Universidad de Málaga

Doctorado en Ingeniería de Telecomunicación
Departamento de Tecnología Electrónica
Escuela Técnica Superior de Ingeniería de Telecomunicación

Methods and elements of graph theory and fuzzy logic for communication network management

PhD Thesis Dissertation

submitted by

Lissette Valdés Valdés

Supervisor: Prof. Dr. Gonzalo Joya Caparrós

April 14, 2022

AUTOR: Lissette Valdés Valdés

https://orcid.org/0000-0001-6174-137X

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña LISSETTE VALDÉS VALDÉS
Estudiante del programa de doctorado EN INGENIERÍA DE TELECOMUNICACIÓN de la Universidad de Málaga, autor/a de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: METHODS AND ELEMENTS OF GRAPH THEORY AND FUZZY LOGIC FOR COMMUNICATION NETWORK MANAGEMENT.

Realizada bajo la tutorización y dirección de GONZALO JOYA CAPARRÓS (si tuviera varios directores deberá hacer constar el nombre de todos)

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 14 de ABRIL de 2022

Fdo.: LISSETTE VALDÉS VALDÉS Doctorando/a	Fdo.: GONZALO JOYA CAPARRÓS Tutor/a
Fdo.: GONZALO JOYA CAPARRÓS Director/es de tesis	

Edificio Pabellón de Gobierno. Campus El Ejido.
29071
Tel.: 952 13 10 28 / 952 13 14 61 / 952 13 71 10
E-mail: doctorado@uma.es

To my father

ACKNOWLEDGMENTS

Firstly, both my advisor, Prof. Dr. Gonzalo Joya, and I want to express our thanks and appreciation to Prof. Dr. Sira Allende Alonso, Emeric professor at the University of Havana, who, from the beginning of my thesis, has participated in all its stages as a real co-director. I am honored that “Profe” Sira took me on as a student and continued to have faith in me over the years. She has been an ideal teacher and mentor, offering advice and encouragement with a perfect blend of insight and kindness. I am extremely grateful for her support and patience. Administrative problems have prevented her from being officially listed as such.

I would like to express my sincere gratitude to my advisor Prof. Dr. Gonzalo Joya Caparrós, for his continuous support during my Ph.D. study and related research, his patience, motivation, and immense knowledge. I could not have imagined having a better advisor for my Ph.D. study.

I would like to thank Dr. Alfonso Ariza Quintana for his valuable collaboration during the implementation necessary for the simulations provided in this thesis.

This work has been developed in the context of the different international cooperation projects between the University of Havana and the University of Malaga, sponsored by the Agencia Española de Cooperación Internacional para el Desarrollo A2/038418/11, D/030223/10 and D/023835/09. Finally, I want to thank the responsibles for the projects, Dr. Gonzalo Joya Caparrós, Dr. Héctor de Arazoza, and Aymée Marrero, for the trust they placed in me when proposing my membership in these projects.

My sincere thanks to the Department of Electronic Technology of the University of Malaga and the Department of Applied Mathematics of the Faculty of Mathematics and Computer Science at the University of Havana for the technical and human support provided in all these years of work.

Very special gratitude goes out to Prof. Dr. Martin Spiess and the rest of the Psychological Methods Department members at the University of Hamburg for all the knowledge and moral support and for letting me be part of their group.

And my biggest thanks to my parents, Eva and Juan Luis, for all their support, unconditional love and emotional support in my life. To my children Adriana Marie and Erik Luis, for filling my days of happiness. Last but not least, I would like to thank my husband Volker Wandmaker for his valuable contributions as Software Developer and showing his love with no limits, for constantly listening to me rant and talk things out during difficult times. To my family, I give everything, including this.

AGRADECIMIENTOS

En primer lugar, tanto mi tutor de tesis, el Prof. Dr. Gonzalo Joya, como yo, queremos expresar nuestro agradecimiento y reconocimiento a la Prof. Dra. Sira Allende Alonso, profesora Emerica de la Universidad de La Habana, quien desde el inicio de mi tesis ha participado en todas sus etapas como un verdadero cotutor. Me siento honrada de que la "profe" Sira me haya aceptado como su estudiante y haya confiando en mí a lo largo de los años. Ella ha sido una profesora y mentora ideal, ofreciendome valiosos consejos y aliento con una combinación perfecta de rigurosidad y amabilidad. Estoy muy agradecida por su apoyo y paciencia. Problemas administrativos le han impedido figurar oficialmente como mi tutora.

Quisiera expresar mi más sincero agradecimiento a mi asesor Prof. Dr. Gonzalo Joya Caparrós, por su continuo apoyo, su paciencia, motivación e inmenso conocimiento durante toda mi etapa como estudiante de doctorado. No podría haber imaginado tener un mejor asesor en mi investigación de doctorado.

Quisiera agradecer al Dr. Alfonso Ariza Quintana por su valiosa colaboración durante la implementación necesaria para las simulaciones proporcionadas en esta tesis.

Este trabajo se ha desarrollado en el contexto de los diferentes proyectos de cooperación internacional entre la Universidad de La Habana y la Universidad de Málaga, patrocinados por la Agencia Española de Cooperación Internacional para el Desarrollo A2/038418/11, D/030223/10 y D/023835/09. Finalmente, quiero agradecer a los responsables de los proyectos, Dr. Gonzalo Joya Caparrós, Dr. Héctor de Arazoza y Dra. Aymée Marrero, por la confianza que depositaron en mí al proponer mi membresía en estos proyectos.

Mi más sincero agradecimiento al Departamento de Tecnología Electrónica de la Universidad de Málaga y al Departamento de Matemática Aplicada de la Facultad de Matemática y Ciencia de la Computación de la Universidad de La Habana por el apoyo técnico y humano brindado en todos estos años de trabajo.

Un agradecimiento muy especial para el Prof. Dr. Martin Spiess y el resto de los miembros del Departamento de Métodos Psicológicos de la Universidad de Hamburgo por todo su conocimiento y apoyo moral, y por permitirme ser parte de su equipo.

Y mi mayor agradecimiento va hacia mis padres, Eva y Juan Luis, por todo su apoyo, amor incondicional y apoyo emocional en mi vida. A mis hijos Adriana Marie y Erik Luis, por llenar mis días de felicidad. Por último, pero no menos importante, me gustaría agradecer a mi esposo Volker Wandmaker, por sus valiosas contribuciones como desarrollador de software y por mostrarme su amor sin límites, por escucharme siempre y conversar conmigo durante los momentos difíciles.

RESUMEN EN ESPAÑOL

El objetivo de esta tesis es el estudio de la aplicabilidad de elementos de la teoría de grafos y lógica borrosa a diversas tareas de gestión en una red de comunicación.

En una red de comunicación es esencial el establecimiento de conexiones seguras entre los pares de nodos que garanticen la robustez de dicho sistema, dependiendo del objetivo de su operación (funcionamiento máximo, pérdida mínima o retraso de la información enviada, etc.). A la vez, es de crucial importancia garantizar un uso general adecuado de los recursos de la red. Debe realizarse la gestión de caminos en las conexiones teniendo en cuenta los cambios en la carga de tráfico, la duración del funcionamiento de los nodos, conexiones y servidores, y las posibles interrupciones que pueden ocurrir en una red. Por esta razón, el diseño, la construcción y la administración de la infraestructura de la red y plataformas de servicios son tareas vitales y constituyen un excelente reto para los expertos en esta área.

Dirigimos nuestro trabajo hacia dos grandes campos: las redes de comunicación y la lógica borrosa con sus dos áreas fundamentales (inferencia borrosa y propiedades aritméticas de los números borrosos). Además, definimos y tratamos cada problema abordado sobre la base de la teoría de grafos. A partir de esta visión, a las redes incluidas en nuestro desarrollo investigativo se les asocia un grafo, donde cada uno de sus componentes son interpretados como elementos de dicho grafo. Intentamos describir cada problema con un alto nivel de abstracción matemática y lo adaptamos a la aplicación de problemas ingenieriles. Estos problemas están relacionados a la selección del nodo servidor en una red Peer to Peer (P2P), la ruta óptima en una red de comunicación genérica teniendo en cuenta distintas métricas definidas en sus conexiones y, en un mismo sistema, la determinación de caminos arista-disjuntos entre clientes y servidores. Las técnicas que usamos son, principalmente, heurísticas ad hoc para un problema determinado y están basadas en la lógica borrosa.

En este sentido llevamos a cabo tres estudios:

Capítulo 3 La selección del nodo servidor de acuerdo a un índice de bondad en el camino entre el servidor y el cliente en una red P2P.

Chapter 4 El análisis de la eficiencia de distintas funciones de costos en las conexiones de una red de alta capacidad para optimizar la carga de tráfico entre dos nodos servidor y cliente.

Chapter 5 La generación de un par de caminos arista-disjuntos usando costos borrosos en los enlaces en una red de alta capacidad en situaciones de sobrecarga.

En el esquema 0-1 resumimos el objetivo fundamental de nuestro trabajo investigativo, así como las diferentes áreas que conforman el trabajo de tesis.

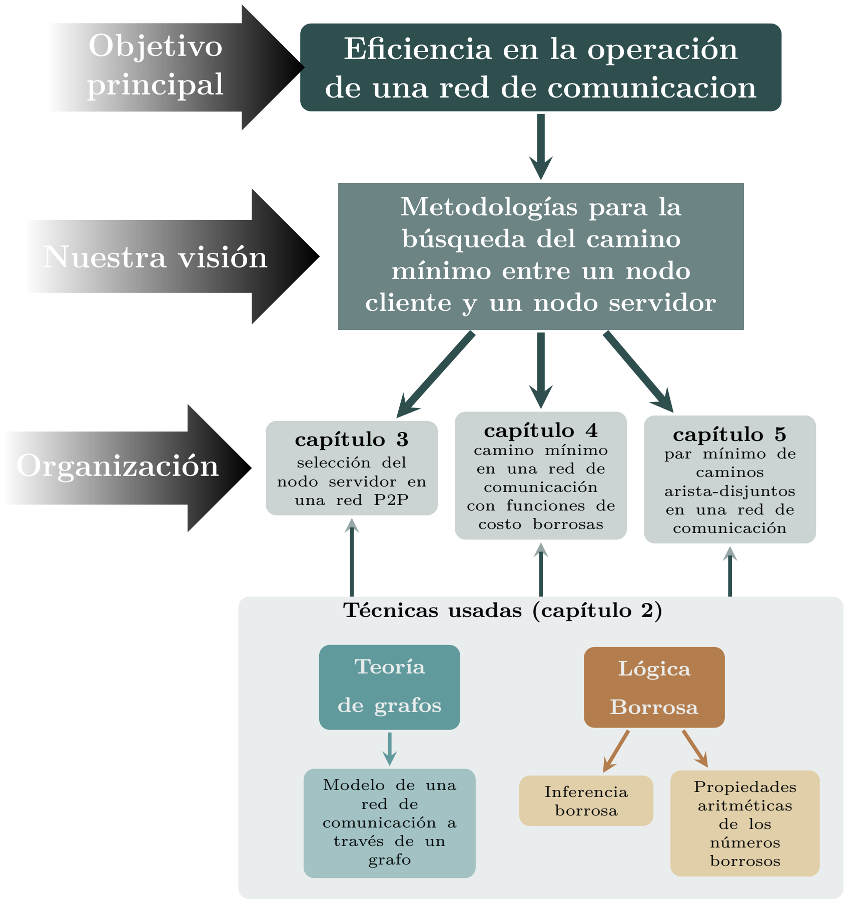


Figure 0-1: *Esquema general de nuestro desarrollo investigativo*

Nuestra intención es que los algoritmos y metodologías desarrolladas puedan aplicarse en la optimización de los recursos o el tiempo en diferentes procesos realizados por distintos tipos de redes. De esta forma, nuestras propuestas tendrían un impacto económico debido al ahorro de los resursos, así como a la reducción de la compra de dispositivos altamente costosos.

Capítulo 3. Algoritmo de Inferencia Borrosa aplicado a una red P2P

En una red P2P, los compañeros (peers) son relativamente autónomos y pueden unirse o dejar el sistema en cualquier momento. Esta es una red distribuida que, de manera usual, puede llegar a tener un gran número de compañeros debido al almacenamiento de datos, al procesamiento, y al ancho de banda de los compañeros autónomos.

Una de las limitaciones de las redes P2P en Internet radica en que la mayoría de los nodos no usan una dirección IP permanente. Con el propósito de seleccionar el nodo servidor, aplicamos un sistema de control borroso que brinda una solución para este problema.

La selección del nodo servidor en cada segmento de intercambio de información es una tarea esencial en redes inhalámbricas, donde los recursos del ancho de banda son muy limitados y se comparten entre todos los usuarios, conllevando al malgasto de recursos. Este es un problema especialmente importante en redes inhalámbricas multihop, debido a que, si un nodo que está distante hace función de cliente, la información debe viajar por un gran número de nodos e incrementar así la probabilidad de interferir con la transmisión de otros nodos.

Actualmente la estrategia más usada, en el caso donde todos los nodos son iguales, está basada en una *selección aleatoria* del nodo servidor entre aquellos a los que se les ha pedido la información. Esta estrategia es eficiente en algunas situaciones ya que nunca usa el mismo nodo como servidor todo el tiempo, sino que distribuye la carga aleatoriamente entre todos los nodos. Sin embargo, esta estrategia no siempre funciona eficientemente, ya que no considera algunos factores de la red como son: la longitud de cada camino, la carga de cada camino o el ancho de banda disponible.

Otro de los criterios para la selección del nodo servidor es la estrategia *Min-Hop*, la cual usa como métrica el número de saltos. Esta estrategia es bastante eficiente en redes homogéneas, donde el costo de cada salto se considera el mismo. Sin embargo, no considera la sobrecarga en algunos nodos, por tanto no siempre es eficiente en redes que no son homogéneas, como es el caso de redes con obstáculos.

Pensamos que una solución basada en la lógica borrosa para el problema de la selección del nodo servidor podría ser adecuada. Esto se debe a que al usar la lógica borrosa, se crea un compromiso entre los diferentes factores, cuyos efectos pueden ser evaluados en el intervalo $[0, 1]$ al aplicar *inferencia borrosa*. En nuestro análisis, los factores son *la calidad de los enlaces* y *la longitud del camino*. Los caminos más cortos son interesantes ya que con ellos se ahorran recursos que pueden ser usados en otros tráficos. Sin embargo, a la vez es tambien necesario que el camino pueda ofrecer una alta probabilidad de éxito en la transmisión de la información y así evitar retransmisiones futuras.

Nuestra propuesta es una versión mejorada del sistema borroso presentado en [Valdés et al. 2013]. Realizamos una comparación del funcionamiento de nuestro sistema con respecto a otras propuestas como son la selección aleatoria y la selección min-hop.

El algoritmo de inferencia borrosa que realiza nuestro sistema consta de cuatro componentes o módulos básicos:

1. Borrosificación de las variables de entrada
2. La base del conocimiento (base de reglas borrosas)
3. La toma de decisiones (maquinaria de inferencia borrosa)
4. Desborrosificación de la salida

Cada uno de los módulos antes mencionados así como la interconexión entre ellos se muestran en la figura 0-2.

Las variables de salida de nuestro sistema son las siguientes:

- Número de saltos (NHops)
- valor de la métrica “Expected Transmission Count” (ETX)

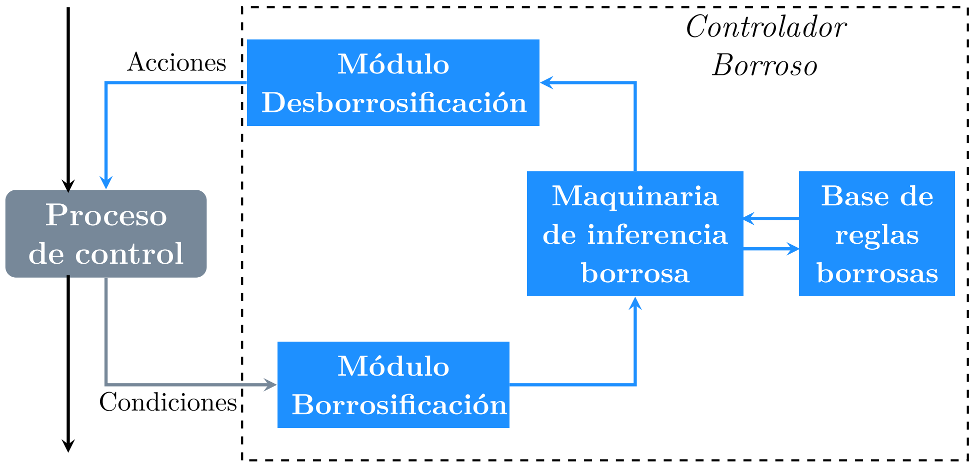


Figure 0-2: *Esquema general de un controlador borroso*

y la variable de entrada:

- Índice de bondad del camino entre el servidor y el cliente (GPath)

Definimos una base de reglas con 16 reglas del tipo "IF-THEN" con antecedentes compuestos mediante el conector "AND". Como método de implicación usamos el *Método de Mamdani* (o truncamiento) y como proceso de combinación empleamos el *valor máximo que resulta de la composición de las funciones de pertenencia de las salidas de cada regla*. Finalmente, usamos el *método del centroide* como el método de desborrosificación.

En nuestra etapa experimental, usamos una red inhalámbrica que está basada en la extensión mallada inhalámbrica que existe en IEEE 802.11-2012, donde los mecanismos de ruteo y envío son implementados en el nivel del enlace. El ambiente experimental en todos los experimentos realizados consiste en una red cuadrada regular 8×8 (64 nodos). Sobre esta red simulamos las estrategias de *selección aleatoria*, *Min-Hop* e *Inferencia borrosa*, esta última propuesta por nosotros. Analizamos dos escenarios de transmisión diferentes: la red sin contener obstáculos y la red con obstáculos. En cada simulación consideramos tres parámetros:

- tiempo de descarga que cada nodo emplea para obtener todos los segmentos de la información (valores promedio y máximo),
- el número de bytes enviados por nodo a nivel de aplicación (valores promedio y máximo),
- y el número de nodos enviados por nodo a nivel de red (valores promedio y máximo).

Basándonos en estos parámetros, nuestro sistema de inferencia borrosa es comparado con las otras dos estrategias.

En la red sin obstáculos, la estrategia de selección aleatoria es la menos eficiente, tanto en lo que respecta al tiempo de descarga requerido como a la carga de tráfico a nivel de red. Debido a que usamos una red regular, no hay diferencias importantes entre Min-Hop y nuestra estrategia, ya que Min-Hop es muy eficiente en este tipo de redes. Por lo tanto, el impacto de la lógica borrosa no puede mostrarse completamente en este experimento. Por otro lado, en la red con obstáculos, nuestro sistema de inferencia borrosa produce los mejores resultados con respecto al tiempo de descarga de un nodo. Además, en este escenario, la estrategia Min-Hop es la menos eficiente, ya que no considera el estado real de la red, sino sólo el número de saltos.

Capítulo 4. Búsqueda del camino mínimo en una red de comunicación usando funciones de costo borrosas

Un sistema de gestión de una red de comunicación toma las mediciones de sus variables de estado en instantes específicos de tiempo, considerándolas constantes en el intervalo entre dos mediciones consecutivas. De manera específica, debido al comportamiento dinámico de una red de telecomunicación, su sistema de gestión usualmente calcula los valores de las variables de estado de la red en un intervalo de tiempo dado, a partir de las mediciones obtenidas en el intervalo de tiempo inmediatamente anterior. Generalmente, los protocolos de ruteo actualizan el estado (costo) de los enlaces de dos formas posibles:

(A) Por intervalos de tiempo: cada valor de costo (y por tanto las tablas de enrutamiento) se actualiza periódicamente con una periodicidad fija. Esta actualización es tanto a partir del valor del estado instantáneo al comienzo del período (el comienzo del nuevo intervalo) o del valor medio de los costos durante el intervalo de tiempo anterior.

(B) Por umbrales: Algunos umbrales de actualización se fijan para que la actualización de una variable de estado se realice cuando la diferencia entre su valor real actual y el último determinado supera el umbral correspondiente.

Sin embargo, los métodos de actualización (A) y (B) introducen una incertidumbre en las mediciones de los valores de costo, ya que existe una probabilidad real de que las variables de estado cambien de un intervalo de tiempo a otro.

Para enfrentar este problema, consideramos el uso de elementos de la lógica borrosa para introducir la incertidumbre que existe en la red. Para ello, modelamos la red de comunicación como un grafo borroso de tipo V, donde los nodos y los enlaces se describen con precisión, pero los costos en los enlaces se asumen como números borrosos triangulares. En particular, para representar los costos hemos usado la variable *Ancho de Banda Usado Normalizado* [Ariza 2001].

Hemos propuesto un algoritmo de Dijkstra borroso (FDA en inglés). Este algoritmo encuentra el camino más corto entre dos vértices en un grafo borroso de tipo V, donde los costos en las aristas son números borrosos triangulares. Para la comparación de los costos borrosos que realiza el algoritmo, aplicamos el método de ranking propuesto en [Yu and l. Q. Dat 2014].

Hemos utilizado una red de 56 nodos basada en la topología de la red troncal de "Nippon Telegraph and Telephone (NTT)". En nuestra aplicación, se supone que todos los enlaces tienen la misma capacidad de 1 Gb/s. Las conexiones utilizan la conmutación de etiquetas multiprotocolo (MPLS) [Rosen et al. 2001] de modo que, una vez que se selecciona la ruta entre dos nodos, esta permanece sin cambios durante el tiempo de conexión. Por otro lado, las conexiones se realizan sin reserva de recursos, lo que significa que la conexión nunca se rechaza, pero puede haber una pérdida puntual de información en los enlaces cuando se excede su capacidad.

En nuestro escenario de simulación, los nodos de origen y destino se seleccionan aleatoriamente (con igual probabilidad) entre todos los nodos de la red. Por lo tanto, hay varios pares activos de origen y destino simultáneamente. Nuestra magnitud de interés en la red es el *número total de bytes enviados y recibidos* por cada nodo en cada intervalo de simulación. A partir de los valores anteriores calculamos las siguientes variables:

Tasa de entrega media, (MDR): Número total de bytes recibidos dividido por el número total de bytes enviados a lo largo de un experimento. Este valor indica la probabilidad de que finalmente se reciban los datos enviados.

Tasa de entrega global media, (GMDR): Media de la MDR en diez repeticiones del experimento.

Intervalo de confianza: Calculado con una probabilidad de 0.95.

Para comparar la eficiencia del rendimiento de la red basada en lógica borrosa frente a la basada en valores no borrosos, implementamos las funciones y estrategias de costos más comúnmente utilizadas en la gestión de redes reales basadas en valores no borrosos (crisp). Comparamos las estrategias crisp con otras similares basadas en nuestra definición de costos borrosos. En particular, las funciones de costo usadas son las siguientes:

- Ancho de Banda Usado Normalizado Instantáneo .
- Ancho de Banda Usado Normalizado Medio.
- Ancho de Banda Residual Medio.

Para cada una de las funciones de costo anteriores, hemos propuesto una variante borrosa, donde se define la métrica como un número triangular borroso.

Estudiamos ocho estrategias donde aparecen las versiones borrosas o crisp de las funciones de costo antes nombradas. Estas son:

Estrategia 1: Aplicación del algoritmo de Dijkstra clásico usando el Ancho de Banda Usado Normalizado Instantáneo en los enlaces. Para el cálculo del costo total del camino encontrado por el algoritmo, usamos la suma del costo de los enlaces que conforman dicho camino.

Estrategia 2: Aplicación del algoritmo de Dijkstra clásico usando el Ancho de Banda Usado Normalizado Medio como función de costo en los enlaces.

Estrategia 3: Aplicación del FDA usando el *Ancho de Banda Usado Normalizado Borroso* como función de costo. Esta estrategia es directamente comparable con las estrategias 1 y 2.

Estrategia 4: Aplicación del algoritmo "Shortest-Widest (SW)" [Wang and Crowcroft 1996], usando el Ancho de Banda Residual Medio y el algoritmo de Dijkstra clásico para encontrar el camino más corto.

Estrategia 5: Aplicación del algoritmo "Fuzzy Shortest-Widest (FSW)" usando el *Ancho de Banda Residual Borroso* y el FDA para encontrar el camino más corto. Esta estrategia es directamente comparable con la estrategia 4.

Estrategia 6: Aplicación del algoritmo "Widest-Shortest, (WS)" [Guerin et al. 1997], usando el Ancho de Banda Residual Medio y el algoritmo de Dijkstra clásico para encontrar el camino más corto.

Estrategia 7: Aplicación del algoritmo "Fuzzy Widest-Shortest, (FWS)" usando el Ancho de Banda Residual Borroso y el FDA para encontrar el camino más corto. Esta estrategia es directamente comparable con la estrategia 6.

Estrategia 8: Puede verificarse que mientras menor sea el ancho de banda usado en un intervalo de tiempo $(n-1)$, mayor es el grado de incertidumbre en las mediciones del ancho de banda usado del enlace que se realizan en el intervalo de tiempo n. Por tanto, hemos redefinido el Ancho de Banda Usado Normalizado Instantáneo Borroso al multiplicar este por un coeficiente crisp. De esta manera hemos propuesto el *Ancho de Banda Usado Normalizado Borroso Modificado* y hemos aplicado la estrategia 3 usando este como función de costo.

La gráfica 0-3 muestra los resultados de la GMDR para cada una de las estrategias, así como sus respectivos intervalos de confianza.

Basado en el comportamiento de las estrategias que muestra la gráfica 0-3, resumimos

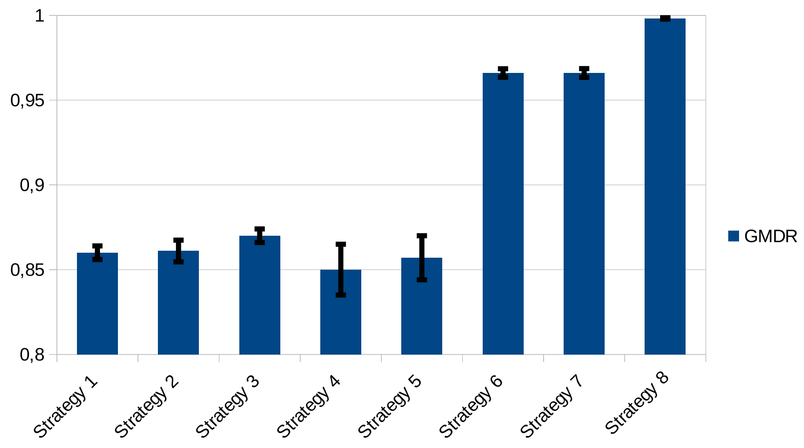


Figure 0-3: *Diagrama de barras de la GMDR para cada estrategia*

a continuación los resultados obtenidos una vez realizada nuestra experimentación:

(i) Los resultados muestran que la Estrategia 3 (borrosa) supera levemente a sus análogas crips (estrategias 1 y 2) pero aún así de manera estadísticamente significativa. El GMDR de la Estrategia 5 (fuzzy) supera a su equivalente borrosa

(Estrategia 4), pero en este caso sus intervalos de confianza se superponen. Por lo tanto, consideramos ambos resultados estadísticamente iguales. Las estrategias 6 y 7 tienen los mismos valores de GMDR con intervalos de confianza prácticamente iguales. Además, dado que estas estrategias buscan caminos con el mínimo número de saltos, contribuyen a un menor flujo de información a través de la red. Por tanto, las estrategias 6 y 7 se benefician del bajo consumo de recursos en comparación con las Estrategias 1 a 5.

(ii) La Estrategia 8, propuesta por nosotros, es la más eficiente con una clara ventaja sobre el desempeño del resto de estrategias, al obtenerse un valor de GMDR cercano a 1. Esta ventaja proviene de la definición de una nueva función de costo borrosa que incorpora nuestro conocimiento empírico acerca de los efectos de la incertidumbre en la medición.

Capítulo 5. Búsqueda del par mínimo de caminos arista-disjuntos en una red de comunicación. Visión borrosa.

La capacidad de supervivencia de las redes de comunicación es extremadamente importante, debido a los diferentes servicios que las redes proporcionan a la sociedad y la economía. La capacidad de supervivencia se puede definir como la capacidad de la red para respaldar la Calidad de los Servicios (QoS) comprometida de forma continua en la presencia de varios escenarios de fallos. Relacionado con la capacidad de supervivencia se encuentra el concepto de “Self-Healing”, donde, en una situación de saturación, el tráfico entre dos nodos puede organizarse dividiéndose éste entre dos caminos alternativos. Esto reduciría las condiciones de saturación y mejoraría el ancho de banda de ambos caminos. Por otro lado, la búsqueda de caminos alternativos mejora la seguridad de la comunicación de las fuentes de prioridad, ya que le resulta más difícil a elementos externos capturar un mensaje completo. Para resolver estos problemas, hemos propuesto una estrategia para encontrar el par mínimo de caminos, disjuntos en sus enlaces, entre dos nodos servidor y cliente, considerando la incertidumbre presente en la red.

No nos interesamos únicamente en que los caminos que forman el par no tengan enlaces comunes, sino tambien nos proponemos que la suma de sus costos sea la mínima. De esta manera, cuando ambos caminos sean empleados al mismo tiempo, estaríamos optimizando el costo de enviar la información y contribuyendo a la reducción de la saturación de la red.

Al igual que en el capítulo 4, asociamos la red a un grafo borroso de tipo V, cuyos vértices y aristas corresponden a los nodos y enlaces de la red, respectivamente. Nos enfrentamos a la incertidumbre en el sistema de gestion de la red al considerar el costo de las aristas como números borrosos triangulares. Basándonos en el método de ranking propuesto por [Yu and l. Q. Dat 2014], establecemos una condición necesaria para que un camino sea el mínimo en un grafo borroso de tipo V.

El par mínimo de caminos aristas-disjuntos puede tener la estructura de cualquiera de los dos pares de caminos que aparecen en la ecuación 1,

$$\{S \times P',\, P_1 \times P_2\} \qquad (1)$$

donde S es el camino mínimo del grafo (el cual puede ser único o no), P' es un camino que no tiene ninguna arista en común con S y los caminos P_i ($i = \overline{1,2}$) tienen una o varias aristas en común con S. Hemos inferido una fórmula para el costo total del par de caminos, la cual usa el costo de ciertas aristas ("breaks" en inglés) que pertenecen al camino S pero no son parte de ninguno de los caminos P_1 ó P_2.

Hemos propuesto un algoritmo (con siglas FSPPA en inglés) que encuentra un par de caminos con cualquiera de las estructuras mostradas en la expresión 1. Este algoritmo encuentra primeramente el camino S en el grafo original mediante la aplicación del algoritmo FDA descrito en el capítulo 4. Posteriormente se hace una modificación en las aristas que son parte del camino S, por lo que se obtiene un nuevo grafo que presenta la misma borrosidad del original, pero es mixto. En particular, el nuevo grafo se define como:

- El conjunto de vértices del nuevo grafo coincide con el del grafo original.
- Las aristas del camino S en el grafo original se sustituyen por arcos dirigidos hacia el vértice servidor. El costo de estos arcos se define como el *número borroso complementario* del costo borroso de sus aristas correspondientes, el cual es un número borroso negativo. El resto de las aristas se mantienen invariantes.

Los costos de los arcos introducidos en el nuevo grafo son números borrosos triangulares negativos. Por lo tanto, al aplicar el FDA, podríamos caer en ciclos cuyo costo total sea un número borroso triangular negativo (ciclos negativos). Para la búsqueda del camino mínimo sobre el nuevo grafo, el FSPPA aplica una modificación del FDA llamado Algoritmo de Dijkstra Borroso Modificado (MFDA en inglés) tambien propuesto por nosotros, el cual se aplica sobre un grafo borroso de tipo V que presenta arcos con costos definidos como números borrosos triangulares negativos. Para la comparación de los costos totales de los caminos, el MFDA utiliza el método de ranking propuesto por [Yu and l. Q. Dat 2014]. Este método compara los números borrosos a través de la comparación de sus integrales totales, siendo este un valor crisp que depende de un índice, α, que representa el grado de optimismo al realizar la comparación. Por tanto, podrían existir valores de α para los cuales el MFDA no convergería. Es por ello que realizamos un análisis para encontrar el rango de valores que puede tomar este índice tal que no existan ciclos negativos en el nuevo grafo definido. Para un valor de α dentro del rango permitido, el MFDA encuentra el camino mínimo P_{aux} en el nuevo grafo. Una vez que se han obtenido los caminos S y P_{aux}, el FSPPA elimina las aristas comunes entre ambos, tambien conocidas como "breaks". De esta forma, quedan establecidos los caminos que forman el par mínimo de caminos arista-disjuntos $P_1 \times P_2$. Note que, de no haber "breaks" el par de caminos es de la forma $S \times P'$.

Como etapa experimental hemos realizado dos experimentos diferentes:

Experimento 1. (Validación del FSPPA): Realizamos el experimento 1 para validar al FSPPA. Este experimento es una adaptación de la experimentación que realizamos en el capítulo 4, en el cual nos hemos asegurado de que todos los nodos tengan al menos dos caminos alternativos. De esta manera, podemos aplicar nuestro algoritmo cuando buscamos de forma aleatoria la comunicación entre dos nodos cualquiera. En esencia, el experimento 1 consiste en aplicar el FSPPA para todas las posibles combinaciones de pares de nodos servidor y cliente en una red.

Debido al alto costo computacional al realizar una búsqueda exhaustiva, este experimento no se puede realizar en redes de gran dimensión. Sin embargo, esta prueba es suficiente para mostrar que nuestro algoritmo encuentra el par mínimo de caminos arista-disjuntos. Utilizamos para nuestra experimentación la red formada por las principales ciudades de Estados Unidos. Simulamos un tráfico en el que el sistema presenta condiciones de saturación muy altas. Los costos de los enlaces varían con el tiempo y dependen del tráfico en cada intervalo de tiempo. Usamos como la función de costo de cada enlace el Ancho de Banda Usado Normalizado Borroso Modificado definido en la estrategia 8 descrita en el capítulo 4. Realizamos una única repetición del experimento.

Experimento 2. (Búsqueda del par mínimo de caminos arista-disjuntos en una red de alta funcionalidad con tráfico prioritario usando costos borrosos): En este experimento usamos una red de 57 nodos inspirada en el Nippon Telegraph and Telephone Corporation (NTT) [Varga 2001]. Ajustamos la red original para que se pueda acceder a cada nodo por, al menos, dos caminos. La simulación está orientada al flujo, es decir, solo se simulan dos eventos: el inicio y el final de una "ráfaga". Si tuviéramos que simular el envío de la información como tal, no podríamos simular el envío utilizando pequeños paquetes de información. Por lo tanto, se establece una única llamada para enviar toda la información, y no se termina hasta que la operación se completa totalmente. En otras palabras, enviamos toda la información como si fuera un bloque, por lo que debemos observar el principio y el final del bloque o ráfaga. Además, el almacenamiento no se simula mediante colas en los nodos. Si un nodo no tiene la capacidad suficiente para transmitir la ráfaga, los datos se pierden hasta que haya espacio libre o la ráfaga finalice, en cuyo caso se perderá por completo. Por lo tanto, simulamos un sistema sin demoras. En consecuencia, la variable que determinará la calidad de la red será *la tasa de entrega de bytes*, de forma tal que sabremos si los datos se pierden o no.

El tiempo de simulación ha sido de 10^4 segundos y todos los enlaces tienen la misma capacidad de 1 Gb/s. El tráfico se genera a través de llamadas con una conexión. Es decir, una vez que se establece una llamada, la ruta elegida no cambia durante toda la duración de la llamada. Estas llamadas no hacen reservas de recursos, por lo que el establecimiento de llamadas nunca se rechazará, pero puede haber una pérdida de datos en ellas. Esta restricción facilita la visualización de la pérdida de datos debido a la saturación de los enlaces. Al igual que en el experimento 1, usamos como indicador variable del costo de un enlace el Ancho de Banda Usado Borroso Normalizado Modificado.

Realizamos diez repeticiones por experimento con diferentes semillas y todos los flujos de información que se envían entre los nodos se realizan de acuerdo con distribuciones de probabilidad. Cada nodo puede tener dos tipos de fuentes de comunicación: la fuente F1 que siempre envía la información por la ruta más corta entre la fuente y los nodos de destino, y la fuente F2 (fuente de mensaje de prioridad) donde la información no solo puede enviarse por la ruta más corta entre los nodos fuente y destino (fuente F1), sino tambien la información se envía por las rutas que conforman el par de caminos que encuentra nuestro algoritmo FSPPA.

Medimos la calidad de la transmisión en la red mediante *el radio de entrega de bytes, (BDR en inglés)* definido como el radio entre los bytes entregados y enviados. La fuente F1 se implementa en todos los nodos, mientras que la fuente F2 solo se activa en un número limitado de nodos en cada experimento. Los experimentos se llevaron a cabo para grupos de 5, 10, 15, 20, 25 y 30 nodos con fuente F2. Para cada conjunto de nodos con la fuente F2, realizamos diez simulaciones en la que estos son escogidos aleatoriamente cada vez. Para los nodos con fuente F2, generamos un tráfico donde la información se envía a través del par de caminos encontrado por el FSPPA ó a través del camino mínimo encontrado por el FDA. En cada caso, el valor medio del BDR (MBDR) se calcula tanto para los nodos con fuente F1 como para los nodos con fuente F2. Posteriormente, comparamos el rendimiento de ambos tráficos generados.

La figura 0-4 muestra los resultados del experimento 2.

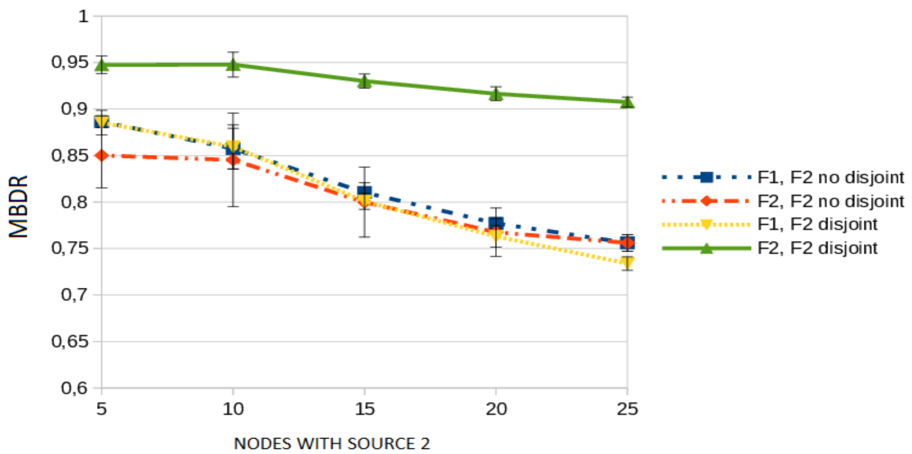


Figure 0-4: *Resultados de la simulación del experimento 2*

Línea azul: Muestra el MBDR correspondiente al tráfico creado por la fuente F1 cuando los nodos con la fuente F2 no aplican el FSPPA.

Línea amarilla: Corresponde al MBDR del tráfico de la fuente F1 cuando los nodos con la fuente F2 aplican el FSPPA.

Línea naranja: Corresponde al MBDR del tráfico creado por los nodos con la fuente F2 cuando estos no aplican el FSPPA.

Línea verde: Muestra la MBDR del tráfico creado por los nodos con fuente F2 cuando estos aplican el FSPPA.

Comparación entre las líneas azul y amarilla: En caso de haber pocos nodos con fuente F2, ambas líneas son muy similares. Sin embargo, a medida que aumenta el número de nodos con fuente F2, la línea amarilla se separa por debajo de la azul, es decir, los nodos con fuente F1 son afectados aún más debido al aumento del tráfico (alcanzan el valor inadmisible de 0.75 aproximadamente).

Comparación entre las líneas verde y naranja: Debido al tráfico adicional que enfrentan los nodos con la fuente F2, observamos la gran diferencia entre las líneas verde y naranja. El MBDR correspondiente a los nodos con fuente F2 cuando estos aplican el FSPPA (línea verde) se mantiene muy alto. Sin embargo, cuando estos nodos no aplican el FSPPA (línea naranja) su MBDR alcanza el valor inadmisible de 0.72.

Comparación entre las líneas azul y naranja: Ambas líneas se comportan de manera similar, aunque los nodos con fuente F2 disminuyen su MBDR cuando no aplican el FSPPA. Por otro lado, observamos que el tráfico generado por los nodos con fuente F2 (línea naranja), a medida que el número de estos aumenta, se acerca al tráfico creado por los nodos con fuente F1 (línea azul). Este es un comportamiento lógico ya que cuando no se aplica el FSPPA, cuanto mayor sea el número de nodos con fuente F2, más regular es el tráfico de la red.

Podemos decir que tener una estrategia donde la red tiene un número pequeño de nodos "privilegiados" con fuente F2, donde la información se envía mediante un par de caminos arista-disjuntos, es bastante interesante y efectiva. El algoritmo propuesto por nosotros proporciona una solución para que esta estrategia, de hecho, funcione.

Actualmente, en muchas redes de comunicación, los nodos que se seleccionan como "privilegiados" forman una red por separado solamente formada por ellos, significando esto un mayor costo de los recursos. Por lo tanto, otra ventaja de nuestra estrategia es la no necesidad de crear una red separada para lograr altos valores de la MBDR.

Como conclusión general, decimos que es cierto que en las redes actuales, las condiciones de sobrecarga no son habituales, puesto que los operadores sobredimensionan la red con un factor de seguridad elevado. Por otra parte, los métodos clásicos aplicados en estas condiciones resuelven satisfactoriamente los problemas planteados. Por tanto, nuestro objetivo no consiste tanto en confrontar la aplicación de técnicas borrosas con las actuales, sino en estudiar su viabilidad. En cualquier caso, hemos visto que estas resultan ser competitivas, ya que al menos en todos los casos hemos encontrado un método borroso que funciona igual o mejor que el método clásico.

Los sistemas de gestión de redes son muy conservadores en cuanto a los métodos y algoritmos empleados, y en la mayoría de los casos, las situaciones problemáticas están bien cubiertas con duplicidad de recursos. Sin embargo, considerar la exploración de otras técnicas alternativas, como las analizadas en este trabajo, puede ser interesante, no solo desde un punto de vista teórico y matemático, sino también práctico en futuros escenarios.

SUMMARY

This thesis aims to study the applicability of graph theory and fuzzy logic elements to various management tasks of a communication network.

We focus our work on two major fields: communication networks and fuzzy logic with their two fundamental aspects (fuzzy inference and arithmetic properties of fuzzy numbers). Also, every addressed problem is defined and treated based on *graph theory*. From this perspective, we include networks associated with graphs in our research, where we interpret each of its components as graph elements. We intend to describe each problem with a high mathematical abstraction level and specify them in engineering problems. They are related to selecting the server node in a P2P (Peer to Peer) network, the optimal route in a generic communication network considering different metrics defined on its links, and establishing the shorter edge-disjoint pair of paths between servers and clients. We apply mainly established ad hoc algorithms techniques for a given problem and fuzzy logic-based.

We develop and propose solution to three problems in chapters 3, 4, and 5. In the following, we summarize each of them:

Chapter 3: The server node selection according to a goodness index in the server-client path in a Peer to Peer (P2P) Network

We discuss the complexity of implementing our fuzzy inference algorithm by fuzzifying the input and output variables (in our case, given a server-node path, the inputs are the *number of hops* and the *Expected Transmission Count*, and the output is the *goodness index of the path*). We also examine the introduction of the inference rules, the inference engine, and the defuzzification of the fuzzy solution to convert it into a single crisp value.

We analyze two different transmission scenarios: a network without obstacles and a network with obstacles between nodes. We compare our strategy with the *Random Selection* of the server node, which is currently the most used, and the *Min-Hop* strategy, which chooses the path with the minimum number of hops.

The Random Selection strategy is the least efficient in a network without obstacles concerning the required transmission time and network-level traffic load. There are no crucial differences between Min-Hop and our approach in a regular system since the Min-Hop is very efficient in this kind of network, so the impact of fuzzy logic cannot be shown entirely in this experiment.

On the other hand, our strategy produces the best results concerning the download time for a node when obstacles are present. Also, in this scenario, the Min-Hop strategy is the least efficient because it does not consider the network's

actual state but only the number of hops.

Chapter 4: Analysis of the efficiency of different cost functions in a high capacity network links to optimize the traffic load between two nodes

A Communication Network Management System takes the measurements of its state variables at specific times, considering them constant in the interval between two consecutive measurements. Nevertheless, this assumption is not correct since these variables evolve in real-time. Therefore, uncertainty in measures cannot be efficiently managed using crisp variables or control based on fuzzy inference models. We face this problem by modeling the communications network as a type V fuzzy graph, where we defined the sets of nodes and links as crisp sets, but we modeled each link's cost as a triangular fuzzy number. We consider each cost function's crisp and fuzzy variant (fuzzy number) applied to the search for the shortest path between two nodes.

We based the optimal search strategies on a Dijkstra algorithm adapted to the fuzzy case where triangular fuzzy numbers are compared through their *Total Integrals*.

We perform an experimental study using a 56-nodes network based on the topology of the backbone network of Nippon Telegraph and Telephone Corporation as a reference. In the network, we calculate the total number of received bits divided by the total number of sent bits throughout an experiment (MDR). This value indicates the probability that sent data is finally received. We compare our functions and strategies with their crisp equivalents. We use for the comparison the Mean of the MDR in ten experiments. The results show that fuzzy strategy 3 surpasses their analogous crisp ones (strategies 1 and 2) slightly but in a statistically significant way. The GMDR of Strategy 5 (fuzzy) surpasses its crisp equivalent (Strategy 4), but their confidence intervals are overlapped. Thus we consider both results statistically the same. Strategies 6 and 7 have the same GMDR values with practically equal confidence intervals. Also, since these strategies search for paths with the minimum number of hops, they contribute to a lesser flow of information through the network. Therefore, Strategies 6 and 7 benefits from the low consumption of resources compared to Strategies 1 to 5. Finally, our new Strategy 8 (fuzzy), with a Global Mean Delivery Rate (GMDR) very close to 1, has a wide superior performance compared to the rest of the analyzed strategies.

Chapter 5 Search of the shortest pair of edge-disjoint paths using fuzzy costs in a high-performance network with priority traffic

The communication network's survivability is very important due to the systems' different services for society and the economy. Survivability can be defined as the network's ability to continuously support the committed Quality of Services (QoS) in the presence of various failure scenarios. The system must remain operational regardless of whether a failure occurs (in a node or a line). Related to survivability is the concept of Self-Healing, in which in a saturation situation, the traffic between two nodes can be organized by dividing it between two alternative paths. That would lower the saturation conditions and improve the bandwidth of both paths. It is not our only interest that both paths are link-disjoint, but also the sum of their costs is minimal. In this way, when both paths are used simultaneously, we would be optimizing the cost of sending the information and reducing the network's saturation. On the other hand, the search for alternative paths improves the communication security of priority sources since it is more challenging to capture a complete ordered message by external elements. The difference with finding the shortest and backup paths is that using both paths simultaneously (the pair of edge-disjoint paths whose total sum of their costs is the minimum) reduces the probability of information losses and increases the security (or privacy) of communications.

As in chapter 4, we associate the network to a type V fuzzy graph whose vertices and edges correspond to the network's nodes and links, respectively. We face uncertainty in the network's operating system by considering the edges' cost as triangular fuzzy numbers. First, we briefly provide the general configuration that the shortest pair of paths should have and define a formula to compute the pair's total cost. Then, we describe an algorithm that finds the shortest pair of edge-disjoint paths in the graph (FSPPA). The FSPPA uses a fuzzy adaptation of the Modified Dijkstra algorithm (MFDA) as a sub-algorithm. We can apply the MFDA in a mixed graph containing edges and some arcs whose costs are negative triangular fuzzy numbers.

To illustrate the algorithm's effectiveness, we apply FSPPA to a network with a high traffic load. We carry out ten replications per experiment with different seeds each, and all the information flows sent between the nodes are made according to probability distributions. Each node can have two types of communication sources: source F1 (regular sending of information), which always sends the data by the shortest path between the source and destination nodes, and source F2 (priority message source), which sends the data throughout the paths of the shortest pair of edge-disjoint paths. Note that a node with source F2 can apply the FSPPA and work as source F1. We measure the transmission quality in the network through the Packet Delivery Ratio (ratio between delivered and sent packets). Having a strategy with a small number of privileged nodes with source F2 is quite interesting and useful. The algorithm proposed by us provides a solution to make this strategy works.

Overload conditions are not common in current networks since operators oversize the system with a high safety factor. On the other hand, classical methods applied in these conditions effectively solve the problems posed. In any case, we have seen that these are competitive since, at least in all cases, we have found a fuzzy method that works the same or better than the classical ones. Therefore, this thesis aims to confront the application of fuzzy techniques with current ones and study their viability.

CONTENTS

ACRONYMS

The following table describes the significance of various acronyms used throughout the thesis. The page on which each one is defined or first used is also given.

Acronyms	Meaning	Page
AVCHD	Advanced Video Coding High Definition	81
BRR	Bandwidth Rejection Ratio	70
BDR	Bytes Delivery Ratio	3
CoA	Center of Area method	58
CoM	Center of Maximum method	58
ETT	Expected Transmission Time	70
ETX	Expected Transmission Count	48
FCM	Fuzzy Cognitive Maps	48
FDA	Fuzzy Dijkstra Algorithm	71
FIS	Fuzzy Inference System	50
FLC	Fuzzy Logic Controller	49
FMP	Fuzzy Modus Ponens	38
FSPPA	Fuzzy Shortest Pair of Edge-Disjoint Paths Algorithm	91
FSW	Fuzzy Shortest-Widest	79
FWS	Fuzzy Widest-Shortest	79
Gbps	Gigabits per sec	7
GMDR	Global Mean Delivery Rate	66
IP	Internet Protocol	7
LAN	Local Area Network	49
MBDR	Mean of the Bytes Delivery Ratio	120
Mbps	Megabits per sec	7
MDR	Mean Delivery Rate	83
MFDA	Modified Fuzzy Dijkstra Algorithm	113

Acronyms	Meaning	Page
MFI	Mamdani Fuzzy Implication	38
MoM	Mean of MAximum method	58
MPLS	Multi-Protocol Label Switching	75
NTT	Nippon Telegraph and Telephone Corporation	75
P2P	Peer to Peer	2
PPS	Packets per sec	7
OBS	Optical Burst Switching Network	81
QoS	Quality of Service	75
SINR	Signal to Interference and Noisy Ratio	48
SW	Shortest-Widest	69
TCP	Transmission Control Protocol	7
TISO	Two-inputs/single-output system	56
WAN	Wide Area Network	49
WS	Widest-Shortest	70

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

1.1 Problem Statement and Motivation

The proper and efficient functioning of a computer network, telecommunication network, and electrical power networks, among other factors, consists of the distribution of sensory, measurement, or generator devices. This fact has significant economic implications in the world nowadays. However, possible decision alternatives confront the randomness of data and, in general, the imprecision of data. For instance, the signal propagation's speed transmitted through a specific link may vary according to traffic density, external conditions, etc. On the other hand, there could be modifications in the network topology caused by natural, technological, or social factors.

In a communication network, it is essential to establish secure connections between pairs of nodes that guarantee such a system's robustness, depending on the objective of its operation (maximum performance, minimum loss or delay of sent information, etc.). At the same time, it is crucial to guarantee the proper general utilization of network resources. The management of paths in the connections between nodes must be carried out, taking into account:

- Slow and fast changes in the traffic load,
- Long and short lifetimes of nodes, links and servers, and,
- Short and long-term interruptions in the network, where many nodes or links may stop working simultaneously, or a critical server in the service platforms may stop functioning.

Therefore, the design, construction, and administration of network infrastructure and service platforms are vital tasks and represent a tremendous challenge for experts in this area.

The research, development, and application of new algorithmic methods and models for routing design in graphs under uncertainty and the possibility that the obtained results can be directly applied to different types of real networks is a scientific need due to its incredible relevance and importance. Thus, it is possible to use the resulting solutions and methodologies to resolve various applications in various science areas through fundamental mathematical research. It is worth mentioning that the search for solving large systems' optimization problems considering uncertainty is not a much-explored area. We base our motivation for writing this thesis on these aspects, whose primary goal, expressed in mathematical terms, is the study of heuristics and computational intelligence techniques for solving the routing in graph-defined problems.

Recently, many scientists have focused on searching for efficient algorithms for solving routing problems and optimal network design, assuming complete information about network data. The computational complexity of these problems already imposes a challenge to searching for efficient and robust solutions in problems with large dimensions. During my research, my attention has been addressed to obtain a methodology to deal with routing problems defined on networks, considering uncertainty in the information and network topology; and formulating the corresponding models and algorithmic proposals to solve these problems.

The term *fuzzy* seems to have been first introduced in [Zadeh 1962]. Zadeh expressed that "we need a radically different kind of mathematics, the mathematics of fuzzy or cloudy quantities which are not describable in terms of probability distributions. Indeed the need for such mathematics is becoming increasingly apparent even in the realms of inanimate systems". This paper was followed in 1965 by a technical exposition of such mathematics, now termed the theory of *fuzzy sets*, [Zadeh 1965]. The reasons supporting the representation of inexact concepts by fuzzy sets have been given by [Goguen 1967]. Perhaps his most convincing argument is a Representation Theorem, which states that any system satisfying certain axioms is equivalent to a system of fuzzy sets. Since the axioms are intuitively plausible for the system of all inexact concepts, the theorem allows us to conclude that fuzzy sets can represent vague concepts. Moreover, the Representation Theorem is a precise mathematical result in the Theory of Categories so that an exact meaning is given to the concepts "system", "equivalent", and "represented".

1.2 Contribution of the thesis

We focus our work on two major fields: communication networks and fuzzy logic. We treat two fundamental aspects of fuzzy logic: fuzzy inference and arithmetic properties of fuzzy numbers. Also, every addressed problem is defined and treated based on *graph theory*. From this perspective, we include networks associated with graphs, where we interpret each of its components as the elements of the graph. We intend to describe each problem with a high mathematical abstraction level and specify them in applications to engineering problems. They are related to selecting the server node in a Peer to Peer network, the optimal route in a generic communication network considering different metrics defined on its links, and, in the same type of system, the establishment of edge-disjoint paths between servers and clients. We apply established ad hoc algorithms for a given problem and fuzzy logic-based techniques.

In general, we deal with three problems:

Problem 1 - In a Peer to Peer network (P2P), we intend to choose the server node according to a goodness index in the server-client path. As a strategy, we propose a fuzzy-inference-based system. First, we have physical measurements with well-defined values of the input variables. These values may have some errors in their measurement, but we do not consider them. Then, by creating fuzzy inference rules and applying the fuzzy engine existing in the system, we obtain precise values of the output variables. We use the *number of hops* and the *Expected Transmission Count* as input variables, and as output variable, the *goodness index of the server-client path*. Then, we show that our fuzzy-inference-based system produces better results than traditional strategies used as the "random selection" and the "minimum number of hops", especially when facing a network with obstacles between nodes.

Problem 2 - To minimize information loss in high saturation conditions, we intend to find the shortest path between the server node and each client node in a communication network. We use different magnitudes or metrics defined on the links, considering these as imprecise. In particular, we interpret the network's uncertainty from the fuzzy logic view, representing the cost of each link as a triangular fuzzy number. In particular, in a *backbone network of Nippon Telegraph and Telephone Corporation*, we analyze the behavior of different cost functions in their fuzzy and crisp versions. Also, we propose a new fuzzy cost function that incorporates how much increases the degree of uncertainty in the measurements of the used bandwidth in a time interval when these measurements in the previous time interval were small. In general, fuzzy solutions improve crisp solutions but obtain values in the same confidence interval. In particular, the solution provided by our new fuzzy cost function is superior to the others. In this problem, we highlight the design and application of a fuzzy version of the Dijkstra algorithm used to search for the shortest path in a type V fuzzy graph, where fuzzy numbers are compared through their Total Integrals.

Problem 3 - In a similar network under the same conditions as in problem 2, we focus on finding the pair of paths that disjoint in edges, whose total cost is the minimum. Our interest is not only that both paths are link-disjoint, but that the sum of their costs is minimal. Thus, when both paths are used simultaneously, the information's sending is optimized, and the network's saturation is reduced. Besides, using an alternative path in parallel for sending the information improves the communication security of priority sources. We propose an algorithm that finds the shortest pair of edge-disjoint paths in a graph associated with the network. This algorithm uses a modification of the Fuzzy Dijkstra Algorithm mentioned in problem 2. This new version of the Dijkstra algorithm finds the solution in a mixed graph containing edges and some arcs whose costs are negative fuzzy numbers. To illustrate our algorithm's effectiveness, we apply this to a network with a high traffic load. We simulate traffic, where each node can have two types of communication sources: source F1 (regular sending of information) and source F2 (priority sending of information). We measure the network's transmission quality through the Bytes Delivery Ratio (BDR), defined as the ratio between delivered and sent bytes. Our algorithm provides a very good solution in a network with a small number of nodes with privileged source.

We summarize the different areas and stages of our work in figure 1-1.

We intend that the developed algorithms and methodologies can be applied to optimize

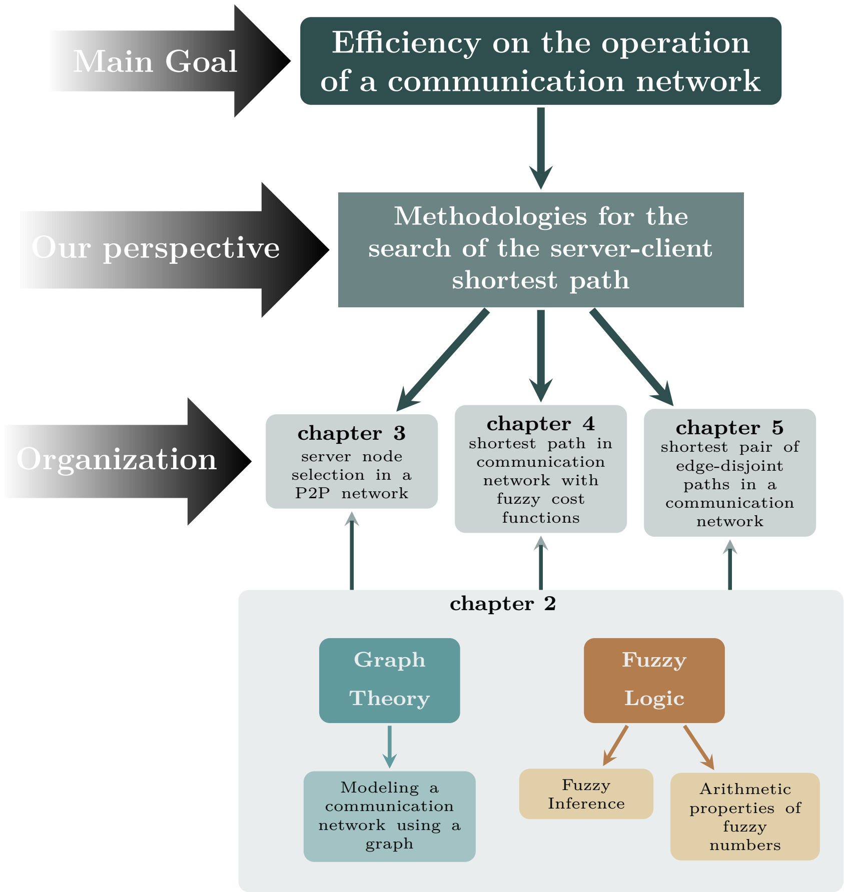


Figure 1-1: *General overview of the research development*

resources or time in different processes performed by different types of communication networks. In this way, our approaches would have an economic impact, leading to resource-saving and reducing the purchase of highly expensive measuring devices.

1.3 Thesis Structure

The structure of the thesis is as follows:

In Chapter 2, we provide a short review of the most interesting elements of the three conceptual fields involved in the thesis: communications networks, graph theory, and fuzzy logic. Thus, at first, we give a short overview of the main features of communication networks. In particular, we briefly describe the concept of the Internet Protocol (IP) network and the interpretation of its traffic. We further discuss the modeling of a communication network using graph theory elements, where we provide some concepts of this extensive area in mathematics. At last, we discuss the role of fuzzy logic for solving problems defined on networks. We provide some essential definitions and properties of the fuzzy inference and arithmetics of fuzzy numbers. This chapter may help facilitate the specific consultation of different concepts that appear throughout the thesis by those readers who may be experts on one of the topics mentioned above but not in others. Consequently, we consider that the reading of this chapter can be done independently of reading the central nucleus of the thesis, which is composed of chapters 3, 4, 5, and 6.

In Chapter 3, we propose a fuzzy logic-based system applied to select the server node in a P2P network. We discuss the complexity of implementing our fuzzy inference algorithm by fuzzifying the input and output variables, introducing the inference rules together with the inference engine, and the defuzzification of the fuzzy set given by the system as the solution. We further discuss the experimental environment to validate our approach.

In Chapter 4, we propose a fuzzy version of the Dijkstra algorithm to find the server-client shortest path in a communication network. We consider the costs of the links as triangular fuzzy numbers and describe different cost functions. To compare the efficiency of a communication network based on crisp and fuzzy costs, we give several strategies (crisp and fuzzy) to search the path between the server and client nodes with minimum cost. At last, we provide an experimental study where we perform a flow-oriented simulation. We show and discuss the simulation results using each strategy proposed.

In Chapter 5, we present a strategy to find the shortest pair of edge-disjoint paths between two fixed nodes in a communication network. Once again, we face uncertainty in the network's operating system by considering the cost of the links as triangular fuzzy numbers. At first, we describe the general structure of the shortest pair of edge-disjoint paths and give a formula for its total cost. Further, we formulate an algorithm that finds the pair of paths, which uses a modification of the Fuzzy Dijkstra Algorithm proposed in chapter 4, converging in a mixed graph containing arcs with costs defined as negative triangular fuzzy numbers. At last, in the experimentation stage, we perform the first experiment to verify our algorithm's effectivity and a second experiment designed to confirm if the parallel use of two edge-disjoint paths contributes to guaranteeing the quality of priority traffic in conditions of saturation.

Finally, in Chapter 6, we present the main conclusions and introduce some ideas for future work.

CHAPTER 2

PRELIMINARIES.

2.1 Introduction

From the initial overview of our work exposed in the introduction, we will provide this chapter with a certain level of common ground, the essential concepts of the three central topics to which we dedicate this thesis. Due to these topics being different between them, it is foreseeable that experts' knowledge in some areas is far from the others. Therefore, we intend to understand all the areas addressed by explaining concepts that an expert may find necessary with this chapter. These are: (a) central ideas about the topology and functioning of communication networks; (b) some elements of graph theory necessary for the modeling of a system; and (c) the essential concepts related to the Fuzzy Logic that are used for the different views of the uncertainty that we consider in the development of our research.

2.2 Communication network

Nowadays, people are increasingly connected, which gives us the feeling that the world is getting smaller. It is essential to long-distance communication if we want a planet to have significant coverage. Telecommunication has played a crucial role in establishing this world as *connected* and connecting telecommunication and data networks. Being connected has profound effects on the dissemination of information. How we are connected plays an essential role in the speed and robustness of such diffusion, among other issues.

Communication networks are part of all the networks that many people are aware of. What should immediately become clear is that networks are present in very different scientific disciplines: economics, organizational studies, social sciences, biology, logistics, among others.

Telecommunication networks were well established when people began to connect computers and create data communication networks. Of course, the many existing networks, such as telegram, have already made it possible to send data. Nevertheless, the new challenge was connecting these separate networks into a single one from a logical point of view, which computers could use using the same protocol. Thus, the idea arises by building a communication system where large messages are split into smaller units called *packets*. Each packet would be tagged with the address of its destination and subsequently routed through the various networks. Notice that packets from the same message could each follow their route to the destination, where they would then be finally used to reassemble the original message.

2.2.1 Internet Protocol Network

A *protocol* is a set of rules governing how things work in a particular technology to have some standardization ([Medhi and Ramasamy 2007]). Into the context of a communication network, an *Internet Protocol* (IP) network is a set of rules that govern how packets are transmitted over a network. An internet protocol describes how data packets move through a system. Along with addressing, *routing* is one of the main functions of the IP protocol. Routing consists of forwarding IP packets from source to destination machines over a network based on their IP addresses. An *IP address* is a unique address identifying a machine (a computer, a server, an electronic device, a router, a phone, etc.) on a network, thus serving for routing and forwarding IP packets from source to destination.

An IP network provides many services such as web and email. The Transmission Control Protocol (TCP) ensures reliability in a transmission. Thus, there is no packet loss, the packets are in the proper order, the delay is to an acceptable level, and there is no duplication of packets. When the TCP couples with IP, one gets the internet highway traffic controller. TCP and IP work together to transmit data over the internet but at different levels. All this ensures that the data received is consistent, in order, complete, and smooth. TCP bundles data into TCP packets before sending these to IP, which encapsulates these into IP packets. More detailed, an application's message content is broken into smaller TCP pieces, called TCP segments, transmitted over the IP network after including IP header information. The data entity at the IP level is *IP datagrams*, while a *packet* is also a commonly used term. Thus, traffic in an IP network is IP datagrams generated by various applications, without wondering which applications are for.

Traffic in IP Network

When we talk about *traffic volume* on an IP network link, we are interested in knowing the number of IP packets flowing on a link in a particular unit of time. Usually, the time unit is considered in seconds. Thus, we can specify the traffic volume in IP packets offered per second or packets per sec (PPS). On the other hand, another measure of traffic volume is often used: raw data rate units such as Megabits per sec (Mbps) or Gigabits per sec (Gbps).

A delay is a critical performance parameter in an IP network environment since we are interested in ensuring that a packet generated from one end reaches the other end as soon as possible. Interestingly, there is an analogy between road transportation networks and IP networks. In road transportation networks, the delay depends on the volume of traffic and the number of street lanes (and speed limit) imposed by the system. Similarly, the delay in an IP network depends on the amount of traffic and the system's capacity, i. e.

$$\text{Delay} = F(\text{Traffic volume data rate, Capacity})$$

Specifically, the above relation is valid only in a single-link system. However, when we consider a network where routing is also a factor, then a more general functional relation is shown in equation 2.1

$$\text{Delay} = F(\text{Traffic volume data rate, Capacity, Routing}) \tag{2.1}$$

2.3 Modeling a communication network

Understanding complex networks requires the right set of tools. For this purpose, we will be using elements of graph theory in our work.

Graph theory is a field in mathematics that gained popularity in the 19th and 20th centuries, mainly because it allowed the description of the phenomena from very different areas: communication infrastructures, scheduling tasks, social structures, etc., where there exists an interaction between elements of diverse nature. When we assign a graph to such a scenario, we represent the elements by nodes, the connection between them by edges and arcs, and the interaction of the elements by weights or costs. For instance, in a road network, the nodes represent the cities, the edges are the roads between every two cities, and the interaction or communication can be given by the distance or driving time for a car between cities. In a communication network, the nodes represent the signal emitting/receiving devices, the edge, or link, describes the communication channel (electromagnetic signals, fiber optic, etc.), and the weight in edges represent metrics such as the number of information packets sent from two specific sender and receiver devices. We can also model an epidemic by using a graph: nodes represent the beings affected by a disease (humans, animals, etc.), the edges can be the form of contact between these beings, and the interaction represents the contagion, which can have a determined probability of occurrence on each contact.

Using methods created on a graph representing a network makes the formulation of various problems defined in such network. For instance, when assigning a graph to a road network, a problem could be finding the shortest path between two cities in a road network or in a city itself. Furthermore, in an epidemic network having an assigned graph, it would be interesting to build a possible model of the evolution of such an epidemic, making it possible to isolate the initial focus or cut its development. Finally, and of great interest to us, when we assign a graph to a communication network, interesting problems would be finding the path between two nodes with the lowest loss of information and designing the network with the shortest number of nodes with certain restrictions on the traffic quality.

We will restrict our attention to communication networks. Therefore, in a graph $G = (V, E, \mathcal{C})$ assigned to a communication network, V and E are the set of nodes and links in the network, respectively; and $\mathcal{C}$ represents the set of costs assigned to the links. Each link has an associated cost of transferring one unit of information (for instance, the estimated time necessary for delivering an information packet) through it. The cost can vary from one link to another. Notice that we can define whether the graph is directed or undirected depending on the problem's nature. In a direct graph, the edge e_{ij} represents the directed link connecting node i to node j, whereas e_{ji} represents the directed link connecting node j to node i, so that e_{ij} is used to transfer the unit of information from node i to node j and e_{ji} is used to transfer the unit of information from node j to node i.

2.3.1 Necessary elements of Graph Theory

To provide a clear explanation of the studies given in our work, we need to use terminology that allows us to be precise. By adopting a "language" from graph theory, we will formulate statements such as distance between two nodes in a network, among others, accurately.

This section will refer to a few aspects of graph theory, giving some basic concepts and notations and fundamental properties that characterize networks.

Graph and vertex degrees

We have introduced a network that is represented mathematically by a *graph*. Using a formal notation, we define a graph in definition 2.3.1.

Definition 2.3.1. *Graph*
A graph G consists of a collection V of vertices and a collection of edges E, for which we write $G = (V, E)$. Each edge $e \in E$ is said to join two vertices called its endpoints. If the endpoints of an edge e are u and v, we write $e = (u, v)$ or $e = (v, u)$. Vertices u and v, in this case, are said to be ***adjacent****. Edge e is said to be* ***incident*** *with vertices u and v, respectively.*

Simply speaking, a graph is a collection of vertices that can be connected using edges. In particular, each edge of the graph joins exactly two vertices. We denote the number of vertices in graph G by $n = |V|$ and the number of edges by $|E|$. A graph for which every pair of distinct vertices defines an edge is called a *complete* graph.

In the remainder of this work, we will refer to the graph as defined in definition 2.3.1 with the following properties:

Simple graph: A graph that does not have loops or multiple edges.

Empty graph: An empty graph is a particular case where the graph has no vertices and, consequently, no edges.

A convenient definition in graph theory concerns *neighbors* of a vertex v.

Definition 2.3.2. *Neighbor Set*
For any graph $G = (V, E)$ and vertex $v \in V$, the neighbor set $\Gamma(v)$ of v is the set of vertices (other than v) adjacent to v, that is,

$$\Gamma(v) \stackrel{def}{=} \{u \in V | \, v \neq u, \, \exists e \in E : e = (u, v)\} \tag{2.2}$$

An essential property of a vertex is the number of edges that are incident with it. This number is called the *degree of a vertex*.

Definition 2.3.3. *Degree of a vertex*
In a graph $G = (V, E)$, the number of edges incident with a vertex v is called the degree of v and denoted as deg(v). Loops are counted twice.

Assuming $|V| = n$, the total sum of all the degrees satisfies the equation 2.3,

$$\sum_{i=1}^{n} \deg(v_i) = 2|E| \tag{2.3}$$

In a *regular* graph, every vertex has the same degree.

Simple Graph representation

There are different ways to represent graphs. When we consider their formal definition, graphs are described in terms of vertices and edges. One of the most important ways to represent a graph is using the *adjacency matrix*.

Definition 2.3.4. *Adjacency Matrix*
Consider a graph G with n vertices and m edges. The ***adjacency matrix*** *of G is a table A, with n rows and n columns with entry $A[i, j]$, denoting the number of edges joining the vertices v_i and v_j.*

Some important properties of this type of matrix are:

- A is *symmetric*, that is, for all i, j, $A[i, j] = A[j, i]$. This property reflects that an edge is represented as an unordered pair of vertices $e = (v_i, v_j) = (v_j, v_i)$.
- A graph G is *simple* if and only if for all i, j, $A[i, j] \leq 1$ and $A[i, i] = 0$. As we stated above, there can be at most one edge joining vertices v_i and v_j and, in particular, no edge joining a vertex to itself.
- the sum of values in row i equals the degree of vertex v_i, that is, $\delta(v_i) = \sum_{j=1}^{n} A[i, j]$.

The different representations of a graph (by its adjacency matrix, incidence matrix, or edge list) are independent of how we draw it. However, if we properly attach labels to vertices and edges, we will find their respective representations are the same. This similarity is formalized through the term *graph isomorphism*.

Definition 2.3.5. *Isomorphic graphs*
Consider two graphs $G = (V, E)$ and $G' = (V', E')$. G and G' are ***isomorphic*** *if there exists a one-to-one mapping $\phi : V \to V'$ such that for every edge $e \in E$ with $e = (u, v)$, there is a unique edge $e' \in E'$ with $e' = (\phi(u), \phi(v))$.*

Stated differently, two graphs G and G' are isomorphic if there is a one-to-one correspondence between the vertices of G and the vértices of G' such that the number of edges joining any two vertices in G is equal to the number of edges joining the corresponding two vertices in G'.

In many cases, checking whether two graphs are isomorphic is relatively simple as some crucial requirements need to be fulfilled. That is the case that the graphs have the same ordered degree sequence,

Theorem 2.3.1.
If two graphs G and G' are isomorphic, their respective ordered degree sequences should be the same.

For the proof of theorem 2.3.1, we refer the reader to [van Steen 2010].

Theorem 2.3.1 gives only a necessary condition for two graphs to be isomorphic. Thus, if two graphs have the same ordered degree sequence, that fact alone is insufficient to conclude that they are also isomorphic. In fact, there are no known easy sufficient conditions that tell us in general whether two graphs are isomorphic or not. Therefore, once we have found all necessary conditions have been fulfilled, we will have to fall into a trial-and-error method.

Connectivity.

The concept of *connectivity* in a graph is related to the availability of each vertex v to be reached from any other vertex u through a chain of adjacent vertices between them.

Definition 2.3.6. *Walk, Trail, Path, Cycle.*
*Given a graph G. A $(u, v)-$**walk** in G is a sequence $\langle u, e_1, v_1, e_2, v_2, \dots v_{k-1}, e_k, v\rangle$ of alternating vertices and edges from G with $e_i = (v_{i-1}, v_i)$. In a closed walk, $u = v$. A **trail** is a walk-in in which all edges are distinct; a simple **path** is a trail in which all vertices are also distinct. A **cycle** o circuit is a closed trail where all vertices except u and v are distinct.*

In a simple graph, we can specify more simply a path or cycle $\langle u, e_1, v_1, e_2, v_2, \dots v_{k-1}, e_k, v\rangle$ by the sequence of vertices $\langle u, v_1, v_2, \dots v_{k-1}, v\rangle$. Using the notion of a path, we define a graph as connected when there is a path between each pair of distinct vertices. Definition 2.3.7 formalizes the previous statement,

Definition 2.3.7.
In graph G, two distinct vertices u and v are connected if there is a path between them in G. G is connected if all pairs of distinct vertices are connected.

The notion of connectivity is essential, notably when considering the robustness of networks. *Robustness* means how well the system remains connected when removing vertices or edges. For example, we can see the internet as a (huge) graph in which routers form the vertices and communication links between routers the edges. In a formal sense, the internet is connected. However, if it were possible to partition the network into multiple components by removing only a single vertex (i.e., router) or edge (i.e., communication link), we could hardly claim the internet to be robust. This kind of network must sustain severe attacks and failures by which routers and links are brought down, so connectivity is still guaranteed. In general, there are many networks for which robustness plays a vital role in one way or another.

Subgraph

Definition 2.3.1 does not say anything that all vertices in a graph should be connected. Intuitively, this means that a graph could also consist of a collection of components, where each component is a connected subgraph.

Definition 2.3.8.
*A **subgraph** of G, denoted as H, is called a component of G if H is connected and not contained in a connected subgraph of G with more vertices or edges.*

We can obtain a proper subgraph H of G by removing a (nonzero) number of edges and (or) vertices of G. The removal of a vertex necessarily implies removing every edge incident to it. In contrast, removing an edge does not remove a vertex, although it may result in one (or even two) isolated vertices.

Tree

A tree T is a connected graph containing no cycles. A tree where one vertex r, the *root*, is distinguished is called a *rooted tree* $T(r)$. Any vertex of degree one in a rooted tree is called a *leaf* unless it is the root. Theorem 2.3.2 shows that there is one path between any two vertices of a tree, among other properties.

Theorem 2.3.2.
If T is a tree with n vertices, then,

(i) Any two vertices of T are connected by precisely one path.

(ii) For any edge e, not in T, but connecting two vertices of T, the graph $(T + e)$ contains exactly one cycle.

(iii) T has $(n - 1)$ edges.

The reader will find the detailed proof of theorem 2.3.2 in [Gibbons 1985].

These special types of graphs are essential to study for their standard and widespread use in diverse fields of practice and science. Typical examples of the application of trees are transportation networks, including communication and traffic networks. In many cases, we need to solve the problem of minimizing transportation costs from source to multiple destinations (or vice versa).

Trees play a prominent role in communication networks, whose main job is ensuring that messages are sent from their source to their intended destination(s), also referred to as *message routing*. How message routing is accomplished is laid down in a routing protocol: a collection of specifications describing what to do when a node in a network receives a message from source A sent to the destination node B. In general, a node in a communication network can be viewed as consisting of several interfaces, where each interface connects that node to precisely one other node in the system. Thus, we can represent a communication network as a graph with nodes represented as vertices and links between two nodes as edges. An interface is actually the endpoint of a link, and its representation coincides with the vertex representing the node to which that link is attached.

Directed graphs

It is natural to assign a *direction* to each graph's edge in some applications. For example,

- When we model a street plan as a network, the one-way streets are represented as a directed edge.
- In social networks, the "who knows whom" is also represented as a directed edge.
- In computer networks (especially wireless networks) where links between two different nodes are often not symmetric in the sense that messages can generally be successfully sent from station A to B, but not the other way around. Thus, modeling these connections is more conveniently done using directed edges.

When we make a diagram of such graphs, an arrow represents each edge. Such a graph is called a *directed graph* or *digraph*.

Definition 2.3.9. *Digraph*
A ***directed graph*** *or digraph D consists of a collection of vertices V, and a collection of directed edges or* ***arcs*** *$\mathcal{A}$, for which we write $D = (V, \mathcal{A})$. Each arc $a = (\overrightarrow{u, v})$ is said to join the vertex $u \in V$ to another (not necessarily distinct) vertex v. Vertex u is called the* ***tail*** *of a, whereas v is its* ***head****.*

If a digraph contains the edge (u, v), it may or may not contain the edge (v, u). In a *symmetric* digraph, there is an edge (v_j, v_i) for every edge (v_i, v_j). Every digraph D has an underlying (undirected simple) graph $G(D)$ obtained by replacing each arc $a = (\overrightarrow{u, v})$ with its undirected counterpart. Analyzing the underlying graph is often more convenient than directly considering the original digraph. Conversely, we can transform any undirected graph G into a directed one, $D(G)$, by associating a direction with each edge.

Weighted graphs

Besides adding a direction to an edge, we can also associate a *weight* with an edge, representing some *cost* or *distance*. The weight is a real number associated with an edge (or arc) in a graph. This extension is also natural in some applications where it is necessary to model real-world networks as graphs. Thus, the weights describe distances between vertices, travel time, links capacities, and in general, any metric measured in the link between the end vertices.

Definition 2.3.10.
A weighted graph $G = (V, \mathcal{E}, \mathcal{W})$ is a graph for which each edge $e \in \mathcal{E}$ has an associated number $w(e) \in \mathcal{W}$ called its weight. For any subgraph $H \subseteq G$, the weight of H is simply the sum of weights of its edges: $w(H) = \sum_{e \in E(H)} w(e)$

Shortest path

It is often that the interest lies on a path (or cycle), in which case it may be appropriate to refer to the length rather than the weight of the path (or cycle). However, it should not be confused with the length of a path (or cycle) in an unweighted graph which we defined earlier.

Among the uses of the weights, we can use them to determine the distance between two vertices, which is formally defined as follows.

Definition 2.3.11. *Distance, Shortest Path*
Consider an undirected graph G and two vertices $u, v \in V$. Let P be a (u, v)-path having minimal weight among all (u, v)-paths in G. The weight of P is known as the (geodesic) ***distance*** *$d(u, v)$ between u and v. Path P is called a* ***shortest*** $(\mathbf{u}, \mathbf{v})$***-path****, or a geodesic between u and v.*

Finding the shortest paths is a central problem in virtually all communication networks. Several efficient algorithms solve the well-known problem of finding the shortest path in a weighted graph. In this work, we will refer to one that stands out for its efficiency and simplicity.

Dijkstra Algorithm

There are many different variants for the Dijkstra algorithm. Although the original variant finds the shortest path between only two nodes r and t, a more common variant fixes a single node r as the "source" node and finds shortest paths from r to all other nodes in the graph, creating a shortest-path *tree* $T(r)$ that is said to be rooted at r. In general, using this algorithm for a different vertex yields a different rooted tree. Besides, there may be more than one shortest path between two vertices r and t. In other words, there may be several (r, t)-paths, all having the same minimal weight.

Dijkstra algorithm forms the core of many so-called *routing algorithms* used on the internet. This algorithm, created by the Dutch mathematician Edsger Dijkstra (1930 - 2002) in 1959, is undoubtedly one of the most important algorithms in modern communication networks. It is an efficient algorithm for finding the shortest path between a given pair of vertices. Also, this algorithm is valid for nonnegative graphs, and its efficiency is $O|V^2|$, where O denotes the order of complexity of the algorithm. The arrival of the Dijkstra algorithm was a significant development because most practical applications involve nonnegative graphs. Over time, researchers from different scientific areas have attempted to develop new algorithms, which are either modifications or improvements valid for specific types of graphs. Some of our approaches in this work are examples of what we previously stated.

Dijkstra algorithm solves the single-source shortest-paths problem on a weighted, directed graph $G = (V, E, \mathcal{W})$ for the case in which all edges weights are nonnegative $(w(u, v) \leq 0, \forall (u, v) \in E)$. Consider a vertex $r \in V$, and the set $S(r)$ of vertices whose shortest path from r has already been found. Given the vertex $x \in S(r)$, at each step, the algorithm analyzes the vertices of the neighbor set of x (definition 2.3.2) that do not belong to $S(r)$ yet. Among these vertices, the closest to r is picked and then added to $S(r)$. In other words, the algorithm repeatedly selects the vertex $u \in V - S(r)$ with the minimum shortest path estimate, adds u to $S(r)$, and relaxes all edges leaving u.

Algorithm 1 corresponds to the pseudo-code of the Dijkstra algorithm.

Algorithm 1 Dijkstra($\tilde{G}', r, t$)

$S(r) \leftarrow \{r\}$;
$d(r) \leftarrow (r, 0)$; $\forall v \in V : r \neq v$, **do** $d(v) \leftarrow (-, \infty)$;
while $S(r) \neq V$ **do**
  Select $u \in V - S(r)$ where $d(u)$ is minimal;
  $S(r) \leftarrow S(r) \cup \{u\}$;
  for $v \in \Gamma(u)$ **do**
    $d^{\text{new}}(v) := d(v) + w(u, v)$;
    if $d^{\text{new}}(v) < d(v)$ **then**
      $x(v) = u$ and $d(v) = d^{\text{new}}(v)$; ▷ $x(v)$: predecessor of v in the shortest path r-v path.
    end if
  end for
end while

2.4 The role of Fuzzy Logic

Fields of sciences, such as engineering, chemistry, or physics, construct exact mathematical models of empirical phenomena and then use them to make predictions. Nevertheless, some aspects of the "real world" always escape such precise mathematical models, and usually, there is an elusive inexactness as part of the original model.

Essentially, fuzziness is a type of imprecision that stems from grouping elements into classes that do not have sharply denned boundaries. Such classes, called *fuzzy sets*, arise, for example, whenever we describe ambiguity, vagueness, and ambivalence in mathematical models of empirical phenomena. Since certain aspects of reality always escape such models, the strictly binary or ternary approach to treating physical phenomena is inadequate to describe real-world systems. Besides, the attributes of the system' variables often emerge from an elusive fuzziness, a readjustment to context, or an effect of human imprecision.

In our research, we deal with some issues within the telecommunication engineering framework from the point of view of Fuzzy Logic. We intend to recreate a possible realistic performance environment. Thus, we consider the uncertainty present in the performance of a telecommunication network. For this, we focus our approaches mainly on two important areas of fuzzy logic theory: *Fuzzy inference* and *arithmetic properties of fuzzy numbers*. In particular, we try to give both a new interpretation and a more efficient solution to some problems related to the traffic and distribution of information in telecommunication networks (optical fiber, P2P, etc.) by using either the arithmetic properties of fuzzy numbers or elements of fuzzy inference.

2.4.1 Necessary Basic Concepts

In this section, we introduce some basic concepts and terminology of fuzzy sets that, either directly or indirectly, we will use in chapters 3 to 5. We refer to the latter as *crisp sets* to distinguish between fuzzy sets and classical (nonfuzzy) sets.

Fuzzy Set and Membership Function

Definition 2.4.1. *Universe of discourse*
A universal set (universe of discourse), A, is defined as a collection of objects having all the same features.

Intuitively, a *fuzzy set* is a class that admits the possibility of partial membership in it. Let A denote a space of objects. Then a fuzzy set $\tilde{A}$ in A is a set of ordered pairs,

$$\tilde{A} = \{(x, \mu_A(x)), x \in A\} \tag{2.4}$$

where $\mu_{\tilde{A}}(x) \in [0,1]$ is termed "the grade of membership of x in $\tilde{A}$".

The theory of fuzzy sets deals with a subset $\tilde{A}$ of the universe of discourse A, where the transition between full membership and no membership is gradual rather than abrupt. Moreover, the *fuzzy subset* has no well-defined boundaries where the universe of discourse covers a limited range of objects. These properties of fuzzy numbers can be used to model many aspects of human activity when we intend to classify a set of things or individuals into different categories -for example, size (high, medium, low); speed (slow, medium, fast, extra fast); human attitude (nice, indifferent, unpleasant).

A fuzzy set can be defined mathematically by assigning each possible individual in the universe of discourse a value representing its grade of membership in the fuzzy set. This grade corresponds to the degree to which that individual is similar or compatible with the concept represented by the fuzzy set. Thus, individuals may belong to a greater or lesser degree in the fuzzy set, as indicated by a larger or smaller membership degree. Definition 2.4.2 formalizes the above stated.

Definition 2.4.2. *Membership function of a fuzzy set*
Any fuzzy set $\tilde{A}$ defined on A has an associated membership function $\mu_{\tilde{A}}$ *that associates all elements of A with a value in the interval* $[0,1]$,

$$\mu_{\tilde{A}}: \ A \to [0,1] \tag{2.5}$$

Larger values of $\mu_{\tilde{A}}$ denote higher membership degrees to $\tilde{A}$. The most commonly used range of values of membership functions is the unit interval $[0,1]$. In this case, each membership function maps elements of a given universe of discourse A into real numbers in $[0,1]$. The membership function can also be generalized to take its values in intervals $[a,b] \in (0,1)$.

In general, any function $\mu: \ A \to [0,1]$ can be used as a membership function describing a particular fuzzy set. From a practical standpoint, the term "membership function" should reflect the problem from which the fuzzy set is defined.

α-cuts and support of a fuzzy set

Definition 2.4.3. *Normality*
The fuzzy set $\tilde{A}$ is said to be normal (normalized) if and only if its membership function fulfills the condition 2.6,

$$\max_{x \in A} \mu_{\tilde{A}}(x) = 1 \tag{2.6}$$

A non-normal and non-empty fuzzy set [1] *can be normalized by dividing each* $\mu_{\tilde{A}}(x)$ *by the term* $\max_{x \in X} \mu_{\tilde{A}}(x)$.

One of the most important concepts in fuzzy sets theory corresponds to the *α-cut* and its variant, the *strong α-cut*. Both concepts can be viewed as a bridge connecting fuzzy sets and crisp sets. When we want to exhibit an element $x \in A$ that typically belongs to a fuzzy set $\tilde{A}$, we may demand its membership degree to be greater than some threshold $\alpha \in [0,1]$.

Definition 2.4.4. *α-cut and strong α-cut sets*
The α-cut set of a fuzzy set $\tilde{A}$*, denoted by* $\tilde{A}_\alpha$*, is a crisp set that contains all the elements of A whose membership degree is greater than or equal to the specified value* α*, i.e.:*

$$\tilde{A}_\alpha = \{x \in A| \ \mu_{\tilde{A}}(x) \geq \alpha\}, \quad \alpha \in [0,1] \tag{2.7}$$

The strong α-cut of A, denoted as $\tilde{A}_{\alpha+}$*, is the crisp set containing all the elements of A with membership function greater than* α*,*

$$\tilde{A}_{\alpha+} = \{x \in A| \ \mu_{\tilde{A}}(x) > \alpha\}, \quad \alpha \in [0,1] \tag{2.8}$$

Let $\tilde{A}$ be a fuzzy set, and the real numbers α_1 and α_2 ($\alpha_1, \alpha_2 \in [0,1]$):

$$\text{If } \alpha_1 \leq \alpha_2 \ \Rightarrow \ \tilde{A}_{\alpha_2} \subseteq \tilde{A}_{\alpha_1}$$

An important special case of α-cut is the *support* of a fuzzy set.

Definition 2.4.5. *Support of* $\tilde{A}$
The support of fuzzy set $\tilde{A}$ *within a universal set A is the crisp set that contains all the elements of A that have nonzero membership degrees in* $\tilde{A}$*., i.e.:*

$$supp(\tilde{A}) = \{x \in A| \ \mu_{\tilde{A}}(x) > 0\} \tag{2.9}$$

Definition 2.4.6. *Level set of* $\tilde{A}$
The value $\alpha \in [0,1]$ *explicitly shows the value of the membership function. The* level set *of* $\tilde{A}$*, denoted as* Λ*, is defined as,*

$$\Lambda_{\tilde{A}} = \{\alpha \,|\, \mu_{\tilde{A}}(x) = \alpha, \ \alpha \in [0,1], \, x \in X\} \tag{2.10}$$

Convexity

Convexity is an essential property of fuzzy sets defined on $\mathbb{R}$ (for some $n \in \mathbb{N}$). It is a generalization of the classical concept of convexity of crisp sets.

The generalized convexity of fuzzy sets should be consistent with the classical definition. Therefore, it requires that the α-cuts of a convex set are convex for all $\alpha \in (0,1]$ in the classical sense (0-cut is excluded here since it is always equal to $\mathbb{R}^n$ in this case, and includes $-\infty$ to $+\infty$).

The definition of convexity for fuzzy sets does not mean that the membership function of a convex fuzzy set is convex. Instead, the membership functions of convex fuzzy sets are concave and not convex according to standard definitions.

Theorem 2.4.1 provides an alternative formulation of convexity of fuzzy sets restricted to fuzzy sets on $\mathbb{R}$.

[1] The fuzzy set $\tilde{A}$ is empty if and only if $\forall x \in A, \ \mu_{\tilde{A}}(x) \equiv 0$.

Theorem 2.4.1. *Convexity*
The fuzzy set $\tilde{A}$ *is convex if its membership function satisfies the following condition:* $\forall\, x_1, x_2 \in X$ *and* $\forall \lambda \in [0,1]$,

$$\boxed{\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2))}$$

where min *denotes the minimum operator.*

2.4.2 Set-Theoretic Operations for Fuzzy sets

The concepts suggested by Zadeh in [Zadeh 1965] constitute a consistent framework for the theory of fuzzy sets. The operations with fuzzy sets are defined via their membership function. Two of the most used and intuitive operators for modeling the intersection and union of fuzzy sets (also known as Zadeh's t-operators) are the *min* and *max* operators, respectively.

Definition 2.4.7. *Intersection of two fuzzy sets*
The membership function $\mu_{\tilde{C}}(x)$ *of the intersection* $\tilde{C} = \tilde{A} \cap \tilde{B}$ *(operator AND) is pointwise defined by,*

$$\mu_{\tilde{C}}(x) = \min\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\} = \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x), \quad \forall x \in X \tag{2.11}$$

Definition 2.4.8. *Union of two fuzzy sets*
The membership function $\mu_{\tilde{D}}(x)$ *of the union* $\tilde{D} = \tilde{A} \cup \tilde{B}$ *(operator OR) is pointwise defined by,*

$$\mu_{\tilde{D}}(x) = \max\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\} = \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x), \quad \forall x \in X \tag{2.12}$$

In [Bellman and Giertz 1973], the authors give an axiomatic justification about why *min* and *max* operators are preferably used over others that are also valid. Their statements are provided from a logical point of view, where the intersection is interpreted as "*logical and*", and the union as "*logical or*". Together with other operators that have also been suggested, they are part of the two basic and classical operators, referred to as triangular norms (*t-norms*) and conorms (*t-conorms*).

Definition 2.4.9. *t-norm*
The operator t *defined as,*

$$t: \; [0,1] \times [0,1] \rightarrow [0,1]$$

and, given $\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x), \mu_{\tilde{C}}(x) \in [0,1]$ *with* $x \in X$*, satisfies the conditions:*

i) $t(0,0) = 0,\; t(1,1) = 1,\; t(\mu_{\tilde{A}}(x), 0) = 0,\; t(\mu_{\tilde{A}}(x), 1) = \mu_{\tilde{A}}(x)$	:	*boundary condition*
ii) $t(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)) = t(\mu_{\tilde{B}}(x), \mu_{\tilde{A}}(x))$	:	*conmutativity*
iii) If $\mu_{\tilde{B}}(x) \leq \mu_{\tilde{C}}(x)$ *then* $t(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)) \leq t(\mu_{\tilde{A}}(x), \mu_{\tilde{C}}(x))$	:	*monotonicity*
iv) $t\,(t(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)), \mu_{\tilde{C}}(x)) = t\,(\mu_{\tilde{A}}(x), t(\mu_{\tilde{B}}(x), \mu_{\tilde{C}}(x)))$	:	*associativity*

is a t-norm.

Definition 2.4.10. *t-conorm*
The operator t^* *defined as:*

$$t^*: \; [0,1] \times [0,1] \rightarrow [0,1]$$

and, given $\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x), \mu_{\tilde{C}}(x) \in [0,1]$ with $x \in X$, satisfies the conditions:

i) $t^*(0,0)=0,\ t^*(1,1)=1,\ t^*(\mu_{\tilde{A}}(x),0)=\mu_{\tilde{A}}(x),\ t^*(\mu_{\tilde{A}}(x),1)=1$	:	*boundary condition*
ii) $t^*(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(x)) = t^*(\mu_{\tilde{B}}(x),\mu_{\tilde{A}}(x))$	:	*conmutativity*
iii) If $\mu_{\tilde{B}}(x) \leq \mu_{\tilde{C}}(x)$ *then* $t^*(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(x)) \leq t^*(\mu_{\tilde{A}}(x),\mu_{\tilde{C}}(x))$	:	*monotonicity*
iv) $t^*\left(t^*(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(x)),\mu_{\tilde{C}}(x)\right) = t^*\left(\mu_{\tilde{A}}(x),t^*(\mu_{\tilde{B}}(x),\mu_{\tilde{C}}(x))\right)$	:	*associativity*

is a t-conorm.

In particular, the intersection operator holds the conditions for t-norm, and the union operator holds the conditions for t-conorm.

2.4.3 Representation of Fuzzy Sets

The principal role of α-cuts and strong α-cuts is their capability to represent fuzzy sets. Each fuzzy set can uniquely be represented by the family of all its α-cuts or the family of all its strong α-cuts. These representations extend various properties and operations on crisp sets to their fuzzy counterparts. In each extension, a given crisp property or operation is required to be valid for each crisp set involved in the representation.

Let us first recall the definition of *Characteristic function* of the α-cut of a fuzzy set $\tilde{A}$.

Definition 2.4.11. *Characteristic function of $\tilde{A}_\alpha$*
Given the α-cut of the fuzzy set $\tilde{A}$. The application

$$\chi_{\tilde{A}_\alpha} : \tilde{A}_\alpha \to \{0,1\}$$

such that

$$\chi_{\tilde{A}_\alpha}(x) = \begin{cases} 1, & \text{if } x \in \tilde{A}_\alpha \\ 0, & \text{if } x \notin \tilde{A}_\alpha \end{cases} \tag{2.13}$$

is called the Characteristic function of $\tilde{A}_\alpha$.

The representation of a fuzzy set by its α-cut is universal, regardless of whether it is based on a finite or infinite universal set.

Definition 2.4.12. *Decomposition of $\tilde{A}$*
The decomposition of a fuzzy set $\tilde{A}$ is the representation of $\tilde{A}$ in terms of a special fuzzy set ${}_\alpha\tilde{A}$ which is defined in terms of the α-cuts of $\tilde{A}$ with membership function defined as:

$$\mu_{{}_\alpha\tilde{A}} = \alpha \cdot \chi_{\tilde{A}_\alpha}, \qquad \alpha \in [0,1] \tag{2.14}$$

Figure 2-1 shows an example of the special fuzzy set ${}_\alpha\tilde{A}$.

Theorems 2.4.2, 2.4.3, and 2.4.4 describe the different ways a fuzzy set can be decomposed based on the special fuzzy set ${}_\alpha\tilde{A}$. $\mathcal{F}(X)$ denotes the family of fuzzy sets having X as their universal set.

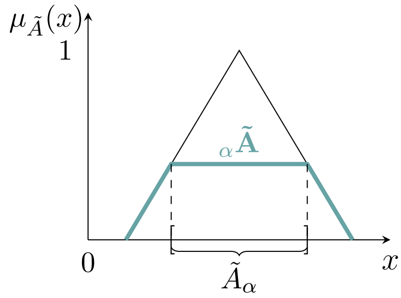


Figure 2-1: *Special fuzzy set* $_\alpha\tilde{A}$

Theorem 2.4.2. *First Decomposition Theorem*
For every $\tilde{A} \in \mathcal{F}(X)$,

$$\tilde{A} = \bigcup_{\alpha \in [0,1]} {}_\alpha\tilde{A}$$

where $_\alpha\tilde{A}$ *is defined by equation 2.14 and "*$\cup$*" denotes the standard fuzzy union.*

Theorem 2.4.3. *Second Decomposition Theorem*
For every $\tilde{A} \in \mathcal{F}(X)$,

$$\tilde{A} = \bigcup_{\alpha \in [0,1]} {}_{\alpha+}\tilde{A}$$

where $_{\alpha+}\tilde{A}$ *denotes a special fuzzy set defined by* $\mu_{\alpha+\tilde{A}} = \alpha \cdot \chi_{\tilde{A}_{\alpha+}}$.

Theorem 2.4.4. *Third Decomposition Theorem*
For every $\tilde{A} \in \mathcal{F}(X)$,

$$\tilde{A} = \bigcup_{\alpha \in \Lambda_{\tilde{A}}} {}_\alpha\tilde{A}$$

where $\Lambda_{\tilde{A}}$ *is the level set of* $\tilde{A}$.

2.4.4 Extension principle of Zadeh

The transformations of elements using functions are omnipresent. Generalizations of these transformations are those between points involving sets transformations between spaces and mappings of fuzzy sets between universes. Thus, point transformations can be expanded to cover transformations involving fuzzy sets. One of the mechanisms to transform fuzzy sets is the *extension principle of Zadeh*. The intuitive idea can be described as "Given a function that goes from a particular domain X to image Y, the extension principle provides a mechanism for transforming a fuzzy set defined on X in another fuzzy set defined in Y".

For an arbitrary number of sets $X_1, \ldots, X_n$, the set of all n-tuples $(x_1, \ldots, x_n)$ such that $x_1 \in X_1, \ldots, x_n \in X_n$ is called Cartesian product and is denoted by $X_1 \times \ldots \times X_n$.

Definition 2.4.13. *Extension Principle between two fuzzy sets*
Let $\mathcal{F}(X)$ and $\mathcal{F}(Y)$ be the family of all the fuzzy sets defined in the universes of discourse X and Y, respectively, and let $f : X \rightarrow Y$, be a crisp function. f is fuzzified when it is extended to act on fuzzy sets defined on X and Y. That is, the fuzzified function, for which the same symbol f is usually used, has the form

$$f : \mathcal{F}(X) \rightarrow \mathcal{F}(Y)$$

such that if $\tilde{A}$ is a fuzzy set in X, then its image under f is also a fuzzy set $\tilde{B} = f(\tilde{A}) = \{(y, \mu_{\tilde{B}}(y))\}$ in Y with membership function defined by equation 2.15,

$$\mu_{\tilde{B}}(y) = \max_{x|y=f(x)} \mu_{\tilde{A}}(x) \tag{2.15}$$

The Cartesian product of n fuzzy sets is also a fuzzy set. We formalize this concept in definition 2.4.14.

Definition 2.4.14.
Let X be the Cartesian product of universal sets, $X_1 \times X_2 \times \ldots \times X_n$, and $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n$ be n fuzzy sets in the universal sets $X_1, X_2, \ldots, X_n$, respectively. The Cartesian product of sets $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n$ leads to the fuzzy set $\tilde{A}_1 \times \tilde{A}_2 \times \ldots \times \tilde{A}_n$ on X defined by equation 2.16,

$$\mu_{\tilde{A}_1 \times \ldots \times \tilde{A}_n}(x_1, x_2, \ldots, x_n) = \min[\mu_{\tilde{A}_1}(x_1), \ldots, \mu_{\tilde{A}_n}(x_n)] \tag{2.16}$$

When a given function is defined on a Cartesian product, the extension principle is still applicable.

Definition 2.4.15. *Generalized Extension Principle*
Given the Cartesian product $\tilde{A}_1 \times \tilde{A}_2 \times \ldots \times \tilde{A}_n$ defined in 2.4.14 and let f be the function from $X_1 \times X_2 \times \ldots \times X_n$ to Y,

$$f(x_1, x_2, \ldots, x_n) : X_1 \times X_2 \times \ldots \times X_n \ \rightarrow \ Y$$

Then the fuzzy set $\tilde{B}$ in Y can be obtained by the function f and the fuzzy sets $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n$ as follows:

$$\mu_{\tilde{B}}(y) = \begin{cases} 0, & \text{if } f^{-1}(y) = \emptyset \\ \max_{(x_1,\ldots,x_n)\in f^{-1}(y)} \left[\min(\mu_{\tilde{A}_1}(x_1), \ldots, \mu_{\tilde{A}_n}(x_n))\right] & \text{if } f^{-1}(y) \neq \emptyset \end{cases} \tag{2.17}$$

where

- *f^{-1} is the inverse of f.*
- *$\mu_{\tilde{B}}(y)$ is the membership degree of $y = f(x_1, \ldots, x_n)$ with $(x_1, \ldots, x_n)$ having membership function $\mu_{\tilde{A}_1 \times \tilde{A}_2 \times \ldots \times \tilde{A}_n}(x_1, \ldots, x_n)$.*

Fuzzy sets as $\tilde{B}$ defined in definition 2.4.15 are referred to as *fuzzy relations*.

2.4.5 Fuzzy Relations

A crisp relation represents the presence or absence of association, interaction, or interconnection between elements of two or more sets. Relations can be generalized to allow the presence of various degrees of association or interaction between elements. Membership degrees can represent these degrees in a fuzzy relation in the same way as degrees of set membership are represented in the fuzzy set. A fuzzy relation generalizes a classical relation to one that allows partial membership and describes a relationship that holds between two or more objects.

Definition 2.4.16. *Fuzzy Relation*
A fuzzy relation is a fuzzy set defined on the Cartesian product of crisp sets $X_1, X_2, \ldots, X_n$ *where tuples* $(x_1, x_2, \ldots, x_n)$ *may have varying membership degrees within the relation. The membership degree indicates the strength of the relation present between the tuple elements.*

Among n-dimensional relations, *binary relations* have a special significance since they are, in some sense, generalized mathematical functions. Contrary to functions from X to Y, binary relations $\tilde{R}(X, Y)$ may assign to each element of X two or more elements of Y. Some basic operations on functions, such as the inverse and composition, are also applicable to binary relations.

Definition 2.4.17. *Binary fuzzy relation*
A binary fuzzy relation $\tilde{R}$ *between variables "*x, y*" is a fuzzy relation defined by equation 2.18,*

$$\tilde{R} = \{(x, y) \mid \mu_{\tilde{R}}(x, y) \geq 0, \;\; x \in X, \;\; y \in Y\} \tag{2.18}$$

where $\mu_{\tilde{R}} : \tilde{A} \times \tilde{B} \rightarrow [0, 1]$

The membership function $\mu_{\tilde{R}}(x, y)$ is interpreted as the "strength" of the relation between x and y. When $\mu_{\tilde{R}}(x_1, y_1) \geq \mu_{\tilde{R}}(x_2, y_2)$, we say that (x_1, y_1) is more strongly related than (x_2, y_2).

Definition 2.4.18. *Domain and Range of* $\tilde{R}(X, Y)$
Given a fuzzy relation $\tilde{R}(X, Y)$*, its domain is a fuzzy set on* X*,* $\mathrm{dom}\tilde{R}$*, whose membership function is defined by*

$$\mu_{\mathrm{dom}\tilde{R}}(x) = \max_{y \in Y} \tilde{R}(x, y)$$

for each $x \in X$*.*
The range of $\tilde{R}(X, Y)$ *is a fuzzy relation on* Y*,* $\mathrm{ran}\tilde{R}$*, whose membership function is defined by*

$$\mu_{\mathrm{ran}\tilde{R}}(y) = \max_{x \in X} \tilde{R}(x, y)$$

for each $y \in Y$*.*

Composition of fuzzy relations

The $\sup -i$ compositions of fuzzy relation are widely used in some applications such as *approximate reasoning* and *fuzzy control*.

Definition 2.4.19.
Given a particular t-norm i and two fuzzy relations $\tilde{R}_1(X,Y)$ and $\tilde{R}_2(Y,Z)$, the $\sup -i$ *composition of $\tilde{R}_1$ and $\tilde{R}_2$ is a fuzzy relation, $\tilde{R}_1 \overset{i}{\circ} \tilde{R}_2$ on $X \times Y$, defined by,*

$$\mu_{\tilde{R}_1 \overset{i}{\circ} \tilde{R}_2}(x,z) = \sup_{y \in Y} -i[\mu_{\tilde{R}_1}(x,y), \mu_{\tilde{R}_2}(y,z)] \tag{2.19}$$

for all $x \in X$, $z \in Z$.

When the t-norm i is the min operator, $\tilde{R}_1 \overset{i}{\circ} \tilde{R}_2$ becomes the *standard composition* or $\max - \min$ *composition* of $\tilde{R}_1 \circ \tilde{R}_2$.

Definition 2.4.20. $\max - \min$ *composition of fuzzy relations*
Given the fuzzy relations $\tilde{R}_1$ and $\tilde{R}_2$ defined on sets X, Y and Z. That is $\tilde{R}_1 \subseteq X \times Y$, $\tilde{R}_2 \subseteq Y \times Z$. The standard composition $\tilde{R}_1 \circ \tilde{R}_2$ is expressed by the relation from X to Z, and is defined by equation 2.20,

$$\begin{aligned} \mu_{\tilde{R}_1 \circ \tilde{R}_2}(x,z) &= \max_{y \in Y} \left[\min(\mu_{\tilde{R}_1}(x,y), \mu_{\tilde{R}_2}(y,z))\right] \\ &= \vee_{y \in Y}[\mu_{\tilde{R}_1}(x,y) \wedge \mu_{\tilde{R}_2}(y,z)] \end{aligned} \tag{2.20}$$

for $(x,y) \in X \times Y$ and $(y,z) \in Y \times Z$.

Proposition 2.4.1 shows the basic properties that, under the standard fuzzy union and intersection, follow directly from the corresponding properties of *t*-norms,

Proposition 2.4.1.
Given fuzzy relations $\tilde{R}_1(X,Y)$, $\tilde{R}_2(Y,Z)$ and $\tilde{R}_3(Z,V)$, then

1. $(\tilde{R}_1 \circ \tilde{R}_2) \circ \tilde{R}_3 = \tilde{R}_1 \circ (\tilde{R}_2 \circ \tilde{R}_3)$
2. $\tilde{R}_1 \circ (\bigcap_{j \in J} \tilde{R}_2^j) \subseteq \bigcap_{j \in J} (\tilde{R}_1 \circ \tilde{R}_2^j)$
3. $(\bigcap_{j \in J} \tilde{R}_1^j) \circ \tilde{R}_2 \subseteq \bigcap_{j \in J} (\tilde{R}_1^j \circ \tilde{R}_2)$

2.4.6 Fuzzy numbers

Among the various types of fuzzy sets, of particular significance are fuzzy sets defined on the real line $\mathbb{R}$. Membership functions of these sets, which have the form:

$$\mu : \mathbb{R} \to [0,1]$$

have a quantitative meaning. They capture the intuitive conceptions of approximate numbers or intervals, such as "numbers that are closed to a given real number b" or "numbers that are around a given interval of real numbers $[b,c]$."

Definition 2.4.21. ***Fuzzy number***
A fuzzy number is a fuzzy set with the following conditions,

- *is convex*
- *is normalized*
- *its membership function is piecewise continuous*
- *it is defined in* $\mathbb{R}$

To qualify as a fuzzy number, a fuzzy set $\tilde{A}$ on $\mathbb{R}$ must fulfill the following three properties:

1. $\tilde{A}$ *must be a normalized fuzzy set*
2. *supp*$(\tilde{A})$ *must be bounded*
3. $\tilde{A}_\alpha$ *must be a closed interval for every* $\alpha \in [0, 1]$

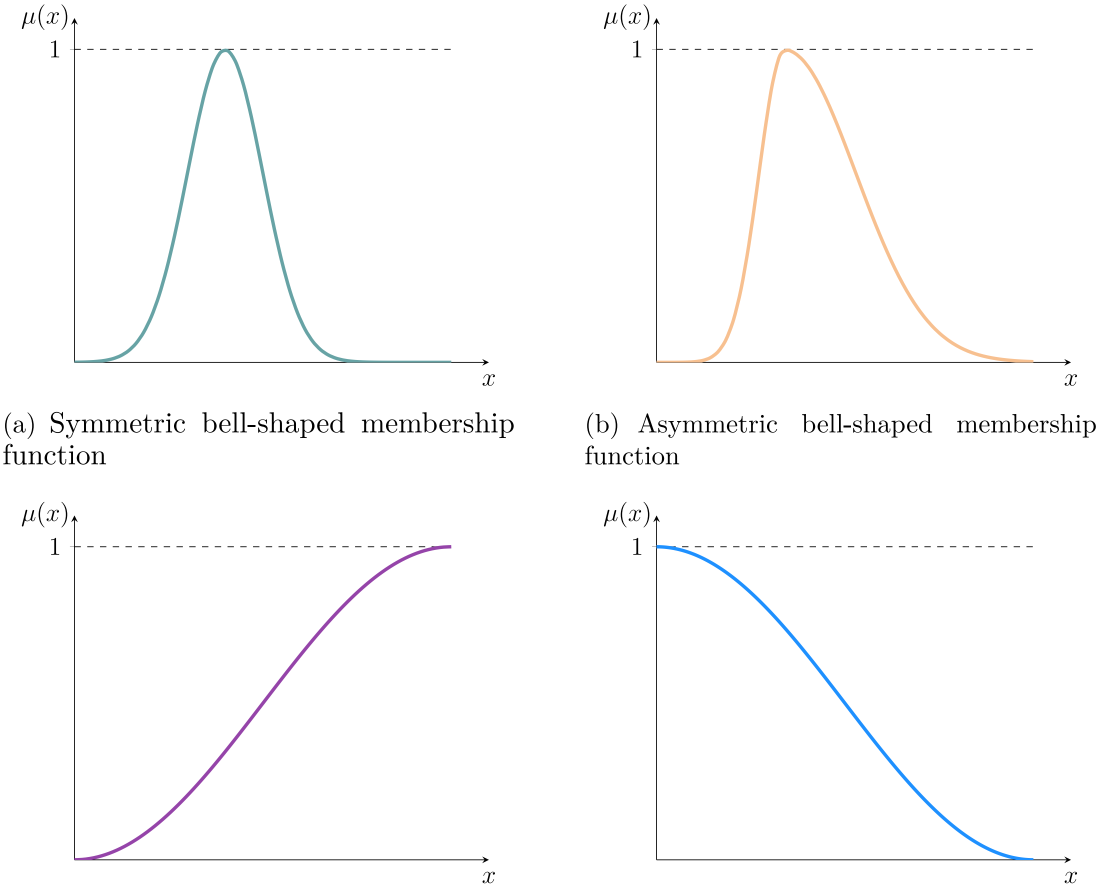
Given the real number b, due to the notion that a set of "real numbers close to b" is fully satisfied by b itself, the membership degree of b in any fuzzy set satisfying this notion (i.e., a fuzzy number) must be 1. The bounded support of $\tilde{A}$ and all its α-cuts for $\alpha \neq 0$ must be closed intervals to define meaningful arithmetic operations on fuzzy numbers in terms of standard arithmetic operations on real closed intervals. Since the α-cuts of any fuzzy number are required to be closed intervals for all $\alpha \in [0, 1]$, every fuzzy number is a convex fuzzy set.

Although the triangular and trapezoidal shapes of membership functions are the most often used for representing fuzzy numbers, other shapes may be preferable in some applications. Moreover, membership functions of fuzzy numbers do not need to be symmetric. Figure 2-2 shows some commonly used membership function shapes of fuzzy numbers. Cases (a) and (b) exemplify the bell-shaped membership function in symmetric and asymmetric forms, respectively. Cases (c) and (d), respectively, show membership functions that only increase or decrease, which also qualify as fuzzy numbers. They capture the conception of a "large number" or a "small number" in the context of each particular application.

Theorem 2.4.5 shows that membership functions of fuzzy numbers can be, in general, piecewise-defined functions,

Theorem 2.4.5. $f_L - f_R$ *Membership function of a fuzzy number*
Let $\tilde{A} = (a, b, c, d)$ *with* $a, b, c, d \in \mathbb{R}$ *be a fuzzy number. The membership function of* $\tilde{A}$ *is defined in equation 2.21,*

$$\mu_{\tilde{A}}(x) = \begin{cases} f_L(x) & a \leq x \leq b \\ 1 & b \leq x \leq c \\ f_R(x) & c \leq x \leq d \\ 0 & \textit{otherwise} \end{cases} \tag{2.21}$$



(a) Symmetric bell-shaped membership function

(b) Asymmetric bell-shaped membership function

(c) Increasing membership function

(d) Decreasing membership function

Figure 2-2: *Basic types for the membership function of a fuzzy number.*

where $f_L : [-\infty, b] \to [0,1]$ *is a monotonic increasing and continuous from the right function, and such that* $f_L(x) = 0$ *for* $x \in (-\infty, a)$*; and* $f_R : [c, \infty] \to [0,1]$ *is monotonic decreasing, continuous from the left function, and such that* $f_R(x) = 0$ *for* $x \in (d, \infty)$*.*

We refer the reader to [Klir and Yuan 1995] for the proof of theorem 2.4.5. Figure 2-3 shows a fuzzy number in a piecewise manner.

Triangular and Trapezoidal fuzzy numbers

Membership functions with linear functions f_L and f_R correspond to *triangular and trapezoidal shaped membership functions* and are often used for representing fuzzy numbers. Let $\tilde{A} = (a, b, c, d)$ be a triangular or trapezoidal fuzzy number,

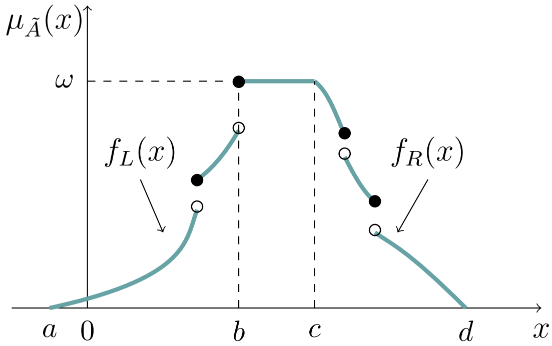


Figure 2-3: *General shape of the membership function of the fuzzy number $\tilde{A}$.*

the functions $f_L(x)$ and $f_R(x)$ that form its membership function are defined as:

$$f_L(x) = \begin{cases} 0, & x \in (-\infty, a) \\ \dfrac{x-a}{b-a}, & a \leq x \leq b \end{cases} \tag{2.22}$$
$$f_R(x) = \begin{cases} \dfrac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x \in (d, \infty) \end{cases}$$

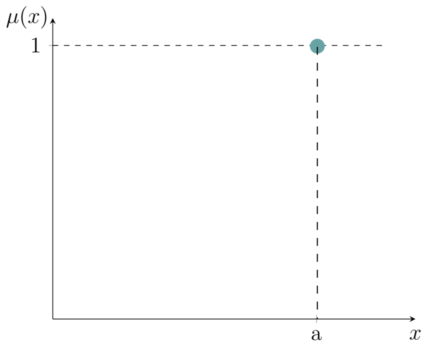
Functions $f_L(x)$ and $f_R(x)$ as described in expression 2.22 correspond to the membership function of a trapezoidal fuzzy number as long as $a \leq b < c \leq d$. When $b = c$, $\tilde{A}$ is a triangular fuzzy number.

Special cases of fuzzy numbers include ordinary real numbers and intervals of real numbers. These have the characteristic function instead of the membership function. Table 2.1 shows the different adaptations of a fuzzy number in trapezoidal and triangular and the crisp cases (real number and real interval) with graphic representation shown in figure 2-4.

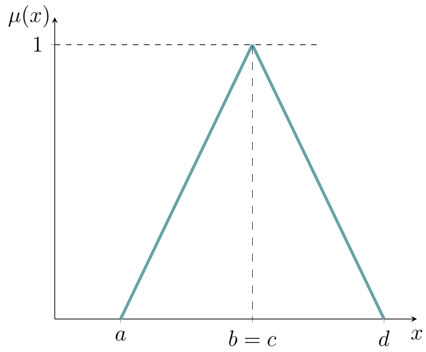
Fuzzy Number	Triangular ($b = c$)	Trapezoidal ($a \leq b < c \leq d$)
Crisp Number	Real Number ($a = b = c = d$)	Real Interval ($a = b < c = d$)

Table 2.1: *Special adaptations of a fuzzy number $\tilde{A} = (a, b, c, d)$.*

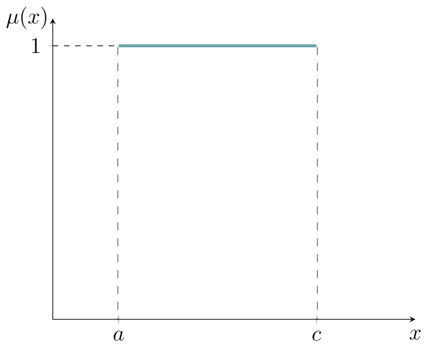
The α-cuts of a fuzzy number $\tilde{A} = (a, b, c, d)$ are obtained by the interception of functions $f_L(x)$ and $f_R(x)$ with the line $y = \alpha$ with $\alpha \in (0, 1]$. We summarize the α-cuts of triangular and trapezoidal fuzzy numbers in table 2.2 and graphically represent them in figure 2-5.



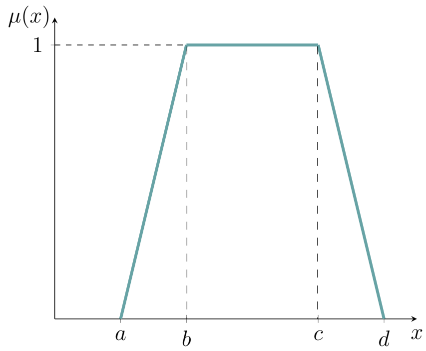
(a) Real number



(b) Triangular fuzzy number



(c) Real interval



(d) Trapezoidal fuzzy number

Figure 2-4: *Representations of real and fuzzy numbers.*

Fuzzy Number	α-cut
TFN $\tilde{A} = (a, b, c)$	$\tilde{A}_\alpha = [a + \alpha(b - a), c - \alpha(c - b)]$
TrapFN $\tilde{A} = (a, b, c, d)$	$\tilde{A}_\alpha = [a + \alpha(b - a), d - \alpha(d - c)]$

Table 2.2: *α-cuts of triangular and trapezoidal fuzzy numbers.*

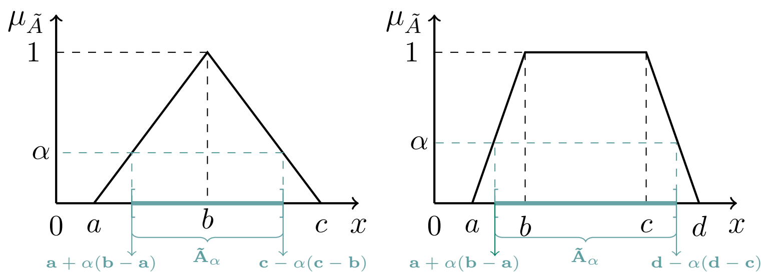


Figure 2-5: *α-cuts in triangular and trapezoidal fuzzy numbers.*

Arithmetic operations of Fuzzy Numbers

Arithmetic operations of fuzzy numbers are mainly based on two methods: *the extension principle method*, by which operations on real numbers are extended to operations on fuzzy numbers, and *the interval arithmetic method*, which is based on the arithmetic operations of intervals [[Klir and Yuan 1995] and [Banerjee and Roy 2012]]. We will assume that fuzzy numbers are represented by continuous membership functions for the description of both methods. Notice that if X is a set of real numbers bounded superiorly (that is, there is an M such that $x \leq M$, $\forall x \in X$), then $\sup(X)$ is the least upper bound for X. If X has a maximum element, then $\sup(X) = \max(X)$.

Two essential properties for the arithmetic operations of triangular and trapezoidal fuzzy numbers are described below:

1. The results from addition and subtraction between triangular or trapezoidal fuzzy numbers are also triangular o trapezoidal fuzzy numbers, respectively.

2. The results from multiplication or division as in maximum or minimum operations are not a triangular or trapezoidal fuzzy number.

We will denote by $\oplus$, $\ominus$, $\otimes$ and $\oslash$ the operation on fuzzy numbers, extended from the normal algebraic operations $+$, $-$, $\times$ and $/$, respectively.

Extension Principle Method: Classic fuzzy arithmetic is based on Zadeh's Extension Principle, where, mainly, standard arithmetic operations on real numbers are extended to fuzzy numbers by applying the Extension Principle described in section 2.4.4.

Let "$\circledast$" denote any of the four basic arithmetic operations ($\oplus, \ominus, \otimes, \oslash$) and let $\tilde{A}$ and $\tilde{B}$ denote two fuzzy numbers. Assuming, in definition 2.4.15, the function f as the operator "$\circledast$", where

$$f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$$

then the membership function of the fuzzy set $(\tilde{A} \circledast \tilde{B})$ on $\mathbb{R}$ is defined by the equation 2.23,

$$\mu_{(\tilde{A}\circledast\tilde{B})}(z) = \max_{z=x*y} \min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)] \tag{2.23}$$

for all $z \in \mathbb{R}$. More precisely,

$$\begin{aligned}
\mu_{(\tilde{A}\oplus\tilde{B})}(z) &= \max_{z=x+y} \min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)] \\
\mu_{(\tilde{A}\ominus\tilde{B})}(z) &= \max_{z=x-y} \min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)] \\
\mu_{(\tilde{A}\otimes\tilde{B})}(z) &= \max_{z=x\cdot y} \min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)] \\
\mu_{(\tilde{A}\oslash\tilde{B})}(z) &= \max_{z=x\div y} \min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)]
\end{aligned} \tag{2.24}$$

Theorem 2.4.6, whose proof the reader can find in [Klir and Yuan 1995], shows that more than a fuzzy set on $\mathbb{R}$, $(\tilde{A} * \tilde{B})$ is a fuzzy number for each $\circledast \in \{\oplus, \ominus, \otimes, \oslash\}$.

Theorem 2.4.6.
Let $\circledast \in \{\oplus, \ominus, \otimes, \oslash\}$, and let $\tilde{A}$,$\tilde{B}$ denote fuzzy number with continuous membership functions. The fuzzy set $(\tilde{A} \circledast \tilde{B})$ with membership function defined in expression 2.23 is a fuzzy number with a continuous membership function.

Interval Arithmetic Method: Using the Extension Principle for fuzzy arithmetic is highly complicated, even for simple operations of two small fuzzy numbers. Also, this method does not work, in general, with membership functions that do not intersect each other. This problem leads to some proposed solutions and algorithms for fuzzy arithmetic, such as the Approximate Methods of Extension [[Dubois and Prade 1980], [Ross 2004]]. Another well-known is the Interval Arithmetic method, where the operations of fuzzy numbers are reduced to the operations of ordinary intervals.

The following two properties of fuzzy numbers are based on the representation of a fuzzy set and the definition of the α-cut of a fuzzy set,

1. Each fuzzy set, and thus also each fuzzy number, can fully and uniquely be represented by its α-cut.
2. α-cuts of each fuzzy number are closed crisp intervals for all $\alpha \in (0, 1]$.

These properties enable the definition of arithmetic operations on fuzzy numbers in terms of the arithmetic operations of its α-cuts. The result of an arithmetic operation on a closed interval is again a closed interval, as is shown in proposition 2.4.2.

Proposition 2.4.2.
Let $*$ denote any of the four arithmetic operations on closed intervals, that is, $* \in \{+, -, \cdot, \div\}$. Then,

$$[a, b] * [c, d] = \{f * g \,|a \leq f \leq b,\, c \leq g \leq d\}$$

is a general property, except for the case $[a, b] \div [c, d]$ which is not defined when $c \leq 0 \leq d$.

The basic operations between the crisp intervals $[a, b]$ and $[c, d]$ are described in expression 2.25,

$$\begin{aligned}
&\text{Addition:} && [a, b] + [c, d] = [a + c, b + d] \\
&\text{Subtraction:} && [a, b] - [c, d] = [a - d, b - c] \\
&\text{Multiplication:} && [a, b] \cdot [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \\
&\text{Division:} && [a, b] \div [c, d] = \left[\min\left(\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\right), \max\left(\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\right)\right] \\
& && \text{with } 0 \notin [c, d]
\end{aligned} \tag{2.25}$$

Definition 2.4.22 provides the necessary elements to define the α-cut of the fuzzy number $(\tilde{A} * \tilde{B})$ by the α-cuts of the fuzzy numbers $\tilde{A}$ and $\tilde{B}$.

Definition 2.4.22.
Let $\tilde{A}$ and $\tilde{B}$ denote fuzzy numbers and let $*$ and $\circledast$ denote any of the four basic arithmetic operations and their extension to fuzzy numbers, respectively. Then we define a fuzzy number $(\tilde{A} \circledast \tilde{B})$, by defining its α-cut, $(\tilde{A} \circledast \tilde{B})_\alpha$, as

$$\boxed{(\tilde{A} \circledast \tilde{B})_\alpha = \tilde{A}_\alpha * \tilde{B}_\alpha}$$

for any $\alpha \in (0,1]$. When $* =$ "$\div$", clearly, $0 \notin \tilde{B}_\alpha$ $\forall \alpha \in (0,1]$.

By definition 2.4.22, we can use the interval operations in 2.25 to find $(\tilde{A} \circledast \tilde{B})_\alpha$. On the other hand, due to theorem 2.4.2, we can write $(\tilde{A} \circledast \tilde{B})$ as the union of the fuzzy numbers ${}_\alpha(\tilde{A} \circledast \tilde{B})$, i.e.:

$$\tilde{A} * \tilde{B} = \bigcup_{\alpha \in (0,1]} {}_\alpha(\tilde{A} \circledast \tilde{B})$$

According to definition 2.4.12, we can find the fuzzy number ${}_\alpha(\tilde{A} \circledast \tilde{B})$ by using $\chi_{(\tilde{A} * \tilde{B})_\alpha}(x)$, the characteristic function of $(\tilde{A} \circledast \tilde{B})_\alpha$. Finally, we can quickly compute the membership function of $(\tilde{A} \circledast \tilde{B})$ by applying the definition of the union of fuzzy sets (definition 2.4.8), i.e.,

$$\mu_{(\tilde{A} * \tilde{B})}(x) = \max_{\alpha \in (0,1]} \left\{ \alpha \cdot \chi_{(\tilde{A} * \tilde{B})_\alpha}(x) \right\}$$

We propose the example 2.4.6.1 to clarify the above mentioned methods.

Example 2.4.6.1. *Using the interval arithmetic method, we want to compute $\tilde{A} \oplus \tilde{B}$ where $\tilde{A} = (-3, 2, 4)$ and $\tilde{B} = (-1, 0, 6)$ with membership functions defined in expression 2.26 and shown in figure 2-6.*

$$\begin{aligned}
\mu_{\tilde{A}}(x) &= \begin{cases} 0, & x \leq -3,\ x \geq 4 \\ \dfrac{x+15}{5}, & -3 \leq x \leq 2 \\ \dfrac{4-x}{2}, & 2 \leq x \leq 4 \end{cases} \\
\mu_{\tilde{B}}(x) &= \begin{cases} 0, & x \leq -1,\ x \geq 6 \\ x+1, & -1 \leq x \leq 0 \\ \dfrac{6-x}{6}, & 0 \leq x \leq 6 \end{cases}
\end{aligned} \tag{2.26}$$

The fuzzy number $(\tilde{A} \oplus \tilde{B})$ is also a triangular fuzzy number. Therefore, we will find $\mu_{(\tilde{A} \oplus \tilde{B})}(x)$ following the Interval Arithmetic method.
The α-cuts of $\tilde{A}$ and $\tilde{B}$ are $\tilde{A}_\alpha = [5\alpha - 3, -2\alpha + 4]$ and $\tilde{B}_\alpha = [\alpha - 1, -6\alpha + 6]$,

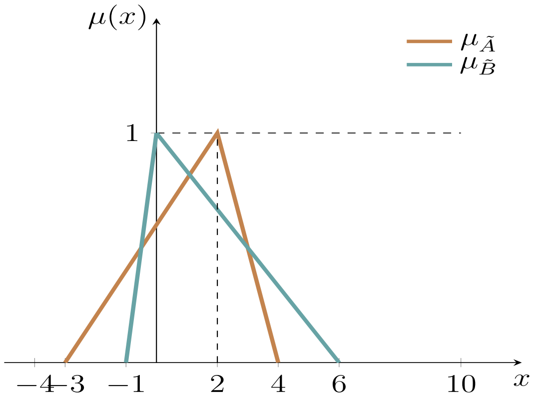


Figure 2-6: *Triangular fuzzy numbers $\tilde{A}$ and $\tilde{B}$.*

respectively. By applying the sum operation of ordinal intervals, described in expression 2.25, between $\tilde{A}_\alpha$ and $\tilde{B}_\alpha$, we obtain,

$$\tilde{A}_\alpha + \tilde{B}_\alpha = [6\alpha - 4, -8\alpha + 10]$$

We can define $(\tilde{A} \oplus \tilde{B})_\alpha$, by definition 2.4.22, for any $\alpha \in (0, 1]$. i.e.,

$$(\tilde{A} \oplus \tilde{B})_\alpha = [6\alpha - 4, -8\alpha + 10]$$

According to First Decomposition Theorem,

$$(\tilde{A} \oplus \tilde{B}) = \bigcup_{\alpha \in (0,1]} {}_\alpha(\tilde{A} \oplus \tilde{B})$$

thus, the membership function of $(\tilde{A} \oplus \tilde{B})$ can be computed by the definition of the union and decomposition of fuzzy sets, i.e.,

$$\mu_{(\tilde{A}\oplus\tilde{B})}(x) = \max_{\alpha \in (0,1]} \left\{ \mu_{\alpha(\tilde{A}\oplus\tilde{B})} \right\} = \max_{\alpha \in (0,1]} \left\{ \alpha \cdot \chi_{(\tilde{A}\oplus\tilde{B})_\alpha}(x) \right\}$$

where $\chi_{(\tilde{A}\oplus\tilde{B})_\alpha}(x) = \begin{cases} 1, & \text{if } x \in [6\alpha - 4, -8\alpha + 10] \\ 0, & \text{if } x < 6\alpha - 4 \quad \text{or} \quad x > -8\alpha + 10 \end{cases}$

We are now able to build $\mu_{(\tilde{A}\oplus\tilde{B})}(x)$, i.e.,

$$x = 6\alpha - 4 \quad \Rightarrow \quad \alpha = \frac{x+4}{6} \quad \Rightarrow \quad \begin{cases} \text{if } \alpha = 0 \;\Rightarrow\; x = -4 \\ \text{if } \alpha = 1 \;\Rightarrow\; x = 2 \end{cases}$$

$$x = -8\alpha + 10 \quad \Rightarrow \quad \alpha = \frac{x-10}{8} \quad \Rightarrow \quad \begin{cases} \text{if } \alpha = 0 \;\Rightarrow\; x = 10 \\ \text{if } \alpha = 1 \;\Rightarrow\; x = 2 \end{cases}$$

therefore,

$$\mu_{(\tilde{A}\oplus\tilde{B})}(x) = \begin{cases} \dfrac{x+4}{6}, & \text{if } -4 \le x \le 2 \\ \dfrac{10-x}{8}, & \text{if } 2 \le x \le 10 \\ 0, & \text{if } x \le -4 \quad \text{or} \quad x \ge 10 \end{cases}$$

which is the membership function of the triangular fuzzy number $(\tilde{A} \oplus \tilde{B}) = (-4, 2, 10)$*. Notice that* $(\tilde{A} \oplus \tilde{B})$ *is obtained by* $(\tilde{A} \oplus \tilde{B})_\alpha$ *for* $\alpha = 0$ *and* $\alpha = 1$*. Figure 2-7 shows the membership functions of* $\tilde{A}$ *and* $\tilde{B}$ *and the fuzzy number* $(\tilde{A} \oplus \tilde{B})$*.*

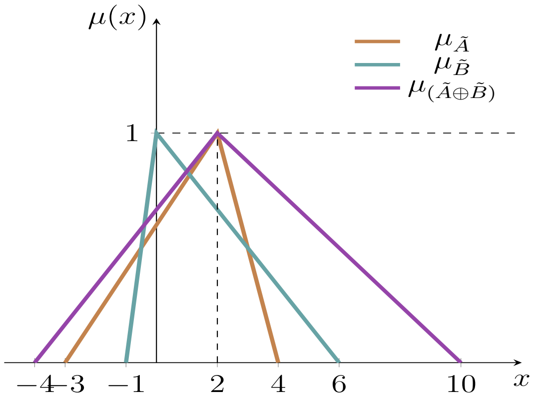


Figure 2-7: *Triangular fuzzy numbers* $\tilde{A}$*,* $\tilde{B}$*, and* $(\tilde{A} \oplus \tilde{B})$*.*

In summary, based on the interval arithmetic method, we can perform the addition and difference of triangular and trapezoidal fuzzy numbers, also resulting in triangular and trapezoidal fuzzy numbers, respectively, by using their α-cuts.

$\tilde{A}$, $\tilde{B}$ triangular fuzzy numbers:

$$\begin{aligned} \tilde{A} + \tilde{B} &= (a_1, b_1, c_1) \oplus (a_2, b_2, c_2) = (a_1 + a_2, b_1 + b_2, c_1 + c_2) \\ \tilde{A} + \tilde{B} &= (a_1, b_1, c_1) \ominus (a_2, b_2, c_2) = (a_1 - a_2, b_1 - b_2, c_1 - c_2) \end{aligned} \tag{2.27}$$

$\tilde{A}$, $\tilde{B}$ trapezoidal fuzzy numbers:

$$\begin{aligned} \tilde{A} + \tilde{B} &= (a_1, b_1, c_1, d_1) \oplus (a_2, b_2, c_2, d_2) = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2) \\ \tilde{A} + \tilde{B} &= (a_1, b_1, c_1, d_1) \ominus (a_2, b_2, c_2, d_2) = (a_1 - a_2, b_1 - b_2, c_1 - c_2, d_1 - d_2) \end{aligned} \tag{2.28}$$

Studies on fuzzy numbers like those presented in [Dubois and Prade 1979] and [Yager 1979] show that there are no opposite and reverse fuzzy numbers in the sense of group structure. It is easy to see that, mathematically, for any fuzzy number $\tilde{A}$, $\tilde{A} + (-\tilde{A}) \neq 0$ and $\tilde{A} \otimes (\frac{1}{\tilde{A}}) \neq 1$. We summarize in definitions 2.4.23 and 2.4.24 *negative* and *complementary* triangular and trapezoidal fuzzy numbers.

Definition 2.4.23. *Positive, nonnegative, and negative triangular fuzzy numbers.*
A triangular fuzzy number $\tilde{A} = (a, b, c)$ *is a nonnegative triangular fuzzy number, i.e.,* $\tilde{A} \geq 0$ *if and only if* $a \geq 0$*.* $\tilde{A}$ *is said to be a positive (negative) triangular fuzzy number, i.e.,* $\tilde{A} > 0$ $(\tilde{A} < 0)$ *if and only if* $a > 0$ $(a < 0)$*.*

Definition 2.4.24. *Complementary of a triangular fuzzy number.*
Let $\tilde{A} = (a, b, c)$ *be a triangular fuzzy number. The complementary of* $\tilde{A}$ *is a triangular fuzzy number defined by* $-\tilde{A} = (-c, -b, -a)$*.*

There is a special class of trapezoidal fuzzy numbers that do not satisfy the criterion for both positivity and negativity. These fuzzy numbers are neither positive nor negative and are interpreted as *near-zero fuzzy numbers*.

Definition 2.4.25. *N-zero fuzzy number.*
A fuzzy number $\tilde{A}$ is called N-zero fuzzy number, if its membership function $\mu_{\tilde{A}}(x)$ satisfies $\mu_{\tilde{A}}(0) = \mu_{\tilde{A}}(0^+) = \mu_{\tilde{A}}(0^-) = \mu_{\tilde{A}}(0) \neq 0$. In particular:

- *A triangular fuzzy number $\tilde{A} = (a, b, c)$ is a N-zero triangular fuzzy number if and only if $a < 0 < c$.*
- *A trapezoidal fuzzy number $\tilde{A} = (a, b, c, d)$ is a N-zero trapezoidal fuzzy number if and only if $a < 0 < d$.*

In table 2.3, we summarize the three types of N-zero triangular and trapezoidal fuzzy numbers:

	N_1-zero FN	N_2-zero FN	N_3-zero FN
Triangular fuzzy number	iff $a < b < 0 < c$	iff $a < 0 < c$	iff $a < 0 < b < c$
Trapezoidal fuzzy number	iff $a < b \leq c < 0 < d$	iff $a << b < 0 < c < d$	iff $a < 0 \leq b \leq c < d$

Table 2.3: *Types of N-zero fuzzy numbers.*

The class of N-zero fuzzy numbers shares the same arithmetic expression for extended addition and subtraction as with nonnegative fuzzy numbers. However, in the class of basic arithmetic functions, these numbers behave differently for extended multiplication, division, and inverse.

2.4.7 Fuzzy Graphs

As we mentioned before, graph theory is an important tool to represent different types of networks. Thus, solving problems defined on graphs has extensive applications in many real-world problems. However, due to the uncertainty or haziness of the parameters of networks, it is becoming increasingly frequent not to represent these systems appropriately through a graph. This uncertainty motivated to define the concept of *fuzzy graph*, where the union of graph and fuzzy set concepts allowed to consider the uncertainty that is present in a problem defined on a network. Thus, more realism is provided in applications of mathematics in different areas of science and technology, obtaining greater precision and realism in the results obtained. The applications of fuzzy graphs include networking, communication, data mining, image segmentation, image capturing, clustering, planning, scheduling, etc.

Although crisp and fuzzy graphs are structurally similar, the uncertainty on vertices and/or edges makes the fuzzy graphs particularly important. [Rosenfeld 1975] first introduced the concept of fuzzy graphs, along with fuzzy analogs of some basic concepts in graph theory like paths, cycles, trees, and connectedness, etc. Subsequently, many authors contributed to developing fuzzy graphs, becoming this a vast research area. We refer the reader to [Li and Yi 2017] and [Mordeson and Nair 2000], respectively, for a detailed exposition about different concepts related to fuzzy graphs, together with theoretical and applied aspects of fuzzy graphs.

From the general definition given by [Rosenfeld 1975], [Blue et al. 2002] proposes a more extensive study about the classification of fuzzy graphs. In particular, in [Blue et al. 2002] the authors presented a taxonomy of fuzzy graphs that treats fuzziness in vertex existence, edge existence, edge connectivity, and edge weight. Thus, the authors define five types of fuzziness possible in graphs, where each can be effectively exemplified according to the problem's nature at which it is applied. For our purposes in this work, we only consider the type V fuzzy graph, although we present in the following a brief description of each of them.

Type I: Fuzzy set of crisp graphs.
It consists of a fuzzy set $\tilde{G}$ of crisp graphs.

$$\tilde{G} = \{G_i, \mu_{G_i}\}, \quad i = \overline{1, n_{\tilde{G}}}$$

This fuzziness is trivial and is not interesting unless the graphs G_i have some vertices or edges in common. Even in this case, the analysis is complicated unless the commonality has a regular structure. The case of most interest occurs when each of the crisp graphs has the same set of vertices, so the presence and configuration of the edges are fuzzy for these graphs.

Type II: Crisp vertices set and fuzzy edge set.
The graph has known vertices but unknown edges. In this case, the vertices set is crisp, and the edge set is fuzzy.

Type III: Crisp vertices and edges with fuzzy connectivity.
The graph has known vertices and edges but unknown edge connectivity. The vertices and edges sets are crisp, but the edges have fuzzy heads and tails.

Type IV: Fuzzy vertices set and crisp edges set.
The graph has unknown vertices but known edges. In this case, the vertices set is fuzzy, and the edges set is crisp.

Type V: Crisp graph with fuzzy weights.
The graph has known vertices and edges but unknown weights on the edges. In this case, only the weights are fuzzy. In particular, fuzzy numbers.

Definition 2.4.26. *Type V fuzzy graph*
The graph $\tilde{G} = (V, E, \mathfrak{C})$ *where,*

V : *set of vertices*
E : *set of edges*
$\mathfrak{C}$: *set of fuzzy weights*

is a type V fuzzy graph

This type of graph fuzziness is one of the more interesting and widely used. Therefore, we will focus our interest in type V fuzzy graphs throughout our work.

2.4.8 Elements of Fuzzy Logic and Fuzzy Inference

Linguistic Variable

In section 2.4.6, we gave a brief description and principal concepts about fuzzy numbers. This particular case of fuzzy sets plays a fundamental role in formulating *quantitative fuzzy variables*, among others. These are variables whose states are fuzzy numbers. Also, when the fuzzy numbers represent linguistic concepts, such as very small, small, medium, etc., as interpreted in a particular context, the resulting constructs are usually called *linguistic variables*.

A *base variable* is a variable in the classical sense, exemplified by any physical variable (e.g., temperature, speed, humidity, etc.) and any other numerical variable (e.g., age, performance, salary, reliability, etc.). A linguistic variable is defined in terms of a base variable.

Appropriate fuzzy numbers capture the linguistic terms representing approximate values of a base variable in a linguistic variable relevant to a particular application. Definition 2.4.27 gives a formal summary of the concept of a linguistic variable.

Definition 2.4.27. *Linguistic Variable*
A linguistic variable is defined by the following quintuple,

$$Linguistic\ Variable = (x, \mathcal{T}(x), X, G, m)$$

where:

x - the name of the variable

X - universal set which defines the characteristics of the variable.

$\mathcal{T}(x)$ - Set of linguistic terms of x that refer to a base variable whose values range over a universal set X.

G - syntactic rule (a grammar), which produces terms in $\mathcal{T}(x)$

m - a semantic rule that assigns to each linguistic term $t \in \mathcal{T}$ its "meaning", $m(t)$, which is a fuzzy set on X (i.e., $m : \mathcal{T} \to \mathcal{F}(X)$)

Example 2.4.8.1. *Figure 2-8 shows an example of a linguistic variable called* Age. *This variable expresses the age (which is the base variable in this example) of an individual in a given context by five linguistic terms (very young, young, adult young, middle age and old), as well as other linguistic terms (very very young, very old, etc.) generated by a syntactic rule. The figure shows that each linguistic term is assigned to one of the five fuzzy numbers by a semantic rule. In this case, with the trapezoidal membership function, the fuzzy numbers are defined in the interval* $[0, 100]$*, which is the range of the base variable. Each of them expresses a fuzzy restriction on this range.*

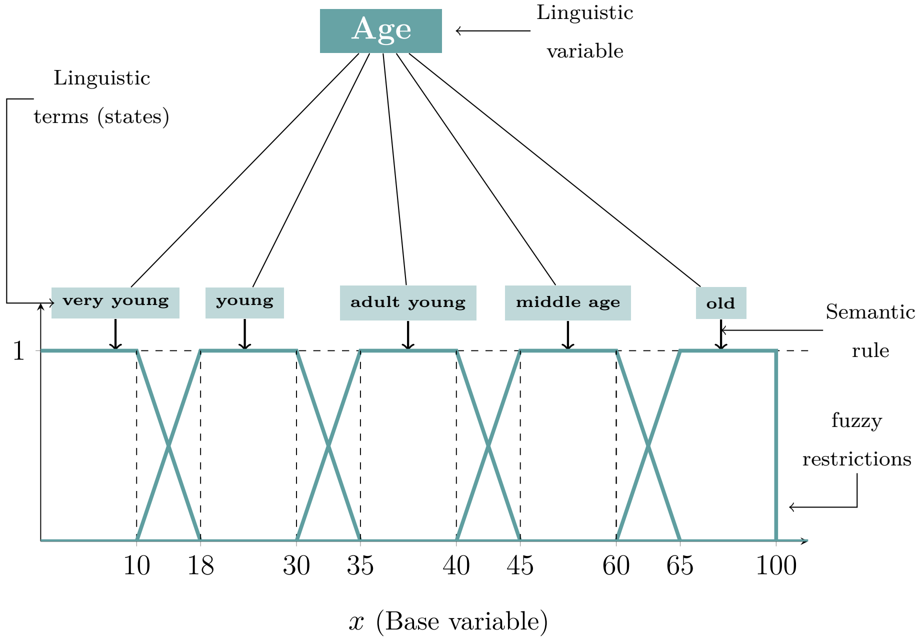


Figure 2-8: *An example of a linguistic variable.*

Fuzzy Propositions and Fuzzy Rules

Propositions are sentences expressed in some language. Each sentence representing a proposition can be divided into a *subject* and a *predicate*. The main difference between classical and fuzzy propositions lies in the range of their truth values. In a classical proposition, the truth and falsity are expressed by values 1 and 0, respectively. On the other hand, in a fuzzy proposition, the truth's degree is represented by a number in the interval $[0, 1]$. In this way, we obtain what is known as an *unqualified and unconditional proposition* defined in expression 2.29.

$$p^{uu} : \; x \text{ is } \tilde{A} \tag{2.29}$$

where x is a variable taking values in some universal set X, and $\tilde{A}$ is a fuzzy set representing a fuzzy predicate. Given a particular value of x (say, x_0), this value belongs to $\tilde{A}$ with membership degree $\mu_{\tilde{A}}(x_0)$. This membership degree is then interpreted as the degree of truth of proposition p^{uu} for each given particular value x_0 of variable x in proposition p^{uu}. Then, p^{uu} is a fuzzy set on $[0, 1]$, which assigns the membership degree $\mu_{\tilde{A}}(x_0)$ to each value x_0 of variable x.

An affirmation of type "x is $\tilde{A}$" is called *simple* since it only contains one proposition, p^{uu}. However, the propositions can be interconnected, creating compound affirmations. In logic, the connections are made by the following operators:

$\wedge$ for the connection AND,

$\vee$ for the connection OR,

$\neg$ for the connection NOT, and

$\rightarrow$ for the implication connection (IF-THEN)

Given the simple propositions "x is $\tilde{A}$", "y is $\tilde{B}$" and "z is $\tilde{C}$". Connecting these proposition using the operators $\wedge$ and $\rightarrow$ we obtain the compound propositions 2.30 and 2.31,

$$\text{if } \ x \text{ is } \tilde{A} \ \text{ then } \ y \text{ is } \tilde{B} \tag{2.30}$$

or

$$\text{if } \ x \text{ is } \tilde{A} \ \text{ and } \ y \text{ is } \tilde{B} \ \text{ then } \ z \text{ is } \tilde{C} \tag{2.31}$$

where $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are linguistic values defined by fuzzy sets on the universes of discourse X, Y, and C, respectively, and x, y, and z are variables in X, Y, and Z, respectively.

The equation 2.30 shows a *fuzzy if-then rule* in its simple form, and the equation 2.31 shows a fuzzy if-then rule in its compound form. The if-part is called *antecedent or premise*, and the then-part is called the *consequence or conclusion*.

Fuzzy Modus Ponens

In propositional logic, there are many schemes of reasoning called *tautologies*. A widely used scheme is the so-called *Modus Ponens* and establishes that if an implication and its premise are true, then it can be inferred that the conclusion is also true.

In classical propositional logic, reasoning deals with always entirely true propositions (assigning the numerical value 1) or entirely false (assigning the value 0). This form of inference is very efficient from the mathematical point of view. However, it rarely properly works when translated into the human language since most propositions in ordinary human reasoning cannot be considered entirely true or entirely false. Therefore, classical logic has problems with representing and inferring imprecise knowledge. On the other hand, the classical implication using fuzzy sets to represent vague knowledge does not satisfactorily solve the reasoning process.

It is appropriate to consider the classical implication process as a particular case of the fuzzy implication process. Therefore, the inference models used in classical logic can be extended to the fuzzy case. In particular, we define the Fuzzy Modus Ponens as shown in the definition 2.4.28,

Definition 2.4.28. *Fuzzy Modus Ponens (FMP)*
Let us assume, without loss of generality, a fuzzy if-then rule in the form 2.30 and a fact in the form,

$$x \text{ is } \tilde{A}$$

Then we can infer and obtain the new result,

$$y \text{ is } \tilde{B}$$

Thus, the Fuzzy Modus Ponens reasoning is defined in expression 2.32,

$$\begin{array}{ll} \textit{Fact:} & x \text{ is } \tilde{A} \\ \textit{Rule:} & \text{if } x \text{ is } \tilde{A} \text{ then } y \text{ is } \tilde{B} \\ \textit{result:} & y \text{ is } \tilde{B} \end{array} \tag{2.32}$$

Mamdani Fuzzy Implication

In a fuzzy rule, the entry to the implication process is a number resulting from evaluating the antecedent of the rule. On the other hand, the output is a fuzzy set that constitutes the inference provided by the rule.

There is no loss of generality in assuming a fuzzy if-then rule in its simple form (equation 2.30). To represent the membership function of the fuzzy set resulting from the implication, $\mu_{\tilde{A}\to\tilde{B}}(x,y)$, several methods based on the interpretations of the Cartesian product and various T-norms and T-conorms can be formulated.

One of the most known is the *Mamdani Fuzzy Implication (MFI)*. The MFI interprets the fuzzy implication as the minimum operation. The fuzzy implication function, proposed by Mamdani [[Lee 2005]], determine $\mu_{\tilde{R}}$ according to equation 2.33,

$$\begin{aligned} \mu_{\tilde{A}\to\tilde{B}}(x,y) &= \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(y) \\ &= \min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)] \end{aligned} \tag{2.33}$$

In the inference model Modus Ponens, the “min” operator expresses that the certainty value of the consequent cannot be higher than that of the antecedent. Thus, given a fuzzy rule with form as in expression 2.32, the MFI truncates the superior part of the membership function of the consequent with the larger certainty value. For example, figure 2-9 shows the Mamdani implication in a specific rule with one fuzzy set as consequent represented by the membership function $\mu_{\tilde{B}}(y)$. The resultant fuzzy set of the inference, $\mu_{\tilde{A}\to\tilde{B}}(x,y)$, is the highlighted area.

The decision about which operator to use depends on the specific problem being addressed. Therefore, it is always convenient to run tests before applying the implication operator.

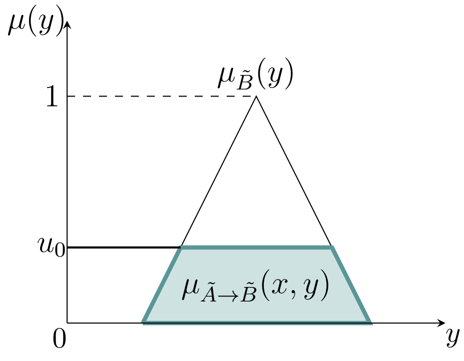


Figure 2-9: *Example of Mamdani operator*

Results of inference in one rule with two inputs

Once we determine the implication method of our interest, we will describe the result of inference in an FMP consisting of a single fuzzy if-then rule in the form 2.31 with two inputs and one output.

Given the fuzzy sets $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ with universes of discourse X, Y and, Z, respectively. The FMP is defined by schema 2.34,

$$\begin{array}{|lr|}\hline \text{Fact:} & x \text{ is } \tilde{A} \text{ and } y \text{ is } \tilde{B} \\ \text{Rule:} & \text{if } x \text{ is } \tilde{A} \text{ and } y \text{ is } \tilde{B} \text{ then } z \text{ is } \tilde{C} \\ \text{consequence:} & z \text{ is } \tilde{C} \\ \hline \end{array} \tag{2.34}$$

The antecedent of the rule 2.34 is composed through the AND connector by two simple propositions. Thus, the membership degrees of each antecedent are related by the “min" operator described in equation 2.11. This operation provides a numerical value that defines the grade of truth (matching degree), α, which varies according to the operator used and the values that the input variables take. The matching degree of the sets $\tilde{A}$ and $\tilde{B}$ for the values x_0 and y_0, respectively, is defined in equation 2.35,

$$\alpha(x_0, y_0) = \min\{\mu_{\tilde{A}}(x_0), \mu_{\tilde{B}}(y_0)\} \tag{2.35}$$

The inference result is then determined by the application of the MFI, i.e.,

$$\mu_{\tilde{C}'}(z) = \mu_{(\tilde{A}\cap\tilde{B})\to\tilde{C}}(z) = \min\{\alpha(x_0, y_0), \mu_{\tilde{C}}(z)\} \tag{2.36}$$

where $\alpha(x_0, y_0)$ is the matching degree defined in equation 2.35.

Figure 2-10 shows an example of the inference result for a fuzzy rule in the form 2.31.

Rule Base

A *knowledge system* is often represented by a *fuzzy rule base* consisting of fuzzy if-then rules. We will consider a rule base with two inputs and a single output, and

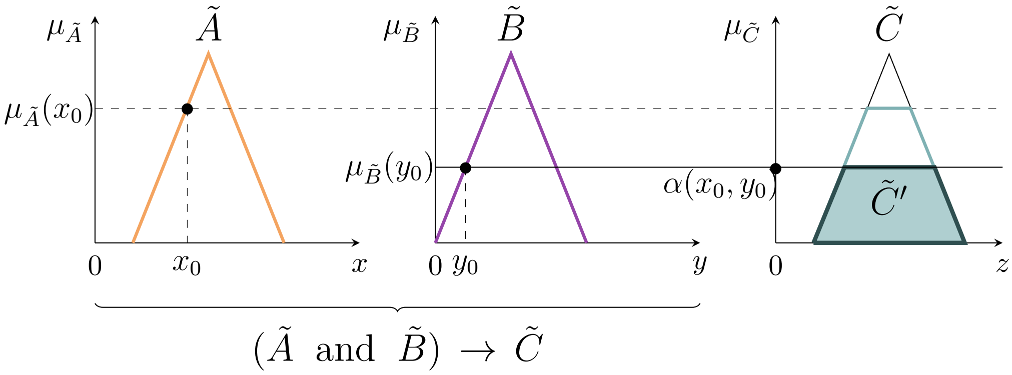


Figure 2-10: *Result of inference in one rule with two inputs and a single output.*

where the rules are compound by the AND connector. In the inference scheme, the rule base will be based on the FMP. Expression 2.37 summarizes a fuzzy system with the rule base previously described,

$$\boxed{\begin{array}{rcr} \text{Fact} & : & x \text{ is } \tilde{A} \text{ and } y \text{ is } \tilde{B} \\ \text{Rule 1} & : & \text{if } x \text{ is } \tilde{A}_1 \text{ and } y \text{ is } \tilde{B}_1 \text{ then } z \text{ is } \tilde{C}_1 \\ \text{Rule 2} & : & \text{if } x \text{ is } \tilde{A}_2 \text{ and } y \text{ is } \tilde{B}_2 \text{ then } z \text{ is } \tilde{C}_2 \\ & \ldots & \\ & \ldots & \\ \text{Rule n} & : & \text{if } x \text{ is } \tilde{A}_n \text{ and } y \text{ is } \tilde{B}_n \text{ then } z \text{ is } \tilde{C}_n \\ \text{consequence} & : & z \text{ is } \tilde{C} \end{array}} \tag{2.37}$$

where x, y, and z represent linguistic variables. For all $i \in \mathbb{N}$, the fuzzy sets $\tilde{A}_i \in \mathcal{F}(X)$, $\tilde{B}_i \in \mathcal{F}(Y)$ and $\tilde{C}_i \in \mathcal{F}(Z)$ are linguistic terms of x, y, and z in the universes of discourse X, Y and Z, respectively.

Result of inference with two inputs/single output Rule Base)

We can now determine the result of inference in the fuzzy system 2.37. For convenience, we ignore the connectives "also" and "else" between the rules since they are not part of our approaches in this work.

After the implication process, each rule will produce a fuzzy set as output. These sets are combined so that the fuzzy system provides an output resulting from the global inference process. We aim to define the overall result of inference. Thus, we treat the rules in 2.37 in either of two different ways, *disjunctive* and *conjunctive*. The interpretation of the rules depends on their intended use and how we obtain each rule i ($i = \overline{1, n}$). We address our attention to disjunctive rules.

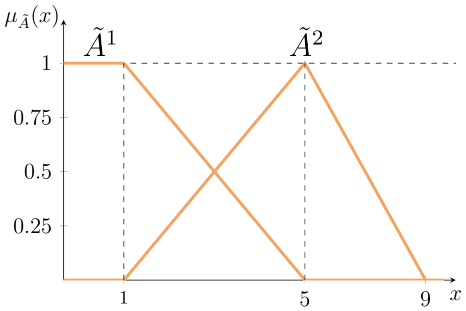
Rules treated as disjunctive: We obtain a conclusion for a given fact(s) whenever the grade of truth is positive for at least one rule.

Given this interpretation for the rules, we use one of the most popular combination methods: the maximum value that results from composing the fuzzy sets output of each rule.

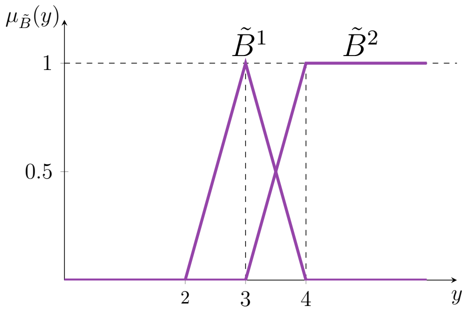
$$\tilde{C}' = \bigcup_{i=1}^{n} \tilde{C}'_i \tag{2.38}$$
$$\mu_{\tilde{C}'} = \max_{i\in[1,n]}\{\mu_{\tilde{C}'_i}\}$$

To gain clarity, we give an example of a fuzzy system with two rules where we find the overall result after applying the inference process.

Example 2.4.8.2. *Given the input variables x and y taking values in the universes of discourse X and Y, respectively. Also, consider the fuzzy sets $\tilde{A}^1$ and $\tilde{A}^2$ defined on X, and $\tilde{B}^1$ and $\tilde{B}^2$ defined on Y corresponding to the variables x and y, respectively (figures 2-11(a)-2-11(b)).*



(a) Fuzzy sets $\tilde{A}^1$ and $\tilde{A}^2$



(b) Fuzzy sets $\tilde{B}^1$ and $\tilde{B}^2$

Figure 2-11: *Fuzzy sets corresponding to input variables.*

The output variable z takes values in the universe of discourse Z, i. e. its corresponding fuzzy sets $\tilde{C}^1$ and $\tilde{C}^2$ are defined on Z, figure 2-12.

The rule base is defined in expression 2.39

$$\begin{aligned} \text{Rule 1} &: \text{ if } x \text{ is } \tilde{A}^1 \text{ and } y \text{ is } \tilde{B}^2 \text{ then } z \text{ is } \tilde{C}^1 \\ \text{Rule 2} &: \text{ if } x \text{ is } \tilde{A}^2 \text{ and } y \text{ is } \tilde{B}^1 \text{ then } z \text{ is } \tilde{C}^2 \end{aligned} \tag{2.39}$$

Let us take $x = 2$ and $y = 3.5$. The antecedent of each rule in the rule base is compound. Therefore, we will compute the matching degree $\alpha(2, 3.5)$ for each rule according to equation 2.35. Table 2.4 summarizes the values of the matching degrees (grade of truth) for each rule.

We can now apply the implication process (MFI operator) to each rule. The

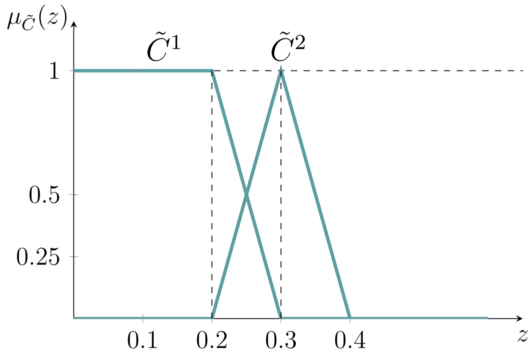


Figure 2-12: *Fuzzy sets corresponding to output variable.*

Rule	$\mu_{\tilde{A}^i}$	$\mu_{\tilde{B}^j}$	$\alpha(2, 3.5)$
1	0.75	0.5	0.5
2	0.25	0.5	0.25

Table 2.4: *Matching degrees for each rule*

inference result for each rule is described in equation 2.40

$$\begin{aligned} \mu_{\tilde{C}^{1'}}(z) &= \mu_{(\tilde{A}^1 \cap \tilde{B}^2) \to \tilde{C}^1}(z) = \min\{0.5, \mu_{\tilde{C}^1}(z)\} \\ \mu_{\tilde{C}^{2'}}(z) &= \mu_{(\tilde{A}^2 \cap \tilde{B}^1) \to \tilde{C}^2}(z) = \min\{0.25, \mu_{\tilde{C}^2}(z)\} \end{aligned} \tag{2.40}$$

Finally, once the output fuzzy sets $\tilde{C}^{1'}$ and $\tilde{C}^{2'}$ are obtained from each rule, these are combined according to expression 2.38 to determine the overall inference result $\tilde{C}'$. Figure 2-13 shows the methods of implication and combination in the rule given the values of the input variables.

In chapter 3, we propose a complete fuzzy inference process in a fuzzy system applied to a specific problem in a P2P network. We will describe the different steps that are part of this fuzzy reasoning in detail.

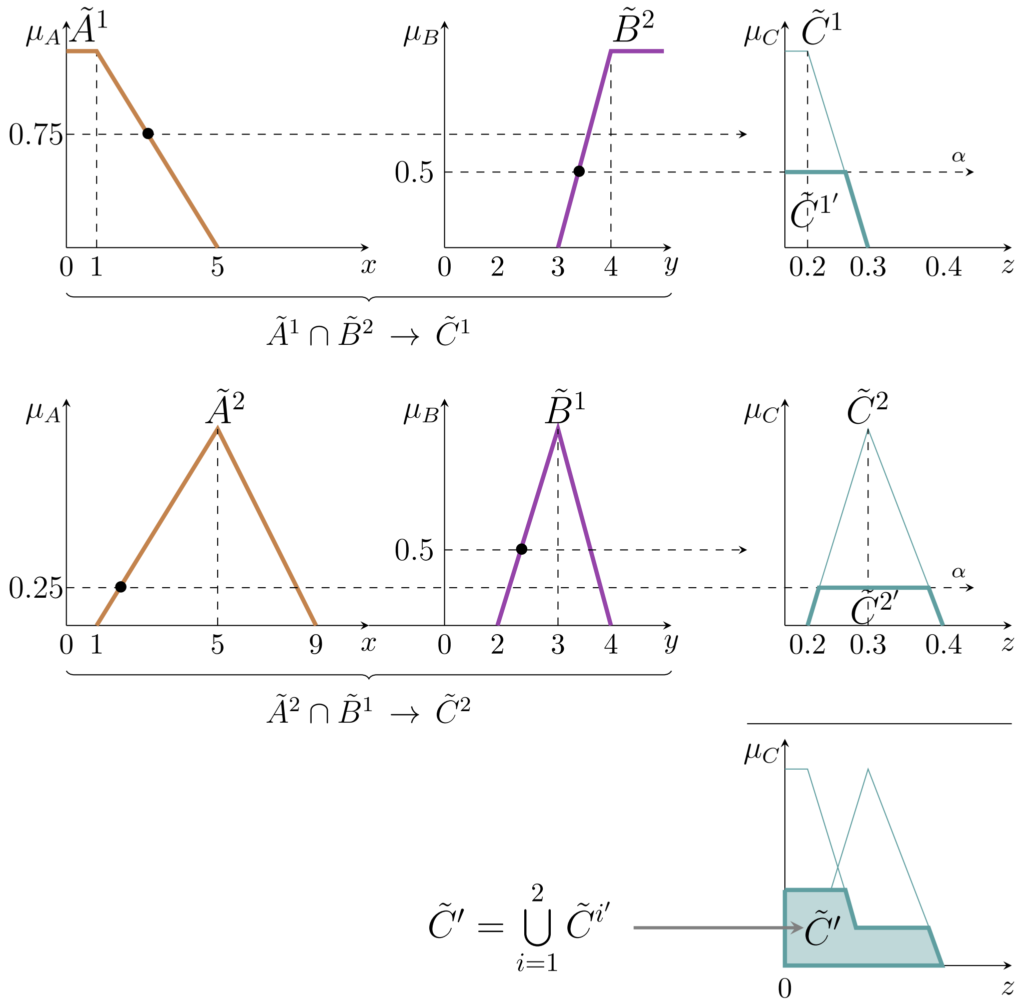


Figure 2-13: *Methods of implication and combination in the fuzzy system.*

2.5 Conclusions

In this chapter we provided the concepts that will be necessary to understand the technical terms and approaches referred in chapters 3 to 5. In particular, we showed the essential concepts of the three central areas we address in the thesis work:

1. Topology and functioning of communication networks, where we presented concepts like Protocol and Traffic.

2. Elements of Graph Theory, where we provided the basic definitions necessary for the modeling of a communication network, to later declare problems defined on this.

3. Elements of Fuzzy Logic, where we focuss our attention on Fuzzy Inference System and arithmetic properties of fuzzy numbers.

We exposed each theoretical concept concisely. In this way, the chapter serves as a reference when it is necessary for the reader to remember or verify the theoretical foundations of some result.

CHAPTER 3

FUZZY INFERENCE ALGORITHM APPLIED TO P2P NETWORK

- L. Valdés, A. Ariza, S. M. Allende, R. Parada, G. Joya; "Study of alternative strategies to the selection of peer in P2P wireless mesh networks"; In: *Advances in Computational Intelligence. IWANN 2013. Lecture Notes in Computer Science. Springer Berlin Heidelberg*; Vol. 7902; pp. 124–132; (2013);
DOI: 10.1007/978-3-642-38679-4_11.
- L. Valdés, S. Montesinos, A. Ariza, S. M. Allende, G. Joya; "Peer selection in P2P wireless mesh networks: comparison of different strategies"; In: *Soft Computing, Springer-Verlag Berlin Heidelberg*; Vol. 19; pp. 2447–2455; (2015);
DOI: 10.1007/s00500-014-1572-6.

In this chapter, we study the use of various strategies for selecting the server node in Peer to Peer networks, which are especially oriented networks with limited resources such as wireless mesh networks based on WiFi technology. We examine three different strategies: Random, the currently most used one, in which the server node is randomly chosen, Min-Hop, that selects the path with the least number of hops, and Purely Fuzzy, where the selection is made from a fuzzy inference process using the number of hops and the ETX cost (Expected Transmission Count) as fuzzy inputs. We analyze two different transmission scenarios: without obstacles and with obstacles between nodes. Results show that the currently more extended random strategy is the least efficient in most of cases, Min-Hop and Purely Fuzzy have a very similar behavior in networks without obstacles, and Purely Fuzzy is clearly more efficient in networks with obstacles. The study of performance was carried out using a wireless network simulation tool for discrete event simulation OMNeT++.

3.1 Introduction

The peers in a P2P network are relatively autonomous and can join or leave the system at any time. As a result, they have been successfully used for sharing computation, internet services, or data. A P2P system can usually scale up to many peers due to distributing data storage, processing data, or the presence of bandwidth across autonomous peers.

Applications P2P are responsible for a significant percentage of total traffic generated on the Internet in recent years. Such applications were initially designed to share information among multiple users, and in them, there is no clear distinction between client and server nodes. Instead, each node can act as both client and server (peer network) [Buford et al. 2008].

P2P networks are highly efficient in information sharing, allowing quick dissemination of information avoiding bottlenecks created on dedicated servers. The network achieves its efficiency at first dividing data into segments. Then, these segments are distributed in the network to maximize the number of nodes acting as servers of that information. Figure 3-1 shows the evolution of the data distribution in a P2P network in three-time instants. Initially, only node A has the four segments that constitute the data, and this node serves a different segment for each of the nodes that request. In stage 2, node A sends segments 3 and 4 to nodes B and C, respectively, as long as they exchange segments 1 and 2. Finally, nodes B and C exchange segments 3 and 4; thus, all nodes obtain all segments. In this process, node A has sent every segment only once. In [Kang 2011] and [Luo 2012], we can find two interesting surveys about P2P networks.

One limitation in Internet P2P networks is that most nodes do not use a permanent IP address. Depending on how this problem is solved, a classification of P2P networks is established:

Centralized: All transactions are carried out based on a central server that stores and distributes information about the contents of the nodes.

Hybrid: A central server manages some resources, but nodes are responsible for maintaining the data.

Completely decentralized: There is no central server.

Selecting the server node in each information segment exchanging is an essential question in wireless networks where bandwidth resources are minimal and shared among all users since it can waste resources. This issue is especially critical in multihop wireless networks because if a distant server node is serving a client, the information must cross a large number of nodes, increasing the probability of interfering with the transmissions of other nodes. Equation 3.1 shows the probability B_P of occurrence of an error in the information transmission in a path P.

$$B_P = 1 - \prod_{\forall i \in P} (1 - B_i) \tag{3.1}$$

where i is the i-th hop in path P and B_i is the probability of error in the i-th hop. Increasing the size of the route increases the chance of losing information. Furthermore, each error B_i increases the total traffic in the network because the link layer will retransmit the lost package several times until the package arrives

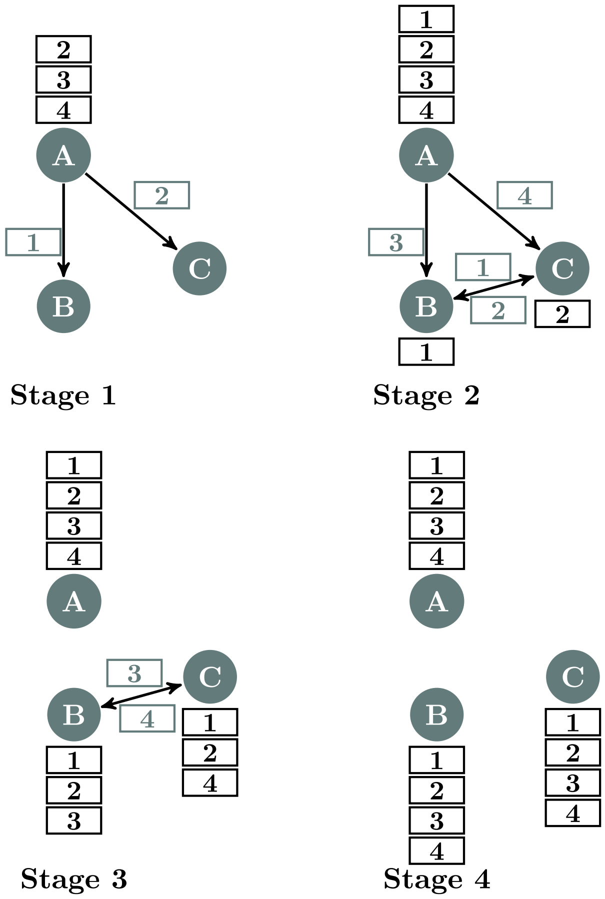


Figure 3-1: *An illustrative example of the evolution of data distribution in a P2P network.*

successfully or is considered lost. With this in mind, we conclude that in wireless networks, the choice of server node must reply to criteria that minimize the probability of error in the information provided, avoiding the need to retransmit damaged packages.

3.2 Strategies for selecting the server node

When all nodes are equal, the most used strategy is based on a *random selection* of the server node among those with the required information. This strategy can be efficient in some situations because it never uses one node as a server all the time, but it distributes the load between all the nodes randomly. However, this strategy is not always efficient because it does not consider any network factors, such as the length of every path, the load of every link, or the available bandwidth.

Some other well-known criteria are used for server selection. The *Min-Hop* strategy uses the number of hops as a metric. This strategy is very efficient in homogeneous networks, where the cost of every hop can be considered the same. However, it does not consider possible overload in some nodes, so it is not always efficient in networks that are not homogeneous, such as those with obstacles.

The Expected Transmission Count (ETX) [[Couto 2004]] is an easily implementable metric that measures the guarantee to deliver a package in one link successfully. Indeed, using this dynamic metric can introduce instabilities in the network, so it is recommended to apply it with caution. These instabilities are due to the modifications in the routing tables. When a link is overloaded, the routing protocol modifies the paths excluding this link, which quickly becomes unused. Thus, other links previously not overloaded will become overloaded. Furthermore, due to this behavior, the traffic continuously balances between different sections on the network, reducing the throughput.

We think that a fuzzy logic-based solution for the problem of selecting the server node could be appropriate. Using fuzzy logic to solve this problem acts as a compromise between different factors whose effect could be evaluated into a continuous $[0, 1]$ interval when used in the inference system. Also, a fuzzy solution will be less vulnerable to changes in the network.

In our study, the factors are the quality of the links and the length of the path. Short paths are interesting because they save resources used by other traffic. However, at the same time, it is necessary that the path can offer a high probability of success in the transmission of the information in order to avoid future retransmissions. In summary, we aim to balance link quality and the length of paths.

3.3 Related Works

Some studies using fuzzy logic to select the server node have also been developed. In [[Arom-oon and Keeratiwintakorn 2007]], the authors present a fuzzy system that takes the number of hops and the available energy in every node as inputs. The system goes after selecting the path with the most energy capacity to extend the time of life of the node, allowing fewer disconnections of the nodes. This aspect is good to consider, especially in ad-hoc sensor networks. However, it does not provide an interesting solution to networks using WIFI [IEEE 2012] because the consumption of their nodes is similar in both waiting and transmission states. In [Li et al. 2009b], [Li et al. 2009a], authors use Fuzzy Cognitive Maps (FCM) as the method of selection and analyze the following factors: level of energy in each node, number of hops, SINR (Signal to Interference and Noisy Ratio), time of connection, movement speed and security. The targets of these implementations are reducing the transference time, [Li et al. 2009b], and recovering the path when a node fails, [Li et al. 2009a]. The nodes do not present high mobility in current wireless mesh networks, so movement speed is not critical. In addition, SINR can change rapidly, so its use may produce some instabilities. In [Umezaki et al. 2012], authors propose a fuzzy system to select the server node depending on its degree of trustworthiness for file sharing; it uses two fuzzy controllers: the first one uses as inputs the number of jobs, the number of connections, and the connection lifetime, whose output is the Actual Behavioral Criterion; the second controller takes as inputs the Actual Behavioral Criterion and the Reputation of the node, obtaining as output the Peer Reliability. This solution is especially interesting when considering Wide Area Networks (WAN), where we can not control the connected peers. However, in Local Area Networks (LAN), where control over nodes is possible, it can be more interesting to prioritize other criteria that improve the throughput. Moreover, the time required to get the information is the most crucial parameter in a P2P network from a user perspective. Nevertheless, the majority of the previous strategies do not consider this aspect.

3.4 Proposed strategy: Fuzzy Inference based solution

We propose a *fuzzy inference-based system* to solve the problem of selecting the server node in Local Area wireless mesh networks. In general, a *fuzzy system* is any system whose at least some of its variables range over states that are fuzzy sets. For each variable, the fuzzy sets are defined on the real line for our application. Thus, the fuzzy sets are fuzzy numbers, and the associated variables are linguistic variables (section 2.4.8). We use the *fuzzy control*, the most successful application area of fuzzy systems.

A *Fuzzy Logic Controller* (FLC) is a special expert system. Expert systems are computer-based systems that emulate the reasoning process of a human expert within a specific domain of knowledge. They can be designed for various specific activities, such as diagnosis, design, and planning. The core of any expert system consists of

- a *knowledge base*, which contains general knowledge of the problem domain. In the fuzzy expert system, the knowledge is usually represented by a set of relevant fuzzy inferences rules connecting antecedents with consequences. We use the if-then rules which are the most commonly used,
- a *database* whose purpose is to store data for each specific task of the expert system. The data may be obtained through a dialog between the expert system and the user. In addition, the inference of the expert system may obtain other data,
- and an *inference engine* that operates on a serie of production rules and makes fuzzy inferences.

Fuzzy controllers vary substantially according to the nature of the control problems they are supposed to solve. The main difference between fuzzy with classical controllers lies in the capacity of the first for using knowledge elicited from human operators. This knowledge is crucial in control problems for which it is extremely challenging to build precise mathematical models or for which the acquired models are difficult or expensive to use.

Our proposed fuzzy system is an improved version of the so-called Purely Fuzzy system developed in [Valdés et al. 2013]. We compare the performance of our system respect to other ones using the most extended criteria: *random selection and min-hop selection.*

3.4.1 Description of the fuzzy strategy for server node selection in a P2P network

As we previously explained, a fuzzy Inference System (FIS) is an intelligent system that uses fuzzy set theory to map inputs to outputs according to a set of inference rules described by experts. The configuration of an FLC consists of four components:

- Fuzzification interface
- Knowledge base (fuzzy rule base)
- Decision-making logic (fuzzy inference engine)
- Defuzzification interface

Figure 3-2 shows a schema with the interconnections among the four modules and the controlled process.

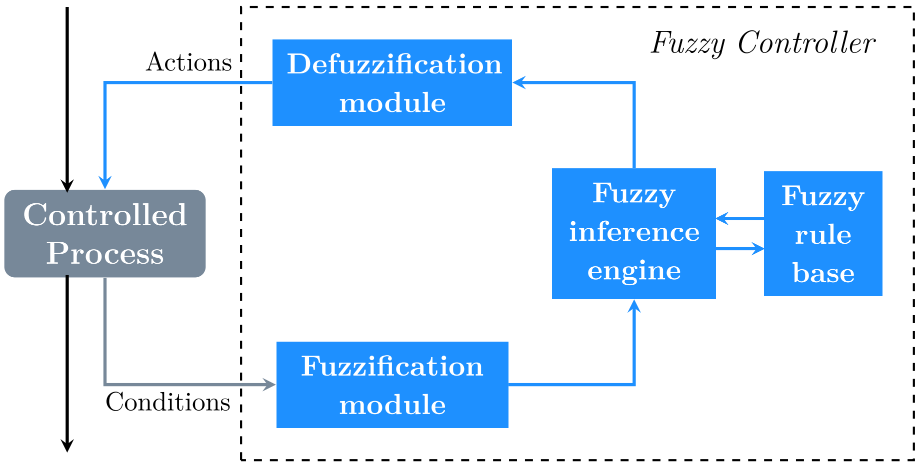


Figure 3-2: *General scheme of a Fuzzy Controller.*

Our FLC operates by repeating a cycle of five steps: First, the crisp measurements of the input variables are taken. The input and output variables are converted into appropriate fuzzy sets to express the uncertainty measurements through the fuzzification process. Then, the *knowledge* of our fuzzy system is formulated by the creation of fuzzy inference rules. The inference engine then uses the fuzzy measurements to evaluate the control rules stored in the fuzzy rule base. The result of this evaluation is also a fuzzy set defined on the universe of possible actions. In the final step of the cycle (defuzzification), this fuzzy set is converted into a single crisp value that is, in some sense, the best representative of the fuzzy set.

In the following, we will describe our proposal of fuzzy system for the server node selection in a P2P network.

Step 1: Fuzzification of input and output variables

The fuzzy system is designed to control a particular process. Therefore, we need to determine the state variables that become input variables and the control variables that become output variables. In every path, we take as inputs (state) variables,

- Number of hops (NHops)
- ETX metric (ETX)

and as a single output (control) variable,

- Goodness index of the server-client path (GPath)

We perform the fuzzification once the numerical values of input variables (NHops_0 and ETX_0) are entered. This process transforms the range of values of the input variables into the corresponding universe of discourse and selects meaningful linguistic states for each variable to express them by appropriate fuzzy numbers. Table 3.1 describes the ranges and linguistic terms for the inputs and output variables in our fuzzy system.

	Input Variables	Output variable
Ranges	NHops: $[0, 13]$ ETX: $[0, 4]$	GPath: $[0, 1]$
Linguistic terms	NHops: Low, Middle-Low, Middle-High, High ETX: Low, Middle-Low, Middle-High, High	GPath: Low, Middle-Low, Middle, Middle-High, High

Table 3.1: *Ranges and linguistic terms determined for the input and output variables in our fuzzy system.*

We represent the linguistic terms by triangular and trapezoidal fuzzy numbers equally spread over each range. Notice that, depending on specific applications, other shapes of the membership functions might be preferable to triangular or trapezoidal. Also, these shapes do not need to be symmetric nor equally spread over the given ranges. Nevertheless, we choose triangular and trapezoidal shaped membership functions as preliminary candidates due to their reasonable, intuitive definitions.

We can now state the input and output variable as linguistic variables (section 2.4.8):

Input variables:

$$\{\text{NHops}, T(\text{NHops}), X_{\text{NHops}}, G_{\text{NHops}}, m\}$$

where: NHops = Number of hops in the shortest "server-client" path for a given server.

$$T(\text{NHops}) = \{\text{NHLow, NHMiddel1, NHMiddle2, NHHigh}\}$$
$$X_{\text{NHops}} = [0, 14]$$
$$m_{\text{ETX}} = \{\mu_{\text{NHLow}}, \mu_{\text{NHMiddle1}}, \mu_{\text{NHMiddle2}}, \mu_{\text{NHHigh}}\}$$

$$\{\text{ETX}, T(\text{ETX}), X_{\text{ETX}}, G_{\text{ETX}}, m\}$$

where: ETX = ETX metric

$$T(\text{ETX}) = \{\text{ETXLow}, \text{ETXMiddle1}, \text{ETXMiddle2}, \text{ETXHigh}\}$$
$$X_{\text{ETX}} = [1, 4]$$
$$m_{\text{ETX}} = \{\mu_{\text{ETXLow}}, \mu_{\text{ETXMiddle1}}, \mu_{\text{ETXMiddle2}}, \mu_{\text{ETXHigh}}\}$$

The semantic functions, elements of $m(t)$, for NHops and ETX values are described in equation 3.2 and equation3.3, respectively.

$$\mu_{\text{NHLow}}(x) = \begin{cases} 1, & 0 \leq x < 1 \\ -0.25x + 1.25, & 1 \leq x < 5 \\ 0, & x \geq 5 \end{cases}$$

$$\mu_{\text{NHMiddle1}}(x) = \begin{cases} 0, & 0 \leq x < 1 \\ 0.25x - 0.25, & 1 \leq x < 5 \\ -0.25x + 2.25, & 5 \leq x < 9 \\ 0, & x \geq 9 \end{cases} \tag{3.2}$$

$$\mu_{\text{NHMiddle2}}(x) = \begin{cases} 0, & 0 \leq x < 5 \\ 0.25x - 1.25, & 5 \leq x < 9 \\ -0.25x + 3.25, & 9 \leq x < 13 \\ 0, & x \geq 13 \end{cases}$$

$$\mu_{\text{NHHigh}}(x) = \begin{cases} 0, & 0 \leq x < 9 \\ 0.25x - 2.25, & 9 \leq x < 13 \\ 1, & x \geq 13 \end{cases}$$

$$\mu_{\text{ETXLow}}(x) = \begin{cases} 1, & 0 \leq x < 1 \\ -x + 2, & 1 \leq x < 2 \\ 0, & x \geq 2 \end{cases}$$

$$\mu_{\text{ETXMiddle1}}(x) = \begin{cases} 0, & 0 \leq x < 1 \\ x - 1, & 1 \leq x < 2 \\ -x + 3, & 2 \leq x < 3 \\ 0, & x \geq 3 \end{cases} \tag{3.3}$$

$$\mu_{\text{ETXMiddle2}}(x) = \begin{cases} 0, & 0 \leq x < 2 \\ x - 2, & 2 \leq x < 3 \\ -x + 4, & 3 \leq x < 4 \\ 0, & x \geq 4 \end{cases}$$

$$\mu_{\text{ETXHigh}}(x) = \begin{cases} 0, & 0 \leq x < 3 \\ x - 3, & 3 \leq x < 4 \\ 1, & x \geq 4 \end{cases}$$

Figures 3-3 and 3-4 show the membership functions corresponding to input variables.

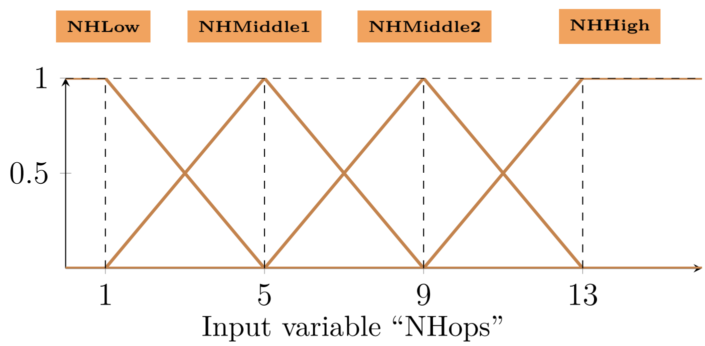


Figure 3-3: *Fuzzy numbers corresponding to each linguistic term in $T(NHops)$.*

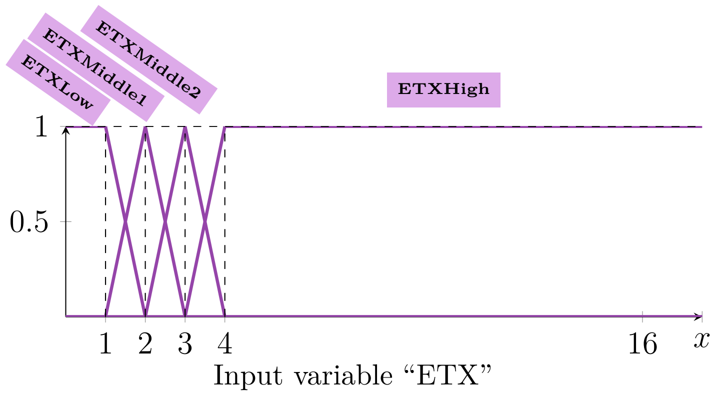


Figure 3-4: *Fuzzy numbers corresponding to each linguistic term in $T(ETX)$.*

Output Variable:

$$\{\mathrm{GPath}, T(\mathrm{GPath}), X_{\mathrm{GPath}}, G_{\mathrm{GPath}}, m\}$$

where:

$$\mathrm{GPath} = \text{Goodness index of server-client path}$$

$$T(\mathrm{GPath}) = \{\mathrm{GPLow}, \mathrm{GPMiddleL}, \mathrm{GPMiddle}, \mathrm{GPMiddleH}, \mathrm{GPHigh}\}$$

$$U_{\mathrm{GPath}} = [0, 1]$$

$$m_{\mathrm{ETX}} = \{\mu_{\mathrm{GPLow}}, \mu_{\mathrm{GPMiddleL}}, \mu_{\mathrm{GPMiddle}}, \mu_{\mathrm{GPMiddleH}}, \mu_{\mathrm{GPHigh}}\}$$

We describe the semantic functions for the output variable in equation 3.4,

$$\mu_{\text{GPLow}}(x) = \begin{cases} 1, & 0 \le x < 0.2 \\ -10x+3, & 0.2 \le x < 0.3 \\ 0, & 0.3 \le x \end{cases}$$

$$\mu_{\text{GPMiddleL}}(x) = \begin{cases} 0, & 0 \le x < 0.2 \\ 10x-2, & 0.2 \le x < 0.3 \\ -10x+4, & 0.3 \le x < 0.4 \\ 0, & 0.4 \le x \end{cases}$$

$$\mu_{\text{GPMiddle}}(x) = \begin{cases} 0, & 0 \le x < 0.3 \\ 10x-3, & 0.3 \le x < 0.4 \\ 1, & 0.4 \le x < 0.6 \\ -10x+7, & 0.6 \le x < 0.7 \\ 0, & 0.7 \le x \end{cases} \tag{3.4}$$

$$\mu_{\text{GPMiddleH}}(x) = \begin{cases} 0, & 0 \le x < 0.6 \\ 10x-6, & 0.6 \le x < 0.7 \\ -10x+8, & 0.7 \le x < 0.8 \\ 0, & 0.8 \le x \end{cases}$$

$$\mu_{\text{GPHigh}}(x) = \begin{cases} 0, & 0 \le x < 0.7 \\ 10x-7, & 0.7 \le x < 0.8 \\ 1, & 0.8 \le x \end{cases}$$

Figure 3-5 shows the membership function of the output variable,

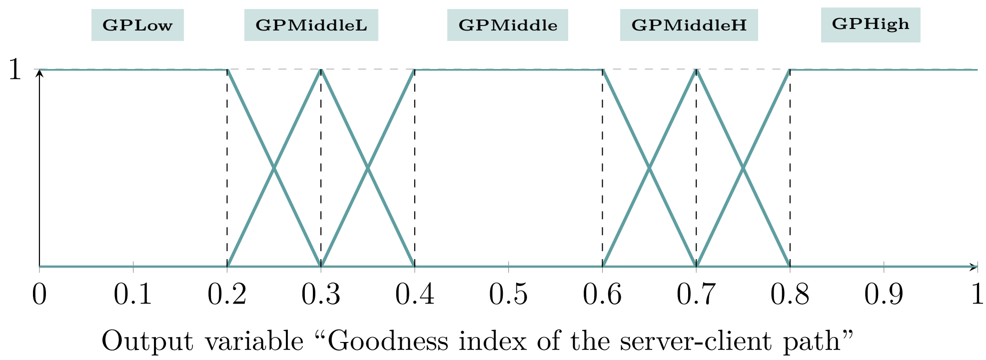


Figure 3-5: *Fuzzy numbers corresponding to each linguistic term in $T(GPath)$.*

Step 2: Formulation of fuzzy inference rules

With the fuzzy inference rules we formulate the knowledge of the problem. Suitable learning methods can determine the inference rules from experienced human operators or empirical data.

The rule base of the two-inputs/single-output system (TISO) that we use contains I fuzzy if-then rules whose antecedent is compound by the connector "AND" (equation 2.31), i.e.,

$$\mathrm{R}_i^{\mathrm{TISO}}: \quad \text{if}\, x \text{ is } \tilde{A} \text{ and } y \text{ is } \tilde{B}, \text{ then } z \text{ is } \tilde{C} \tag{3.5}$$

where $i = \overline{1, I}$, and $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ are fuzzy sets that represent the linguistic terms NHLow, NHMiddel1, NHMiddle2, NHHigh; ETXLow, ETXMiddle1, ETXMiddle2 and ETXHigh, and GPLow, GPMiddleL, GPMiddle, GPMiddleH, GPHigh, respectively.

We use *state evaluation fuzzy control rules*, where the state variables are in the antecedent part of each rule, and the control variable is in the consequent part. In linguistic terms, we interpret each rule as: if performance index x is $\tilde{A}_i$ and index y is $\tilde{B}_i$ when a control command z_i is $\tilde{C}_i$, then this rule is selected, and the control command $\tilde{C}_i$ is the controller's output.

The number of linguistic terms in the input space determines the maximum number of fuzzy control rules. Since each input variable has four linguistic terms, the total number of possible nonconflicting fuzzy inference rules is $4^2 = 16$. We use expert experience and control engineering knowledge (operating manual and questionnaire) to obtain the rules. We represent the rules conveniently in a matrix form in table 3.2.

<table>
<tr><td></td><td></td><td colspan="4">NHops</td></tr>
<tr><td></td><td>GPath</td><td>NHLow</td><td>NHMiddle1</td><td>NHMiddle2</td><td>NHHigh</td></tr>
<tr><td rowspan="4">ETX</td><td>ETXLow</td><td rowspan="2">GPHigh</td><td>GPMiddle-High</td><td rowspan="2">GPMiddle-Low</td><td rowspan="4"></td></tr>
<tr><td>ETXMiddle1</td><td>GPMiddle</td></tr>
<tr><td>ETXMiddle2</td><td>GPMiddle-High</td><td>GPMiddle-Low</td><td rowspan="2">GPLow</td></tr>
<tr><td>ETXHigh</td><td></td><td></td></tr>
</table>

Table 3.2: *Fuzzy rule base in our system.*

We also write the fuzzy rules in the if-then form,

1. If (NHops is Low) AND (ETX is Low) THEN (GPath is High)
2. If (NHops is Low) AND (ETX is Middle1) THEN (GPath is High)
3. If (NHops is Low) AND (ETX is Middle2) THEN (GPath is MiddleHigh)

4. If (NHops is Low) AND (ETX is High) THEN (GPath is Low)
5. If (NHops is Middle1) AND (ETX is Low) THEN (GPath is MiddleHigh)
6. If (NHops is Middle1) AND (ETX is Middle1) THEN (GPath is Middle)
7. If (NHops is Middle1) AND (ETX is Middle2) THEN (GPath is MiddleLow)
8. If (NHops is Middle1) AND (ETX is High) THEN (GPath is Low)
9. If (NHops is Middle2) AND ETX is (Low) THEN (GPath is MiddleLow)
10. If (NHops is Middle2) AND (ETX is Middle1) THEN (GPath is MiddleLow)
11. If (NHops is Middle2) AND (ETX is Middle2) THEN (GPath is (Low)
12. If (NHops is Middle2) AND (ETX is High) THEN (GPath is Low)
13. If (NHops is High) AND (ETX is Low) THEN (GPath is Low)
14. If (NHops is High) AND (ETX is Middle1) THEN (GPath is Low)
15. If (NHops is High) AND (ETX is Middle2) THEN (GPath is Low)
16. If (NHops is High) AND (ETX is High) THEN (GPath is Low)

The matrix representation of rules 3.2 and the definitions of the linguistic terms shown in figures 3-3, 3-4, and 3-5 form the fuzzy rule base of our fuzzy controller.

The connector "AND" composes the compound antecedent of each rule. Therefore, at first, the domain change is made in each antecedent's part separately, and then both parts are related by the operator MIN. Then, the grade of truth (matching degree) for each rule is calculated according to equation 2.35. The matching degree for each rule R_i, with $i = \overline{1, 16}$, is calculated according to equation 3.6,

$$\alpha_i(\mathrm{NHops}_0, \mathrm{ETX}_0) = \min\left\{\mu_{\tilde{A}}(\mathrm{NHops}_0), \mu_{\tilde{B}}(\mathrm{ETX}_0)\right\} \tag{3.6}$$

At this point, our system is ready to perform the inference engine.

Step 3: Inference Engine

Implication: The purpose of the inference engine is to combine the numerical values of the input variables of our fuzzy inference system (NHops_0, ETX_0) with the fuzzy rules to make inferences regarding the output variable.

Once the compound antecedent of each rule R_i has provided its matching degree α_i, we apply the implication process. This method works on the output fuzzy set, depending on the matching degree given by the antecedent, to generate the result of each rule. In our case, we use the Mamdani method (truncating the output fuzzy set), whose formula corresponds to equation 2.36. Thus, when applying the Mamdani method to each rule R_i of our fuzzy system, we obtain the expression 3.7.

$$\mu_{\tilde{C}'_i}(\mathrm{GPath}) = \mu_{[(\tilde{A}\cap\tilde{B})\to\tilde{C}]_i}(\mathrm{GPath}) = \min\left\{\alpha_i(\mathrm{NHops}_0, \mathrm{ETX}_0), \mu_{\tilde{C}}(\mathrm{GPath})\right\} \tag{3.7}$$

Combination: Once each rule R_i provides its output, this method combines the rules to produce the total inference of the rule base, that is, to make a decision. Let $\mu_{\tilde{C}'_i}$ be the set of all the output functions created in the Implication process of each rule. Since we interpret each rule as disjunctive (we obtain a conclusion whenever the grade of truth is positive for at least one rule), we use the maximum value resultant of composing the output functions of each rule described in equation 3.8,

$$\mu_{\tilde{C}'} = \max_{i\in[1,16]} \left\{ \mu_{\tilde{C}'_i} \right\} \tag{3.8}$$

Step 4: Defuzzification

The defuzzification is the last step of the design process. This process aims to convert the output obtained by the inference engine, which is expressed in terms of a fuzzy set, to a single real number. This numerical value is not arbitrary but must, in some sense, summarize the elastic constraint imposed on possible values of the output variable by the fuzzy set. There are three commonly used defuzzification methods proposed in the literature: Center of Area method (CoA), also known as Centroid Method or Center of Gravity, Center of Maximum method (CoM), and Mean of Maximum method (MoM).

Our analysis applies the CoA, the most frequently used in the simple fuzzy controller. This method calculated z_0, defined as the value within the range of the variable *GPath* for which the area under the graph of membership function $\mu_{\tilde{C}'}$ is divided into two equal subareas (center of gravity of $\mu_{\tilde{C}'}$). In our case, z_0 is generated by the formula 3.9,

$$z_0 = \frac{\int_0^1 z\mu_{\tilde{C}'}(z)\mathrm{d}z}{\int_0^1 \mu_{\tilde{C}'}(z)\mathrm{d}z} \tag{3.9}$$

3.5 Experimental Environment

Using the framework inetmanet-2.2 [Ariza 2014], we simulate the strategies: *random selection* of server node (Random), *minimum number of hops* between server and client nodes (Min-Hop), and our *fuzzy inference based system* (Purely fuzzy). We also simulate IP networks on OMNET++[1] [Varga 2014]. Also, we developed two dierent environments to compare the performance of the three strategies under analysis: a network without obstacles between nodes and another one with obstacles.

We start with a fixed number of server nodes containing all the information,

[1]OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators.

which provides a series of datum segments with a predetermined size, which has to be distributed to all network nodes. Our simulations end when all nodes have all segments. These segments may be further divided into smaller units to be introduced into IP packets without fragmentation. To simplify the implementation (without loss of generality), we assume that all nodes know the network's status at every moment and the information available in other nodes. This assumption is reasonably viable in a small LAN.

When a client node has to choose among several server nodes, the analysis of each one is carried out on the path with the minimum number of hops for each server-client pair since this criterion forces stable routes throughout the time. Thus, this factor has been explicitly forced to prevent the dynamic selection of paths that influence our study.

We use a wireless network based on the wireless mesh extension present in the IEEE 802.11-2012[2], where the routing and forwarding mechanisms are implemented at link level [IEEE 2012]. Table 3.3 shows the simulation conditions.

The environment we use in both experiments consists of a regular 8x8 (64 nodes) squared network, in which we implemented the strategies for the server node selection.

3.6 Simulation and results

3.6.1 Information distribution in a regular network without obstacles

In this experiment, we consider a network without obstacles. We performed three different simulations in which a different number of nodes (one node in the first simulation, two nodes in the second one, and three nodes in the third one), randomly selected, have all the information segments. Each client node starts making requests in an instant of time randomly set. The simulation ends when all the nodes get all the information segments. Each simulation is repeated ten times with different seeds to select the initial server nodes.

In order to study the performance of the different strategies for server selection, we analyzed three parameters in each simulation:

- the download time that every node spends to get all the segments of the information (average and maximum),
- the number of bytes that are sent in the application-level by node (average and maximum),

[2] IEEE 802.11 is used in most home and office networks to allow laptops, printers, smartphones, and other devices to communicate with each other and access the Internet without connecting wires.

Simulation area	1000 × 1000 m^2
Nodes in the backbone network	64
Maximum transmission distance	130m
Propagation model	Two ray
Separation between nodes in the backbone network	80m
Simulation period	Simulation end when all nodes have the complete information
Interference model	Additive
WIFI model	802.11g
Bit rate	54Mbit/s
Number of segments to be transmitted	10
Size of the segments	100000B
Maximum packet size	1000B
Number of repetitions with different seeds	10
Confidence interval	95%
Routing protocol	OLSR ([Clausen and Jacquet 2003])

Table 3.3: *Simulation conditions.*

- and the number of bytes sent by a node at the network level (average and maximum).

Since the download time is a measure of the service quality perceived by the users, we consider it the most important parameter. Furthermore, the number of bytes sent by a server node at the application level measures the size of downloaded data. Thus, it can give information about the punctual overload in some nodes. Finally, the number of bytes sent by a node at the network-level is a measure of the real total load traffic in the network, so it is an essential parameter for the scalability of the network.

The results obtained from the different situations are very similar in value and distribution. Therefore, we finally show the average of each. Table 3.4 and figure 3-6 show the numerical and graphical results of the download time, respectively.

According to these results we can realize that both the average time and the

	Random	Min-hop	Fuzzy
Avg. Download Time (in secs)	212.6	145.05	145.03
Confidence Interval	± 2.49	± 3.05	±3.01
Max. Download Time (in secs)	315.86	283.56	283.33
Confidence Interval	±10.06	±9.91	±9.98

Table 3.4: *Download time in every node (in seconds) to get all the information segments.*

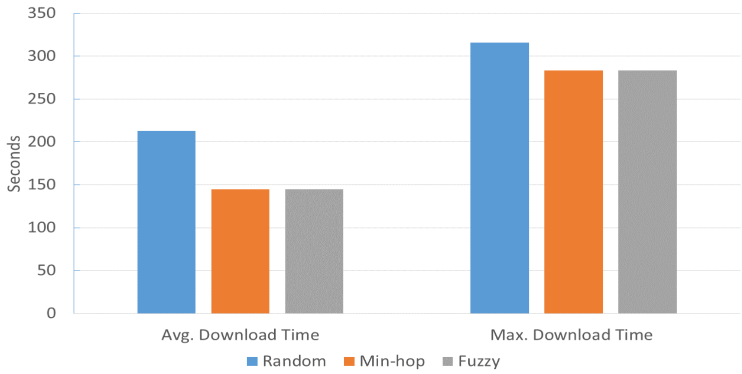


Figure 3-6: *Download time in every node (in seconds) to get all the information segments.*

maximal time required for nodes to get the information is clearly larger for the Random strategy, whereas Min-Hop and Purely Fuzzy strategies do not present relevant differences.

Table 3.5 and Figure 3-7 show the numerical and graphical results for the sent bytes at the application level.

	Random	Min-hop	Fuzzy
Sent Bytes (average)	1203768	1000860	1001271
Confidence Interval	± 18191	± 1442	±1446
Sent bytes (Maximum)	3873609	4926555	4808877
Confidence Interval	±321168	±443056	±516119

Table 3.5: *Number of bytes sent by a node at the application-level.*

We observe that the Random strategy reduces the maximal number of bytes sent by a node. As a result, however, it produces a larger number of total sent bytes. This result is reasonable since: on the one hand, a random selection does not charge any particular node, but it distributes requests in an aleatory way; on

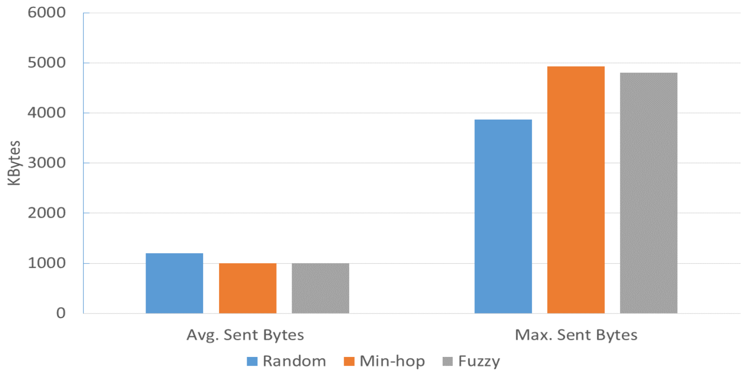


Figure 3-7: *Number of application-level sent bytes, average and maximum.*

the other hand, the server-client paths are longer in this strategy, so enlarges the number of lost packets and, consequently, the average number of sent bytes.

Table 3.6 and Figure 3-8 show numerical and graphical results for the sent bytes at the network-level.

	Random	Min-hop	Fuzzy
Avg. Sent Bytes	4231654	1295977	1299659
Confidence Interval	± 47147	± 46817	±47147
Max. Sent Bytes	12497080	5304649	5243647
Confidence Interval	±674741	±506359	±583006

Table 3.6: *Number of network-level sent bytes by each node, average and maximum*

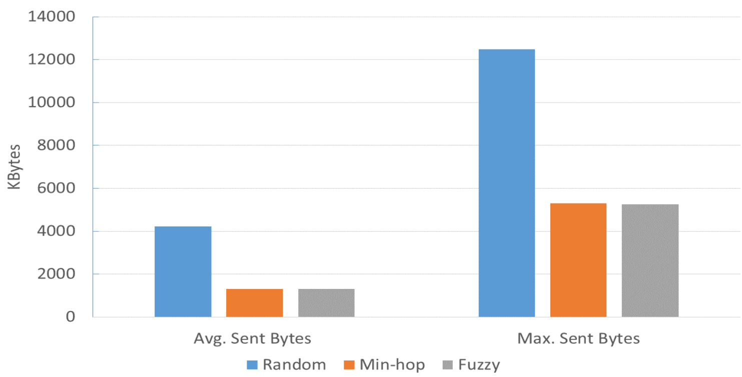


Figure 3-8: *Number of network-level sent bytes, average and maximum*

These results show that the Random strategy is the least efficient because its number of network-level sent bytes is the highest. Notice that this parameter represents the real traffic load in the network, where a packet sent from the server to the client will be re-sent by each of the intermediate nodes in the path.

Summarizing the conclusion in the first set of experiments we have:

Random selection strategy is the least efficient concerning the required transmission time and network-level traffic load. We must consider that these parameters are the most important ones for network users and administrators. There are no crucial differences between Min-Hop and Purely Fuzzy strategies in our regular network because the Min-Hop strategy is very efficient in this kind of network, so the impact of fuzzy logic cannot be shown entirely in this experiment.

3.6.2 Distribution of information in a regular network with obstacles

This experiment analyzes the effect on the network service quality for the three analyzed strategies when we introduce an obstacle. This obstacle will force a fixed packet lost probability.

We keep the same network structure from the previous experiment, but we analyze a specific situation in which two peers have all the segments (possible servers) and one peer requests for these segments (client). One of the nodes with the information is located one hop from the client node behind the obstacle, and the other node is located two hops from the client peer in a path without obstacles. We studied this situation with three nodes instead of analyzing the entire network because, in this last case, we would not appreciate the impact of the obstacle properly. We do not consider the other nodes, so they do not request any information from the simulation. We set to 50% the probability of losing a packet in a link with an obstacle.

In this simulation, we analyze the time required for the client node to get the complete information (the other parameters have been omitted because they are not significant with a single client). Table 3.7 and Figure 3-9 show the numerical and graphical results.

	Random	Min-hop	Fuzzy
Download Time (in secs)	159	407.5	119.3
Confidence Interval	± 14.44	± 25.15	± 2.3

Table 3.7: *Download time in every node to get all the segments of the information.*

We observe that the Purely Fuzzy strategy produces the best results concerning the download time for a node when obstacles are present. On the other hand, we can also realize that, in cases in which obstacles appear, the Min-Hop strategy is the least efficient because it does not consider the real state of the network but only the number of hops.

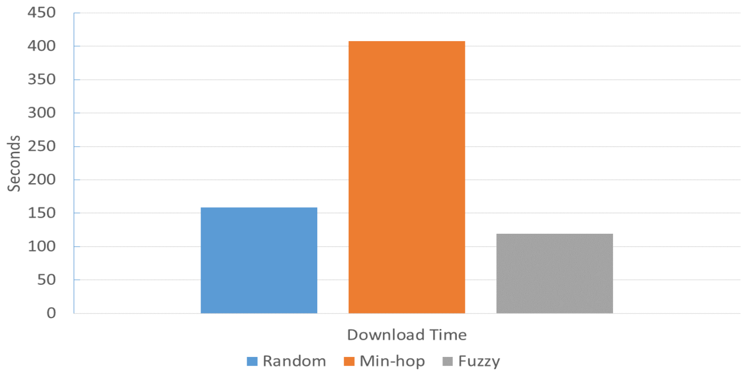


Figure 3-9: *Download time in every node (in seconds) to get all the segments of the information*

3.7 Conclusions

This work is a considerable extension of the content and significance of a previous study that we performed [Valdés et al. 2013]. We analyze the performance of three strategies to select the server node in a wireless P2P network. The first one is the random strategy which probably is the currently more often implemented one and carries out the selection randomly regardless of the distance between client and server. The second one, called the Min-Hop strategy, uses the minimum number of hops as a selection criterion between server and client. Finally, the third strategy, called Purely Fuzzy, is an improved version of that presented in [Valdés et al. 2013], and it selects the best server using a fuzzy logic-based system considering as input variables the Number of Hops and ETX of each path.

We perform the study using a wireless network simulation tool for discrete even simulator OMNEeT++. We implemented an 8x8 node regular network in which we consider two different scenarios: a network without obstacles and a network with obstacles between nodes. We realize a significant number of simulations to obtain results related to the download time and the traffic load at both the application and network levels.

Based on the results that we obtain, we conclude that the Purely Fuzzy logic is the most effective strategy because it adapts to all network situations (with and without obstacles). Besides, although the random strategy works better than Min-Hop in concrete circumstances, it is the least efficient in most cases because it does not consider any factor of the network. Finally, we must consider that the most important factor considered by users is the total download time; so, in this aspect, we conclude that the fuzzy strategy is the most efficient one considering the general conditions of networks.

We must assume that most of the parameters and hyperparameters intervening in the definition of a FIS (parameters for the definition of the membership functions of each fuzzy value, number of fuzzy values associated with each fuzzy variable, set of rules, etc.) are chosen empirically. Hence, it isn't easy to guarantee that the system will perform optimally. However, we are aware Fuzzy Inference Systems are quite robust concerning these possible variations. Therefore, the behavior shown by different reasonably defined models will probably give very similar results in terms of efficiency.

CHAPTER 4

SEARCH OF THE SHORTEST PATH IN A COMMUNICATION NETWORK WITH FUZZY COST FUNCTIONS

- LISSETTE VALDÉS, ALFONSO ARIZA, SIRA M. ALLENDE, ALICIA TRIVIÑO, GONZALO JOYA; "Search of the shortest path in a communication network with fuzzy cost functions"; In: *Symmetry. Manuscript ID: symmetry-1281387*; Vol. 13(8); 1534; (2021); https://doi.org/10.3390/sym13081534.

A Communication Network Management System takes the measurements of its state variables at specific instants of time, considering them constant in the interval between two consecutive measurements. This assumption is, nevertheless, not valid since these variables evolve in real time. Therefore, uncertainty is introduced into the measurements that cannot be efficiently managed using crisp variables or using a control based on fuzzy inference models. In this chapter, we face this problem by modeling the communications network as type V fuzzy graph, where both the nodes and the links are described with precision, but the cost of each link is modeled as a triangular fuzzy number. Different fuzzy cost allocation functions and fuzzy optimization strategies are described and applied to the search for the shortest path between two nodes. An experimental study has been conducted using the backbone network of Nippon Telegraph and Telephone Corporation as a reference, where our fuzzy cost functions and strategies have been compared with the well-known crisp equivalents. The results show that, the fuzzy alternatives surpass the crisp equivalents with statistically significant values in all cases. Specifically, the so-called Strategy 8, proposed here for the first time, presents the best throughput, significantly exceeding the performance of all those evaluated, achieving a Global Mean Delivery Rate (GMDR) close to 1. The optimal search strategies are based on a Dijkstra algorithm adapted to the fuzzy case where the comparison between triangular fuzzy numbers is made through their total integrals.

4.1 Introduction

The management of a communication network mainly aims to maximize the ratio between sent and received information for any pair of source-destination nodes, reducing transmission delays. This objective is especially relevant in high occupancy conditions. Of course, this concept can be extended to other networks such as paths and commercial distributions. Necessary resources to achieve this objective are algorithms and routing protocols, which seek to provide the optimal path between any pair of source-destination nodes. It is crucial to know the network's state by assigning a cost to every link to evaluate the path's goodness level. In this way, we compute the cost of a particular path using variables such as *Used Bandwidth*, *Residual Bandwidth (non-used bandwidth)*, *Packet Delivery Ratio (ratio between delivered and sent packets)*, *Packet Loss Ratio (1 - the Packet Delivery Ratio)*, *Packet Delay (time spent from the creation to the delivery of a packet)*, etc. From this knowledge, we can establish the routing tables, which contain the path communicating nodes for the demanded information transmission at each moment.

Because of the network's dynamic behavior, and consequently, the high variability in time of its performing conditions, the usual way of working is to compute the state variables over a given time interval using measurements obtained in the immediately previous time interval. So, typical routing protocols update the state (cost) of links in two possible ways, [Ariza et al. 2000]:

(A) By time intervals: Each cost value (and, therefore, the routing tables) is periodically updated with a fixed periodicity. This update is either from the instantaneous state value at the beginning of the period (the beginning of the new interval) or from the mean value of costs over the previous time interval.

(B) By thresholds: Some update thresholds are fixed so that the actualization of a state variable is carried out when the difference between its current real value and the last determined exceeds the corresponding threshold.

It is evident that, although these used cost values are well-defined real numbers (crisp numbers in our terminology), these procedures introduce a degree of uncertainty about the current state due to the existence of a real probability that variables have changed over the time interval. Consequently, we believe that considering this uncertainty into the decision-making process on the management should allow us to find better solutions. In this sense, some elements of Fuzzy Logic, particularly the application of arithmetic operations and properties of fuzzy numbers, can be helpful. We first interpret a triangular fuzzy number as a triplet of real numbers as a first approach. The numbers at the ends delimit the interval of values assigned to the fuzzy number, and the central number is the value considered more probable.

An essential aspect of our work is representing the cost measurements using fuzzy numbers, which reasonably incorporate this uncertainty. For this objective, option (B) has the apparent advantage that, when fixing the thresholds, we are also setting the upper and lower limits of the corresponding triangular fuzzy numbers. However, the central value of this fuzzy number, which is theoretically considered the more probable value, does not have a physical justification since it responds to an instantaneous value from which we have no information about its feasibility. Also, if the central value is given, we compute the upper and lower values by adding and subtracting the threshold. As a result, the crisp number corresponding to this fuzzy cost function will always be constant and equal to the central value. Therefore, the sense of fuzzification is lost.

On the other hand, option (A) does not directly provide the extreme values of the *normalized* triangular fuzzy number. However, it allows us to define the triangular fuzzy number by using statistical methods in some way consistent with the physics of the problem. Thus, option (A) incorporates the fuzzy nature of the problem.

Thus, in our case, the proposed fuzzy numbers for the time interval t are defined as follows, the central value corresponds to the mean value of the costs over the entire $t-1$ time interval (most "probable" value). The extreme values are the maximum and minimum cost measurements in the above-mentioned $t-1$ time interval.

Our proposal models the network as a *non-directed and weighted type V fuzzy graph*, where nodes and links are established with clarity, but the cost functions on links are defined as triangular fuzzy numbers. In particular, to represent these costs, we use the variable called *Normalized Used Bandwidth* [Ariza 2001], which is defined in equation 4.1,

$$bw_{ij}(t) = \frac{BW_{ij}(t)}{C_{ij}} \frac{C_{\max}}{C_{ij}} \tag{4.1}$$

where $BW_{ij}[t]$ is the used bandwidth in the link (i, j), C_{ij} is the total capacity (bandwidth) of the link (i, j), and $C_{\max}$ is the link's capacity with the larger bandwidth in the network. Expression 4.1 weights the Used Bandwidth ratio, taking into account the own capacity of each link.

At last, the application of a fuzzy version of the Dijkstra algorithm, like the shortest path search algorithm, should give us the path between any pair of source-destination nodes with the minimum cost. Moreover, this fuzzy version of the Dijkstra algorithm can be considered classical since it does not incorporate any uncertainty consideration on its execution, but in the value of the variables (here considered as fuzzy) and in the definition of the mathematical operation between them. Therefore, we are interested in highlighting the differences between this methodology and the Fuzzy Inference proposed in chapter 3 ([Valdés et al. 2013], [Valdés et al. 2015] [Shamshirband et al. 2016]) where the measurements are exact values, and the uncertainty lies in the decision process itself.

To summarize, the main contribution of this work is the presentation of a methodology to optimize the search for the shortest path between two nodes during the global management of a communication network. This methodology is characterized by modeling the network on a Non-directed Type V Fuzzy Graph and applying a fuzzy Dijkstra algorithm adapted to operate with fuzzy numbers ([Valdés et al. 2021]). To show the goodness of our approach, we have two objectives:

i) To show the competitiveness of the methodology based on the fuzzy model versus that based on the traditional model using crisp magnitudes. To do this, we reconstruct the currently used classical cost functions and strategies for searching the optimal path to the fuzzy model. Then, we compare both visions (classical and fuzzy) in the same experimental environment. The result shows that the fuzzy model is more efficient (or equal in the worst case) with statistical validation.

ii) To illustrate that the fuzzy model allows us to define new metrics that take advantage of the experimental uncertainty mentioned above. Thus, we provide a new strategy and its cost function (referred to as Strategy 8), which has no equivalent in the classic model and by far exceeds in effectiveness all the other strategies analyzed.

Thus, the organization of this chapter is as follows: We devote section 4.2 to compile some related works from the points of view of both the application in the communication environment and the use of Fuzzy Logic. In section 4.3, we describe the fuzzy model of the network and the definition of various types of fuzzy numbers and their possible interpretation. Section 4.4 introduces our proposal of the Fuzzy Dijkstra Algorithm applied on a network with fuzzy costs using the Total Integral Method proposed in [Yu and l. Q. Dat 2014] as the ranking procedure for fuzzy numbers. Section 4.5 presents the application of these theoretical concepts to a communication network with a description of the communication network used, the different cost functions used in the comparison, and the different strategies to search the optimal path (with a particular interest in Strategy 8). Section 4.6 deals with our experimental study, where the reader finds the experimental results and their discussion. Finally, Section 4.7 summarizes the main and conclusive ideas of this work together with future lines of work.

4.2 Related works

Concerning classic strategies (based on crisp magnitudes) used for the search of the shortest (or more efficient) path in a communication network, we consider some approaches to be among the most known and used at the moment. These are the following: the Shortest-Widest (SW) strategy presented in [Wang and Crowcroft 1996] searches for the path with the largest residual bandwidth path (that is, the more efficient path). If there are two paths with the same residual bandwidth, the path with the minimum number of hops is selected. The

Widest-Shortest (WS) strategy, explained in [Guerin et al. 1997], proceeds contrary to SW. WS searches for the path with the minimum number of hops, and in case there are two paths with the same number, the one with maximum residual bandwidth is selected. [Ma et al. 1996] and [Ma and Steenkiste 1997] describe the Shortest Dist Path strategy using a hyperbolic cost function based on the inverse of the residual bandwidth, where the authors perform an exhaustive comparison between SW, WS, and Shortest Dist Path algorithm. In the case of traffic with no reserve of resources, the Shortest dist Path algorithm is clearly advantageous, especially under overload conditions. In [Tomovic et al. 2015], to find the optimum path in the sense of reducing the Bandwidth Rejection Ratio (BRR), the authors perform a comparison of cost functions applied to software-defined networks. Based on a cost function that uses a Shortest Dist Path algorithm variant, the algorithm DORA 0.9 gives the best results in [Boutaba et al. 2002].

Other approaches to the optimal path search problem in a communications network are based on the adaptation of control techniques based on fuzzy inference. In this case, the values of the cost functions on the links are crisp, but the uncertainty applies to the inference process itself. Thus, in [Valdés et al. 2015], authors develop a fuzzy inference system using the Expected Transmission Time (ETT) and the Number of Hops as input variables to find the best server node in a P2P wireless mesh. The fuzzification of these variables and their introduction in the fuzzy decision rules allow us to outperform the efficiency of traditional methods, which select the server node based on a min-hop or random criteria. In [Umezaki et al. 2012], the server node is selected using its degree of trustworthiness for file-sharing as a control variable. In [Tian et al. 2017], the authors propose a fuzzy inference system to find the path in a sensor network considering the remaining energy, the minimum number of hops, and the node traffic load.

An alternative fuzzy approach to network modeling assumes that the uncertainty lies in the value of the cost functions. This uncertainty can be modeled by using fuzzy numbers. In these cases, the application of a classic path search algorithm such as the Dijkstra Algorithm involves solving the problem of operating with fuzzy numbers (e.g., adding up the fuzzy costs of the links in a path) and comparing fuzzy numbers (e.g., the comparison between the total cost of two paths). In [Chou 2003], the Graded Mean Integration Representation definitions and the expression for the sum and product of two (triangular or trapezoidal) fuzzy numbers are developed with particular mathematical rigor. The authors use an integral representation allowing the operation and comparison of fuzzy numbers. [Deng et al. 2012] and [Mullai 2016] use this representation to adapt the Dijkstra algorithm to fuzzy costs in a generical transportation network.

We base our proposal on modeling the network by assigning a triangular fuzzy number to the costs of the links. To compare triangular fuzzy numbers, we use a generalized definition for the *Total Integral of a fuzzy number*, proposed in [Yu and l. Q. Dat 2014], which is different from the above-mentioned Graded Mean Integration Representation. The different metrics discussed above (WS, SW, Normalized Used Bandwidth) are adapted to the definition of triangular fuzzy number and compared with the classic versions on the simulated environment of a communication network, based on the topology of the NTT network. Our fuzzy versions generally surpass or equal the crisp variants in all cases by a small statistically significant margin. Also, a new cost function based on the general model of the Total Integral of a triangular fuzzy number is developed and given a physical explanation. Our optimization strategy based on this latter function surpasses the other tested algorithms, resulting from a Delivery Bit Ratio close to 1.

4.3 Fuzzy modeling of a communication network

Without loss of generality, we will refer to the cost to the weight of a link because it makes more sense using this term in our application described in this chapter.

We assign a Type V fuzzy graph $\tilde{G}$ to our network. For the definition of a Type V fuzzy graph, we refer the reader to section 2.4.7. The associated graph $\tilde{G}$ is then defined as:

- $\tilde{G} = (V, E, \mathfrak{C})$ is a triplet where,
  - V is the set of vertices
  - E is the set of edges
  - $\mathfrak{C}$ is the set of costs. $\tilde{C}_{ij} \in \mathfrak{C}$: fuzzy cost of edge (i, j), $i, j \in V$
- $\tilde{G}$ is connected; that is, there is a path between each pair of different nodes in $\tilde{G}$.
- To each edge $e_{ij} = (i, j)$, $i, j \in V$, $e_{ij} \in E$ corresponds a cost $\tilde{C}_{ij} \in \mathfrak{C}$ defined as a triangular fuzzy number (in detail explained in section 2.4.6). We refer to $\tilde{C}_{ij}$ as the *fuzzy cost* of edge e_{ij}. The cost of path P, denoted as $\tilde{C}_P$ (fuzzy cost of path P), is a function of the cost of its constituent edges, i.e.:

$$\tilde{C}_P = f(\tilde{C}_{e_{ij}}), \quad \forall e_{ij} \in P$$

4.4 Dijkstra Algorithm for type V fuzzy graph

We propose a *Fuzzy Dijkstra Algorithm* (FDA) applied to a type V fuzzy graph. This algorithm finds the shortest path between the source vertex r and any other vertex on $\tilde{G}$, and deals with the fuzzy costs defining these as triangular fuzzy numbers. To introduce the FDA is essential to establish the arithmetic operations and a ranking method for triangular fuzzy numbers. In section 4.4.1, we will briefly explain the ranking method that we applied, and in section 4.4.2, we will describe the Fuzzy Dijkstra algorithm for a type V fuzzy graph.

4.4.1 Method for the arithmetic operations of normalized triangular fuzzy numbers

To compare the fuzzy costs in the FDA, we use the ranking criterion proposed by [Yu and l. Q. Dat 2014]. This method compares the Total Integral of fuzzy numbers depending on a parameter α, called the *index of optimism*. This index represents the degree of optimism of the decision-maker. For α taking values greater than 0.5, the comparison between the fuzzy numbers gives priority to numbers higher than the central value. The opposite occurs when α is lower than 0.5.

Let $\tilde{C}_{(u,v)} = (a, b, d)$ be a triangular fuzzy number that represents the fuzzy cost in a link (u, v). Then, for a fixed α, the *Total Integral of* $\tilde{C}_{(u,v)}$ is defined in equation 4.2:

$$S_T^{\alpha}(\tilde{C}_{(u,v)}) = \frac{1}{2}\left[(1-\alpha)a + b + \alpha d\right] - X_{\min} \tag{4.2}$$

where $X_{\min} \leq a$. For simplicity, since every cost in $\tilde{G}$ is assumed to be a positive fuzzy number in our study, we set $X_{\min} = 0$ without changing the analysis.

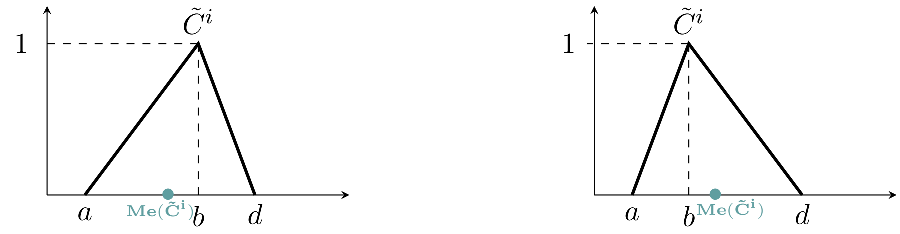
Notice that $S_T^{\alpha}(\tilde{C}_{(u,v)})$ is a crisp value. Thus, we can define an *order relation* of fuzzy numbers from the standard order of real numbers. Let $\tilde{C}^1$ and $\tilde{C}^2$ be two fuzzy costs (defined either in a link or a path), for $\alpha \in [0, 1]$, each $\tilde{C}^i$, $i = 1, 2$ has total integral $S_T^{\alpha}(\tilde{C}^i)$, then,

- If $S_T^{\alpha}(\tilde{C}^1) > (<)\, S_T^{\alpha}(\tilde{C}^2)$ then $\tilde{C}^1$ is greater (smaller) than $\tilde{C}^2$, denoted as $\tilde{C}^1 \succ (\prec)\, \tilde{C}^2$
- If $S_T^{\alpha}(\tilde{C}^1) = S_T^{\alpha}(\tilde{C}^2)$ and $\mathrm{Me}(\tilde{C}^1) > (<)\, \mathrm{Me}(\tilde{C}^2)$ then $\tilde{C}^1 \succ (\prec)\, \tilde{C}^2$
- If $S_T^{\alpha}(\tilde{C}^1) = S_T^{\alpha}(\tilde{C}^2)$ and $\mathrm{Me}(\tilde{C}^1) = \mathrm{Me}(\tilde{C}^2)$ then $\tilde{C}^1 = \tilde{C}^2$

where $\mathrm{Me}(\tilde{C}^i)$ $(i = 1, 2)$ denotes the median of the fuzzy number $\tilde{C}^i$. To compute the median of the fuzzy cost $\tilde{C}^i = (a, b, d)$ it is necessary first to identify if the median is smaller or greater than the value b.

- If $b - a \geq \frac{d-a}{2}$ then $a \leq \mathrm{Me}(\tilde{C}^i) \leq b$ and $\mathrm{Me}(\tilde{C}^i) = a + \sqrt{\frac{(b-a)(d-a)}{2}}$, figure 4-1(a).
- If $d - b \geq \frac{d-a}{2}$ then $b \leq \mathrm{Me}(\tilde{C}^i) \leq d$ and $\mathrm{Me}(\tilde{C}^i) = d - \sqrt{\frac{(d-b)(d-a)}{2}}$, figure 4-1(b).

This relation is an order relation because it meets the antisymmetry, reflexivity, and transitivity properties. On the other hand, since the Total Integral satisfies the linearity property, we can apply it to compare the cost of two links and the cost of two paths using the comparison between their Total Integrals.



(a) $a \leq \text{Me}(\tilde{A}) \leq b$ (b) $b \leq \text{Me}(\tilde{C}^i) \leq d$

Figure 4-1: *Median of a triangular fuzzy number* $\tilde{C}^i = (a, b, d)$.

4.4.2 Fuzzy Dijkstra Algorithm

Let $\Psi(r, v)$ denote the set of all paths between r and v ($v, r \in V$ with $v \neq r$), and $P^* \in \Psi(r, v)$ be the shortest path between r and v. The cost of P^* is defined in equation 4.3,

$$\tilde{\delta}_{(r,v)} = \tilde{C}_{P^*(r,v)} = \begin{cases} \min\limits_{\forall P \in \Psi(r,v)} \{\tilde{C}_{(P(r,v))}\} & \text{if } \Psi(r,v) \neq \emptyset \\ \infty & \text{if } \Psi(r,v) = \emptyset \end{cases} \tag{4.3}$$

Then, for each vertex $v \in V$, the FDA defines an attribute $\tilde{d}(v)$, which is an upper bound on the cost of P^*,

$$\tilde{d}(v) \geq \tilde{\delta}_{(r,v)}$$

Given a vertex $v \in V$, we will denote by $\Gamma(v)$ the set of neighbors of v, that is, adjacent vertices to v. Moreover, we will denote by vertex $w(v)$ the *predecessor of* v in P^*. The algorithm considers $w(v)$ to be either a vertex of $\Gamma(v)$ or NIL. $w(v)$ is the predecessor of v in the shortest path to v known so far during the algorithm's execution. Only when the algorithm has ended can we say that $w(v)$ is the predecessor of v by the shortest path from r to v.

Every vertex has got assigned a *label*, which adapts throughout the algorithm execution. At each stage, the label of vertex $v \in V$ contains its predecessor in the (known so far) shortest path from r to v and the corresponding cost of this path. We describe the label of a vertex v in equation 4.4,

$$\text{Label}(v) = \left[\tilde{d}(v), w(v)\right] \quad \text{with} \quad \begin{aligned} &\tilde{d}(v) := \tilde{d}(w(v)) \oplus \tilde{C}_{(w(v),v)} \\ &w(v): \text{ predecessor vertex of } v \text{ in the} \\ &\qquad\quad \text{provisional path } P^*(r,v) \end{aligned} \tag{4.4}$$

where $\tilde{C}_{(w(v),v)}$ is the cost of link $(w(v), v)$.

At the end of the algorithm, for each vertex v, $\tilde{d}(v)$ will coincide with $\tilde{\delta}_{(r,v)}$, and $w(v)$ will be its predecessor in $P^*(r, v)$.

The FDA contains two main sub-algorithms: Initialization and Relaxation.

The Initialization (algorithm 2) assigns the label Label(v) to every vertex in $\tilde{G}$. The cost of the provisional shortest path from r to any other vertex, $\tilde{d}(v)\ \forall v \in V$, is initialized as ∞, except for r, which is set to be equal to 0.

Algorithm 2 Initialization

function INITIALIZE-SINGLE-SOURCE($\tilde{G}, r, \mathfrak{C}$)
  $\tilde{d}(r) \leftarrow (0, 0, 0)$
  $w(r) \leftarrow$ NIL
  for each vertex $v \in V - \{r\}$ **do**
    $\tilde{d}(v) \leftarrow \infty$
  end for
end function

The FDA proceeds by choosing and extracting vertices from a set, denoted as H, initially defined as V. At each iteration, the algorithm selects the vertex u with minimum cost in H according to the ranking method described in section 4.4.1. Then, it analyzes the cost $\tilde{d}$ of the neighbors of vertex u in the Relaxation (algorithm 3). Thus, the algorithm updates the label of each neighbor $v \in \Gamma(u)$ if, at this point, the path from r to v with $u = w(v)$ has the minimum cost.

Algorithm 3 Relaxation of v

function RELAXATION($u, v, \tilde{C}_{(u,v)}, \alpha$)
  $\tilde{d}^{\text{new}}(v) := \tilde{d}(u) \oplus \tilde{C}_{(u,v)}$
  if $S_T^{\alpha}(\tilde{d}(v)) > S_T^{\alpha}(\tilde{d}^{\text{new}}(v))$ **or** $\left[S_T^{\alpha}(\tilde{d}(v)) = S_T^{\alpha}(\tilde{d}^{\text{new}}(v))\right.$ **and** $\text{Me}(\tilde{d}(v)) > \left.\text{Me}(\tilde{d}^{\text{new}}(v))\right]$ **then**
    $\tilde{d}(v) \leftarrow \tilde{d}^{\text{new}}(v)$
    $w(v) \leftarrow u$
  end if
end function

The algorithm ends when set H is empty. At this point, the label of each vertex v contains the cost of $P^*(r, v)$ $(\tilde{\delta}_{(r,v)})$ with its predecessor $w(v)$ on this path. The shortest path between r and each vertex is found following a *backward procedure*. Algorithm 4 shows the pseudocode of the FDA.

Algorithm 4 Fuzzy-Dijkstra($\tilde{G}, r, t, \alpha$)

Initialize-single-source($\tilde{G}, r, \mathfrak{C}$) ▷ Initialization
$H \leftarrow V$
while $H \neq \emptyset$ **do**
  $u \leftarrow t \mid \tilde{d}(t) = \min_{\forall x \in H} \left\{ \tilde{d}(x) \right\}$
  Update $H \leftarrow H - \{u\}$
  if $\Gamma(u) \cap H \neq \emptyset$ **then**
    for each vertex $v \in \Gamma(u) \cap H$ **do**
      Relaxation($u, v, \tilde{C}_{(u,v)}, \alpha$) ▷ Relaxation of v
    end for
  end if
end while

4.5 Application to a communications network. Fuzzy Functions and Strategies

4.5.1 Description of the communications network

We use a 56-nodes network based on the topology of the backbone network of the Nippon Telegraph and Telephone (NTT). For the particular characteristics of the network, we refer the reader to the examples of the use of the OMNeT ++ simulator [Varga 2001] (figure 4-2). In our application, links are all assumed to have the same capacity of 1 Gbit/s. Furthermore, the connections use Multi-Protocol Label Switching (MPLS) [Rosen et al. 2001] so that once the path between two nodes is selected, it remains unchanged during the connection time. This practice is widely used in Networks of Quality of Service (QoS) to avoid unstable networks when applying dynamic routing. On the other hand, the connections are made without resource reservation, meaning that the system never rejects a connection. Still, there can be a one-off loss of information in the links when their capacity is exceeded. This way of proceeding is usual in IP networks (since the management necessary for the resource reservation would have a very high computational cost), resulting in a loss of performance. The model and source code required to simulate this network is found in [Ariza 2015].

In the simulated network, we identify each node by an index i and the link between nodes determined by the source-destination index pair ij. Also, the capacity (or bandwidth) of link (i, j), (C_{ij}) and its *Used Bandwidth* at time t, $([BW]_{ij}[t])$ will be the primary magnitudes.

4.5.2 Description of cost functions and strategies

We describe in this section the different cost functions used in our experiments. These functions are defined in their *classic form* (as crisp numbers) and their

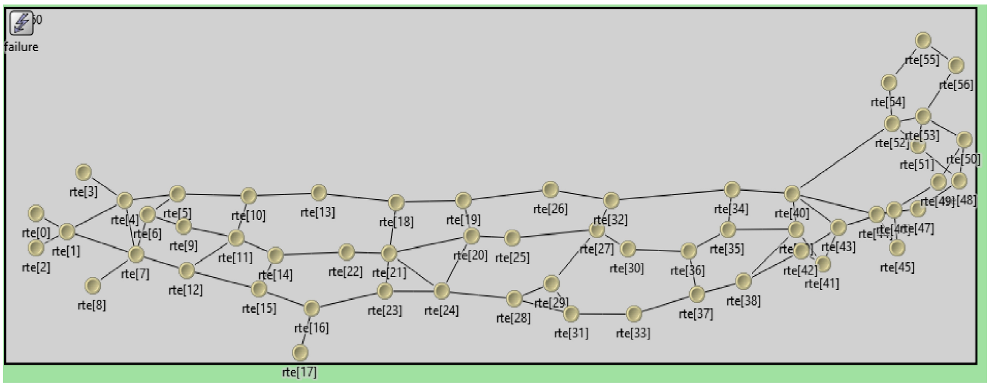


Figure 4-2: *Topology used in our experiments, based on the backbone network of the NTT.*

fuzzy version as triangular fuzzy numbers, allowing us to compare both methodologies.

Additionally, as one of the main contributions of this chapter, we propose a new fuzzy cost function that has no crisp equivalent since it is an empirical adaptation of the generalized fuzzy number model described above. This new function will be explained in detail when presenting strategy 8.

In short, the simulated classic cost functions simulated in a link (i, j) are the following:

- *Normalized Instantaneous Used Bandwidth*, [Ch.Xin and He 2005], [Ariza 2001]: It is the value of the bandwidth occupied in a link at the instant t, weighted according to equation 4.5,

$$bw_{ij}(t) = \frac{BW_{ij}(t)}{C_{ij}} \frac{C_{\max}}{C_{ij}} \tag{4.5}$$

  where,

  $bw_{ij}(t)$ is the Normalized Instantaneous Used Bandwidth in link (i, j) at the instant of adaptation t.

  $BW_{ij}(t)$ is the Instantaneous Used Bandwidth in link (i, j) at the instant of adaptation t.

  C_{ij} is the capacity of link (i, j).

  $C_{\max}$ is the capacity of the link with the highest bandwidth in the network.

  This value will be updated in the routing tables with a predefined periodicity. Therefore, it will be considered "fixed" or "constant" during the update period.

- *Normalized Mean Used Bandwidth*, [Ariza 2001]:
  It is the mean value of the Used Bandwidth in a particular link over a

considered time interval. This value is normalized to the link's capacity and is weighted according to the ratio between the network's larger capacity and the current link capacity. Equation 4.6 shows the definition of the Normalized Mean Used Bandwidth,

$$\overline{[bw]}_{ij} = \frac{\overline{[BW]}_{ij}}{C_{ij}} \frac{C_{\max}}{C_{ij}} \tag{4.6}$$

where,

$\overline{[bw]}_{ij}$ is the Normalized Mean Used Bandwidth in link (i, j) at a measurement time interval.

$\overline{[BW]}_{ij}$ is the Mean Used Bandwidth in link (i, j) at the measured period of time.

C_{ij} and $C_{\max}$ as defined above.

- *Mean Residual Bandwidth associated with a link (or a path)*:
  In a link (i, j), the Mean Residual Bandwidth $(\overline{r}_{ij})$ is defined by the difference between the link capacity and $\overline{[BW]}_{ij}$ at the considered time interval, as shown in equation 4.7,

$$\overline{r}_{ij} = C_{ij} - \overline{[BW]}_{ij} \tag{4.7}$$

  The Mean Residual Bandwidth of a path $(\overline{r}_P)$ is given by the minimum among the Mean Residual Bandwidths of the links that are part of the path, that is, the Residual Bandwidth of the "worst" link.

Equations 4.5, 4.6, and 4.7 will be transformed into triangular fuzzy numbers. In particular, in our experimental study, for each link (i, j) at the n-th time interval, we will use triangular fuzzy numbers (a, b, c), where a and c are computed from the minimum and maximum values of the variable measured at the $(n - 1)$-th time interval, and b is computed from the mean value of the variable over the $(n - 1)$-th time interval.

Once the cost functions previously defined are simulated for each link, we describe the different strategies of cost assignment to search the shortest path using the Dijkstra algorithm for crisp and fuzzy versions.

4.5.3 Strategies for the search of the path with minimum cost

As was summarized in the Introduction, our first goal is to show the competitiveness of the methodology based on the fuzzy model of the network versus that based on the traditional model using crisp magnitudes when searching the best communication path by the application of the Dijkstra algorithm. Therefore, we compare both methodologies for different well-known optimization strategies, regardless of performance on the absolute terms, considering only the relative ones.

Strategy 1: Application of Dijkstra algorithm using the Normalized Instantaneous Used Bandwidth as the cost function (equation 4.5). This strategy aims to find the path, P^*, with the minimum sum of the Normalized Instantaneous Used Bandwidths of their links, as shown in equation 4.8,

$$P^* : \quad C_{P^*} = \min_{\forall P \in \Psi(r,v)} \{C_P\} = \min_{\forall P \in \Psi(r,v)} \left\{ \sum_{(i,j)\in P} bw_{ij}(t) \right\} \tag{4.8}$$

The instants at which we measure this magnitude will be given by $t = t_0 + nT$, where t_0 is the instant of the first measurement, and the measured value will be considered constant during T (time interval between measurements). The total cost of a path is given by the algebraic sum of their links costs; thus, this is a strategy of additive costs.

Strategy 2: Application of Dijkstra algorithm using the Normalized Mean Used Bandwidth as the cost function (equation 4.6). With this strategy, we intend to find the path that minimizes the sum of the Normalized Mean Used Bandwidths of their links (equation 4.9),

$$P^* : \quad C_{P^*} = \min_{\forall P \in \Psi(r,v)} \{C_P\} = \min_{\forall P \in \Psi(r,v)} \left\{ \sum_{(i,j)\in P} \overline{[bw]}_{ij} \right\} \tag{4.9}$$

In strategies 1 and 2, we consider the magnitudes as crisp numbers.

Strategy 3: Application of the FDA using the *Fuzzy Normalized Used Bandwidth*. The cost function for link i, j at the n-th time interval is defined in equation 4.10,

$$\widetilde{[bw]}_{ij} = \left([bw]_{ij}^{\min}, \overline{[bw]}_{ij}, [bw]_{ij}^{\max} \right) \tag{4.10}$$

where $[bw]_{ij}^{\min}$ and $[bw]_{ij}^{\max}$ are the minimum and maximum values of the Normalized Instantaneous Used Bandwidths measured in the $(n-1)$-th time interval, respectively, and $\overline{[bw]}_{ij}$ is the Normalized Mean Used Bandwidth in the $(n-1)$-th time interval. In this instance, we apply the FDA considering additive fuzzy costs. This strategy is directly comparable to strategies 1 and 2.

Strategy 4: (Shortest-Widest, SW) [Wang and Crowcroft 1996]. This strategy search for the path with the maximum Mean Residual Bandwidth. If several paths have the same maximum value, the strategy chooses the path with the minimum number of edges (hops). While bw_{ij} and $\overline{[bw]}_{ij}$ are additive magnitudes (the cost of the path is the sum of the costs of their links), the Residual Bandwidth is concave (the cost of the path is the cost of the link with minimum cost). Note that this metric is not properly a cost, but quite the opposite; it is a metric that measures the goodness of a link. Therefore, the goodness of a path is associated with the goodness of its worst link (link with minimal residual bandwidth).

Figure 4-3 helps understand how this strategy works. On each link appears its Residual Bandwidth. According to the SW, paths P_A and P_C are first chosen (paths with the maximum Residual Bandwidth in their "worst" link), and between those two, the strategy chooses the path P_C that is the one with the minimum number of hops.

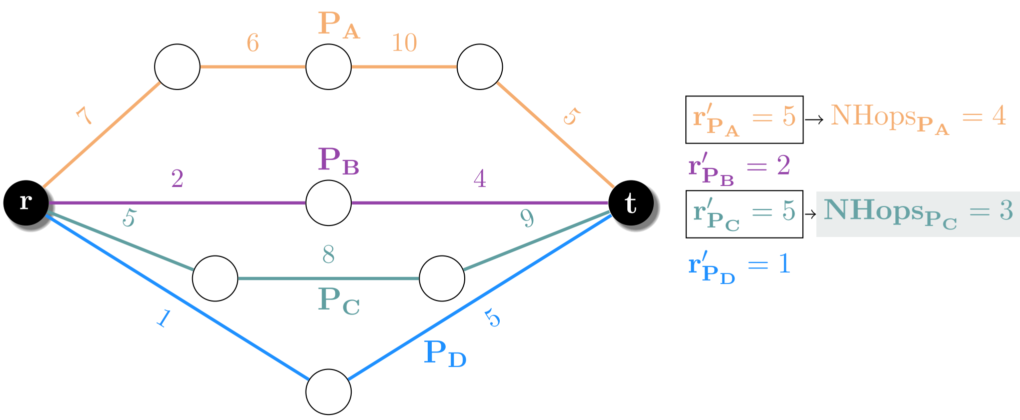


Figure 4-3: *SW applied to a 9-nodes network with explicit Mean Residual Bandwidth in every link*

Strategy 5: (Fuzzy Shortest-Widest, FSW). It is the same as strategy 4, but defining the Fuzzy Mean Residual Bandwidth measurement of each link (i, j) as a triangular fuzzy number defined in expression 4.11,

$$\tilde{\bar{r}}_{ij} = (r_{ij}^{\min}, \bar{r}_{ij}, r_{ij}^{\max}) \quad \text{where} \quad \begin{array}{l} r_{ij}^{\min} = C_{ij} - [BW]_{ij}^{\max} \\ \bar{r}_{ij} = C_{ij} - \overline{[BW]}_{ij} \\ r_{ij}^{\max} = C_{ij} - [BW]_{ij}^{\min} \end{array} \tag{4.11}$$

We apply the FDA that finds the shortest path using $\tilde{\bar{r}}_{ij}$ as the cost function on each link (i, j). This strategy is directly comparable with strategy 4.

Strategy 6: (Widest-Shortest, WS) [Guerin et al. 1997]. WS consists of the inverse process of the SW. First, WS searches for the path with the minimum number of hops. In the case of several paths with the same number of hops, the strategy searches for the one with the maximum Residual Bandwidth. Figure 4-4 shows the application of WS to the same network as figure 4-3. The paths with the minimum number of hops are P_B and P_D. Among them, the strategy WS chooses the path P_B because it has the maximum Residual Bandwidth used bandwidth in its worst link.

Strategy 7: (Fuzzy Widest-Shortest, FWS) Using the same definition of the fuzzy cost as strategy 5, we follow the same selection pattern as strategy 6. This strategy competes with strategy 6.

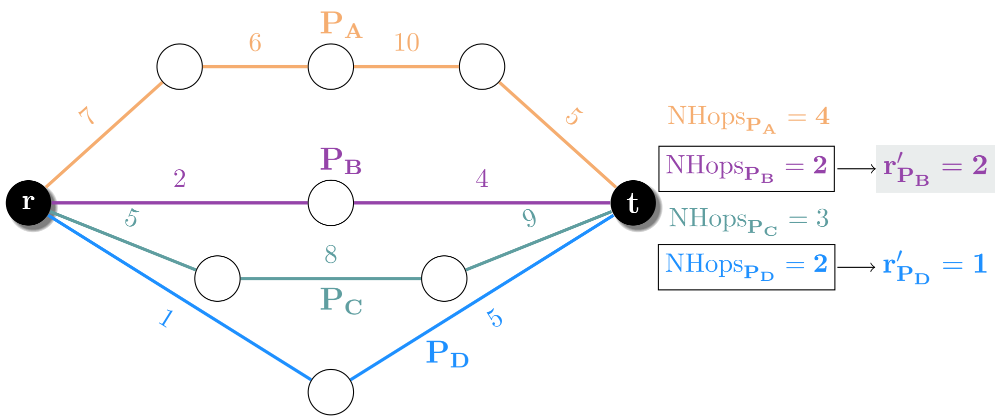


Figure 4-4: *WS applied to a 9-nodes network with explicit Mean Residual Bandwidth in every link*

Finally, we propose a new strategy (strategy 8), which uses a new fuzzy cost function with no crisp equivalent since it is an empirical adaptation of the generalized fuzzy number model described above. In the following, we explain this new strategy and its cost function.

Strategy 8: It can be verified that the smaller the used bandwidth at the $(n-1)$-th time interval, the higher the degree of uncertainty in the values considered for the used bandwidth of a link at the n-th time interval. We can explain the above as the following:

When a link has a very low used bandwidth in the $(n-1)$-th time interval, the dynamics of the network tend to increase the information transmitted by that link. That is, it considerably increases the used bandwidth of the link in the n-th time interval (an analogous situation occurs with the traffic system in a city). In this case, the new connection will cause a high relative variation and, therefore, a higher uncertainty in the value of the used bandwidth in the n-th time interval. On the contrary, when the used bandwidth is high in the $n-1$-th time interval, the new connection will vary little in the n-th time interval. Thus the uncertainty in the used bandwidth will be lower in the n-th time interval.

These reasons lead us to redefine the fuzzy number that we describe in strategy 3 (equation 4.10), which is now multiplied by a greater coefficient, the smaller the used bandwidth is. In this way, we achieve a double objective:

a) we widen the extreme values of the fuzzy number representing the used bandwidth when its measured value is small.

b) we shift the central value to the right, thus correcting the error in the measurement of the used bandwidth.

Thus, the *Modified Fuzzy Normalized Used Bandwidth* in link (i, j) is defined in equation 4.12.

$$\widetilde{\text{Cost}}_{\text{St8}} = A * B^{-\overline{[bw]}_{ij}} \widetilde{[bw]}_{ij} \tag{4.12}$$

In practice, this operation produces a displacement to the right and a widening of $\widetilde{[bw]}_{ij}$. In our experiments, we use the values $A = 10$ and $B = 20$.

4.6 Experimentation and results

4.6.1 Description of the experiment

We use the simulator OMNET ++ [Varga 2001] for the experimentation. The code used for the experiment and the configuration files can be downloaded from [Ariza 2015]. In addition, we perform a flow-oriented simulation to reduce the execution time of each simulation (the alternative packet-oriented simulation would require an execution time in the order of a thousand times greater).

Our data loss model simulates an Optical Burst Switching network (OBS) without storage. If a burst does not have enough bandwidth on a link, we discard the information (that is, information is lost) until there are enough resources. Also, in the case that during the burst lifetime, there are not enough resources at any time, we discard the complete burst.

To facilitate the reproduction of the experiments, we have carried out ten simulations with different seeds for each of the different strategies, where every node generates traffic with the same probability.

Table 4.1 indicates the traffic characteristics. [1] [2] [3]
In the following, we describe each parameter listed in table 4.1,

Call rate: Mean number of connections requests.

Call duration: Time interval at which the connection is active once it is established.

Type of traffic: Data is transmitted for a specific time (burst with constant bandwidth) and not transmitted for another time interval.

Used bandwidth over an ON period: Resources required for a connection over an ON period. Since we are considering an ON period of 5s and an OFF period of 1s, the Mean Used Bandwidth for a connection is 25 Mb/s (the Used Bandwidth of an Advanced Video Coding High Definition (AVCHD) connection, approximately). We use this value intending to work in conditions of saturation.

[1] The simulator can be downloaded at https://omnetpp.org/download/
[2] The source code of the model used can be downloaded in https://github.com/aarizaq/flowsimulator
[3] The configuration for the running of the experiments can be downloaded in https://github.com/aarizaq/configurationFuzzy/blob/master/omnetpp.ini

Description of the parameter	Value
Call rate	Poisson Distribution with mean value 0.5 s
Call duration	Exponential Distribution with mean value $\mu = 120$ s
Type of traffic	ON/OFF
Used Bandwidth over an ON period	30 Mb/s
ON period	Exponential Distribution with mean value $\mu = 5$ s
OFF period	Exponential Distribution with mean value $\mu = 1$ s
Type of connection	Symmetrical and bidirectional
Simulation time	10000 s
Table updating time interval	300 s
Number of replications with different seeds	10

Table 4.1: *Characteristics of the traffic used in the experiments.*

ON period: Transmission period. It is modeled by random distribution. In our case, we use an exponential distribution with a mean of 5 s.

OFF period: Period without transmission, that is, without consumption of resources. We use the same exponential function with a mean of 1 s. We force an off period to guarantee the competition of the burst.

Connection type: Given a connection, both nodes can be source and destination simultaneously with the same consumption of resources. Thus, the connection is symmetrical and bidirectional.

Simulation time: It is the simulated time assigned to each experiment (do not confuse with the duration of the simulation)

Table updating time interval: Time interval used to measure the average occupation of links and update the routing tables. Higher values in this parameter imply higher uncertainty in the measurements.

In our simulation scenario, source and destination nodes are randomly selected (with equal probability) among all the nodes in the network. Thus, there are multiple active source-destination pairs simultaneously.

In the network, our magnitudes of interest are *the total number of sent and received bytes by each node at each simulation interval.*

From the previous values, we calculate the following variables:

Mean Delivery Rate (MDR): Total number of received bits divided by the total number of sent bits throughout an experiment. This value indicates the probability that sent data is finally received.

Global Mean Delivery Rate, (GMDR): Mean of the MDR in ten experiments.

Confidence interval: (Computed with a probability of 0.95)

4.6.2 Results and discussion

We calculated the GMDR for each strategy with its respective cost function. Table 4.2 shows these values together with the associated confidence intervals. Figure 4-5 shows the same information in a bar diagram.

Strategies	GMDR	Error	Conf. Interval $[\mathbf{CI}_{\min}, \mathbf{CI}_{\max}]$	
			$\mathrm{CI}_{\min}$	$\mathrm{CI}_{\max}$
Strategy 1	0.860	0.004	0,856	0,864
Strategy 2	0.861	0.0064	0,856	0,866
Strategy 3	0.870	0.004	0,866	0,874
Strategy 4	0.8500	0.015	0,835	0,865
Strategy 5	0.857	0.013	0,843	0,870
Strategy 6	0.966	0.0025	0,963	0,968
Strategy 7	0.966	0.0026	0,9632	0,9684
Strategy 8	0.9982	0.0004	0,9978	0,9986

Table 4.2: *GMDR and Confidence Interval associated with each Strategy.*

When applying Strategy 3, we observe that the value of the GMDR surpasses its crisp equivalents (Strategies 1 and 2), although in a small percentage. In addition, we find that confidence intervals are not overlapping. From the statistical point of view, the results are distinguishable so that the resulting comparison has statistical validation.

Similarly, with Strategy 5, the value of the GMDR surpasses its crisp equivalent (Strategy 4), but their confidence intervals are overlapped. Thus, we must consider both results as equal from a statistical point of view.

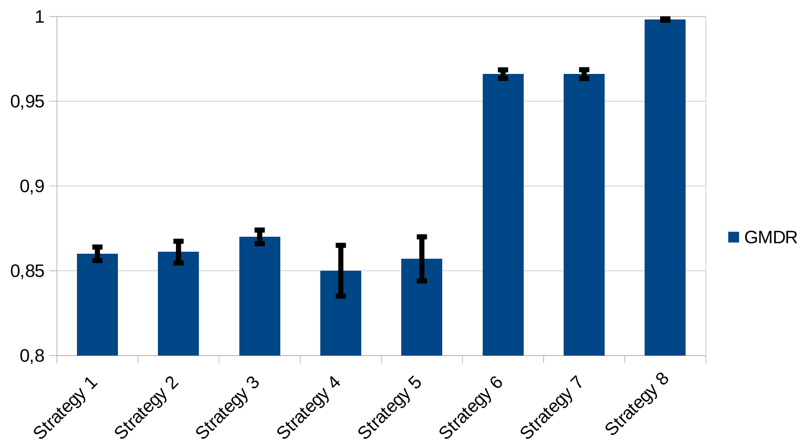


Figure 4-5: *Bar diagram of the GMDR for each Strategy*

Strategies 6 and 7 have the same GMDR values with practically equal confidence intervals. We also observe that Strategies 6 and 7 are superior to the previous ones (in the order of 10%). Our hypothesis in this regard is the advantage of these strategies working in a network close to saturation. Because these strategies search for paths with the minimum number of hops, they contribute a lesser flow of information through the network. Therefore, Strategies 6 and 7 benefit from the low consumption of resources compared to Strategies 1 to 5.

Finally, as the main result of our study, we verify that Strategy 8 (proposed for the first time in this work) achieves widely superior performance to all the previous ones, with a value of GMDR close to 1.

Therefore, we summarize the obtained results with the following two statements:

1. Fuzzy versions of the classical strategies are more efficient (or practically equal) than their corresponding crisp versions, although in a very narrowly. Thus, we can say that our fuzzy modeling of a communications network correctly incorporates the uncertainty in the measurement variables, being therefore competitive.

2. Strategy 8, proposed by us, is the most efficient with a clear advantage over the other strategies, having a GMDR close to 1. This advantage comes from the definition of a new fuzzy cost function incorporating our empirical knowledge about the effects of measurement uncertainty.

4.7 Conclusions

In this chapter, we addressed the problem of searching the shortest path (or more efficient, in a more general sense) between two nodes in a communications network, considering uncertainty in cost function measures of its connections.

The above-referred uncertainty is inevitably caused by the network dynamics and the impossibility of making decisions with the instantaneous measurement of the state variables at each moment. Thus, in general, over a given time interval, decisions on the routing of information are made using measurements obtained in the previous time intervals, which, naturally, can differ from the current ones.

We associate the network to a Type V fuzzy graph to incorporate our fuzzy cost functions and strategies into a communication network management system. In this graph, nodes and links are located (without uncertainty), but a triangular fuzzy number defines the cost for each connection. Moreover, a fuzzy version of Dijkstra's routing algorithm was developed and exhaustively described, called Fuzzy Dijkstra Algorithm (FDA). Simulated experiments have shown its correct operation and competitiveness.

It is essential to distinguish this approximation, where the uncertainty lies in the value of the variables, from other approximations based on Fuzzy Inference, where these values are considered crisp (precise data) and the uncertainty lies in the decision rules used.

For our experimental study, we implemented the most commonly used cost functions and strategies in the management of real networks based on crisp values (e.g., these using the Instantaneous or Mean Used Bandwidth or Residual Bandwidth as cost function, Shortest-Widest (SW), Widest-Shortest (WS)). We confronted the strategies with similar ones based on our definition of fuzzy costs.

As an especially interesting contribution, we proposed a new fuzzy strategy (strategy 8) which has no correspondence with a classic one. Strategy 8 adapts the definition of the fuzzy cost to the fact that the smaller the Used Bandwidth at the $(n-1)$-th time interval (where the measures have been obtained), the more significant uncertainty in the Used Bandwidth value considered in the n-th time interval.

We did the experiments on a 56-nodes network based on the topology of the NTT backbone network, whose particular characteristics can be found in the examples of the use of OMNET ++ simulator.

Fuzzy strategy 3 surpasses their analogous crisp ones (strategies 1,2) slightly but in a statistically significant way. In the fuzzy strategies 5 and 7, we observed that they do not present statistically significant differences with their crisp analogous strategies 4 and 6, respectively. However, our new fuzzy strategy (strategy 8), with an effectiveness ratio very close to unity, clearly surpasses the rest of the analyzed strategies.

We conclude that our methodology of introducing the inherent uncertainty to the dynamic nature of the network in routing algorithms has been successful. On the other hand, we must remember that our models, based on fuzzy numbers, model the real uncertainty generated in a network management system. This uncertainty is due to the imponderable fact that we use values of the system variables that do not correspond to the real ones at each moment. As seen in our experimental study, our models, especially strategy 8, produce better results than those methods that do not consider uncertainty. However, it is clear that this uncertainty will always be present. Therefore, we cannot be sure that our solution is optimal in each case, especially since we empirically find many of the parameters used to define that uncertainty.

CHAPTER 5

SHORTEST PAIR OF EDGE-DISJOINT PATHS IN A COMMUNICATION NETWORK. A FUZZY APPROACH

- L. Valdés, A. Ariza, S.M. Allende, G. Joya; "Searching the Shortest Pair of Edge-Disjoint Paths in a Communication Network. A Fuzzy Approach."; In: *Advances in Computational Intelligence. IWANN. Lecture Notes in Computer Science, Springer, Cham.*; Vol. 11507; pp. 640-652; (2019); `DOI: 10.1007/978-3-030-20518-8_53`.

Communication networks' survivability is extremely important due to the different services that the networks provide for society and the economy. Survivability can be defined as the ability of the network to support the committed QoS continuously in the presence of various failure scenarios. Related to survivability is the concept of Self-Healing, wherein in a saturation situation, the traffic between two nodes can be organized by dividing it between two alternative paths. In this sense, we address the problem of finding the shortest pair of edge-disjoint paths with fuzzy costs between two nodes in a communication network. We use a new cost function named Modified Fuzzy Normalized Used-bandwidth, described as a triangular fuzzy number, thus incorporating the uncertainty generated in calculating this magnitude in a real network. The proposed algorithm uses as a sub-algorithm an adaptation of a Modified Fuzzy Dijkstra algorithm applied in a type V mixed fuzzy graph with arcs whose costs are negative triangular fuzzy numbers. We prove its effectiveness by simulating traffic close to overload with two types of communication sources: regular and priority sending of information. The addressed problem presents a considerable interest in financial entities or government services, where privacy and security against external attacks must be considered.

5.1 Introduction

Survivability can be defined as the network's ability to support the committed QoS continuously in the presence of various failure scenarios. *Communication* *networks' survivability* is extremely important due to the different services that networks provide for society and the economy. Communication networks are used by banking and finance entities and for government services. Thus, the survival of these networks is essential in critical applications where, by contract or prior requirement, they can not stop working. That is, there can be no loss of data at any time. On the other hand, the survivability of the network becomes even more crucial to face the unstoppable increasing impact and rate of failures due to, among other causes, the intricate design that modern systems have. In particular, the problem of the increased bandwidth of links. This issue is essential in fiber optic networks since its sending speed is very high, and the loss of data originated, while an alternative to the outage is found or repaired, is very high.

The network must remain operational regardless of whether a failure occurs (in a node or a link). The network's survival consists of two components: the analysis, which understands failures and system functionality after failures, and the design, which adopts network procedures and architecture to prevent and minimize the impact of failures/attacks on network services.

Another concept related to the survivability of the network is the capacity of *Self-healing* that can be associated with the redistribution of traffic in the system by using alternative paths between two nodes when is appropriate. The traffic between two nodes can be organized by dividing it between two alternative paths in a saturation situation. That would lower the saturation conditions and improve the bandwidth of both paths. When each path is close to saturation, possibly a packet with a certain length cannot completely arrive if a single path sends it. However, this packet could arrive if we divide it into two smaller packets sent through two paths with no common links under the same bandwidth conditions. That is, the bandwidth of each path can be sufficient for a segment of the packet to be sent but not enough to send the entire packet.

In this chapter, to address the above problems, we propose to analyze the problem of finding the shortest pair of paths, disjoint in links, between two origin and destination nodes in a communication network. The pair of paths can have different purposes:

(i) Deviate the information by one of the paths (replacement) when the other suffers a failure. Thus, the network avoids losing information while repairing the issue or looking for an alternative path.

(ii) Distribute the data sent between the two paths simultaneously when the network is near saturation. In this case, possibly a packet of a specific length cannot completely arrive if a single path sends it. However, this packet could reach its destination if we divide it into two smaller packets transmitted through two paths with no mutual links under the same bandwidth conditions.

(iii) Distribute the information in a not predictable way between both paths, thus making it impossible to capture a complete message by an intruder.

These tasks have a considerable interest in contexts such as financial entities or government services, where privacy and security against external attacks have to be considered, and in networks with a continuous increase in their transmission speed, such as the optical ones. Thus, different works have addressed one or several of them in the last two decades: [Gottschau et al. 2018], [He and Rexford 2008], [Lou and Fang 2001], [R. Maaloul and Cousin 2018], [Yang and Papavassiliou 2001], [D. Zhu 2002]. In most of these works, a common characteristic is that a real number represents the cost of edges, and, in some cases, this value is invariant in time. The authors do not simulate the real traffic in a credible network and only present a mathematical and/or computational analysis of the proposed methods. Their goal is not to find the shortest pair but only a disjoint pair of paths.

In contrast with these characteristics, we start from the fact that a fuzzy number represents the function that defines the edges' weight (or cost). We justify this hypothesis under the same terms as in chapter 4: the values of the cost functions of edges measured and used over a time interval are calculated from the state of the network in the previous time interval. Therefore, there is a high probability of discrepancy between the value we compute and the real value at each moment. A fuzzy number can model this uncertainty.

The network reliability can be characterized by parameters like the degree of each node, the average distance between every pair of nodes, connectivity, saturation level at each moment, available flow capacity, etc. We represent the network as a set of nodes (switches, routers, satellites, base stations, etc.) and another set of links between them (optical fibers, electrical wires, etc.) represented by edges. To measure the network functionality, we will consider as a metric the occupied bandwidth in the links at each moment, which we assume to be a fuzzy value. From this perspective, we analyze the network reliability considering the connectivity parameter by searching a pair of paths, disjoint by edges, between two fixed nodes ([Valdés et al. 2019]).

In our work, we do not consider the possibility of designing the network to guarantee the duplicity of paths, but we start from a network already in use and search for a pair of edge-disjoint paths if possible.

5.2 Problem statement

In the introduction, we refer to the importance of *survival* in a communication network since it deals with its ability to continue operating, even when ruptures and interruptions occur at specific points. A fundamental problem would be when one or several links are damaged due to external or internal factors, disrupting the sending of information between two network nodes. We can interpret this situation as eliminating a path between the source and destination nodes. A solution to this problem would be the immediate replacement of the interrupted path by a new one not containing any damaged link. In this way, the network can maintain its operation uninterruptedly. On the other hand, another interesting challenge occurs when in saturation conditions, the information sent from the source node to the destination node is divided into two packets and transmitted in parallel by two paths that do not have mutual links. In this case, it is not of interest only that both paths are link-disjoint, but that the sum of their costs is minimal. In this way, when both paths are used simultaneously, we would optimize the cost of sending the information and reduce the network's saturation. The first problem is relatively simple to solve since after finding the shortest path between the source and destination nodes, all its edges are eliminated, and a new shortest path between both nodes is searched again. However, the second problem presents a higher complexity in its resolution and is where we make a greater focus.

We model the network by defining a type V fuzzy graph associated with the system. We model the uncertainty generated by operating in the time interval t with the values of the variables obtained in the $t-1$ interval using cost functions defined by triangular fuzzy numbers.

Let $\tilde{G} = (V, E, \mathfrak{C})$ be a non-directed type V fuzzy graph where:

- We fix a pair of vertices as the source and destination vertices, denoted by r and t, respectively.
- Let $\tilde{C}_e \in \mathfrak{C}$, $e \in E$, be the cost of edge e. We define $\tilde{C}_e = (a_e, b_e, c_e)$ as a positive triangular fuzzy number referred to the cost of edge e, which is close to value b_e and is measured with an uncertainty bounded by the values a_e and c_e. In the same way, the cost of a path P, $\tilde{C}_P = (a_P, b_P, c_P)$, is also a positive triangular fuzzy number. We consider the cost of a link an additive metric; therefore, in equation 5.1 we define the cost of the path P,
$$\tilde{C}_P = \sum_{\forall e_i \in P} \tilde{C}_{e_i} \tag{5.1}$$
$\tilde{C}_P$ is referred to as "cost of path P, which is close to value b_P." In general, when we do not use additive metrics, the total cost of a path will be a function of the costs of all its edges.
- Each edge is equivalent to a pair of oppositely directed arcs whose costs correspond to the length of the original edge and its complementary.
- The graph is connected.
- More than one path may exist between vertices r and t.

For short, we refer to "fuzzy cost" the cost of an edge or path when this is defined as a triangular fuzzy number.

Our goal is to find the shortest pair of edge-disjoint paths (paths with no common edges) between the source and destination vertices r and t (pair of paths with no common edges where the sum of the costs of both paths is minimum)

[Bhandari 1999] already proposes a solution to search for the shortest pair of edge-disjoint paths in a graph with crisp and positive costs. In essence, this proposal consists of the following steps:

1. Finding the shortest path of the graph, S, using the Dijkstra algorithm.

2. Modifying the original graph by replacing the edges of S with arcs oppositely oriented, and whose costs are defined as the opposite (negative) number of the costs of said edges (this new graph is called *modified graph*).

3. Finding the shortest path in the modified graph, P_{aux}, by modifying the Dijkstra algorithm to allow its convergence in a graph with negative costs.

4. Finally, the shortest pair of edge-disjoint paths is either (S, P_{aux}) or another pair of paths resultant from the combination of S and P_{aux}.

Our algorithmic proposal, called *Algorithm to find the Fuzzy Shortest Pair of Edge-Disjoint Paths (FSPPA for short)*, constitutes an adaptation of the previous strategy for a type V fuzzy graph. This new approach requires the change of the operations involved in the original procedure, where we base our proposal on adapting the crisp analysis to the fuzzy character of the cost functions of edges. Therefore,

first, we must have a criterion of comparison of fuzzy numbers;

second, we must find a way to express the negative character of a triangular fuzzy number necessary in the construction of the modified graph $\tilde{G}'$;

third, we must create a fuzzy adaptation and modification of the Dijkstra algorithm that can be applied to the graphs $\tilde{G}$ and $\tilde{G}'$ ($\tilde{G}'$ is a type V fuzzy graph containing arcs with costs defined as negative triangular fuzzy numbers).

These new operations are an essential part of our contribution, and we will briefly explain them in detail in section 5.3.

5.3 Remarkable concepts involved in the FSPPA

5.3.1 Necessary conditions under which a path is the shortest in $\tilde{G}$

Given the total cost of a path, $\tilde{C}_P = (a_P, b_P, c_P)$, we intend to find the condition, based on the values a_P, b_P, and c_P, which allows us to compare the cost of two paths. As in chapter 4, we base the comparison of fuzzy costs on the ranking criterium proposed in [Yu and l. Q. Dat 2014] (briefly described in section 4.4.1).

Let $\tilde{C}_{P_1}$ and $\tilde{C}_{P_2}$ be fuzzy costs of paths P_1 and P_2, respectively. Without loss of generality, we assume that path P_1 is shorter in terms of its total cost than path P_2. Therefore, for a fixed $\alpha \in [0,1]$ the inequality 5.2 holds,

$$S_T^{\alpha}(\tilde{C}_{P_1}) < S_T^{\alpha}(\tilde{C}_{P_2}) \tag{5.2}$$

where $S_T^{\alpha}(\tilde{C}_{P_1})$ and $S_T^{\alpha}(\tilde{C}_{P_2})$ are the total integrals of P_1 and P_2, respectively.

We can rewrite the inequality 5.2 based on the definition of total integral,

$$\begin{aligned} &\alpha S_R(\tilde{C}_{P_1}) + (1-\alpha) S_L(\tilde{C}_{P_1}) < \alpha S_R(\tilde{C}_{P_2}) + (1-\alpha) S_L(\tilde{C}_{P_2}) \\ \Rightarrow &\left(S_R(\tilde{C}_{P_1}) - S_L(\tilde{C}_{P_1})\right) - \left(S_R(\tilde{C}_{P_2}) - S_L(\tilde{C}_{P_2})\right) < \frac{1}{\alpha}\left(S_L(\tilde{C}_{P_2}) - S_L(\tilde{C}_{P_1})\right) \end{aligned} \tag{5.3}$$

where $S_L(\tilde{C})$ and $S_R(\tilde{C})$ are the left and right integrals of the fuzzy number $\tilde{C}$, respectively.

Assuming that $\tilde{C}_{P_1}$ and $\tilde{C}_{P_2}$ are triangular fuzzy numbers, the replacement of their left and right integrals by their respective expressions leads us to equation 5.4,

$$\begin{aligned} &\left(\left(\frac{b_{P_1}}{2} + \frac{c_{P_1}}{2} - X_{\min}\right) - \left(\frac{a_{P_1}}{2} + \frac{b_{P_1}}{2} - X_{\min}\right)\right) - \left(\left(\frac{b_{P_2}}{2} + \frac{c_{P_2}}{2} - X_{\min}\right) - \right. \\ &\left. - \left(\frac{a_{P_2}}{2} + \frac{b_{P_2}}{2} - X_{\min}\right)\right) < \frac{1}{\alpha}\left(\frac{a_{P_2}}{2} + \frac{b_{P_2}}{2} - X_{\min} - \frac{a_{P_1}}{2} - \frac{b_{P_1}}{2} + X_{\min}\right) \\ \Rightarrow &\frac{c_{P_1}}{2} - \frac{a_{P_1}}{2} - \frac{c_{P_2}}{2} + \frac{a_{P_2}}{2} < \frac{1}{\alpha}\left(\frac{a_{P_2}}{2} + \frac{b_{P_2}}{2} - \frac{a_{P_1}}{2} - \frac{b_{P_1}}{2}\right) \\ \Rightarrow &(1-\alpha)a_{P_1} + b_{P_1} + \alpha c_{P_1} < (1-\alpha)a_{P_2} + b_{P_2} + \alpha c_{P_2} \end{aligned} \tag{5.4}$$

where $\tilde{C}_{P_1} = (a_{P_1}, b_{P_1}, c_{P_1})$ and $\tilde{C}_{P_2} = (a_{P_2}, b_{P_2}, c_{P_2})$.

Equation 5.4 is an operational condition to compare two triangular fuzzy numbers. We summarize this result in proposition 5.3.1.

Proposition 5.3.1.
Let P_1 and P_2, be two paths from a source vertex r to a destination vertex t on a type V fuzzy graph $\tilde{G} = (V, E, \mathfrak{C})$ whose costs are triangular fuzzy numbers $\tilde{C}_{P_1} = [a_{P_1}, b_{P_1}, c_{P_1}]$ and $\tilde{C}_{P_2} = [a_{P_2}, b_{P_2}, c_{P_2}]$, respectively. The comparison among both paths is performed based on their total integrals, calculated as in [Yu and l. Q. Dat 2014]. For a fixed $\alpha \in [0,1]$, P_1 is shorter (equal) than P_2 if and only if,

$$(1-\alpha)a_{P_1} + b_{P_1} + \alpha c_{P_1} < (=)(1-\alpha)a_{P_2} + b_{P_2} + \alpha c_{P_2} \tag{5.5}$$

Based on proposition 5.3.1, for a fixed $\alpha \in [0,1]$, we can establish the necessary condition for being a path S the shortest path between the vertices r and t in $\tilde{G}$. Let $\tilde{C}_S$ be the cost of the shortest path S where $\tilde{C}_S = [a_S, b_S, d_S]$ is a triangular fuzzy number defined as the sum of the costs of all edges contained in S. $\tilde{C}_S$ is less than the cost of any other path P between r and t. Equation 5.6 expresses the said above and is considered a necessary condition for S to be the shortest path in $\tilde{G}$,

$$(1-\alpha)a_S + b_S + \alpha d_S \leq (1-\alpha)a_P + b_P + \alpha d_P, \quad \forall S, P \in \tilde{G}: \; P \neq S \tag{5.6}$$

where $\tilde{C}_P = [a_P, b_P, d_P]$ is the cost of path P.

5.3.2 Risk of Suboptimality in search of the shortest pair of edge-disjoint paths

A method to solve the search for the shortest pair of edge-disjoint paths in a graph would be to apply twice the FDA (proposed by us in chapter 4) in a type V fuzzy graph. This algorithm finds the shortest path for a fixed pair of source and destination vertices in the graph. Then, the edges constituting the path are eliminated, and in the resultant modified graph, the algorithm finds a new shortest path when it is newly applied. This methodology solves the problem created when some links get damaged, and the path containing them is interrupted. Thus, the damaged path can not be used to search the other edge-disjoint path. However, applying this algorithm could lead us to a suboptimality when the problem is to find, besides the shortest path, an alternative one, so the information is divided into two parts and sent simultaneously. In the following, we show an example where we cannot get the optimal solution for the shortest pair of edge-disjoint paths problem when applying the above described method..

Example 5.3.2.1.
Figure 5-1 shows a type V fuzzy graph $\tilde{G}$, where the costs on edges are triangular fuzzy numbers and r and t are the source and destination vertices, respectively. After the first run of the search algorithm, we base our approach on the algorithm's analysis but adapt this to the fuzzy character of the edges cost functions for the comparison of paths. We find in $\tilde{G}$ the shortest path $P_1 = \langle r, v_2, v_3, v_4, t\rangle$ with cost $\tilde{C}_{P_1} = (1,1,1)+(1,2,3)+(1,2,3)+(1,1,1) = (4,6,8)$, (blue path). Removing the edges of path P_1 on $\tilde{G}$, we obtain a modified graph. In this new graph, after applying once again the search algorithm, we verify that the shortest path is $P_2 = \langle r, v_6, v_2, v_7, t\rangle$ with cost $\tilde{C}_{P_2} = (1,1,1)+(1,1,1)+(1,2,3)+(3,4,6) = (6,8,11)$ (green path). Therefore, we obtain the pair of edge-disjoint paths (P_1, P_2) with total cost $\tilde{C}_{P_1,P_2} = \tilde{C}_{P_1} + \tilde{C}_{P_2} = (4,6,8)+(6,8,11) = (10,14,19)$.

Nevertheless, if we consider the pair (P_3, P_4) of edge-disjoint paths where $P_3 = \langle r, v_6, v_2, v_7, v_4, t\rangle$ with $\tilde{C}_{P_3} = (1,1,1)+(1,1,1)+(1,2,3)+(1,2,4)+(1,1,1) = (5,7,10)$ and $P_4 = \langle r, v_2, v_3, v_5, t\rangle$ with $\tilde{C}_{P_4} = (1,1,1)+(1,1,1)+(1,2,3)+(1,2,4) = (4,6,9)$ (figure 5-2) we can easily verify that the total cost $\tilde{C}_{P_3,P_4} = \tilde{C}_{P_3} + \tilde{C}_{P_4} = (9,13,19)$ is smaller than $\tilde{C}_{P_1,P_2}$.

The total integrals for the total costs of both pairs of paths $\tilde{C}_{P_1,P_2}$ and $\tilde{C}_{P_3,P_4}$, depending on the index α are

$$S_T^{\alpha}(\tilde{C}_{P_1,P_2}) = \frac{9}{2}\alpha + 12 \quad \textit{and} \quad S_T^{\alpha}(\tilde{C}_{P_3,P_4}) = 5\alpha + 11$$

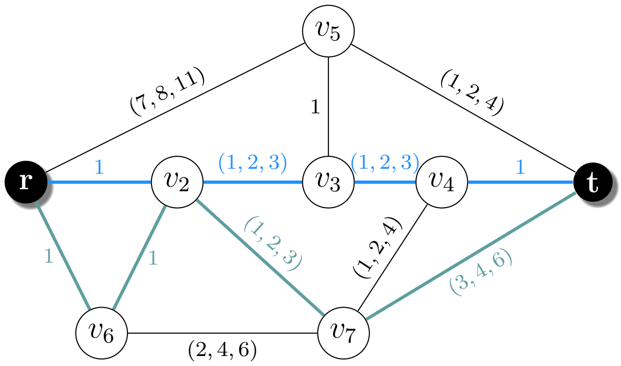


Figure 5-1: *Graph $\tilde{G}$ with the pair of edge-disjoint paths (P_1, P_2).*

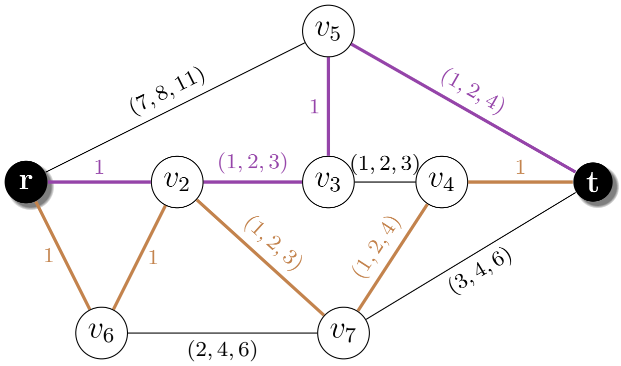


Figure 5-2: *Graph $\tilde{G}$ with the pair of edge-disjoint paths (P_3, P_4).*

respectively. Thus, we can conclude that $\forall \alpha \in [0, 1]$, $S_T^{\alpha}(\tilde{C}_{P_3,P_4}) < S_T^{\alpha}(\tilde{C}_{P_1,P_2})$ and, therefore, $\tilde{C}_{P_3,P_4} \prec \tilde{C}_{P_1,P_2}$. This means that the pair of edge-disjoint paths (P_1, P_2) found is not an optimal solution.

Our goal is to find the pair of edge-disjoint paths whose sum of costs is the minimum, so the solution obtained in the example is not optimal. Therefore, we will dedicate ourselves to finding an algorithm that solves the stated problem and avoids the risk of suboptimality.

5.3.3 General structure for the shortest pair of edge-disjoint paths

This section briefly describes the appropriate structure that the shortest pair of edge-disjoint paths should have. Assuming that S is the shortest path between the pair of vertices r and t, the following notations are necessary to give the valid configurations,

$\mathfrak{S}$: Set of all paths with some segments overlapping with path S,

$\mathfrak{S}'$: Set of all paths without any segment overlapping with path S,

P: Path that belongs to $\mathfrak{S}$,

P': Path that belongs to $\mathfrak{S}'$,

where we will call a *segment* of a path to an edge or several consecutive edges that belong to the path.

We denote $\gamma_1 \times \gamma_2$ as a pair of paths in $\tilde{G}$, where "$\times$" stands for the combination of individual paths γ_1 and γ_2. Due to the condition of edge-disjointness, $\gamma_1 \times \gamma_2$ has one of the structures in expression 5.7,

$$(\gamma_1 \times \gamma_2)_{\text{edge-disjoint}} = \{S \times P', P_1 \times P_2, P_1' \times P_2', P \times P'\} \tag{5.7}$$

For simplicity, we consider a pair of paths $P_1 \times P_2$ where each path contains a single (but different) segment of path S. From this condition, there exist three possible cases about the paths P_i $(i = 1, 2)$ and P':

(a) Only one of these two segments is at one of the endpoint vertices.

(b) Neither of the two segments is at an endpoint vertex.

(c) Each of these segments is at an endpoint vertex.

We also assume that the paths constituting the pair $P_1 \times P_2$ do not intersect each other in any vertex.

In the following, we will analyze the appropriate topological structure of the shortest pair of edge-disjoint paths, both when the shortest path S is unique and when other paths have equal cost.

When path S is unique

When S is the only shortest path in $\tilde{G}$, neither configurations $P_1' \times P_2'$ or $P \times P'$ can be candidates to the shortest pair of edge-disjoint paths. It is easy to notice that the total cost of each of these configurations will always be larger than the cost of the configuration $S \times P'$ with the structure shown in figure 5-3.

Finally, the shortest pair of edge-disjoint paths have one of the two configurations

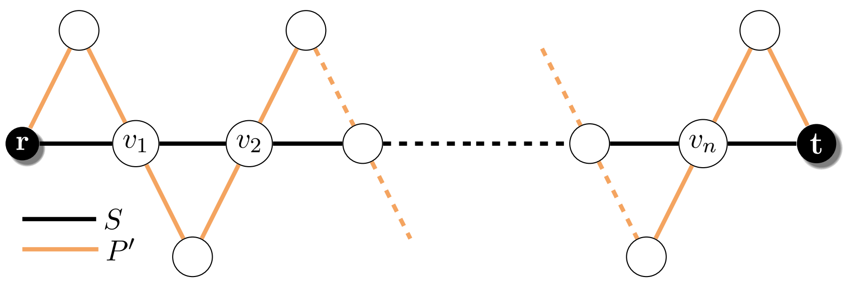


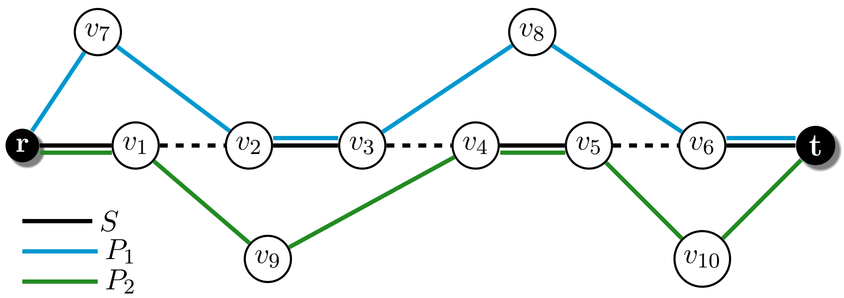
Figure 5-3: *Pair configuration $S \times P'$ where P' intercepts path S at certain number of vertices and has no overlapping edges with S.*

in equation 5.8,

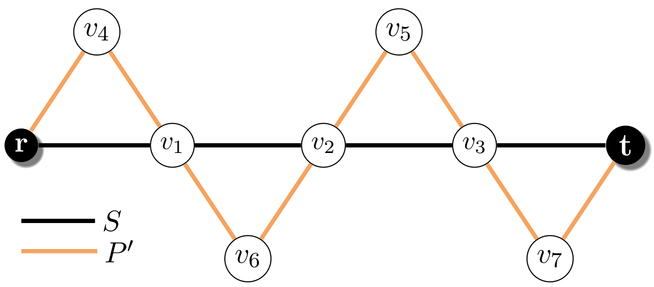
$$(\gamma_1 \times \gamma_2)_{\substack{\text{edge-disjoint} \\ \text{shortest pair}}} \in \{S \times P',\, P_1 \times P_2\} \tag{5.8}$$

where paths P_1 and P_2 compose the configuration $P_1 \times P_2$ described in case (c).

A *break* is a segment of path S that does not belong to either of the paths forming the pair of paths but is adjacent at each endpoint vertices to an edge of each path. Figures 5-4(a) and 5-4(b) show a particular example of valid configurations for the pairs $P_1 \times P_2$ and $S \times P'$, respectively. Figure 5-4(a) shows the shortest path $S =< r, v_1, \ldots, v_2, v_3, \ldots, v_4, v_5, \ldots, v_6, t >$ where the segments $(v_1, \ldots, v_2)$, $(v_3, \ldots, v_4)$ and $(v_5, \ldots, v_6)$ are breaks. In figure 5-4(b), path S is one of the paths that form the pair. The other path, P', does not contain any segment of S but intersects it at the vertices v_1, v_2, and v_3.



(a) Configuration $P_1 \times P_2$



(b) Configuration $S \times P'$

Figure 5-4: *Valid configurations for the shortest pair of edge-disjoint paths when the shortest path S is unique.*

To summarize, the shortest pair of edge-disjoint paths when S is the only shortest path in $\tilde{G}$ is among the valid configurations given in equation 5.8.

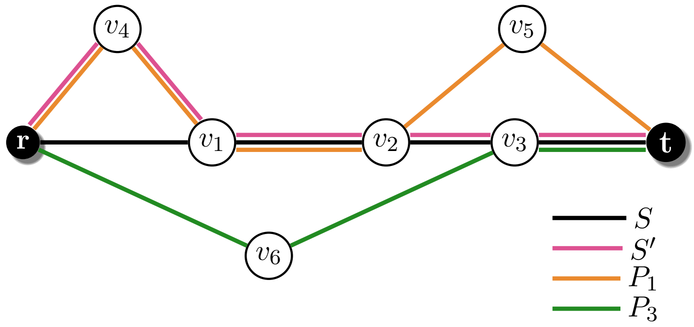
When path S is not unique

Let us assume there exist a path S' such that $\tilde{C}_{S'} = \tilde{C}_S$. If S' is edge-disjoint with S, then the shortest pair of edge-disjoint paths would be $S \times S'$. Throughout this section, we will assume that, in addition to S, there is a second shortest path S'

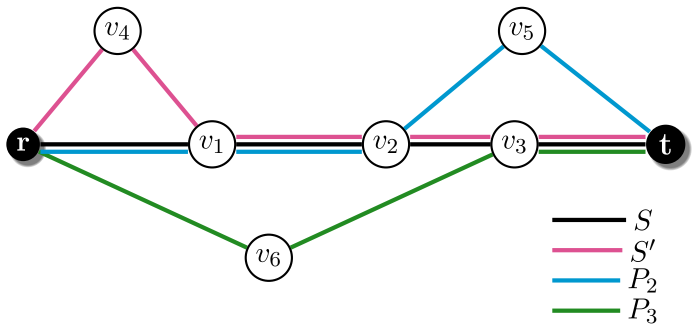
with one or more common edges with S. In this case, besides the valid configurations when the shortest path is unique ($P_1 \times P_2$ as shown in figure 5-4(a) and $S \times P'$ as shown in figure 5-4(b)), $P \times P'$ and additional configurations $P_1 \times P_2$ (satisfying the cases (a) and (b) in section 5.3.3) are also valid.

Figure 5-5 shows an example where an additional configuration of type $P_1 \times P_2$ that satisfies the case (a) is a valid configuration when S is not unique. Let $S = \langle r, v_1, v_2, v_3, t\rangle$ and $S' = \langle r, v_4, v_1, v_2, v_3, t\rangle$ be the shortest paths, represented in black and pink lines, respectively. S and S' have common edges (v_1, v_2), (v_2, v_3) and (v_3, t). Let us consider the paths P_1 and P_3, represented in orange and green lines in 5-5(a), respectively, and path P_2 in a blue line shown in 5-5(b). The pair $P_1 \times P_3$ is edge-disjoint, and only P_3 overlaps the shortest path S in the edge (v_3, t) (case (a)). On the other hand, the pair $P_2 \times P_3$ is also edge-disjoint, but both paths P_2 and P_3 overlap S in the edges (r, v_1) and (v_3, t), respectively (case (c)). Since path S' has the same cost as S, $P_1 \times P_3$ has the same total cost as the pair $P_2 \times P_3$. Therefore, the pair $P_1 \times P_3$, which satisfies case (a), is also valid for the shortest pair of edge-disjoint paths.

Figure 5-6 shows an example of another valid configuration structured as in case



(a) Valid configuration $P_1 \times P_3$ where only P_3 overlaps S in a segment at one of its endpoint vertices (case (a))



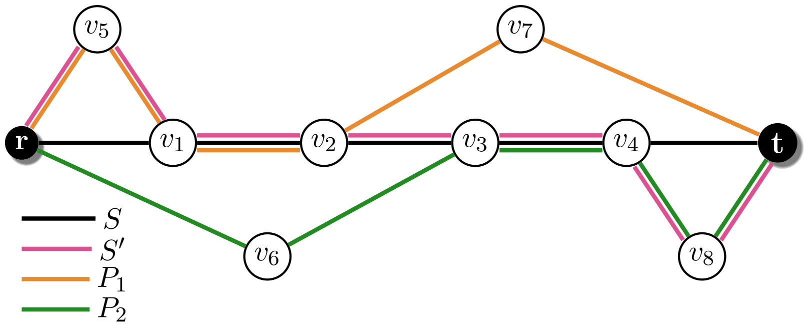
(b) Configuration $P_2 \times P_3$ with the same structure as in figure 5-4(a) and the same total cost as $P_1 \times P_3$.

Figure 5-5: *Example of an additional valid configuration satisfying the case (a) for the pair of edge-disjoint paths when the shortest path S is not unique.*

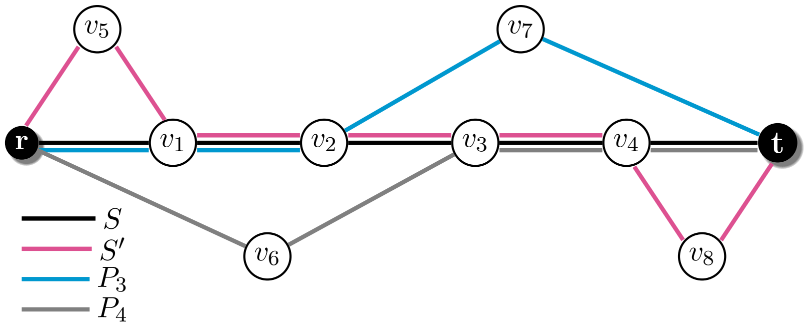
.

(b) when S is not unique. $S = \langle r, v_1, v_2, v_3, v_4, t\rangle$ and $S' = \langle r, v_5, v_1, v_2, v_3, v_4, v_8, t\rangle$ are the shortest paths, both with the same cost and represented in black and pink lines, respectively. Both paths have the common segment $\langle v_1, v_2, v_3, v_4\rangle$. In the pair $P_1 \times P_2$ shown in 5-6(a), with P_1 and P_2 in orange and green lines, respectively, neither of the paths overlap S at its endpoint vertices (case (b)). On the other hand, the pair $P_3 \times P_4$ shown in 5-6(b) is also edge-disjoint, but both paths P_3 and P_4 overlap S in the edges (r, v_1) and (v_4, t), respectively (case (c)). Notice that, since path S' has the same cost as S, $P_1 \times P_2$ has the same total cost as $P_3 \times P_4$. Therefore, the pair $P_1 \times P_2$, which satisfies case (b), is also valid for the shortest pair of edge-disjoint paths.

Figure5-7 shows an example of a configuration $P \times P'$, which can also be



(a) Valid configuration $P_1 \times P_2$ where neither of its paths overlaps S in a segment at its endpoint vertices (case (b))



(b) Configuration $P_3 \times P_4$ with the same structure as in figure 5-4(a) and the same total cost as $P_1 \times P_2$

Figure 5-6: *Example of an additional valid configuration satisfying the case (b) for the shortest pair of edge-disjoint paths when the shortest path S is not unique.*

.

valid for the shortest pair of edge-disjoint paths when S is not unique. If path $P = \langle r, v_1, v_2, v_3, v_6, t\rangle$ has the same cost as path $S = \langle r, v_1, v_2, v_3, v_4, t\rangle$, we can define P as S'. Then path S is not unique. Additionally, P has the common segment $\langle v_1, v_2, v_3\rangle$ with S. Therefore, the pair $P \times P'$, where path P' is represented in blue lines, has the same total cost as the pair $S \times P'$. Notice that $(S \times P')$ has the same structure as the valid configuration shown in figure 5-4(b) when S is unique.

To summarize, when S is not the only shortest path in $\tilde{G}$, besides the configurations that are valid when S is unique ($P_1 \times P_2$ and $S \times P'$ shown in figure 5-3), the pair $P \times P'$ and additional configurations of type $P_i \times P_j$, $i, j \in \mathbb{N}$, $i, j \neq 1, 2$ also become authentic. However, the configuration $P \times P'$ is valid with equal total cost as $S \times P'$, and any of the additional valid configurations $P_i \times P_j$ has the same total cost as the pair $P_1 \times P_2$. Therefore, *when S is not unique, we can consider the configurations given in equation 5.8 as a sufficient solution for the shortest pair of edge-disjoint paths.* Section 5.3.4 provides a detailed explanation of the total cost for the configurations $P_1 \times P_2$ and $S \times P'$.

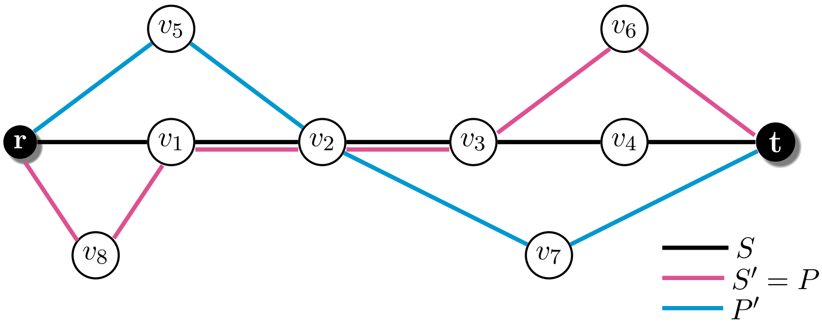


Figure 5-7: *Example of a valid configuration $P \times P'$ for the shortest pair of edge-disjoint paths when S is not unique.*

5.3.4 Total cost of the shortest pair of edge-disjoint paths

Once the topology of the shortest pair of edge-disjoint paths is given, we analyze in this section the total cost of the pair of paths, whether it has the configuration $P_1 \times P_2$ or $S \times P'$. For this purpose, we must pay special attention to the fuzzy character of the graph.

Given the configuration $P_1 \times P_2$, we will consider the *coalescence of the endpoint vertices at each break*. We acknowledge that this transformation is not physically possible, but from a formal or abstract perspective, we refer to it. For instance, in the configuration $P_1 \times P_2$ shown in figure 5-4(a), segments $(v_1, \ldots, v_2)$, $(v_3, \ldots, v_4)$ and $(v_5, \ldots, v_6)$ are the breaks of the pair of paths. Thus, the coalescence of vertices v_1 and v_2, vertices v_3 and v_4, and vertices v_5 and v_6 lead us to the vertices v_1, v_2, and v_3 in the configuration $S \times P'$ in figure 5-4(b), respectively. Notice that these vertices belong to path S and have, at least, degree 4. Therefore, let us assume that both paths intercept in a vertex with degree 4 of path S by the coalescence of the endpoint vertices at each break.

With the transformation exposed above, we intend to find a correct way to describe, only through the costs of the edges, the assumption of coalescence between the endpoint vertices on breaks. To do this, we use the term *complementary of a triangular fuzzy number* whose detailed definition the reader can find in chapter 2.

Figure 5-8 shows the edges and vertices that belong to paths P_1, P_2, and S. We replace each edge (i, j), $i, j \in \mathbb{N}$ in path S by two overlapped arcs with opposite directions, $\overrightarrow{(v_i, v_j)}$ and $\overrightarrow{(v_j, v_i)}$. Thus, from a topological vision, besides S, we obtain the *auxiliary path* $P_{\text{aux}} = \langle r, v_7, v_2, v_1, v_9, v_4, v_3, v_8.v_6, v_5, v_{10}, t\rangle$. Notice that each of the oppositely directed arcs in every break belongs to S and P_{aux}, respectively.

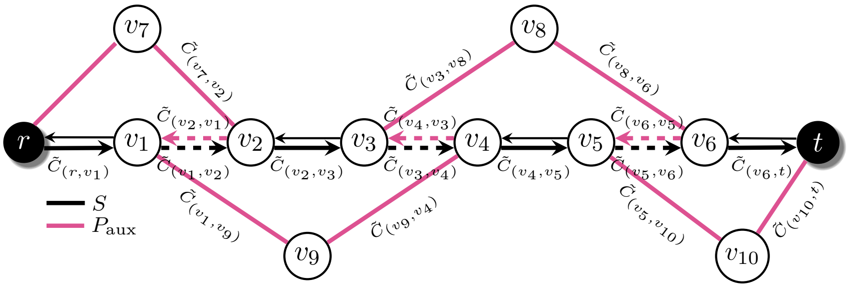


Figure 5-8: *Paths S and P_{aux} are created after replacing the edges in S with oppositely directed arcs.*

In path S, the cost of each segment $\overrightarrow{(v_i, v_j)}$ coincides with the cost of its corresponding edge (v_i, v_j), and its opposed arc $\overrightarrow{(v_j, v_i)}$ has a cost equal to the complementary of the cost of (v_i, v_j). For instance, lets take the segments $(v_1, \ldots, v_2)$, $(v_3, \ldots, v_4)$ and $(v_5, \ldots, v_6)$ in figure 5-8. Notice that these segments are the breaks for the valid configuration $P_1 \times P_2$ referred to in figure 5-4(a). Also, these segments are, in fact, the common segments between the paths S and P_{aux}. Due to the arithmetic properties of fuzzy numbers, assuming the cost of edges in $\tilde{G}$ as triangular fuzzy numbers, the sum of the costs of edges in a segment result in a triangular fuzzy number. Therefore, without loss of generality, in the rest of this section we will assume the segments $(v_1, \ldots, v_2)$, $(v_3, \ldots, v_4)$ and $(v_5, \ldots, v_6)$ as single edges (v_1, v_2), (v_3, v_4) and (v_5, v_6), respectively, with cost defined as a triangular fuzzy number. Denoting $\tilde{C}_{(v_1,v_2)} = (a_{1,2}, b_{1,2}, c_{1,2})$, $\tilde{C}_{(v_3,v_4)} = (a_{3,4}, b_{3,4}, c_{3,4})$ and $\tilde{C}_{(v_5,v_6)} = (a_{5,6}, b_{5,6}, c_{5,6})$, their oppositely directed arcs will have the costs described in equation 5.9,

$$
\begin{aligned}
\tilde{C}_{\overrightarrow{(v_1,v_2)}} = \tilde{C}_{(v_1,v_2)} = (a_{1,2}, b_{1,2}, c_{1,2}) \;\Rightarrow\; & \tilde{C}_{\overrightarrow{(v_2,v_1)}} = -\tilde{C}_{(v_1,v_2)} \\
& = (-c_{1,2}, -b_{1,2}, -a_{1,2}) \\
\tilde{C}_{\overrightarrow{(v_3,v_4)}} = \tilde{C}_{(v_3,v_4)} = (a_{3,4}, b_{3,4}, c_{3,4}) \;\Rightarrow\; & \tilde{C}_{\overrightarrow{(v_4,v_3)}} = -\tilde{C}_{(v_3,v_4)} \\
& = (-c_{3,4}, -b_{3,4}, -a_{3,4}) \\
\tilde{C}_{\overrightarrow{(v_5,v_6)}} = \tilde{C}_{(v_5,v_6)} = (a_{5,6}, b_{5,6}, c_{5,6}) \;\Rightarrow\; & \tilde{C}_{\overrightarrow{(v_6,v_5)}} = -\tilde{C}_{(v_5,v_6)} \\
& = (-c_{5,6}, -b_{5,6}, -a_{5,6})
\end{aligned} \tag{5.9}
$$

For each break (v_i, v_j), the sum of the costs of its corresponding opposite arcs results in a new triangular fuzzy number described in equation 5.10,

$$\tilde{C}_{(v_i,v_j)} + (-\tilde{C}_{(v_i,v_j)}) = (a_{i,j} - c_{i,j}, 0, c_{i,j} - a_{i,j}), \quad i = \{1,3,5\},\; j = \{2,4,6\} \tag{5.10}$$

The fuzzy number $(a_{i,j}-c_{i,j},0,c_{i,j}-a_{i,j})$ belongs to the class of special triangular fuzzy numbers called *N-zero fuzzy number*, defined in definition 2.4.25. More precisely, it is an N_2-zero fuzzy number since $a_{i,j}-c_{i,j}<0<c_{i,j}-a_{i,j}$. Due to the interpretation of N-zero fuzzy numbers, we say that the sum of the costs of both arcs is a " value around zero". Thus, the cancellation of the costs of both arcs is also subjected to uncertainty.

The total cost of the pair $S \times P_{\text{aux}}$ is defined in expression 5.11,

$$
\begin{aligned}
\tilde{C}_{S\times P_{\text{aux}}} &= \tilde{C}_S + \tilde{C}_{P_{\text{aux}}} \qquad (5.11)\\
&= \tilde{C}_{(r,v_1)} + \tilde{C}_{(v_1,v_2)} + \tilde{C}_{(v_2,v_3)} + \tilde{C}_{(v_3,v_4)} + \tilde{C}_{(v_4,v_5)} + \tilde{C}_{(v_5,v_6)} + \tilde{C}_{(v_6,t)} +\\
&\quad + \tilde{C}_{(r,v_7)} + \tilde{C}_{(v_7,v_2)} + \tilde{C}_{\overrightarrow{(v_2,v_1)}} + \tilde{C}_{(r,v_7)} + \tilde{C}_{(v_7,v_2)} + \tilde{C}_{(v_2,v_1)} + \tilde{C}_{(v_1,v_9)} +\\
&\quad + \tilde{C}_{(v_9,v_4)} + \tilde{C}_{\overrightarrow{(v_4,v_3)}} + \tilde{C}_{(v_3,v_8)} + \tilde{C}_{(v_8,v_6)} + \tilde{C}_{\overrightarrow{(v_6,v_5)}} + \tilde{C}_{(v_5,v_{10})} + \tilde{C}_{(v_{10},t)}
\end{aligned}
$$

By the replacement of $\tilde{C}_{\overrightarrow{(v_2,v_1)}}$, $\tilde{C}_{\overrightarrow{(v_4,v_3)}}$, and $\tilde{C}_{\overrightarrow{(v_6,v_5)}}$ with their corresponding complementary of the cost of the associated edge, $\tilde{C}_{S\times P_{\text{aux}}}$ contains the total cost of the configuration $P_1 \times P_2$, $\mathbf{\tilde{C}_{P_1\times P_2}}$, as shown in figure 5-9 and described in equation 5.12.

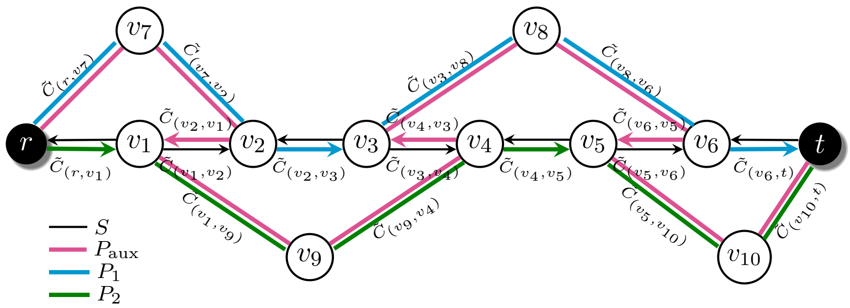


Figure 5-9: *The pair $P_1 \times P_2$ is included in $S \times P_{aux}$ configuration*

$$
\begin{aligned}
\tilde{C}_{S\times P_{\text{aux}}} &= \tilde{C}_{(r,v_7)} + \tilde{C}_{(v_7,v_2)} + \tilde{C}_{(v_2,v_3)} + \tilde{C}_{(v_3,v_8)} + \tilde{C}_{(v_8,v_6)} + \tilde{C}_{(v_6,t)} + \tilde{C}_{(r,v_1)} +\\
&\quad + \tilde{C}_{(v_1,v_9)} + \tilde{C}_{(v_9,v_4)} + \tilde{C}_{(v_4,v_5)} + \tilde{C}_{(v_5,v_{10})} + \tilde{C}_{(v_{10},t)} +\\
&\quad + \tilde{C}_{(v_1,v_2)} + (-\tilde{C}_{(v_1,v_2)}) + \tilde{C}_{(v_3,v_4)} + (-\tilde{C}_{(v_4,v_3)}) + \tilde{C}_{(v_5,v_6)} + (-\tilde{C}_{(v_6,v_5)})\\
&= \mathbf{\tilde{C}_{P_1\times P_2}} + \tilde{C}_{(v_1,v_2)} + (-\tilde{C}_{(v_1,v_2)}) + \tilde{C}_{(v_3,v_4)} + (-\tilde{C}_{(v_3,v_4)}) + \tilde{C}_{(v_5,v_6)} +\\
&\quad + (-\tilde{C}_{(v_5,v_6)}) \qquad (5.12)
\end{aligned}
$$

The remaining members in equation 5.12 correspond to the sum of the costs of each pair of oppositely directed arcs in the breaks (v_1,v_2), (v_3,v_4), and (v_5,v_6). The fuzzy number resulting from this summation has a structure corresponding to the N_2-zero triangular fuzzy numbers described in equation 5.10, as is shown in equation 5.13,

$$\begin{aligned}\tilde{C}_{S\times P_{\text{aux}}} &= \tilde{\mathbf{C}}_{\mathbf{P_1}\times\mathbf{P_2}} + (a_{1,2}-c_{1,2},0,c_{1,2}-a_{1,2}) + (a_{3,4}-c_{3,4},0,c_{3,4}-a_{3,4})+\\ &\quad + (a_{5,6}-c_{5,6},0,c_{5,6}-a_{5,6})\\ &= \tilde{\mathbf{C}}_{\mathbf{P_1}\times\mathbf{P_2}}+\\ &\quad + \Big(a_{1,2}+a_{3,4}+a_{5,6}-c_{1,2}-c_{3,4}-c_{5,6},0,c_{1,2}+c_{3,4}+c_{5,6}-a_{1,2}-\\ &\quad - a_{3,4}-a_{5,6}\Big) \qquad (5.13)\end{aligned}$$

where $a_{1,2}+a_{3,4}+a_{5,6}-c_{1,2}-c_{3,4}-c < 0 < c_{1,2}+c_{3,4}+c_{5,6}-a_{1,2}-a_{3,4}-a_{5,6}$

We can generalize the result in equation 5.13. Let us assume we have a pair $S\times P_{\text{aux}}$ where a valid configuration for the shortest pair of edge-disjoint paths with structure $P_1\times P_2$ is included. Also, let us assume there are M breaks between the paths P_1 and P_2. The cost of $S\times P_{\text{aux}}$ is equal to the sum of the total cost of $P_1\times P_2$ and an N_2-zero triangular fuzzy number, which corresponds to the sum of the costs of the overlapping arcs associated with each break. When there are no breaks ($M=0$), the shortest pair of edge-disjoint paths is of type $S\times P'$ and the cost of the pair $S\times P_{\text{aux}}$ is exactly equal to the cost of $S\times P'$.

$$\tilde{C}_{S\times P_{\text{aux}}} = \begin{cases}\tilde{C}_{P_1\times P_2} + \sum_{m=1}^{M}(a-c,0,c-a)_m, & \text{if } M\neq 0\\ \tilde{C}_{S\times P'}, & \text{if } M=0\end{cases} \qquad (5.14)$$

By arithmetic properties of triangular fuzzy numbers, the sum of the N_2-zero triangular fuzzy numbers that correspond to each break is also an N_2-zero triangular fuzzy number, as is described in equation 5.15,

$$\begin{aligned}\sum_{m=1}^{M}(a-c,0,c-a)_m &= \left(\sum_{m=1}^{M}(a-c)_m, 0, \sum_{m=1}^{M}(c-a)_m\right)\\ &= \left(\sum_{m=1}^{M}a_m-\sum_{m=1}^{M}c_m, 0, \sum_{m=1}^{M}c_m-\sum_{m=1}^{M}a_m\right)\\ &= (A_M-C_M,0,C_M-A_M) \qquad (5.15)\end{aligned}$$

where $A_M=\sum_{m=1}^{M}a_m$ and $C_M=\sum_{m=1}^{M}c_m$.

To summarize, once we find the paths S and P_{aux}, we calculate the total cost of $S\times P_{\text{aux}}$ and search for breaks. We perform the summation of the costs of its corresponding oppositely directed arcs for each break, obtaining an N_2-zero fuzzy number. Then, we apply the equation 5.15, where the sum of the M N_2-zero triangular fuzzy numbers is implemented. At last, we calculate the total cost of the shortest pair of edge-disjoint paths $\gamma_1 \times \gamma_2$ described in expression 5.8, according to equation 5.16,

$$\tilde{C}_{\gamma_1 \times \gamma_2} = \begin{cases} \tilde{C}_{S \times P_{\text{aux}}} - (A_M - C_M, 0, C_M - A_M) & \begin{array}{c}\text{if } M \neq 0 \\ (\gamma_1 \times \gamma_2 = P_1 \times P_2)\end{array} \\ \tilde{C}_{S \times P_{\text{aux}}} & \begin{array}{c}\text{if } M = 0 \\ (\gamma_1 \times \gamma_2 = S \times P')\end{array} \end{cases} \tag{5.16}$$

Returning to the example in section 5.3.2,

Example 5.3.4.1.
The shortest path of $\tilde{G}$ *is* $S = \langle r, v_2, v_3, v_4, t\rangle$ *with* $\tilde{C}_S = (4, 6, 8)$. *Replacing each edge in* S *by its oppositely directed arc, we obtain the new shortest path* $P_{aux} = \langle r, v_6, v_2, v_7, v_4, v_3, v_5, t\rangle$ *with* $\tilde{C}_{P_{aux}} = (3, 7, 13)$. *We observe that both paths contain the oppositely directed arcs in the edge* (v_3, v_4), *making this a* break. *Thus, we get the valid configuration* $P_1 \times P_2$ *with* $P_1 = \langle r, v_2, v_3, v_5, t\rangle$ *and* $P_2 = \langle r, v_6, v_2, v_7, v_4, t\rangle$. *Figure 5-10 shows the pairs* $S \times P_{aux}$ *and* $P_1 \times P_2$ *and the costs of each edge and arc.*

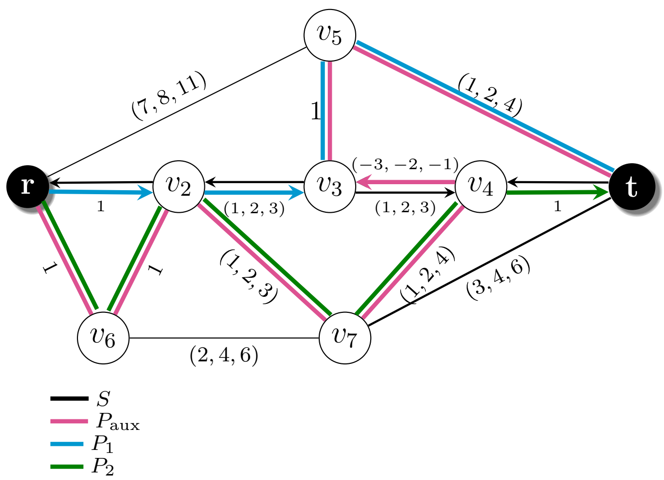


Figure 5-10: *Graph* $\tilde{G}$ *with the paths* S, P_{aux}, P_1 *and* P_2.

Applying equation 5.16, we compute the total cost of $P_1 \times P_2$,

$$\begin{aligned} \tilde{C}_{P_1 \times P_2} &= \tilde{C}_{S \times P_{aux}} - \left[\tilde{C}_{\overrightarrow{(3,4)}} + \tilde{C}_{\overrightarrow{(4,3)}}\right] \\ &(7, 13, 21) - (-2, 0, 2) \\ &= (\mathbf{9}, \mathbf{13}, \mathbf{19}) \end{aligned}$$

$P_1 \times P_2$ *is the shortest pair of edge-disjoint paths already found by us in example 5.3.2.1. We compute* $\tilde{C}_{P_1 \times P_2}$ *using the pair* $S \times P_{aux}$ *and the break* (v_3, v_4) *included in the shortest pair of edge-disjoint paths.*

5.3.5 Definition of the modified graph $\tilde{G}'$

From the analysis made in previous sections, finding the shortest pair of edge-disjoint paths in $\tilde{G}$ requires, at first, the search of the shortest path S. Secondly, it is necessary to find a second path P_{aux}. P_{aux} is the shortest path found in a new *mixed graph*, denoted as $\tilde{G}' = (V, A', \mathfrak{C}')$, which is a modification of the original $\tilde{G}$ and is defined as follows:

- Set V of vertices is the same as in the original graph.
- Edges and arcs in A' are defined as: each edge (v_i, v_j) belonging to the path S in $\tilde{G}$ is replaced by the arc $\overrightarrow{(v_j, v_i)}$. The rest of the edges in $\tilde{G}$ remain the same.
- Costs in $\mathfrak{C}'$ are defined as: the cost of each arc is the complementary of the corresponding edge cost in $\tilde{G}$. The costs of the remaining edges are the same as in $\tilde{G}$.

Under the conditions stated above, path P_{aux} has segments overlapping with S where:

(i) On the overlapped segments, the arcs of P_{aux} in $\tilde{G}'$ are oriented towards r, and the arcs of the shortest path S in $\tilde{G}$ are oriented towards t (oppositely directed arcs).

(ii) We define the cost of each arc in P_{aux} as the *complementary* of the original cost (triangular fuzzy number) of its corresponding edge in $\tilde{G}$.

5.4 Algorithm to find the shortest pair of edge-disjoint paths

This section proposes an algorithmic solution to searching the shortest pair of edge-disjoint paths in a type V fuzzy graph. So far, we described the structure that the pair of paths must have, together with its total cost. Also, to obtain the pair of paths, it is necessary to find the path P_{aux} on a new graph $\tilde{G}'$. This graph is a modification of the original graph. We can now create an algorithm based on our concepts capable of finding a solution to the problem.

We obtain the shortest pair of edge-disjoint paths with fuzzy costs by applying twice an algorithm that searches for the shortest path in a type V fuzzy graph. The first run is made on $\tilde{G}$, where we obtain the shortest path S. The second run is performed on $\tilde{G}'$ and finds the path P_{aux} that must satisfy the conditions (i)-(ii). Once we complete both runs of the algorithm, the pair $\gamma_1 \times \gamma_2$ is obtained by either of two cases: first, by deleting the overlapped segments between paths S and P_{aux} (breaks) resulting in the pair $P_1 \times P_2$; and second, in case there are no breaks, the pair is $S \times P'$ where $P' = P_{\text{aux}}$.

5.4.1 Conditions for the nonexistence of negative cycles

Since we define the cost of edges and arcs as triangular fuzzy numbers, in the following, we will name this as *triangular fuzzy cost* without loss of generality. Also, we will call the *total cost* of a cycle to the sum of the costs of its edges and arcs.

The costs of the introduced arcs in $\tilde{G}'$ are negative triangular fuzzy numbers. Thus, cycles whose total cost is a negative triangular fuzzy number (negative cycles) could be possible. Let us assume that a standard algorithm for the search of the shortest path in a graph is applied on $\tilde{G}'$, and for the comparison of the costs of the paths, we use the ranking method proposed by [Yu and l. Q. Dat 2014]. Then, given an arbitrary α, if a path falls into a negative cycle, at every tour made, chains with smaller fuzzy numbers could be created, leading to the non-convergence of the algorithm. Therefore, our focus lies on *finding the conditions under which graph $\tilde{G}'$ has no negative cycles.*

Any cycle in $\tilde{G}'$ is either:

Simple cycle: Cycle composed only of edges with non-negative triangular fuzzy costs.

Mixed cycle: Cycle composed of arcs with negative triangular fuzzy costs replacing the edges of the shortest path S in $\tilde{G}$ and edges of $\tilde{G}'$ with non-negative triangular fuzzy costs. According to its total cost, a mixed cycle can be either positive or negative.

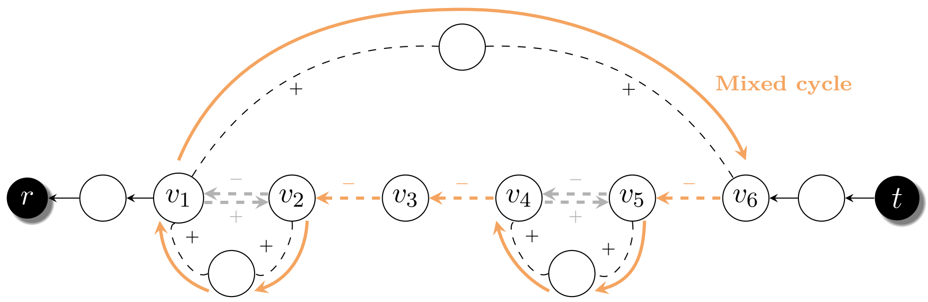
Our goal is to find the conditions under which any mixed cycle in $\tilde{G}'$ is guaranteed to be non-negative. In other words, *under which circumstances the total cost of every mixed cycle in $\tilde{G}'$ is a non-negative triangular fuzzy number.*

Non-negative mixed cycle in $\tilde{\mathbf{G}}'$

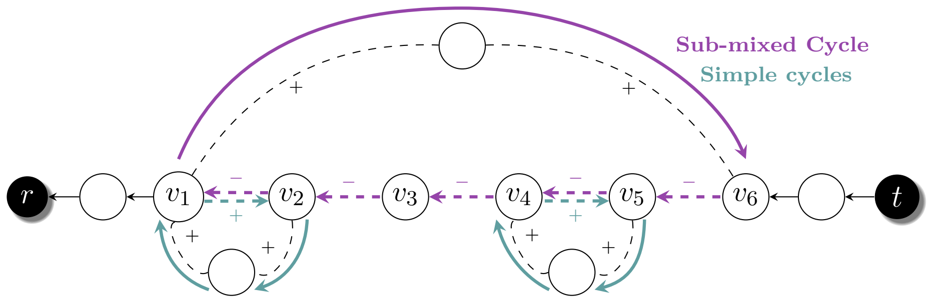
Figure 5-11(a) shows an example of a mixed cycle in $\tilde{G}'$ wherein its corresponding original graph $\tilde{G}$ the shortest path is $S = \langle r, \ldots, v_1, v_2, v_3, v_4, v_5, v_6, \ldots, t\rangle$. Each of the edges (v_1, v_2) and (v_4, v_5), since they belong to path S, are replaced by two opposite directed arcs whose costs correspond to the value of the original edge cost and the complementary of this value, respectively. The signs "+" and "-" on edges and arcs represent a non-negative and negative triangular fuzzy cost, respectively. Thus, the mixed cycle in figure 5-11(a) is equivalent to the set of cycles in figure 5-11(b). Arcs $\overrightarrow{(v_2, v_1)}$ and $\overrightarrow{(v_5, v_4)}$ do not belong to the mixed cycle; they are part of a new cycle called a *sub-mixed cycle.*

Sub-mixed cycle : Cycle composed of a set of contiguous arcs with non-negative fuzzy cost, and another set of contiguous arcs, with negative fuzzy cost, directed towards the source vertex.

In figure 5-11(b), arcs $\overrightarrow{(v_2, v_1)}$, $\overrightarrow{(v_3, v_2)}$, $\overrightarrow{(v_4, v_3)}$, $\overrightarrow{(v_5, v_4)}$, and $\overrightarrow{(v_6, v_5)}$ are part of the sub-mixed cycle (red cycle). On the other hand, arcs $\overrightarrow{(v_1, v_2)}$ and $\overrightarrow{(v_4, v_5)}$, both with positive triangular fuzzy costs, are part of simple cycles (green cycles). Consequently, the mixed cycle in figure 5-11(a) contains a set of contiguous arcs with negative triangular fuzzy cost included in a sub-mixed cycle and a set of edges with positive triangular fuzzy cost which are part of simple cycles. In other words, any mixed cycle in $\tilde{G}'$ is composed of a sub-mixed cycle and a set of edges with positive triangular fuzzy costs that are not part of the sub-mixed cycle. *To guarantee the non-negativity of a mixed cycle, we will find under which conditions the included sub-mixed cycle is non-negative.*



(a)



(b)

Figure 5-11: *Mixed cycle in $\tilde{G}'$ composed of edges and arcs from a sub-mixed cycle and edges from simple cycles.*

Non-negative sub-mixed cycle $\tilde{G}'$

Let us consider a sub-mixed cycle in $\tilde{G}'$ with the structure shown in figure 5-11(b), composed of J edges and I arcs, with $I, J \in \mathbb{N}$.

Below, we define the *total cost of the sub-mixed cycle.*

The following paths are included in the sub-mixed cycle:

- $\mathcal{S} = S_{(v_n,v_m)}$: subpath of S from v_n to v_m composed by I arcs.
  We denote $(a_i^{\mathcal{S}}, b_i^{\mathcal{S}}, d_i^{\mathcal{S}})$ with $i = 1, \dots I$ the cost of each arc in $\mathcal{S}$. The total

cost of $\mathcal{S}$ is defined in equation 5.17.

$$\begin{aligned}\tilde{C}_{\mathcal{S}} &= \tilde{C}_{\text{arc}_1} + \tilde{C}_{\text{arc}_2} + \ldots + \tilde{C}_{\text{arc}_I} \\ &= (a_1^{\mathcal{S}}, b_1^{\mathcal{S}}, d_1^{\mathcal{S}}) + (a_2^{\mathcal{S}}, b_2^{\mathcal{S}}, d_2^{\mathcal{S}}) + \ldots + (a_I^{\mathcal{S}}, b_I^{\mathcal{S}}, d_I^{\mathcal{S}}) \\ &= \left(\sum_{i=1}^{I} a_i^{\mathcal{S}}, \sum_{i=1}^{I} b_i^{\mathcal{S}}, \sum_{i=1}^{I} d_i^{\mathcal{S}} \right) \\ &= \left(A^{\mathcal{S}}, B^{\mathcal{S}}, D^{\mathcal{S}} \right) \end{aligned} \tag{5.17}$$

where $A^{\mathcal{S}} = \sum_{i=1}^{I} a_i^{\mathcal{S}}$, $B^{\mathcal{S}} = \sum_{i=1}^{I} b_i^{\mathcal{S}}$, $D^{\mathcal{S}} = \sum_{i=1}^{I} d_i^{\mathcal{S}}$ and $A^{\mathcal{S}} \leq B^{\mathcal{S}} \leq D^{\mathcal{S}}$.

- $\mathcal{S}^- = S_{(v_m,v_n)}$: subpath of S from v_m to v_n.
  The total cost of $\mathcal{S}^-$ is the complementary of $\tilde{C}_{\mathcal{S}}$ as the result of the sum of the complementary cost of each arc in $\mathcal{S}$, this is,

$$\begin{aligned}\tilde{C}_{\mathcal{S}^-} &= (-d_1^{\mathcal{S}}, -b_1^{\mathcal{S}}, -a_1^{\mathcal{S}}) + (-d_2^{\mathcal{S}}, -b_2^{\mathcal{S}}, -a_2^{\mathcal{S}}) + \ldots + (-d_I^{\mathcal{S}}, -b_I^{\mathcal{S}}, -a_I^{\mathcal{S}}) \\ &= \left(-\sum_{i=1}^{I} d_i^{\mathcal{S}}, -\sum_{i=1}^{I} b_i^{\mathcal{S}}, -\sum_{i=1}^{I} a_i^{\mathcal{S}} \right) \\ &= \left(-D^{\mathcal{S}}, -B^{\mathcal{S}}, -A^{\mathcal{S}} \right) \\ &= -\tilde{C}_{\mathcal{S}} \end{aligned} \tag{5.18}$$

- $P_{(v_n,v_m)}$: Path between the vertices v_n and v_m included in the sub-mixed cycle that does not belong to S. Let J be the number of edges in $P_{(v_n,v_m)}$ and (a_j^P, b_j^P, d_j^P) the cost of each edge in $P_{(v_n,v_m)}$, the total cost of this path is defined in equation 5.19,

$$\tilde{C}_{P_{(v_n,v_m)}} = \left(\sum_{j=1}^{J} a_j^P, \sum_{j=1}^{J} b_j^P, \sum_{j=1}^{J} d_j^P \right) = \left(A^P, B^P, D^P \right), \quad \text{with } A^P \leq B^P \leq D^P \tag{5.19}$$

Finally, the sub-mixed cycle is created by the paths $\mathcal{S}^-$ and $P_{(v_n,v_m)}$, as figure 5-12 shows.

The *total cost of the sub-mixed cycle* is the sum of the total cost of the paths $\mathcal{S}^-$ and $P_{(v_n,v_m)}$, i.e.,

$$\begin{aligned}\tilde{C}\left(\begin{array}{c}\text{sub-mixed}\\ \text{cycle}\end{array} \right) &= \tilde{C}_{P_{(v_n,v_m)}} + \tilde{C}_{\mathcal{S}^-} \\ &= \left(A^P, B^P, D^P \right) + \left(-D^{\mathcal{S}}, -B^{\mathcal{S}}, -A^{\mathcal{S}} \right) \\ &= \left(A^P - D^{\mathcal{S}}, B^P - B^{\mathcal{S}}, D^P - A^{\mathcal{S}} \right) \end{aligned} \tag{5.20}$$

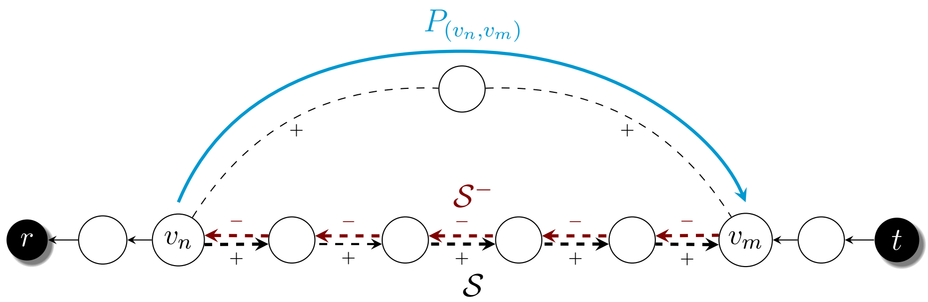


Figure 5-12: *Paths $\mathcal{S}^-$ and $P_{(v_n,v_m)}$ form the sub-mixed cycle in graph $\tilde{G}'$.*

where $A^P - D^{\mathcal{S}} < B^P - B^{\mathcal{S}} < D^P - A^{\mathcal{S}}$.

As a remark we have that $D^P - A^{\mathcal{S}} \geq 0$:

Proof.

(Proof by contradiction)

Let us assume that $D^P - A^{\mathcal{S}} < 0$

$\Rightarrow \; D^P < A^{\mathcal{S}}$

$\Rightarrow \; 0 \leq A^P \leq B^P \leq D^P \underset{\text{(assumption)}}{<} A^{\mathcal{S}} \leq B^{\mathcal{S}} \leq D^{\mathcal{S}}$

$\Rightarrow \; A^P \underset{\text{(I)}}{<} A^{\mathcal{S}}, \quad B^P \underset{\text{(II)}}{<} B^{\mathcal{S}} \quad \text{and} \quad D^P \underset{\text{(III)}}{<} D^{\mathcal{S}}$

In order to obtain the expression of the Total Integral for $\tilde{C}_{P_{(v_n,v_m)}}$ and $\tilde{C}_{\mathcal{S}}$,

first, we multiply (III) by $\alpha \in [0,1]$ and (I) by $(1-\alpha) \in [0,1]$

$\Rightarrow \; (1-\alpha)A^P \underset{\text{(IV)}}{\leq} (1-\alpha)A^{\mathcal{S}} \quad \text{and} \quad \alpha D^P \underset{\text{(V)}}{\leq} \alpha D^{\mathcal{S}}$

Performing the sum of the inequalities (II), (IV), and (V), we obtain the inequality 5.21,

$$(1-\alpha)A^P + B^P + \alpha D^P < (1-\alpha)A^{\mathcal{S}} + B^{\mathcal{S}} + \alpha D^{\mathcal{S}} \tag{5.21}$$

meaning that $\tilde{C}_{P_{(v_n,v_m)}} \prec \tilde{C}_{\mathcal{S}}$.
However, S is the shortest path of G, then by theorem A.0.1 its subpath $\mathcal{S}$ is also the shortest path from v_n to v_m. Therefore the inequality 5.21 does not hold.
Contradiction!

$\Rightarrow \; D^P - A^S \geq 0$ □

Therefore, $\tilde{C}\left(\begin{array}{c}\text{sub-mixed}\\ \text{cycle}\end{array}\right)$ is either an N-zero triangular fuzzy number or a positive triangular fuzzy number, i.e.:

$$\text{If } A^P - D^S \begin{cases} < 0 \quad \text{and} \begin{cases} B^P - B^S < 0 & \Rightarrow \quad N_1\text{-zero triangular fuzzy number} \\ B^P - B^S = 0 & \Rightarrow \quad N_2\text{-zero triangular fuzzy number} \\ B^P - B^S > 0 & \Rightarrow \quad N_3\text{-zero triangular fuzzy number} \end{cases} \\ \geq 0 \quad \Rightarrow \quad \text{positive triangular fuzzy number} \end{cases}$$

N-zero fuzzy numbers are not characterized as positive or negative (Chapter 2). Therefore, we need to establish a criterion to identify the *total cost of the sub-mixed cycle* as non-negative. Definition 5.4.1 exposes the conditions under which a sub-mixed cycle is non-negative.

Definition 5.4.1. *Non-negative sub-mixed cycle*

Given the type V fuzzy mixed graph $\tilde{G}'$, we say that a sub-mixed cycle with a structure as shown in figure 5-12 is non-negative if the inequality 5.22 holds,

$$S_T^\alpha\left(\tilde{C}\begin{pmatrix}\textit{sub-mixed}\\ \textit{cycle}\end{pmatrix}\right) \geq S_T^\alpha((0,0,0)) \tag{5.22}$$

for, at least, one value of $\alpha \in [0,1]$.

Following the above definition, we compute the Total Integral for the *total cost of a sub-mixed cycle* and the real number 0 (which could also be written in terms of a triangular fuzzy number as $(0,0,0)$), this is,

$$\begin{aligned} S_T^\alpha\left(\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}\right) &= \alpha S_L\left(\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}\right) + (1-\alpha) S_R\left(\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}\right) \\ &= \alpha\left(\frac{B^P - B^S}{2} + \frac{D^P - A^S}{2} - X_{\min}\right) + \\ &\quad (1-\alpha)\left(\frac{A^P - D^S}{2} + \frac{B^P - B^S}{2} - X_{\min}\right) \\ &= \alpha\left(\frac{D^P - A^S}{2}\right) + \frac{B^P - B^S}{2} + (1-\alpha)\left(\frac{A^P - D^S}{2}\right) - X_{\min} \end{aligned} \tag{5.23}$$

and

$$\begin{aligned} S_T^\alpha((0,0,0)) &= \alpha(0 + 0 - X_{\min}) + (1-\alpha)(0 + 0 - X_{\min}) \\ &= -X_{\min} \end{aligned} \tag{5.24}$$

where $X_{\min} = \inf\{(A^P - D^S), 0\} \leq 0$ and $\alpha \in [0,1]$.

Replacing the expressions of $S_T^\alpha\left(\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}\right)$ and $S_T^\alpha((0,0,0))$ in 5.22, we obtain that any sub-mixed cycle in $\tilde{G}'$ is non-negative for the values of α that make the inequality 5.25 to hold,

$$(1-\alpha)(A^P - D^S) + B^P - B^S + \alpha(D^P - A^S) \geq 0 \tag{5.25}$$

Given $S_{(v_n,v_m)}$ to be the shortest path from v_n to v_m, the inequality 5.26 is true:

$$\tilde{C}_{P_{(v_n,v_m)}} \succ \tilde{C}_{S_{(v_n,v_m)}} \tag{5.26}$$

which can also be written in terms of the Total Integral for the cost of paths $P_{(v_n,v_m)}$ and $S_{(v_n,v_m)}$, i.e.,

$$\Rightarrow\ S_T^\alpha\left(\tilde{C}_{P_{(v^n,v^m)}}\right) \geq S_T^\alpha\left(\tilde{C}_{S_{(v^n,v^m)}}\right) \qquad \text{for some } \alpha \in [0,1]$$

$$\Rightarrow\ (1-\alpha)A^P + B^P + \alpha D^P \geq (1-\alpha)A^S + B^S + \alpha D^S$$

$$\Rightarrow\ (1-\alpha)A^P + B^P + \alpha D^P \geq A^S - \alpha A^S + B^S + \alpha D^S$$

$$\Rightarrow\ (1-\alpha)A^P - \alpha D^S + \mathbf{B^P} - \mathbf{B^S} + \alpha D^P - A^S + \alpha A^S \geq 0^{1}$$

$$\Rightarrow\ (1-\alpha)A^P - \alpha D^S + 2\alpha\mathbf{D^S} - \mathbf{D^S} - 2\alpha\mathbf{D^S} + \mathbf{D^S} + \mathbf{B^P} - \mathbf{B^S} +$$
$$+\alpha D^P - A^S + \alpha A^S \geq 0, \qquad \text{since } D^S \geq 0$$

$$\Rightarrow\ (1-\alpha)A^P + (\alpha D^S - \mathbf{D^S}) - 2\alpha\mathbf{D^S} + \mathbf{D^S} + \mathbf{B^P} - \mathbf{B^S} +$$
$$+\alpha D^P - A^S + \alpha A^S \geq 0$$

$$\Rightarrow\ \left[(1-\alpha)A^P - (1-\alpha)D^S\right] - 2\alpha\mathbf{D^S} + \mathbf{D^S} + \mathbf{B^P} - \mathbf{B^S} +$$
$$+\alpha D^P - A^S + \alpha A^S \geq 0$$

$$\Rightarrow\ (1-\alpha)(\mathbf{A^P} - \mathbf{D^S}) + \mathbf{B^P} - \mathbf{B^S} + \alpha D^P - A^S$$
$$+\alpha A^S - 2\alpha\mathbf{D^S} + \mathbf{D^S} \geq 0$$

$$\Rightarrow\ (1-\alpha)(\mathbf{A^P} - \mathbf{D^S}) + \mathbf{B^P} - \mathbf{B^S} + \alpha D^P - A^S + \alpha A^S - 2\alpha\mathbf{A^S} +$$
$$+2\alpha\mathbf{A^S} - 2\alpha\mathbf{D^S} + \mathbf{D^S} \geq 0, \qquad \text{since } A^S \geq 0$$

$$\Rightarrow\ (1-\alpha)(\mathbf{A^P} - \mathbf{D^S}) + \mathbf{B^P} - \mathbf{B^S} + (\alpha D^P - \alpha A^S) + 2\alpha\mathbf{A^S} - A^S -$$
$$-2\alpha\mathbf{D^S} + \mathbf{D^S} \geq 0$$

$$\Rightarrow$$

$$(1-\alpha)(\mathbf{A^P} - \mathbf{D^S}) + \mathbf{B^P} - \mathbf{B^S} + \alpha(\mathbf{D^P} - \mathbf{A^S}) \geq A^S - 2\alpha A^S + 2\alpha D^S - D^S \tag{5.27}$$

Note that the left term in 5.27 coincides with the left term in equation 5.25. We are interested in searching for some $\alpha \in [0,1]$ for which the left term in 5.27

[1] In green: wanted terms. In orange: auxiliary terms

is non-negative. Thus, it is enough to consider only one of the following two possible cases:

(a) When $A^S - 2\alpha A^S + 2\alpha D^S - D^S \geq 0$

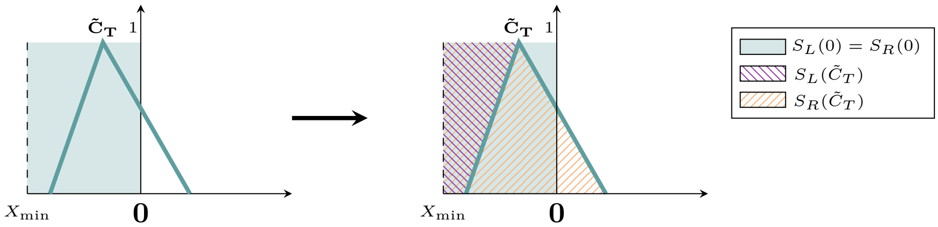
(b) When $(1-\alpha)(A^P - D^S) + B^P - B^S + \alpha(D^P - A^S) \geq 0$ with $A^S - 2\alpha A^S + 2\alpha D^S - D^S < 0$

Considering the case (a), we isolate α in the inequality, i.e.:

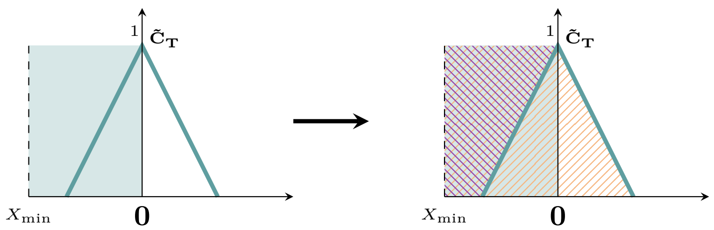
$$\begin{aligned} & A^S - 2\alpha A^S + 2\alpha D^S - D^S \geq 0 \qquad \text{for some } \alpha \in [0,1] \\ & \Rightarrow (A^S - D^S) - 2\alpha(A^S - D^S) \geq 0 \\ & \Rightarrow (1-2\alpha)(A^S - D^S) \geq 0, \qquad \text{with } \ A^S - D^S \leq 0 \\ & \Rightarrow 1 - 2\alpha \leq 0 \\ & \Rightarrow \alpha \geq \frac{1}{2} \end{aligned}$$

Thus, the term $(1-\alpha)(A^P - D^S) + B^P - B^S + \alpha(D^P - A^S)$ is non-negative for $\forall\alpha \in \left[\frac{1}{2}, 1\right]$. *Therefore, as long as we use the ranking criterium proposed by [Yu and l. Q. Dat 2014], if a sub-mixed cycle in $\tilde{G}'$ has a total cost defined as in equation 5.20, the condition 5.25 is true, and therefore, the sub-mixed cycle is non-negative for every α in the interval $\left[\frac{1}{2}, 1\right]$.*

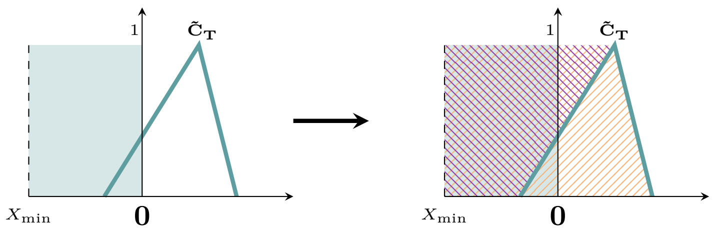
The results stated above are shown in figures 5-13(a) to 5-13(d). These figures graphically show the total cost of the sub-mixed cycle in cases where it is N_1-zero, N_2-zero, N_3-zero, and a positive triangular fuzzy numbers, respectively. We compare the total cost of the sub-mixed cycle with the real number 0 and illustrate the Left and Right Integrals for both numbers in all cases. $S_L(0)$ and $S_R(0)$ are both equal to the fixed $X_{\min} = \inf\{\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}, 0\} < 0$. $S_L[\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}]$ and $S_R[\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}]$ are the areas filled in violet and orange, respectively. For the cases 5-13(a) to 5-13(c), if $\alpha < \frac{1}{2}$, when computing S_T^α for both numbers, the higher ponderation lies on S_L. Therefore, we do the comparison giving more priority to numbers lower than the value with the highest membership degree (entirely negative values for the cases 5-13(a) and 5-13(b)). Note that $S_L(0)$ can be greater than $S_L\left(\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}\right)$, and consequently, there is no guarantee that the Total Integral for the total cost of the sub-mixed cycle has a value greater than 0 for $\forall\alpha \in \left[0, \frac{1}{2}\right)$. Therefore, we would need to find the values for $\alpha \in [0, \frac{1}{2})$ for which $S_T^\alpha\left[\tilde{C}\begin{pmatrix}\text{sub-mixed}\\ \text{cycle}\end{pmatrix}\right] > S_T^\alpha[(0,0,0)]$. On the other hand, when $\alpha \geq \frac{1}{2}$, the higher ponderation is on the right integral, which considers the numbers to the left of the value with the highest membership degree and those to the right of this value. Therefore, for these values of α, the total cost of the sub-mixed cycle, represented as a triangular fuzzy number, is always greater than $(0, 0, 0)$.



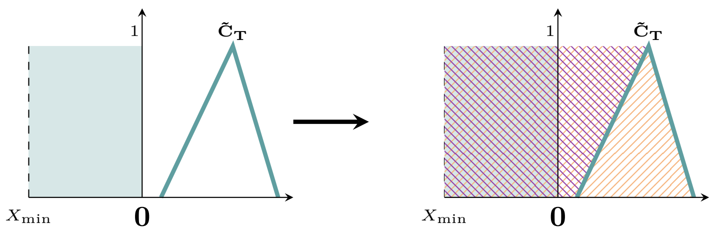
(a) The total cost of the sub-mixed cycle as N_1-zero triangular fuzzy number.



(b) The total cost of the sub-mixed cycle as N_2-zero triangular fuzzy number.



(c) The total cost of the sub-mixed cycle as N_3-zero triangular fuzzy number.



(d) The total cost of the sub-mixed cycle as positive triangular fuzzy number.

Figure 5-13: *Analysis of the non-negativity of the total cost of the sub-mixed cycle by its comparison with the real number* 0*.*

Consequently, the mixed cycle containing a sub-mixed cycle with a shape as 5-12 is non-negative for $\alpha \in \left[\frac{1}{2}, 1\right]$. We summarize the previous results in proposition 5.4.1.

Proposition 5.4.1.

Given the type V fuzzy mixed graph $\tilde{G}' = (V, A', \mathfrak{C}')$ defined in section 5.3.5 under the following conditions:

- *The costs in $\mathfrak{C}'$ are defined as triangular fuzzy numbers.*
- *The ranking method proposed by [Yu and l. Q. Dat 2014] is applied to compare costs.*
- *Any mixed cycle in $\tilde{G}'$ is composed of sub-mixed cycles with a structure as shown in figure 5-12 and positive cycles.*

The following properties are satisfied:

(i) A sub-mixed cycle has a non-negative total cost if the parameter α, used for the ranking method, takes any value in the interval $[\frac{1}{2}, 1]$.

(ii) Any mixed cycle is non-negative if every sub-mixed cycle included has a non-negative total cost.

With proposition 5.4.1, we give a condition that guarantees the convergence of an algorithm to search the shortest path in $\tilde{G}'$. In particular, we modify the FDA proposed in Chapter 4 and incorporate it into our algorithmic proposal discussed in section 5.4.2.

5.4.2 Algorithm to find the Fuzzy Shortest Pair of Edge-Disjoint Paths

Modified Fuzzy Dijkstra Algorithm

The Modified Fuzzy Dijkstra Algorithm (MFDA) is a slight variation of the FDA that finds the path P_{aux} in $\tilde{G}'$. The term *modified* comes from the fact that it converges to the feasible solution when applied in the graph $\tilde{G}'$ containing arcs with negative triangular fuzzy costs.

We use the same terms and notations proposed in section 4.4.2. We show the pseudo-code of the MFDA in algorithm 5.

Algorithm 5 Mod-FuzzyDijkstra($\tilde{G}', r, t, \alpha$)

1: Initialize-single-source($\tilde{G}', r$) ▷ Initialization
2: $H \leftarrow V$
3: **while** $H \neq \emptyset$ **do**
4:     $u \leftarrow z | \tilde{d}(z) = \min_{\forall x \in H} \left\{ \tilde{d}(x) \right\}$
5:     Update $H \leftarrow H - \{u\}$
6:     **if** $u = t$ **then -end while-**
7:     **end if**
8:     **if** $\Gamma(u) \neq \emptyset$ **then**
9:         **for** each vertex $v \in \Gamma(u)$ **do**
10:             **if** $v \neq w(u)$ **then**
11:                 RelaxationMFDA($u, v, \tilde{C}_{(u,v)}, \alpha$) ▷ Relaxation 2 of v
12:             **end if**
13:         **end for**
14:     **end if**
15: **end while**

The initialization algorithm is the same as algorithm 2. In the case of the Relaxation, we make a small variation to algorithm 3, as is shown in algorithm 6

Algorithm 6 Relaxation of v in MFDA

1: **function** RELAXATIONMFDA($u, v, \tilde{C}_{(u,v)}, \alpha$)
2:     $\tilde{d}^{\text{new}}(v) := \tilde{d}(u) \oplus \tilde{C}_{(u,v)}$
3:     **if** $S_T^\alpha(\tilde{d}(v)) > S_T^\alpha(\tilde{d}^{\text{new}}(v))$ **or** $\left[S_T^\alpha(\tilde{d}(v)) = S_T^\alpha(\tilde{d}^{\text{new}}(v)) \right.$ **and** $\left. \text{Me}(\tilde{d}(v)) > \text{Me}(\tilde{d}^{\text{new}}(v)) \right]$ **then**
4:         $\tilde{d}(v) \leftarrow \tilde{d}^{\text{new}}(v)$
5:         $w(v) \leftarrow u$
6:         Update $H \leftarrow H \cup \{v\}$
7:     **end if**
8: **end function**

The MFDA guarantees the reentry to H of vertices previously labeled due to $\tilde{G}'$ containing arcs with negative fuzzy costs. Each vertex $v \in \Gamma(u)$, whose label is updated, is added again to the set H if this does not contain it (step 6 in Algorithm 6), meaning that its label can be updated again. Consequently, for each vertex u, all its neighboring vertices are analyzed, regardless of whether they belong to set H or not (step 9 in Algorithm 5). These modifications are redundant for non-negative type V fuzzy graphs but are essential for graphs with the structure of $\tilde{G}'$ that we can find when searching the shortest pair of edge-disjoint paths.

Algorithm for the search of the fuzzy shortest pair of edge-disjoint paths (FSPPA)

To search for the shortest pair of edge-disjoint paths in a type V fuzzy graph, we propose the FSPPA, whose pseudo-code is shown in Algorithm 7. In the algorithm, we look first for the shortest path of $\tilde{G}$, denoted as S, by the application of the FDA (Algorithm 4). Then, we create a new graph $\tilde{G}'$ using the transformations explained in section 5.3.5 (Algorithm 8). On the new graph, $\tilde{G}'$, we apply the MFDA (Algorithm 5) to find the shortest path P_{aux}. Once the paths S and P_{aux} have been obtained, their mutual edges, called breaks, are eliminated to create a pair of paths, disjoint in edges, whose total sum of costs is the minimum. Algorithm 9 calculates the total cost of the shortest pair of edge-disjoint paths. According to the uniqueness of path S and the existence of breaks, the pair of paths found by the FSPPA has the structure of pairs described in section 5.3.3.

Algorithm 7 FSPPA($\tilde{G}, r, t$)

$\alpha = \alpha_0$ ▷ Setting α according to proposition 5.4.1
$(S, \tilde{C}_S)$ = Fuzzy-Dijkstra($\tilde{G}, r, t, \alpha$) ▷ To apply FDA to $\tilde{G}$ to obtain S
$(\tilde{G}')$ = Build-ModifiedGraph($\tilde{G}, r, t, S$) ▷ Creating graph $\tilde{G}'$
$(P_{\text{aux}}, \tilde{C}_{P_{\text{aux}}})$ = Mod-FuzzyDijkstra($\tilde{G}', r, t, \alpha$) ▷ To apply MFDA to $\tilde{G}'$ to obtain P_{aux}
$M := \{e_i | \, e_i \in S \cap P_{\text{aux}}\}$ ▷ M: set of breaks between S and P_{aux}
if $M = \emptyset$ **then**
  $\gamma_1 := S$ and $\gamma_2 := P_{\text{aux}}$ ▷ If $M = \emptyset$ the pair has the configuration of figure 5-4(b)
else
  $(\gamma_1, \gamma_2) := (S \cup P_{\text{aux}}) - \{M\}$ ▷ If $M \neq \emptyset$ the pair has the configuration of figure 5-4(a)
end if
$(\tilde{C}_{(\gamma_1,\gamma_2)})$ = Calc-TotalCost($\tilde{G}, M, S, P_{\text{aux}}, \tilde{C}_S, \tilde{C}_{P_{\text{aux}}}$) ▷ To calculate the total cost of (γ_1, γ_2)

Algorithm 8 Build-ModifiedGraph($\tilde{G}, r, t, S$)

▷ Creating graph $\tilde{G}'$
for each $e_i = (u, v)_i \in S$ **do**
  Update $e_i \leftarrow \overrightarrow{(v, u)}_i$ ▷ Each edge of S is replaced by an unique arc directed to r
  Update $\tilde{C}_{(v,u)_i} = -\tilde{C}_{(u,v)_i}$ ▷ The cost of each arc is defined as the complementary of the original edge cost
end for

Algorithm 9 Calc-TotalCost($\tilde{G}, M, S, P_{\text{aux}}, \tilde{C}_S, \tilde{C}_{P_{\text{aux}}}$)

▷ Computing the total cost of pair (γ_1, γ_2)

1: $\tilde{C}_{(S,P_{\text{aux}})} = \tilde{C}_S + \tilde{C}_{P_{\text{aux}}}$

2: $A_M = \text{sum}(a_{e_i} | \, e_i \in M)$, $C_M = \text{sum}(c_{e_i} | \, e_i \in M)$

3: $\tilde{C}_{(\gamma_1,\gamma_2)} = \tilde{C}_{(S,P_{\text{aux}})} - (A_M - C_M, 0, C_M - A_M)$

5.5 Experimentation and Results

We apply the FSPPA in a network with a high traffic load. We intend to illustrate the effectiveness of the algorithm when we want to find the shortest pair of edge-disjoint paths where both paths can be used simultaneously to distribute the information delivery and contribute to a decrease in network saturation. [2] [3] [4]

5.5.1 Experiment 1. FSPPA validation

Experiment 1 applies the FSPPA for all the possible pairs of server and client nodes. We focus on seeing that our algorithm always finds the shortest pair of edge-disjoint paths with this experiment. It is an adaptation of the experimentation we perform in Chapter 4, in which we have ensured that all nodes have at least two alternative paths. In this way, we can apply our algorithm when we randomly search for communication between any two nodes.

We can not perform this experiment in large networks due to the high computational cost of carrying out the exhaustive search. However, this test is sufficient to show that our algorithm finds the shortest pair of edge-disjoint paths. We use a network of 12 US cities whose graphical representation is shown in figure 5-14, considering the costs of its links as triangular fuzzy numbers. On this network, we simulate traffic in which the system presents very high saturation conditions. The costs on links vary with time and depend on the traffic in each time interval (the reader can find a more detailed explanation in Chapter 4). The variable that we measure as the link's cost is the *modified fuzzy normalized used bandwidth*, already defined in Chapter 4. We perform a single repetition due to its high computational cost.

For each node, we apply an algorithm to find the K-shortest paths between this and any other node. If K is large enough, we ensure we have all possible paths between both nodes. The costs of the paths found are calculated and sorted in ascending order concerning their costs. Then, we choose the first path of the list (shortest path) and continue searching through the list until finding the first path that is disjoint in edges with the one chosen above. The total cost of the pair of

[2] The simulator can be downloaded at https://omnetpp.org/download/

[3] The source code of the model used can be downloaded in https://github.com/aarizaq/flowsimulator

[4] The configuration for the running of the experiments can be found in https://github.com/aarizaq/configurationFuzzy

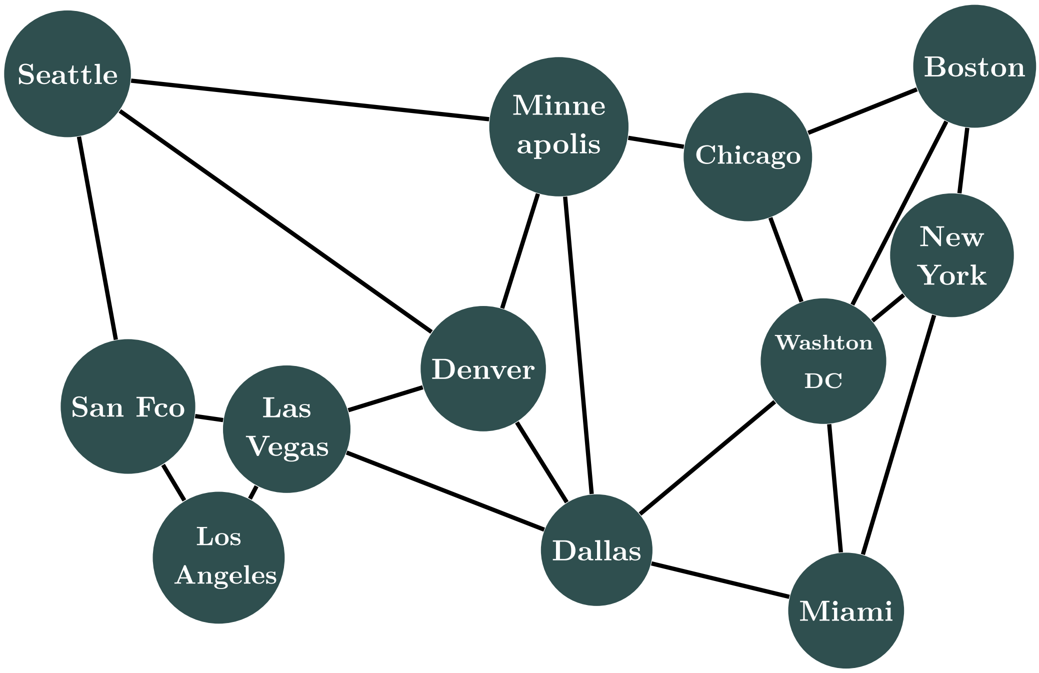


Figure 5-14: *Network of 12 US cities*

paths found is calculated and serves as a reference.

On the other hand, we apply the FSPPA algorithm to find the shortest pair of edge-disjoint paths. Besides, we perform the exhaustive search described above to find a second pair of edge-disjoint paths. Then we compare the total cost of both pairs of paths. If both total costs match, the FSPPA has found an optimal solution. Both pairs of paths do not have to contain the same paths; our interest lies in finding the same total cost.

In conclusion, we verified that our algorithm finds the shortest pair of edge-disjoint paths in all cases.

5.5.2 Experiment 2. A search of the shortest pair of edge-disjoint paths using fuzzy costs in a high-performance network with priority traffic

Once we show that our algorithm finds the shortest pair of edge-disjoint paths in all cases, experiment 2 aims at checking whether the use of edge-disjoint paths can help guarantee the quality of a given priority traffic in conditions of network overload.

We understand as *Priority traffic* the one generated by a set of nodes to which, for security reasons, a certain level of communication privilege is granted.

We emphasize that we do not refer to the problem whose solution is to find, at first, the shortest path, then eliminate the edges of it, and, eventually, to find the shortest path on the modified network (backup path) again. Solving this problem provides the possibility of having an alternative replacement path if the shortest path fails (the backup path would be the second shortest path and would be operational once the failure is detected). Thus, the only guarantee when solving this problem is to ensure that the information is sent through the shortest path. If this path suffers a failure, the second shortest path is the replacement. Conversely, our goal is to find the shortest pair of edge-disjoint paths, that is, the pair of paths with minimum total cost. Thus, it is clear that we are not merely looking for an alternative redundant path, but a pair of paths, disjoint in edges, to use them simultaneously to send the fractionated information through both paths. This strategy improves the communication security of priority sources since it is more challenging to capture a complete ordered message by external elements. Also, it increases the transmission quality since it decreases the bandwidth necessary in each connection of the edge-disjoint paths, reducing the probability of losses in conditions close to saturation. Also, we could create redundant traffic, that is, always sending the same message through two paths at once. This action increases the network's traffic, but the information nevertheless arrives by the other path when one of the paths fails. To summarize, the difference between finding the pair of edge-disjoint paths with the minimum total cost and finding the shortest and backup paths consists of when using both paths simultaneously, we reduce the probability of information losses and increase the security (or privacy) of communications.

We use a 57-nodes network inspired by the NTT backbone network [Varga 2001]. We adjust the original network so that at least two paths can access each node.

The simulation is flow-oriented; we simulate only two events: the start and end of a burst. If we had to simulate sending the information burst like this, we could not simulate all the shipments in time using small information packets. Therefore, a single call is established to send complete information, and it is not finished until the operation ends. In other words, we send the complete information as if it were a block, so we have to observe the beginning and end of the block or burst. Also, the storage is not simulated by queues at the nodes. If a node does not have sufficient capacity to transmit the burst, the data is lost until there is free space or the burst ends, in which case it will be entirely lost. Therefore, we are simulating a system without delays, except propagation, which could be seen as an Optical Network without delay elements. Consequently, the variable that will determine the quality of the network will be *the byte delivery ratio*. Thus, we will know whether the data sent is lost or not.

The simulation time has been 10^4 seconds, and all links have the same capacity (1Gb/s).

We generate the traffic through calls with a connection. Once a call is established, the chosen path does not change during the entire duration of the call. These calls do not make reservations of resources, so the establishment of calls will never be rejected, but there may be a loss of data. This restriction facilitates the visualization of the data loss due to the saturation of links.

As in experiment 1, we use as the indicator variable of the cost of a link the *modified fuzzy normalized used bandwidth*, which, to facilitate the reading, we redefine in equation 5.28 for each link $e = (i, j)$,

$$[\widetilde{bw}]_{ij}^{A,B} = A * B^{-[bw]'_{ij}} [\widetilde{bw}]'_{ij} \tag{5.28}$$

where $[bw]'_{ij}$ and $[\widetilde{bw}]'_{ij}$ are defined in the expressions 4.6 and 4.10, respectively. $[\widetilde{bw}]_{ij}^{A,B}$ is based on the cost function defined in strategy 8 explained in Chapter 4. We consider this variable a triangular fuzzy number and weigh it with the values A and B to achieve a displacement to the right and a widening of the fuzzy number. As in the experiments performed in Chapter 4, we use the values $A = 10$ and $B = 20$. We update the links every 300 secs. This interval is realistic in a 57-nodes network and justifies using fuzzy numbers to measure the link costs due to the high uncertainty in the variable's value.

Traffic is of the ON/OFF type with the following parameters:

- We perform ten replications per experiment with different seeds each. In each simulation, the seed is different, making the order in which the nodes are activated and different the flow of information. On the other hand, the flow of information sent between nodes is made according to probability distributions.

- Each node can have two communication sources (or independent traffic generators), F1 and F2. Both types of sources differ in the priority of sending specific messages.

  Source F1: corresponds to the regular sending of the information. It always sends the information by a single path, precisely, by the shortest path between the source and destination nodes.

  Source F2: It is a priority message source; that is, it sends messages that we treat with additional security (for example, when a bank or state agency has a serie of priority messages, these are sent in different conditions concerning the rest of the information transmitted). In nodes with source F2, we can make the information travel by two alternative paths. In particular, these are the paths forming the shortest pair of edge-disjoint paths found by the FSPPA. Note that a node with source F2 can apply the FSPPA and function as a node with source F1. Also, the destination of a node with source F2 is always another node with source F2. This procedure facilitates the delimitation of the possible number of source-destination pairs in traffic F2.

Description of the simulation

We measure the transmission quality in the network through the BDR, defined as the ratio between delivered and sent bytes.

We performed the experimentation in a network with conditions very close to saturation. The source F1 is implemented in all nodes, while the source F2 is only active in a limited number of randomly selected nodes in each experiment. We do the tests for sets of 5, 10, 15, 20, 25, and 30 nodes with source F2.

For each set of nodes with source F2, we perform ten simulations. We generate trac for these nodes where the information is sent either by the pair of paths that the FSPPA finds or the shortest path that the FDA finds. In each case, we compute the MBDR, defined as the Mean of the Bytes Delivery Ratio, for the nodes with source F2 and the nodes with source F1. We compare the performance of both generated traffics.

Analysis of the results

Figure 5-15 shows the results of experiment 2. On the X-axis, the numbers represent the size of the set of F2 source nodes used in the tests, and the Y-axis represents the values of MBDR. Each line shows the MBDR of the network for each repetition of the traffic simulations under different conditions.

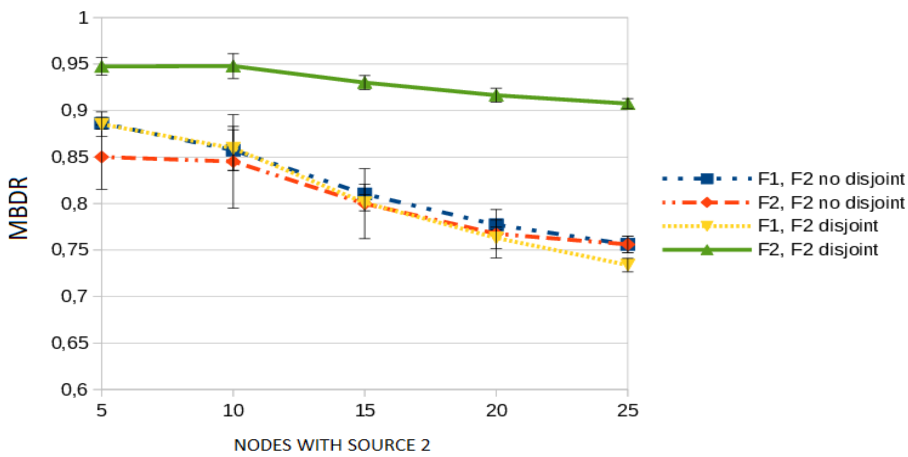


Figure 5-15: *Simulation results in Experiment 2*

Blue Line: The blue line shows the MBDR corresponding to the traffic by source F1 when the nodes with source F2 do not apply FSPPA. As the number of nodes with source F2 increases, network traffic also increases, and as a consequence, the MBDR decreases. This line represents the effect of the system's saturation and its inability to deal with it when the FSPPA is not applied.

Yellow Line: The yellow line shows the MBDR corresponding to the traffic by source F1 when nodes with source F2 apply FSPPA. As nodes with source F2 increase, these nodes generate more traffic, which affects the nodes with source F1. Thus, coherently, this traffic is the worst behaved with the rise of nodes with source F2.

Orange Line: The orange line shows the MBDR corresponding to the traffic by nodes with source F2 when these do not apply the FSPPA.

Green Line: The green line shows the MBDR corresponding to the traffic by nodes with source F2 when applying the FSPPA. Notice that, although the number of nodes with source F2 increases to almost 50% of the total nodes, the MBDR remains high (greater than 0.9).

Comparison between lines blue and yellow: We compare the traffic behavior by source F1 according to the application of the FSPPA by nodes with source F2. When there are few nodes with source F2, both lines are very similar. However, as the number of nodes with source F2 increases, the yellow line separates below the blue one. This separation means that the nodes with source F1 are harmed even more due to the rise of the MBDR in traffic (they reach the impermissible value of 0.75 approximately). Logically, the MBDR of nodes with source F1 when the nodes with source F2 apply FSPPA is lower than when the nodes with source F2 do not apply the FSPPA. This difference increases as the number of nodes with source F2 applying FSPPA increases.

Comparison between lines green and orange: We compare the traffic behavior by source F2, whether when the nodes with source F2 apply FSPPA or not. We recall that nodes with source F2 have additional traffic that they do not use when not applying FSPPA. In such a case, the traffic capacity of these nodes is saturated. Due to this extra traffic, we observe the high difference between the green and orange lines. The MBDR corresponding to the green line remains between 0.9 and 0.95, both very high values. However, when these nodes do not apply FSPPA (orange line), the MBDR starts from 0.87 (for a small number of nodes with source F2) and reaches the impermissible value of approximately 0.72 (for a large number of nodes with source F2).

Comparison between lines blue and orange: We compare the behavior of traffics F1 and F2 when the FSPPA is not applied. For a small number of both sources, the difference between their MBDR is significant (0.85 for traffic F2 and 0.9 for traffic F1). Still, we observe that their confidence intervals are very large, indicating high variability in this case. This aspect is because of the extra unused traffic that nodes with source F2 when not applying FSPPA. On the other hand, notice that as the number of nodes with source F2 increases, the MBDR of traffics in F1 and F2 sources are closer. This behavior is logical because when nodes with source F2 do not apply the FSPPA, the more nodes with source F2 are, the more regular the network traffic is. Thus, the nodes with both sources use the same kind of traffic.

We conclude that having a strategy where the network has a small number of privileged nodes with source F2 where a pair of paths sends the information is very interesting and helpful. The algorithm proposed by us provides a solution to make this strategy works.

Currently, in many communication networks, the “privileged” nodes constitute a separate system where only they are part of, which means a higher cost of resources. Therefore, another advantage of our strategy is that it is unnecessary to create a separate network to achieve high values of the MBDR.

5.6 Conclusions

In this chapter, we faced the problem of finding the shortest pair of edge-disjoint paths in a communication network under uncertain conditions. We designed an algorithm to solve the problem from the graph theory vision, and we described the uncertainty in the network using elements of fuzzy logic. In particular, we associate the network to a type V fuzzy graph and propose an algorithm (FSPPA) that finds the shortest pair of edge-disjoint paths in the graph.

We described the applications of the FSPPA to guarantee security in a communication network. Thus, to illustrate the algorithm’s effectiveness, we apply the FSPPA to a well-known network with a high traffic load, using a new fuzzy cost function, also proposed by us. The results showed the competence of our algorithm in communication networks with specific nodes having a “privileged” traffic (nodes with source F2). In other words, our strategy makes it unnecessary to achieve high values of the MBDR, to create a separate network with source F2 nodes.

As a general conclusion, it is true that in current networks, overload conditions are not common since operators over-size the network with a high safety factor. On the other hand, the classical methods applied under these conditions satisfactorily solve the presented problems. Therefore, our goal is not to compare the application of fuzzy techniques with the current ones but rather to study their feasibility. In any case, we have seen that these turn out to be competitive since, at least in all cases, we found a fuzzy method that works equal to or better than the classic method.

Network management systems are very conservative in terms of the methods and algorithms used, and in most cases, problematic situations are well covered by duplication of resources. However, exploring alternative techniques, such as those analyzed in this work, can be interesting from a theoretical and practical point of view in future scenarios.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, we approach the study of some tasks related to communication networks' management through graph theory and fuzzy logic. Based on these techniques, we designed and applied different methodologies to search for the optimal path between the server and client nodes in a communication network. As a result, three main lines of work have been developed:

(i) The study of server node selection in a P2P network using a fuzzy inference-based system.

(ii) The search of the shortest path in a communication network using different fuzzy cost functions in the links. This hypothesis is quite viable because of how updating the network parameters in each time interval.

(iii) The search of the shortest pair of links-disjoint paths using fuzzy costs in a high-performance communication network in an overload situation.

Next, we will expose a summary of the conclusions extracted throughout our investigative work:

Chapter 3

1. We have implemented a fuzzy inference algorithm to select the server node in a P2P network. The input variables in the fuzzy system are the number of hops and the ETX metric in each server-node path. On the other hand, the output variable is the goodness index of the path. The rule base is composed of fuzzy if-then rules with antecedent compound by the connector AND. We apply the MFI as the implication method and the maximum value resultant of composing the output functions of each rule as a combination process. Lastly, we use the CoA as the defuzzification method.

2. We analyzed two different transmission scenarios: a network without obstacles and a network with obstacles between nodes. In addition, we compared

the fuzzy inference system with two other strategies: the Random Selection of the server node and the *Min-Hop*.

3. In a network without obstacles, the Random Selection strategy is the least efficient concerning the required transmission time and the network-level traffic load. In a network with obstacles, the fuzzy inference system produces the best results concerning the download time for a node. Also, in this scenario, the Min-Hop strategy is the least efficient because it does not consider the real state of the network but only the number of hops. There are no critical dierences between Min-Hop and our approach in a regular network since the Min-Hop is very ecient in this kind of network, so the impact of fuzzy logic cannot be shown entirely in this experiment.

Chapter 4

4. To search for the shortest path between two nodes, we analyzed and compared the efficiency of different strategies with the crisp and their equivalent fuzzy cost functions to improve the delivery rate. We face this problem by modeling the communications network as a type V fuzzy graph. In the graph, we describe both the nodes and the links with precision, but we modeled each link's cost as a triangular fuzzy number.

5. We proposed an FDA that finds the shortest path between two vertices in a type V fuzzy graph where the costs in the edges are triangular fuzzy numbers. To compare the fuzzy costs, we applied the ranking method proposed in [Yu and l. Q. Dat 2014].

6. We were interested in comparing the network performance efficiency based on fuzzy logic versus based on crisp values. Thus, we implemented the most commonly used cost functions and strategies to manage real networks based on crisp values (e.g., the instantaneous or mean used bandwidth and residual bandwidth as cost functions and SW or WS strategies). We confronted the crisp strategies with similar ones based on our definition of fuzzy costs. We performed an experimental study using the NTT backbone network as a reference, where we implemented each strategy.

7. As an interesting contribution, we proposed a new strategy (strategy 8), which does not correspond with any classic one. In this fuzzy strategy, we defined the Modified Fuzzy Normalized Used Bandwidth. This cost function modifies the fuzzy normalized used bandwidth such that the smaller the used bandwidth at the (n-1)-th interval (where the measurements have been obtained), the higher uncertainty in the used bandwidth value considered in the depth of the values in the (n+1)-th interval.

8. Fuzzy strategy 3 surpasses in a slightly but statistically significant way their analogous crisp strategies 1 and 2. On the other hand, the fuzzy strategies 5 and 7 do not present statistically substantial differences with

their equivalent crisp strategies 4 and 6, respectively. Specifically, our new fuzzy strategy (strategy 8) provides the best results, significantly exceeding the performance of the rest of the strategies, achieving a GMDR close to 1.

Chapter 5

9. We described the possible applications of the shortest pair of edge-disjoint paths: the capacity of the network to find an alternative path when part of the system is affected by an external cause; the redistribution of traffic when the network is under saturation conditions, where the information is divided into two segments and sent at the same time by two paths with no common links; and the communication security where the data travels by two paths in parallel, so it is more challenging to capture a complete message by external elements.

10. As in chapter 4, we associated the network to a type V fuzzy graph whose vertices and edges correspond to the nodes and links of the network, respectively. We faced uncertainty in the network's operating system by considering the cost of the edges as triangular fuzzy numbers.

11. We proposed an algorithm that finds the shortest pair of edge-disjoint paths in the graph (FSPPA). The FSPPA uses a sub-algorithm, an adaptation of the FDA, called the Modified Fuzzy Dijkstra Algorithm (MFDA). The MFDA can find the shortest path in a mixed type V fuzzy graph containing edges and some arcs whose costs are negative triangular fuzzy numbers.

12. Intending to illustrate the effectiveness of the algorithm, we applied the FSPPA to a network with a high traffic load, using the new fuzzy cost function defined in strategy 8 (chapter 4) as the indicator variable. We simulated traffic with two types of communication sources: source F1 (regular sending of information) that always sends the data by the shortest path between the source and destination nodes, and source F2 (priority sending of information) which sends the data throughout the paths of the shortest pair of edge-disjoint paths. We measured the transmission quality in the network through the BDR (ratio between delivered and sent bytes). Our algorithm provided a very efficient solution in a scenario where the system has a relatively small number of nodes with priority communication sources.

13. Currently, in many communication networks, the privileged nodes form a separate network where only they are part of, meaning a higher cost of resources. Therefore, another advantage of our strategy is that it is unnecessary to create a separate network to achieve high BDR values.

Like the one we implemented in chapter 3, Fuzzy Inference Systems are quite robust concerning possible variations in the definition of the membership functions that define the fuzzy values of their variables. However, it could be interesting to perform a systematic study of different models about the parameters of these functions and their form, and different network configurations. In this way, it would be possible to find more efficient models or, at least, study their applicability in the field of teaching.

It is interesting to explore the applicability of the algorithms proposed in chapter 4 to maximize the throughput of a communication network or to search for optimal paths in other transportation networks. In particular, it can be very interesting in the autonomous vehicle routing problem. In this context, the network behaves very similarly to the communication networks that we studied because the traffic conditions are updated periodically, generating uncertainty in the value of each variable between two updates. In the case of vehicle transport networks, this uncertainty is much more significant since the factors that can generate a change in traffic conditions occur more chaotically.

Chapter 5 addressed the problem of finding the minimum pair of edge-disjoint paths in a V-type fuzziness graph from a theoretical point of view and its application to a communication network with two types of traffic from a practical point of view. From a theoretical point of view, we consider that an interesting future line of work would be to address the problem of finding the minimum pair of node-disjoint paths.

In chapters 4 and 5, we have focused on using the Dijkstra algorithm. However, it would be interesting to study the behavior of other complete search algorithms, such as A* ([Peter E. et al. 1968]), which could provide a faster search of the solution.

In general, network management systems are very conservative regarding the methods and algorithms used, and in most cases, problematic situations are well covered with duplication of resources. However, exploring other alternative technical techniques, such as the analytical ones in this work, can be interesting, not only from a theoretical and mathematical perspective but also practical in future scenarios.

APPENDIX A

SUBPATH BETWEEN TWO VERTICES IN THE SHORTEST PATH IN A TYPE V FUZZY GRAPH

Theorem A.0.1.
Let $\tilde{G} = (V, E, \tilde{W})$ be a type V fuzzy graph with the weights defined as positive triangular fuzzy numbers. Let $S =< v_1, v_2, \ldots, v_k >$ be the unique shortest path between the vertices v_1 and v_k and, for every i and j such that $1 \leq i \leq j \leq k$, let $S_{ij} =< v_i, v_{i+1}, \ldots, v_j >$ be the subpath of S between vertices v_i and v_j. Then S_{ij} is the shortest path from v_i to v_j of $\tilde{G}$.

Proof. The length of the shortest path S is defined as

$$\begin{aligned} L_S &= L_{S_{1i}} + L_{S_{ij}} + L_{S_{jk}} \\ &= (a_{S_{1i}}, b_{S_{1i}}, d_{S_{1i}}) + (a_{S_{ij}}, b_{S_{ij}}, d_{S_{ij}}) + (a_{S_{jk}}, b_{S_{jk}}, d_{S_{jk}}) \\ &= \Big(a_{S_{1i}} + a_{S_{ij}} + a_{S_{jk}}, b_{S_{1i}} + b_{S_{ij}} + b_{S_{jk}}, d_{S_{1i}} + d_{S_{ij}} + d_{S_{jk}}\Big) \end{aligned}$$

Lets assume that S_{ij} is not the shortest path between v_i and v_j.
This means that there exist an alternative path, S'_{ij}, which length $L_{S'_{ij}} = (a_{S'_{ij}}, b_{S'_{ij}}, d_{S'_{ij}})$ is less than the length of S_{ij}, i.e.:

$$\begin{aligned} &L_{S_{ij}} \succ L_{S'_{ij}} \\ &\Rightarrow (1-\alpha)a_{S_{ij}} + b_{S_{ij}} + \alpha d_{S_{ij}} > (1-\alpha)a_{S'_{ij}} + b_{S'_{ij}} + \alpha d_{S'_{ij}} \qquad \forall \alpha \in [0,1] \end{aligned}$$

If we add the positive expressions $(1-\alpha)(a_{S_{1i}} + a_{S_{jk}})$, $(b_{S_{1i}} + b_{S_{jk}})$ and $\alpha(d_{S_{1i}} + d_{S_{jk}})$ in the previous inequality, we obtain:

$$\begin{aligned} \Rightarrow\ & (1-\alpha)(a_{S_{1i}} + a_{S_{ij}} + a_{S_{jk}}) + b_{S_{1i}} + b_{S_{ij}} + b_{S_{jk}} + \alpha(d_{S_{1i}} + d_{S_{ij}} + d_{S_{jk}}) \\ & > (1-\alpha)(a_{S_{1i}} + a_{S'_{ij}} + a_{S_{jk}}) + b_{S_{1i}} + b_{S'_{ij}} + b_{S_{jk}} + \alpha(d_{S_{1i}} + d_{S'_{ij}} + d_{S_{jk}}) \end{aligned}$$

Hence, path $S' =< S_{1i}, S'_{ij}, S_{jk} >$ is a shortest path of $\tilde{G}$. But this is a contradiction with the hyphotesis of being S' the unique shortest path.
$\therefore$ S_{ij} is the shortest path between the vertices v_i and v_j. □

BIBLIOGRAPHY

[Ariza 2001] Ariza, A.: *Encaminamiento en redes con calidad de servicio*, Malaga University, PhD dissertation, 2001

[Ariza 2014] Ariza, A.: *inetmanet framework*. 2014

[Ariza 2015] Ariza, A.: *GitHub. Flow simulator to testing QoS wired protocols*. 2015. – URL https://github.com/aarizaq/flowsimulator

[Ariza et al. 2000] Ariza, A. ; Casiliari, E. ; Sandoval, F.: Strategies for updating link states in QoS routers, 2000, pp. 1749–1750

[Arom-oon and Keeratiwintakorn 2007] Arom-oon, U. ; Keeratiwintakorn, P.: Fuzzy path selection for routing in energy-limited mobile ad hoc networks. In: *Proceedings of IEEE ISCIT* (2007), 10, pp. 860–864

[Banerjee and Roy 2012] Banerjee, S. ; Roy, T. K.: Arithmetic Operations on Generalized Trapezoidal Fuzzy Number and its Applications. In: *Turkish Journal of Fuzzy Systems* 3 (2012), 03, Nr. 1, pp. 16–44. – URL http://www.tjfs-journal.org/tjfs_v3n1_pp016_044.pdf

[Bellman and Giertz 1973] Bellman, R. ; Giertz, M.: On the analytic formalism of the theory of fuzzy sets. In: *Information Sciences* 5 (1973)

[Bhandari 1999] Bhandari, R.: *Survivable networks. Algorithms for Diverse Routing*. Kluwer Academic Publishers, 1999. – ISBN 0792383818

[Blue et al. 2002] Blue, M. ; Bush, B. ; Puckett, J.: Unified approach to fuzzy graph problems. In: *Fuzzy Sets and Systems* (2002), 08, Nr. 125, pp. 355–368. – URL http://citeseerx.ist.psu.edu/viewdoc/download?

[Boutaba et al. 2002] Boutaba, R. ; Szeto, W. ; Iraqi, Y.: DORA: Efficient Routing for MPLS Traffic Engineering. In: *Journal of Network and Systems Management* 10 (2002), 09, Nr. 3

[Buford et al. 2008] Buford, J. F. ; Yu, H. ; Lua, E. K.: *P2P Networking and Applications*. Elsevier Inc, 2008. – URL http://www.sciencedirect.com/science/book/9780123742148. – ISBN 9780123742148

[Chou 2003] Chou, CH-CH.: The Canonical Representation of Multiplication Operation on Triangular Fuzzy Numbers. In: *Pergamon Computers and Mathematics with Applications* 45 (2003), pp. 1601–1610. – URL www.elsevier.nl/locate/camwa

[Ch.Xin and He 2005] Ch.Xin ; He, F.: Cross-layer Path Computation for Dynamic Traffic Grooming: In Mesh WDM Optical Networks, ACM, 2005, pp. 9–13. – URL http://doi.acm.org/10.1145/1167253.1167257. – ISBN 1-59593-059-0

[Clausen and Jacquet 2003] Clausen, T. ; Jacquet, P.: *RFC 3626 : Optimized link state routing protocol (olsr)*. 2003

[Couto 2004] Couto, D. S. J. D.: *High-Throughput Routing for Multi-Hop Wireless Networks*, Massachusetts Institute of Technology, Dissertation, 2004

[D. Zhu 2002] D. Zhu, D. R. C.: Feedback based routing. In: *ACM SIGCOMM Computer Communication Review* (2002)

[Deng et al. 2012] Deng, Y. ; Chen, Y. ; Zhang, Y. ; Mahadevan, S.: Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. In: *Applied Soft Computing* 12 (2012), pp. 1231–1237

[Dubois and Prade 1979] Dubois, D. ; Prade, H.: Fuzzy real algebra: Some results. In: *Fuzzy Sets Systems* (1979), Nr. 2, pp. 327–349

[Dubois and Prade 1980] Dubois, D. ; Prade, H.: *Fuzzy Sets and Systems: Theory and Applications*. Bd. 144. Academic Press, Inc, 1980. – ISBN 0-012-222750-6

[Gibbons 1985] Gibbons, Alan: *Algorithmic Graph Theory*. Press Syndicate of the University of Cambridge, 1985. – ISBN 0 521 24659 8

[Goguen 1967] Goguen, J. A.: L-Fuzzy Sets. In: *Journal of Mathematical Analysis and Applications* 18 (1967), 04, pp. 145–174. – URL http://dx.doi.org/10.1016/0022-247X(67)90189-8

[Gottschau et al. 2018] Gottschau, M. ; Kaiser, M. ; Waldmann, C.: The Undirected Two Disjoint Shortest Paths Problem. (2018), 09. ISBN arXiv:1809.03820 [math.CO]

[Guerin et al. 1997] Guerin, R. ; Orda, A. ; Williams, D.: QoS Routing Mechanism and OSPF Extensions, 1997

[He and Rexford 2008] He, J. ; Rexford, J.: Toward internet-wide multipath routing. In: *IEEE Network: The Magazine of Global Internet working* 22 (2008), 03, pp. 16–21

[IEEE 2012] IEEE: Ieee standard for information technology-telecommunications and information exchange between systems local and metropolitan area networks. In: *specific requirements part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications* ieee std 802.11-2012 (2012), 11

[Kang 2011] Kang, C.: Survey of search and optimization of P2P networks. In: *Peer-to-Peer Networking and Applications* 4 (2011), Nr. 3, pp. 211–218. – URL http://dblp.uni-trier.de/db/journals/ppna/ppna4.html#Kang11

[Klir and Yuan 1995] Klir, George J. ; Yuan, Bo ; Guerrieri, Patti (Hrsg.): *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice Hall PTR, 1995. – ISBN 0-130101171-5

[Lee 2005] Lee, Kwang H. ; Kacprzyk, Janusz (Hrsg.): *First Course on Fuzzy Theory and Applications*. Bd. 27. Springer Berlin Heidelberg, 2005. – ISBN 3-540-22988-4

[Li et al. 2009a] Li, X. ; Ji, H. ; Yu, F. R. ; Zheng, R.: A FCM-based Peer Grouping Scheme for Node Failure Recovery in Wireless P2P File Sharing. In: *Communications, 2009. ICC'09. IEEE International Conference on* (2009), 06, pp. 1–5

[Li et al. 2009b] Li, X. ; Ji, H. ; Zheng, R. ; Li, Y. ; Yu, F. R.: A novel team-centric peer selection scheme for distributed wireless p2p networks. In: *Wireless Communications and Networking Conference* (2009), 04, pp. 1–5

[Li and Yi 2017] Li, X. ; Yi, H.: Structural properties of fuzzy graphs. In: *Iranian Journal of fuzzy systems* 14 (2017), Nr. 1, pp. 131–144

[Lou and Fang 2001] Lou, W. ; Fang, Y.: A multipath routing approach for secure data delivery. In: *MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No.01CH37277)* (2001). ISBN 0-7803-7225-5

[Luo 2012] Luo, Y.: *Survey on P2P Traffic Managements*. Bd. 145. Springer Berlin Heidelberg, 2012. – 191–196 S

[Ma and Steenkiste 1997] Ma, Q. ; Steenkiste, P.: On Path Selection for Traffic with Bandwidth Guarantees. In: *Proceedings of IEEE International Conference on Network Protocols* (1997), 10, pp. 191–202

[Ma et al. 1996] Ma, Q. ; Steenkiste, P. ; Zhang, H.: Routing High-bandwidth Traffic in Max-min Fair Share Networks. In: *In Conference proceedings of ACM SIGCOMM'96* (1996), 08, pp. 206–217

[Medhi and Ramasamy 2007] MEDHI, D. ; RAMASAMY, K. ; ELSEVIER (Hrsg.): *Network Routing. Algorithms, Protocols and Architectures.* Morgan Kaufmann Publishers, 2007. – ISBN 10: 0-12-088588-3

[Mordeson and Nair 2000] MORDESON, J. N. ; NAIR, P. S.: *Fuzzy graphs and fuzzy hypergraphs*. Bd. 46. Physica-Verlag Heidelberg, 2000. – ISBN 978-3-7908-1286-2

[Mullai 2016] MULLAI, M.: Railway Route Optimization System Using Fuzzy Dijkstra's Algorithm. In: *International Journal of Applied Mathematical Sciences* 9 (2016), Nr. 1, pp. 117–123. – URL http://www.ripublication.com. – ISSN ISSN 0973-0176

[Peter E. et al. 1968] PETER E., Hart ; NILSSON, Nils J. ; RAPHAEL, Bertram: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In: *IEEE Transactions on Systems Science and Cybernetics* 4 (1968)

[R. Maaloul and Cousin 2018] R. MAALOUL, L. Ch. F. ; COUSIN, B. A.: Two Node-Disjoint Paths Routing for Energy-Efficiency and Network Reliability. In: *25th IEEE International Conference on Telecommunication (ICT 2018)* (2018), 06

[Rosen et al. 2001] ROSEN, E. ; VISWANATHAN, A. ; CALLON, R.: Multiprotocol Label Switching Architecture. In: *RFC 3031, IETF* (2001)

[Rosenfeld 1975] ROSENFELD, A.: Fuzzy Graphs. In: *Fuzzy Sets and Their Applications* (1975), pp. 77–95

[Ross 2004] ROSS, T. J.: *Fuzzy Logic with Engineering Applications.* Second Edition. John Wiley and Sons, Ltd., 2004. – ISBN 0-470-86075-8

[Shamshirband et al. 2016] SHAMSHIRBAND, S. ; BANJANOVIC-MEHMEDOVIC, L. ; BOSANKIC, I. ; KASAPOVIC, S. ; WAHAB, A. W. Bin A.: Adaptive Neuro-Fuzzy Determination of the Effect of Experimental Parameters on Vehicle Agent Speed Relative to Vehicle Intruder. In: *Plos One* (2016). – URL http://doi.org/10.1371/journal.pone.0155697

[van Steen 2010] STEEN, M. van: *Graph Theory and Complex Networks. An Introduction.* 2010. – ISBN 978-9081540612

[Tian et al. 2017] TIAN, Y. ; ZHOU, Q. ; ZHANG, F. ; LI, J.: Multi-hop clustering routing algorithm based on fuzzy inference and multi-path tree. In: *International Journal of distributed Sensor Networks* 13 (2017), Nr. 5

[Tomovic et al. 2015] TOMOVIC, S. ; RADUSINOVIC, I. ; PRASAD, N.: Performance comparison of QoS routing algorithms applicable to large-scale SDN networks. In: *International Conference on Computer as a Tool (EUROCON 2015)* 7313698 (2015), pp. 191–202

[Umezaki et al. 2012] Umezaki, K. ; Spaho, E. ; Barolli, L. ; Xhafa, F. ; Younas, M. ; Iwashige, J.: An Integrated Fuzzy-based Trustworthiness System for P2P Communications in JXTA-Overlay. In: *Seventh International Conference on Broadband, Wireless Computing, Communication and Applications* (2012), 11, pp. 278–284

[Valdés et al. 2019] Valdés, L. ; Ariza, A. ; Allende, S. M. ; Joya, G.: Searching the Shortest Pair of Edge-Disjoint Paths in a Communication Network. A Fuzzy Approach. In: *Advances in Computational Intelligence. IWANN. Lecture Notes in Computer Science, Springer, Cham.* 11507 (2019), pp. 640–652. – URL https://doi.org/10.1007/978-3-030-20518-8_53

[Valdés et al. 2013] Valdés, L. ; Ariza, A. ; Allende, S. M. ; Parada, R. ; Joya, G.: Study of alternative strategies to selection of peer in P2P wireless mesh networks. In: *Advances in Computational Intelligence, IWANN 2013, Lecture Notes in Computer Science. Springer Berlin Heidelberg* 7902 (2013), pp. 124–132. – URL http://dx.doi.org/10.1007/978-3-642-38679-4_11

[Valdés et al. 2021] Valdés, L. ; Ariza, A. ; Allende, S. M. ; Triviño, A. ; Joya, G.: Search of the shortest path in a communication network with fuzzy cost functions. In: *Symmetry. Manuscript ID: symmetry-1281387* 13(8) (2021)

[Valdés et al. 2015] Valdés, L. ; Montesinos, S. ; Ariza, A. ; Allende, S. M. ; Joya, G.: Peer selection in P2P wireless mesh networks: comparison of different strategies. In: *Soft Computing* 19 (2015), pp. 2447–2455

[Varga 2001] Varga, A.: *OMNeT++ Discrete Event Simulator*. 2001. – URL https://omnetpp.org/

[Varga 2014] Varga, A.: *Omnet++*. 2014

[Wang and Crowcroft 1996] Wang, Z. ; Crowcroft, J.: Quality-of-Service Routing for Supporting Multimedia Applications. In: *IEEE Journal on Selected Areas in Communications* 14 (1996), Nr. 7, pp. 1228–1234

[Yager 1979] Yager, R. R.: On solving fuzzy mathematical relationships. In: *Information Control* (1979), Nr. 41, pp. 29–55. – URL https://core.ac.uk/download/pdf/82404516.pdf. ISBN 0019-9958/79/040029-27502.00/0

[Yang and Papavassiliou 2001] Yang, J. ; Papavassiliou, S.: Improving network security by multipath traffic dispersion. In: *Proceedings Communications for Network-Centric Operations: Creating the Information Force* (2001), 02

[Yu and l. Q. Dat 2014] Yu, V. F. ; Q. Dat l.: An improved ranking method for fuzzy numbers with integral values. In: *Applied Soft Computing* (2014), Nr. 14, pp. 603–608. – URL http://www.elsevier.com/locate/asoc

[Zadeh 1962] Zadeh, L. A.: From circuit theory to system theory. In: *Proceedings I.R.E.*, 1962, pp. 856–865

[Zadeh 1965] ZADEH, L. A.: Fuzzy Sets. In: *Information and Control* 8 (1965), Nr. 3