ファジィ組合せ最適化

石井博昭

大阪大学大学院工学研究科応用物理学専攻

〒565-0871 吹田市山田丘2-1
TEL: 06-6879-7868　FAX: 06-68797871
E-mail: ishiiha@ap.eng.osaka-u.ac.jp

概要：ファジィOR延いてはファジィ組合せ最適化は際物という人もあるようですが、これは最近の社会情勢から必然だと思っています。個々人の価値の多様化、社会状況の不確実、不確定性の下で、意思決定する際には、多様な情報の下で最適化が必要になってくる。このような情報をうまく利用しなければならない状況では、ますますORの役割が重要になり、ファジィ理論とORのマッチングが行われなければならない。本来、ファジィ理論は制御よりもORこそ相性がよい筈であるが、これまではそうは意識されて来なかった。データの精度、基準の融通性、環境変化に対応して最適化がこれからは必要であるからである。　まず、ファジィ理論の基礎と組合せ最適化に導入すべきファジィ概念について述べ、その後、ネットワーク理論、スケジューリング理論のおさらいをした後、幾つかのファジィ組合せ最適化問題についてその解法も含めて述べる。その際、ファジィ概念化する幾つかの要素について、その妥当性と意味について考える。確率計画法の時もそうであるが、情報をモデルにとり入れる時は、最適化の意味も重要となるからである。最後にこれからの展望について述べたいと思う。

1.　はじめに

ファジィ組合せ最適化については1995年春に行われたOR春季大会での第33回シンジウムでお話しており、くしくも中国・四国支部での開催であった。そのときは、まだファジィOR自身もあまりOR学会のなかで認知されていなかったのですが、最近は40周年記念事業でファジィORの本が刊行されるなど、だんだん認知されてきているのは嬉しいかぎりです。この6年の間にいろいろな研究も行われたので、前回とは違う部分を中心にお話したいと思います。我々がファジィ組合せ最適化を研究し始めた動機はChanas et. alによるファジィ容量の概念[6, 7]であった。彼らはこのようなファジィ概念の下で、最大フロー最小カットの定理が拡張できることを鮮やかに示したのである。我々はこれがファジィ組合せ最適化の始まりであると思っている。ファジィOR延いてはファジィ組合せ最適化は際物という人もあるようですが、これは最近の社会情勢から必然だと思っています。個々人の価値の多様化、社会状況の不確実、不確定性の下で、意思決定する際には、多様な情報の下で最適化が必要になってくる。このような情報をうまく利用しなければならない状況では、ますますORの役割が重要になり、ファジィ理論とORのマッチングが行われなければならない。本来、ファジィ理論は制御よりもORこそ相性がよい筈であるが、

これまではそうは意識されて来なかった。データの精度、基準の融通性、環境変化に対応して最適化がこれからは必要であるからである。　まず、ファジィ理論の基礎と組合せ最適化に導入すべきファジィ概念について述べ、その後、ネットワーク理論、スケジューリング理論のおさらいをした後、幾つかのファジィ組合せ最適化問題についてその解法も含めて述べる。その際、ファジィ概念化する幾つかの要素について、その妥当性と意味について考える。確率計画法の時もそうであるが、情報をモデルにとり入れる時は、最適化の意味についても重要となるからである。最後にこれからの展望について述べたいと思う。

2.　ファジィ理論の基礎と関連するファジィ概念

[ファジィ集合]　　L. A .Zadeh は1965年通常の集合（ファジィ集合との対比からクリスプ集合といわれる）の概念を一般化してファジィ部分集合を定義した。先駆者の常として最初は受け入れてもらえずに苦労したようである。また、彼の集合は従来客観性こそもっとも優れたものだという西洋の科学観を打ち破り、主観をもとり入れるものであった。この点で、東洋人たる我々にはあまり違和感なくうけいれられるものであると思われる。従来の集合（クリスプ集合）は、ある要素がその集合に属するか、属さないかがはっきりしており、この帰属性を属するときは1、属さないときは0で示す。この集合を表す0または1の関数は特性関数と呼ばれている。この帰属性を一般化し0と1の間の連続量としたのがメンバシップ関数（図2.1参照、これは下限つきファジィ容量の例）と呼ばれるもので、このメンバシップ関数で定義される集合がファジィ部分集合である。すなわち、クリスプな全体集合 X があって、次の順序対で定義されるのがファジィ集合 A である。

$A=\{(x,\mu_A(x))|x\in X\}$。これに対してクリスプ集合は特性関数 $C_A(x)=\begin{cases}1 & x\in A,\\ 0 & x\notin A,\end{cases}$

で示される。ファジィ部分集合は通常 "部分" をとってファジィ集合と呼ばれる。

ファジィ集合とクリスプ集合を関連づける概念として, α-レベル集合と呼ばれるクリスプ集合が次のように定義される。

α-レベル集合

全体集合 X におけるファジィ集合 A について, A のメンバシップ関数 $\mu_A(x)$ が任意の実数 $\alpha\in[0,1]$ 以上となるような $x\in X$ のクリスプ集合を α-レベル集合 A_α という。

$$A_\alpha=\{x\in X|\mu_A(x)\geqq\alpha\}\quad(0\leqq\alpha\leqq 1)\tag{1}$$

α-レベル集合を定義することにより, ファジィ集合を通常の数学体系で取り扱うことが可能となるが, クリスプ集合間の数学的関係（写像）をファジィ集合に拡張した概念として拡張原理 (extension principle) が存在する。

拡張原理

写像 $f:X\to Y$ (X, Y はクリスプ集合) に関して, X におけるファジィ集合 A の f による像 $f(A)$ は, Y におけるファジィ集合として次のようなメンバシップ関数によって定義される。

$$\mu_{f(A)}(y)=\begin{cases}\sup\limits_{y=f(x)}\mu_A(x) & (f^{-1}(y)\neq\phi)\\ 0 & (f^{-1}(y)=\phi)\end{cases}\tag{2}$$

[ファジィ関係]

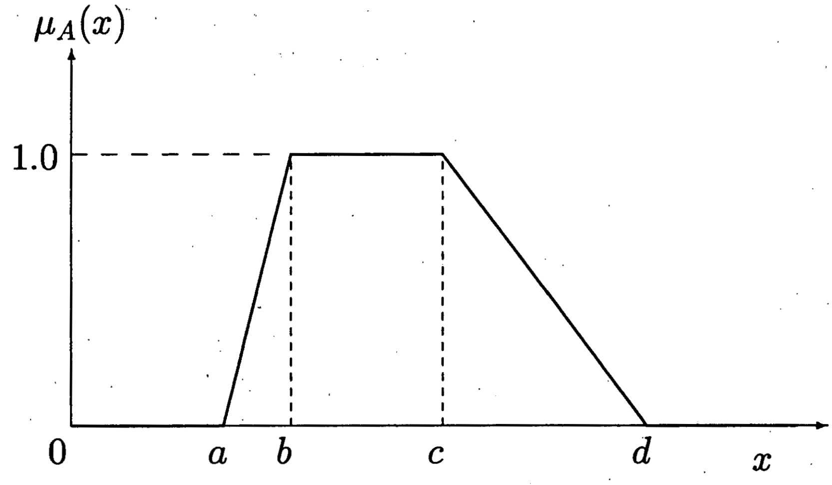

図 2.1　ファジィ集合

通常、関係を考えるとき，２つの対象の間の関係が基本となる。特に、人間関係や社会的関係を考えるときは、この様な１対１の関係から全体の関係が構成されていることが多いが、これを２項関係という。勿論、一般には、n 個の対象を同時に考えることもあり、n 項関係という。x と y の間に関係 R が成り立つとき，xRy で表す。x が属する集合 X と y が属する集合 Y が異なる場合も，$X=Y$ の場合もある。通常の２項関係は X と Y の直積集合 $X\times Y$ の部分集合 $\{(x,y)|xRy, x\in X, y\in Y\}$ を与えることと等価である。そして $X=Y$ の場合，R を X 上の２項関係というが，ＯＲではこの場合をファジィ概念化した X 上のファジィ２項関係を考えることが多い。このファジィ関係は，各要素 $(x,y), x,y\in X$ に対して，関係の成り立つ度合いとしてのメンバシップ関数 $\mu_R(x,y)$ を与える事により定義される。これは直積集合上のファジィ部分集合と等価である。また、有限集合上の２項関係はグラフとして表現できるが、同様に有限集合上のファジィ関係はアークの存在度合いをメンバシップ関数とするファジィグラフとなる。ファジィ組合せ最適化への応用としては、スケジューリング問題での仕事間の先行関係のファジィ概念化としてのファジィ先行関係が考えられる。X,Y が有限個の要素からなる場合は以下のような関係行列で表されることが多い。この行列の各要素 (i,j) は $\mu_R(x_i,y_j)$ すなわち x_i と y_j の関係の度合いを表している。例えば $(1,1)$ 要素の 0.7 はこの関係の度合いが 0.7 であることを示している。

$$\begin{bmatrix} 0.7 & 0.4 & 0.3 \\ 1.0 & 0.8 & 0.6 \\ 0.5 & 1.0 & 0.7 \\ 0.1 & 0.9 & 0.6 \end{bmatrix}$$

[ファジィグラフ]

これはファジィ関係の特別な場合とも考えられるが、点と枝からなるグラフ（通常の2項関係を示すもの）（後述）をファジィ概念化したものと考えることができる。点 i, j の間に枝が存在する可能性 $\mu(i, j)$ は、点 i, j で表される要素間に関係がある度合いを表している。さらに、点の存在可能性を考慮したものもあり、要素自身からなるファジィ集合上のファジイ関係を表している。ファジィグラフの点や枝に長さ、容量などの数値を付随させたものをファジィネットワークという。さらには以下にのべるファジィ数を長さとして考えるものもあり、これらはファジィネットワーク最適化問題の構成要素として重要となる。

[ファジィ数]

通常実数のファジィ概念化がファジィ数といわれ、だいたい m ぐらい、若干名、40名程度などのおおよその数や推定された値、さらには人数に少しぐらい融通が利く状況で用いられる。例えば、入学定員などは成績が良い学生が多ければ、募集人員より少し余分にとることもある一方、レベルが低ければ、少なくしか取らない場合もある。このようなデータの曖昧性を表すファジィ概念はORなどの数理的意思決定には非常に有用であり、これからどんどんモデル化に使われるあるいはもっとも用いられるべきファジィ概念である。以上のイメージから数学的には以下のように定式化される。

ファジィ数

実直線上で定義された正規かつ凸ファジィ集合で、メンバシップ関数が区分的に連続なものをファジィ数という。ただし、正規性はメンバシップ値の上限が1であることをいい、ファジィ集合 A が凸であるとは、$\forall x_1, x_2 \in X, 0 \leqq \forall\lambda \leqq 1$ に対して

$$\mu_A(\lambda x_1 + (1-\lambda)x_2 \geqq min(\mu_A(x_1), \mu_A(x_2))$$

が成り立つことをいう。ここで正規とはメンバシップ関数 $\mu_A(\cdot)$ の最大値が1であることを意味している。また凸性はその α　レベル集合が単一区間になることと等価であり、上記のような性質をもつ関数を準凹関数という。R^1 上に拡張原理を適用すれば, 2項演算 $*$ を2つのファジィ数 M, N の2項演算$\circledast$に拡張することができ, そのメンバシップ関数は

$$\mu_{M \circledast N}(z) = \sup_{z=x*y} \min\left(\mu_M(x),\, \mu_N(y)\right) \tag{3}$$

となる。特に, 2項演算として $+$, $-$, $\times$, $\div$ を考えれば, 2つのファジィ数 M, N の和, 差, 積, 商については次のようになる。

$$\text{加法 } M \oplus N: \quad \mu_{M\oplus N}(z) = \sup_{z=x+y} \min\left(\mu_M(x),\, \mu_N(y)\right) \tag{4}$$

$$\text{減法 } M \ominus N: \quad \mu_{M\ominus N}(z) = \sup_{z=x-y} \min\left(\mu_M(x),\, \mu_N(y)\right) \tag{5}$$

$$\text{乗法 } M \otimes N: \quad \mu_{M\otimes N}(z) = \sup_{z=x\times y} \min\left(\mu_M(x),\, \mu_N(y)\right) \tag{6}$$

$$\text{除法 } M \oslash N: \quad \mu_{M\oslash N}(z) = \sup_{z=x\div y} \min\left(\mu_M(x),\, \mu_N(y)\right) \tag{7}$$

しかし, 各々のファジィ数のメンバシップ関数が複雑になると, これらを実際に求めることは非常に困難である。そこで, ファジィ数の演算を計算機を用いて効率よく行うために, Dubois と Prade [14] は ***L-R* ファジィ数** (*L-R* fuzzy number) を導入した。

***L-R* ファジィ数**

次のようなメンバシップ関数に制限されるファジィ数 M を L-R ファジィ数と呼ぶ。

$$\mu_M(x) = \begin{cases} L\left(\dfrac{m-x}{\alpha}\right) & (x \leqq m,\ \alpha > 0) \\ \\ R\left(\dfrac{x-m}{\beta}\right) & (x \geqq m,\ \beta > 0) \end{cases} \tag{8}$$

ただし, L, R は以下の条件を満たすよう定義される**型関数** (shape function) である。

- $L(x) = L(-x),\ R(x) = R(-x)$
- $L(0) = R(0) = 1$
- $L(x),\ R(x)$ は $[0,\ \infty)$ で非増加

また, m は**平均** (mean) と呼ばれ, パラメータ α, β はメンバシップ関数の横方向への**拡がり** (spread) を表す。これらによって L-R ファジィ数は

$$M = (m,\ \alpha,\ \beta)_{LR} \tag{9}$$

と表すことができる。(9) の L-R ファジィ数 M は, 例えば 図 2.2 で示されるメンバシップ関数によって制限される。 L-R ファジィ数の基本演算に関して, 加法, 減法に対する次

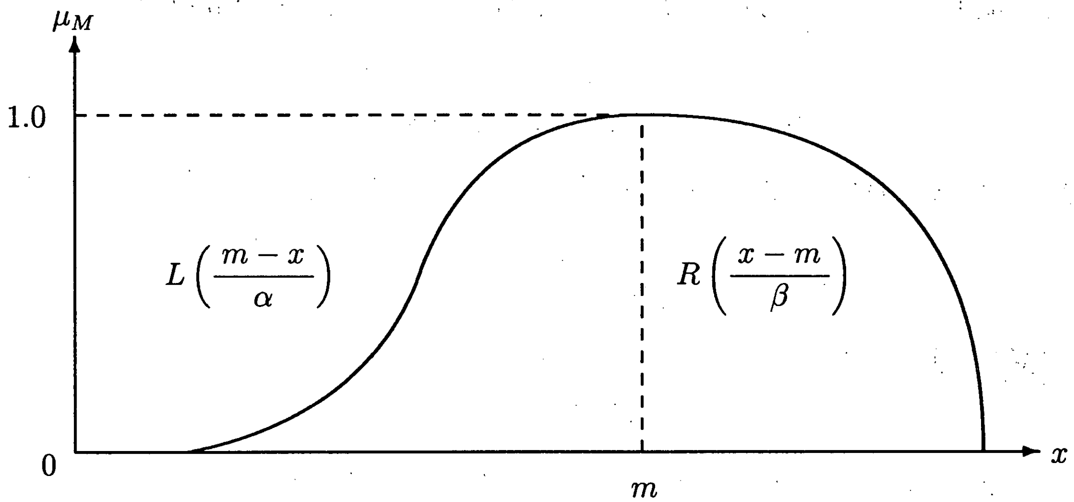

図 2.2 $M = (m,\ \alpha,\ \beta)_{LR}$ のメンバシップ関数 μ_M

のような公式が Dubois と Prade [13] によって示された。

$$\text{加法：}\quad (m,\ \alpha,\ \beta)_{LR} \oplus (n,\ \gamma,\ \delta)_{LR} = (m+n,\ \alpha+\gamma,\ \beta+\delta)_{LR} \tag{10}$$

$$\text{減法：}\quad (m,\ \alpha,\ \beta)_{LR} \ominus (n,\ \gamma,\ \delta)_{RL} = (m-n,\ \alpha+\gamma,\ \beta+\delta)_{LR} \tag{11}$$

これらは拡張原理からも明らかである。
また型関数 $L(\cdot)$ が $R(\cdot)$ と同じでメンバシップ関数が偶関数であるとき、特に L ファジィ数といい、$(m,\alpha)_L$ で示される。

[様相性とその尺度]

様相性とは古くから**様相論理学** (modal logic) で扱われている "可能性", "必然性" をさし, これら両者は双対な関係にある。すなわち, 命題 P に関して

$$\text{P は “可能” である} \Leftrightarrow \text{P でないことが “必然” でない}$$
$$\text{P は “必然” である} \Leftrightarrow \text{P でないことが “可能” でない}$$

が成立する。

この様相性に対して数理的な解析を行うために, ファジィ理論ではファジィ数 M を可能性変数 M と解釈し, そのメンバシップ関数 μ_M を M の**可能性分布関数** π_M と同一視する。可能性は確率と同様 “40% (0.4)” 等のように表現されるため, 日常では無意識のうちに混同されている場合が多いが, この違いを示すために Zadeh による以下の例を考える [62]。

“今日, ハンスは朝食に何個の卵を食べたか という質問に対して, ハンスは毎朝必ず卵を食べるので, 正解 x_0 は $X = \{1, 2, \cdots\}$ のなかにある。この際の各 $x \in X$ に対する可能性を表す可能性分布関数 $\pi(x)$, 確率 $p(x)$ は, 例えば次のようになる。

x	1	2	3	4	5	6	7	8
$\pi(x)$	1.00	1.00	1.00	1.00	0.80	0.40	0.10	0
$p(x)$	0.70	0.25	0.05	0	0	0	0	0

この例から, “確率”, “可能性” は以下のように区別されるものであると解釈できる。

確率 $\cdots$ 事象の生起に関するもの
可能性 $\cdots$ 事象生起能力に関するもの

さらに, 各々の総和に関して “確率” と “可能性” で最も異なる点は

$$\sum_{x \in X} p(x) = 1 \tag{12}$$

であるのに対し, 基本的に

$$\sum_{x \in X} \pi(x) \neq 1 \tag{13}$$

となる点である。これらのことから “確率”, “可能性” について, 例えば次のようにイメージすることができる。

- ＜確率についてのイメージ＞
  ハンスが朝食時に食べる卵の数を統計にとると, 1 個の場合が最も多く, 2 個以上の場合は徐々に少なくなる。
- ＜可能性についてのイメージ＞
  ハンスが任意時点において, 簡単（平気）に食べられる卵の数は 4 個迄で, 5~7 個では徐々に苦しくなり, 8 個目は全然食べられない。

したがって, “確率” と “可能性” の関係が以下のように整理される。

$$\begin{array}{lcl} \text{確率が高い} & \Longrightarrow & \text{可能性も高い} \\ \text{可能性が低い} & \Longrightarrow & \text{確率も低い} \\ \text{可能性が高い} & \not\Longrightarrow & \text{確率も高い} \\ \text{確率が低い} & \not\Longrightarrow & \text{可能性も低い} \end{array}$$

また, 双対関係より, 必然性について変換すると

$$\begin{array}{lcl} \text{必然性が高い} & \Longrightarrow & \text{確率も高い} \\ \text{確率が低い} & \Longrightarrow & \text{必然性も低い} \\ \text{確率が高い} & \not\Longrightarrow & \text{必然性も高い} \\ \text{必然性が低い} & \not\Longrightarrow & \text{確率も低い} \end{array}$$

Zadeh はこれらの関係を**可能性／確率調和原理** と呼んでいる。

様相測度

確率論においては確率測度が基礎として定義されているように, あるクリスプ集合に関する様相性の程度を表す尺度として, 様相測度が定義されている。

(可能性測度 1) 次の (1) ～ (3) の性質を満たす集合関数 Π を**可能性測度** (possibility measure) と呼ぶ。

(1) $\Pi : {}^{\forall}A \subseteq X \to [0,\ 1]$

(2) $\Pi(\phi) = 0,\ \Pi(X) = 1$

(3) $\Pi(A \cup B) = \max(\Pi(A),\ \Pi(B)) \quad ({}^{\forall}A,\ B \subseteq X)$

また双対関係より, 必然性測度も次のように定義できる。

(必然性測度 1) 次の (1) ～ (3) の性質を満たす集合関数 $\mathcal{N}$ を**必然性測度** (necessity measure) と呼ぶ.

(1) $\mathcal{N} : {}^{\forall}A \subseteq X \to [0,\ 1]$

(2) $\mathcal{N}(\phi) = 0,\ \mathcal{N}(X) = 1$

(3) $\mathcal{N}(A \cap B) = \min(\mathcal{N}(A),\ \mathcal{N}(B)) \quad ({}^{\forall}A,\ B \subseteq X)$

定義の内容からわかるように, 様相測度については一般の測度のような**加法性**が必ずしも要求されているわけではない。

これらは確率測度の場合と同様, 対象とする集合の各要素 x が制限される可能性分布関数 $\pi(x)$ を基に考えられ, 実際には次のような等価な定義が多く用いられる。

(可能性測度 2) X を空でない集合とする. Π が X 上の可能性測度であるとは

$$\sup_{x \in X} \pi(x) = 1 \tag{14}$$

を満たす関数 $\pi : X \to [0,1]$ が存在して

$$\Pi(A) = \sup_{x \in A} \pi(x) \quad ({}^{\forall}A \subseteq X) \tag{15}$$

となることをいう。また, 必然性測度については, 可能性測度との双対性を利用して

$$\mathcal{N}(A) = 1 - \Pi(A^c) \quad ({}^{\forall}A \subseteq X) \tag{16}$$

とすることができる。すなわち

(必然性測度 2) X を空でない集合とする。$\mathcal{N}$ が X 上の必然性測度であるとは

$$\mathcal{N}(A) = \inf\{1 - \pi(x) | x \in A^c\} \quad ({}^{\forall}A \subseteq X) \tag{17}$$

となることをいう。特に対象となる集合がファジィ集合の場合, Π, $\mathcal{N}$ はそれぞれ

$$\Pi(A) = \sup_x \min\left(\pi(x),\ \mu_A(x)\right) \tag{18}$$

$$\mathcal{N}(A) = \inf_x \max\left(1 - \pi(x),\ \mu_A(x)\right) \tag{19}$$

として定義される。

これらの様相測度を用いることにより, Dubois と Prade は **Zadeh** による定性的な原理, 可能性／確率調和原理を次のような不等式で表現した [34]。

$$\mathcal{N}(A) \leqq P(A) \leqq \Pi(A) \quad ({}^{\forall}A \subseteq X) \tag{20}$$

[ファジィマックス順序][19]

2つのファジィ数 A, B に対して

$$A \preceq B \Leftrightarrow \begin{cases} (a) & m \leqq n \\ \text{or} & \\ (b) & m \leqq^{\exists} d \leqq n : \\ & \quad \mu_A(x) \geqq \mu_B(x), \quad {}^{\forall}x < d \\ & \quad \mu_A(x) \leqq \mu_B(x), \quad {}^{\forall}x > d \end{cases}$$

m, n はそれぞれ, ファジィ数 A, B の平均を表す。 上の定義は表現は異なるが, 一般に用いられている Dubois と Prade [14] が提案したファジィマックス順序と同値である。

定理 2.1[19]

2つの L ファジィ数 $A = (m,\ \alpha)_L, B = (n,\ \beta)_L$ に対して

$$A \preceq B \Leftrightarrow n - m \geqq t_0|\alpha - \beta|$$

ここで, t_0は $\inf_{t>0,}\{t \mid L(t) = 0\}$ として定義され, L の零点と呼ばれる。

[ファジィランダム変数]

ファジィランダム変数は1978年に Kwakernaak [42] によって導入され, その後, Puri と Ralescu [49] によって理論的な土台が構築された。ファジィランダム変数とは確率変数の実現値がファジィ数となっている変数である。Kwakernaak による定義は多値論理の立場より導出されたものであり, ランダム集合を拡張した Puri-Ralescu による定義とは少し違いがあるが, Puri-Ralescu の定義のほうがより一般的である。また, 両者の定義がある条件の下では一致することも知られている [63].

ファジィランダム変数の定義にはその他にもいくつかある [18, 39, 46, 57] が, 本研究では Watanabe [58] の定義を紹介することにする。 この定義は従来のファジィランダム変数の包括的な定義であると同時に, ここで用いるファジィランダム変数と深い関わりを持っている。

ファジィランダム変数 [58]

Ω を標本空間, Λ をファジィ集合の族とし, B_Ω, B_Λ をそれぞれの $\sigma-$ 集合体, P を確率測度とする. (Ω, B_Ω, P), (Λ, B_Λ) をそれぞれ確率空間, 可測空間とするとき, Ω から Λ への可測写像 X をファジィランダム変数という。

このことは任意の $A \in B_\Lambda$ に対して $\{\omega | X(\omega) \in A\} \in B_\Omega$ が成り立つことを意味する。

次の定理は上記ファジィランダム変数の定義の十分条件になっていることが示されている。

定理 2.2 [58]

x を確率空間 (Ω, B_Ω, P) から可測空間 (Γ, B_Γ) への可測写像とし, X を Ω から Λ への写像とする。もし全単射 $h : \Lambda \to \Gamma$ が存在すれば, 可測空間 (Λ, B_Λ) と (Ω, B_Ω, P) から (Λ, B_Λ) への写像 X はファジィランダム変数である。

この定理から次の系が成り立つ。

系 2.1 [58]

X を Ω から Λ への写像とする。$\forall\omega \in \Omega$ に対して, ファジィ集合 $X(\Omega)$ のメンバシップ関数 $\mu_{X(\omega)}$ がある関数 $f(u;\theta)$ に対して $\mu_{X(\Omega)}(u) = f(u; x(\omega))$ と表されるとする。ここで θ に関して $\theta_1 \neq \theta_2$ のとき $f(u;\theta_1) \neq f(u;\theta_2)$ が成り立つならば X はファジィランダム変数である。

ファジィ集合 X のメンバシップ関数がパラメータによって決定され x が確率変数ならば, 系 2.1 から X はファジィランダム変数である。系 2.1 における条件はかなり強いが, これを満たすものは応用上有用であり, 本研究で導入するファジィランダム変数もこの条件を満たしている。 また, Kaufmann と Gupta [34] によって導入されたハイブリット数や Puri と Ralescu [49] による正規ファジィランダム変数もこの条件を満たす。

[ファジィ目標]

目的関数に "**だいたい** f_0" 以下にしたいということをあらわすファジィ集合で、そのメンバシップ値は満足度を表す。これは主観的に考えられることを多く、例えば "線形のメンバシップ関数" であれば、以下のようなメンバシップ関数となる。

$$\mu_G(z) = \begin{cases} 1, & z \leqq f_0 \\ 1 - \dfrac{z - f_0}{f_1 - f_0}, & f_1 \geqq z \geqq f_0 \\ 0, & z \geqq f_1 \end{cases}$$

逆に "**だいたい** f_0 以上にしたいというファジィ目標も考えられる。

3. スケジューリングとは

一般的に、**スケジューリング問題**とは、処理すべき対象（仕事という）とそれらを処理すべき機械（場合によってはプロセッサーという）があるときに、仕事をどういう順序に処理していくのが最適かという計画問題のことをいう。スケジューリング問題は**機械**の台数とそれにかけられる順序の制約のあるなし、あったときはその形でいくつかの場合に分けられる。まず、

1. 機械が 1 台の場合
2. 機械が複数台の場合にわけられ、複数台のときは並列型とショップ型に分かれる。並列型はその機械が、同一の時は等価並列型、一様に処理速度に差がある一様並列型、仕事によっては得手不得手がある非一様並列型がある。ショップとはもともと機械が

並べてある所を意味し、各仕事は複数の部分作業からなっていて、各部分作業についてはそれを処理する機械が決まっている。このとき、

(a) 複数の機械にかける順序があらかじめ決まっていて各仕事で同じであるフローショップ（流れ作業）型

(b) 順序があらかじめ決まっていなくて自由性があるオープンショップ型

(c) 複数の機械にかける順序があらかじめ決まっていて各仕事毎にそれが異なっているジョブショップ型

に分類できる。

最近はさまざまな工具 (Tool) を使い分けて処理を行う FMS(Flexible Manufacturing Sytems) における問題などを考えるための一般化が考えられている。すなわち仕事はいずれの機械においても処理を行うことができるが，定められた工具を必要とする。これをモデル化したものを**多機能機械** (Multi-Purpose Machines:MPM) と呼ぶ [5, 29] 。また $U_{i,j}$ の要素である全ての機械が $O_{i,j}$ を処理する際に同時に用いられるようなときも考えられており，そのような問題は**多機械処理** (Multiprocessor Task)[2, 3, 11, 21, 41] 。

スケジューリング問題を決定する要因は仕事については**処理時間**、**納期**、**先行関係**、**資源**、**仕事の分割の可否**，**最早開始可能時間**等がある。処理時間は仕事の処理を完了するのに必要な時間を意味し、納期はその時点までに仕事を完了すべき時刻、先行関係は仕事間の処理順の制約、資源とは人員や工具，空間，資金，エネルギーなどのことである。現実的なスケジュールを考えるためにはこのような資源の制約を考慮する必要があり，またそれ自体が目的関数となることも多い。資源はその属性によって離散量と連続量または再生可能か再生不可能という2つの方法で分類される。離散量と連続量という分類は資源の単位の属性によるものである。その仕事を処理するに当たり必要とされる資源量である。仕事の分割の可否は仕事の処理を分割して行えるかどうかで、時間分割処理のように、何個かの連続しない時間区間で処理ができるかである。突然重要な仕事が入ったから学生とのゼミをやめさせられてあとで続きをする（実際は忘れることが多いですが）とかである。当然分けることが可能である方が以下述べるスケジュールの基準をよくする。最早開始可能時間は、その仕事の処理が開始できるもっとも早い時間である。実際、一般には最初から、全ての仕事が来ているわけでは途中から入ってくるのが自然である。また、スケジュールの基準としては、以下のような目的関数がよく使われる。
各仕事 J_i の完了時間の C_i に対する何らかのコスト関数 $f(C_i)$ で評価される場合が多い。このような関数 f の要素として処理時間 p_i だけではなく，r_i，納期 d_i，重み w_i などがよく用いられる。今ある実行可能なスケジュール π が存在し，仕事 J_i の処理の完了時間が C_i であるとする。そのとき以下のように表わされる2種類の目的関数が考えられる。

$f_{\max}(\pi) = \max\{f_i(C_i) \mid i = 1, \cdots, n\}$ (1)

$f_{sum}(\pi) = \sum_{i=1}^{n} f_i(C_i)$ (2)

これらの形で最もよく使われる目的関数は以下のものがある。

最大完了時間すべての仕事が終了する時間を示すものであり，通常 C_{max} で示され、以下の式で表される。

$$C_{\max} = \max\{C_i \mid i = 1, \cdots, n\}$$

総滞留時間

各仕事が処理を含めてどれだけ完了待ちをしたかを（**滞留時間**）といい、完了時間ー最早開始時間である）の総和であり、通常 F_{sum} で示し，以下の式で表される。

$F_{sum}=\sum_{i=1}^{n} C_i$

重み付き総滞留時間滞留時間に対し各仕事 J_i に重み w_i が与えられているものであり，F_{wsum} で示され、以下の式で表される。

$F_{wsum}=\sum_{i=1}^{n} w_i C_i$

ここで w_i は各仕事に定義された重みである。

また各仕事 J_i に対し納期 d_i が設定されているとき，各仕事に対し以下を定義する。

納期遅れ (Tardiness)

L_i は納期よりどれだけ遅れて仕事の処理が完了するかを示し，T_i として以下の式で表される。

$T_i=\max\{0, C_i-d_i\}$

また納期よりどれだけ早く終わるかも問題とされることもあり，E_i として以下の式で表される。

$E_i=\max\{0, d_i-C_i\}$

納期ずれ

各仕事が納期に対し実際の完了時間がどれだけずれているか示すものである。

$L_i=\max\{0, C_i-d_i\}$

これより以下の目的関数が考えられる。

納期ずれコスト

仕事の処理の完了時刻と納期のずれにコストがかかる場合の総和を示し，以下の式で表される。

$f_{et\_cost}=\sum_{i=1}^{n}(\alpha_i E_i+\beta_i T_i)$

ここで α_i, β_i はそれぞれ各仕事に定義された納期ずれに対する重みである。

納期遅れ仕事数

仕事の処理の完了時刻が納期より遅れてしまう仕事の総数で L_{number} で通常示され，以下の式で表される。

$L_{number}=|\{J_i \mid L_i>0\}|$

最大納期ずれ納期ずれの最大値のことで通常 L_{max} で示され，以下のように表される。

$L_{max}=\max_{1\leq i\leq n} L_i$

これらの目的関数は一般に同時に最適化できず，むしろこれらのトレードオフが問題となる。例えば最大完了時間や総滞留時間は工場の生産ライン，すなわち生産者側の要求であり，納期遅れ仕事数や最大納期遅れは顧客側の要求である。生産現場においてはこのように異なる立場からの要求が複数あり，これらにバランス良く応えていくことが必要である。

このような立場からは多目的スケジューリングが必要となる。スケジューリング問題の研究についてはもっともっとあるが、ここではこの辺りにとどめ、ファジィスケジューリングの紹介の所で、また必要な分は補っていく。

4. ネットワークとは

まず、**グラフ** (graph)$G(V, E)$ とは**点** (vertex, node) $v_1, v_2, \cdots, v_n$ の集合 V（誤解の恐れがなければ, 単に $V=\{1, 2, \cdots, n\}$ と表現することもある）と, V に属する2点を接続する**枝** (edge) の集合 E から構成されるものであり, v_i と v_j を接続する枝を e_{ij} と表記し, v_i と v_j は**隣接している** (adjacent) という。このとき, グラフ $G_s(V_s, E_s)$ ($V_s \subseteq V, E_s \subseteq E$) はグラフ G の**部分グラフ** (subgraph) と呼ばれる。

また, 枝に方向性が与えられていて, e_{ij} と e_{ji} が区別される場合, 特にそれらを**アーク** (arc) と呼ぶ。枝集合としてアーク集合が用いられるグラフを**有向グラフ** (directed graph) といい, **無向グラフ** (undirected graph) と区別する. 以降, 単にグラフというときは, 無向グラフを指すものとする。さらに, 同じ2点間に2本以上の枝が存在するものを**多重枝** (multiple edge), そのグラフを**多重グラフ** (multiple graph) という。2点を接続するのではなく, 両端点が同一の点となるような枝を**自己ループ** (self loop) といい, 自己ループや多重枝が存在しないようなグラフを**単純グラフ** (simple graph) と呼ぶ。

グラフ理論において中心的役割をなす**経路** (path) とは, ある2点間において, 複数の点を連続的に接続する枝の列である. すなわち, ある点から他の点への道を表す。また, 有向グラフにおける経路は, それを構成する全アークの方向が一致している場合に限る。同一点を2度以上通らない経路は**単純経路** (simple path), 最初と最後の点が等しい経路を**サイクル** (cycle), サイクル中に同一の点を含まないものを**閉路** (circuit) とそれぞれ呼んでいる。点 v_i から v_j への経路が存在すれば, v_i から v_j へ**到達可能である** (reachable) といい, グラフ内の任意の2点間が到達可能なものを**連結グラフ** (connected graph), そうでないものを**非連結グラフ** (disconnected graph) と呼ぶ。

閉路を含まない連結グラフは**木** (tree) と呼ばれ, いろいろなデータ構造を解析する基礎となり, 最小の連結グラフであるという点でも重要な概念である. 木に関する性質として, 次のようなものが知られている [16, 22]。

命題 4.1

木は少なくとも2つの終点（ここでは, 接続する枝が一つの点を指す）をもつ。

ある連結グラフ G の部分グラフについて, G と共通の点集合により構成される木を G の**スパニング・ツリー** (spanning tree) という。G のスパニング・ツリーは一意的ではなく複数存在し, その数については次の定理が成り立つ [16, 22]。

定理 4.1

単純グラフ $G(V, E)$ のスパニング・ツリーは高々$|V|^{|V|-2}=n^{n-2}$ 個である。

[スパニング・ツリー問題]

$G(V, E)$ の各枝 e_{ij} にコスト c_{ij} が付随していて, あるスパニング・ツリーのコストをそれに属する枝のコストの総和と定義した場合に, コストが最小となるスパニング・ツリーを見出す問題である（**図 4.1(b)** 参照; 各枝に付した数値はコストを表す。 通信網や送電網の構築において, 最も費用の少ない配線パターンを決定するような場合には, この問題

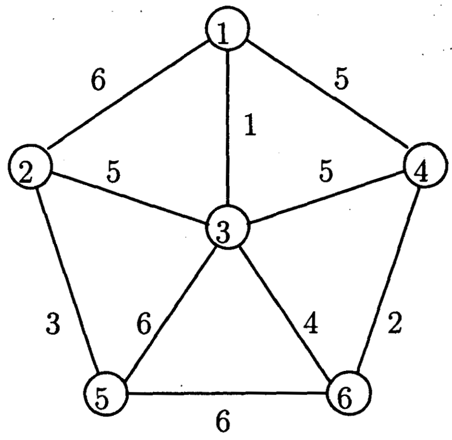

図 4.1(a) グラフ $G(V,\ E)$

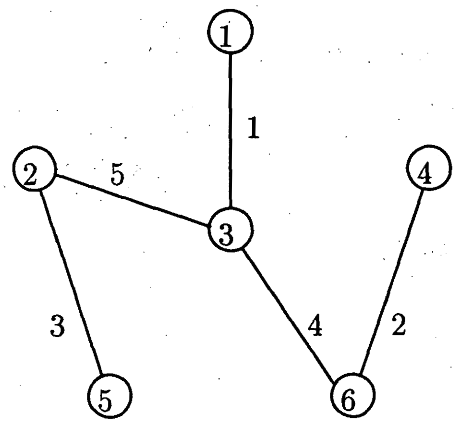

図 4.1(b) $G(V,\ E)$ のコスト最小スパニング・ツリー

として定式化することができる。

スパニング・ツリー問題を解く効率的アルゴリズムには, 以下のような **Prim, Kruskal** 等によるものが存在する。

Prim's Algorithm

Step 0 対象とするグラフを $G(V,\ E)$ とし, $S=\phi,\ U=\{1\}$ とする.

Step 1 $U=V$ であればグラフ（木）$T(V,\ S)$ をコスト最小のスパニング・ツリーとして終了. そうでなければ, **Step 2** へ。

Step 2 $e_{ij}\in E$ について, $e_{uv}=\min_{\{i,\ j\mid i\in U,\ j\in V\setminus U\}} e_{ij}$ とした後,

$$S=S\cup\{e_{uv}\}$$
$$U=U\cup\{v\}$$

とし, **Step 1** へ。

Prim のアルゴリズムは $O(n^2)$ でコスト最小のスパニング・ツリーを見出し [1, 48], **図 4.1(a)** に適用すれば, **図 4.1(b)** のような結果を得る。

Kruskal's Algorithm

Step 0 対象とするグラフを $G(V,\ E)$ とし, $k=1,\ S=\phi$ とする。$e_{ij}\in E$ をソートし, それらを

$$e_1\leqq e_2\leqq\cdots\leqq e_m$$

とする。

Step 1 グラフ $G'(V,\ S\cup\{e_k\})$ が閉路を含むならば, **Step 2** へ. そうでなければ, $S=S\cup\{e_k\}$ とせよ。もし, $|S|=n-1$ なら, グラフ（木）$T(V,\ S)$ をコスト最小のスパニング・ツリーとして終了。そうでなければ **Step 2** へ。

Step 2 $k=k+1$ として, **Step 1** へ。

Kruskal のアルゴリズムは $O(m \log n)$ でコスト最小のスパニング・ツリーを見出し [1, 40], **図 4.1(a)** に適用すれば, やはり**図 4.1(b)** のような結果を得る。

[最短経路問題]

有向グラフ $G(V, A)$ を考え, 各アーク a_{ij} には数値的長さ D_{ij} が与えられているとする. このとき, ある点からある点への経路のうち, その長さが最短となるものを求める問題を**最短経路問題** (shortest path problem) という。この問題はアークが道路に対応し, その長さを道路距離とすれば, 現実的な意味で最短経路を求めることになる。

最短経路問題には特定の点から他の点への最短経路を求める場合と, 任意の 2 点間の最短経路を求める場合が考えられるが, ここでは前者に限定して紹介する。なお, 後者の場合を扱ったモデルとしては **Floyd-Warshall** 法 [22] 等がある. また, アークの長さが負の場合（**Bellman-Ford** 等）も考えられるが, 非負の場合についてのみ言及する。

いま, 特定の点を v_1 とし, アークが存在しない 2 点 i, j 間に対して $D_{ij} = \infty$ とすれば, 点 v_1 から他の点 v_j $(j = 1, 2, \cdots, n)$ への経路長 D_j は, 次の非線形連立方程式を解くことにより求められる。

$$D_j = \begin{cases} 0 & (j = 1) \\ \min\limits_{i \neq j} \{D_i + D_{ij}\} & (j \neq 1) \end{cases} \tag{21}$$

この方程式は **Bellman** の方程式と呼ばれ, v_1 から v_j への有向経路が存在すれば, 唯一の有限解が得られる。しかしながら, **Bellman** の方程式自身は, どのように解けば良いかを示しておらず, ネットワーク上の問題として効率的解法が必要となる。その代表的なものとして, **Dijkstra** 法が知られている。

Dijkstra's Algorithm

Step 0 仮ラベル l_i $(i = 1, 2, \cdots, n)$ を以下のように設定する。ただし, v_1 は既に永久ラベル l_1^* を有するものとする。

$$l_i = \begin{cases} D_{1i} & (a_{1i} \in A) \\ \infty & (a_{1i} \notin) \end{cases}$$

さらに,

$$S(i) = \{\text{ 点 } v_i \text{に隣接する点 }\} \quad (i = 2, 3, \cdots, n-1)$$

とする.

Step 1 $l_k = \min\limits_i l_i$ なる k を選択する。l_k を永久ラベル l_k^* とする. 永久ラベルを持っていない点が無ければ終了。

Step 2 $i \in S(k)$ について

$$l_i = \min(l_i, l_k^* + D_{ki})$$

とし, **Step 1** へ.

実際に経路を特定するには, **Step 2** で l_i が変更される毎に直前の点 v_k を記憶しておけば良い。また, **Dijkstra** のアルゴリズムは $O(n^2)$ で最短経路を見出す [10, 16, 22].

[最大流問題]

多くの現実的な物の流れを扱う問題や, 他の組合せ最適化問題が**ネットワーク流量問題** (network flow problem) として定式化される．例えば, ネットワーク中の各点を供給点, 中継点, 需要点に対応させる物流問題などが挙げられる．このような問題の解法を考える際に, 重要かつ基本となる問題として**最大流問題** (maximum flow problem) がある。

有向グラフ $G(V, A)$ があり, 各アーク a_{ij} には i から j の方向に流すことのできる物質の最大量として**容量** (capacity)c_{ij} が与えられているとする。このとき, ある点 v_s から他のある点 v_t へ流すことのできる最大流量を求める問題を考える。v_s は**ソース** (source), v_t は**シンク** (sink) と呼ばれる．アーク a_{ij} を流れる流量を f_{ij} とすると, すべてのアークについて

$$0 \leqq f_{ij} \leqq c_{ij} \tag{22}$$

であり, v_s, v_t を除いて,

$$\sum_j f_{ji} = \sum_j f_{ij} \tag{23}$$

が成立するものとする。すなわち, ある点において物質が停留することはなく, 流入量と流出量は等しいという**Kirchhoff の流量保存則**が成り立つ。また, ソースとシンクについては

$$\begin{cases} \sum_j f_{sj} - \sum_j f_{js} = v \\ \sum_j f_{tj} - \sum_j f_{jt} = -v \end{cases} \tag{24}$$

である。

このとき, 最大流問題は次のように表すことができる。

$$v \rightarrow \textbf{最大化}$$

$$\text{制約条件} \ \textstyle\sum_j f_{sj} - \sum_j f_{js} = v, \sum_j f_{tj} - \sum_j f_{jt} = -v$$

$$\sum_j f_{ji} = \sum_j f_{ij} \quad (i \neq s,\ t), 0 \leqq f_{ij} \leqq c_{ij}$$

上記の最大流問題の最適解, すなわち最大流量, およびその流量パターンを求める際に有用な**最大流-最小切断定理**を示す。点集合 V を部分集合 $X\,(v_s \in X)$ とその補集合 $\overline{X}\,(v_t \in \overline{X})$ に分割したとき, X と $\overline{X}$ の各々に属する点間の全アークにより, **切断集合** (cut set)$(X, \overline{X})$ が形成される．この切断集合に対して, 以下のように容量 $C(X, \overline{X})$ を定義する。

$$C(X, \overline{X}) = \sum_{\{i,\, j | v_i \in X,\, v_j \in \overline{X}\}} c_{ij} \tag{25}$$

定理 4.2(最大流-最小切断定理)

整数のアーク容量をもつ任意のネットワークに対して, ソースからシンクへの最大流量は, ソースとシンクを分離する最小切断集合の容量に等しい。

直感的に, ネットワークに流すことのできる最大流量は各アークの容量による制約を受け, 必ずどこかネックとなる部分が生じるはずであり, さらにそのネックとなる部分のアーク容量の和が最大流量と等しくなると思われる。この定理に基づき, 最大流問題を解くための効率的アルゴリズムが開発されてきた [17, 22]。あといろいろと無数に述べる点があるが紙面の都合でファジィ組合せ最適化に必要なこと+アルファでの準備を行ってきた。

5. ファジィ組合せ最適化問題

5.1. ファジィスケジューリング

これまでに様々なスケジューリング問題が提案され，その最適スケジュールについて研究がなされてきたが，それらのほとんどはクリスプなデータを取り扱っている。しかし，仕事の処理時間は実際は実行してみないと分からない。また，処理工程が機械だけではなく人間の作業を含むような場合，やはり作業時間は一定でない上，機械においても突発的な不調が生じるかもしれない。このようなことから，処理時間があいまいなもの，すなわちファジィ化したモデルを考える意味がでてくる。

納期についても、納期は仕事の締め切りと考えれば，きっちりと守られなければならない期日である場合も，単なる目安であり、すこしであればそれに遅れてもよいこともしばしば見受けられる。あるいは納期を遅れることを見越して，ある程度余裕をもって納期を設定している節もあるだろうから，それほど絶対的なものと考える必要がないのかもしれない。抱えている仕事が沢山あって、私のようにすべての仕事を納期までに完了することが無理な状況が存在するため，納期に遅れることは前提で仕方がないとして，その遅れ具合をどの程度に抑えるか，つまり遅れに対する満足度をスケジューリングの基準とすることもある。さらには、ある仕事を処理するためには，その前段階として別の仕事が完了している必要がある場合もあり，仕事間の関係を先行関係と言う。このときも状況によっては完全に前段階の処理が完了していなくても次の処理に移れることもある。また,「終了している必要がある」ではなく「終了していることが望ましい」というような緩和された先行関係が設定されるときもある。あるいは先行の順番そのものも理由がつけばかえることができる場合もある。

以上のような観点から、ここ数年，一機械は勿論，複数機械でショップ型等種々多様なモデルについてファジィスケジューリング問題が研究されてきた（例えば [24])。まず一機械上でのモデルについて幾つか紹介する。

仕事の数は n 個とし，仕事を表す記号として J_i $(i=1,\cdots,n)$，また各仕事のクリスプな場合の納期，処理時間，完了時間をそれぞれ d_i, p_i, c_i を用いて表す。

ファジィ納期を考慮したスケジューリング問題

処理時間は通常の確定的な時間で，納期がファジィ，つまり完了時間に対して満足度がメンバシップ関数によって与えられる場合を考え，満足度の最小値が最も大きくなるようなスケジュールを求める [26]。いま，そのメンバシップ関数を次のように設定する（図 5.1 参照）.

$$\mu_{D_i}(x)=\begin{cases} 1 & (0\leqq x<d_i) \\ \dfrac{d_i+\theta_i-x}{\theta_i} & (d_i\leqq x<d_i+\theta_i) \\ 0 & (d_i+\theta_i\leqq x)\end{cases} \tag{26}$$

各仕事の完了時間 c_i に対してその満足度が $\mu_{D_i}(c_i)$ で与えられ，全仕事間での最小値を t とすれば,

$$\mu_{D_i}(c_i)\leqq t \Leftrightarrow c_i\leqq d_i+\theta_i(1-t)$$

がすべての i について成立する。

この不等式は完了時間が $d_i + \theta_i(1-t)$ 以内であれば良いことを示しているので，この値を通常納期と考えEDDルール（納期の早い順に仕事をスケジュールする）[28] を適用すれば最適スケジュールは求まるが，θ_i と t の値によってEDDルールに基づくスケジュールは変化する。ただし，t の値域 [0, 1] は，対応するスケジュールが不変であるような幾つかの区間に分割することができ，それらを

$$[0,t_1),\ [t_1,t_2),\ \cdots,\ [t_l,t_{l+1}),\ \cdots,\ [t_m,1]$$

とする。通常，納期が遅くなるにつれて満足度は減少するので，ある値 t_{l+1} に対応する $d_i + \theta_i(1-t)$ を納期としてすべての仕事を完了することはできないが，$[t_l,t_{l+1})$ 内の値に対してなら完了できる l が存在する筈である。その時のスケジュールが最適スケジュールを与える。

ファジィ処理時間を考慮したスケジューリング問題

ここでは処理時間がファジィ，納期がクリスプである幾つかのモデルを紹介する。

(最大納期遅れ最小化)

最大納期遅れを最小にするようなスケジュールを求めるには納期の早いものの順，すなわちEDDルールに基づいてスケジュールを生成すればよいことが知られている．これはファジィ処理時間ではない，通常のクリスプな処理時間の場合と同じスケジュールが得られることになる。

(納期遅れ仕事最小化)

Mooreは [47] で納期遅れ仕事の数を最小にするスケジュールが以下のようなアルゴリズムで求められることを示した。

Step 1 仕事を処理時間の小さいものの順にソートし，現在のシーケンスを $(J_{i1}, J_{i2}, \cdots, J_{in})$ とする $(p_{i1} \leqq \cdots \leqq p_{in})$.

Step 2 現在のシーケンス中，最も若い納期遅れの仕事を J_{iq} として **Step 3** へ。
そのような仕事が存在しなければ，現在のシーケンス中の仕事をEDDルールに基づき並べ替え，最適スケジュールとして終了。

Step 3 仕事 $J_{i1}, \cdots, J_{iq}$ をEDDルールに基づき並べ替え，

$$J_{l1},\ \cdots,\ J_{lq},\ J_{i\,q+1},\ \cdots,\ J_{in}$$

を構成する。

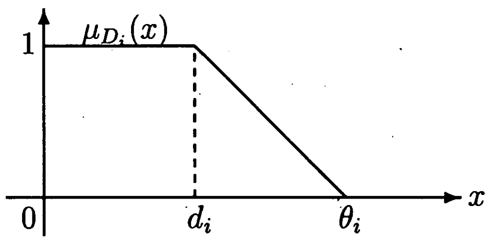

図5.1　ファジィ納期　$\mu_{D_i}(x)$

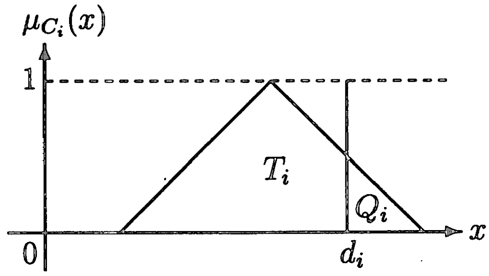

図 5.2　完了時間 C_i に対する T_i と Q_i

1. もし，部分シーケンス $J_{l1}, \cdots, J_{lq}$ 中のすべての仕事が納期遅れでないなら，上の列を現在のシーケンスとして Step 2 へ。
2. そうでなければ $J_{l1}, \cdots, J_{lq}, J_{i\,q+1}, \cdots, J_{in}$ から J_{lq} を削除し，残った列を現在のシーケンスとして Step 2 へ。

ここでは，そのファジィ概念化モデルを紹介する。

各仕事の処理時間は次のようなメンバシップ関数で制限される正の L-R ファジィ数を $P_i = (m_i,\ \alpha,\ \beta)_{LR}$ であるとする．

$$\mu_{P_i}(x) = \begin{cases} L\left(\dfrac{m_i - x}{\alpha}\right) & (x \leqq m_i) \\ R\left(\dfrac{x - m_i}{\beta}\right) & (x \geqq m_i) \end{cases} \tag{27}$$

ただし，$L,\ R$ は $L(0) = R(0) = 1$ で，$L,\ R : [0,\ +\infty) \to [0,\ 1]$ なる狭義減少関数とする．

処理時間がファジィ数であるため，先行仕事の拡張和として定義される各仕事の完了時間 C_i も L-R ファジィ数となる。完了時間がファジィ数であるので納期遅れの定義をファジィ概念化する必要があり，以下の通りとする。

λ-S 納期遅れ

J_i について $\dfrac{Q_i}{T_i} \geqq \lambda\ (0 < \lambda \leqq 1)$ が成立するとき，J_i は λ-S 納期遅れである。

ただし，T_i は完了時間のメンバシップ関数と時間軸とで囲まれる部分の面積，Q_i は T_i から納期直線 d_i によって切り取られる部分の面積である（図 5.2　参照)。

このモデルに対する最適スケジュールは，オリジナルの Moore モデルにおける諸定理が本モデルの設定においても成り立つことから，Moore のアルゴリズムにおける仕事の処理時間をメンバシップ関数のセンター m_i，納期遅れを λ-S 納期遅れと変更することで得られる。詳細は [47] を参照されたい。

(ファジィ処理時間，ファジィ納期を考慮したスケジューリング問題)

ここでは処理時間，納期が共にファジィの場合のモデルを紹介する。

J_i の納期は以下のようなメンバシップ関数をもつファジィ納期 D_i とする $(d_i > 0)$。

$$\mu_{D_i}(x) = \begin{cases} 1 & (0 \leqq x \leqq d_i) \\ D(x - d_i) & (d_i \leqq x < d_i + \theta) \\ 0 & (d_i + \theta \leqq x) \end{cases} \tag{28}$$

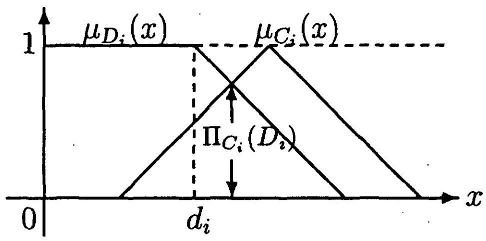

図 5.3　可能性測度 $\Pi_{C_i}(D_i)$

ただし，$D(x)$ は $D(0)=1$, $D: R^+ \to [0,1]$ なる狭義減少関数（線形である必要はない），$D(\theta)=0$ である。

D_i はその性格上，“仕事 J_i をだいたい d_i までに完了したい．”というファジィ目標と同一視することができる。仕事の処理時間は正の L-R ファジィ数 $P_i=(m_i, \alpha_i, \beta_i)_{LR}$ であり，完了時間 C_i は拡張加算により定義されるものとする。完了時間もまた正の L-R ファジィ数となるので，ファジィ納期（ファジィ目標）が満足される可能性を可能性測度 $\Pi_{C_i}(D_i)$ により表現することができる（**図 5.3**　参照)。

$$\Pi_{C_i}(D_i) = \sup_x \min\{\mu_{C_i}(x),\ \mu_{D_i}(x)\} \tag{29}$$

以上より，全スケジュールの集合を K として，最大納期遅れ最小化に対応した次のような問題を考える．

$$\text{目的関数}\ \min_{J_i \in K_l} \Pi_{C_i}(D_i) \ \to\ \text{最大化}\ (K_l \in K) \tag{30}$$

(ファジィ先行関係を考えたモデル)

いくつかの仕事があるとき、ある仕事をすませないと次の仕事に入れないということがよくある。こういう制約を先行関係という。通常のスケジューリング問題では、仕事間に先行関係があるかあるいはないかのどちらかに峻別してしまっている。また先の仕事が完了しないと次の仕事に入れないと仮定している。

しかし、実際の製造現場をみると、先行関係があるかないかのどちらかに限定できないような状況がある。部品の調達状況、費用、もしくはどの顧客に対する納品を優先するか、さらに「この仕事を先に処理する方が望ましい」とか「必ずこれを先に処理しなければならない」など、スケジュールを組む際に先行関係を状況に応じてフレキシブルに考える必要が生じる。

また、1つ1つの先行関係をよく観察すると、ときには、前の作業が完全に終らなくても次の作業に入れることがある。このように、先行関係を過度に厳密に考えると、不都合が生じることが多いので、柔軟に対応した方がよいといえる。こういう先行関係をファジィ先行関係という。

先行関係は2つの仕事 J_i, J_j の間に、J_i が J_j に先行するとき、$J_i < J_j$ と表し、この先行関係を $\mu_{ij}=1, \mu_{ji}=0$ によって表す。ファジィ先行関係では、μ_{ij}, μ_{ji} を0と1の間の数値で表す。ファジィ先行関係は、全仕事上のファジィ関係であり、ファジィグラフによって表すことができる。ただし、先行関係では、2つの仕事が独立で「先行関係がない」ということは、どちらを先に処理してもよい、ということを意味するので、$\mu_{ij}=1, \mu_{ji}=1$

と約束する。そこで、最大納期ずれの最小化と先行関係の尊重という２つの目的をもつスケジューリング問題を考える。ここで、先行関係については、各仕事間に定めたファジィ先行関係から求めた満足度の最小値で測ることにする。通常の先行関係を考慮したスケジュール問題について解法を説明する。

まず、仕事 J_i が先行する仕事の集合を $T_i = \{J_k | J_i < J_k, k \in \{1, 2, \ldots, n\}\}$ で示す。ただし、記号 $<$ はこの記号の前の仕事が後の仕事に先行して処理されなければならないことを示す。Lawler and Moore [45] に従って次のように修正納期 d_i' を計算する。

$d_i' = min\{d_i, min\{d_j | J_j \in T_i\}\}, i = 1, 2, \ldots, n$

d_i' の非減少順に先行関係を考慮しながら、仕事を処理すれば、最適スケジュールがえられる。

仕事 J_j と J_i についてそのファジィ先行関係を表すメンバシップのグレード μ_{ij} に関して

1. $\mu_{ij} = 1, \mu_{ji} = 0$ はクリスプな先行関係があることを表し、
2. $\mu_{ij} = 1, \mu_{ji} = 1$ は、２つの仕事のうちどちらを先に処理してもよいことを表し、
3. $\mu_{ij} = 1, \mu_{ji} = \alpha\ (0 < \alpha < 1)$ は、ファジィ先行関係があることを示す。ここで、μ_{ij} と μ_{ji} のいずれかは必ず１、すなわちどちらかは相手より先行するものと仮定する。

さて、仕事 J_i を仕事 J_j に先行して処理したときの満足度は μ_{ij} であると定める。いま、スケジュール π に対して第 k 番目に処理する仕事を $\pi(k)$ で示し、このスケジュールの最大納期ずれを L^{π}_{max}、ファジィ先行関係の満足度の最小値を μ^{π}_{min} で示す。また、スケジュールベクトル $v^{\pi} = (L^{\pi}_{max}, \mu^{\pi}_{min})$ を考え、非劣スケジュールを以下のように定義する。

非劣スケジュール

２つのスケジュール π_1, π_2 に関して, π_1 が π_2 に優越するというはこれらのスケジュールベクトル $v^{\pi_1} = (v_1^{\pi_1}, v_2^{\pi_1}), v^{\pi_2} = (v_1^{\pi_2}, v_2^{\pi_2})$ について次の関係が成り立つ時である。

$$v_1^{\pi_1} \leqq v_1^{\pi_2}, v_2^{\pi_1} \geqq v_2^{\pi_2}$$

π に優越するスケジュールが存在しないとき、π は非劣スケジュールであるという。

一般に２目的以上のスケジューリング問題では全ての目的関数を最適化するスケジュールは存在しないので、非劣スケジュールを求めることになる。この問題は以下のようにファジィ先行関係の満足度を順に１から下げながら非劣スケジュールを求める手法で解く事ができる。まず、μ_{ij} を降べきの順に並べ、この結果を

$$\mu^0 \equiv 1 > \mu^1 > \mu^2 > \cdots > \mu^k \geqq 0$$

とする。ここで k は異なった μ_{ij} の数である。次に先行関係グラフ PG を

$PG = [N; A]$, N は仕事 J_i に対応する点の集合、枝集合 A は $J_i < J_j$ に対応する (J_i, J_j) からなる　として定義する。

この先行グラフの列 $PG^{\ell}, \ell = 1, 2, \ldots, k$ を以下のように作る。

$$A^0 \equiv \{(J_i, J_j) | \mu_{ij} = \mu^0, \mu_{ji} \fallingdotseq \mu^0\}, \bar{A}^{\ell} \equiv \{(J_i, J_j) | \mu_{ij} = \mu^0, \mu_{ji} = \mu^{\ell}\}$$
$$A^{\ell} = A^{\ell-1} - \bar{A}^{\ell}, PG^{\ell} = [N; A^{\ell}], \ell = 1, 2, \ldots, k$$

Step 0 先行関係グラフ $PG^0 = [N; A^0]$ に基づく先行関係のもとで、最大納期ずれ最小化問題を解き、その最適値 L^0_{max} 及び最適スケジュール π^0 を求めよ。$DV = \{(L^0_{max}, 1)\}, DS = \{\pi^0\}, \ell = 1$ とおいてステップ1へ。

Step 1 先行関係 PG^ℓ に基づく先行関係のもとで、最大納期ずれ最小化問題を解き、その最適値 L^ℓ_{max} 及び最適スケジュール π^ℓ を求めよ。対応するスケジュールベクトル $v^\ell = (L^\ell_{max}, \mu^\ell_{min})$ をつくれ。ここで $\mu^\ell_{min} = min\{\mu_{\pi^\ell(i)\pi^\ell(j)} | i, j = 1, 2, \ldots, n, i < j\}$ である。このスケジュールが、π^ℓ が DS に属するベクトルに優越されているか、このスケジュールに対するスケジュールベクトルがすでに DV にあれば、ステップ2へ。そうでなければ $DV = DV \cup \{v^\ell\}, DS = DS \cup \{\pi^\ell\}$ と更新してステップ2へ。

Step 2 $\ell = \ell + 1$ と更新せよ。もし、$\ell = k + 1$ なら終了。DS が非劣スケジュールを与える。そうでなければ、ステップ1へ戻れ。

このアルゴリズムの詳細等は [24] を参照されたい。

[1 機械以外のファジィスケジューリング問題]

1機械問題に較べて2機械以上のファジィスケジューリング問題は少ない上、1機械で大体のファジィ概念が出てきたので、ここでは1つだけにとどめておく。詳しくは [50] の第3章「離散システムにおけるソフト最適化」を見られたい。D.Werra[59] はオープンショップでの各仕事の各部分作業が単位時間あたり資源 R の単位量を必要とするとき、各区間で使用可能なト-タル資源量制約のもとで、最大完了時間を最小にする問題を議論した。我々は仕事の実行可能時間のファジィ概念化と使えるト—タルの資源量をフレキシブルにしたファジィ資源量を導入し、次のような問題を考えた ([38])。

1. m 台の機械 $P_i, i = 1, 2, \ldots, m$ とこの機械で各部分作業が処理される n 個の仕事 $J_1, J_2, \cdots, J_n$ がある。
2. 各仕事 J_j は各機械 P_i で処理されるべき m 個の部分作業 T_{ij} からなり、T_{ij} の処理時間は単位時間とする。
3. 各仕事はオープンショップ型、すなわち各仕事の部分作業の処理順序は任意である。
4. 各部分作業の処理には単位資源量が必要である。
5. 各実行可能時間帯 $\tilde{u}_j$、すなわち処理開始時間 $\tilde{s}_j$（非負整数）と完了時間 $\tilde{e}_j$（非負整数）はフレキシブルで、次のような単峰なメンバシップ関数で示される満足度で表現される。

$$\mu_{\tilde{s}_j}(s_j) = \begin{cases} 0 & (s_j \leqq r_j) \\ m_j(s_j) & (r_j < s_j \leqq r_j + g_j) \\ 1 & (s_j \geqq r_j + g_j) \end{cases}$$

ここで s_j は整数値をとり、r_j, g_j は非負整数, m_j は単調非減少とする。

$$\mu_{\tilde{e}_j}(e_j) = \begin{cases} 1 & (e_j \leqq d_j) \\ k_j(e_j) & (d_j < e_j \leqq d_j + f_j) \\ 0 & (e_j \geqq d_j + f_j) \end{cases}$$

ここで e_j は整数値をとり、d_j, f_j は非負整数, k_j は単調非増加とする。また $r_j+g_j \leqq d_j$ とする。

6. 各区間 k で使用可能な資源量（非負整数と仮定）もフレキシブルで非増加階段関数で示される満足度が付随している。
7. 各区間の資源の余剰分は次の区間に繰り越すことができる。
8. 1から7までの条件のもとで、資源量、実行可能時間の条件を満たしながら、資源量の満足度の最小値、実行可能時間の満足度の最小値について、これらの最小値の最小値を最大にするスケジュールを求める。

解法は特別なネットワークを作り、最大フロー問題を利用するものであるが詳細は [38] を見られたい。このような関係のモデルとしては [37] がある。またその他幾つかのモデルが考えられている。

5.2. ファジィネットワークモデル

はじめにも述べたようにファジィ組合せ最適化の最初の大きな研究は Chanas 等によるファジィアーク容量の最大流問題への導入であるので、まずこの辺りからはじめよう。[6, 7] 参照。すなわち，少しぐらいならば容量を越えてもかまわない（だいたい〜以下の容量）ならば図 5.4 に示されるようなメンバシップ関数が，上限のみならず下限もある程度考慮したい（だいたい〜の容量）ときは図 5.5 に示されるような台形型メンバシップ関数がそれぞれ利用される。これらの関数は，アーク (i,j) を流れるフローの量 $f(i,j)$ について意思決定者（decision-maker; 以下 "DM" と略記）が主観的にどう思うかをファジィ容量のメンバシップ値 $\mu_{ij}(f(i,j))$ として与える。

この問題の目的は，通常の最大流問題の制約のもとで，

$$\min[\min\{\mu_{ij}(f(i,j)) \mid arc(i,j) \in \mathcal{A}\}, \mu_G(v)] \tag{31}$$

を最大にするフローを求めることで， 総流量 v に対する満足度 μ_G とアークの容量制約に対する帰属度 μ_{ij} の最小値の 2 つを同時に最大化する二目的関数になっている。上の (31) 式を $\mu_D(f_v)$ とおくと，$\max_f \mu_D(f_v) = \mu_D(\widehat{f_v})$ を与える $\widehat{f_v}$ のことを 極大フローと呼び，極大フローの中で最大の流量 v を与えるフローのことを 最適フロー f_v^* と定義する。

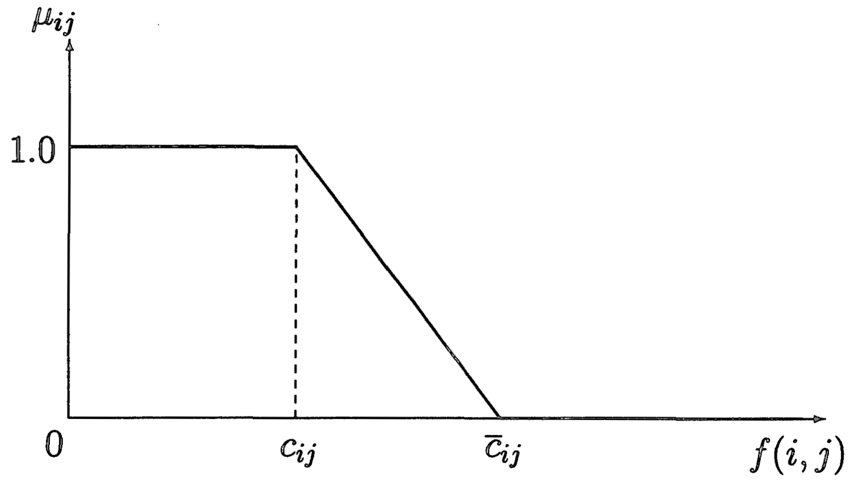

図 5.4　ファジィ容量

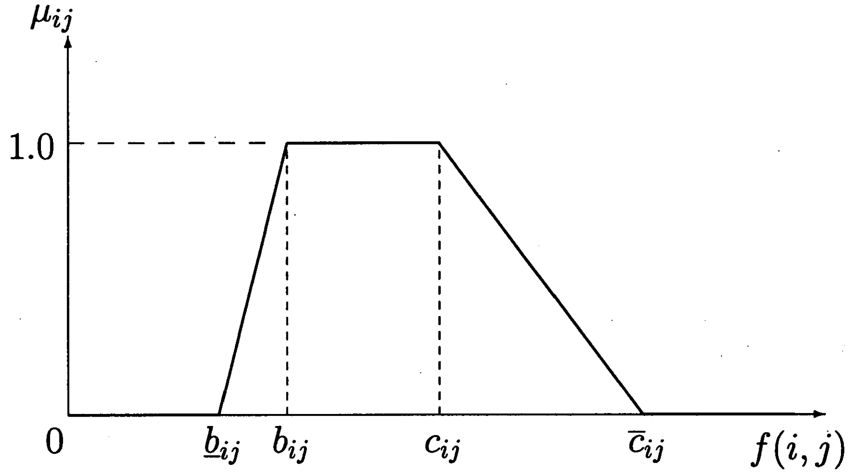

図 5.5　下限制約つきファジィ容量

このとき，前述の最大流-最小切断定理のファジィ版が Chanas 等によって与えられた。これはクリスプな最大流問題に対する前述の最大流-最小切断定理を含む拡張になっている。この証明およびその解法については，[6] を参照されたい。

[定理 5.1] 最大流－最小切断定理（ファジィ版）

f_v^* をソース s からシンク t への最適フローとするとき，以下の 2 つの条件を満たす切断 $(X,\overline{X})$ が存在する。

- アーク $(i,j) \in (X,\overline{X}) \Rightarrow \mu_{ij}(f(i,j)) = \mu_D(f_v^*)$,
- $i \in \overline{X}, j \in X \Rightarrow f(i,j) = 0.$

ファジィランダムスパニングツリー問題

$G=(N,E)$ を点集合 $N=\{v_1,v_2,\ldots,v_n\}$ と枝集合 $E=\{e_1,e_2,\ldots,e_m\} \subset N \times N$ からなる無向グラフとし, e_i にはコスト c_i が設定されているものとする。 G におけるスパニングツリー $T=T(N,S)$ は $S \subseteq E$ かつ閉路を含まない連結部分グラフである。

T は次のように 0-1 変数の $x_1,x_2,\ldots,x_m$ で表すことができる。

$$T \ : \ \begin{array}{ll} x_i = 1, & e_i \in S \\ x_i = 0, & e_i \notin S \end{array}$$

以下では X を $T(N,S)$ に対応する 0-1 ベクトルの集合とし, X をスパニングツリーの集合とみなす. このとき, 最小スパニングツリー問題は次のように定式化される。

$$\mathbf{P}_{s1} : \text{minimize} \quad \boldsymbol{cx} \tag{32}$$

$$\text{subject to} \quad \boldsymbol{x} \in X \tag{33}$$

ここで， $\boldsymbol{c}=(c_1,\ldots,c_m)$ $\boldsymbol{x}=(x_1,\ldots,x_m)^t$ であり，それぞれの c_i は次のメンバシップ関数 $\mu_{C_i(\omega)}$ で特性づけられるファジィランダム変数であるとする。

$$\mu_{C_i(\omega)}(c_i) = \max\left\{ L\left(\frac{c_i - d_i(\omega)}{\alpha_i}\right),\ 0\right\} \tag{34}$$

ただし，L は $\mathbf{R} \to [0,\ 1]$ である単調減少連続関数で $L(x) = L(-x)\ \ \forall x \in \mathbf{R}$ かつ $L(0) = 1$ を満たすものとする。また, $d_i(\omega)$ は平均 μ_i, 分散 σ_i^2 をもつ正規分布に従う互いに独立な確率変数とする。すなわち, ファジィ数において中心が確率変数になっている場合であり, $d_i(\omega)$ に伴って $\mu_{C_i(\omega)}(c_i)$ も確率的に変動する。

$y = \boldsymbol{c}\boldsymbol{x}$ とおくと，y は次のメンバシップ関数で特性づけられるファジィランダム変数 $Y(\omega)$ となる。

$$\mu_{Y(\omega)}(y) = \max\left\{L\left(\frac{y - \sum_{i=1}^{m} d_i(\omega)x_i}{\sum_{i=1}^{m}\alpha_i x_i}\right),\ 0\right\} \tag{35}$$

次に目的関数に関する目標値を “だいたい y_1 以下である” とし, それをメンバシップ関数 μ_G で特性づけられるファジィ集合 G で表す. ただし，μ_G は $0 \leqq y \leqq y_1$ で 1 であり，$y \geqq y_1$ において単調減少である連続関数とする。総コストを表すファジィランダム変数 $Y(\omega)$ のメンバシップ関数 $\mu_{Y(\omega)}(y)$ を可能性分布とみなすとき，$\mu_{Y(\omega)}(y)$ の下で G である可能性測度は次のように与えられる。

$$\Pi_{Y(\omega)}(G) \triangleq \sup_y \min\{\mu_{Y(\omega)}(y),\ \mu_G(y)\} \tag{36}$$

問題 $\mathbb{P}_{s1}$ から次の $\mathbb{P}_{s2}$ を考える。

$$\begin{aligned}
\mathbb{P}_{s2}\ :&\text{maximize}\ \ h && (37)\\
&\text{subject to}\ Pr\left(\Pi_{Y(\omega)}(G) \geqq h\right) \geqq \alpha && (38)\\
&\boldsymbol{x}\ \in X && (39)
\end{aligned}$$

ここで Pr は確率測度を表し, 確率レベル α は $1/2 < \alpha < 1$ を満たす確定値とする。$\Pi_{Y(\omega)}(G) \geqq h$ を変形すると

$$\sum_{i=1}^{m} d_i(\omega)x_i - L^*(h)\sum_{i=1}^{m}\alpha xi \leqq \mu_G^*(h)$$

となり（詳細は [30, 33] 参照）よって，$\mathbb{P}_{s2}$ は次の $\mathbb{P}_{s3}$ になる。

$$\begin{aligned}
\mathbb{P}_{s3}\ :&\text{maximize}\ \ h && (40)\\
&\text{subject to}\quad Pr\left(\sum_{i=1}^{m}\{d_i(\omega) - L^*(h)\alpha_i\}x_i \leqq \mu_G^*(h)\right) \geqq \alpha && (41)\\
&\boldsymbol{x}\ \in X && (42)
\end{aligned}$$

ただし, $L^*(\cdot)$ および $\mu_G^*(\cdot)$ は擬逆関数であり次のように表される.

$$L^*(h) = \begin{cases} \sup\{r|L(r) > h, r \geqq 0\} & (0 < h \leqq 1)\\ \infty & (h = 0)\end{cases} \tag{43}$$

$$\mu_G^*(h) = \sup\{r|\mu_G(r) \geqq h\} \tag{44}$$

正規分布の性質から

$$\frac{\sum_{i=1}^{m} d_j(\omega)x_j - \sum_{i=1}^{m} \mu_i x_i}{\sqrt{\sum_{i=1}^{m} \sigma_i^2 x_i}} \tag{45}$$

は標準正規分布 N(0,1) に従う確率変数となる。K_α を α 分位点とするともとの確率的条件は次の等価確定条件に変換される.

$$\sum_{i=1}^{m} \{\mu_i - L^*(h)\alpha_i\} x_i + K_\alpha \sqrt{\sum_{i=1}^{m} \sigma_i^2 x_i} - \mu_G^*(h) \leqq 0 \tag{46}$$

となる. ゆえに $\mathbf{P}_{s2}$ は次の確定問題 $\mathbf{P}_s$ と等価になる。

$$\mathbf{P}_s \text{ :maximize } \; h \tag{47}$$

$$\text{subject to } \; \sum_{i=1}^{m} \{\mu_i - L^*(h)\alpha_i\} x_i + K_\alpha \sqrt{\sum_{i=1}^{m} \sigma_i^2 x_i} \leqq \mu_G^*(h) \tag{48}$$

$$\boldsymbol{x} \in X \tag{49}$$

解法の便宜上, $L^*(h) = q$ とおいた次の問題を考える。

$$\mathbf{P}_s' \text{ :minimize } \; q \tag{50}$$

$$\text{subject to } \; \sum_{i=1}^{m} \{\mu_i - q\alpha_i\} x_i + K_\alpha \sqrt{\sum_{i=1}^{m} \sigma_i^2 x_i} \leqq \mu_G^*(L(q)) \tag{51}$$

$$\boldsymbol{x} \in X \tag{52}$$

$L(\cdot)$ の単調非増加性より $L(q)$ を最大化するためには q を最小化すればよいことになる。問題 $\mathbf{P}_s'$ を解くために次の部分問題を導入する。

$$\mathbf{P}_s^q \text{ :minimize } \; \sum_{i=1}^{m} \{\mu_i - q\alpha_i\} x_i + K_\alpha \sqrt{\sum_{i=1}^{m} \sigma_i^2 x_i} \tag{53}$$

$$\text{subject to } \; \boldsymbol{x} \in X \tag{54}$$

問題 $\mathbf{P}_s^q$ の最適解を $\boldsymbol{x}(q)$ とし, 最適値を $z(\boldsymbol{x}, q)$ とする。 また, $\mathbf{P}_s$ の最適解を $\boldsymbol{x}^*$ とし, 最適値を q^* とする。このとき, 部分問題 $\mathbf{P}_s^q$ と元の問題 $\mathbf{P}_s$ に対して次の補題が成り立つ。

補助定理 5.1

問題 $\mathbf{P}_s^q$ の最適解が $\mathbf{P}_s$ の最適解である必要十分条件は $\mathbf{P}_s^q$ における最適解 $\boldsymbol{x}(q)$ に対して $z(\boldsymbol{x}^*,\ q^*) = \mu_G^*(L(q^*))$ が成り立つことである。

問題 $\mathbf{P}_s^q$ は目的関数に非線形の項を含んでいるが, この問題を解くために, 次のパラメータ λ を導入した問題を考える。

$$\mathbf{P}_s(q,\lambda) \text{ :minimize } \; \sum_{i=1}^{m} \left(\mu_i - q\alpha_i + \lambda K_\alpha \sigma_i^2\right) x_i \tag{55}$$

$$\text{subject to } \; \boldsymbol{x} \in X \tag{56}$$

q 一定のとき, $\mathbf{P}_s(q,\lambda)$ は一つのパラメータ λ をもつパラメトリック最小スパニングツリー問題となる。$\mathbf{P}_s$ の解法を考える. まず, 次の 3 目的問題を考える。

$$\mathbf{P}_{3opt}\text{ :minimize } \sum_{i=1}^{m} \mu_i x_i \tag{57}$$

$$\text{maximize } \sum_{i=1}^{m} \alpha_i x_i \tag{58}$$

$$\text{minimize } \sum_{i=1}^{m} K_\alpha \sigma_i^2 x_i \tag{59}$$

$$\text{subject to } \vec{x} \in X \tag{60}$$

任意の q,λ に対して $\mathbf{P}_s(q,\lambda)$ の最適解は $\mathbf{P}_{3opt}$ の非劣解集合に属する. また, Geetha ら [20] によって非劣解のうち目的空間において凸包の端点に相当するものだけが $\mathbf{P}_s$ の最適解の候補になることが示されている。$\mathbf{P}_{3opt}$ の非劣解集合を求めるアルゴリズムを考える。次に示すアルゴリズムは Geetha ら [20] の方法が基になっており, 伊藤 [27] の方法を 3 目的計画問題に拡張したものである。

非劣解集合を求めるアルゴリズム

手順 1 $k=1$ とする. $z_1^{(1)} = \min_{x\in X}\sum_{i=1}\mu_i x_i$ を求め, 対応する $z_2^{(1)} = \sum_{i=1}^m \alpha_i x_i$, $z_3^{(1)} = \sum_{i=1}^m K_\alpha \sigma_i^2 x_i$ を計算する。 $(z_1^{(1)}, z_2^{(1)}, z_3^{(1)})$ と対応するスパニングツリー $\boldsymbol{x}^{(1)}$ を保持する。 $k=2$ とし, 同様にして $z_2^{(2)} = max\{\sum_{i=1}^m \alpha_i x_i | \boldsymbol{x} \in X\}$ と対応する $z_1^{(2)} = \sum_{i=1}^m \mu_i x_i, z_3^{(2)} = \sum_{i=1}^m K_\alpha \sigma_i^2 x_i$ とその時の解 $\boldsymbol{x}^{(2)}$ を求める。$k=3$ とし, また同様に, $z_3^{(3)} = \min\{\sum_{i=1}^m K_\alpha \sigma_i^2 x_i | \boldsymbol{x} \in X\}$ と対応する $z_1^{(3)} = \sum_{i=1}^m \mu_i x_i$, $z_2^{(3)} = \sum_{i=1}^m K_\alpha \sigma_i^2 x_i$ を求め, その時の解 $\boldsymbol{x}^{(3)}$ を求める。集合 $DS=\{1,2,3\}$ を定義して手順 2 へ進む。

手順 2 $k=k+1$ とする。 ある $\boldsymbol{x}^{(s)}, \boldsymbol{x}^{(t)}, \boldsymbol{x}^{(u)}$ を選択し,

$$q_{(s,t,u)} = \frac{(z_1^{(t)} - z_1^{(u)})(z_3^{(s)} - z_3^{(t)}) - (z_1^{(s)} - z_1^{(t)})(z_3^{(t)} - z_3^{(u)})}{(z_2^{(t)} - z_2^{(u)})(z_3^{(s)} - z_3^{(t)}) - (z_2^{(s)} - z_2^{(t)})(z_3^{(t)} - z_3^{(u)})} \tag{61}$$

$$\lambda_{(s,t,u)} = \frac{(z_1^{(t)} - z_1^{(u)})(z_2^{(s)} - z_2^{(t)}) - (z_1^{(s)} - z_1^{(t)})(z_2^{(t)} - z_2^{(u)})}{(z_2^{(t)} - z_2^{(u)})(z_3^{(s)} - z_3^{(t)}) - (z_2^{(s)} - z_2^{(t)})(z_3^{(t)} - z_3^{(u)})} \tag{62}$$

を計算し, 次の問題を解く。

$$\mathbf{P}^{(s,t,u)}\text{ :minimize } \sum_{i=1}^{m}\left(\mu_i - q_{(s,t,u)}\alpha_i + \lambda_{(s,t,u)} K_\alpha \sigma_i^2\right) x_i \tag{63}$$

$$\text{subject to } \boldsymbol{x} \in X \tag{64}$$

この問題の最適解を $\boldsymbol{x}^{(k)}$ とする。$\boldsymbol{x}^{(s)}$ がこの問題の最適解ならば $\boldsymbol{x}^{(k)}$ を放棄し, $DS = DS\backslash\{(s,t,u)\}$ として手順 3 へ進む. そうでなければ

$$DS = DS \cup \{(s,t,k),(s,k,u),(k,t,u)\}\backslash\{(s,t,u)\}$$

として $\boldsymbol{x}^{(k)}$ を保持して手順 3 へ進む。

手順 3 $DS=\emptyset$ なら, 終了する。 そうでなければ手順 2 へ戻る。

上記のアルゴリズムは目的空間の凸包の端点に対応する非劣解を順に効率良く求めているが，この非劣解の個数は3次元におけるk-集合問題と深い関わりがある。k-集合問題とはユークリッド空間において，ある点集合をk個とそれ以外の点に分割ときの場合の数を求める問題であり，上記の問題においては，非劣解の個数がn個の点をm個と$n-m$個に分割するときの場合の数に対応している。$T_{MST}(n,m)$をスパニングツリーを求めるための計算時間とし，$N_{kset}(m)$をm-集合の個数の上界とするとき次の定理が成り立つ。

定理 5.2

上記のアルゴリズムによって非劣解集合は$O(T_{MST}\cdot N_{kset}(m))$の計算時間で求めることができる。

Dey [9] によってm-集合の個数の上界は$O(m^{8/3})$であることが示されており，[8, 60] などのアルゴリズムによってスパニングツリーを求めるならば非劣解を求めるアルゴリズムの計算時間は全体で$O(m^{(11/3)}\log n \log\log n)$となる。

求められた非劣解の個数を$n(s)$とする。$\boldsymbol{x}^{(k)}, i=1,\ldots,n(s)$に対して次の値を計算する。

$$
\begin{aligned}
a^{(k)} &= \sum_{i=1}^{m}\mu_i x_i^{(k)} + K_\alpha\sqrt{\sum_{i=1}^{m}\sigma_i^2 x_i^{(k)}} \\
b^{(k)} &= \sum_{i=1}^{m}\alpha_i x_i^{(k)}
\end{aligned}
$$

さらに次の関数を定義する。

$$
\begin{aligned}
y^{(k)}(q) &= a^{(k)} - qb^{(k)} \\
f(q) &= \min_k y^{(k)}(q)
\end{aligned}
$$

$y^{(k)}$がqの線形関数であるから$f(q)$はqに関して区分的線形かつ狭義減少であることがわかる。$f(q)$の区分点を求め，次のように非減少順に並べ替える。

$$
0 = q_0 < q_1 < \cdots < q_{n(p)} < q_{n(p)+1} = L^*(0)
$$

ここで$n(p)$は求めた区分点の個数を表す．また，$f(q_l)$, $l=0,\ldots,n(p)+1$を与えるyに対応する非劣解を$\boldsymbol{x}^{(l)}$, $l=0,\ldots,n(p)+1$として改めて番号付けする。求めたqの集合をQとし，LB, UBをそれぞれ求めたlの下限と上限とすると，$\mathbf{P}_s$を解くアルゴリズムは次のようになる。

アルゴリズム

手順 1 非劣解集合を求め，集合Qを求める。

手順 2 $LB\leftarrow 0$とし，$\mu_G(f(q_0)) \geq L(q_0)$ならば$\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{(0)}$として終了する。
そうでなければ手順3へ進む。

手順 3 $UB\leftarrow n(p)+1$とし，$\mu_G(f(q_{n(p)+1})) < L(q_{n(p)+1})$ならば終了する．そうでなければ手順4へ進む。

手順 4 $UB-LB=1$ならば$\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{(LB)}$として終了する。そうでなければ$k\leftarrow\lceil(LB+UB)/2\rceil$として手順5へ進む。

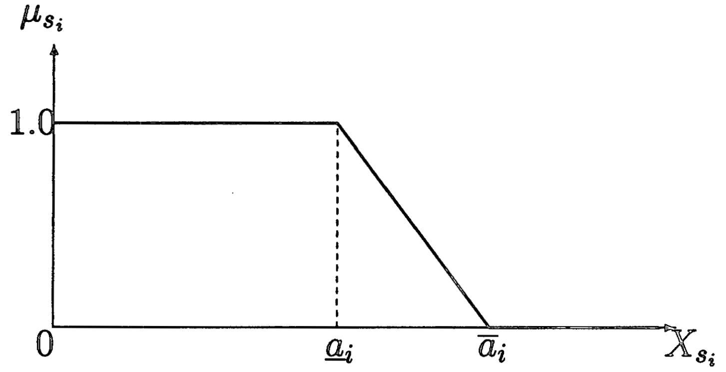

図 5.6 ファジィ供給量 $\tilde{F}_{s_i}$

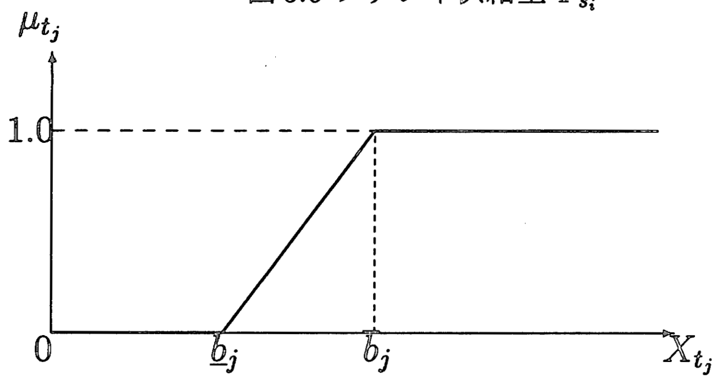

図 5.7 ファジィ需要量 $\tilde{G}_{t_j}$

手順 5 $\mu_G(f(q_k)) > L(q_k)$ ならば $UB \leftarrow k$ として手順 4 へ戻る。 $\mu_G(f(q_k)) < L(q_k)$ ならば $LB \leftarrow k$ として手順 4 へ戻る。$\mu_G(f(q_k)) = L(q_k)$ ならば $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{(k)}$ として終了する。

定理 6.3
上記のアルゴリズムによって P_s の最適解は多くとも $O(T_{MST} \cdot N_{kset}(m))$ の計算時間で求められる。

この他にもファジィランダムスパニング問題のモデルが幾つか考えられている（例えば [31] 等）。

ファジィ輸送問題
通常の輸送問題では，供給量 a_i と需要量 b_j について，供給量の合計値 $\sum_{i=1}^{m} a_i$ は需要量の合計値 $\sum_{j=1}^{n} b_j$ 以上でなければならない。しかし，現実にはこのような条件が満たされず、供給側に無理言って，供給量を a_i より増やしてもらうとか、需要側では受け取る量はできれば b_j 以上だが，少し我慢してもよいというような状況が考えられる。すなわち、足らない時は三方一両損の処世術である。このような観点から、自然状態では総供給量が総需要量を下回っている状況を考えて、図 5.6、図 5.7 で示されるファジィ供給量、ファジィ需要量を輸送問題に導入する。これらは供給ノード s_i，需要ノード t_j ごとに定められ，供給量 $(X_{s_i} = \sum_{t_j \in T} f(s_i, t_j),\ i = 1, \cdots, m)$，需要量 $(X_{t_j} = \sum_{s_i \in S} f(s_i, t_j),\ j = 1, \cdots, n)$ に対する満足度をメンバシップ関数 $\mu_{s_i}(X_{s_i})$, $\mu_{t_j}(X_{t_j})$ が示すので，$\underline{a}_i$ や $\overline{a}_i$ （$\underline{b}_j$ や $\overline{b}_j$）の値を調整することによって，現実的な供給量（需要量）を表すことができる。ただし，現実

的な意味から，$\sum_{s_i \in S} \overline{a}_i > \sum_{t_j \in T} \underline{b}_j$ と仮定する。.

$$
\begin{aligned}
FTP: \text{Maximize } & \min\{\mu_{s_i}(X_{s_i}), \mu_{t_j}(X_{t_j}) | s_i \in S, t_j \in T\}, && (65)\\
\text{subject to} & \\
& \sum_{s_i \in S} \sum_{t_j \in T} c_{ij} f(s_i, t_j) \leqq \overline{C}, && (66)\\
& f(s_i, t_j) \geqq 0, \qquad s_i \in S, t_j \in T. && (67)
\end{aligned}
$$

この問題の目的は，総輸送コストの上限 $\overline{C}$ を超えない範囲で，供給・需要双方のメンバシップの最小値を最大にする輸送パターンを求める事である。もし総輸送コストの制約式がなければ，下記の手順で各供給ノード，需要ノードにおける輸送量
X_{s_i}, X_{t_j} が定められるので，得られた値を用いて通常の輸送問題を解くことによって，最適なフローパターンが求められる。

[供給量・需要量の計算]

1. メンバシップ値を α とおいて X_{s_i}, X_{t_j} について解く。
すなわち，$X_{s_i} = \mu_{s_i}^{-1}(\alpha), X_{t_j} = \mu_{t_j}^{-1}(\alpha)$ を得る。
2. $\sum_{s \in S} \mu_{s_i}^{-1}(\alpha) = \sum_{t \in T} \mu_{t_j}^{-1}(\alpha)$ を満たすような α を計算した結果を，ステップ1で得られた X_{s_i}, X_{t_j} の式に代入する。この計算で得られた値で通常の輸送問題を解き，総輸送コストが上限 $\overline{C}$ を超えるときは，総輸送量を減らす。このとき，供給側の満足度（メンバシップ値）は上がるが，需要側は下がるので，目的関数の意味マックスミニから，需要側の満足度を均一にできるだけ高く保ちながら輸送量を減少させす。このため，現在のフローパターンで使われているアークのうち，コスト c_{ij} が最大のものを各需要ノードごとに求め（このアークを $(\hat{s}_i, t_j)$，コストを $c_{\hat{s}_i, t_j}$ とする），これら n 本のアーク $(\hat{s}_i, t_j)$ $j = 1, \cdots, n$ への輸送量を減らす。この操作を少なくとも1本のアークの輸送量が0になるまで行い，0となったときに総輸送コストの制約を満足したかどうか調べる。まだ減少量が不十分なため制約を満たしていなければ，同様の操作を繰り返す。コスト制約 (3.7) 式が等号で成立していれば現在の解が最適となるが，真の不等式（$<$）ならば、輸送量を減らしすぎかもしれないので，総輸送コストがちょうど $\overline{C}$ になるように増やすことができるか調べる。今，輸送量を減らす操作を k 回繰り返したときに，コスト制約を不等式で満たしたと仮定する。また，この問題の解を表すために m 行 n 列のマトリックス $(\mathcal{F})$ を導入し，$(k-1)$ 回目の（その満足度と対応する総コストをそれぞれ α'，C' とする）解を $(\mathcal{F}')$，総輸送コストがちょうど $\overline{C}$ になるような満足度 α^* に対する解を $(\mathcal{F}^*)$ とする。このとき，LP の基底解 (basic solution) の観点から，$(\mathcal{F}^*)$ と $(\mathcal{F}')$ は「解の形」（基底解と非基底解の配置）が同じになるので，
$$\sum_{j=1}^{n} c_{\hat{s}_i, t_j} \{\mu_{t_j}^{-1}(\alpha') - \mu_{t_j}^{-1}(\alpha^*)\} = C' - \overline{C} \tag{68}$$
を解くことによって α^* の値が決まり，最適な X_{s_i}, X_{t_j} が求められることになる。

以上の考え方を基に，ファジィ輸送問題 FTP を解くアルゴリズムが [25] で導かれている。さらに，多田等は，整数制約をつけたファジィ輸送問題も考えた [55, 53]。

その他のファジィネットワーク問題

シェアリング問題に対するファジィシェアリング問題 ([56]), 古川長太による最近のファジィ最短経路問題の研究 ([19])、人員配分にファジィ理論を応用した研究 ([35, 36])、ファジィ割り当て問題 ([54]) などいろいろ考えられている。

6. おわりに

ファジィ組合せ最適化としては，この他ナップサック問題にファジィランダム変数を導入したファジィランダムナップサック問題 ([30]) 等沢山あるが、ここでは割愛した。また，ファジィランダム線形計画などの研究もある ([32]) がこれも紙面上は割愛した。最近の有用なファジィ概念としては、ランダムファジィ変数というものがあり、注目している。これはある可能性で、確率現象が起こるとしたもので最近の社会情勢における意思決定に有効ではないかと思っている。最後に宣伝を 1 つ。最近 OR 学会から 40 周年記念出版の 1 つとして「ファジィOR」[23] の本を岩本先生（九州大学)、坂和先生（広島大学）と私の編で玄先生（足利工業大学）をはじめとする何人かのファジィOR の研究者の執筆で出させて頂いた。一度手に取っていただければ幸いであります。この本からの参照をはじめ、多田実氏（龍谷大学)、伊藤健氏（流通科学大学)、片桐英樹氏（広島大学）には博士論文からの引用をさせて頂いた。ここに感謝致します。

参考文献

[1] A.V.Aho, J.E.Hopcroft, J.D.Ullman: *Data Structures and Algorithms*, Addison Wesley Publishing, 1983.

[2] S.K.Baruah: The multiprocessor scheduling of precedence constrained task systems in the presence of interprocessor communication delay, Operations Research, Vol.46, pp.65-72 (1998).

[3] L. Bianco, J. Blazewicz, P. Dell7Olmo and M. Drozdowski: Linear algorithm for preemptive scheduling of multiprocessor tasks subject to minimal lateness, Discrete Applied Mathematics, Vol. 72, pp. 25-46 (1997).

[4] J.Blazewicz and Z.Liu: Scheduling multiprocessor tasks with chain cnstraint, European J. of Operational Research, Vol. 94, pp. 231-241 (1996)

[5] P.Brucker: *Scheduling algorithms*, Springer (1995).

[6] S.Chanas and W.Kolodziejczyk: Maximum flow in a network with fuzzy arc capacities, Fuzzy Sets and Systems, Vol. 8, pp. 165-173 (1982).

[7] S.Chanas and W.Kolodziejczyk: Integer flows in a network with fuzzy capacity constraint, Networks, Vol.16, pp. 17-31 (1986).

[8] D. Cherition and R. E. Tarjan:Finding minimum spanning tree trees, SIAM Journal on Computing, Vol. 5, pp. 724-742 (1976).

[9] T. K. Dey:On counting triangulations in *d* dimensions, Discrete and Computational Geometry, Vol. 12, pp. 281-289 (1994)

[10] E.W.Dijkstra: “A note on two Problem in connection with graphs”, Numerische Mathematik, Vol.1, pp. 269-271(1959).

[11] M.Drozdowski: Scheduling multiprocessor tasks- A overview, European J. of Operational Research, Vol. 94, pp. 215-230 (1996).

[12] D. Dubois: Linear programming with fuzzy data, in: J.C. Bezdek (ed.), Analysis of Fuzzy Information, Volume 3: Applications in Engineering and Science, CRC Press, pp.241–263 (1987).

[13] D. Dubois and H. Prade: Operations on fuzzy numbers, International Journal of Systems Science, Vol. 9, pp. 613-626 (1978).

[14] D. Dubois and H. Prade: *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York (1980).

[15] D. Dubois and H. Prade: *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenumn Press (1988).

[16] S.Even: *Graph Algorithms*, Computer Science Press, 1979.

[17] L.R.Ford, D.R.Fulkerson: Maximal flow through a network, Canadian Journal of Mathematics, Vol. 8, pp. 399-404(1956).

[18] T. Fukuda: On a class of fuzzy random vectors, Journal of Japan Society for Fuzzy Theory and Systems, Vol. 10, pp. 499-505 (1998).

[19] N. Furukawa: A parametric total order on fuzzy numbers and a fuzzy shortest route problem, Optimization, Vol. 10, pp. 367-377 (1994).

[20] S. Geetha and K. P. K. Nair: On stochastic spanning tree problem, Networks, Vol. 23, pp. 675-679 (1993).

[21] J.A. Hoogeven, S.L.van de Velde and B.Veltman: Complexity of scheduling multiprocessor tasks with prespecified processor allocations, Discrete Applied Mathematics, Vol. 17, pp. 205-208 (1994) .

[22] T.C.Hu: *Combinatorial Algorithms*, Addison Wesley, 1982.

[23] 石井博昭，坂和正敏，岩本誠一編：ファジィOR，朝倉書店 (2001).

[24] H. Ishii and M. Tada: : Single machine scheduling problem with fuzzy precedence relation，European Journal of Operational Research，Vol. 87，pp. 284-288（1995）.

[25] 石井博昭，多田実，西田俊夫:ファジィ輸送問題，日本ファジィ学会誌，Vol. 2, pp.79-84(1990)

[26] H.Ishii，M.Tada and T.Masuda：Two scheduling problems with fuzzy duedates，Fuzzy Sets and Systems，Vol. 46, pp. 339-347（1992）.

[27] 伊藤健，石井博昭:必然性測度に基づくファジィ・スパニングツリ問題の一解法，Journal of the Operations Research Society of Japan, Vol. 39，pp. 247-256 (1996).

[28] J.R.Jackson：Scheduling a Production Line to Minimize Maximum Tardiness，*Research Report* **43**，*Management Sciences Research Project*，*UCLA*（1955）.

[29] B.Jurisch: Lower bounds for the job shop scheduling problem on multi-purpose machines, Discrete Applied Mathematics，Vol.58, pp.145-156 (1995).

[30] 片桐英樹:不確実・不確定状況下での数理的意思決定の基礎的研究，大阪大学博士論文 (2000).

[31] H. Katagiri and H. Ishii: Chance constrained bottleneck sapnning tree problem with fuzzy random edge costs, Journal of the Operations Research Society of Japan, Vol. 43, pp. 128-137 (2000)

[32] H. Katagiri and H. Ishii:Linear programming problem with fuzzy random con-

straint, Mathematica Japonica, Vol. 52, pp. 123-129, (2000).

[33] 片桐英樹，石井博昭，坂和正敏:可能性および必然性に基づくファジィ極大木問題日本ファジィ学会誌，Vol. 12, pp. 797-805, (2000).

[34] A. Kaufmann and M.M. Gupta: *Introduction to fuzzy arithemetic: Theory and applications*, Van Nostrand (1985).

[35] 今野勤、石井博昭:ファジィ人員配分問題，日本ファジィ学会誌，Vol. 7, pp. 624-629 (1995).

[36] 今野勤、石井博昭:ファジィ人員配分問題，日本ファジィ学会誌，Vol. 7, pp. 630-636 (1995).

[37] T. Konno and H. Ishii: Two machine scheduling problem with fuzzy allowable time constraint, Jopurnal of the Operations Research Society of Japan, Vol. 41, pp. 487-491 (1998).

[38] T. Konno and H. Ishii: An Open shop scheduling problem with fuzzy allowable time and fuzzy resource constraint, Fuzzy Sets and Systems, Vol. 109, pp. 141-147 (2000)

[39] R. Kruse and K. D. Meyer: *Statistics with Vague Data*, D. Reidel Publishing Company, Dordrecht (1987).

[40] J. B. Kruskal: On the shortest spanning subtree of a graph and the traveling salesman problem, *Proceedings of AMS*, Vol. 7, pp.48-50 (1956).

[41] M.Kubale: Preemptive versus nonpreemptive scheduling of biprocessor tasks on dedicated processors, European J. of Operational Research, Vol. 94, pp. 242-251 (1996).

[42] H. Kwakernaak: Fuzzy random variable- I . Definitions and theorems, Information Sciences, Vol.15, pp. 1-29 (1978).

[43] Y.J. Lai: Hierarchical optimization: a satisfactory solution, Fuzzy Sets and Systems, Vol. 77, pp. 321-335 (1996).

[44] Y.J. Lai and C.L. Hwang: *Fuzzy Multiple Objective Decision Making*, Springer-Verlag, Berlin (1994).

[45] E. L. Lawler and J. M. Moore:A functional equation and its application to resource allocation and sequencing problems, Management Science, Vol. 16, pp. 77-84(1969).

[46] S. Li and Y. Ogura: Fuzzy random variables, conditional expectations and fuzzy valued martingales, Journal of Fuzzy Mathematics, pp. 905-927 (1996).

[47] J. M. Moore：An n Job, One machine sequencing algorithm for minimizing the number of late Jobs, Management Science, Vol. 15, pp. 102-109（1968）.

[48] R.C.Prim: Shortest connection networks and some generalizations, Bell System Technical Journal, Vol. 36, pp. 1389-1401(1957).

[49] M. L. Puri and D. A. Ralescu: Fuzzy random variables, Journal of Mathematical Analysis and Application, Vol. 114, pp. 409-422 (1986).

[50] 坂和正敏, 石井博昭, 西崎一郎： ソフト最適化, 朝倉書店 (1995).

[51] R. Slowinski (eds.): *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, Kluwer Academic Publishers, Boston (1998).

[52] R. Slowinski and J. Teghem: *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming Under Uncertainty*, Kluwer Academic Publishers, Dor-

drecht (1990).
[53] M. Tada and H. Ishii: An integer fuzzy transportation problem, Computers and Mathematics, Vol. 31, pp. (1996)
[54] 多田実、石井博昭:2目的ファジィ割り当て問題, 日本ファジィ学会誌, Vol. 10, pp. 867-875 (1998).
[55] M. Tada, H. Ishii and T. Nishida:Fuzzy transportation problem with integral flow, Mathematica Japonica, Vol. 35, pp. 335-341 (1990).
[56] M. Tada, H. Ishii, T. Nishida and T. Masuda:Fuzzy sharing problem, Fuzzy Sets and Systems, Vol. 33, pp.303-313 (1989).
[57] G. Wang and Y. Zhang: The theory of fuzzy stochastic process, Fuzzy Sets and Systems, Vol. 51, pp. 161-178 (1992).
[58] N. Watanabe: Fuzzy random variables and statistical inference, Journal of Japan Society for Fuzzy Thoery and Systems, Vol. 8, pp. 126-135 (1996).
[59] D.Werra et. al:A preemptive open shop scheduling problem with one resource, Operations Research Letters, Vol.10, pp. 9-15 (1991).
[60] A. C. Yao: An $O(|E|loglog|V|)$ algorithm for finding minimum spanning trees, Information Processing Letters, Vol. 4, pp. 21-23 (1975).
[61] L.A. Zadeh: Fuzzy sets, Information and Control, Vol. 8, pp. 338-353 (1965).
[62] L.A.Zadeh: Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems, Vol.1 pp 3-28 (1978).
[63] C. Zhong and Z. Guohua: The equivalence of two definitions of fuzzy random variable, Preprints of Second IFSa Congress, pp. 59-62 (1986)
[64] H.-J. Zimmermann: Fuzzy programming and linear programming with several objective functions, Fuzzy Sets and Systems, Vol. 1, pp. 45-55 (1978).
[65] H.-J. Zimmermann: *Fuzzy Sets, Decision-Making and Expert Systems*, Kluwer Academic Publishers, Boston (1987).
[66] H.-J. Zimmermann: *Fuzzy Set Theory and Its Applications*, 2nd Revised Edition, Kluwer Academic Publishers, Boston (1991).