**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : Samodro Bagus Prasetyanto**

**NRP : 5110100228**

**DOSEN WALI : Dr.techn. Ir. Raden Venantinus Hari Ginardi, M.Sc.**

**DOSEN PEMBIMBING : 1. Bilqis Amaliah, S.Kom., M.Kom.  
 2. Dr*.* Chastine Fatichah*,* S.Kom., M.Kom.**

# JUDUL TUGAS AKHIR

“Metode *fuzzy node combination* untuk menyelesaikan masalah pencarian rute terpendek pada lingkungan yang tidak pasti”

# LATAR BELAKANG

Pada umumnya algoritma untuk menyelesaikan masalah pencarian rute terpendek adalah algoritma *Dijkstra*. Namun, algoritma *Dijkstra* tidak dapat menyelesaikan masalah pencarian rute terpendek pada lingkungan yang tidak pasti. Lingkungan yang tidak pasti dalam pencarian rute terpendek adalah jika bobot di tiap *edge* tersebut terdiri dari tiga bobot atau empat bobot, sehingga dibutuhkan parameter *fuzzy*. Pertama, untuk menentukan bagaimana penambahan dari dua jalur yang terhubung. Lainnya, untuk menentukan bagaimana membandingkan jarak dari dua rute yang direpresentasikan oleh bilangan *fuzzy .*

Teori himpunan *fuzzy* adalah teori yang dapat menangani suatu permasalahan dengan informasi yang kabur dan dapat digunakan untuk berbagai macam permasalahan semisal penilaian lingkungan, pengenalan pola dan pembuat keputusan. Metode *node combination* adalah metode untuk menyelesaikan masalah pencarian rute terpendek selain algoritma *Dijkstra.* Metode *node combination* mudah dipahami karena dalam pembuatan program tersebut tidak membutuhkan himpunan yang berisi informasi jarak pada *node*.

Oleh karena itu, Tugas Akhir ini akan menerapkan penggabungan dari metode *node combination* dan teori himpunan *fuzzy*. Sehingga, masalah pencarian rute terpendek pada lingkungan yang tidak pasti dapat diselesaikan.

# RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana menerapkan bilangan *fuzzy* untuk menambahkan dua jalur terhubung?
2. Bagaimana menerapkan bilangan *fuzzy* untuk membandingkan dua rute?
3. Bagaimana mengkombinasikan metode *node combination* dan teori himpunan *fuzzy* untuk mencari rute terpendek?
4. Bagaimana menampilkan *node* asal pada aplikasi yang dibangun?

# BATASAN MASALAH

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki batasan sebagai berikut :

1. Teori himpunan *fuzzy* yang digunakan adalah segitigabilangan *fuzzy* dan trapesiumbilangan *fuzzy*
2. Jumlah bilangan *fuzzy* di tiap rute pada satu graf memiliki jumlah yang sama
3. Jumlah *node* pada satu graf tidak melebihi 200 *node*.
4. Bahasa pemrograman yang digunakan adalah bahasa C/C++
5. Angka pada bobot *edge* harus naik dan positif

# TUJUAN PEMBUATAN TUGAS AKHIR

Tugas Akhir ini memiliki tujuan sebagai berikut :

1. Menerapkan bilangan *fuzzy* untuk menambahkan dua jalur terhubung
2. Menerapkan bilangan *fuzzy* untuk membandingkan dua rute
3. Mengkombinasikan metode *node combination* dan teori himpunan *fuzzy* untuk mencari rute terpendek
4. Menampilkan *node* asal pada aplikasi yang dibangun

# MANFAAT TUGAS AKHIR

Manfaat yang diharapkan dari Tugas Akhir ini adalah metode ini dapat diterapkan untuk aplikasi di sistem transportasi, manajemen logistik dan berbagai macam permasalahan lainnya yang membutuhkan perhitungan rute terpendek

# TINJAUAN PUSTAKA

Pada bab ini akan dibahas tinjauan pustaka yang digunakan dalam mengerjakan Tugas Akhir, yaitu algoritma *Dijkstra,* metode *node combination,* teori himpunan *fuzzy,* bilangan *fuzzy,* dan pemodelan matematis untuk operasi pada bilangan *fuzzy*.

## 8.1 Algoritma *Dijkstra*

Menurut Edsger W. Dijkstra, *Dijkstra* adalah algoritma yang dipakai dalam memecahkan permasalahan satu jarak terpendek untuk sebuah graf berarah dengan bobot-bobot sisi yang bernilai tidak negatif [[1](#EWD59),[2](#Cor09)]. Misalnya, bila *vertices* dari sebuah graf melambangkan kota-kota dan bobot sisi melambangkan jarak antara kota-kota tersebut, maka algoritma *Dijkstra* dapat digunakan untuk menemukan jarak terpendek antara dua kota.

*Pseducode* dari algoritma *Dijkstra* adalah sebagai berikut [[3](#LuX11)] :

*Dijkstra*(G,s)

1 d[s] := 0, vu := vs, V := V – {s} /\*Inisialisasi\*/

2 while d[u] < ∞ and |V| > 0

3 V := V – {u} /\*tandai u telah dikunjungi/

4 for each j in V

5 d[j] := min{d[j],d[u] +W[u, j]} /\*perbarui nilai jarak pada d \*/

6 d[u] := nilai jarak pada d yang paling kecil di V

/\*diakhir algoritma, rute terpendek berada pada vektor d\*\

Algoritma *Dijkstra* memiliki tiga langkah, langkah-langah dari algoritma *Dijkstra* adalah sebagai berikut [[3](#LuX11)]:

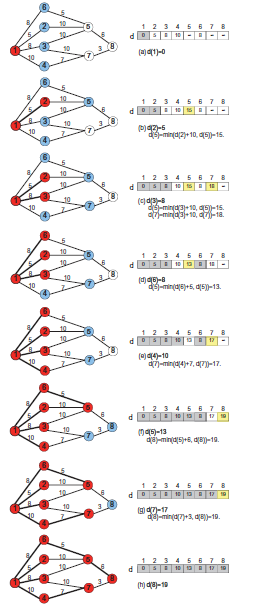
Langkah 0. Inisialisasi. d[s] := 0, vu := vs, V := V – {s}

Langkah 1. Memilih *node* yang belum dikunjungi dengan nilai jarak pada d yang paling kecil di V dan nilai jarak tersebut kurang dari ∞. Jika tidak ada *node* di V maka hentikan perulangan.

Langkah 2. Tandai *node* yang terpilih bahwa telah dikunjungi, V := V – {u}.

Langkah 3. Perbarui nilai jarak pada d, d[j] := min{d[j],d[u] +W[u, j]}. Selanjutnya, pergi ke Langkah 1

Gambar **1** merupakan ilustrasi dari algoritma *Dijkstra.* Ilustrasi tersebut adalahuntuk menemukan jarak terpendek dari *node* asal ke *node* lainnya, dimana d(i) adalah jarak *node* asal ke *node* i [[3](#LuX11)].



Gambar 1. Algoritma Dijkstra [3]

## 8.2 Metode *Node Combination*

Metode *node combination* adalah metode untuk mencari rute terpendek dengan cara menggabungkan *node* asal ke *node* terdekatnya [[3](#LuX11)]*.* Metode *node combination* adalah pengembangan dari algoritma *Dijkstra.* Metode *node combination* mudah dipahami karena dalam pembuatan program tersebut tidak membutuhkan himpunan yang berisi informasi jarak pada *node* [[3](#LuX11)].

*Pseducode* dari metode *node combination* adalah sebagai berikut [[3](#LuX11)] :

*Node\_Combination*(G,s)

1 W[s,u] := 0, vu := vs, V := V – {s} /\*Inisialisasi\*/

2 while W[s,u] < ∞ and |V| > 0

3 V := V – {u} /\*Gabungkan *node* \*/

4 for each j in V

5 W[s,j] := min{W[s, j],W[s, u] + W[u, j]} /\*perbarui nilai

6 Vu := *node* terdekat dengan s di V bobot pada sisi\*/

/\*diakhir metode, jarak terpendek berada pada nilai bobot baris s di W\*\

Metode *node combination* memiliki tiga langkah, langkah-langah dari metode *node combination* adalah sebagai berikut [[3](#LuX11)] :

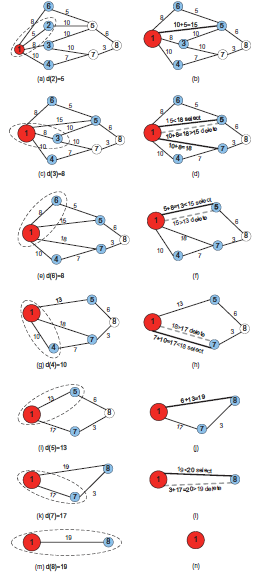
Langkah 0. Inisialisasi. W[s,u] := 0, vu := vs, V := V – {s}

Langkah 1. Memilih *node* yang terdekat dengan Vs. Jika tidak ada *node* yang terhubung dengan Vs, maka hentikan perulangan

Langkah 2. Gabungkan *node* tersebut dan hapus *node* Vk, V := V – {u}.

Langkah 3. Perbarui nilai bobot pada tiap sisi, W[s,j] := min{W[s, j],W[s, u] + W[u,j]}. Selanjutnya, pergi ke Langkah 1

Gambar **2** merupakan ilustrasi dari metode *node combination.* Ilustrasi tersebut adalahuntuk menemukan jarak terpendek dari *node* asal ke *node* lainnya, dimana W(s,i) adalah jarak terpendek *node* s ke *node* i [[3](#LuX11)].



Gambar 2. Metode *Node Combination* [3]

## 8.3 Teori Himpunan *Fuzzy*

Teori himpunan *fuzzy* adalah teori yang dapat menangani suatu permasalahan dengan informasi yang kabur dan dapat digunakan untuk berbagai macam permasalahan [[4](#Den113)] semisal penilaian lingkungan [[5](#Sad08),[6](#Den11)], pengenalan pola [[7](#Den04),[8](#Liu11),[9](#YeJ11)] dan pembuat keputusan [[10](#Den111),[11](#Den112)]. Himpunan *fuzzy* adalah misalkan X semesta, dimana Ã adalah himpunan bagian dari X untuk setiap x X, terdapat bilangan µÃ(x) [0,1] yang merupakan nilai derajat keanggotaan dari x di Ã, dan bilangan itu disebut fungsi keanggotaan Ã [[12](#Dub80)]

## 8.4 Bilangan *Fuzzy*

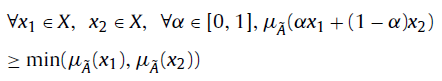
Bilangan *fuzzy* adalah bilangan *fuzzy* Ã adalah normal dan *convex fuzzy* adalah bagian dari X [[12](#Dub80)]

Normal yang dimaksud adalah sebagai berikut [[4](#Den113)] :

F:\Kuliah\Kuliah Semester 7\TA\Gambar\fuzzynumbernormal.png

(1)

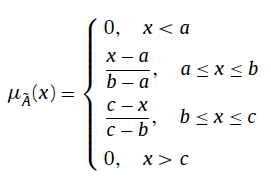
*Convex* yang dimaksud adalah sebagai berikut [[4](#Den113)] :



(2)

s

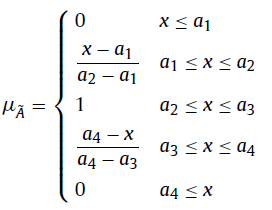
Gambar 3 adalah sebuah segitiga bilangan *fuzzy.* Sebuah segitigabilangan *fuzzy* Ã dapat didefinisikan oleh Ã = (a,b,c) [[4](#Den113)], dimana keanggotaan tersebut ditentukan sebagai berikut [[4](#Den113)] :



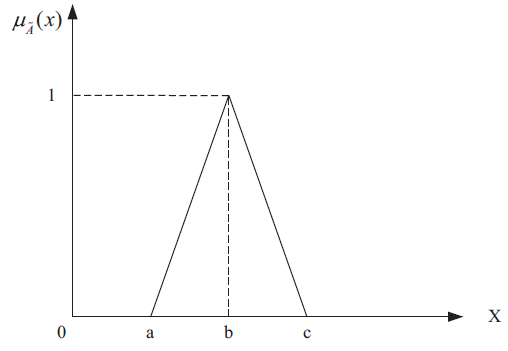
(3)

S

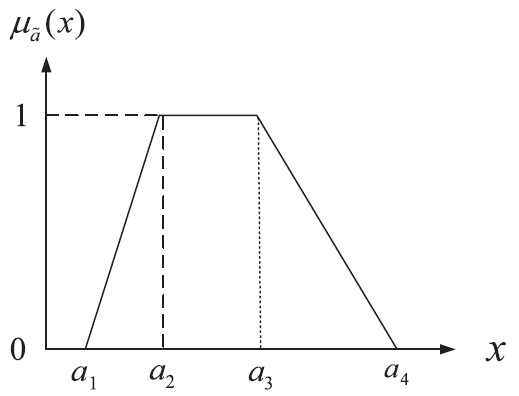
Gambar 4 adalah sebuahtrapesiumbilangan *fuzzy.* Sebuah trapesiumbilangan *fuzzy* Ã dapat didefinisikan oleh Ã = (a1,a2,a3,a4) [[4](#Den113)], dimana keanggotaan tersebut ditentukan sebagai berikut [[4](#Den113)] :



(4)



Gambar 3. Sebuahsegitigabilangan *fuzzy* [4]



Gambar 4. Sebuah trapesiumbilangan *fuzzy* [4]

## 8.5 Pemodelan matematis untuk operasi pada bilangan *fuzzy*

Pemodelan matematis untuk operasi pada segitigabilangan *fuzzy* yang didasarkan pada metode *graded mean integration representation* [[13](#Cho03)], biasanya digunakan untuk algoritma mencari rute terpendek pada lingkungan yang tidak pasti [[4](#Den113)]. Sebuah segitiga bilangan *fuzzy* denganÃ = (a1,a2,a3) [[4](#Den113)], *The graded mean integration representation* pada segitiga bilangan *fuzzy* dapat didefinisikan sebagai berikut [[4](#Den113)] :

P(Ã) = (a1 + 4 a2 + a3) (5)

Misalkan Ã = (a1,a2,a3) dan B = (a1,a2,a3) adalah dua segitiga bilangan *fuzzy*. *The graded mean integration representation* pada segitiga bilangan *fuzzy* Ã dan Badalah sebagai berikut [[4](#Den113)] :

P(Ã) = (a1 + 4 a2 + a3)

P(B) = (b1 + 4 b2 + b3)

Representasi operasi penambahan pada segitiga bilangan *fuzzy* Ã dan B dapat didefinisikan sebagai berikut [[4](#Den113)] :

P(Ã B) = P(Ã) + P(B) = (a1 + 4 a2 + a3) + (b1 + 4 b2 + b3) (6)

Representasi operasi pengalian pada segitiga bilangan *fuzzy* Ã dan B dapat didefinisikan sebagai berikut [[4](#Den113)] :

P(Ã B) = P(Ã) P(B) = (a1 + 4 a2 + a3) (b1 + 4 b2 + b3) (7)

Sebuah trapesium bilangan *fuzzy* denganÃ = (a1,a2,a3,a4) [[4](#Den113)], *The graded mean integration representation* pada trapesium bilangan *fuzzy* dapat didefinisikan sebagai berikut [[4](#Den113)] :

P(Ã) = (a1 + 2 a2 + 2 a3 + a4) (8)

Misalkan Ã = (a1,a2,a3,a4) dan B = (a1,a2,a3,a4) adalah dua trapesium bilangan *fuzzy*. *The graded mean integration representation* pada trapesium bilangan *fuzzy* Ã dan Badalah sebagai berikut [[4](#Den113)] :

P(Ã) = (a1 + 2 a2 + 2 a3 + a4)

P(B) = (b1 + 2 b2 + 2 b3 + b4)

Representasi operasi penambahan pada trapesium bilangan *fuzzy* Ã dan B dapat didefinisikan sebagai berikut [[4](#Den113)] :

P(Ã B) = P(Ã) + P(B) = (a1 + 2 a2 + 2 a3 + a4)

(b1 + 2 b2 + 2 b3 + b4) (9)

Representasi operasi pengalian pada trapesium bilangan *fuzzy* Ã dan B dapat didefinisikan sebagai berikut [[4](#Den113)] :

P(Ã B) = P(Ã) P(B) = (a1 + 2 a2 + 2 a3 + a4)

(b1 + 2 b2 + 2 b3 + b4) (10)

# RINGKASAN ISI TUGAS AKHIR

Pada Tugas Akhir ini penulis mengusulkan implementasi metode *node combination* dan teori himpunan *fuzzy* untuk menyelesaikan masalah pencarian rute terpendek pada lingkungan yang tidak pasti.

*Pseducode* dari metode *fuzzy node combination* adalah sebagai berikut :

*Fuzzy\_Node\_Combination*(G,s)

1 W[s,u] := {0}, vu := vs, V := V – {s}, fuzzy := 3 atau 4 /\*Inisialisasi\*/

2 while W[s,u] < ∞ and |V| > 0

3 V := V – {u} /\*Gabungkan *node* \*/

4 for each j in V

5 alt := dist(W[s,u], fuzzy) + dist(W[u,j], fuzzy)

6 if alt < dist(W[s,j], fuzzy)

7 W[s,j] := add(W[s, u], W[u, j], fuzzy) /\*perbarui nilai pada *node* W[s,j]\*/

8 Vu := *node* terdekat dengan s di V

/\*diakhir metode, jarak terpendek berada pada baris s di W\*\

*Pseducode* **dist** dan **add** dari metode *fuzzy node combination* adalah sebagai berikut :

dist(a, fuzzy)

1 if fuzzy = 3

2 **return**  (a1 + 4 a2 + a3)

3 else

4 **return**  (a1 + 2 a2 + 2 a3 + a4)

add(a, b, fuzzy)

1 if fuzzy = 3

2 a1 = a1 + b1

3 a2 = a2 + b2

4 a3 = a3 + b3

5 **return** a

6 else

7 a1 = a1 + b1

8 a2 = a2 + b2

9 a3 = a3 + b3

10 a4 = a4 + b4

11 **return** a

Metode *fuzzy node combination* memiliki tiga langkah, langkah-langah dari metode fuzzy *node combination* adalah sebagai berikut [[3](#LuX11)] :

Langkah 0. Inisialisasi. W[s,u] := 0, vu := vs, V := V – {s}

Langkah 1. Memilih *node* yang terdekat dengan Vs. Jika tidak ada *node* yang terhubung dengan Vs, maka hentikan perulangan

Langkah 2. Gabungkan *node* tersebut dan hapus *node* Vk, V := V – {u}.

Langkah 3. Perbarui nilai bobot pada tiap sisi jika bobot yang baru lebih kecil, W[s,j] := add(W[s, u], W[u, j], fuzzy). Selanjutnya, pergi ke Langkah 1

# METODOLOGI

## Penyusunan proposal tugas akhir

Proposal ini berisi tentang implementasi metode *node combination* dan teori himpunan *fuzzy* untuk mencari rute terpendek pada lingkungan yang tidak pasti. Metode *node combination* adalah modifikasi dari algoritma *Dijkstra* yang lebih mudah diimplementasi dalam pembuatan program. Teori himpunan *fuzzy* yang digunakan adalah segitiga bilangan *fuzzy* dan trapesium bilangan *fuzzy*. Sehingga, masalah pencarian rute terpendek pada lingkungan yang tidak pasti dapat diselesaikan.

## Studi literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur yang diperlukan. Informasi dan studi literatur yang diperlukan adalah implementasi algoritma *Djikstra,* metode *node combination* maupun teori himpunan *fuzzy.*

## Implementasi perangkat lunak

Kakas bantu yang digunakan untuk mengimplementasikan Tugas Akhir ini adalah Dev-C++. Adapun bahasa pemrograman yang digunakan adalah bahasa C/C++.

## Pengujian dan evaluasi

Algoritma tersebut akan diuji coba dengan beberapa studi kasus. Studi kasus tersebut secara garis besar dibagi menjadi dua, yaitu :

1. Studi kasus segitiga bilangan *fuzzy*
2. Studi kasus trapesium bilangan *fuzzy*

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | 2013 | | | | | | | | | | | | | | | | | | | | |
| September | | | | Oktober | | | | November | | | | Desember | | | | | Januari | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian dan Evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan Buku Tugas Akhir |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, pp. 269-271, 1959. |
| [2] | T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Dijkstra's Algorithms," in *Introduction to Algorithms, Third Edition*. London, England: The MIT Press, 2009, ch. 6, pp. 658-664. |
| [3] | X. Lu and M. Camitz, "Finding the shortest paths by node combination," *Applied Mathematics and Computation 217*, pp. 6401-6408, 2011. |
| [4] | Y. Deng, Y. Chen, Y. Zhang, and S. Mahadevan, "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment," *Applied Soft Computing 12*, pp. 1231-1237, 2011. |
| [5] | R. Sadiq and S. Tesfamariam, "Developing environmental using fuzzy numbers ordered weighted averaging (FN-OWA) operators," *Stochastic Environmental Research and Risk Assesment 22*, pp. 495-505, 2008. |
| [6] | Y. Deng, W. Jiang, and R. Sadiq, "Modeling contaminant intrusion in water distribution network: a new similarity-based DST method," *Expert Systems with Applications 38*, pp. 571-578, 2011. |
| [7] | Y. Deng, W. K. Shi, and Q. Liu, "A new similarity measure of generalized fuzzy numbers and its application to pattern recognition," *Pattern Recognition Letters 25*, pp. 875-883, 2004. |
| [8] | H. W. Liu, "New similarity measures between intuitionistic fuzzy sets and between elements," *Mathematical and Computer Modelling 42*, pp. 61-70, 2005. |
| [9] | J. Ye, "Cosine similarity measures for intuitionistic fuzzy sets and their applications," *Mathematical and Computer Modelling 53*, pp. 91-97, 2011. |
| [10] | Y. Deng and F. T. S. Chan, "A new fuzzy Dempster MCDM method and its application in supplier selection," *Expert Systems with Applications 38*, pp. 6985-6993, 2011. |
| [11] | Y. Deng, F. T. S. Chan, Y. Wu, and D. Wang, "A new liguistic MCDM method based on multiple-criterion data fusion," *Expert Systems with Applications 38*, pp. 9854-9861, 2011. |
| [12] | D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Application*. New York: Academic Press, 1980. |
| [13] | C. C. Chou, "The canonical representation of multiplication operation on triangular fuzzy numbers," *Computer and Mathematics with Applications 45*, pp. 1601-1610, 2003. |