,5.25,4.125>	(18,22)	<3,4,4,5> <4,5,6>	<3.875,5,4.4375>
(9,10)	<4,5,7> <5,8,9>	<5.25,7.5,6.375>	(18,23)	<3,3,5,5> <4,5,5,6,5>	<3.75,5.375,4.5625>
(9,15)	<2.5,3,5> <3,6,7>	<3.375,5.5,4.4375>	(19,22)	<3.5,4,5,6> <5,7,8>	<4.625,6.75,5.6875>
(9,16)	<3.5,5,6> <5,7,8>	<4.875,6.75,5.8125>	(20,21)	<4,5,7> <5,7,5,8,5>	<5.25,7.125,6.1875>
(10,11)	<3,4,5> <5,6,8>	<4,6.25,5.125>	(21,22)	<4,5,6> <5.5,6,5,8>	<5,6.625,5.8125>
(10,14)	<2,3,4> <9,12,15>	<3,12,7.5>	(22,23)	<4,6,7> <6,7,5,9>	<5.75,7.5,6.625>
(10,15)	<4,4,5,5> <5,6,8>	<4.5,6.25,5.375>	(22,25)	<6.5,7,5,9> <8,9,5,11>	<7.625,9.5,8.5625>
(11,13)	<2.5,4,5> <4.5,5,6>	<3.875,5.125,4.5>	(23,24)	<5,6,8> <6.5,9,10>	<6.25,8.625,7.4375>
(12,13)	<3.5,5,5,5> <5,6,7,5>	<4.75,6.125,5.4375>	(23,25)	<3,4,4,5> <4.5,5,5,6,5>	<3.875,5.5,4.6875>
(12,20)	<2,3,4> <3,4,5,5>	<3,4.25,3.625>	(24,25)	<4,5,6,5> <6,7,9>	<5.125,7.25,6.1875>

در نهایت شکل ۴ به خوبی اختلاف مسیرها را در گراف نمایش می‌دهد. رنگ سبز نشان‌دهنده حالت فازی معمولی و رنگ قرمز نشان‌دهنده حالت فازی شهودی می‌باشد. همچنین گره‌های آبی ابتدا و انتهای مسیر می‌باشند. دلیل این اختلاف می‌تواند مواردی مانند: ایجاد شرایط خاص که باعث شود مسیری که در ظاهر کوتاه‌تر است، زمان بیشتر برای طی شدن ببرد و یا زمانی که اطلاعات مناسبی از مسیر خاصی وجود ندارد در نتیجه آن مسیر با احتیاط بیشتر (طولانی‌تر) برآورد شود و یا موارد مشابه باشد.



شکل ۴- اختلاف مسیر حاصل از الگوریتم‌های مسیریابی فازی شهودی (قرمز) و فازی معمولی (سبز)

سه مرحله اول الگوریتم با تمامی جزئیات در جدول‌های ۲ و ۳ آورده شده است. فرض شود S مجموعه یال‌هایی است که بازدید شده‌اند و U مجموعه یال‌هایی است که بازدید نشده‌اند.

در مرحله اول گره شماره (۱) که نقطه مبدا می‌باشد، از مجموعه U به مجموعه S انتقال می‌یابد، و کوتاه‌ترین فاصله (۱) تا (۱) برابر صفر می‌باشد. سپس کوتاه‌ترین فاصله از نقطه (۱) به نقاط قابل دسترس محاسبه می‌شود. حال کوتاه‌ترین مسیر یعنی (۱)←(۸) با طول ۴,۰۶۲۵ انتخاب می‌شود.

در مرحله دوم گره شماره (۸) از مجموعه U به مجموعه S انتقال داده می‌شود. کوتاه‌ترین فاصله از (۱)←(۸) به یال-های متصل به گره (۸) محاسبه می‌شود. در نهایت مشاهده می‌شود که هیچ کدام از مسیرها از مسیر (۱)←(۲) کوتاه‌تر نمی‌باشند. در نتیجه این مسیر به عنوان کوتاه‌ترین مسیر با طول ۴,۲۵ انتخاب می‌شود.

در مرحله سوم گره شماره (۲) از مجموعه U به مجموعه S انتقال داده می‌شود. کوتاه‌ترین فاصله از (۱)←(۲) به یال-های متصل به گره (۲) محاسبه می‌شود. در نهایت مشاهده می‌شود که هیچ کدام از مسیرها از مسیر (۱)←(۷) کوتاه‌تر نمی‌باشند. در نتیجه این مسیر به عنوان کوتاه‌ترین مسیر با طول ۶,۸۱۲۵ انتخاب می‌شود.

اگر به همین ترتیب ادامه داده شود، در نهایت مشاهده می‌شود که در مرحله ۲۵ الگوریتم به پایان کار خود می‌رسد. طولانی‌ترین مسیر (۱)←(۸)←(۹)←(۱۵)←(۱۷)←(۱۸)←(۲۲) می‌باشد که دارای طول ۲۷,۹۳۷۵ می‌باشد.

همانطور که در جدول‌های ۲ و ۳ مشاهده می‌شود در بعضی از مسیرها در دو حالت فازی و فازی شهودی نتایج متفاوتی وجود دارد. برای مثال در حالت فازی برای رفتن از نقطه ۱ به نقطه ۱۸ از مسیر (۱)←(۷)←(۱۰)←(۱۴)←(۱۸) استفاده می‌شود. در حالیکه در حالت فازی شهودی از مسیر (۱)←(۸)←(۹)←(۱۵)←(۱۷)←(۱۸) استفاده می‌شود. این موضوع همچنین در مسیرهای دیگر مانند نقطه ۱ به نقاط ۱۹ یا ۲۲ اتفاق افتاده است. همچنین با مراجعه به جدول ۴ می‌توان میزان اختلاف مسیرها را در حالت‌های فازی معمولی و فازی شهودی مشاهده کرد.

جدول ۲- مراحل حل الگوریتم ارائه شده

حالت فازی معمولی		مرحله
Set U (unsettled) (fuzzy)	Set S (shortest path) (fuzzy)	
$U = \{(2), (3), \dots (25)\}$ $(1) \rightarrow (2) = <3.25>$ $(1) \rightarrow (7) = <5.75>$ $(1) \rightarrow (8) = <2.875>$ $(1) \rightarrow (\text{other}) = \infty$ Find out the shortest path: $(1) \rightarrow (8) = <2.875>$	$S = \{(1)\}$ $(1) \rightarrow (1) = 0$ Seek from (1)	۱
$U = \{(2), (3), \dots (25)\}$ $(1) \rightarrow (8) \rightarrow (9) = <5.875>$ $(1) \rightarrow (8) \rightarrow (\text{other}) = \infty$ Find out the shortest path: $(1) \rightarrow (2) = <3.25>$	$S = \{(1), (8)\}$ $(1) \rightarrow (1) = 0$ $(1) \rightarrow (8) = <2.875>$ Seek from $(1) \rightarrow (8)$	۲
$U = \{(3), (4), \dots (25)\}$ $(1) \rightarrow (2) \rightarrow (3) = <6.25>$ $(1) \rightarrow (2) \rightarrow (\text{other}) = \infty$ Find out the shortest path: $(1) \rightarrow (7) = <5.75>$	$S = \{(1), (8), (2)\}$ $(1) \rightarrow (1) = 0$ $(1) \rightarrow (8) = <2.875>$ $(1) \rightarrow (2) = <3.25>$ Seek from $(1) \rightarrow (2)$	۳
		.
		.
		.
$U = \{\}$ پایان الگوریتم!	$S = \{(1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19), (20), (21), (22), (23), (24), (25)\};$ $(1) \rightarrow (1) = 0$ $(1) \rightarrow (8) = 2.875$ $(1) \rightarrow (2) = 3.25$ $(1) \rightarrow (7) = 5.75$ $(1) \rightarrow (8) \rightarrow (9) = 5.875$ $(1) \rightarrow (2) \rightarrow (3) = 6.25$ $(1) \rightarrow (7) \rightarrow (10) = 9.125$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (15) = 9.25$ $(1) \rightarrow (7) \rightarrow (6) = 10$ $(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) = 10.5$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (16) = 10.75$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (14) = 12.125$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (11) = 13.125$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (15) \rightarrow (17) = 13.375$ $(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \rightarrow (5) = 14.125$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (16) \rightarrow (24) = 14.75$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (14) \rightarrow (18) = 16.125$ $(1) \rightarrow (7) \rightarrow (6) \rightarrow (12) = 16.625$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (11) \rightarrow (13) = 17$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (15) \rightarrow (17) \rightarrow (23) = 17.375$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (14) \rightarrow (18) \rightarrow (19) = 19$ $(1) \rightarrow (7) \rightarrow (6) \rightarrow (12) \rightarrow (20) = 19.625$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (16) \rightarrow (24) \rightarrow (25) = 19.875$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (14) \rightarrow (18) \rightarrow (22) = 20$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (11) \rightarrow (13) \rightarrow (21) = 22.75$	۲۵

توسعه یک الگوریتم بهترین مسیر در شرایط نایقینی و کمبود...

جدول ۳- مراحل حل الگوریتم ارائه شده

حالت فازی شهودی		مرحله
Set U (unsettled) (Intuitionistic fuzzy)	Set S (shortest path) (Intuitionistic fuzzy)	
$U = \{(2), (3), \dots, (25)\}$ $(1) \rightarrow (2) = \langle 3.25, \dots, 5.25 \rangle, \langle 4.25 \rangle$ $(1) \rightarrow (7) = \langle 5.75, \dots, 7.875 \rangle, \langle 6.8125 \rangle$ $(1) \rightarrow (8) = \langle 2.875, \dots, 5.25 \rangle, \langle 4.0625 \rangle$ $(1) \rightarrow (\text{other}) = \infty$ Find out the shortest path: $(1) \rightarrow (8) = \langle 2.875, \dots, 5.25 \rangle, \langle 4.0625 \rangle$	$S = \{(1)\}$ $(1) \rightarrow (1) = 0$ Seek from (1)	۱
$U = \{(2), (3), \dots, (25)\}$ $(1) \rightarrow (8) \rightarrow (9) = \langle 5.875, \dots, 10.5 \rangle, \langle 8.1875 \rangle$ $(1) \rightarrow (8) \rightarrow (\text{other}) = \infty$ Find out the shortest path: $(1) \rightarrow (2) = \langle 3.25, \dots, 5.25 \rangle, \langle 4.25 \rangle$	$S = \{(1), (8)\}$ $(1) \rightarrow (1) = 0$ $(1) \rightarrow (8) = \langle 2.875, \dots, 5.25 \rangle, \langle 4.0625 \rangle$ Seek from $(1) \rightarrow (8)$	۲
$U = \{(3), (4), \dots, (25)\}$ $(1) \rightarrow (2) \rightarrow (3) = \langle 6.25, \dots, 10 \rangle, \langle 8.125 \rangle$ $(1) \rightarrow (2) \rightarrow (\text{other}) = \infty$ Find out the shortest path: $(1) \rightarrow (7) = \langle 5.75, \dots, 7.875 \rangle, \langle 6.8125 \rangle$	$S = \{(1), (8), (2)\}$ $(1) \rightarrow (1) = 0$ $(1) \rightarrow (8) = \langle 2.875, \dots, 5.25 \rangle, \langle 4.0625 \rangle$ $(1) \rightarrow (2) = \langle 3.25, \dots, 5.25 \rangle, \langle 4.25 \rangle$ Seek from $(1) \rightarrow (2)$	۳
		۲۵
$U = \{\}$ پایان الگوریتم!	$S = \{(1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19), (20), (21), (22), (23), (24), (25)\};$ $(1) \rightarrow (1) = 0$ $(1) \rightarrow (8) = 4.0625$ $(1) \rightarrow (2) = 4.25$ $(1) \rightarrow (7) = 6.8125$ $(1) \rightarrow (2) \rightarrow (3) = 8.125$ $(1) \rightarrow (8) \rightarrow (9) = 8.1875$ $(1) \rightarrow (7) \rightarrow (10) = 10.9375$ $(1) \rightarrow (7) \rightarrow (6) = 12.3125$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (15) = 12.625$ $(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) = 13.375$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (16) = 14$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (11) = 16.0625$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (15) \rightarrow (17) = 17.625$ $(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \rightarrow (5) = 18.0625$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (14) = 18.4375$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (16) \rightarrow (24) = 18.8125$ $(1) \rightarrow (7) \rightarrow (6) \rightarrow (12) = 20$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (11) \rightarrow (13) = 20.5625$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (15) \rightarrow (17) \rightarrow (23) = 22.3125$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (11) \rightarrow (13) \rightarrow (19) = 23.375$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (15) \rightarrow (17) \rightarrow (18) = 23.5$ $(1) \rightarrow (7) \rightarrow (6) \rightarrow (12) \rightarrow (20) = 23.625$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (16) \rightarrow (24) \rightarrow (25) = 25$ $(1) \rightarrow (7) \rightarrow (10) \rightarrow (11) \rightarrow (13) \rightarrow (21) = 27.125$ $(1) \rightarrow (8) \rightarrow (9) \rightarrow (15) \rightarrow (17) \rightarrow (18) \rightarrow (22) = 27.9375$	

جدول ۴- مقایسه نتایج مسیرها در حالت‌های فازی و فازی شهودی

مسیر	طول حالت فازی	طول حالت فازی شهودی
۱ به ۷ به ۱۰ به ۱۴ به ۱۸ به ۱۹	۱۹	۲۹,۶۸۷۵
۱ به ۷ به ۱۰ به ۱۱ به ۱۳ به ۱۹	۱۹,۱۲۵	۲۳,۳۷۵
مسیر	طول حالت فازی	طول حالت فازی شهودی
۱ به ۷ به ۱۰ به ۱۴ به ۱۸	۱۶,۱۲۵	۲۶,۲۵
۱ به ۸ به ۹ به ۱۵ به ۱۷ به ۱۸	۱۸,۳۷۵	۲۳,۵
مسیر	طول حالت فازی	طول حالت فازی شهودی
۱ به ۷ به ۱۰ به ۱۴ به ۱۸ به ۲۲	۲۰	۳۰,۶۸۷۵
۱ به ۸ به ۹ به ۱۵ به ۱۷ به ۱۸ به ۲۲	۲۲,۲۵	۲۷,۹۳۷۵

۵- نتیجه گیری

الگوریتم ارائه شده قابل استفاده در مسائل دنیای واقعی به خصوص در زمینه مسیریابی در شرایط بحرانی و جاده های روستایی و جاده های مرزی می باشد. همچنین نکته قابل توجه دیگر این است که در بعضی کاربردها به دلیل عدم وجود یک مسیر با طول و یا دیگر شرایط مشخص، این روش پتانسیل زیادی برای استفاده خواهد داشت. به عنوان یک نمونه می توان از مسئله مسیریابی خودروهای قاچاق کالا در مرز کشورها یاد نمود. همچنین استفاده از منطق فازی شهودی در شرایط ناکافی بودن داده و ابهام هندسی منجر به نتایج مناسب تری نسبت به منطق فازی معمولی می شود. زیرا در این شرایط مسیره های ممکن و پیشنهادی را می توان بصورت های بدبینانه و در کمال احتیاط، خوشبینانه و یا حالت های بین آن دو محاسبه کرد. در این صورت نه تنها اطلاعات بلکه عدم اطلاعات و همچنین شک و دودلی های موجود نیز در نظر گرفته شده اند. در نتیجه استفاده از این الگوریتم نتایج منطبق تر با شرایط داده شده و موجود را به تصمیم گیرنده ارائه می دهد. نکته نهایی که باید به آن اشاره کرد این است که از آنجا که هدف مقاله حاضر ارائه یک راه حل قابل قبول و با مبانی استدلالی برای حل مشکل بوده است، لذا در این تحقیق به مسایل Performance پرداخته نشده است. در اینجا باید یک راه حل مناسب ارائه داد که توانایی پشتیبانی از چنین عدم قطعیتی را داشته باشد و سپس آنرا بهبود داده تا بتوان بصورت تجاری و عملی نیز استفاده کرد.

این مقاله الگوریتم دایجسترا را برای حل مسئله کوتاه ترین مسیر با یال های فازی شهودی توسعه داده است. دو مشکل اصلی برای انجام این کار وجود داشت. اولی نحوه جمع مقادیر دو یال بود که با توجه به استفاده از اعداد فازی شهودی مثلی این مشکل حل شد. دومی نحوه مقایسه دو یال بود که برای حل این مسئله از روش امتیازدهی انتگرال گیری و محاسبه شاخص امتیاز دقت استفاده شد. در نهایت یک مثال عددی برای نمایش کارایی الگوریتم ارائه شد. همچنین مقایسه این الگوریتم در دو حالت فازی و فازی شهودی ارائه شد که دلایل اختلاف در انتخاب مسیر نهایی به صورت زیر می باشد:

ایجاد شرایط خاص که باعث شود مسیری که در ظاهر کوتاه تر است، زمان بیشتر برای طی شدن ببرد. مثلا در شرایط بحرانی ممکن است خرابی در خیابان ایجاد شود که باعث افزایش زمان عبور وسایل نقلیه امداد از مسیر شود. زمانی که اطلاعات مناسبی از مسیر خاصی وجود ندارد در نتیجه آن مسیر با احتیاط بیشتری (طولانی تر) برآورد می شود. مثلا در هنگام تعقیب اشرار در مرز به دلیل نبود اطلاعات قابل اطمینان از جزئیات مسیر آنها سعی می شود تا با احتیاط بیشتری مسیر برآورد شود.

مراجع

- [1] Okada, S., & Soper, T. (2000). "A shortest path problem on a network with fuzzy arc lengths." *Fuzzy Sets and Systems*. Vol.109, No.1, PP. 129-140.
- [2] Dubois, D. J. (1980). "Fuzzy sets and systems: theory and applications" (Vol. 144). Academic press.
- [3] Kristianto, Y., Gunasekaran, A., Helo, P., & Hao, Y. (2014). "A model of resilient supply chain network design: A two-stage programming with fuzzy shortest path." *Expert Systems with Applications*. Vol.41, No.1, PP. 39-49.
- [4] Malek, M. (2010). "Describing, modeling and topological relationships of fuzzy regions, using intuitionist fuzzy theory." *Remote Sensing & GIS*. Vol.2, No.3, PP. 63-74.
- [5] Zadeh, L. A. (1965). "Fuzzy sets." *Information and control*. Vol.8, No.3, PP. 338-353.
- [6] Klein, C. M. (1991). "Fuzzy shortest paths." *Fuzzy Sets and Systems*. Vol.39, No.1, PP. 27-41.
- [7] Lin, K. C., & Chern, M. S. (1993). "The fuzzy shortest path problem and its most vital arcs." *Fuzzy Sets and Systems*. Vol.58, No.3, PP. 343-353.

- [8] Okada, S., & Gen, M. (1994). "Fuzzy shortest path problem." *Computers & industrial engineering*. Vol.27, No.1, PP. 465-468.
- [9] Ji, X., Iwamura, K., & Shao, Z. (2007). "New models for shortest path problem with fuzzy arc lengths." *Applied Mathematical Modelling*. Vol.31, No.2, PP. 259-269.
- [10] Hernandez, F., Lamata, M. T., Verdegay, J. L., & Yamakami, A. (2007). "The shortest path problem on networks with fuzzy parameters." *Fuzzy Sets and Systems*. Vol.158, No.14, PP. 1561-1570.
- [11] Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment." *Applied Soft Computing*. Vol.12, No.3, PP. 1231-1237.
- [12] Karunambigai, M. G., Rangasamy, P., Atanassov, K., & Palaniappan, N. (2007). "An intuitionistic fuzzy graph method for finding the shortest paths in networks." In *Theoretical Advances and Applications of Fuzzy Logic and Soft Computing* (pp. 3-10). Springer Berlin Heidelberg.
- [13] Gani, A. N., & Jabarulla, M. M. (2010). "On searching intuitionistic fuzzy shortest path in a network." *Applied Mathematical Sciences*. Vol.4, No.69, PP. 3447-3454.
- [14] Mukherjee, S. (2012). "Dijkstra's algorithm for solving the shortest path problem on networks under intuitionistic fuzzy environment." *Journal of Mathematical Modelling and Algorithms*. Vol.11, No.4, PP. 345-359.
- [15] Rangasamy, P., Akram, M., & Thilagavathi, S. (2013). "Intuitionistic fuzzy shortest hyperpath in a network." *Information Processing Letters*. Vol.113, No.17, PP. 599-603.
- [16] Chu, T. C., & Tsao, C. T. (2002). "Ranking fuzzy numbers with an area between the centroid point and original point." *Computers & Mathematics with Applications*. Vol.43, No.1, PP. 111-117.
- [17] Wang, Y. J., & Lee, H. S. (2008). "The revised method of ranking fuzzy numbers with an area between the centroid and original points." *Computers & Mathematics with Applications*. Vol.55, No.9, PP. 2033-2042.
- [18] Chen, S. H. (1985). "Ranking fuzzy numbers with maximizing set and minimizing set." *Fuzzy sets and Systems*. Vol.17, No.2, PP. 113-129.
- [19] Liou, T. S., & Wang, M. J. J. (1992). "Ranking fuzzy numbers with integral value." *Fuzzy sets and systems*. Vol.50, No.3, PP. 247-255.
- [20] Yu, V. F., & Dat, L. Q. (2014). "An improved ranking method for fuzzy numbers with integral values." *Applied Soft Computing*. Vol.14, PP. 603-608.
- [21] Cheng, C. H. (1998). "A new approach for ranking fuzzy numbers by distance method." *Fuzzy sets and systems*. Vol.95, No.3, PP. 307-317.
- [22] Choobineh, F., & Li, H. (1993). "An index for ordering fuzzy numbers." *Fuzzy Sets and Systems*. Vol.54, No.3, PP. 287-294.
- [23] Wang, J. Q., Nie, R. R., Zhang, H. Y., & Chen, X. H. (2013). "Intuitionistic fuzzy multi-criteria decision-making method based on evidential reasoning." *Applied Soft Computing*. Vol.13, No.4, PP. 1823-1831.
- [24] Atanassov, K. T. (1986). "Intuitionistic fuzzy sets." *Fuzzy sets and Systems*. Vol.20, No.1, PP. 87-96.
- [25] Atanassov, K. T. (1989). "More on intuitionistic fuzzy sets." *Fuzzy sets and systems*. Vol.33, No.1, PP. 37-45.
- [26] Atanassov, K. T. (1999). "Intuitionistic Fuzzy Sets: Theory and Applications." *Physica Verlag Heidelberg*.
- [27] Lakshmana Gomathi Nayagam, V., & Sivaraman, G. (2011). "Ranking of interval-valued intuitionistic fuzzy sets." *Applied Soft Computing*. Vol.11, No.4, PP. 3368-3372.