



# HOKKAIDO UNIVERSITY

Title	ハイブリッド型遺伝的アルゴリズムによるシステムにおけるファジィ信頼性の最適設計とその応用に関する研究
Author(s)	佐々木, 正仁
Degree Grantor	北海道大学
Degree Name	博士(工学)
Dissertation Number	乙第5498号
Issue Date	1999-06-30
DOI	<a href="https://doi.org/10.11501/3154946">https://doi.org/10.11501/3154946</a>
Doc URL	<a href="https://hdl.handle.net/2115/51620">https://hdl.handle.net/2115/51620</a>
Type	doctoral thesis
File Information	000000340134.pdf



ハイブリッド型遺伝的アルゴリズムによるシステムにおける  
ファジィ信頼性の最適設計とその応用に関する研究

佐々木 正 仁

# ハイブリッド型遺伝的アルゴリズムによるシステムにおける ファジィ信頼性の最適設計とその応用に関する研究

佐々木 正仁

# 目次

第1章	序論	1
第2章	システム信頼性の最適設計	7
2.1	システム信頼性の最適設計問題	7
2.1.1	システム信頼性の最適設計問題	8
2.1.2	大規模なシステム信頼性の最適設計問題	9
2.1.3	多目的信頼性最適設計モデル	10
2.2	不完全なFDSを伴うシステム信頼性の最適設計問題	10
2.2.1	最適ユニット選択と冗長配分モデル	10
2.2.2	数種類の故障モードを伴うシステム信頼性の最適設計問題	13
第3章	遺伝的アルゴリズムとハイブリッド化	15
3.1	まえがき	15
3.2	遺伝的アルゴリズムの基礎	16
3.2.1	遺伝的アルゴリズムの概要	16
3.2.2	遺伝子表現	17
3.2.3	評価関数	17
3.2.4	遺伝的操作	18
3.3	非線形最適化問題への応用	19
3.3.1	制約条件のない非線形最適化問題	19
3.3.2	0-1非線形計画問題	21
3.3.3	非線形整数計画問題	21
3.4	ハイブリッド型遺伝的アルゴリズム	22
3.4.1	ハイブリッド型遺伝的アルゴリズムの概要	22
3.4.2	GUB構造に基づくハイブリッド型遺伝的アルゴリズム	22
第4章	ファジィ数理計画法と多目的計画法	25
4.1	まえがき	25
4.2	ファジィ理論と数理計画法	27
4.2.1	ファジィ集合とファジィ関係	27
4.2.2	ファジィ数理計画法	29

4.3	多目的計画法	33
4.3.1	多目的計画問題	33
4.3.2	ファジィ多目的計画問題	34
4.4	ファジィ多目的計画問題の会話型解法	36
4.4.1	計算法	36
4.4.2	数値例	38
4.4.3	評価	43
4.5	むすび	46
第5章 ファジィ多目的信頼性の最適設計		47
5.1	まえがき	47
5.2	ファジィ多目的信頼性設計I	49
5.2.1	ファジィ多目的信頼性設計問題	49
5.2.2	計算法	51
5.2.3	最適設計問題への応用	54
5.2.4	評価	61
5.3	不完全なFDSをもつ冗長ユニット配分問題	62
5.3.1	Sタイプ故障の負荷状態待機冗長システム	62
5.3.2	Mタイプ故障の負荷状態待機冗長システム	63
5.4	ファジィ多目的信頼性設計II	64
5.4.1	ファジィ多目的信頼性設計問題	64
5.4.2	計算法	66
5.4.3	数値例	69
5.4.4	評価	75
5.5	むすび	76
第6章 遺伝的アルゴリズムによるファジィ信頼性の最適設計		77
6.1	まえがき	77
6.2	ファジィ信頼性最適設計問題	79
6.3	遺伝的アルゴリズムによる解法	82
6.3.1	遺伝的操作	82
6.3.2	評価関数	83
6.3.3	計算法	83
6.3.4	数値実験	86
6.4	むすび	90

第7章	ハイブリッド型遺伝的アルゴリズムによるシステム信頼性の最適設計	93
7.1	まえがき	93
7.2	システム信頼性設計問題	94
7.2.1	GUB構造を伴う混合0-1線形計画問題	94
7.2.2	GUB構造を伴うナップサック問題への変換	96
7.3	ハイブリッド型遺伝的アルゴリズム	97
7.3.1	遺伝子表現	97
7.3.2	評価関数	98
7.3.3	遺伝的操作	99
7.3.4	決定変数のランクに基づく並換え	99
7.4	計算法	100
7.5	数値例	101
7.6	定量的評価	106
7.7	むすび	109
第8章	ハイブリッド型遺伝的アルゴリズムによる二目的のシステム信頼性の最適設計	111
8.1	まえがき	111
8.2	二目的のシステム信頼性設計問題	112
8.2.1	GUB構造を伴う二目的ナップサック問題	112
8.3	ハイブリッド型遺伝的アルゴリズム	113
8.3.1	遺伝子表現	113
8.3.2	評価関数	114
8.3.3	遺伝的操作	115
8.3.4	決定変数のランクに基づく並換え	115
8.4	計算法	116
8.5	数値例	117
8.6	むすび	122
第9章	結 言	123
	参 考 文 献	125
	謝 辞	137

# 第 1 章

## 序 論

近年の半導体技術の急速な進歩により，コンピュータをベースとした情報処理システムの機能は，ますます巨大化，統合化，また複雑化している．電気・ガスなどのエネルギー供給システム，テレビ・電話などの情報通信システム，鉄道・航空・道路などの交通システム，病院・消防などの医療・救急システムなど，いずれも現代社会にとって必要不可欠なシステムである．社会におけるこうした情報処理システムの役割はその重要度を増しており，万が一故障が発生した場合に及ぼす社会的・経済的影響は計り知れないほど大きく，時に人命にかかわるほど大きな役割を担っている．現代の情報処理システムのおかれたこのような社会的環境のもとでは，システムを長時間故障なく，その処理機能を満足に稼働させるためのシステム信頼性の最適設計は今後ますます重要な課題となると考えられる．

最近，遺伝的アルゴリズム (Genetic Algorithm : GA) がコンピュータサイエンス，生物科学，計測／制御，オペレーションズ・リサーチ／経営工学などの分野において大きな注目を集めている．また，組合せ問題の有効な解法として注目されており，各種の最適化問題に応用されている<sup>[21, 22, 24, 25, 86]</sup>．GA は，1960年代に Holland によって生物学の遺伝的システム，いわゆる生物進化（自然淘汰・交叉・突然変異）の現象を，コンピュータ・シミュレーションによって模倣する確率的探索・最適化・学習の一手法として提案された，自然淘汰と遺伝現象のメカニズムを単純化した数式モデルであり，個体と呼ばれる染色体の集合が外部環境に適応するように，集団の構成を世代ごとに生成させるものである．このGAは，非線形整数計画問題として定式化されるシステム信頼性の最適設計問題に対しても有効な解法の一つとなる<sup>[24, 25, 125, 126, 127, 128, 129, 130]</sup>．ただしGAでは，一般的に最適解周辺には早く近づくが局所探索能力が低いという問題点が指摘されている．この問題を解決する手段の一つとしてハイブリッドGAが提案されている<sup>[2, 43, 44]</sup>．ハイブリッドGAは，GAとヒューリスティック手法を組み合わせたもので，大域的な探索をGAによって行い，局所探索をヒューリスティック手法で行うものである．つまりGAによって高速に最

適解周辺に近づき、GAの苦手とする局所探索をヒューリスティック手法に行わせるのである。これによりハイブリッドGAは、優れた解をより高速に探索することができる。

他方、オペレーションズ・リサーチ／経営工学の分野でもっともよく利用され、効果を発揮している数理計画法は、1940年代の後半にDantzigがシンプレックス法を開発し、線形計画法が実用的な手法として認められて以来、急速に発展した。数理計画法は、自然科学や社会科学等のさまざまな現象を数学的モデルとして表し、とるべき方策を数理的に最適化することを目的として、各種の解法が次々に誕生してきた。特に、多目的数理計画法は、近年の社会的要求の多様化に伴い、従来のような断片的な意思決定問題にとどまらず多数の互いに競合する目的関数を同時に考慮することができる現実的な最適化手法として注目されている。多目的線形計画問題は、複数個の相競合する目的関数を対象とするため、一般にある目的関数を改善するために少なくとも一つの目的関数を犠牲にしなくてはならないような解、すなわちパレート最適解 (Pareto optimal solution) が数多く存在する。現実の意思決定においては、意思決定者 (decision maker) は自己の選好に基づいて、パレート最適解の集合の中から最終的に最良と考えられる解を選択しなければならない。ところが、意思決定者の選好構造を十分に反映させるいわゆる選好関数は本来未知で、また直接同定することが困難である場合が多いため、意思決定者の選好関数を大域的に同定することなく、対話により得られる局所的な選好情報に基づいて意思決定者の選好解を導出するという、いわゆる対話型手法が数多く提案されてきた<sup>[15, 32, 45, 48, 51, 57, 70, 72, 138, 139]</sup>。ここで、人間の判断のあいまいさから、意思決定者は複数の目的関数それぞれに対して”だいたいある値以上にしたい”という、いわゆるファジィ目標をもつものと、一般には考えられる。このような観点から、多目的線形計画問題における意思決定者のファジィ目標が線形のメンバシップ関数で表される場合、BellmanとZadehの最大化決定に従えば、線形計画問題になることがZimmermann(1976)によって示され<sup>[135]</sup>て以来、ファジィ数理計画法が急速に発展した。

本研究の目的は、システム信頼性の最適設計問題のためのファジィ目標とファジィ制約を伴う会話型多目的意思決定問題の解法、ならびにハイブリッド型GAによる解法を提案することにある。提案する解法は、GUB構造を伴うシステム信頼性の最適設計問題を対象とするもので、従来解法と比較してファジィ目標を導入することによって意思決定者の主観的判断に起因するあいまいさを考慮することができるほか、ファジィ制約を導入することによってシステムの使用環境や経済情勢などの変化に対応できる、より現実的で柔軟な意思決定やシステム設計が可能となっている。ファジィ目標を伴う目標計画問題の会話型解法では、意思決定者の選好構造

をより反映させることを目的とした実用的な解法について、その定式化および計算法を提案する。また、GUB (generalized upper bounding) と呼ばれる大規模なシステム特有の構造を考慮した遺伝子表現と探索過程をGAに導入したハイブリッド型のGAにより、GUB構造を伴うシステム信頼性の最適設計問題を効率的に解くことが可能となった。

また、これらの解法の応用として、より現実的なシステム信頼性の最適設計問題である故障検出切換え装置 (fault detecting and switching: FDS) の故障を考慮に入れたシステムに対する解法、さらに De Novo 計画問題やナップサック問題として定式化されるシステム信頼性の最適設計問題の解法を提案している。以下に本論文の各章の内容を要約して述べる。

第2章および第3章において、第4章以降で取り上げるシステム信頼性の最適設計問題およびその解法の一つであり、組合せ最適化問題に対する解法の一つとして有効なGAについて基礎概念を述べる。第4章および第5章では、ファジィ目標およびファジィ制約をもつ多目的意思決定問題の会話型解法およびシステム信頼性の最適設計問題の解法について取りあげる。第6章において、ファジィ目標とファジィ制約を伴うシステム信頼性のGAによる解法、第7章および第8章では、GUB構造を伴う De Novo 計画問題およびその変換された形式であるナップサック問題のハイブリッド型GAによる解法について取り上げる。

第2章においては、システム信頼性の最適設計技法として、大規模なシステム信頼性の最適設計として設計案の選択とユニットの冗長配分問題、不完全なFDSを伴うシステム信頼性の最適設計、および数種類の故障モードを伴うシステム信頼性の最適設計について紹介する。

第3章では、最近、組合せ最適化問題に対する有効な解法として数多くの成果が報告されているGAの基本的な概念および計算法について紹介する。遺伝子表現、評価関数、基本的な遺伝的操作のほか、ハイブリッドGAについて概説する。

第4章では、ファジィ理論の基礎概念について説明するとともに、ファジィ目標およびファジィ制約条件を伴う多目的意思決定問題の会話型解法を提案する。この解法では、ファジィ目標を導入することによって、意思決定者の主観的な判断に起因するあいまいさを考慮に入れることができるのと同時に、ファジィ制約を導入することによってシステムの使用環境の変化や経済情勢の変動に対応することができる、より現実的で柔軟な意思決定が可能となる。また、意思決定者の各目的関数に対する選好を反映させるための重み付けに、多評価意思決定のための支援手段として評価されているAHP (analytic hierarchy process) を用いるとともに、意思決定の補助手段として、複数のパレート最適解の中から最良な妥協解を決定するためにTOPSIS (technique for order preference by similarity to ideal solution) を導入するこ

とによって、意思決定者の意向をより反映した解を容易に得ることができる。さらに、数値例として6種類の食品による最適献立問題を取りあげ、従来解法と比較することによって提案する手法の有効性を示している。

第5章では、システム信頼性の最適設計問題に対して、ファジィ目標およびファジィ制約を導入することによって、意思決定者であるシステム設計者の意向を反映した設計を可能とした、現実的で柔軟な解法を提案する。この解法は、玄らの提案するGUB構造を活用した効率的な解法<sup>[13, 16, 37, 38, 40]</sup>を多目的計画問題に拡張して適用したこと、及びコンピュータの計算速度、さらにメモリの使用量の点で従来解法より優れている。提案する解法の有効性を示すために、数値例としてユニットの冗長配分の設計問題を取りあげる。さらに、ここで提案する解法の応用として、現実のシステム信頼性の最適設計問題におけるFDSの故障を考慮した解法を提案する。提案する解法の有効性を示すために、3種類の非線形なシステム資源の制約の下で、4種類の異なる故障のタイプをもつサブシステムから構成されたシステムの信頼度を最大に、またコスト関数を最小にする最適な冗長ユニット配分問題を取り扱う。

第6章では、ファジィ目標とファジィ制約をもつシステム信頼性の最適設計問題を解くためにGAを導入した解法を提案する。この解法では、一般に非線形整数計画問題として定式化される信頼性最適設計問題を、線形化することなくそのまま解くことができるため、変数の数を増やしたりする必要がなく、コンピュータ上のメモリ制約や処理速度の点で有利である。さらに、提案する解法は、従来の方法では得ることができなかった解を得ることができる柔軟性を持っていることを、数種類の故障モードを伴うシステム信頼性の最適設計問題を数値例として取りあげることにより示している。

第7章では、与えられた予算額の中で資源の最適配分を含む計画問題の最適な設計を行うことができるDe Novo計画問題として定式化されるシステム信頼性の最適設計問題を解くためのハイブリッド型GAによる解法を提案する。提案する解法は、GUB構造を有効に表現する新しい遺伝子表現を導入することにより、取り扱いが容易である。また、バイナリ変数によるGUB構造の表現を行った場合に比べ、計算効率および必要メモリ量の点で有利であることを示す。さらに、各GUB制約の決定変数を効率指標に基づいてランク付けし、それに基づいて解の改善をはかる過程をGAに組み込む、すなわちハイブリッド化により、GUB構造性を利用した効率的な解の探索が可能になっている。

第8章では、第7章で提案した解法を発展させ、ナップサック問題として定式化される二目的のシステム信頼性の最適設計問題のためのハイブリッド型GAによる解法について提案する。

第9章では、結論として、本研究で得られた成果を要約する。以上の各章間の関連図を図1.1に示す。

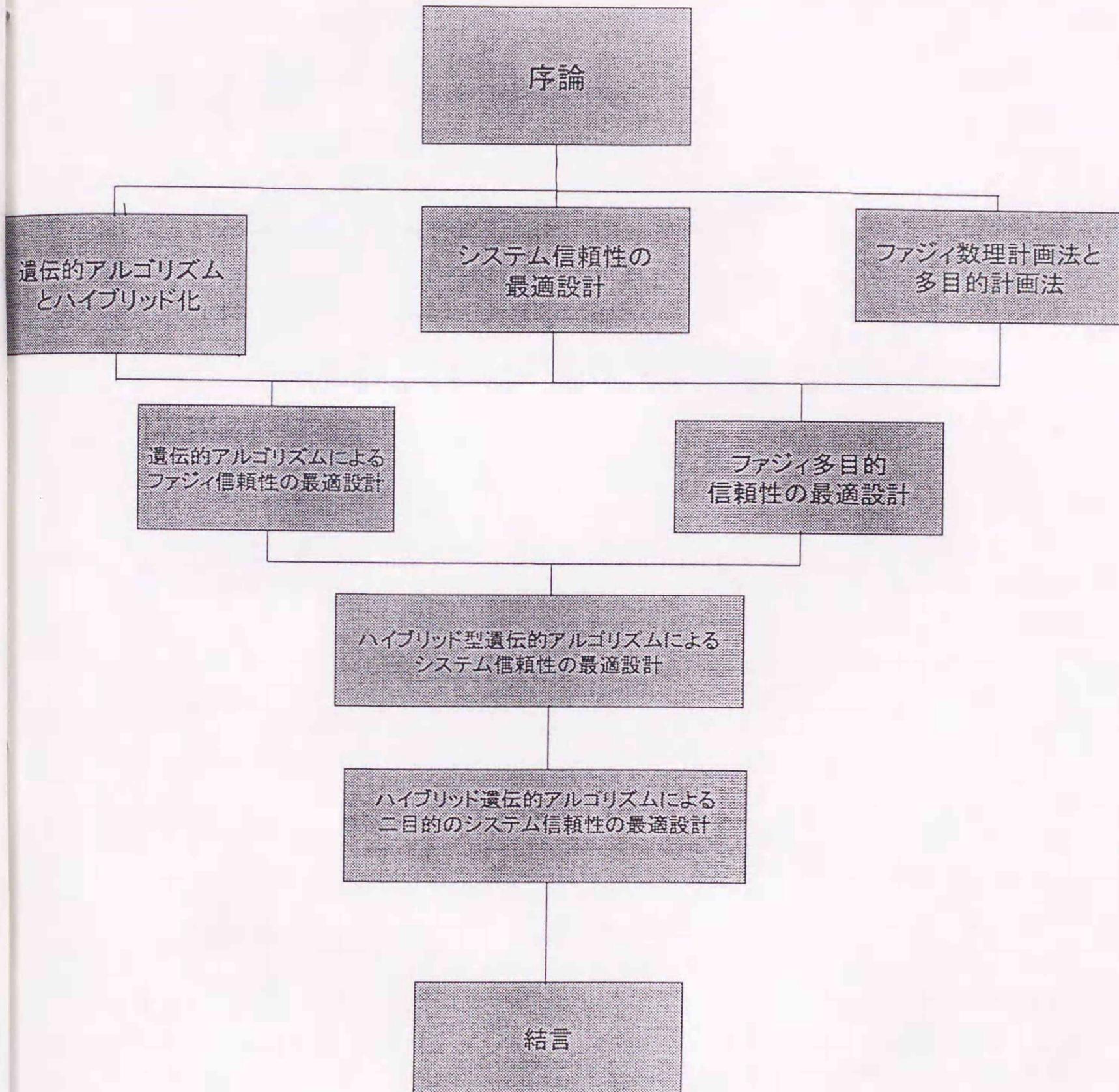


図 1.1 相互関連図



## 第 2 章

# システム信頼性の最適設計

### 2.1 システム信頼性の最適設計問題

近年の半導体技術の急速な進歩により、コンピュータをベースにした情報システムの処理機能は、巨大化、統合化および複雑化している。それに伴い、社会における情報処理システムの役割はますますその重要度を増しており、万が一故障が発生した場合の及ぼす影響は計り知れないほど大きく、時に人命にかかわるほど重大な役割を担っている。現代の情報処理システムのおかれたこのような社会的環境の下では、システムを長時間故障なく、その処理機能を満足に稼働させるためのシステム信頼性の最適設計は今後ますます重要な課題となると考えられる。システム信頼性を向上させる設計上の手段としては、

1. より信頼度の高いユニットの選択
2. 冗長ユニット系の付加
3. 故障の代用モードの付加
4. ディレーティングの強化

などがある<sup>[10, 11, 63]</sup>。特に、2の場合として限られた資源、すなわちコストや重量や容積などの制約条件の下で、システムの信頼度を最大にする最適冗長設計問題は、今まで種々の最適化技法が開発され、解決されてきた<sup>[5, 6, 9, 31, 41, 35, 36, 69]</sup>。

また、システムの運用・保全上からは、故障が起こらないこと、故障が起こった場合にはその検知と保全が重要である。したがって、保全作業を伴うシステムの定常的な使用状態では、その信頼度と保全度を同時に考慮したシステム・アベイラビリティの最適なユニットの選択と冗長配分が問題になってくる。このシステム・アベイラビリティやシステム信頼性の最適ユニット冗長配分などの最適設計問題は、一般

に非線形計画問題として定式化される。この種の非線形計画問題の解法としては、Genや玄・奥野の線形変換化により0-1計画法<sup>[10, 11]</sup>、FyffeらのLagrange乗数を用いた動的計画法、Nakagawaらの2種類のシステム制約条件の場合だけの専用解法アルゴリズム、また、0-1線形計画問題に変換することにより、平面切除法、陰的列挙法やヒューリスティック・アルゴリズムを用いる方法が提案されている<sup>[31, 38]</sup>。さらに、井田らは大規模な計画問題に特有な構造の一つであるGUB (generalized upper bounding) 構造をもつ0-1線形計画問題を効率的に解くための効率的な解法を提案し、米国NASAのLife-Supportシステム信頼性の最適化問題など、現実の問題に適用してその有効性を明らかにしている<sup>[12, 13, 35, 39]</sup>。

### 2.1.1 システム信頼性の最適設計問題

コスト、重量、容積などの限られた諸資源の下でシステムの信頼度を最大化する、あるいはコストを最小化する、システムの最適設計問題を考える。 $n$ 個のサブシステムからなり、各サブシステムでは $m_i$ 個のユニットが並列または待機に冗長構成され、各ユニットは $\alpha_i$ 種類 ( $i = 1, 2, \dots, n$ ) の交換可能なユニットの中からいずれか一種類を選択して用いるものとする。高信頼化システムを設計するために必要な諸資源、例えばコスト、重量、容積など $T$ 種類の非線形な制約条件  $G_r(\alpha, m)$  ( $r = 1, 2, \dots, T$ ) の下で、目的関数である非線形な信頼度を最大化する問題は、次のような一般形で表される。

$$\max R(\alpha, m) = \prod_{i=1}^n R_i(\alpha_i, m_i) \quad (2.1)$$

$$\text{s. t. } G_r(\alpha, m) = \sum_{i=1}^n g_{ri}(\alpha_i, m_i) \leq b_r, \quad r = 1, 2, \dots, T \quad (2.2)$$

$$\alpha_i, m_i \geq 0, \quad : \text{integer}, \quad i = 1, \dots, n \quad (2.3)$$

ここで、

$R_i(\alpha_i, m_i)$ : サブシステム  $i$  に  $m_i$  個のユニットを配分し、設計案  $\alpha_i$  を選択した場合の信頼度

$g_{ri}(\alpha_i, m_i)$ : サブシステム  $i$  に  $m_i$  個のユニットを配分し、設計案  $\alpha_i$  を選択した場合の資源

$b_r$ : 資源  $r$  の制限量

である。この高信頼化システムを実現する最適設計問題は、各サブシステムのユニット  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$  の選択と冗長ユニット  $m = (m_1, m_2, \dots, m_n)$  の配分を同時

に行う非線形整数計画問題となる。この種のシステム信頼性最適設計問題の解法として、いろいろな最適化技法が研究されている。これまでに提案されている主な技法としては、・動的計画法、・線形計画法、・整数計画法、・離散型最大原理、・Convex計画法、・Backtracking法、・スライド法、・ラグランジュ乗数法、・幾何計画法、・SUMT(sequential unconstrained minimization technique)法、・分岐限定法、・直接簡略化法、・離散型最適化法、・パラメトリック法、・0-1計画法、・ヒューリスティック計画法などがある。

### 2.1.2 大規模なシステム信頼性の最適設計問題

取り扱うシステムの規模が大きくなるに伴い、そこにはそのシステム特有な構造が必ず発生する。したがって、システムを解析する場合には、そのシステムを持っている特有な構造を十分に活用することによって、効率的な計算法を開発することが要求される。このような大規模システム特有な構造の一つに、GUBと呼ばれる特殊な構造があることが知られている。

#### 0-1線形計画問題への変換

非線形整数計画問題を0-1線形計画問題として線形定式化するために、次のような0-1変数を定義する。

$$x_{ij} = \begin{cases} 1: \text{サブシステム } i \text{ において } j \text{ 個のユニットを配分するとき} \\ 0: \text{ そうでないとき} \end{cases} \quad (2.4)$$

システム信頼性の最適設計問題を、0-1線形計画問題に変換する場合、このGUB構造が出現する。 $T$ 個のシステム制約と $n$ 個のGUB制約の下で、目的関数を最大又は最小化する0-1線形計画モデルは、次のように表される。

0-1LP(GUB):

$$\max/\min \quad z = \sum_{i=1}^n \sum_{j=1}^{n_i} c_{ij} x_{ij} \quad (2.5)$$

$$\text{s. t.} \quad g_r(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} a_{rij} x_{ij} \leq b_r, \quad r = 1, 2, \dots, T \quad (2.6)$$

$$g_{r+i}(\mathbf{x}) = \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (2.7)$$

$$x_{ij} = 0 \text{ or } 1, \quad j = 1, 2, \dots, n_i, \quad i = 1, 2, \dots, n \quad (2.8)$$

ここで、式(2.5)をシステム制約、式(2.6)をGUB制約と呼び、 $c_{ij}$ はGUB制約 $i$ における $j$ 番目の要素に対応する目的関数の係数、 $a_{rij}$ はGUB制約 $i$ における $j$ 番目の要素に対応する $r$ 番目のシステム制約の係数、 $b_r$ は $r$ 番目のシステム制

約の右辺定数, さらに  $n_i$  は G U B 制約  $i$  に含まれる決定変数の数である. また,  $x_{ij}, (j = 1, 2, \dots, n_i)$  は G U B 制約  $i$  における決定変数であり, 0 または 1 のどちらかの値をとる.

### 2.1.3 多目的信頼性最適設計モデル

近年, 社会的要求の多様化に伴い, 信頼度のみを最適化するシステム信頼性の最適設計にとどまらず, コストや重量, 容積など複数の互いに競合する複数の目的関数を同時に考慮する多目的信頼性最適設計が注目されている. 多目的信頼性の最適設計問題についても, 0-1変数を導入することによって0-1線形計画問題に変換する場合, このGUB構造が出現する.  $T$ 個のシステム制約と  $n$ 個のGUB制約の下で, 複数の目的関数を最大又は最小化する多目的0-1線形計画モデルは, 次のように表される.

0-1MLP(GUB):

$$\max / \min \quad z_k = \sum_{i=1}^n \sum_{j=1}^{n_i} c_{kij} x_{ij}, \quad k = 1, 2, \dots, q \quad (2.9)$$

$$\text{s. t.} \quad g_r(\mathbf{x}) = \sum_{r=1}^T \sum_{j=1}^{n_i} a_{rij} x_{ij} \leq b_r, \quad r = 1, 2, \dots, T \quad (2.10)$$

$$g_{T+i}(\mathbf{x}) = \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (2.11)$$

$$x_{ij} = 0 \text{ or } 1, \quad j = 1, 2, \dots, n_i, \quad i = 1, 2, \dots, n \quad (2.12)$$

$c_{kij}$  は  $k$  番目の目的関数に対する G U B 制約  $i$  における  $j$  番目の要素に対応する目的関数の係数を表す.

## 2.2 不完全な F D S を伴うシステム信頼性の最適設計問題

従来の待機冗長システムの信頼度解析においては, ほとんどの場合, 故障したユニットの故障検出切換え装置 (fault detecting and switching: FDS) が完全である, すなわち FDS 装置の信頼度が常に 100% であると仮定してきた. しかし, 現実のシステムでは切換え装置が完全に信頼できるとは限らない. したがって, もし FDS が故障していると冗長ユニットがあっても活用できなくなるため, FDS の故障を考慮したシステム信頼性の最適設計はより現実的な問題である.

### 2.2.1 最適ユニット選択と冗長配分モデル

ここで取り上げる負荷状態待機冗長 (loaded standby redundant: LSR) システムの信頼度関数は, その故障のタイプによって次の二つに分けることができる.

Sタイプの故障：サブシステム内の待機冗長グループ(基本ユニットと冗長ユニット)はFDSが故障すると同時に動作を停止する．このタイプの故障をするFDSをもつLSRシステムをS-LSRシステムと呼ぶ．

この待機冗長グループの構成を図2.1に示す．

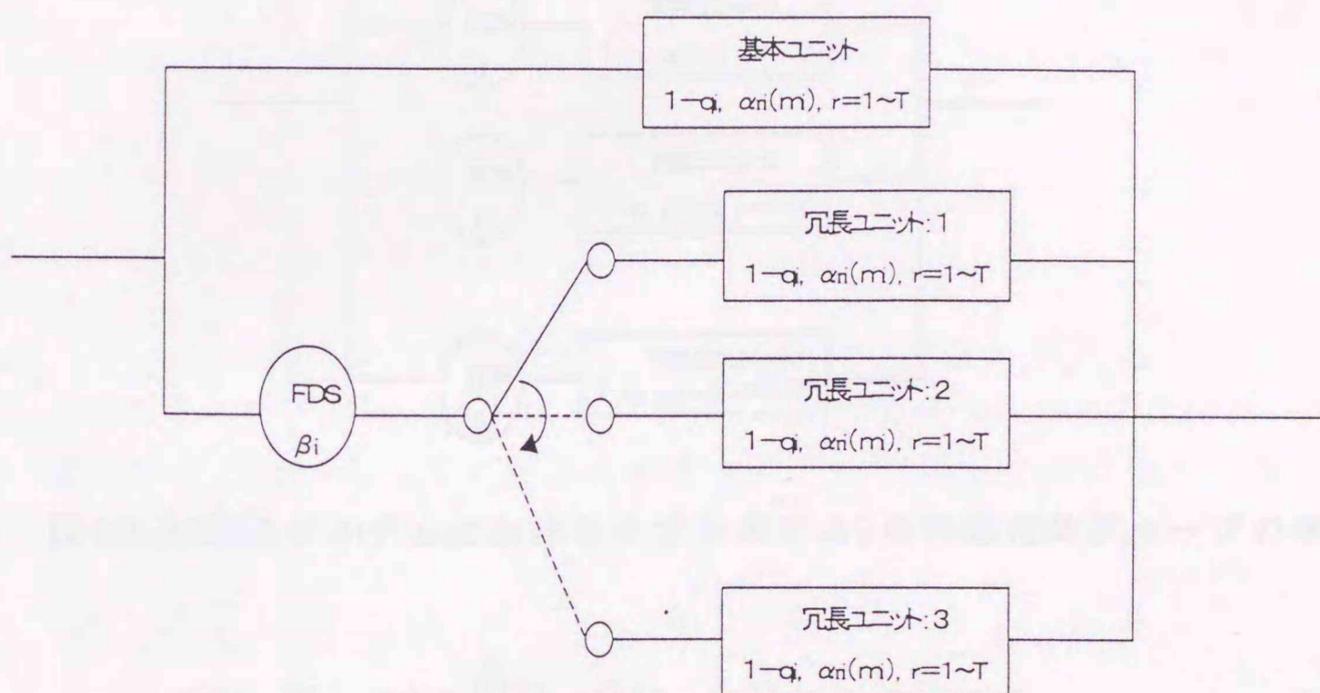


図2.1 S-LSRシステムにおけるサブシステム*i*の待機冗長グループの構成

$$\begin{aligned} \max R(\alpha, m) &= \prod_{i=1}^n \{1 - Q_i(\alpha_i, m_i)\} \\ &= \prod_{i=1}^n \left\{ 1 - \frac{\beta_i q_i \alpha_i + p_i \alpha_i q_i \alpha_i^{m_i} (1 - \beta_i)^{m_i}}{p_i \alpha_i + \beta_i q_i \alpha_i} \right\} \end{aligned} \quad (2.13)$$

Mタイプの故障：サブシステム内の待機冗長グループの各ユニットが専用のFDSをもち、任意のFDSが故障した場合次のFDSが動作する．このタイプの故障をするFDSをもつLSRシステムをM-LSRシステムと呼ぶ．

M-LSRシステムにおけるサブシステム*i*の待機冗長グループの構成を図2.2に示す。

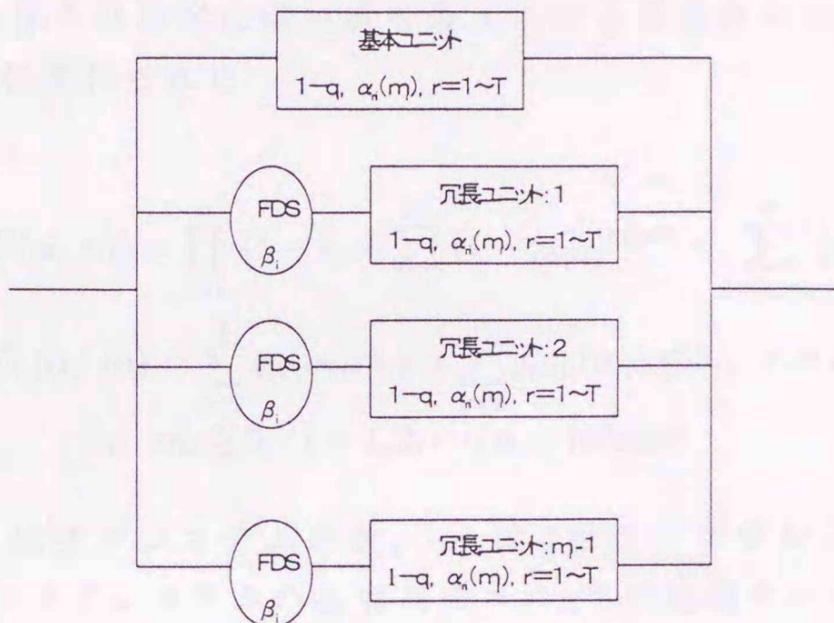


図2.2 M-LSRシステムにおけるサブシステム*i*の待機冗長グループの構成

$$\begin{aligned} \max R(\alpha, m) &= \prod_{i=1}^n \{1 - Q_i(\alpha_i, m_i)\} \\ &= \prod_{i=1}^n [1 - q_{i\alpha_i} \{(1 - \beta_i)q_{i\alpha_i} + \beta_i\}^{m_i-1}] \end{aligned} \quad (2.14)$$

したがって、不完全なFDSをもつシステムのユニット選択と冗長配分の最適設計問題とは、システム資源が*T*種類でかつ次のような非線形な制約条件

$$\begin{aligned} \text{s. t. } G_r(\alpha, m) &= \sum_{i=1}^n g_{ri}(\alpha_i, m_i) = \sum_{i=1}^n g_{ri\alpha_i}(m_i) \leq b_r, \quad r = 1, 2, \dots, T \\ \alpha_i, m_i &\geq 0, \quad i = 1, 2, \dots, n : \text{integer} \end{aligned} \quad (2.15)$$

の下で、上記の非線形な信頼度関数を最大にするような各サブシステムのユニット選択  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$  と冗長配分  $m = (m_1, m_2, \dots, m_n)$  を同時に行う非線形整数計画問題である。ここで、 $\beta_i, p_{i\alpha_i} (= 1 - q_{i\alpha_i})$  はそれぞれサブシステム*i*のFDSの故障率とユニットの信頼度、 $g_{ri\alpha_i}(m_i)$  は*r*番目のサブシステムでの資源の必要量で、 $m_i$  に関して非線形で分離可能な関数、 $b_r$  はシステムで利用可能な資源である。

これらのシステムにおいて、各サブシステムのFDSが完全な信頼性をもつ(すなわち  $\beta_i = 0, i = 1, 2, \dots, n$ ) とすれば、従来の待機冗長(standby redundant:SR)システムに帰着される。

## 2.2.2 数種類の故障モードを伴うシステム信頼性の最適設計問題

システム資源が  $T$  個の非線形な制約条件の下で、高信頼化システムに  $s_i$  種類の故障モードを伴う非線形な信頼度を最大にする最適なユニット選択と配分の問題は、次のように表わされる

$$\max R(\alpha, m) = \prod_{i=1}^n \left\{ 1 - h_i + \sum_{e=1}^{h_i} (1 - q_{ie\alpha_i})^{m_i+1} - \sum_{e=h_i+1}^{s_i} (q_{ie\alpha_i})^{m_i+1} \right\} \quad (2.16)$$

$$\text{s. t. } G_r(\alpha, m) = \sum_{i=1}^n g_{ri}(\alpha_i, m_i) = \sum_{i=1}^n g_{ri\alpha_i}(m_i) \leq b_r, \quad r = 1, 2, \dots, T \quad (2.17)$$

$$\alpha_i, m_i \geq 0, \quad i = 1, 2, \dots, n : \text{integer}$$

ここで、 $n$  はサブシステムの数、 $m_i$  は  $i$  番目のサブシステムのユニットの数、 $q_{ie\alpha_i}$  は  $i$  番目のサブシステムの  $\alpha_i$  番目のユニットの故障モード  $e$  の故障率、 $g_{ri\alpha_i}(m_i)$  は  $r$  番目のサブシステムでの資源の必要量で、 $m_i$  に関して非線形で分離可能な関数、 $b_r$  はシステムに利用可能な資源、 $h_i$  は故障モード  $e$  の数である。

## 第3章

# 遺伝的アルゴリズムとハイブリッド化

### 3.1 進化計算

遺伝的アルゴリズムは、生物進化のメカニズムを模倣した最適化アルゴリズムである。適者生存の原理に基づき、個体群を繰り返し選択・交配・変異させることで、最適な解を探索する。この章では、遺伝的アルゴリズムの基本原理と、ハイブリッド化の意義について解説する。

#### 3.1.1 遺伝的アルゴリズムの基本原理

#### 3.1.2 ハイブリッド化の意義

遺伝的アルゴリズムは、大規模な探索空間において、局所最適解に陥りやすいという欠点がある。これを克服するために、局所最適化アルゴリズムと組み合わせたハイブリッド化が有効である。

ハイブリッド化の具体的な手法としては、遺伝的アルゴリズムで大規模な探索を行い、その後、局所最適化アルゴリズムで細かい最適化を行う方法がある。また、遺伝的アルゴリズムの探索過程で、局所最適化アルゴリズムを自動的に起動させるような手法も存在する。このように、遺伝的アルゴリズムと局所最適化アルゴリズムを組み合わせることで、探索の効率性と最適解の精度を向上させることができる。

## 第 3 章

# 遺伝的アルゴリズムとハイブリッド化

### 3.1 まえがき

遺伝的アルゴリズム(GA)とは、1960年代に発表された生物学の遺伝的システム、いわゆる生物進化(自然淘汰・交叉・突然変異)の現象を、コンピュータ・シミュレーションによって模倣する確率的探索・最適化・学習の一手法である。このGAは、自然淘汰と遺伝現象のメカニズムを単純化した数式モデルであり、個体と呼ばれる染色体(対象問題の解候補)の集合が外部環境(対象問題では評価関数)に適応するように、次に示す規則に基づく集団の構成を世代ごとに生成させるものである。

(1) 適合性の高い個体ほど生存確率が高い(自然淘汰)

(2) 古い個体をもとに新しい個体を生成させる(遺伝現象)

GAを組合せ最適化問題の一解法とした場合、規則(1)を確率的探索法、また規則(2)を経験的探索法と見なすことができ、GAは両者の側面をもっている。

GAでは染色体が遺伝情報を伝える役目をしており、これを一次元あるいは二次元のビット又は文字/数値データ列として表現する。これが対象問題での遺伝子表現あるいはコード化である。GAではまず、必要な個数(集団サイズ)だけの染色体の初期集団をランダムに生成させる。この初期集団に対して、遺伝的操作を繰り返し作用させることによって、生存競争のシミュレーション、すなわち対象問題に対する適合性の高い子孫を残すことができる。一般に、GAの進化シミュレーション結果は、染色体集団の初期染色体に強く依存するため、いろいろな初期集団で数値実験を行うことが重要である。

## 3.2 遺伝的アルゴリズムの基礎

### 3.2.1 遺伝的アルゴリズムの概要

GAは生物の進化過程，つまり自然淘汰と遺伝現象を単純化し，工学的にモデル化したものである．したがって，この考え方に基づくものはすべてGAの範疇に入る．実際，問題の特性を十分に考慮した具体的解法が多く存在する．しかし，それらの多くは，最も基本的なGAである単純GA(Simple GA : SGA)<sup>[24, 52]</sup>を基礎とする．SGAの基本的な仕組みは次のとおりである．

(1) 対象問題の解の遺伝子表現 GAでは染色体(chromosome)が遺伝子情報を伝える役目をし，一次元或いは二次元のビットまたは文字／数値データ列として表現される．これが対象問題の遺伝子表現(representation)或いはコード化である．

(2) GAにおける各パラメータ値の設定

集団サイズ(population size) : 集団に含まれる染色体の数

最大世代数(maximum generation) : 進化過程の反復回数

交叉確率(crossover rate) : 交叉操作の発生確率

突然変異確率(mutation rate) : 突然変異操作の発生確率

(3) 初期の染色体集団の生成 初期解として，与えられた集団サイズの染色体をランダムに作成し，初期集団を構築する．

(4) 各染色体の適合度を表す評価関数 初期集団の各染色体に対して，定義された評価関数(evaluation function)により外部環境，すなわち対象問題への適合度を求める．この適応度が高い染色体ほど次世代へ生き残る可能性が高くなる．

(5) 遺伝的操作 次世代を生成するために遺伝的操作(genetic operation)を行う．一般に，遺伝的操作には次に挙げる三種類がある．

選択(selection): 各染色体の評価関数に基づき，適合度の高さに応じて染色体を選択し，次世代の集団を構成するための操作である．

交叉(crossover): 交叉は二つの親(parents)染色体間で，それぞれの部分解を交換する操作である．すなわち，一方の親の良い部分解と他方の親の部分解を組み合わせることにより，さらに良い染色体(これを子(offspring)と呼ぶ)を生成する操作である．

突然変異(mutation): 突然変異は，染色体内の遺伝子の一部を変えることで，染色体集団の多様性を保持するための操作である．

(2)の各パラメータ値は，GAの解探索効率や最終的に得られる解の質に大きく影響を与えるもので，予備実験などにより，各パラメータ値を経験的に決定しておく必要がある．

単純GAの処理の流れを，リスト3.1に示す．工学問題へ適用する場合は，問題に応じて変更・改善が要求されるが，基本的にはリスト3.1の手順に従う．

リスト3.1: GAの手順

```

procedure: genetic algorithms
begin
   $t \leftarrow 0$ ;
  initialize  $P(t)$ ;
  evaluate  $P(t)$ ;
  while (not termination condition) do
    begin
      recombine  $P(t)$  to yield  $C(t)$ ;
      evaluate  $C(t)$ ;
      select  $P(t+1)$  from  $P(t)$  and  $C(t)$ ;
       $t \leftarrow t+1$ ;
    end
  end
end

```

ここで， $P(t)$ は第 $t$ 世代における親染色体， $C(t)$ は第 $t$ 世代における子染色体を表す．

### 3.2.2 遺伝子表現

GAの工学問題への適用において最も重要なことは，遺伝子表現と評価関数をどのように設定するかにある．遺伝子表現は，解くべき問題の解空間を遺伝子表現空間，すなわち探索空間に写像するものであり，探索過程の良否に大きく影響を及ぼす．遺伝子表現では一般に，数値列，ビット列，あるいは文字列を用いるが，解空間内の実行可能解空間が一对一に遺伝子表現空間に写像されるような表現が与えられれば，単純GAをそのまま，或いは極僅かな変更で適用することが可能である．しかし，遺伝子表現空間内の一部の点が実行不可能解に写像される場合には，何らかの変更・改善をGAに施す必要がある．したがって，対象とする問題の構造や性質を十分に把握した上で，遺伝子表現の方法を決めることが重要である．また，解空間を直接遺伝子表現空間に写像できない場合には，解空間を一旦別の空間に写像して，さらに遺伝子表現空間に写像する方法がとられ，前者を直接表現 (direct representation)，後者を間接表現 (indirect representation) という．さらに，一般に一つの個体は一つの染色体から構成されるが，問題によっては複数の染色体で構成される個体を用いる場合もある．

### 3.2.3 評価関数

評価関数は探索方向を与えるもので，最終的に得られる解の質の良否や探索効率に直接関係する．したがって，評価関数の決定は慎重に行う必要がある．工学問

題への適用においては、一般に尺度となり得る物理量（コスト、時間等）を用いる。数学モデルとして定式化される問題では、その目的関数が用いられることが多い。ただし、多目的の場合には単一目的に変換するなどの工夫が必要となる。

ペナルティ関数 (penalty function) は、制約付き最適化問題を解く場合に発生する実行不能解 (infeasible solution) 生成の問題を解決するための技術である。これは、実行不可能な解にペナルティを課すことによって次世代への生存率を低くしていく方法である。しかし、GAでは実行不能解の中に、最適解に近い解が含まれる可能性があることから、実行不能解を各世代に少数を生存させる。

ペナルティ関数を有効に設定する一般的な方法は確立されておらず、対象問題ごとに有効な設定を考える必要がある。

### 3.2.4 遺伝的操作

遺伝的操作については、数多くの提案があるが<sup>[24]</sup>、ここでは本研究で取り扱う遺伝的操作の概要を述べる。

#### 選択 (selection)

選択の方法として、適合度比例法、期待値選択法、ランク選択法、エリート保存法などがある。適合度比例法はルーレット・モデルまたはモンテカルロ・モデルとも呼ばれ、各染色体の適合度に比例した確率で子孫を選ぶ方法である。期待値選択法は染色体集団の中から残す子孫の期待値を計算し、ある染色体が選択された場合にはその期待値から0.5を引く。この方法により、最悪の場合でも期待値0.5の偏差で子孫を残す事が可能となる。ランク選択法は、事前に適合度に基づいて各染色体をランク付けし、各ランクに対して与えられた確率で子孫を残す方法である。また、エリート保存法では、染色体集団の中で適合度の高いものから順に次の世代に残す。この方法は、継続して使用すると、エリート染色体が集団中に急速に広がる可能性が高く、局所解に陥ることがあるため、一般に、他の選択操作と組み合わせて使用する。

#### 交叉 (crossover)

交叉は遺伝子表現に最も影響を受ける操作である。単純(一点)交叉 (simple crossover)、複数点交叉 (multipoint crossover)、一様交叉 (uniform crossover) などがある。多くの工学問題ではこれらの交叉により生成された子染色体は実行不能解となる。したがって、一般にGAの適用にあたり、適切な交叉方法を新たに考えるか、子染色体を実行可能に修正するなどの方策を必要とする。例えば基本的なスケジュー

リング問題の一つである巡回セールスマン問題のような染色体に重複を許さない問題に適するものとして、PMX(partial-mapped crossover)法、OX(order crossover)法、CX(cycle crossover)法、PBX(position-based crossover)法などが提案されている。

リスト3.2において、 $k$ は染色体の番号、 $r_k$ は $k$ の交叉確率、また $V_k$ は染色体を表す。

リスト3.2: 単純交叉の手順

```

procedure: crossover
begin
   $k \leftarrow 0$ ;
  while ( $k \leq 10$ ) do
    begin
       $r_k \leftarrow$  random number from  $[0, 1]$ ;
      if ( $r_k < 0.25$ ) then
        select  $v_k$  as one parent for crossover;
       $k \leftarrow k + 1$ ;
    end
  end

```

#### 突然変異 (mutaiton)

突然変異は遺伝子表現に依存して決定する。一般に、遺伝子表現にビット列を用いた場合には、任意の位置の遺伝子ビットを反転させたり、数値や文字を用いた場合には、任意の位置の遺伝子を前後の遺伝子や別の任意位置の遺伝子と入れ換える。しかし、染色体の実行可能性に大きく関わるため、問題に適した方法をとる必要がある。例えば、巡回セールスマン問題のような要素の重複を許さない問題に対する突然変異法として、逆位 (inversion)、挿入 (insertion)、転座 (displacement)、重複 (duplication) などがある。

### 3.3 非線形最適化問題への応用

#### 3.3.1 制約条件のない非線形最適化問題

非線形最適化問題の例として、次に示すような非線形関数の最小化を考える。複数の局所解をもつ多峰性関数の最適化問題となることから、従来の解析的手法では最適解が得られないことが多い。

$$\max f(x_1, x_2) = -c_1 \exp \left( -c_2 \sqrt{\frac{1}{2} \sum_{j=1}^2 x_j^2} \right) - \exp \left( \frac{1}{2} \sum_{j=1}^2 \cos(c_3 x_j) \right) + c_1 + e$$

$$-3 \leq x_j \leq 3, \quad j = 1, 2$$

ここで  $c_1 = 20$ ,  $c_2 = 0.2$ ,  $c_3 = 2\pi$ , 及び  $e = 2.71282$

また, 関数が複雑で線形近似が不可能な場合には, GAのようなヒューリスティック手法で非線形最適化問題の近似解や準最適解を求めることが要求される.

一般に,  $n$ 個の決定変数の非線形最適化問題は, 次のように定式化される.

$$\begin{aligned} \max \quad & f(\mathbf{x}) \\ & x_i \in D_i, \quad i = 1, 2, \dots, n \\ & D_i = [a_i, b_i] \subseteq R \end{aligned}$$

ここで,  $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$ ,  $R$ は実数値で,  $D_i$ は変数 $x_i$ の範囲 $[a_i, b_i]$ を示す.

各変数の計算に必要なビット数 $m_i$ 及び, 総ビット数 $m$ を計算する. ただし, 有効桁数を6桁とする.

$$2^{m_i-1} < (b_i - a_i) \cdot 10^6 \leq 2^{m_i} - 1, \quad i = 1, 2, \dots, n$$

$$m = \sum_{i=1}^n m_i$$

$m$ ビットの染色体 $V_k$ , ( $k = 1, 2, \dots, pop\_size$ )を作成する. ( $pop\_size$ :集団サイズ)

(例)  $(01101001110101001)_2$

このとき最初のビットから $m_1$ 番目のビットまでを1つのグループとし,  $m_i$ 個ずつの $n$ 個のグループから構成されると考える. また, それぞれのグループは変数 $x_i$ に対応し, 次式より変数 $x_i$ を求めることができる.

$$x_i = a_i + (010 \dots 01) \cdot \frac{b_i - a_i}{2^{m_i} - 1}, \quad i = 1, 2, \dots, n$$

非線形最適化問題における評価関数 $eval(V_k)$ , ( $k = 1, 2, \dots, pop\_size$ )は, 与えられた関数 $f(\mathbf{x}_k)$ をそのまま定義する.

$$eval(V_k) = f(\mathbf{x}_k), \quad k = 1, 2, \dots, pop\_size$$

したがって, この評価関数の値が大きい染色体 $V^*$ ほど適合度が高いことになり, 次式で表される.

$$V^* = \underset{V_k}{\operatorname{argmax}} \{ eval(V_k) \mid k = 1, 2, \dots, pop\_size \}$$

ここで,  $\operatorname{argmax}$ の $\operatorname{arg}$ は $\operatorname{argument}$ の省略形であり, 関数 $eval(V_k)$ の最大値の時の引数 $V_k$ , すなわち $V^*$ を採用することを意味する.

### 3.3.2 0-1 非線形計画問題

$m$  個の非線形なシステム制約条件の下で、非線形な目的関数を最小化する整数計画問題を考える。  $n$  個の 0-1 変数をもつ 0-1 非線形計画問題は、次のように定式化される。

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s. t.} \quad & g_r(\mathbf{x}) \geq 0, \quad r = 1, 2, \dots, T \\ & x_j = 0 \text{ or } 1, \quad j = 1, 2, \dots, n \end{aligned}$$

ここで、 $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$  とする。

$n$  個の 0-1 変数の解集合を、 $n$  ビットからなる 1 つの染色体  $V_k$  ( $k = 1, 2, \dots, pop\_size$ ) として表現する。

$$V_k = [x_1 \ x_2 \ \dots \ x_n]$$

$$\text{(例)} \quad V_k = [010010001001110101]$$

評価関数において、システム制約条件を満たさない染色体に対しペナルティ  $M$  を付加する ( $M$ : 正の大きな数)。

$$eval(V_k) = \begin{cases} f(\mathbf{x}); g_i(\mathbf{x}) \geq 0, \quad i = 1, 2, \dots, m \\ M; g_i(\mathbf{x}) < 0, \quad i = 1, 2, \dots, m \end{cases}$$

したがって、最小化問題の場合には、この評価関数の値が低いものほど適合度が高くなる。

$$V^* = \underset{V_k}{\operatorname{argmin}} \{ eval(V_k) \mid k = 1, 2, \dots, pop\_size \}$$

### 3.3.3 非線形整数計画問題

$n$  個の決定変数と  $T$  個のシステム制約条件を伴う NIP 問題は、次のように定式化される。

$$\begin{aligned} \max \quad & z(\mathbf{x}) \\ \text{s. t.} \quad & g_r(\mathbf{x}) \leq b_r, \quad r = 1, 2, \dots, T \\ & D_j = [l_j, u_j]; \text{ integer,} \\ & j = 1, 2, \dots, n \end{aligned}$$

ここで、 $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]$ で、また  $D_j$ は変数  $x_j$ の範囲を示す。  $b_i$ は  $i$ 番目のシステム制約条件における右辺定数である。

$n$ 個の決定変数の解集合を、 $n$ 要素からなる一つの染色体  $V_k$ , ( $k = 1, 2, \dots, pop\_size$ )として表現する。

$$V_k = [x_{k1} \ x_{k2} \ \cdots \ x_{kn}]$$

### 3.4 ハイブリッド型遺伝的アルゴリズム

#### 3.4.1 ハイブリッド型遺伝的アルゴリズムの概要

GAは最適化問題を解くための効果的なアプローチとして評価されている<sup>[24]</sup>。しかしGAは、一般に最適解周辺には早く近づくが局所探索能力が低いという問題点が指摘されている。この問題点を解決する手段の一つとして、ハイブリッド型GAが提案されている。ハイブリッド型GAは、GAとヒューリスティック手法を組み合わせたもので、大域的な探索をGAによって行い、局所探索をヒューリスティック手法で行うものである。つまりGAによって高速に最適解周辺に近づき、GAの苦手とする局所探索をヒューリスティック手法に行わせる。これによりハイブリッド型GAは、優れた解をより高速に探索することが可能となる。

リスト3.3: ハイブリッド型GAの手順

```

procedure: hybrid GA
begin
   $t \leftarrow 0$ ;
  initialize  $P(t)$ ;
  evaluate  $P(t)$ ;
  while (not termination condition) do
    begin
      recombine  $P(t)$  to yield  $C(t)$ ;
      search  $C(t)$  locally;
      evaluate  $C(t)$ ;
      select  $P(t+1)$  from  $P(t)$  and  $C(t)$ ;
       $t \leftarrow t+1$ ;
    end
  end
end

```

#### 3.4.2 GUB構造に基づくハイブリッド型遺伝的アルゴリズム

これまでに、GAにエキスパートシステム、ニューラルネットワーク、シミュレーテッドアニーリング、タブサーチ等を組み合わせたハイブリッド型GAが提案されているが、GA単独の場合よりも優れた探索能力を示すことが確かめられている

[24, 25, 43]. 本研究では, 前章で取り上げた GUB 構造を伴う大規模な問題の解法として, GUB 構造性を利用した効率的な解の探索を行うヒューリスティクスを GA に組み込んだハイブリッド型 GA を提案する.

次に示す, GUB 構造を伴うナップサック問題 (KP/GUB) を考える.

KP/GUB:

$$\max f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} c_{ij} x_{ij} \quad (3.1)$$

$$\text{s. t. } g(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} a_{ij} x_{ij} \leq b \quad (3.2)$$

$$\sum_{j=1}^{n_j} x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (3.3)$$

$$x_{ij} = 0 \text{ or } 1, \quad \forall i, j \quad (3.4)$$

ここで, 式 (3.2) はナップサック制約, また式 (3.3) は GUB 制約と呼ばれる.  $c_{ij}$  は  $i$  番目の GUB 制約における  $j$  番目の要素に対応する目的関数の係数である. また,  $a_{ij}$  は  $i$  番目の GUB 制約における  $j$  番目の要素に対応するナップサック制約の係数,  $x_{ij}$  は  $i$  番目の GUB 制約における  $j$  番目の要素に対応する決定変数である. さらに,  $n_i$  は  $i$  番目の GUB 制約に含まれる決定変数の数を表す.

本研究で提案するハイブリッド型 GA は, GUB 制約の構造性を利用した解の改善アルゴリズムを, GA に組み込む. すなわち, GA の各世代において, 得られた解集合に対しランクに基づく決定変数の交換を行うことにより, 解の改善を図る. ここで, ランクは GUB 制約の各決定変数について, 制約条件 (コスト) と目的関数 (パフォーマンス) の係数の比, すなわちコストパフォーマンスを表すため, 各世代で得られた解の近傍を局所的に探索し, 改善する方向に探索を進めていくメカニズムを GA に組み込むことにより, 解の改善効果が期待される.

GUB 制約の構造性を利用し, 各 GUB 制約の決定変数に対して次のようなランク付けに基づく並換えを行う.

ステップ 1: 同一集団内において, 各 GUB 制約の決定変数をランク付けするために, 次式に基づいて指標を計算する.

$$e_{ij} = c_{ij} / a_{ij}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n_i \quad (3.5)$$

この指標は, 各決定変数のコストパフォーマンスを表すと考えられることから, 効率指標と呼ぶ.

ステップ2: 次式により, 各GUB制約において指標の高い順に決定変数をランク付けする.

$$\bar{j}_i = [\bar{j}_{i1} \bar{j}_{i2} \cdots \bar{j}_{in_i}] = \underset{i}{\text{indsort}}\{e_{ij} | j = 1, 2, \dots, n_i\}, \forall i \quad (3.6)$$

ここで,  $\text{indsort}$  は降順の並替えを表す.

ステップ3: 各GUB制約において, 効率指標に基づいて各決定変数を降順に並べ換える.

## 第4章

# ファジィ数理計画法と多目的計画法

### 4.1 まえがき

ファジィ理論は、米国カリフォルニア大学のZadehによって1965年に提案されたファジィ集合論を中心として、ファジィ論理とファジィ測度を加えて体系化された。われわれの日常生活で用いられることの多い、“少し大きい”とか“かなり若い”などといった表現にみられるような人間の判断の主観的な側面に基づくあいまいさを定量的に取り扱うことができる理論として提案されたファジィ理論は、オペレーションズ・リサーチ、経営科学、制御理論や人工知能／エキスパート・システムなど、多方面にわたって現実的な問題に応用され、その有用性が示されてきた。ファジィ応用製品も、1970年代後半から実用化され、数多くの製品が市場に送り出されている。

数理計画法への応用としては、1970年、BellmanとZadehが意思決定者のファジィ目標とファジィ制約を統合した決定集合を定義することにより、ファジィ環境における意思決定を試みた。その後、田中らによりはじめて数理計画法に導入された<sup>[115, 116]</sup>のをきっかけに、各種の定式化やその解法が提案されている。近年、乾口らは取り扱われている不確定性の種類から、これらの問題を三種類に大別した<sup>[42, 61]</sup>。すなわち、漠然性を取り扱うフレキシブル計画問題、不明確さを取り扱う可能性計画問題、さらにそれらを同時に取り扱うファジィ数理計画問題である。この漠然性と不明確さを取り扱うファジィ数理計画問題の解法としては、Narasimhan<sup>[59, 60]</sup>やHannan<sup>[28]</sup>などによって提案されたファジィ目標計画法や、坂和<sup>[77, 86]</sup>、Hwang<sup>[32, 45, 46, 47, 48, 49]</sup>などによって提案された対話型ファジィ多目的計画法などがある。

他方、近年、社会的要求の多様化に伴い、従来のような断片的な意思決定問題にとどまらず互いに競合する複数の目的関数を同時に考慮することができる多目的計画法が、現実的な最適化手法として注目されている。多目的計画法は、1951年のKuhn-Tuckerに始まり、その後Ijiri(1965)、Lee(1970)、Zeleny(1974)、Ignizio(1976)、

Zionts(1976), Hwang(1979)らによって取り上げられ、数多くの解法が提案されてきた。さらに、コンピュータの高速化、大容量化の進展に伴い、Zeleny(1982), Chankong(1983), Hwang(1986)らによって新解法の提案や、知識ベースによるエキスパート・システムの構築、さらに生産管理、輸送計画、最適配分、ゲーム理論、環境管理、財務管理、経済などの各分野で、現実問題への応用事例が数多く研究されるようになった[30, 38]。

多目的計画問題では、与えられた制約条件のもとで複数の目的関数値を同時に最適化する完全最適解は一般には存在しないため、一つの目的関数値を改善するためには少なくとも他の一つの目的関数値を改悪せざるを得ない解としてパレート最適解(非優越解)の概念が導入されている。パレート最適解は一般に無限個の点からなる解集合を形成するので、意思決定者は、自己の主観的な価値基準に基づいて解集合の中から一つの解(妥協解)を選択する必要がある。それを補助する手段として、意思決定者の設定する目標にできるだけ近い実行可能解を求めるという目標計画法や、意思決定者との対話を通じて満足解を導出しようという対話型手法が提案されている。

複数の相反する目的関数を同時に考慮する、多目的意思決定(multiple objective decision making: MODM)問題を解くための手法の一つとしてMasudとHwangによって提案されたISGP(interactive sequential goal programming)<sup>[51]</sup>は、目標計画法に基づく会話型の解法で、その実用性の点から評価されている。最近、このISGPの欠点であった、得られた非優越解の中から最良な妥協解を見つけだすことが意思決定者にとって最大の課題であるという問題を、MDI(method of displaced ideal)を導入することによって解決した、ISGPの新しい解法が提案されている<sup>[15]</sup>。しかし、この解法は、目標値の達成度に対する意思決定者のあいまいさについて考慮されていなかったほか、各目標に対する重みづけの方法について示されていない。

本章では、このような観点から、ISGPをさらに発展させたファジィ多目的計画問題の会話型解法を提案する。提案解法は、会話型の目標計画法に基づく実用的な手法で、意思決定者の目標値に対する満足度のあいまいさをメンバシップ関数によって表すほか、右辺定数にゆとりをもたせるファジィ制約、さらに、各目的関数に対する意思決定者の重要度を表すための重みづけに、多評価意思決定のための支援手法として評価されているAHP(analytic hierarchy process)を用いることにより、意思決定者の選好構造をより反映させることを可能とした<sup>[87, 88, 89]</sup>。

## 4.2 ファジィ理論と数理計画法

### 4.2.1 ファジィ集合とファジィ関係

人間の判断の主観的側面におけるあいまいさを定量的に解析するために、Zadehによって提案されたファジィ集合 (fuzzy set) は、メンバシップ関数 (membership function) により、次のように定義される。

[定義 4.1] ファジィ集合  $\tilde{A}$  とは、全体集合  $X$  の要素  $x$  が  $\tilde{A}$  に属する程度をメンバシップ関数  $\mu_{\tilde{A}}(x)$  によって特性づけられる  $x$  の集まりであり、次のように表される。

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}$$

$\mu_{\tilde{A}}$  の値が 1 に近ければ  $x$  の  $\tilde{A}$  に属する度合いが大きく、反対に、0 に近ければ  $x$  の  $\tilde{A}$  に属する度合いが小さいことを示している。

ファジィ集合の  $\alpha$ -レベル集合は、次のような通常の集合である。

[定義 4.2] ファジィ集合  $\tilde{A}$  の  $\alpha$ -レベル集合 ( $\alpha$ -level set) を  $A_\alpha$  で表し、これを次のように定義する。

$$A_\alpha = \{x \in X \mid \mu_{\tilde{A}}(x) \geq \alpha\}; \quad 0 \leq \alpha \leq 1$$

$\alpha$ -レベル集合を用いて、ファジィ集合の凸性、有界性および正規性を次のように定義する。

[定義 4.3] ファジィ集合  $\tilde{A}$  が凸であるとは、すべての  $\alpha \in (0, 1]$  に対して、その  $\alpha$ -レベル集合  $A_\alpha$  が凸集合であることである。この定義は、等価的に次のように表現することができる。

ファジィ集合  $\tilde{A}$  が凸である必要十分条件は、任意の  $x_1 \in X, x_2 \in X$  と  $0 \leq \lambda \leq 1$  であるような任意の  $\lambda$  に対して、

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

が成り立つことである。

[定義 4.4] 全体集合  $X$  が距離空間であるとき、ファジィ集合  $\tilde{A}$  が有界であるとは、すべての  $\alpha \in (0, 1]$  に対して  $A_\alpha$  が有界となることである。

[定義 4.5] ファジィ集合  $\tilde{A}$  が正規であるとは、 $A_1$  が空集合  $\phi$  でないことである。

" $a$  と  $b$  は等しい" というような明確な関係に対して、" $a$  と  $b$  はよく似ている"、あるいは " $a$  はだいたい  $b$  以上" というようなあいまいな関係は、ファジィ関係として定義されている<sup>[81]</sup>。このような通常の関係の一般化としてのファジィ関係は、ファジィ

理論の応用において非常に重要な概念である。ここで、本研究において取り扱うファジィ目標およびファジィ制約の基礎となるファジィ関係の定義とその基本的な性質について述べる。

[定義 4.6] 直積  $X \times Y = \{(x, y) \mid x \in X, y \in Y\}$  におけるファジィ関係 (fuzzy relation)  $R$  は,

$$\mu_R : X \times Y \rightarrow [0, 1]$$

なるメンバシップ関数  $\mu_R$  によって特性づけられた  $X \times Y$  におけるファジィ集合  $R$  であり, メンバシップ関数  $\mu_R(x, y)$  は,  $x$  と  $y$  の関係の度合いを表す。

直積  $X \times Y$  におけるファジィ関係は,  $X \times Y$  におけるファジィ集合であるから, メンバシップ関数により定義されるファジィ集合における基本的演算と同様に, 次のように定義される。

[定義 4.7] 部分集合:  $X \times Y$  におけるファジィ関係  $R, S$  に対して,  $R$  が  $S$  の部分集合であることを  $R \subseteq S$  と表し, 次のように定義する。

$$R \subseteq S \Leftrightarrow \mu_R(x, y) \leq \mu_S(x, y), \quad \forall (x, y) \in X \times Y$$

[定義 4.8] 共通集合: 二つのファジィ関係  $R, S$  の共通集合を  $R \cap S$  と表し, 次のように定義する。

$$R \cap S \Leftrightarrow \mu_{R \cap S}(x, y) = \min(\mu_R(x, y), \mu_S(x, y))$$

[定義 4.9] 和集合: 二つのファジィ関係  $R, S$  の和集合を  $R \cup S$  と表し, 次のように定義する。

$$R \cup S \Leftrightarrow \mu_{R \cup S}(x, y) = \max(\mu_R(x, y), \mu_S(x, y))$$

[定義 4.10] 補関係: ファジィ関係  $R$  の補集合を  $\bar{R}$  と表し, 次のように定義する。

$$\bar{R} \Leftrightarrow \mu_{\bar{R}}(x, y) = 1 - \mu_R(x, y)$$

[定義 4.11] 逆ファジィ関係: 関係  $R$  の逆ファジィ関係を  $R^{-1}$  と表し, 次のように定義する.

$$R^{-1} \Leftrightarrow \mu_{R^{-1}}(y, x) = \mu_R(x, y)$$

[定義 4.12] 代数積: 二つのファジィ関係  $R, S$  の代数積を  $RS$  と表し, 次のように定義する.

$$RS \Leftrightarrow \mu_{RS}(x, y) = \mu_R(x, y)\mu_S(x, y)$$

[定義 4.13] 代数和: 二つのファジィ関係  $R, S$  の代数和を  $R+S$  と表し, 次のように定義する.

$$R+S \Leftrightarrow \mu_{R+S}(x, y) = \mu_R(x, y) + \mu_S(x, y) - \mu_R(x, y)\mu_S(x, y)$$

[定義 4.14] 限界積: 二つのファジィ関係  $R, S$  の限界積  $R \odot S$  と表し, 次のように定義する.

$$R \odot S \Leftrightarrow \mu_{R \odot S}(x, y) = \max(0, \mu_R(x, y) + \mu_S(x, y) - 1)$$

[定義 4.15] 限界和: 二つのファジィ関係  $R, S$  の限界和  $R \oplus S$  と表し, 次のように定義する.

$$R \oplus S \Leftrightarrow \mu_{R \oplus S}(x, y) = \min(1, \mu_R(x, y) + \mu_S(x, y))$$

## 4.2.2 ファジィ数理計画法

### (1) ファジィ環境における意思決定

1970年, Bellman と Zadeh は, ファジィ環境における意思決定として, 代替案の集合  $X$  上にファジィ目標 (fuzzy goal) とファジィ制約 (fuzzy constraint) が与えられた場合のファジィ意思決定 (fuzzy decision)  $D$  を, ファジィ目標  $G$  とファジィ制約  $C$  を同時に満たす

ことを考慮して、ファジィ目標  $G$  とファジィ制約  $C$  との共通集合であると定義した。すなわちファジィ決定  $D$  は

$$D = G \cap C \quad (4.1)$$

であると定義され、そのメンバシップ関数

$$\mu_D(\mathbf{x}) = \min(\mu_G(\mathbf{x}), \mu_C(\mathbf{x})) \quad (4.2)$$

で特性づけられる。

ここで、代替案とは決定にあたってとりうる手段や行動のことであり、また、ファジィ目標  $G$  とファジィ制約  $C$  はそれぞれのメンバシップ関数

$$\mu_G: X \rightarrow [0, 1] \quad (4.3)$$

$$\mu_C: X \rightarrow [0, 1] \quad (4.4)$$

によって特性づけられる代替案の集合  $X$  上のファジィ集合である。

さらにより一般の複数のファジィ目標と複数のファジィ制約が存在する場合への拡張は、容易に行うことができる。いま、 $G_1, G_2, \dots, G_q$  をファジィ目標とし、 $C_1, C_2, \dots, C_m$  をファジィ制約とすれば、ファジィ決定  $D$  は

$$D = G_1 \cap G_2 \cap \dots \cap G_q \cap C_1 \cap C_2 \cap \dots \cap C_m \quad (4.5)$$

と定義され、

$$\begin{aligned} \mu_D(\mathbf{x}) &= \min_{\substack{1 \leq k \leq q \\ 1 \leq j \leq m}} (\mu_{G_k}(\mathbf{x}), \mu_{C_j}(\mathbf{x})) \\ &= \min(\mu_{G_1}(\mathbf{x}), \mu_{G_2}(\mathbf{x}), \dots, \mu_{G_q}(\mathbf{x}), \mu_{C_1}(\mathbf{x}), \mu_{C_2}(\mathbf{x}), \dots, \mu_{C_m}(\mathbf{x})) \end{aligned} \quad (4.6)$$

なるメンバシップ関数で特性づけられる。

ファジィ決定  $D$  における意思決定としては、 $D$  に帰属する度合を最大にするような  $\mathbf{x}$  を選ぶという最大化決定 (maximizing decision) が Bellman と Zadeh により提案されている。すなわち最大化決定とは、ファジィ決定  $D$  のメンバシップ関数  $\mu_D(\mathbf{x})$  の値を最大化するような  $\mathbf{x}$  を選ぶことであり、

$$\mu_D(\mathbf{x}^*) = \max_{\mathbf{x} \in X} \mu_D(\mathbf{x}) \quad (4.7)$$

となるような  $\mathbf{x}^*$  を求めるものである。

一般に、 $q$  個のファジィ目標  $G_1, G_2, \dots, G_q$  と  $m$  個のファジィ制約  $C_1, C_2, \dots, C_m$  を考える場合は、

$$\begin{aligned} \mu_D(\mathbf{x}^*) &= \max_{\mathbf{x} \in X} \min_{\substack{1 \leq k \leq q \\ 1 \leq j \leq m}} (\mu_{G_k}(\mathbf{x}), \mu_{C_j}(\mathbf{x})) \\ &= \max_{\mathbf{x} \in X} \{ \min(\mu_{G_1}(\mathbf{x}), \mu_{G_2}(\mathbf{x}), \dots, \mu_{G_q}(\mathbf{x}), \mu_{C_1}(\mathbf{x}), \mu_{C_2}(\mathbf{x}), \dots, \mu_{C_m}(\mathbf{x})) \} \end{aligned} \quad (4.8)$$

となる $x^*$ を選ぶことによってファジィ決定 $D$ に対する最大化決定がなされる。

Bellman と Zadeh はファジィ目標 $G$ とファジィ制約 $C$ とを統合するファジィ決定の定義としてこのほかに凸ファジィ決定や積ファジィ決定などを提案した。凸ファジィ決定 (convex fuzzy decision) は, ファジィ目標 $G_1, G_2, \dots, G_q$  とファジィ制約 $C_1, C_2, \dots, C_m$  に対し,

$$\left. \begin{aligned} \mu_D^{co}(x) &= \sum_{k=1}^q \alpha_k \mu_{G_k}(x) + \sum_{j=1}^m \beta_j \mu_{C_j}(x) \\ \sum_{k=1}^q \alpha_k + \sum_{j=1}^m \beta_j &= 1 \\ \alpha_k &\geq 0 \quad k=1, 2, \dots, q \\ \beta_j &\geq 0 \quad j=1, 2, \dots, m \end{aligned} \right\} \quad (4.9)$$

また, 積ファジィ決定 (product fuzzy decision) を次のように定義した。

$$\mu_D^{pr}(x) = \left( \prod_{k=1}^n \mu_{G_k}(x) \right) \cdot \left( \prod_{j=1}^m \mu_{C_j}(x) \right) \quad (4.10)$$

これらの凸ファジィ決定や積ファジィ決定についても, ファジィ決定 $D$ に対する最大化決定の場合と同様に, それぞれ,

$$\mu_D^{co}(x^*) = \max_{x \in X} \left\{ \sum_{i=1}^n \alpha_i \mu_{G_i}(x) + \sum_{j=1}^m \beta_j \mu_{C_j}(x) \right\} \quad (4.11)$$

$$\mu_D^{pr}(x^*) = \max_{x \in X} \left\{ \left( \prod_{i=1}^n \mu_{G_i}(x) \right) \cdot \left( \prod_{j=1}^m \mu_{C_j}(x) \right) \right\} \quad (4.12)$$

となるような $x^*$ を選ぶという最大化決定が定義されている。

ここで, これらの $\mu_D^{co}(x), \mu_D^{pr}(x)$  と $\mu_D(x)$  との間には明らかに

$$\mu_D^{pr}(x) \leq \mu_D(x) \leq \mu_D^{co}(x) \quad (4.13)$$

なる関係が成立する。

## (2) ファジィ目標とファジィ制約

従来 of 目標計画法では, 意思決定者は目標値を正確に与える必要があった。しかし, 現実の意思決定問題では, 目標値は”一定計画期間中の目標利益はだいたい1000万円くらい”とか, ”今月の生産量の目標はおよそ2000万トン以上”といった, あいまいな値として与えられる場合が多い。こうした人間の判断のあいまいさを

扱うには、意思決定者の目標値の達成度に対する満足の度合いをメンバシップ関数を導入することによって評価する、ファジィ目標が有効である。

さらに、従来の数理計画法では、例えば生産計画問題を定式化する場合、システム資源に関する制約は”材料1の期間使用量の制限は100kg以下”など、右辺定数を確定値として与えられてきた。ところが、現実の設計問題ではシステム資源の制約は確定値として与えるよりも、あいまいな値として与えるほうがより実用的な場合が多く、また、システムの使用環境の変化や経済情勢の変動により、資源の制約量は変化することも多い。このようなシステム資源の変動をシステム設計段階から考慮するためには、ファジィ制約が有効となる。このようなファジィ目標およびファジィ制約を伴う線形計画問題は、1976年に Zimmermann によって最初に提案された。Zimmermann は、次のような通常の線形計画問題

$$\begin{aligned} \max \quad & f(\mathbf{x}) \\ \text{s. t.} \quad & A\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (4.14)$$

に対して、目的関数を”だいたい  $z_0$  以上”，また制約条件を”だいたい  $b$  以下”にしたいというファジィ目標およびファジィ制約を導入した問題を、次のように表した。

$$\begin{aligned} \max \quad & f(\mathbf{x}) \lesssim h \\ \text{s. t.} \quad & A\mathbf{x} \gtrsim \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (4.15)$$

ここで、 $\lesssim$  は、”だいたい目標値  $h$  以上にしたい”，また  $\gtrsim$  は ”だいたい  $b$  以下にしたい” ということを表している。Zimmermann は、ファジィ目標とファジィ制約は決定に対して同じ役割を果たすと考え、目標と制約をまとめて次のように表した。

$$\begin{aligned} B\mathbf{x} &\gtrsim \mathbf{b}' \\ \mathbf{x} &\geq \mathbf{0} \\ B &= [\mathbf{c} \ A]^T, \quad \mathbf{b}' = [z_0 \ \mathbf{b}]^T \end{aligned} \quad (4.16)$$

Zimmermann はさらに、このような意思決定者のあいまいさを表すために、ファジィ不等式  $\mathbf{x} \geq \mathbf{0}$  に対し、次のようなメンバシップ関数を導入した。

$$\mu_r((B\mathbf{x})_r) = \begin{cases} 1 & ; \quad (B\mathbf{x})_r \geq b'_r \\ 1 - \frac{b'_r - (B\mathbf{x})_r}{d_r} & ; \quad b'_r - d_r \leq (B\mathbf{x})_r \leq b'_r \\ 0 & ; \quad (B\mathbf{x})_r < b'_r - d_r \end{cases} \quad (4.17)$$

ここで、 $d_r$ は、 $r$ 番目の制約条件の右辺定数に対する許容量を表す。このメンバシップ関数を図示すると、図4.1のようになる。

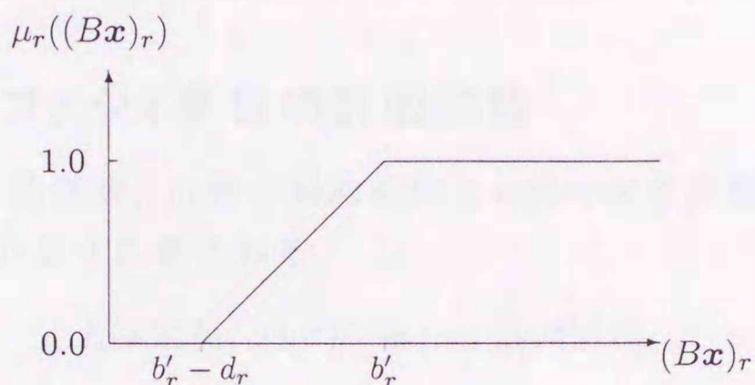


図 4.1. 線形のメンバシップ関数

このファジィ線形計画問題は、BellmanとZadehのファジィ決定に対する最大化決定を導入することにより、次の問題において、 $\mathbf{x}^*$ を求めることになる。

$$\mu_D(\mathbf{x}^*) = \max_{\mathbf{x} \geq 0} \min_{0 \leq r \leq T} \{\mu_r((B\mathbf{x})_r)\} \quad (4.18)$$

## 4.3 多目的計画法

### 4.3.1 多目的計画問題

多目的計画問題は、与えられた制約条件の下で、複数個の互いに相反する目的関数を何らかの意味で最適化するベクトル最小化問題として、次のように表される。

$$\min \quad f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_q(\mathbf{x})) \quad (4.19)$$

$$\text{s. t.} \quad \mathbf{x} \in X = \{\mathbf{x} \in E^n \mid g_r(\mathbf{x}) \leq 0, \quad r = 1, 2, \dots, T\} \quad (4.20)$$

ここで、 $f_1(\mathbf{x}), \dots, f_q(\mathbf{x})$ は $q$ 個の相競合する目的関数、 $\mathbf{x}$ は $n$ 次元の決定変数ベクトル、 $g_1(\mathbf{x}), \dots, g_T(\mathbf{x})$ は $T$ 個の制約条件、また $X$ は制約領域を表す。

多目的計画問題において、 $q$ 個すべての目的関数を同時に最小化する解、すなわち完全最適化は一般には存在しない。そのため、ある目的関数の値を改善するためには、他の目的関数値を犠牲にせざるを得ないという、いわゆるトレード・オフの関係にある複数個の解の集合としてパレート最適解が導入される。完全最適解およびパレート最適解は、次のように定義される。

[定義 4.16] すべての  $x \in X$  に対して  $f_k(x^*) \leq f_k(x)$ , ( $k = 1, 2, \dots, q$ ) となる  $x^* \in X$  が存在するとき,  $x^*$  を完全最適解であるという.

[定義 4.17]  $x^* \in X$  に対して  $f(x^*) \leq f(x)$  となる  $x$  が存在しないとき,  $x^*$  をパレート最適解であるという.

### 4.3.2 ファジィ多目的計画問題

$q$  個の目的関数,  $m$  個の制約条件と  $n$  個の決定変数からなるファジィ多目的計画問題は, 次のように表される.

$$\min \quad f_1(x) \lesssim h_1, f_2(x) \lesssim h_2, \dots, f_q(x) \lesssim h_q \quad (4.21)$$

$$\text{s. t.} \quad g_r(x) = \sum_{t=1}^p a_{rt} x_t \bar{\approx} b_r, \quad r = 1, 2, \dots, T_1 \quad (4.22)$$

$$g_r(x) = \sum_{t=1}^p a_{rt} x_t \gtrsim b_r \quad r = T_1 + 1, \dots, T \quad (4.23)$$

$$x_t \geq 0, \quad t = 1, 2, \dots, p \quad (4.24)$$

ここで,  $f_k(x)$  は,

$$f_k(x) = \sum_{t=1}^p c_{kt} x_t, \quad k = 1, 2, \dots, q \quad (4.25)$$

で表される  $k$  番目の目的関数,  $h_k$  は意思決定者が設定する  $k$  番目の目的関数の目標値で, 最悪値 (negative ideal solution: NIS)  $z_k^-$  と理想値 (positive ideal solution: PIS)  $z_k^*$  の間の値をとる. 記号  $\bar{\approx}$  および  $\gtrsim$  はあいまいさ (大体等しい, および大体等しいか小さい) を表すファジィ等号および不等号である. 式 (4.22) と (4.23) は, 意思決定者によって右辺定数  $b_r$  に上限と下限が与えられるファジィシステム制約, また  $c_{kt}$  は  $k$  番目の目的関数の  $t$  番目の係数,  $a_{rt}$  は  $r$  番目の制約条件の  $t$  番目の係数, および  $x_t$  は  $t$  番目の決定変数である.

ここで, 目的関数  $f_k(x)$  に対する達成度を表すメンバシップ関数  $\mu_k$  およびシステム制約  $g_r(x)$  に対する線形型のメンバシップ関数  $\mu_{q+i}$  をそれぞれ次のように定義する.

$$\mu_k = \begin{cases} 1 & ; \quad f_k(x) < h_k \\ \frac{f_k(x) - z_k^-}{h_k - z_k^-} & ; \quad h_k \leq f_k(x) \leq z_k^- \\ 0 & ; \quad f_k(x) > z_k^- \end{cases} \quad k = 1, 2, \dots, q \quad (4.26)$$

$$\mu_{q+r} = \begin{cases} 0 & ; \quad g_r(x) < b_r - \delta_r \\ \frac{g_r(x) - b_r + \delta_r}{\delta_r} & ; \quad b_r - \delta_r \leq g_r(x) < b_r \\ 1 & ; \quad g_r(x) = b_r \\ \frac{b_r + \delta_r - g_r(x)}{\delta_r} & ; \quad b_r < g_r(x) \leq b_r + \delta_r \\ 0 & ; \quad g_r(x) > b_r + \delta_r \end{cases} \quad r = 1, 2, \dots, T_1 \quad (4.27)$$

$$\mu_{q+r} = \begin{cases} 0 & ; g_r(\mathbf{x}) < b_r - \delta_r \\ \frac{g_r(\mathbf{x}) - b_r + \delta_r}{\delta_r} & ; b_r - \delta_r \leq g_r(\mathbf{x}) < b_r \\ 1 & ; g_r(\mathbf{x}) \geq b_r \end{cases} \quad r = T_1 + 1, \dots, T \quad (4.28)$$

ここで、 $\delta_r$ は意思決定者が設定するシステム制約 $r$ の右辺定数 $b_r$ の幅である。

さらに、AHPによって各目的関数 $f_k(\mathbf{x})$ およびシステム制約 $g_r(\mathbf{x})$ の重要度に対応する重み $w_k$ を設定する。AHPを用いて各目的関数およびシステム制約に対する一対比較を行うには、表4.1のように定義されるスケールが用いられる。

表 4.1 一対比較のための重要度スケール

重要度	定義(前の項目が後に対して)
1	同等
3	やや同等
5	重要
7	かなり重要
9	はるかに重要
2,4,6,8	補間的に用いる
上の数値の逆数	後の項目から見た場合に用いる

このスケールを用いて、各目的関数およびシステム制約に対する一対比較を行い、次のような一対比較行列 $D$ を作成する。

$$D = \begin{bmatrix} 1 & d_{12} & \cdots & d_{1,q+T} \\ d_{21} & 1 & \cdots & d_{2,q+T} \\ \cdots & \cdots & \cdots & \cdots \\ d_{q+T,1} & d_{q+T,2} & \cdots & 1 \end{bmatrix}$$

ここで、 $d_{ij} = 1/d_{ji}$ である。重み $w_k$ は次式により近似的に計算される。

$$w_k = \left( \prod_{l=1}^{q+T} d_{kl} \right)^{1/(q+T)} / \sum_{k=1}^{q+T} \left( \prod_{l=1}^{q+T} d_{kl} \right)^{1/(q+T)} \quad k = 1, 2, \dots, q+T \quad (4.29)$$

目的関数の型が最大化問題の場合、式(4.26)で表されるメンバシップ関数のかわりに次のメンバシップ関数を適用する。

$$\mu_k = \begin{cases} 1 & ; f_k(\mathbf{x}) > h_k \\ \frac{f_k(\mathbf{x}) - z_k^-}{h_k - z_k^-} & ; z_k^- \leq f_k(\mathbf{x}) \leq h_k \\ 0 & ; f_k(\mathbf{x}) < z_k^- \end{cases} \quad k = 1, 2, \dots, q \quad (4.30)$$

意思決定者は、一般に無限個の解集合を形成するパレート最適解の中から自己の選好構造に基づいて一つの妥協解を選択しなければならない。しかし、意思決

定者の選好構造を十分に表現する選好関数は、本来未知のものであり、同定することが困難である。このような状況に対して、選好関数のかわりに、意思決定者との対話を通じて得られる局所的な選好情報に基づいて選好解を導出する、会話型の手法が有効である。会話型手法は、1972年に Geoffrion, Dyerらによって提案された対話型 Frank-Wolfe 法を始めとして、Zeleny(1974), Haimes(1974), Wierzbicki(1980), Hwang(1981), Sakawa(1982)らにより、様々な会話型手法が提案され、発展してきた。近年では、従来の会話型意思決定手法の拡張として、多目的計画問題の定式化における専門家の判断のあいまいさや、意思決定者のファジィ目標を考慮した新しい会話型意思決定法が提案されつつある。

## 4.4 ファジィ多目的計画問題の会話型解法

本章では、ISGPをさらに発展させたファジィ多目的計画問題の会話型の解法を提案する。提案解法は、意思決定者の選好構造をより反映させることができる会話型の目標計画法に基づくもので、意思決定者の目標値に対する満足度のあいまいさをメンバシップ関数によって表すほか、右辺定数にゆとりをもたせるファジィ制約、さらに各目的関数に対する意思決定者の重要度を表すための重みづけに AHP を用いている。

### 4.4.1 計算法

本章で提案する、ファジィ多目的計画問題を解くための会話型解法を次に示す。

ステップ 1: 意思決定者の基本的な設計方針を反映させるために、AHPによって各目的関数およびシステム制約条件の重要度  $w_k$ , ( $k=1, 2, \dots, q+T$ ) を計算する。

ステップ 2: 各システム制約に対する右辺定数の幅  $\delta_r$ , ( $r=1, 2, \dots, T$ ) を設定する。

ステップ 3:  $q$ 個の目的関数のそれぞれについて、次に示す単一目的の線形計画問題

$$\min f_k(\mathbf{x}) = \sum_{j=1}^n c_{kj}x_j \quad (4.31)$$

$$\text{s. t. } g_r(\mathbf{x}) = \sum_{t=1}^p a_{rt}x_t \geq b_r - \delta_r, \quad r=1, 2, \dots, T_1 \quad (4.32)$$

$$g_r(\mathbf{x}) = \sum_{t=1}^p a_{rt}x_t \leq b_r + \delta_r, \quad r=T_1+1, \dots, T \quad (4.33)$$

$$x_t \geq 0, \quad t=1, \dots, p \quad (4.34)$$

を解き、PISを求める。このLP問題でのシステム制約は、最も緩い条件で理想解を得るために、式(4.32)および(4.33)とする。

さらに、目的関数の型 (min / max) を反転させた  $q$  個の単一目的の線形計画問題を解き、NIS を求める。

ステップ 4 : 各目的関数に対する目標値  $h_k$  を、次に示す条件

$$z_k^* \leq h_k < z_k^-, \quad k = 1, 2, \dots, q \quad (4.35)$$

を満足する範囲で意思決定者が設定する。

ステップ 5 : 主問題 P(principle) 問題と補助問題 A(alternative) 題の目的関数を、式 (4.26)~(4.28) で定義されたメンバシップ関数を用いてそれぞれ次のように設定する。

$$z_0 = \sum_{k=1}^q w_k \mu_k + \sum_{i=1}^m w_{q+i} \mu_{q+i} \quad (4.36)$$

$$z_t = \sum_{\substack{k=1 \\ k \neq k'}}^q w_k \mu_k + \sum_{r=1}^m w_{q+r} \mu_{q+r}, \quad k' = 1, 2, \dots, q \quad (4.37)$$

ステップ 6 : 次に示す P 問題を解き、最適解  $x_0$  とそのときの目的関数  $f_0 = [f_{10}, f_{20}, \dots, f_{q0}]$  を得る。

$$\max z_0$$

s. t.

$$f_k(\mathbf{x}) + (z_k^- - h_k) \mu_k = z_k^-, \quad k = 1, 2, \dots, q \quad (4.38)$$

$$g_r(\mathbf{x}) - \delta_r \mu_{q+r} = b_r - \delta_r, \quad r = 1, 2, \dots, T_1 \quad (4.39)$$

$$g_r(\mathbf{x}) + \delta_r \mu_{q+r} = b_r + \delta_r, \quad r = T_1 + 1, 2, \dots, T \quad (4.40)$$

$$x_t \geq 0, \quad t = 1, 2, \dots, p \quad (4.41)$$

ステップ 7 :  $q$  個の A 問題を解き、その最適解  $x'_k$  と目的関数値  $f'_k = [f_{1k'}, f_{2k'}, \dots, f_{qk'}]$  を得る。

$$\max z'_k$$

s. t.

$$f'_k(\mathbf{x}) = h'_k \quad (4.42)$$

$$f_k(\mathbf{x}) + (z_k^- - h_k) \mu_k = z_k^-, \quad k = 1, 2, \dots, q, \quad k \neq k' \quad (4.43)$$

式 (4.39), (4.40) および (4.41)

ステップ8: TOPSISにより, 意思決定者に提示する解を計算する. 重みつきユークリッド距離を次式によって求める.

$$\begin{aligned} s_u^* &= \sqrt{\sum_{k=1}^q w_k^2 (r_{ku} - r_k^*)^2}, \quad u = 1, \dots, v \\ s_u^- &= \sqrt{\sum_{k=1}^q w_k^2 (r_{ku} - r_k^-)^2}, \quad u = 1, \dots, v \end{aligned} \quad (4.44)$$

ここで,  $r_{ku}$ は正規化された $f_{ku}$ , すなわち,

$$r_{ku} = \frac{f_{ku}}{\sqrt{\sum_{u=1}^v (f_{ku})^2 + (z_k^*)^2 + (z_k^-)^2}} \quad (4.45)$$

であり,  $r_k^*$ ,  $r_k^-$ はそれぞれ上式で正規化された $z_k^*$ ,  $z_k^-$ , また $v$ は得られた非優越解の数である. 次式からPISにできるだけ近い点を求める.

$$c_u^* = \frac{s_u^-}{s_u^* - s_u^-} \quad (4.46)$$

ステップ9: 各目的関数値の達成度(目標の満足度) $\mu_k, k=1, \dots, q$ が意思決定者の希望する値より大きければ, 得られた解を最良な妥協解として終了する. そうでなければ, 各目的関数の重要度 $w_k (k=1, 2, \dots, q)$ を再設定してステップ5に戻る.

なお, ステップ8で用いられるTOPSIS(technique for order preference by similarity to ideal solution)は, 多くの非優越解の中から, 最良な妥協解を見つけるための意思決定の補助手段として導入されるもので, PISにできるだけ近い点を計算する[30, 34, ?].

#### 4.4.2 数値例

数値例として, MasudとHwangがISGPに用いた多目的意思決定問題<sup>[51]</sup>に提案する計算法を適用し, その有効性を確認する.

表 4.2 6種類の食品に関する栄養コスト

	ミルク (pint)	ビーフ (pound)	卵 (dozen)	パン (ounce)	レタス (ounce)	オレンジ (pint)
ビタミン A(i.u.)	720	107	7080	0	134	1000
カロリー (kcal)	344	1460	1040	75	17.4	240
コレステロール (unit)	10	20	130	0	0	0
蛋白質 (g)	18	151	78	2.5	0.2	4
炭水化物 (g)	24	27	0	15	1.1	52
鉄分 (mg)	0.2	10.1	13.2	0.75	0.15	1.2
コスト (\$)	0.225	2.2	0.8	0.1	0.05	0.26

$$\min f_1(\mathbf{x}) = 0.225x_1 + 2.2x_2 + 0.8x_3 + 0.1x_4 + 0.05x_5 + 0.26x_6 \lesssim h_1$$

$$\min f_2(\mathbf{x}) = 10x_1 + 20x_2 + 130x_3 \lesssim h_2$$

$$\min f_3(\mathbf{x}) = 344x_1 + 1460x_2 + 1040x_3 + 75x_4 + 17.4x_5 + 240x_6 \lesssim h_3$$

$$\min f_4(\mathbf{x}) = 24x_1 + 27x_2 + 15x_4 + 1.1x_5 + 52x_6 \lesssim h_4$$

s. t.

$$g_1(\mathbf{x}) = 720x_1 + 107x_2 + 7080x_3 + 134x_5 + 1000x_6 \gtrsim 4800$$

$$g_2(\mathbf{x}) = 18x_1 + 151x_2 + 78x_3 + 2.5x_4 + 0.2x_5 + 4x_6 \gtrsim 176$$

$$g_3(\mathbf{x}) = 0.2x_1 + 10.1x_2 + 13.2x_3 + 0.75x_4 + 0.15x_5 + 1.2x_6 \gtrsim 17.5$$

$$x_1 \leq 6.0$$

$$x_2 \leq 1.0$$

$$x_3 \leq 0.25$$

$$x_4 \leq 10.0$$

$$x_5 \leq 10.0$$

$$x_6 \leq 4.0$$

$$x_1, \dots, x_6 \geq 0$$

ステップ 1 : 三つのシステム制約  $g_1(\mathbf{x})$ ,  $g_2(\mathbf{x})$ ,  $g_3(\mathbf{x})$  に対する右辺定数の幅をそれぞれ次のように設定する.

$$\delta_1 = 100.0, \quad \delta_2 = 6.0, \quad \delta_3 = 5.0$$

ステップ2: 意思決定者の基本的な設計方針を反映させるために, AHPによって各目的関数およびシステム制約に重要度によるランク付けを行う. 次のように一対比較行列が得られたとする.

$$D = \begin{bmatrix} 1 & 3 & 1/3 & 3 & 3 & 1/5 & 5 \\ 1/3 & 1 & 1/5 & 1/3 & 1/3 & 1/7 & 3 \\ 3 & 5 & 1 & 3 & 5 & 1/3 & 7 \\ 1/3 & 3 & 1/3 & 1 & 3 & 1/5 & 5 \\ 1/3 & 3 & 1/5 & 1/3 & 1 & 1/7 & 3 \\ 5 & 7 & 3 & 5 & 7 & 1 & 7 \\ 1/5 & 1/3 & 1/7 & 1/5 & 1/3 & 1/7 & 1 \end{bmatrix}$$

これより, 各目的関数及びシステム制約の重要度に対応する重み  $w_k, k=1, \dots, 7$  が次のように与えられる.

$$w_1 = 0.165, \quad w_2 = 0.041, \quad w_3 = 0.225, \quad w_4 = 0.099, \\ w_5 = 0.072, \quad w_6 = 0.375, \quad w_7 = 0.023$$

ステップ3: 各目的関数について, それぞれ単一目的の線形計画問題として解き, 理想値と最悪値  $z_k^*, z_k^-, k=1, \dots, 4$  を求める.

表4.3 理想値および最悪値

k	$z_k^*$	$z_k^-$
1	2.56	6.14
2	16.99	112.50
3	2139.31	5668.00
4	76.77	540.00

ステップ4: 最初の目標値をそれぞれ次のように設定する.

$$h_1 = 2.78(\text{ドル}), \quad h_2 = 30.0(\text{unit}), \quad h_3 = 2800(\text{Kcal}), \quad h_4 = 250.0(\text{g}),$$

ステップ5: 第1目的関数に対するメンバシップ関数  $\mu_1$  は, 次のように表わされる.

$$\mu_1 = \begin{cases} 1 & ; \quad f_1(x) < 2.78 \\ \frac{6.14 - f_1(x)}{6.14 - 2.78} & ; \quad 2.78 \leq f_1(x) \leq 6.14 \\ 0 & ; \quad 6.14 < f_1(x) \end{cases}$$

同様に，第2, 3, 4目的関数に関するメンバシップ関数を式(4.26), (4.30)より得る．

第1システム制約に対するメンバシップ関数 $\mu_5$ は，次のとおりである．

$$\mu_5 = \begin{cases} 1 & ; g_1(\mathbf{x}) > 4900 \\ \frac{g_1(\mathbf{x}) - 4800}{100} & ; 4800 \leq g_1(\mathbf{x}) \leq 4900 \\ 0 & ; 4800 > g_1(\mathbf{x}) \end{cases}$$

同様に，第2, 3システム制約に対するメンバシップ関数 $\mu_6, \mu_7$ を，式(4.27), (4.28)から得る．目標計画のP問題とA問題の目的関数をそれぞれ次のように設定する．

$$P\text{問題} : z_0 = 0.165\mu_1 + 0.041\mu_2 + 0.225\mu_3 + 0.099\mu_4 \\ + 0.072\mu_5 + 0.375\mu_6 + 0.023\mu_7$$

$$A\text{問題} : z_1 = 0.041\mu_2 + 0.225\mu_3 + 0.099\mu_4 \\ + 0.072\mu_5 + 0.375\mu_6 + 0.023\mu_7$$

$$z_2 = 0.165\mu_1 + 0.225\mu_3 + 0.099\mu_4 \\ + 0.072\mu_5 + 0.375\mu_6 + 0.023\mu_7$$

$$z_3 = 0.165\mu_1 + 0.041\mu_3 + 0.099\mu_4 \\ + 0.072\mu_5 + 0.375\mu_6 + 0.023\mu_7$$

$$z_4 = 0.165\mu_1 + 0.041\mu_3 + 0.225\mu_4 \\ + 0.072\mu_5 + 0.375\mu_6 + 0.023\mu_7$$

ステップ6 : 次に示すP問題を解く．

$$\max \quad z_0 = 0.165\mu_1 + 0.041\mu_2 + 0.225\mu_3 + 0.099\mu_4 \\ + 0.072\mu_5 + 0.375\mu_6 + 0.023\mu_7$$

s. t.

$$0.225x_1 + 2.2x_2 + 0.8x_3 + 0.1x_4 + 0.05x_5 + 0.26x_6 + 3.36\mu_1 = 6.14$$

$$10x_1 + 20x_2 + 130x_3 + 82.5\mu_2 = 112.5$$

$$344x_1 + 1460x_2 + 1040x_3 + 75x_4 + 17.4x_5 + 240x_6 + 2868\mu_3 = 5668$$

$$24x_1 + 27x_2 + 15x_4 + 1.1x_5 + 52x_6 + 290\mu_4 = 540$$

$$720x_1 + 107x_2 + 7080x_3 + 134x_5 + 1000x_6 - 100\mu_5 = 4800$$

$$18x_1 + 151x_2 + 78x_3 + 2.5x_4 + 0.2x_5 + 4x_6 - 6\mu_6 = 176$$

$$0.2x_1 + 10.1x_2 + 13.2x_3 + 0.75x_4 + 0.15x_5 + 1.2x_6 - 5\mu_7 = 17.5$$

$$x_1 \leq 6.0$$

$$x_2 \leq 1.0$$

$$x_3 \leq 0.25$$

$$x_4 \leq 10.0$$

$$x_5 \leq 10.0$$

$$x_6 \leq 4.0$$

$$x_1, \dots, x_6 \geq 0$$

ステップ7: 次に示すようなA問題を解く. A問題の1番目(すなわち $t=1$ )は, 次のように定式化される.

$$\begin{aligned} \max \quad z_1 = & 0.041\mu_2 + 0.225\mu_3 + 0.099\mu_4 \\ & + 0.072\mu_5 + 0.375\mu_6 + 0.023\mu_7 \end{aligned}$$

s. t.

$$0.225x_1 + 2.2x_2 + 0.8x_3 + 0.1x_4 + 0.05x_5 + 0.26x_6 = 2.78$$

$$10x_1 + 20x_2 + 130x_3 + 82.5\mu_2 = 112.5$$

$$344x_1 + 1460x_2 + 1040x_3 + 75x_4 + 17.4x_5 + 240x_6 + 2868\mu_3 = 5668$$

$$24x_1 + 27x_2 + 15x_4 + 1.1x_5 + 52x_6 + 290\mu_4 = 540$$

$$720x_1 + 107x_2 + 7080x_3 + 134x_5 + 1000x_6 - 100\mu_5 = 4800$$

$$18x_1 + 151x_2 + 78x_3 + 2.5x_4 + 0.2x_5 + 4x_6 - 6\mu_6 = 176$$

$$0.2x_1 + 10.1x_2 + 13.2x_3 + 0.75x_4 + 0.15x_5 + 1.2x_6 - 5\mu_7 = 17.5$$

$$x_1 \leq 6.0$$

$$x_2 \leq 1.0$$

$$x_3 \leq 0.25$$

$$x_4 \leq 10.0$$

$$x_5 \leq 10.0$$

$$x_6 \leq 4.0$$

$$x_1, \dots, x_6 \geq 0$$

同様にして、 $t=2$ および $t=3$ の場合について定式化し、それぞれの問題を解く。

ステップ8：ステップ6, 7で得られた5つの非優越解について、TOPSISを適用する。

表 4.5 反復1でのTOPSIS法の計算結果

	$s_t^*$	$s_t^-$	$c_t^*$
t=0	0.03125	0.09735	0.7570
t=3	0.03125	0.09735	0.7570
t=4	0.03125	0.09735	0.7570
t=1	0.03347	0.09726	0.7439
t=2	0.03480	0.09735	0.7304

TOPSIS法の結果、 $t=0$ すなわちP問題の解を意思決定者に提示する。

ステップ9：各目的関数の達成度 $\mu_1 = 0.90$ ,  $\mu_2 = 0.55$ ,  $\mu_3 = 1.0$ ,  $\mu_4 = 1.0$ に対し、 $\mu_1$ と $\mu_2$ の値に満足しなかったものとして、4つの目的関数に対する重みを再設定する。

$$w_1 = 0.265, w_2 = 0.091, w_3 = 0.125, w_4 = 0.049$$

ステップ5に戻る。

この後、数回の反復の後、すべての目的関数の達成度に満足したものとして終了する。

### 4.4.3 評価

この問題をここで提案する手法で解いた結果、表4.6に示すような非優越解が得られた。

表4.6 提案する手法での反復1の結果

	目的関数値				制約条件の右辺定数		
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$g_1(x)$	$g_2(x)$	$g_3(x)$
目標値 $h_k$	2.78	30.0	2800.0	250.0	4800	176.0	17.5
重み $w_k$	0.165	0.041	0.225	0.099	0.072	0.375	0.023
t=0	3.13	67.46	2799.72	249.97	4799.66	175.98	16.99
メンバーシップ関数値	0.90	0.55	1.0	1.0	1.0	1.0	0.90
t=1	2.78	77.82	2770.31	249.83	4795.90	172.04	12.50
メンバーシップ関数値	1.0	0.42	0.94	1.0	1.0	0.36	0.0
t=2	3.44	30.0	2799.75	286.96	4799.96	175.98	15.44
メンバーシップ関数値	0.80	1.0	1.0	0.59	1.0	1.0	0.59
t=3	3.13	67.46	2799.72	249.97	4799.66	175.98	16.99
メンバーシップ関数値	0.90	0.55	1.0	1.0	1.0	1.0	0.90
t=4	3.13	67.46	2799.72	249.97	4799.66	175.98	16.99
メンバーシップ関数値	0.90	0.55	1.0	1.0	1.0	1.0	0.90

これらの非優越解についてTOPSIS法を用いると、表4.5より $t=0,3,4$ 、すなわちP問題またはA問題の3番と4番目の解が最良の妥協解として選ばれる。

この解は、 $f_2(x)$ 、すなわちコレステロールに関する目標値（だいたい30ユニット以下）に対し、67.46ユニット（メンバーシップ関数値0.55）とやや高い。また、コストに関する目標の達成度(0.90)もやや悪い。そこで、意思決定者はこの解に満足しなかったものとして、コストおよびコレステロールに対する重みをそれぞれ0.165から0.265、0.041から0.091に上げるとともに、カロリーおよび炭水化物に対する重みをそれぞれ0.225から0.125、0.099から0.049に下げ、反復を繰り返す。

この後、1回の反復により表4.7および表4.8に示すような結果が得られた。

表4.7 提案する手法での反復2の結果

	目的関数値				制約条件の右辺定数		
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$g_1(x)$	$g_2(x)$	$g_3(x)$
重み $w_k$	0.265	0.091	0.125	0.049	0.072	0.375	0.023
t=0	2.99	72.99	3220.14	296.39	4797.41	175.92	12.50
メンバーシップ関数値	0.93	0.48	0.85	0.93	1.0	1.0	0.0
t=1	2.97	56.67	2919.53	270.02	4704.23	170.03	12.51
メンバーシップ関数値	0.94	0.68	0.95	0.93	0.0	0.0	0.0
t=2	3.80	30.0	2861.22	289.93	4700.59	170.0	17.50
メンバーシップ関数値	0.73	1.0	0.98	0.86	0.0	0.0	1.0
t=3	3.15	63.80	2800.82	270.00	4700.87	170.08	17.51
メンバーシップ関数値	0.88	0.59	1.0	0.93	0.0	0.0	1.0
t=4	3.12	47.00	2918.69	270.00	4799.88	176.05	12.50
メンバーシップ関数値	0.89	0.79	0.96	0.93	1.0	1.0	0.0

表 4.8 提案する手法での反復3の結果

	目的関数値				制約条件の右辺定数		
	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$f_4(\mathbf{x})$	$g_1(\mathbf{x})$	$g_2(\mathbf{x})$	$g_3(\mathbf{x})$
重み $w_k$	0.270	0.111	0.125	0.024	0.072	0.375	0.023
t=0	2.69	81.83	2800.72	202.21	4799.29	176.07	12.50
メンバシップ関数値	1.0	0.37	1.0	1.0	1.0	1.0	0.0
t=1	infeasible				infeasible		
t=2	2.90	62.00	2800.44	227.19	4799.89	176.05	12.50
メンバシップ関数値	0.96	0.48	1.0	1.0	1.0	1.0	0.0
t=3	infeasible				infeasible		
t=4	2.90	62.00	2800.44	227.19	4799.89	176.05	12.50
メンバシップ関数値	0.96	0.48	1.0	1.0	1.0	1.0	0.0

TOPSIS法により、得られたすべての非優越解のうち最も理想値に近いのは、表4.8の反復3におけるP問題（すなわち、 $t=0$ ）の解である。本数値例では、意思決定者はこの解に満足したものとして最良な解を得る。すなわち、コレステロールの目標30.0に対して81.03（メンバシップ関数値0.37）と、目標を下回っているが、最大の重みをもつたんぱく質に関する目標のほか、他の目標をすべて満たしている。すなわち、コレステロールの目標をやや犠牲にすることにより、他の目標をすべて達成している。ただし、制約条件 $g_3(\mathbf{x})$ のメンバシップ関数値が0.0と低いが、これは右辺定数 $b_3$ の下限値12.5となっているためである。これは、右辺定数に幅を持たせない通常の制約条件

$$g_3(\mathbf{x}) \geq 17.5$$

では、実行可能解となり得なかった解であり、提案する手法の有効性を示している。

さらに、提案する手法では、次善の解であるA問題の2番目の解（すなわち、 $t=2$ ）を提供できる柔軟性を持ち、意思決定者の満足する解を容易に見つけることが可能となっている。

表 4.9 反復2でのTOPSIS法の計算結果

	$s_t^*$	$s_t^-$	$c_t^*$
t=3	0.03252	0.09431	0.74542
t=1	0.03250	0.09405	0.74321
t=4	0.03252	0.09300	0.74097
t=2	0.03249	0.09405	0.74321
t=0	0.04075	0.08683	0.68058

表 4.10 反復3でのTOPSIS法の計算結果

	$s_t^*$	$s_t^-$	$c_t^*$
t=0	0.03134	0.11935	0.79202
t=2	0.03227	0.11581	0.78205
t=4	0.03227	0.11581	0.78205

## 4.5 むすび

本章では、ISGPをさらに発展させ、たファジィ目標計画問題の一解法として、意思決定者の目標値に対する満足度のあいまいさをメンバシップ関数によって表すほか、右辺定数にゆとりをもたせるファジィ制約を伴う会話型の目標計画法に基づく実用的な手法を提案した。提案解法は、次に示すような特徴を持つことを示した。

- ISGPに、ファジィ目標とファジィ制約を導入することにより、意思決定者の主観的判断に基づくあいまいさや、システムのおかれた環境の変化に柔軟に対応できる意思決定が可能となった。
- さらに、各目的関数に対する意思決定者の重要度を表すための重みづけに、多評価意思決定のための支援手法として評価されているAHPを用いたことにより、意思決定者の選好構造をより反映させることができる。

## 第5章

# ファジィ多目的信頼性の最適設計

### 5.1 まえがき

複数のユニットから構成されるシステムにおいて、その中の1個のユニットの故障がシステム全体の機能に致命的な影響を与えることがある。このような場合、システムの信頼性を高める手段の一つとして、冗長系の採用がある<sup>[63, 108]</sup>。このようなシステムのユニット冗長最適配分などの最適設計問題は、一般に非線形整数計画問題として定式化される。この種の非線形整数計画問題の解法として、0-1線形計画問題に変換した後、ラグランジュ乗数を用いた動的計画法のほか、平面切除法、陰的列挙法やヒューリスティック・アルゴリズムを用いた手法等を適用することによって解決されることが示されている<sup>[17, 119]</sup>。最近、その際に必然的に生じるGUBと呼ばれる特殊な構造を生かした効率的な解法が新しく提案された<sup>[13, 16, 17, 37, 38, 39, 40]</sup>。それらの手法は、コンピュータメモリの使用量が少なく、かつ計算時間が早いため、パーソナルコンピュータ上での大規模な0-1線形計画問題を効率的に扱えるなど、多くの利点があることが明らかにされており、NASA(米・航空宇宙局)で開発された生命維持支援システム(Life-Support System)への応用等が提案されている<sup>[35]</sup>。

他方、システム信頼性やシステム・アベイラビリティの最適配分に関する問題は、これまで単一目的の計画問題として定式化され、多くの優れた手法が報告されてきた。しかし、単一の目的関数である信頼度関数あるいはコスト関数のみを最適化する計画モデルでは解決できない問題点が存在し、互いに利害が相反する複数の達成すべき目的関数を伴う多目的意思決定問題の各手法を適用する必要性が高まってきている。前述の新しい解法は、単一目的または2目的の0-1計画問題を対象としたもので、より一般的な多目的の0-1計画問題に対しては有効な方法は示されていない。

このような多目的意思決定問題に対する代表的な解法の一つに目標計画法があ

る。最近、この目標計画法に対し、前述のような信頼性最適設計問題を解く際に生じるGUB構造を生かした効率的な解法も提案されている。従来の目標計画法では、意思決定者であるシステム設計者によって目標値とその優先順位および重みが確定値として与えられていたが、現実の意思決定においては、それらの値がシステム設計者の主観的判断に基づくあいまいさをもつことについては、考慮されていなかった。また、システムの使用状況や経済環境などの変化によって、システム制約条件の右辺定数が変化することは、現実によくある問題である。

本章では、このような観点に立って、次のような特徴を持つ新しいファジィ多目的0-1計画問題の解法を提案する。

1. システム信頼性最適設計問題を解く際に生じる特有な構造の一つであるGUBを考慮した効率的な解法を、多目的計画問題に拡張した。
2. 各目的関数値に対する目標値の設定にファジィ理論を導入することによって、システム設計者の主観的判断に起因するあいまいさを考慮することができる。
3. システムの使用環境などにより変化するシステム制約の右辺定数の上限と下限を設定することによってゆとりをもたせることができる。
4. 多評価意思決定のための支援手法として評価されているAHPに基づいた各目的関数や制約条件の重要度に対応する重みを、状況に応じて設定し直すことが可能である。

さらに、ここで提案する手法の有効性を示すために、システム資源が二種類でかつ線形な制約条件の下で、線形なコストおよび作業人員関数をそれぞれ最小に、非線形な信頼度関数を最大にする最適ユニット選択と冗長配分の設計問題を取り上げる。

他方、複数のユニットから構成されるシステムの信頼性を高める方法の一つとして、冗長系の採用が有効であることが知られている。この冗長系の一つの形態に、待機冗長系(stand-by redundancy)があり、これは並列系にすることが困難で、かつ切換えの即時性が要求される場合に有効である。このようなシステムのユニット選択や冗長配分などの最適設計問題に対し、これまでに数多くの解法が提案されている[6, 9, 74, 75, 120]。

一般に、この種の問題は非線形整数計画(nonlinear integer programming :NIP)問題として定式化され、0-1線形計画(linear programming : LP)問題に変換後、ラグランジュ乗数を用いた動的計画法、平面切除法、陰的列挙法やヒューリスティック・アルゴリズムを用いた手法等を適用することによって解決される。最近、NIP問題を0-1LP問題に変換する段階で必然的に生じるGUBと呼ばれる特殊な構造を生かした効率的

な解法を発展させた，ファジィ多目的0-1LP問題の解法が提案されている<sup>[90]</sup>．この解法は，従来の解法に対し，多目的0-1LP問題が扱えるように拡張されたほか，各目的関数に対して意思決定者が設定する目標値のあいまいさを評価することが可能となっている．

しかし，この解法では，ユニットの故障が発生した場合の故障検出切換え (fault detecting and switching : FDS) 装置そのものの信頼度が1であることを前提としているため，現実の信頼性最適化問題には十分対処できなかった．

このような観点から，本章では不完全なFDSをもつ待機冗長システムの信頼性を最適化するユニット配分問題を，ファジィ多目的0-1線形計画法によって解くための計算法を提案する．提案する手法では，各目的関数に対してファジィ目標を設定できるため，目標値に対する意思決定者のあいまいさを考慮に入れることが可能であるとともに，各目的関数の目標値および制約条件の達成度に対する意思決定者の満足度を最大にするような決定変数を，各GUB制約の中から効率的に選ぶことができるように改良されている．また，目標値に対する意思決定者のあいまいさやシステム資源の使用状況，さらに経済環境の変化にも対応できるため，より現実的なシステム信頼性の最適設計が可能となる．

さらに，提案する手法の有効性を示すために，三種類の非線形なシステム資源の制約の下で，四種類の異なる故障のタイプをもつサブシステムから構成されたシステムの信頼度を最大に，またコスト関数を最小にする最適な冗長ユニット配分の設計問題を取り上げる．

## 5.2 ファジィ多目的信頼性設計I

### 5.2.1 ファジィ多目的信頼性設計問題

$m$  個のシステム制約と  $p$  個のGUB制約条件の下で， $q$  個の目的関数を最小化(ファジィ最小化)するファジィ多目的信頼性設計問題は，次のように表される．

$$\begin{aligned} \min \quad & f_1(\mathbf{x}) \lesssim h_1 \\ \min \quad & f_2(\mathbf{x}) \lesssim h_2 \\ & \dots \end{aligned} \tag{5.1}$$

$$\begin{aligned} \min \quad & f_q(\mathbf{x}) \lesssim h_q \\ \text{s. t.} \quad & g_r(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} a_{rij} x_{ij} \lesssim b_r, \quad r = 1, 2, \dots, T \end{aligned} \tag{5.2}$$

$$g_{T+i}(\mathbf{x}) \equiv \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, n \tag{5.3}$$

$$x_{ij} = 0 \text{ or } 1, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n_i \quad (5.4)$$

ただし,  $f_k(\mathbf{x})$  は,

$$f_k(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} c_{kij} x_{ij}, \quad k = 1, 2, \dots, q \quad (5.5)$$

で表される  $k$  番目の目的関数,  $h_k$  はシステム設計者が設定する  $k$  番目の目的関数の目標値で, 最悪値 (NIS)  $z_k^-$ , ( $k = 1, 2, \dots, q$ ) と理想値 (PIS)  $z_k^*$ , ( $k = 1, 2, \dots, q$ ) の間の値をとる. 記号  $\lesssim$  はあいまいさ (大体小さいか等しい) を表すファジィ不等号である. 式 (5.2) は, システム設計者によって右辺定数  $b_r$ , ( $r = 1, 2, \dots, T$ ) に上限と下限が与えられるファジィシステム制約である.  $a_{rij}$  は GUB 制約  $i$  における  $j$  番目の要素に対応する  $r$  番目のシステム制約の係数,  $b_r$  は  $r$  番目のシステム制約の右辺定数である. 式 (5.3) は GUB 制約と呼ばれ,  $n_i$  は GUB 制約  $i$  に含まれる決定変数の数である. また,  $x_{ij}$ , ( $j = 1, 2, \dots, n_i$ ) は GUB 制約  $i$  における決定変数であり, 0 または 1 のどちらかの値をとる.

ここで, 目的関数  $f_k(\mathbf{x})$  に対するメンバシップ関数  $\mu_k$ , ( $k = 1, 2, \dots, q$ ) およびシステム制約  $g_r(\mathbf{x})$  に対する線形型のメンバシップ関数  $\mu_{q+r}$ , ( $r = 1, 2, \dots, T$ ) をそれぞれ次のように定義する.

$$\mu_k = \begin{cases} 1 & ; f_k(\mathbf{x}) < h_k \\ \frac{z_k^- - f_k(\mathbf{x})}{z_k^- - h_k} & ; h_k \leq f_k(\mathbf{x}) \leq z_k^- \\ 0 & ; z_k^- < f_k(\mathbf{x}) \end{cases} \quad k = 1, 2, \dots, q \quad (5.6)$$

$$\mu_{q+r} = \begin{cases} 1 & ; g_r(\mathbf{x}) < b_r - \delta_r/2 \\ \frac{b_r + \delta_r/2 - g_r(\mathbf{x})}{\delta_r} & ; b_r - \delta_r/2 \leq g_r(\mathbf{x}) \leq b_r + \delta_r/2 \\ 0 & ; b_r + \delta_r/2 < g_r(\mathbf{x}) \end{cases} \quad r = 1, 2, \dots, T \quad (5.7)$$

ここで,  $\delta_r$  はシステム設計者が設定するシステム制約条件  $r$  の右辺定数  $b_r$  の幅, すなわち右辺定数の下限と上限の距離である.

$g_r(\mathbf{x}) \gtrsim b_r$  の場合, 式 (5.7) は次式と置き換えられる.

$$\mu_{q+r} = \begin{cases} 1 & ; b_r + \delta_r/2 \leq g_r(\mathbf{x}) \\ \frac{g_r(\mathbf{x}) - b_r + \delta_r/2}{\delta_r} & ; b_r - \delta_r/2 \leq g_r(\mathbf{x}) \leq b_r + \delta_r/2 \\ 0 & ; g_r(\mathbf{x}) < b_r - \delta_r/2 \end{cases} \quad r = 1, 2, \dots, T \quad (5.7)'$$

この線形型メンバシップ関数は、各目的関数のPISとNISとの距離に対する目標値とNISの距離の比によって目標に対する達成度を評価するものである。また、システム設計者は、提案するメンバシップ関数の採用により複数の目的関数に対する目標を設定することに専念できるため、得られた非優越解に対する判断が容易になる。このメンバシップ関数を、上に示した $q$ 個の目的関数と $T$ 個のシステム制約、 $n$ 個のGUB制約をもつファジィ多目的0-1計画問題に導入すると、次のような単一目的のGUB型線形計画問題として定式化することができる。

$$\max z = \sum_{k=1}^{q+m} w_k \mu_k \quad (5.8)$$

$$\text{s. t. } \sum_{i=1}^n \sum_{j=1}^{n_i} c_{kij} x_{ij} + (z_k^- - h_k) \mu_k = z_k^-, \quad k = 1, 2, \dots, q \quad (5.9)$$

$$\sum_{i=1}^n \sum_{j=1}^{n_i} a_{rij} x_{ij} + \delta_r \mu_{q+r} = b_r, \quad r = 1, 2, \dots, T \quad (5.10)$$

$$g_{T+i}(\mathbf{x}) \equiv \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (5.11)$$

$$x_{ij} = 0 \text{ or } 1, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n_i \quad (5.12)$$

$$0 \leq \mu_k \leq 1, \quad k = 1, 2, \dots, q+T \quad (5.13)$$

ここで、 $w_k$ はAHPによって得られる各目的関数 $f_k(\mathbf{x})$ およびシステム制約 $g_r(\mathbf{x})$ の重要度に対応する重みである。

## 5.2.2 計算法

$m$ 個のシステム制約と $n$ 個のGUB制約条件の下で、 $q$ 個の目的関数を最適化するファジィ多目的0-1線形計画問題(最小化問題)を解く計算手順を次に示す。

ステップ1: 各システム制約に対する右辺定数の幅 $\delta_r$ , ( $r = 1, 2, \dots, T$ )を設定する。

ステップ2: システム設計者の基本的な設計方針を反映させるために、AHPによって各目的関数およびシステム制約条件の重要度 $w_k$ , ( $k = 1, 2, \dots, q+T$ )をAHPによって計算する。

ステップ3: 次に示す $p$ 個のGUB制約をもつ単一目的の0-1線形計画(0-1LP(GUB))問題

$$\min f_k(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} c_{kij} x_{ij}, \quad k = 1, 2, \dots, q \quad (5.14)$$

$$\text{s. t. } g_r(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} a_{rij} x_{ij} \leq b_r + \delta_r/2, \quad r = 1, 2, \dots, T \quad (5.15)$$

$$g_{T+i}(\mathbf{x}) \equiv \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (5.16)$$

$$x_{ij} = 0 \text{ or } 1, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n_i \quad (5.17)$$

を解き、それぞれ PIS を求める。さらに、上記の目的関数の型 (min/max) を反転させた(もとの目的関数の型が min なら max に、max なら min に)  $q$  個の単一目的の GUB 型 0-1 線形計画問題を解き、それぞれ NIS を求める。この 0-1LP(GUB) 問題でのシステム制約は、最も緩い条件で理想解を得るために、 $g_r(\mathbf{x}) \leq b_r + \delta_r/2$ , ( $r = 1, 2, \dots, T$ ) とする。

ステップ 4: 各目的関数に対する目標値  $h_k$  を、次に示す条件

$$z_k^* < h_k \leq z_k^-, \quad k = 1, 2, \dots, q \quad (5.18)$$

を満足する範囲でシステム設計者が設定する。

ステップ 5: 各 GUB 制約において、低いランクの実行可能解を探すために、式 (5.20) に基づいて各 GUB 制約の集合の変数を昇順にランク付けを行い、そのインデックスを求める。

$$\gamma_{ij} = \sum_{k=1}^q c_{rij}/z_k^- + \sum_{r=1}^m a_{rij}/b_r \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n_i \quad (5.19)$$

$$j_i = [j_{i1} \quad j_{i2} \quad \dots \quad j_{in}] \\ = \underset{j}{\text{indsort}}\{\gamma_{ij} \mid j = 1, 2, \dots, n_i\}, \quad i = 1, 2, \dots, n \quad (5.20)$$

ここで、indsort の ind は index の省略形であり、ソーティング後のインデックスを採用することを意味する。

ステップ 6: すべての決定変数の値を 0 にする。各 GUB 制約の集合において、最高ランク(ランク 1)の変数の値を 1 にする。

$$x_{ij_{i1}} = 1, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n_i \\ k = [k_1 \quad k_2 \quad \dots \quad k_n] = [1 \quad 1 \quad \dots \quad 1]$$

これらの変数の集合を初期解  $x^1$  とする。

ステップ7: 次式から各目的関数に対する理想値と達成値の残差  $s_k, (k=1, 2, \dots, q)$  およびシステム制約における達成値と右辺定数の残差  $s_{q+r}, (r=1, 2, \dots, T)$  をそれぞれ求める.

$$s_k = \begin{cases} z_k^* - f_k(\mathbf{x}^1) & ; f_k(\mathbf{x}^1) < z_k^* \\ 0 & ; z_k^* \leq f_k(\mathbf{x}^1) \leq z_k^- \\ z_k^- - f_k(\mathbf{x}^1) & ; z_k^- < f_k(\mathbf{x}^1) \end{cases} \quad k=1, 2, \dots, q \quad (5.21)$$

$$s_{q+r} = \begin{cases} b_r - \delta_r/2 - g_r(\mathbf{x}^1) & ; g_r(\mathbf{x}^1) < b_r - \delta_r/2 \\ 0 & ; b_r \delta_r / 2 \leq g_r(\mathbf{x}^1) \leq b_r + \delta_r/2 \\ b_r + \delta_r/2 - g_r(\mathbf{x}^1) & ; b_r + \delta_r/2 < g_r(\mathbf{x}^1) \end{cases} \quad r=1, 2, \dots, T \quad (5.22)$$

ステップ8: すべての  $k, (k=1, 2, \dots, q+m)$  に対し,  $s_k=0$  ならば, ステップ9へいく. そうでなければ各GUB制約の集合に対し, 次式を満たすような低いランクの実行可能変数を探す.

$$i^* = \operatorname{indmax}_i \left\{ \sum_{k=1}^q (c_{kij_{i^*+1}} - c_{kij_{i^*}}) / z_k^- + \sum_{r=1}^T (a_{rij_{i^*+1}} - a_{rij_{i^*}}) / b_r \mid s_k \neq 0, i=1, 2, \dots, n \right\} \quad (5.23)$$

ここで,  $\operatorname{indmax}$  の  $\operatorname{ind}$  は  $\operatorname{index}$  の省略形であり, 残差の最大値をとるときのインデックスを採用することを意味する.

残差の最大値が正の場合は低いランクの変数, 負の場合には高いランクの変数の値を1にする. そのGUB制約で, いままで1を持っていた変数を0とし, 最も大きな値をもつ目的関数またはシステム制約の残差を計算する. 残差が改善されていればステップ7へ戻る. そうでなければ, 式(5.23)によってさらに小さい値をもつGUB制約の変数に対して残差を改善する変数を探す. 存在すれば, その変数の値を1とする. 存在しなければ低いランクの変数を探す.

$$x_{ij_{i^*+1}} = 1, \quad x_{ij_{i^*}} = 0, \quad k_i = k_i + 1$$

すべてのGUB制約において, そのような変数が存在しなければ, 非実行可能問題として終了する.

同値の場合は, 各GUB制約において, さらに低いランクの変数を選び, その中で最も大きな  $r_{ij}$  の値をもつ変数を探し, ステップ7へ戻る. 最大残差が正の場合  $j_i = j_{i+1}$ , また負の場合には  $j_i = j_{i-1}$  とおく.

ステップ9: 各GUB制約において,  $\sum_{k=1}^q w_k e_k$  の値を最小にするような実行可能変数を探す. ここで,  $e_k$ , ( $k=1, 2, \dots, q$ ) は各目的関数に対する目標値と達成値との差を表す.

$$e_k = |f_k(\mathbf{x}^1) - h_k| \quad (5.24)$$

存在する場合には, その変数の値を1にし, 同じGUB制約で1をもっていた変数の値を0にする.

ステップ10: 各目的関数値の達成度(目標の満足度)  $\mu_k$ , ( $k=1, 2, \dots, q$ ) がシステム設計者の希望する値より大きければ, 得られた解を最良な妥協解として終了する. そうでなければ, 各目的関数の重要度  $w_k$  ( $k=1, 2, \dots, q$ ) を再設定してステップ7に戻る.

ただし, 最大化問題の場合は, ステップ3において目的関数の型をminからmaxに換えるほか, ステップ4の条件を次のように変更する.

$$z_k^- < h_k \leq z_k^*, \quad k=1, 2, \dots, q \quad (5.18)'$$

さらに, ステップ5におけるランク付けを降順とする. また,  $g_r(\mathbf{x}^1) \lesssim b_r$  の場合は, ステップ3において制約条件(5.15)が, 次式と置き換えられる.

$$g_r(\mathbf{x}^1) \geq b_r - \delta_r/2, \quad r=1, 2, \dots, T \quad (5.15)'$$

### 5.2.3 最適設計問題への応用

システム信頼性最適設計問題の例として, 表5.2に示すようなシステム資源が2種類でかつ線形な制約条件の下で, 線形なコストおよび作業人員関数をそれぞれ最小に, 非線形の信頼度関数を最大にする最適ユニット選択と冗長配分の設計問題を取り上げる.

表 5.2 システム信頼性の最適ユニットの選択の数値例

サブシステム	コスト (千ドル)	信頼度	作業人数 (人)	期日 (日)	重量 (100Kg)
1	45	0.95	7	5	4
	50	0.97	35	10	5
	60	0.98	20	15	7
2	40	0.96	50	2	2
	80	0.99	60	5	4
	60	0.98	12	17	6
	20	0.90	40	25	3
3	35	0.97	15	30	4
	50	0.98	5	50	4
制限量				$60 \leq b_1 \leq 70$	$12.5 \leq b_2 \leq 17.5$

このような最適化問題は、次のような三つのファジィ目的関数、二つのファジィシステム制約および三つの GUB 制約をもつファジィ多目的 0-1 計画問題として定式化できる。

$$\min f_1(\mathbf{x}) = 45x_{11} + 50x_{12} + 60x_{13} + 40x_{21} + 80x_{22} + 60x_{23} + 20x_{24} + 35x_{31} + 50x_{32} \lesssim h_1$$

$$\min f_2(\mathbf{x}) = 5.129x_{11} + 3.046x_{12} + 2.020x_{13} + 4.082x_{21} + 1.005x_{22} + 2.020x_{23} + 10.536x_{24} + 3.046x_{31} + 2.020x_{32} \lesssim h_2$$

$$\min f_3(\mathbf{x}) = 7x_{11} + 35x_{12} + 20x_{13} + 50x_{21} + 60x_{22} + 12x_{23} + 40x_{24} + 15x_{31} + 5x_{32} \lesssim h_3$$

s. t.

$$g_1(\mathbf{x}) = 5x_{11} + 10x_{12} + 15x_{13} + 2x_{21} + 5x_{22} + 17x_{23} + 25x_{24} + 30x_{31} + 50x_{32} \lesssim 65$$

$$g_2(\mathbf{x}) = 4x_{11} + 5x_{12} + 7x_{13} + 2x_{21} + 4x_{22} + 6x_{23} + 3x_{24} + 4x_{31} + 4x_{32} \lesssim 15$$

$$g_3(\mathbf{x}) \equiv \sum_{j=1}^3 x_{1j} = 1$$

$$g_4(\mathbf{x}) \equiv \sum_{j=1}^4 x_{2j} = 1$$

$$g_5(\mathbf{x}) \equiv \sum_{j=1}^2 x_{3j} = 1$$

$$x_{tj} = 0 \text{ or } 1, \quad t = 1, 2, 3, \quad j = 1, 2, \dots, n_t$$

$$n_1 = 3, n_2 = 4, n_3 = 2$$

ステップ 1 : 二つのシステム制約  $g_1(\mathbf{x})$ ,  $g_2(\mathbf{x})$  に対する右辺定数の幅をそれぞれ次のように設定する。

$$\delta_1 = 10.0, \quad \delta_2 = 5.0$$

ステップ2: システム設計者の基本的な設計方針を反映させるために, AHPによって各目的関数およびシステム制約に重要度によるランク付けを行う. 次のように一対比較行列が得られたとする.

$$D = \begin{bmatrix} 1 & 1/8 & 2 & 5 & 3 \\ 8 & 1 & 9 & 7 & 5 \\ 1/2 & 1/9 & 1 & 3 & 1 \\ 1/5 & 1/7 & 1/3 & 1 & 3 \\ 1/3 & 1/5 & 1 & 1/3 & 1 \end{bmatrix}$$

これより, 各目的関数及びシステム制約の重要度に対応する重み  $w_k$ , ( $k = 1, 2, \dots, 5$ ) が次のように与えられる.

$$\begin{aligned} w_1 &= 0.168, \quad w_2 = 0.618, \quad w_3 = 0.090, \\ w_4 &= 0.063, \quad w_5 = 0.060 \end{aligned}$$

ステップ3: 各目的関数について, それぞれ単一目的の0-1線形計画問題として解き, 理想値および最悪値  $z_k^*, z_k^-, k = 1, 2, 3$  を求める.

表 5.3 理想値および最悪値

k	$z_k^-$	$z_k^*$
1	190.000	100.000
2	18.711	5.045
3	110.000	34.000

ステップ4: 最初の目標値として, 各目的関数に対するPISを与える.

$$h_1 = 100.0, \quad h_2 = 5.045, \quad h_3 = 34.0$$

ステップ5: 第一目的関数に対するメンバシップ関数は, 次のように表せる.

$$\mu_1 = \begin{cases} 1 & ; 0 < f_1(\mathbf{x}) < 100 \\ \frac{190.0 - f_1(\mathbf{x})}{190.0 - 100.0} & ; 100 \leq f_1(\mathbf{x}) \leq 190 \\ 0 & ; 190 < f_1(\mathbf{x}) \end{cases}$$

同様に, 第二, 第三目的関数に対するメンバシップ関数  $\mu_2, \mu_3$  を式(5.6)から得る. また, 第一システム制約に対するメンバシップ関数  $\mu_4$  は, 次のとおりである.

$$\mu_1 = \begin{cases} 1 & ; g_1(\mathbf{x}) < 60 \\ \frac{70 - g_1(\mathbf{x})}{10} & ; 60 \leq g_1(\mathbf{x}) \leq 70 \\ 0 & ; 70 < g_1(\mathbf{x}) \end{cases}$$

同様に、第二システム制約に対するメンバシップ関数  $\mu_5$  を、式(5.7)から得る。以上から、三つの目的関数、二つのシステム制約条件および三つのGUB制約条件をもつファジィ多目的0-1計画問題は、次のようなGUB制約を伴う線形計画問題として定式化される。

$$\begin{aligned} \max \quad & z = 0.168\mu_1 + 0.618\mu_2 + 0.090\mu_3 + 0.063\mu_4 + 0.060\mu_5 \\ \text{s. t.} \quad & 45x_{11} + 50x_{12} + 60x_{13} + 40x_{21} + 80x_{22} + 60x_{23} + 20x_{24} + 35x_{31} + 50x_{32} + 90\mu_1 = 190 \\ & 5.129x_{11} + 3.046x_{12} + 2.020x_{13} + 4.082x_{21} + 1.005x_{22} + 2.020x_{23} + 10.536x_{24} \\ & \quad + 3.046x_{31} + 2.020x_{32} + 13.666\mu_2 = 18.700 \\ & 7x_{11} + 35x_{12} + 20x_{13} + 50x_{21} + 60x_{22} + 12x_{23} + 40x_{24} + 15x_{31} + 5x_{32} + 76\mu_3 = 110 \\ & 5x_{11} + 10x_{12} + 15x_{13} + 2x_{21} + 5x_{22} + 17x_{23} + 25x_{24} + 30x_{31} + 50x_{32} + 10\mu_4 = 70 \\ & 4x_{11} + 5x_{12} + 7x_{13} + 2x_{21} + 4x_{22} + 6x_{23} + 3x_{24} + 4x_{31} + 4x_{32} + 5\mu_5 = 17.5 \\ & \sum_{j=1}^3 x_{1j} = 1 \\ & \sum_{j=1}^4 x_{2j} = 1 \\ & \sum_{j=1}^2 x_{3j} = 1 \\ & x_{tj} = 0 \text{ or } 1, \text{ for all } t, j \end{aligned}$$

表 5.4 ランク付けされた変数

ランク t	1	2	3	4
1	$x_{11}$	$x_{12}$	$x_{13}$	
2		$x_{32}$	$x_{23}$	$x_{22}$ $x_{24}$
3		$x_{31}$	$x_{32}$	

式(5.19)に基づいて、各GUB制約の集合の変数を昇順にランク付けを行う。

$$r_{11} = 0.875, r_{12} = 1.173, r_{13} = 1.220,$$

$$r_{21} = 1.026, r_{22} = 1.320, r_{23} = 1.119,$$

$$r_{24} = 1.561, r_{31} = 1.141, r_{32} = 1.359$$

ステップ6: すべての決定変数を  $x_{tj} = 0$  とする。各GUB制約において、最高ランクの変数の値を1にする。初期解  $x^1$  は、次のとおりである。

$$x^1 = \{x_{11}, x_{21}, x_{31}\}$$

$$k = [k_1 \ k_2 \ k_3]^T = [1 \ 1 \ 1]^T$$

ステップ7: 初期解に対する各制約の値は次のとおりである.

$$f_1(\mathbf{x}^1) = 120 \in [100, 190]$$

$$f_2(\mathbf{x}^1) = 12.257 \in [5.045, 18.711]$$

$$f_3(\mathbf{x}^1) = 72 \in [34, 110]$$

$$g_1(\mathbf{x}^1) = 37 \notin [60, 70]$$

$$g_2(\mathbf{x}^1) = 10 \notin [12.5, 17.5]$$

得られた各目的関数値  $f_k(\mathbf{x}^1)$  が  $z_k^*$  と  $z_k^-$  の間の値をとる場合,  $f_k(\mathbf{x}^1) \in [z_k^*, z_k^-]$ , 同様にシステム制約の値  $g_r(\mathbf{x}^1)$  についても  $g_r(\mathbf{x}^1) \in [b_r - \delta_r/2, b_r + \delta_r/2]$  と表す.

これより, 残差  $s$  は, 式(5.21),(5.22)により次のようになる.

$$s = [0 \ 0 \ 0 \ 23 \ 2.5]^T$$

ステップ8:

$$\begin{aligned} i^* &= \underset{i}{\operatorname{indmax}} \{(10-5)/70 + (5-4)/17.5, (17-2)/70 + (6-2)/17.5, (50-30)/70 + (4-4)/17.5\} \\ &= \underset{i}{\operatorname{indmax}} \{0.129, 0.443, 0.286\} = 2 \end{aligned}$$

最も大きい残差  $s_4 > 0$  に対する改善度は

$$(17-2)/70 > 0$$

であるから,  $x_{23}$  が選ばれる.  $x_{23} = 1, x_{21} = 0$

$$\mathbf{x}^1 = \{x_{11}, x_{23}, x_{31}\}$$

ステップ7へ戻る.

[ステップ7: 各制約の値は次のとおりである.]

$$f_1(\mathbf{x}^1) = 140 \in [100, 190]$$

$$f_2(\mathbf{x}^1) = 10.195 \in [5.045, 18.711]$$

$$f_3(\mathbf{x}^1) = 34 \in [34, 110]$$

$$g_1(\mathbf{x}^1) = 52 \notin [60, 70]$$

$$g_2(\mathbf{x}^1) = 14 \in [12.5, 17.5]$$

これより, 残差  $s$  は, 式(5.21),(5.22)により次のようになる.

$$s = [0 \ 0 \ 0 \ 8 \ 0]^T$$

ステップ 8 :

$$i^* = \underset{i}{\operatorname{indmax}}\{(10 - 5)/70, (5 - 17)/70, (50 - 30)/70\} \\ = 3$$

最も大きい残差  $s_4 > 0$  に対する改善度は

$$(50 - 30)/70 > 0$$

であるから,  $x_{32}$  が選ばれる.  $x_{32} = 1, x_{31} = 0$

$$\mathbf{x}^1 = \{x_{11}, x_{23}, x_{32}\}$$

ステップ 7 へ戻る.

ステップ 7 : 各制約の値は次のとおりである.

$$f_1(\mathbf{x}^1) = 155 \in [100, 190]$$

$$f_2(\mathbf{x}^1) = 9.169 \in [5.045, 18.711]$$

$$f_3(\mathbf{x}^1) = 24 \notin [34, 110]$$

$$g_1(\mathbf{x}^1) = 72 \notin [60, 70]$$

$$g_2(\mathbf{x}^1) = 14 \in [12.5, 17.5]$$

これより, 残差  $\mathbf{s}$  は, 次のとおりである.

$$\mathbf{s} = [0 \ 0 \ 10 \ -2 \ 0]^T$$

ステップ 8 :

$$i^* = \underset{i}{\operatorname{indmax}}\{0.183, 0.608, -\} = 2$$

最も大きい残差  $s_3 > 0$  に対する改善度は

$$(60 - 12)/110 > 0$$

であるから,  $x_{22}$  が選ばれる.  $x_{22} = 1, x_{23} = 0$

$$\mathbf{x}^1 = \{x_{11}, x_{22}, x_{32}\}$$

ステップ 7 へ戻る.

ステップ7: 各制約の値は次のとおりである.

$$f_1(x^1) = 175 \in [100, 190]$$

$$f_2(x^1) = 8.154 \in [5.045, 18.711]$$

$$f_3(x^1) = 72 \in [34, 110]$$

$$g_1(x^1) = 60 \in [60, 70]$$

$$g_2(x^1) = 12 \notin [12.5, 17.5]$$

これより, 残差  $s$  は, 次のようになる.

$$s = [0 \ 0 \ 0 \ 0 \ 0.5]^T$$

ステップ8:

$$i^* = \underset{i}{\operatorname{indmax}}\{0.06, -0.06, -\} = 1$$

最も大きい残差  $s_3 > 0$  に対する改善度は

$$(5 - 4)/17.5 > 0$$

であるから,  $x_{12}$  が選ばれる.  $x_{12} = 1, x_{11} = 0$

$$x^1 = \{x_{12}, x_{22}, x_{32}\}$$

ステップ7へ戻る.

ステップ7: 各制約の値は次のとおりである.

$$f_1(x^1) = 180 \in [100, 190]$$

$$f_2(x^1) = 6.071 \in [5.045, 18.711]$$

$$f_3(x^1) = 100 \in [34, 110]$$

$$g_1(x^1) = 65 \in [60, 70]$$

$$g_2(x^1) = 13 \in [12.5, 17.5]$$

これより, 残差  $s$  は, 次のようになる.

$$s = [0 \ 0 \ 0 \ 0 \ 0]^T$$

すべての  $k, k=1, 2, \dots, 5$  に対し,  $s_k = 0$  であるから, ステップ9へ行く.

ステップ9: 得られた解は  $x^1 = \{x_{12}, x_{22}, x_{32}\}$  である. 各目的関数に対する目標値と達成値との差  $e$  はそれぞれ次のとおりである.

$$e = [80 \ 1.026 \ 66]$$

各目標値との差に対する重み付き和は、次のようになる。

$$\sum_{k=1}^3 w_k e_k = 20.004$$

これより小さな値をとる実行可能変数は存在しないから、ステップ10へ行く。

ステップ10: 得られた最良解は、 $x^1 = \{x_{12}, x_{22}, x_{32}\}$ である。

#### 5.2.4 評価

目的関数  $f_1(x)$  の目標値 100.0 に対して 180.0,  $f_2(x)$  の目標値 5.045 に対して 6.071,  $f_3(x)$  の目標値 34.0 に対して 100.0 である。またシステム制約条件式の値  $g_1(x)$  は 70 ( $60 \leq b_1 \leq 70$ ),  $g_2(x)$  は 13 ( $12.5 \leq b_2 \leq 17.5$ ) となっている。このときのメンバシップ関数値はそれぞれ

$$\mu_1 = 0.111, \quad \mu_2 = 0.925, \quad \mu_3 = 0.132, \quad \mu_4 = 0.8, \quad \mu_5 = 0.9$$

である。重要度に対応した重みを考慮すると、二番目に大きなウェイトを持つ  $\mu_1$  のメンバシップ関数値が 0.111 と低いほか、三番目の目的関数のメンバシップ関数値が 0.132 となっており、望ましくない。本数値例ではシステム設計者がこの結果に対して各目標の目標値を十分に満足しているとはいえないと判断したものとして

比較的ウェイトの低い作業人数に関する重みを 0.090 から 0.070 に落とすほか、コストに関する重みを 0.168 から 0.188 に少し上げ、以下のステップを繰り返す。

この後、1回の反復でそれぞれ表 5.5 の反復 2, 3 に示すような結果が得られた。

表 5.5 提案する手法での各反復時の結果

反復 r	目的関数			制約条件の右辺定数		値 1 をもつ決定変数 (他の変数は 0)
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$g_1(x)$	$g_2(x)$	
目的関数値(反復 1)	180.000	6.071	100.000	65.0	13.0	$x_{12}, x_{22}, x_{32}$
重み	0.168	0.618	0.090	$60 \leq b_1 \leq 70$	$12.5 \leq b_2 \leq 17.5$	
メンバシップ関数値	0.111	0.925	0.132	0.50	0.90	
目的関数値(反復 2)	155.000	7.086	47.000	62.0	17.0	$x_{13}, x_{23}, x_{31}$
重み	0.188	0.618	0.070	$60 \leq b_1 \leq 70$	$12.5 \leq b_2 \leq 17.5$	
メンバシップ関数値	0.645	0.719	0.723	0.80	0.10	
目的関数値(反復 3)	155.000	7.086	47.000	62.0	17.0	$x_{13}, x_{23}, x_{31}$
重み	0.298	0.500	0.078	$60 \leq b_1 \leq 70$	$12.5 \leq b_2 \leq 17.5$	
メンバシップ関数値	0.645	0.719	0.723	0.80	0.10	

本数値例では、システム設計者は表 5.5 の反復 3 の結果に満足したものとして最良な解を得る。すなわち、サブシステム 1 ではユニット 3, サブシステム 2 ではユニッ

ト3, サブシステム3ではユニット1をそれぞれ選択する. このとき3つの目的関数, すなわちコスト, 信頼度, 作業人員はそれぞれ155.000ドル, 93.2また, 最大の重み(0.500)をもつ信頼度関数は, 目標値5.045に対して7.086(メンバシップ関数値:0.719), 次に大きな重み(0.298)をもつコスト関数は, 目標値100.0に対して155.0(メンバシップ関数値:0.645)となっている. これは, ほぼ背反関係にある信頼度関数とコストの両者のトレード・オフを考慮して, コストに関する目標達成度を改善するために, 信頼度関数に対する重みを反復1よりやや落した結果である. 重量に関する制約のメンバシップ関数値が0.10と低いが, これは得られた解が重量の制約に対する許容の上限に近いので, 重要度の比較的低い(0.06)重量の制約を多少犠牲にして信頼度, コスト等を高めようという意思決定者の意向が反映されたものである. また, この解は通常の制約条件, すなわち

$$g_2(x^1) \leq 65$$

を満たさないので, ファジィ制約条件を採用しなければ実行可能解として選ばれることはなかったものであり, 本手法の有効性を示している.

### 5.3 不完全なFDSをもつ冗長ユニット配分問題

本節で取り上げる負荷状態待機冗長 (loaded standby redundant : LSR) システムの信頼度関数は, その故障のタイプによって, SタイプとMタイプの二つに分けることができる.

#### 5.3.1 Sタイプ故障の負荷状態待機冗長システム

サブシステム内の基本ユニットおよび冗長ユニットからなる待機冗長グループで, FDSが故障すると同時に動作が停止するシステムを, S-LSRシステムと呼ぶ(図2.1).

FDSを伴うS-LSRシステムの信頼度関数  $R(\mathbf{m})$  は,  $n$  をサブシステムの数,  $m_i$  を  $i$  番目のサブシステムのユニット数,  $q_i$  を  $i$  番目のサブシステムにおけるユニットの故障率, また  $\beta_i$  をサブシステム  $i$  のFDSの故障率とすれば, 次式で表される.

$$\begin{aligned} R(\mathbf{m}) &= \prod_{i=1}^n \{1 - Q_i(m_i)\} \\ &= \prod_{i=1}^n \left\{ 1 - \frac{\beta_i q_i + (1 - q_i) q_i^{m_i} (1 - \beta_i)^{m_i}}{1 - q_i + \beta_i q_i} \right\} \end{aligned} \quad (5.25)$$

また, ここでは次に示すような線形なコスト関数を考える.

$$C(\mathbf{m}) = \sum_{i=1}^n p_i m_i \quad (5.26)$$

ここで、 $p_i$  はサブシステム  $i$  のコストを表す。一方、システム資源の制約(システム制約)は次式のように表される。

$$G_r(\mathbf{m}) = \sum_{i=1}^n g_{ri}(m_i) \leq b_r, \quad r = 1, 2, \dots, T \quad (5.27)$$

$$m_i \geq 0, \quad i = 1, 2, \dots, n : \text{integer}$$

$g_{ri}(m_i)$  は  $r$  番目のシステム資源のサブシステム  $i$  に  $m_i$  個の冗長ユニットを配分するときのシステム資源の要求を表す。

ここで扱う S-LSR システムにおける冗長ユニットの最適配分問題とは、システム資源が  $T$  種類で非線形な制約条件式(5.27)のもとで、目的関数として信頼度関数(5.25)およびコスト関数(5.26)をそれぞれ最大化、最小化するような最適な冗長ユニットの配分  $\mathbf{m} = [m_1 \ m_2 \ \dots \ m_n]$  を決定する非線形整数計画問題である。

### 5.3.2 Mタイプ故障の負荷状態待機冗長システム

サブシステム内の待機冗長グループの各ユニットが専用のFDSを持ち、任意のFDSが故障した場合は次のFDSが動作する。このタイプのFDSをもつシステムを、M-LSRシステムと呼ぶ(図2.2)。

M-LSRシステムの信頼度関数  $R(\mathbf{m})$  は、次式で表される。

$$R(\mathbf{m}) = \prod_{i=1}^n [1 - Q_i(m_i)]$$

$$= \prod_{i=1}^n [1 - q_i \{(1 - \beta_i)q_i + \beta_i\}^{m_i - 1}] \quad (5.28)$$

M-LSRシステムの最適信頼性設計問題とは、システム資源が  $T$  種類でかつ非線形な制約条件式(5.27)のもとで、目的関数として式(5.28)で示される非線形な信頼度関数  $R(\mathbf{m})$  を最大化およびコスト関数(5.26)を最小化するような冗長ユニットの配分  $\mathbf{m} = [m_1 \ m_2 \ \dots \ m_n]$  を決定する非線形計画問題である。

なお、各サブシステムのFDSが信頼度1、すなわち  $\beta_i = 0$  である場合には、S-LSR及びM-LSRタイプの信頼度関数は、次式で表される。

$$R(\mathbf{m}) = \prod_{i=1}^n (1 - q_i^{m_i}) \quad (5.29)$$

ここで、不完全なFDSをもつシステム信頼性の最適設計問題において、システム信頼度およびコストのファジィ目標、さらにファジィ制約を次のように設定する。

$$\max \quad R(\mathbf{m}) = \prod_{i=1}^n \{1 - Q_i(m_i)\} \gtrsim h'_1 \quad (5.30)$$

$$\min \quad C(\mathbf{m}) = \sum_{i=1}^n p_i m_i \lesssim h_2 \quad (5.31)$$

s. t.

$$G_r(\mathbf{m}) = \sum_{i=1}^n g_{ri}(m_i) \lesssim b_r, \quad r = 1, 2, \dots, T \quad (5.32)$$

$$m_i, i = 1, 2, \dots, \quad n : \text{integer}$$

$h'_1$  と  $h_2$  はそれぞれ意思決定者が設定する信頼度およびコストに対する目標値である。記号  $\lesssim$  ( $\gtrsim$ ) はファジィ不等号である。

## 5.4 ファジィ多目的信頼性設計 II

### 5.4.1 ファジィ多目的信頼性設計問題

二つの目的関数をもつNIP問題を、0-1LP問題に変換するために次のような0-1変数を導入する。

$$x_{ij} = \begin{cases} 1: & \text{サブシステム } i \text{ に } j \text{ 個のユニットを} \\ & \text{分配するとき} \\ 0: & \text{そうでないとき} \end{cases} \quad (5.33)$$

この0-1変数を用いると、不完全なFDSをもつシステム信頼性の最適冗長ユニット配分の非線形計画問題は、次のようなファジィ多目的0-1LP問題に変換することができる。

$$\max \quad f_k(\mathbf{x}) = \sum_{i=1}^n \sum_{j=l_i}^{u_i} c_{kij} x_{ij} \gtrsim h_k, \quad k = 1, 2 \quad (5.34)$$

s. t.

$$g_r(\mathbf{x}) = \sum_{i=1}^n \sum_{j=l_i}^{u_i} a_{rij} x_{ij} \lesssim b_r, \quad r = 1, 2, \dots, T \quad (5.35)$$

$$g_{r+i}(\mathbf{x}) = \sum_{j=l_i}^{u_i} x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (5.36)$$

$$x_{ij} = 0 \text{ or } 1, \quad i = 1, 2, \dots, n, \quad j = l_i, l_i + 1, \dots, u_i \quad (5.37)$$

ここで、 $h_1$  は式(5.30)で与えられた信頼度に関する目標値  $h'_1$  の自然対数、すなわち

$$h_1 = -\ln h'_1 \quad (5.38)$$

を表す。式(5.36)はGUB制約であり、 $a_{rij}$  はGUB制約  $i$  における  $j$  番目の要素に対応する  $r$  番目の制約条件の係数で、 $a_{rij} = g_{ri}(j)$  で表される。また、 $l_i$  と  $u_i$  はそれぞれサブシステム  $i$  における冗長系の構成ユニット数  $m_i$  の下限値と上限値を表す。一方、 $c_{kij}$  はGUB制約  $i$  における  $j$  番目の要素に対応する  $k$  番目の目的関数の係数で、コスト関数の係数は  $c_{2ij} = jp_i$ 、また信頼度関数の係数  $c_{1ij}$  は、FDSの故障のタイプに応じてそれぞれ次式によって与えられる。

Sタイプ:

$$c_{1ij} = -\ln \left\{ 1 - \frac{\beta_i q_i + (1 - q_i)(q_i)^j (1 - \beta_i)^j}{1 - q_i + \beta_i q_i} \right\} \quad i = 1, 2, \dots, n, \quad j = l_i, \dots, u_i \quad (5.39)$$

Mタイプ:

$$c_{1ij} = -\ln[1 - q_i \{(1 - \beta_i)q_i + \beta_i\}^{j-1}] \quad i = 1, 2, \dots, n, \quad j = l_i, \dots, u_i \quad (5.40)$$

ここで、目的関数  $f_k(\mathbf{x})$  およびシステム制約  $g_r(\mathbf{x})$  に対する反復  $s$  のメンバシップ関数  $\mu_k(f_k(\mathbf{x}))$ , ( $k = 1, 2$ ),  $\mu_{r+2}(g_r(\mathbf{x}))$ , ( $r = 1, 2, \dots, T$ ) をそれぞれ次のように定義する.

$$\mu_k(f_k(\mathbf{x})) = \begin{cases} 1 & ; f_k(\mathbf{x}) < h_k \\ \frac{z_k^- - f_k(\mathbf{x})}{z_k^- - h_k} & ; h_k \leq f_k(\mathbf{x}) \leq z_k^- \\ 0 & ; f_k(\mathbf{x}) > z_k^- \end{cases} \quad k = 1, 2 \quad (5.41)$$

$$\mu_{r+2}(g_r(\mathbf{x})) = \begin{cases} 1 & ; g_r(\mathbf{x}) < b_r - \delta_r / 2 \\ \frac{b_r + \delta_r / 2 - g_r(\mathbf{x})}{\delta_r} & ; b_r - \delta_r / 2 < g_r(\mathbf{x}) \leq b_r + \delta_r / 2 \\ 0 & ; b_r + \delta_r / 2 > g_r(\mathbf{x}), \end{cases} \quad r = 1, 2, \dots, T \quad (5.42)$$

ここで、 $z_k^-$  は式(5.35), (5.36)および(5.37)で示される制約条件のもとで、式(5.34)の目的関数の型を反転(最大化)して得られる目的関数  $f_k(\mathbf{x})$  に対する最悪値 (NIS) を表す. また、 $\delta_r$  は意思決定者によって与えられる右辺定数の幅である.

上記の 0-1LP 問題は、凸ファジィ決定<sup>[81]</sup> (convex fuzzy decision) に従うことにより、GUB構造を伴う単一目的の 0-1LP 問題として定式化することができる.

$$\max \quad z = \sum_{k=1}^2 w_k u_k(f_k(\mathbf{x})) + \sum_{r=1}^r w_{r+2} u_{r+2}(g_r(\mathbf{x})) \quad (5.43)$$

s. t.

$$\sum_{i=1}^n \sum_{j=1}^{u_i} c_{kij} x_{ij} + (z_k^- - h_k) \mu_k(f_k(\mathbf{x})) = z_k^-, \quad k = 1, 2 \quad (5.44)$$

$$\sum_{i=1}^n \sum_{j=1}^{u_i} a_{rij} x_{ij} - \delta_r \mu_{r+2}(g_r(\mathbf{x})) = b_r - \delta_r / 2, \quad r = 1, 2, \dots, T$$

$$0 \leq u_k(f_k(\mathbf{x})), \quad u_{r+2}(g_r(\mathbf{x})) \leq 1, \quad k = 1, 2, \quad r = 1, \dots, T \quad (5.45)$$

式(5.36)および(5.37)

$$(5.46)$$

ただし、 $w_k$  は多評価意思決定のための支援手法として評価されている AHP によって与えられる各目的関数およびシステム制約に対する重みである.

## 5.4.2 計算法

$T$  個のシステム制約と  $n$  個の GUB 制約条件のもとで、二個の目的関数を最小化する不完全な FDS を伴う二目的 0-1 線形計画問題を解くための計算法を次に示す。この計算法は、各目的関数の目標値および制約条件の達成度を最大化する解を、各 GUB 制約の中から効率的に探すことができる。また、この計算法は、前章で提案されたファジィ多目的 0-1 計画問題を解くための計算法を改良したもので、各 GUB 制約の中で実行可能解を探す過程(ステップ 8~ステップ 11)が拡張されている。主な改良点は次の二点である。

1. 各 GUB 制約の中から低いランクの実行可能変数を探す過程(ステップ 9)と残差の改善度の計算(ステップ 10)を二つのステップに分離し、明確化した。
2. 残差の改善度を示す指標を導入することによって、各反復において得られる解に対する評価が明確になった。

次に、この計算法の手順を示す。

ステップ 1: 各目的関数に対する目標値  $h_k$ , ( $k=1,2$ ) およびシステム制約に対する右辺定数の幅  $\delta_r$  ( $r=1,2,\dots,T$ ) を設定する。

ステップ 2: 意思決定者の基本的な設計方針を反映させるために、AHP によって各目的関数およびシステム制約条件の重要度  $w_k$ , ( $k=1,2,\dots,T+2$ ) を計算する。

ステップ 3: 次に示す  $n$  個の GUB 制約をもつ単一目的の 0-1LP 問題

$$\min f_1(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1_i}^{u_i} c_{1ij} x_{ij} \quad (5.47)$$

$$\text{s. t. } g_r(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1_i}^{u_i} a_{rij} x_{ij} \leq b_r + \delta_r/2, \quad r=1,2,\dots,T \quad (5.48)$$

式(5.36)および(5.37)

を解き、理想値  $z_1^*$  (PIS) を求める。さらに、上記の目的関数の型 (min/max) を反転させた(もとの目的関数の型が min なら max に、max なら min に) 単一目的の 0-1LP 問題を解き、最悪値  $z_1^-$  を求める。同様に、 $f_2(\mathbf{x})$  について解き、 $z_2^*$  および  $z_2^-$  を計算する。この 0-1LP 問題でのシステム制約は、最も緩い条件で理想値を得るために、式(5.48)とする。反復回数を  $s=1$  とおく。

ステップ 4: 各目的関数に対する目標値  $h_k$  が、次に示す条件

$$z_k^* \leq h_k < z_k^-, \quad k=1,2 \quad (5.49)$$

を満たすかどうか調べる。満たしていない場合は、ステップ1に戻る。\$s \ge 2\$ の場合は、各目的関数間のトレード・オフを考慮し、重要度の高い目的関数に対する目標値 \$h\_k\$ をできるだけ下げることなく、下位のレベルの目標値を下げることによって再設定を行う。

ステップ5：式(5.33)の0-1変数を導入することにより、式(5.34)～(5.37)に示されるようなGUB制約をもつファジィ多目的0-1LP問題、さらに式(5.41), (5.42)で表されるメンバシップ関数を導入することによって式(5.43)～(5.46)のようなGUB制約をもつ0-1LP問題として定式化する。

ステップ6：各GUB制約において、低いランクの実行可能解を探すために、各GUB制約の集合の変数を昇順にランク付けを行い、そのインデックスを記録する。

$$\gamma_{ij} = \sum_{k=1}^2 c_{kij} / z_k^- + \sum_{r=1}^r a_{rij} / b_r, \quad i = 1, 2, \dots, n, \quad j = l_i, \dots, u_i \quad (5.50)$$

$$\begin{aligned} j_i &= [j_{i1} \quad j_{i2} \quad \dots \quad j_{i(u_i-1)}] \\ &= \underset{j}{\text{indsort}}\{\gamma_{ij} \mid j = l_i, l_i + 1, \dots, u_i\}, \quad i = 1, 2, \dots, n \end{aligned} \quad (5.51)$$

ここで、indsortのindはindexの省略形であり、ソーティング後のインデックスを採用することを意味する。

ステップ7：すべての決定変数の値を0にする。各GUB制約の集合において、最高ランク(ランク1)の変数の値を1にする。

$$\begin{aligned} x_{ij_{i1}} &= 1, \quad i = 1, 2, \dots, n \\ t &= [t_1 \quad t_2 \quad \dots \quad t_n] = [1 \quad 1 \quad \dots \quad 1] \end{aligned}$$

これらの変数の集合を初期解 \$\mathbf{x}\_1\$ とする。

ステップ8：次式から各目的関数および制約条件に対する理想値と達成値の残差 \$d = [d\_1 \quad d\_2 \quad \dots \quad d\_{T+2}]\$ を次式より計算する。

$$\begin{aligned} d_k &= \begin{cases} z_k^* - f_k(\mathbf{x}^1); & f_k(\mathbf{x}^1) < z_k^* \\ 0 & ; z_k^* \leq f_k(\mathbf{x}^1) \leq z_k^- \\ z_k^- - f_k(\mathbf{x}^1); & z_k^- < f_k(\mathbf{x}^1) \end{cases} \quad k = 1, 2 \\ d_{r+2} &= \begin{cases} b_r - \delta_r/2 - g_r(\mathbf{x}^1); & g_r(\mathbf{x}^1) \leq b_r + \delta_r/2 \\ 0 & ; b_r - \delta_r/2 \leq g_r(\mathbf{x}^1) \leq b_r + \delta_r/2 \\ b_r + \delta_r/2 - g_r(\mathbf{x}^1); & b_r + \delta_r/2 < g_r(\mathbf{x}^1) \end{cases} \end{aligned} \quad (5.52)$$

\$r = 1, 2, \dots, T\$

すべての  $k$ , ( $k=1, \dots, T+2$ ) に対し  $d_k=0$  ならば, ステップ12へ行く.

ステップ9: 次式を満たす変数を含む GUB 制約において, さらに低いランクの実行可能変数を探す.

$$i^* = \operatorname{indmax}_i \left\{ \sum_{k=1}^2 (c_{kij^{*+1}} - c_{kij^*}) / z_k^- + \sum_{r=1}^r (a_{rij^{*+1}} - a_{rij^*}) / b_r \mid d_k \neq 0, i = 1, 2, \dots, n \right\} \quad (5.53)$$

ここで,  $\operatorname{indmax}$  の  $\operatorname{ind}$  は  $\operatorname{index}$  の省略形であり, 残差の最大値をとるときのインデックスを採用することを意味する. 残差  $d_k$  の最大値が正の場合は, 式(5.54)において正の最大値をとる変数, また負の場合には負の最大値をとる変数  $k$ , ( $k=1, 2, \dots, T+2$ ) を  $k^*$  と表す.

ステップ10: ステップ9で選ばれた GUB 制約で, 最も大きな値をもつ目的関数またはシステム制約の残差  $d_k$  の改善度  $I_k$  を次式より計算する.

$$I_k = \begin{cases} (c_{ki^*j_{it^*+1}} - c_{ki^*j^*+1}) / z_k^-; & k^* \leq 2 \\ (a_{ki^*j_{it^*+1}} - a_{ki^*j^*}) / b_k^-; & k^* > 2 \end{cases} \quad (5.54)$$

残差  $d_k > 0 (< 0)$  に対する改善度  $I_{k^*}$  が,  $I_{k^*} > 0 (< 0)$  の場合, 選ばれた GUB 制約で一ランク低い変数を1にし, いままで1を持っていた変数を0とする. ステップ8へ戻る. そうでなければ, 式(5.53)においてさらに小さい値をもつ GUB 制約の変数に対して残差を改善する変数を探す. 存在すれば, その変数の値を1とする.

$$x_{i^*j^*+1} = 1, \quad x_{i^*j_{it^*}} = 0$$

$$t_{i^*} = t_{i^*} + 1$$

存在しなければ, 次に低いランクの変数を探す. 最下位ランクの変数の場合は, 式(5.53)において一ランク高い変数との比較を行う.

ステップ11: すべての GUB 制約において, そのような変数が存在しなければ, 非実行可能問題として終了する. 同値の場合は, 各 GUB 制約においてさらに低いランクの変数を選び, その中で最も大きな  $\gamma_{ij}$  の値をもつ変数を探し, ステップ8へ戻る.  $j_i = j_{i+1}$  とおく.

ステップ12: 各GUB制約において, 現在の解より一ランク低い変数を含めて式(5.43)に示されるメンバシップ関数値の加重和を最大にするような実行可能解を探す. 存在する場合には, その変数の値を1にし, 同じGUB制約で1をもっていた変数の値を0にする. ステップ9へ戻る. そのような組合せが存在しない場合は, ステップ13へ行く.

ステップ13: 各目的関数値の達成度(目標の満足度)  $\mu_k$ , ( $k=1, 2, \dots, q$ )がシステム設計者の希望する値より大きければ, 得られた解を最良な妥協解として終了する. そうでなければ, 各目的関数の重要度  $w_k$  ( $k=1, 2, \dots, q$ )を再設定してステップ8に戻る.

なお, 最大化問題の場合は, ステップ3において目的関数の型をminからmaxに換えるとともに, ステップ4の条件を次のように変更する.

$$z_k^- < h_k \leq z_k^*, \quad k=1, 2 \quad (5.40)'$$

さらに, ステップ6におけるランク付けを降順とする.

また,  $g_r(x^1) \gtrsim b_{ri}$  の場合は, ステップ3において制約条件式(5.48)が, 次式と置き換えられる.

$$g_r(x^1) \geq b_r - \delta_r/2, \quad r=1, 2, \dots, T \quad (5.48)'$$

### 5.4.3 数値例

数値例として, 図5.3に示すように四種類の異なる故障のタイプを持つサブシステムが直列に接続された不完全なFDSをもつ直列システム信頼性の最適設計問題を考える. この例では, 三種類の非線形なシステム資源のもとで, システム信頼度の最大化と, コストの最小化をはかる最適なユニット冗長配分問題を取りあげる.

図5.3 直列システムの構成図

$$\begin{aligned}
\widetilde{\max} \quad & R(\mathbf{m}) = \prod_{i=1}^4 R_i(m_i) \lesssim h'_1 \\
\widetilde{\min} \quad & C(\mathbf{m}) = \sum_{i=1}^4 p_i m_i \lesssim h_2 \\
\text{s. t.} \quad & G_1(\mathbf{m}) = \sum_{i=1}^4 a_{1i} (m_i)^2 \lesssim b_1 \\
& G_2(\mathbf{m}) = \sum_{i=1}^4 a_{2i} \{m_i + \exp(m_i/4)\} \lesssim b_2 \\
& G_3(\mathbf{m}) = \sum_{i=1}^4 a_{3i} m_i \exp(m_i/4) \lesssim b_3 \\
& m_i \geq 0, \quad i = 1, 2, 3, 4, : \text{ integer}
\end{aligned}$$

ここで、 $\mathbf{m} = [m_1 \ m_2 \ m_3 \ m_4]$  は各サブシステムの冗長ユニット数を表すほか、信頼度関数はそれぞれ次式で示される。

$$\begin{aligned}
R_1(m_1) &= 1 - q_1 \{(1 - \beta_1)q_1 + \beta_1\}^{m_1+1} \\
R_2(m_2) &= 1 - \frac{\beta_2 q_2 + (1 - q_2)q_2^{m_2}(1 - \beta_2)^{m_2}}{1 - q_2 + \beta_2 q_2} \\
R_3(m_3) &= 1 - (q_3)^{m_3} \\
R_4(m_4) &= 1 - q_4 \{(1 - \beta_4)q_4 + \beta_4\}^{m_4+1}
\end{aligned}$$

各変数の値は、表 5.6 に示されるとおりである。

表 5.6 数値例における係数の値

i	$a_{1i}$	$a_{2i}$	$a_{3i}$	$\beta_i$	$q_i$	$p_i$
1	1	7	7	0.2	0.07	40
2	2	7	8	0.06	0.08	30
3	3	5	8	0.0	0.06	35
4	4	7	6	0.3	0.09	45

ステップ 1: 信頼度およびコストに関する目標値を、次のように設定する。

$$h'_1 = 0.95, \quad h_2 = 300.0$$

また、各システム制約に対する右辺定数の幅  $\delta_r$ , ( $r = 1, 2, 3$ ) をそれぞれ次のように設定する。

$$\delta_1 = 20.0, \quad \delta_2 = 20.0, \quad \delta_3 = 30.0$$

ステップ2: 意思決定者の基本的な設計方針を反映させるために, AHPによって各目的関数およびシステム制約条件の重要度  $w_1, w_2, \dots, w_5$  を計算する.

$$w_1 = 0.618, w_2 = 0.168, w_3 = 0.090,$$

$$w_4 = 0.063, w_5 = 0.060$$

ステップ3: 二つの目的関数について, それぞれ単一目的のLP問題として解き, PISおよびNISを求める. 反復回数を  $s = 1$  とおく.

表 5.7 理想値と最悪値

k	$z_k^*$	$z_k^-$
1	0.0181	0.1966
	(0.982)	(0.822) (信頼度)
2	235	475

ステップ4: 各目的関数に対する目標値  $h_1 = 0.0513 (= -\ln(0.95))$ ,  $h_2 = 300.0$  はともに PIS と NIS の間にあるので, ステップ5へ行く.

ステップ5: この問題は, 次に示すように四つの GUB 制約を持つファジィ二目的 0-1LP 問題として定式化することができる.

$$\begin{aligned} \min f_1(\mathbf{x}) = & 0.0181x_{12} + 0.0046x_{13} + 0.0012x_{14} + 0.0003x_{15} \\ & + 0.0834x_{21} + 0.0109x_{22} + 0.0056x_{23} \\ & + 0.0052x_{24} + 0.0052x_{25} + 0.0619x_{31} \\ & + 0.0036x_{32} + 0.0002x_{33} + 0.0000x_{34} \\ & + 0.0000x_{35} + 0.0332x_{42} + 0.0119x_{43} \\ & + 0.0043x_{44} + 0.0016x_{45} \lesssim 0.05 \end{aligned}$$

$$\begin{aligned} \min f_2(\mathbf{x}) = & 80x_{12} + 120x_{13} + 160x_{14} + 200x_{15} \\ & + 30x_{21} + 60x_{22} + 90x_{23} + 120x_{24} + 150x_{25} \\ & + 35x_{31} + 70x_{32} + 105x_{33} + 140x_{34} + 175x_{35} \\ & + 90x_{42} + 135x_{43} + 180x_{44} + 225x_{45} \lesssim 300.0 \end{aligned}$$

s. t.

$$\begin{aligned} & 4.0x_{12} + 90x_{13} + 16.0x_{14} + 25.0x_{15} \\ & + 2.0x_{21} + 8.0x_{22} + 18.0x_{23} + 32.0x_{24} + 50.0x_{25} \end{aligned}$$

$$\begin{aligned}
& +3.0x_{31} + 12.0x_{32} + 27.0x_{33} + 48.0x_{34} + 75.0x_{35} \\
& +16.0x_{42} + 36.0x_{43} + 64.0x_{44} + 100.0x_{45} \lesssim 100.0 \\
& 25.541x_{12} + 35.819x_{13} + 47.028x_{14} + 59.432x_{15} \\
& +15.988x_{21} + 25.541x_{22} + 35.819x_{23} + 47.028x_{24} \\
& +59.432x_{25} + 11.420x_{31} + 18.244x_{32} + 25.585x_{33} \\
& +33.591x_{34} + 42.459x_{35} + 32.839x_{42} + 46.053x_{43} \\
& +60.465x_{44} + 76.413x_{45} \lesssim 150.0 \\
& 23.082x_{12} + 44.457x_{13} + 76.112x_{14} + 122.16x_{15} \\
& +10.272x_{21} + 26.380x_{22} + 50.808x_{23} + 86.985x_{24} \\
& +10.272x_{31} + 26.380x_{32} + 50.808x_{33} + 86.985x_{34} \\
& +19.785x_{42} + 38.106x_{43} + 65.239x_{44} + 104.71x_{45} \\
& \lesssim 160.0
\end{aligned}$$

$$\sum_{j=2}^5 x_{1j} = \sum_{j=1}^5 x_{2j} = \sum_{j=1}^5 x_{3j} = \sum_{j=2}^5 x_{4j} = 1 \quad (5.55)$$

$$x_{ij} = 0 \text{ or } 1, \quad i = 1, 2, \dots, 4, \quad j = 1, 2, \dots, 5 \quad (5.56)$$

さらに、この問題は次のように変換される。

$$\max \quad 0.618\mu_1 + 0.168\mu_2 + 0.090\mu_3 + 0.063\mu_4 + 0.060\mu_5$$

s. t.

$$\begin{aligned}
& 0.0181x_{12} + 0.0046x_{13} + 0.0012x_{14} + 0.0003x_{15} \\
& +0.0834x_{21} + 0.0109x_{22} + 0.0056x_{23} + 0.0052x_{24} \\
& +0.0052x_{25} + 0.0619x_{31} + 0.0036x_{32} + 0.0002x_{33} \\
& +0.0000x_{34} + 0.0000x_{35} + 0.0332x_{42} + 0.0119x_{43} \\
& +0.0043x_{44} + 0.0016x_{45} + 0.1466\mu_1 = 0.1966 \\
& 80x_{12} + 120x_{13} + 160x_{14} + 200x_{15} \\
& +30x_{21} + 60x_{22} + 90x_{23} + 120x_{24} + 150x_{25} \\
& +35x_{31} + 70x_{32} + 105x_{33} + 140x_{34} + 175x_{35} \\
& +90x_{42} + 135x_{43} + 180x_{44} + 225x_{45} + 145\mu_2 = 445.0 \\
& 4.0x_{12} + 90x_{13} + 16.0x_{14} + 25.0x_{15} \\
& +2.0x_{21} + 8.0x_{22} + 18.0x_{23} + 32.0x_{24} + 50.0x_{25}
\end{aligned}$$

$$\begin{aligned}
&+3.0x_{31} + 12.0x_{32} + 27.0x_{33} + 48.0x_{34} + 75.0x_{35} \\
&+16.0x_{42} + 36.0x_{43} + 64.0x_{44} + 100.0x_{45} + 20\mu_3 \\
&= 110.0
\end{aligned}$$

$$\begin{aligned}
&25.541x_{12} + 35.819x_{13} + 47.028x_{14} + 59.432x_{15} \\
&+15.988x_{21} + 25.541x_{22} + 35.819x_{23} + 47.028x_{24} \\
&+59.432x_{25} + 11.420x_{31} + 18.244x_{32} + 25.585x_{33} \\
&+33.591x_{34} + 42.459x_{35} + 32.839x_{42} + 46.053x_{43} \\
&+60.465x_{44} + 76.413x_{45} + 20\mu_4 = 160.0
\end{aligned}$$

$$\begin{aligned}
&23.082x_{12} + 44.457x_{13} + 76.112x_{14} + 122.16x_{15} \\
&+10.272x_{21} + 26.380x_{22} + 50.808x_{23} + 86.985x_{24} \\
&+10.272x_{31} + 26.380x_{32} + 50.808x_{33} + 86.985x_{34} \\
&+19.785x_{42} + 38.106x_{43} + 65.239x_{44} \\
&+104.71x_{45} + 30\mu_5 = 175.0
\end{aligned}$$

式(5.55) および式(5.56)

ステップ6: 式(5.50)を用いて各GUB制約の集合の変数を,昇順にランク付けする.

表5.8 ランク付けされた変数(( )内は,  $\gamma_{ij}$ の値を示す)

ランク t	1	2	3	4	5
1	$x_{12}(0.588)$	$x_{13}(0.883)$	$x_{14}(1.641)$	$x_{15}(1.832)$	
2	$x_{22}(0.597)$	$x_{21}(0.678)$	$x_{23}(0.954)$	$x_{25}(1.303)$	$x_{24}(1.456)$
3	$x_{31}(0.559)$	$x_{32}(0.572)$	$x_{33}(0.980)$	$x_{34}(1.542)$	$x_{35}(1.945)$
4	$x_{42}(0.861)$	$x_{43}(1.195)$	$x_{44}(1.852)$	$x_{45}(2.646)$	

ステップ7: すべての決定変数の値を0にする. 各GUB制約の集合において, 最高ランク(ランク1)の変数の値を1にする.

$$x^1 = \{x_{12}, x_{22}, x_{31}, x_{42}\}$$

初期解の各GUB制約中のランキングは, 式(5.51)より次のようになる.

$$t = [1 \quad 1 \quad 1 \quad 1]$$

[反復1]

ステップ8: 初期解に対する各制約の値は次のとおりである.

$$f_1(x^1) = 0.1241 \in [0.0234, 0.1966]$$

$$f_2(\mathbf{x}^1) = 265 \in [235, 475]$$

$$g_1(\mathbf{x}^1) = 31 \notin [90, 110]$$

$$g_2(\mathbf{x}^1) = 95.341 \notin [140, 160]$$

$$g_3(\mathbf{x}^1) = 79.519 \notin [145, 175]$$

これより、残差 $d$ は次のようになる。

$$d = [0 \quad 0 \quad 59 \quad 44.659 \quad 65.481]$$

ステップ9:

$$\begin{aligned} i^* &= \underset{i}{\text{indmax}}\{0.12, -0.206, 0.09, 0.375\} \\ &= 4 \end{aligned}$$

従って、四番目のGUB制約の中でさらに低いランクの変数を探す。

ステップ10: 最も大きい残差 $d_5 > 0$ に対する改善度 $I_5$ は、

$$I_5 = (38.106 - 19.785) / 175 > 0$$

であるから、四番目のGUB制約の中から、1ランク低い変数として $x_{43}$ が選ばれる。 $x_{43} = 1, x_{42} = 0$

$$\mathbf{x}^1 = \{x_{12}, x_{22}, x_{31}, x_{43}\}$$

$$t^* = [1 \quad 1 \quad 1 \quad 2]$$

ステップ8へ戻る。

この後、9回の反復により $d = \mathbf{0}$ となるので、ステップ12へ行く。このときの解は $\mathbf{x}^1 = \{x_{13}, x_{23}, x_{31}, x_{44}\}$ である。

ステップ12: 式(5.43)により、メンバシップ関数の加重和は0.6623となる。これより小さな値をとる実行可能変数は存在しないから、ステップ13へ行く。

ステップ13: この問題を解いた結果、 $x_{13} = x_{23} = x_{31} = x_{44} = 1$ , その他の変数は0が得られる。

目的関数  $f_1(\mathbf{x})$  の目標値 0.05(信頼度 0.95) に対して 0.0764(信頼度 0.73),  $f_2(\mathbf{x})$  の目標値 300.0 に対して 425.0, またシステム制約条件式の値  $g_1(\mathbf{x})$  は  $94(90 \leq b_1 \leq 110)$ ,  $g_2(\mathbf{x})$  は  $143.52(130 \leq b_2 \leq 150)$ , さらに  $g_3(\mathbf{x})$  は  $170.776(145 \leq b_1 \leq 175)$  となっている. また, このときのメンバシップ関数値はそれぞれ

$$\mu_1 = 0.820, \quad \mu_2 = 0.138, \quad \mu_3 = 0.800,$$

$$\mu_4 = 0.824, \quad \mu_5 = 0.141$$

である. 重要度に対応した重みを考慮すると, 一番大きなウェイトを持つ信頼度の満足度が低い(0.820)ほか, メンバシップ関数値  $\mu_5$  が 0.141 となっており, 目標値に対する達成度が不十分である.

本数値例では, 意思決定者がこの結果に対して各目標に対する目標値を十分に満足しているとはいえないと判断したものとして, 信頼度に対する重みを 0.618 から 0.650 に上げるとともに, コストに対する重みを 0.168 から 0.138 に下げ, ステップ 8 へ戻る.

表 5.9 提案する手法での各反復時の結果

	目的関数値		制約条件の右辺定数			値 1 をもつ決定変数 (他の変数は 0)
	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$g_1(\mathbf{x})$	$g_2(\mathbf{x})$	$g_3(\mathbf{x})$	
反復 1	0.73	425.0	94.0	143.52	170.776	$x_{13}, x_{23}, x_{31}, x_{44}$
重み $w_k$	0.618	0.168	$90 \leq b_1 \leq 110$	$130 \leq b_2 \leq 150$	$145 \leq b_3 \leq 175$	
メンバシップ関数値	0.820	0.138	0.800	0.824	0.141	
反復 2	0.92	425.0	84.0	134.79	146.343	$x_{13}, x_{23}, x_{31}, x_{44}$
重み $w_k$	0.650	0.136	$90 \leq b_1 \leq 110$	$130 \leq b_2 \leq 150$	$145 \leq b_3 \leq 175$	
メンバシップ関数値	0.78	0.44	0.867	0.892	0.955	
反復 3	0.998	430.0	94.0	140.05	162.456	$x_{13}, x_{22}, x_{32}, x_{44}$
重み $w_k$	0.700	0.186	$90 \leq b_1 \leq 110$	$130 \leq b_2 \leq 150$	$145 \leq b_3 \leq 175$	
メンバシップ関数値	0.981	0.103	0.850	0.997	0.418	

#### 5.4.4 評価

この後, 本数値例では, 意思決定者は表 5.9 の反復 3 の結果に満足したのものとして最良な解を得る. すなわち, サブシステム 1 からサブシステム 4 にそれぞれ 3, 2, 2, 4 個のユニットを配分する. このとき, 信頼度とコストはそれぞれ 0.998, 430 となる. また, 最大の重み (0.70) をもつ信頼度関数に対する満足度が高く (メンバシップ関数値: 0.981), 次に大きな重み (0.186) をもつコスト関数は, 目標値 300 に対して 430 (メンバシップ関数値: 0.103) となっている. これは, ほぼ背反関係にある信頼度関数とコストの両者のトレード・オフを考慮して, 信頼度に関する目標達成度を改善するために, コストに対する目標値を反復 1 より落した結果である. ここで, コストに関する制約のメンバシップ関数値が 0.103 と低い, これは得られた解がコストに関する最悪値に近い, コストの満足度を多少犠牲にして信頼度を高

めようという意思決定者の意向が反映されたものである。

## 5.5 むすび

本章では、信頼性最適化問題においてより現実的な問題に即した解を得ることができる、新しいファジィ多目的0-1計画問題の一解法を提案した。さらに、提案解法は次に示すような特徴を持つことを示した。

- GUB構造を考慮した効率的な解法をより一般的な多目的0-1計画問題に拡張したほか、ファジィ理論の導入により各目的関数に対し、システム設計者の主観的判断に起因するあいまいさを考慮して”およそ設定した目標値くらい”というかたちで目標値を与えることができる。
- また、各目的関数や制約条件に関して、多評価意思決定のための支援手法として評価されているAHPに基づいた重要度に対応する重みを与えることが可能であるほか、システムの使用環境等により変化するシステム制約条件の右辺定数に上限と下限を設定することによってゆとりをもたせることができるため、システム設計者が満足するような最適設計を行うことができる等、多くの優れた特徴を備えていることを明らかにした。
- 切換え装置の故障を考慮した不完全なFDSをもつシステムの冗長ユニット配分問題において、目的関数に対するファジィ目標を設定できるため、目標値に対する意思決定者のあいまいさを考慮に入れることが可能であるとともに、各目的関数および制約条件に対する目標値の達成度を最大にするような決定変数を、各GUB制約の中から効率的に選ぶことができるように改良した解法を提案した。
- システムの使用環境の変化等を考慮することができるため、より現実的な最適信頼性設計が可能である。

## 第 6 章

# 遺伝的アルゴリズムによるファジィ信頼性の最適設計

### 6.1 まえがき

工業製品の競争力の大きな要素の一つは、信頼性を含む製品の品質である。製品の信頼性を高めるためには、現状の技術水準に基づいた最適な設計を行うことが要求される<sup>[5, 68]</sup>。一般に非線形整数計画 (nonlinear integer programming:NIP) 問題として定式化される信頼性最適化問題の解法は、これまでに数多く提案されているが、大きく分類すると動的計画法 (Dynamic Programming), 整数計画法 (Integer Programming:IP), ヒューリスティックな解法, ラグランジュ乗数法等が挙げられる<sup>[120, 69]</sup>。

信頼性を高めるための技法の一つに、冗長系の採用がある。システムは一般に直列/並列の冗長ユニットからなるサブシステムから構成されることから、サブシステムの最適な設計を行うことによってシステムの信頼性を高めることができる。現実のシステムでは、ユニットの故障はそのタイプによって数種類の故障モードに分類することができる。システム信頼性の最適設計を行うにあたり、システム内部の構成部品の故障の結果発生するシステム全体の故障を数種類の故障モードに分類して考えることは、システム故障の系統的な解析にとって有効な方法の一つである。Tillman *et al.* は数種類の故障モードを伴う信頼性最適化問題の解法を提案している<sup>[119]</sup>。Genはこの問題を0-1IP問題に変換して解くことによって、より効率的に解くことができることを示した<sup>[10]</sup>。また、Hwang *et al.* は非線形な信頼性最適化問題を解くために目標計画法を導入した有効な解法を提案した<sup>[31]</sup>。さらに、大規模なシステム特有の構造 (generalized upper bounding:GUB) を利用した効率的な解法も提案されている<sup>[13]</sup>。

一方、目的関数値に対する意思決定者のあいまいさを考慮するためのファジィ目標、またシステムの使用環境や経済情勢などにより変動すると考えられる制約条件の右辺定数の変化を考慮するためにファジィ制約を導入したファジィ数理計画法は、あいまいな状況下における意思決定問題をモデル化するためのより現実的で柔軟な手法として評価されている<sup>[47, 49]</sup>。Gen *et al.* は、ファジィ目標およびファジィ制約を伴う信頼性の最適化問題の解法を提案している<sup>[20, 100]</sup>が、より現実的な問題である数種類の故障モードを伴うシステム信頼性の最適設計問題については言及していない。

近年、遺伝的アルゴリズム (GA) は、組合せ最適化問題に対して有効な手法として注目されている<sup>[26, 52]</sup>。GA は NIP 問題として定式化されるシステム信頼性の最適設計問題などに対しても効果的なツールの一つとして期待される。特に、問題を非線形のまま解くことができるため、大規模な問題などに対してはコンピュータのメモリ容量などの点からも有利である<sup>[21, 41]</sup>。横田らは、不完全な故障切換え装置を伴うシステムおよび複数の故障モードを伴うシステムの信頼性最適設計問題に対して、GA を導入した解法を提案している<sup>[125, 127]</sup>が、システム設計者の主観的判断に基づくあいまいさや、システムのおかれた環境の変化への対応については考慮されていなかった。また、坂和らは、線形のファジィ多目的 0-1 計画問題に対する改良型の GA を提案し、その有効性について明らかにしている<sup>[86]</sup>が、より一般の非線形の目的関数を伴う問題の解法については今後の課題としている。

このような観点から、本章では、数種類の故障モードを伴い、ファジィ目標とファジィ制約をもつシステムの信頼性最適設計問題の解法として、GA を応用した実用的な解法を新しく提案する。この解法は、次のような特徴を持っている。

- 意思決定者であるシステム設計者の主観的判断のあいまいさを考慮するためのファジィ目標、およびシステムのおかれた環境の変化に対応した制約条件の右辺定数の変動を考慮するためのファジィ制約をもつ。
- 数種類の故障モードを考慮することによって、より現実のシステム信頼性最適設計問題に対応しやすい。
- システム設計段階において、一般に相反する目標であると考えられるシステム設計者の要求するシステムの信頼度と、システムの重量、容積、コストなどに関する制約のトレードオフ関係を同時に考慮できるとともに、それらのあいまいさを考慮できる。そのため、システム設計者にとってより柔軟で満足度の高いシステム設計が可能となる。
- GA を応用することにより、システム信頼性の最適設計問題を非線形のまま扱

うことができるため、取り扱いが容易なうえ、コンピュータメモリの点でも有利である。

- 一般に高精度の近似最適解を保証する Goldberg の提案した単純 GA (SGA) [26, 86] を導入することによって、最良解を比較的簡単なアルゴリズムによって得ることが可能となっている。

次節では、ファジィ目標およびファジィ制約をもつ非線形な信頼性最適設計モデルを定式化するとともに、第 6.3 節で、GA に関する基本的な概念について説明する。また第 6.4 節では、提案する解法のアルゴリズムを示し、第 6.5 節でその数値実験結果を報告する。

## 6.2 ファジィ信頼性最適設計問題

システム信頼性の最適設計を行うにあたり、システム内部の構成部品の故障の結果発生するシステム全体の故障を数種類の故障モードに分類して考えることは、システム故障の系統的な解析にとって有効な方法の一つである。複数の冗長ユニットから構成されるサブシステムの故障確率は、サブシステムの処理機能によって冗長ユニットが直列に接続される場合と並列に接続される場合で異なる。現実のシステムはこうした直列/並列のユニットが複雑に接続されることにより構成されると考えられるため、数種類の故障モードという概念を導入することによって、系統的な解析を行うことができ、よりきめ細かい現実のシステム信頼性の最適設計問題に対応することが可能となる。また、システム信頼性の設計段階において、システム設計者が設定するシステム信頼度に対する目標値は、システムの重量、容積、コストなどの制約とのトレードオフを考慮する必要性などから、厳密に設定すべき値というよりはむしろ”ある一定の値以上になることが望ましい”といったあいまいさを含むと一般に考えられる。さらに、制約条件に関しても、システムの使用環境の変化等を考慮すると、”およそある値以下にしたい”といった設計をおこなうことができれば、より現実的で柔軟なシステムの最適設計を行うことが可能となると考えられる。以上のような観点から、本節では、数種類の故障モードを考慮した、ファジィ目標およびファジィ制約を伴うシステム信頼性の最適設計問題のモデルを提案する。まず、本節で取り扱う数種類の故障モードを伴う信頼性最適設計問題では、次の条件を仮定する。

1. サブシステム  $i$  のすべてのユニットは故障モード  $u$  に関して独立である。
2.  $n$  個のサブシステムのうち、少なくとも一つが故障した場合、システム故障が発生する。

3. すべての故障は、いずれか一つの故障モードに区分される。
4. サブシステム*i*において、 $m_i$ 個のユニットのうちいくつかはA故障モード（冗長ユニットの少なくとも一つが正常の場合、そのサブシステムは正常）であり、残りはO故障モード（冗長ユニットの少なくとも一つが故障の場合、そのサブシステムは故障）である。

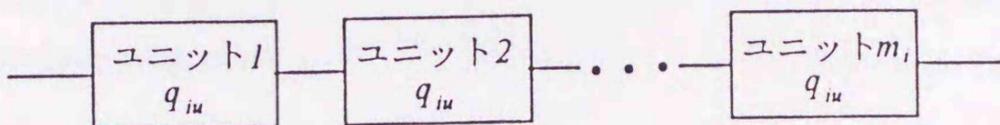


図6.1 O故障モードを伴うサブシステム*i*の論理ダイアグラム

O故障モードは、図6.1に示すようにサブシステム*i*において $m_i$ 個のユニットが直列に配列される場合で、このときサブシステム*i*の故障確率 $Q^O(m_i)$ は、次式で表される。

$$Q^O(m_i) = \sum_{u=1}^{d_i} \{1 - (1 - q_{iu})^{m_i}\} \quad (6.1)$$

ここで、 $q_{iu}$ はサブシステム*i*における冗長ユニットuの故障確率、 $m_i$ はサブシステム*i*の冗長ユニット数、さらに $d_i$ はサブシステム*i*においてO故障モードをもつユニット数を表す。

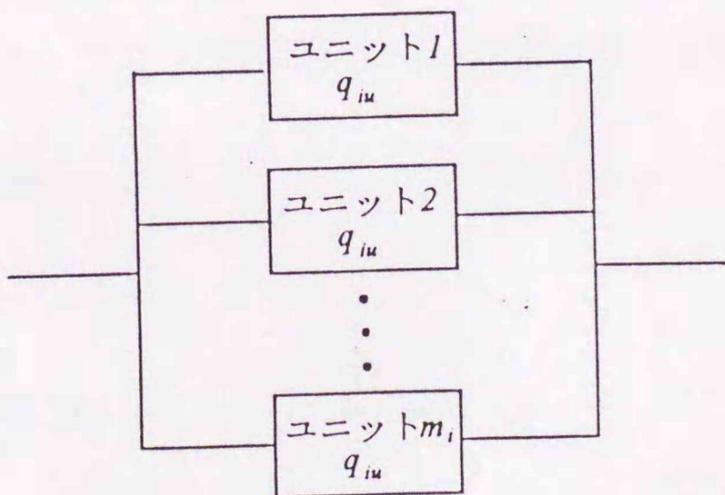


図6.2 A故障モードを伴うサブシステム*i*の論理ダイアグラム

一方、A故障モードは図6.2に示すようにサブシステム*i*において $m_i$ 個のユニットが並列に配列される場合で、このときサブシステム*i*の故障確率 $Q^A(m_i)$ は、次式で表される。

$$Q^A(m_i) = \sum_{u=d_i+1}^{s_i} q_{iu} m_i \quad (6.2)$$

ここで、 $s_i$  はサブシステム  $i$  の故障モードの種類を表す。

ファジィ目標およびファジィ制約を考慮した、数種類の故障モードを伴う信頼性最適設計モデルは、次のように定式化される。このモデルは、限られたシステムの重量、容積、コストなどに関するファジィ制約の下で、システム信頼度を最大化（ファジィ最大化）する最適冗長ユニット配分の決定を行うファジィ非線形整数計画（fuzzy nonlinear integer programming:f-NIP）問題 Opt-1 として表される。

[Opt-1]

$$R(\mathbf{m}) = \prod_{i=1}^n \{1 - (Q^O(m_i) + Q^A(m_i))\} \geq h_0 \quad (6.3)$$

$$\text{s. t. } G_r(\mathbf{m}) = \sum_{i=1}^n g_{ri}(m_i) \leq b_r, \quad r = 1, 2, \dots, T \quad (6.4)$$

$$1 \leq m_i \leq m_i^U; \text{integer}, \quad i = 1, 2, \dots, n \quad (6.5)$$

記号  $\leq$  ( $\geq$ ) はファジィ不等号、また  $m_i^U$  はサブシステム  $i$  に対して配分可能な最大冗長ユニット数を表す。

ここで、信頼度関数に対するメンバシップ関数  $\mu_0(\mathbf{m})$  を次のとおり定義する。

$$\mu_0(\mathbf{m}) = \begin{cases} 1 & ; R(\mathbf{m}) > h_0 \\ \frac{R(\mathbf{m}) - z_0^-}{h_0 - z_0^-} & ; h_0 \geq R(\mathbf{m}) \geq z_0^- \\ 0 & ; z_0^- > R(\mathbf{m}) \end{cases} \quad (6.6)$$

$h_0, z_0^-$  はそれぞれシステム設計者によって設定される信頼度関数の目標値および最悪値である。

一方、制約条件  $G_r(\mathbf{m})$  に対するメンバシップ関数  $\mu_r(\mathbf{m})$  は、次のように定式化される。

$$\mu_r(\mathbf{m}) = \begin{cases} 1 & ; G_r(\mathbf{m}) < b_r \\ 1 - (G_r(\mathbf{m}) - b_r) / l_r & ; b_r \leq G_r(\mathbf{m}) \leq b_r + l_r \\ 0 & ; G_r(\mathbf{m}) > b_r + l_r \end{cases} \quad (6.7)$$

ここで、 $l_r$  ( $r = 1, 2, \dots, T$ ) はシステム設計者によって与えられる制約条件  $r$  の許容量である。

これらのメンバシップ関数を導入することにより、Opt-1 は次のような単一目的の目的関数をもつ NIP 問題 Opt-2 に変換される。

[Opt-2]

$$\max \quad z(\mathbf{m}) = \sum_{t=0}^T w_t \mu_t(\mathbf{m}) \quad (6.8)$$

$$\text{s. t.} \quad R(\mathbf{m}) - (h_0 - z_0^-) \mu_0(\mathbf{m}) = z_0^- \quad (6.9)$$

$$(\mu_r(\mathbf{m}) - 1)l_r + G_r(\mathbf{m}) = b_r \quad (6.10)$$

$$r = 1, 2, \dots, T$$

$$1 \leq m_i \leq m_i^U; \text{integer} \quad (6.11)$$

ここで、 $w_r$  は信頼度関数  $R(\mathbf{m})$  およびシステム制約  $G_r(\mathbf{m})$  の重要度に対応する重みで、 $\sum_{r=0}^T w_r = 1$  である。このモデルは、システム信頼度に関する目標値の達成度について、システムの設計者の満足度と、限られたシステムの重量、容積、コストなどの制約の達成度およびそれらの重要度を同時に考慮することを目的として、信頼度関数および制約条件に対するメンバシップ関数値を最大化する冗長ユニット数  $m_i, i = 1, 2, \dots, n$  の決定問題として定式化されている。

## 6.3 遺伝的アルゴリズムによる解法

### 6.3.1 遺伝的操作

本章で取り上げるGAにおける遺伝的操作は、次のとおりである。

**交叉 (crossover)** : 交叉は、二つの親染色体間でそれぞれの部分解を交換する操作で、両親の良い部分解を組み合わせることにより、さらに良い解を得る効果をもつ。本論文で適用するSGAに基づく単純交叉は交叉する位置を一つ決めて、その前と後でどちらかの親染色体の部分を入れ換える方法で、一点交叉とも呼ばれる。

**突然変異 (mutation)** : 突然変異は、染色体集団内に新しい血を導入して、集団の多様性を保持するための操作で、遺伝子表現のビット列で、確率的に選んだあるビットを反転させるだけであるが、交叉では生成できなかったパターンを作ることができるので、その効果は無視できない。

**選択 (selection)** : SGAでもっとも重要な操作が選択である。一般には、評価関数による適合性の高い染色体を優先して残すことになる。本章で適用されるSGAに基づくエリート保存法は、染色体集団の中で適合度の高いものから次の世代に残す方法である。

### 6.3.2 評価関数

染色体の適合度は、評価関数によって評価される。本章で提案する計算法では、評価関数として Opt-2 における目的関数値，すなわちファジィ目標とファジィ制約を同時に考慮した関数  $z_k(\mathbf{m})$  を用いる。  $z_k(\mathbf{m})$  は、  $k$  番目の集団に対する目的関数の満足度および制約条件の達成度を表す。

$$eval(\mathbf{V}_k) = \begin{cases} z_k(\mathbf{m}); & k = 1, 2, \dots, pop-size, \\ & \text{すべての制約を満たすとき} \\ 0; & \text{そうでないとき} \end{cases} \quad (6.12)$$

ここで、  $\mathbf{V}_p$  は同一集団内における  $k$  番目の染色体を表す。

$$\mathbf{V}_p = [m_{p1} \ m_{p2} \ \dots \ m_{pn}] \quad (6.13)$$

次式により、各世代から最良な遺伝子を選択する。

$$\mathbf{V}^* = \underset{\mathbf{V}_p}{\operatorname{argmax}} \{eval(\mathbf{V}_p) | p = 1, 2, \dots, pop-size\} \quad (6.14)$$

ここで、  $\operatorname{argmax}$  の  $\operatorname{arg}$  は  $\operatorname{argument}$  の省略形で、関数  $eval(\mathbf{V}_p)$  の最大値  $\mathbf{V}_p$  を採用することを意味する。

### 6.3.3 計算法

数種類の故障モードを伴うファジィ信頼性最適設計問題を GA によって解く計算手順を次に示す。提案する解法では、ステップ 2 においてシステム設計者がシステム信頼度と制約条件に対するファジィ目標およびファジィ制約をそれぞれ設定できるようになっている。また、ステップ 2 では、数種類の故障モードを考慮するために、Opt-1 において式 (6.3) で表された信頼度関数を取り扱っている。さらに、ステップ 4 以下では、SGA を導入しているため、比較的簡単なアルゴリズムとなっている点に特徴がある。

ステップ 1 集団サイズ  $pop-size$ 、交叉確率  $p_c$ 、突然変異確率  $p_m$  および最大世代数  $maxgen$  を設定する。初期の世代数  $gen = 1$  の評価関数値  $maxeval$  を 0 とおく。

ステップ 2 信頼度関数  $R(\mathbf{m})$  に対する最悪値  $z_0$  および目標値  $h_0$  をそれぞれシステム設計者が設定する。各制約条件  $G_r(\mathbf{m})$ , ( $r = 1, 2, \dots, T$ ) に対する許容量  $l_r$  を設定する。さらに、信頼度関数および各制約条件に対する重要度  $w_r$  を設定する。

ステップ3 f-NIP問題として定式化される故障モードを伴うシステム信頼性の最適設計問題 (Opt-1) を, 単一目的のNIP問題 (Opt-2) に変換する.

ステップ4  $n$ 要素からなる初期染色体  $V_p$  の集団 ( $p = 1, 2, \dots, pop\_size$ ) をランダムに生成する.

Opt-2の制約条件を満たす染色体が存在しない場合は, 実行可能解が存在しないものとして終了する.

ステップ5  $gen = gen + 1$  とし, 式(6.12)より評価関数値  $eval(V_p)$  を求める.

ステップ6 以下の三つの遺伝的操作を行う.

(6.1) 交叉:

procedure crossover

begin

交叉によって生成された染色体の数を  $ccnt = 0$  とする.

乱数  $r_p$ , ( $p = 1, 2, \dots, pop\_size$ ) を生成する.

$r_p < p_c$  を満たす  $V_p$  を選択する.

選択された  $V_p$  を任意の  $k$  ペアとする.

while ( $ccnt \leq 2k$ ) do

begin

$ccnt = ccnt + 2$  とする.

pos をランダムに決めて交叉を行う.

生成された染色体を  $V'_{ccnt-1}$ ,

$V'_{ccnt}$  とする.

end

end

(6.2) 突然変異:

procedure mutation

begin

突然変異によって生成された染色体の数を

を  $mcnt = 0$  とおく. 乱数  $r_p$

( $p = 1, 2, \dots, n * pop\_size + n * ccnt$ )

を生成する.

$r_p < p_m$  を満足する要素を選択して

$mcnt = mcnt + 1$  とする.

**while** (すべての染色体について処理) **do**

**begin**

    選択された要素  $m_i$  に  $1 \leq m_i \leq m_i^U$

    の範囲で突然変異を起こす.

    生成された染色体を  $V'_{ccnt+mcnt}$

    とする.

**end**

**end**

(6.3) 選択 :

**procedure** selection

**begin**

  生成された染色体の評価関数値

$eval(V'_t)$ , ( $t = 1, 2, \dots, ccnt + mcnt$ )

  を計算する.

  制約条件を満たしていない染色体には

  ペナルティをつける.

**while** (すべての染色体について処理) **do**

**begin**

    親の染色体  $V_p$  ( $p = 1, 2, \dots, pop\_size$ )

    と新たに生成された子の染色体  $V_t$

    ( $t = 1, 2, \dots, ccnt + mcnt$ ) の

    中から評価関数値の大きい順に

$pop\_size$  の数だけ選択する.

    次の世代の  $V_p$  とする.

**end**

**end**

ステップ7  $maxeval < \max\{eval(V_p)\}$  ならば,  $V^*$  を次式より計算する.

$$V^* = \underset{V_p}{\operatorname{argmax}}\{eval(V_p) | p = 1, 2, \dots, pop\_size\}$$

$maxeval = \max\{eval(V_p)\}$  とする.

ステップ8  $gen < maxgen$  ならば, ステップ5に戻る. そうでなければ, 得られた値を出力して終了する.

### 6.3.4 数値実験

ここでは, 数値実験例としてファジィ目標およびファジィ制約を考慮した, 4種類の故障モードを伴うシステム信頼性の最適設計問題を取りあげる. この問題では, 非線形な三つのファジィ制約の下で, 非線形な信頼度関数を目標値に近づけるような, 最適冗長ユニット配分数を決定する.

$$\begin{aligned}
 R(\mathbf{m}) &= \prod_{i=1}^6 [1 - \sum_{u=1}^{d_i} \{1 - (1 - q_{iu})^{m_i+1}\} \\
 &\quad - \sum_{u=d_i+1}^{s_i} (q_{iu})^{m_i+1}] \geq h_0 \\
 \text{s. t. } G_1(\mathbf{m}) &= (m_1 + 3)^2 + (m_2)^2 + (m_3)^2 \\
 &\quad + (m_4 + 3)^2 + (m_5)^2 + (m_6)^2 \leq 51 \\
 G_2(\mathbf{m}) &= 20 \sum_{i=1}^6 \{m_i + \exp(-m_i)\} \geq 260 \\
 G_3(\mathbf{m}) &= 20 \sum_{i=1}^6 \{m_i \exp(-m_i/4)\} \geq 140 \\
 \mathbf{m} &= [m_1 \cdots m_6], m_i: \text{integer for } i = 1, 2, \dots, 6
 \end{aligned}$$

各サブシステムは4つの故障モードを持ち ( $s_i = 4$ ),  $i = 1, \dots, 6$  に対し1つのO故障 ( $d_i = 1$ ) と3つのA故障を伴う. 各サブシステムの故障モードおよび故障確率は表6.1に示すとおりである.

ステップ1 集団サイズ  $pop\_size = 5$ , 交叉確率  $p_c = 0.3$ , 突然変異確率  $p_m = 0.2$  および最大世代数  $maxgen = 300$  とする. 評価関数値を  $maxeval = 0$  とおく.

ステップ2 信頼度関数の最悪値  $z_0^- = 0.86$  および目標値  $h_0 = 0.93$  をそれぞれシステム設計者が設定する. 各制約条件の許容量をそれぞれ  $l_1 = 14.0$ ,  $l_2 = 8.0$  および  $l_3 = 8.0$  と設定する. さらに, 重要度をそれぞれ  $w_0 = 0.85$ ,  $w_1 = 0.05$ ,  $w_2 = 0.05$ ,  $w_3 = 0.05$  と設定する.

ステップ3 Opt-1は, NIP問題 (Opt-2) として次のように変換される.

$$\begin{aligned}
 \max \quad z(\mathbf{m}) &= \sum_{t=0}^3 w_t \mu_t(\mathbf{m}) \\
 \text{s. t. } R(\mathbf{m}) - (h_0 - z_0^-) \mu_0(\mathbf{m}) &= z_0^-
 \end{aligned}$$

表 6.1. 各サブシステムの故障モードおよび故障確率

サブシステム $i$	故障モード	故障確率 $q_{iu}$
1	O	0.002
	A	0.05
	A	0.10
	A	0.18
2	O	0.004
	A	0.02
	A	0.15
	A	0.12
3	O	0.005
	A	0.05
	A	0.20
	A	0.10
4	O	0.003
	A	0.01
	A	0.08
	A	0.10
5	O	0.002
	A	0.02
	A	0.10
	A	0.07
6	O	0.002
	A	0.02
	A	0.10
	A	0.10

$$(\mu_1(\mathbf{m}) - 1)l_1 + (m_1 + 3)^2 + (m_2)^2 + (m_3)^2 + (m_4 + 3)^2 + (m_5)^2 + (m_6)^2 = 51$$

$$(\mu_2(\mathbf{m}) - 1)l_2 + 20 \sum_{i=1}^6 (m_i + \exp(-m_i)) = 260$$

$$(\mu_3(\mathbf{m}) - 1)l_3 + 20 \sum_{i=1}^6 (m_i \exp(-m_i/4)) = 140$$

$$\mathbf{m} = [m_1 \cdots m_6], m_i \geq 0 : \text{integer for } i = 1, 2, \dots, 6$$

ステップ 4 初期の染色体  $V_k$  をランダムに生成する.

$$V_1 = [1 \ 2 \ 3 \ 3 \ 1 \ 3]$$

$$V_2 = [3 \ 1 \ 2 \ 2 \ 1 \ 3]$$

$$\mathbf{V}_3 = [1 \ 2 \ 3 \ 3 \ 3 \ 2]$$

$$\mathbf{V}_4 = [2 \ 2 \ 3 \ 3 \ 1 \ 1]$$

$$\mathbf{V}_5 = [3 \ 1 \ 2 \ 2 \ 3 \ 2]$$

ステップ5  $gen = 2$  とおく. 式 (6.12) より各染色体の評価関数値を計算する.

$$eval(\mathbf{V}_1) = 0.271185$$

$$eval(\mathbf{V}_2) = 0.408735$$

$$eval(\mathbf{V}_3) = 0.416843$$

$$eval(\mathbf{V}_4) = 0.532714$$

$$eval(\mathbf{V}_5) = 0.555988$$

ステップ6 (6.1) 交叉: 生成された乱数は 0.089663, 0.215613, 0.315064, 0.526048 および 0.641652 である. このとき,  $r_p < p_c$  を満足する染色体は  $V_1$  および  $V_2$  である. ランダムに選ばれた交叉の位置が 4 のとき, 次の 2 つの染色体が新たに生成される.

$$\mathbf{V}'_1 = [1 \ 2 \ 3 \ 3 \ 1 \ 3]$$

$$\mathbf{V}'_2 = [3 \ 1 \ 2 \ 2 \ 1 \ 3]$$

(6.2) 突然変異: 生成された乱数 ( $r_p < p_m$  を満足するものを表示) および対応する遺伝子を次に示す.

ビット位置	遺伝子番号	ビット番号	乱数
16	3	4	0.113498
18	3	6	0.038301
29	5	5	0.196936
39	7	3	0.034211
41	7	5	0.154180
42	7	6	0.134129

突然変異によって生成された染色体は, 次の 3 つである.

$$\mathbf{V}'_3 = [1 \ 2 \ 3 \ 2 \ 3 \ 1]$$

$$\mathbf{V}'_4 = [3 \ 1 \ 2 \ 2 \ 2 \ 2]$$

$$\mathbf{V}'_5 = [3 \ 1 \ 3 \ 2 \ 3 \ 1]$$

(6.3) 選択: 生成された各染色体の評価関数値を求める.

$$eval(\mathbf{V}'_1) = 0.271185$$

$$eval(\mathbf{V}'_2) = 0.408735$$

$$eval(\mathbf{V}'_3) = 0.227088$$

$$eval(\mathbf{V}'_4) = 0.565843$$

$$eval(\mathbf{V}'_5) = 0.375953$$

親の染色体と新しく生成された染色体の中から適合度の高い順に選ぶことによって, 次のような次世代の染色体がつくられる.

$$\mathbf{V}_1 = [3 \ 1 \ 2 \ 2 \ 2 \ 2]$$

$$\mathbf{V}_2 = [3 \ 1 \ 2 \ 2 \ 1 \ 3]$$

$$\mathbf{V}_3 = [1 \ 2 \ 3 \ 3 \ 3 \ 2]$$

$$\mathbf{V}_4 = [2 \ 2 \ 3 \ 3 \ 1 \ 1]$$

$$\mathbf{V}_5 = [3 \ 1 \ 2 \ 2 \ 3 \ 2]$$

ステップ7 式(6.14)から  $\mathbf{V}^*$  を計算する.

$$\mathbf{V}^* = \mathbf{V}_1,$$

$$maxeval = \max\{eval(\mathbf{V}_1)\} = 0.565843$$

ステップ8  $gen < maxgen$  を満たすので, ステップ5に戻る.

300世代まで繰り返した後, 得られた評価関数値の値にシステム設計者が満足したとする. このとき, システム信頼度の最大値は0.928987である. また, 選ばれた最良の遺伝子は, 次のとおりである.

$$\mathbf{V}^* = [3 \ 3 \ 3 \ 2 \ 2 \ 2]$$

信頼度関数値は0.928987, また各サブシステムに対する冗長ユニット数はそれぞれ  $m_1 = 3, m_2 = 3, m_3 = 3, m_4 = 2, m_5 = 2$  および  $m_6 = 2$  となる. また, このときの各メンバシップ関数値はそれぞれ  $\mu_0(\mathbf{m}) = 0.986, \mu_1(\mathbf{m}) = 1.0, \mu_2(\mathbf{m}) = 1.0, \mu_3(\mathbf{m}) = 1.0$  となっている. 各世代に対して評価関数値が改善される過程を図6.4に示す.

図6.4から分かるように, 本数値例の場合, ファジィ目標およびファジィ制約を導入しない場合のシステム信頼度が0.911523であるのに対し, これらを導入することにより0.928987と高くなっている. このとき信頼度は, システム設計者が与えた

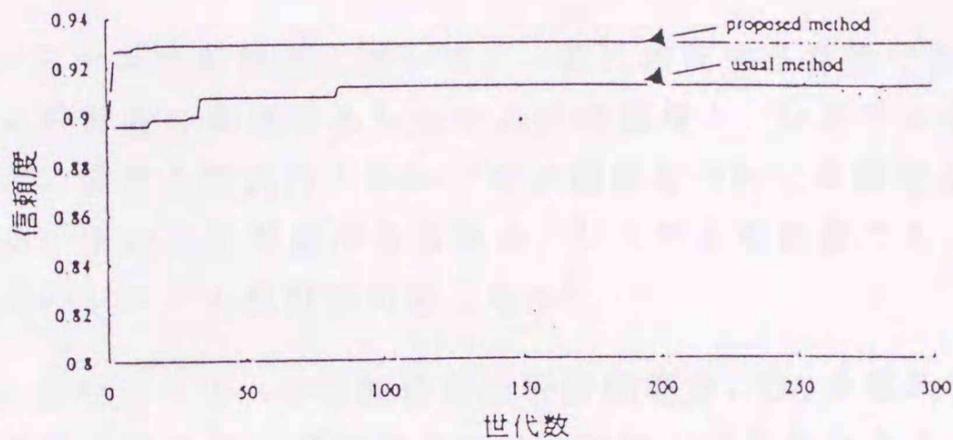


図 6.1. 評価関数値の改善過程

目標値 0.93 に対してやや低い。これは、システム信頼度に対する目標値を完全に達成する解は得られなかったものの（信頼度関数に対するメンバシップ関数値が  $\mu_0(m) = 0.986$ ）、目標値の達成度を多少犠牲にして制約条件を緩めることによって、従来 of 解法では得ることのできなかつた高い信頼度を得たことを示している。

このように、提案する解法は、システム設計段階で数種類の故障モードを考慮に入れることができるほか、ファジィ目標およびファジィ制約を導入することによって、システム設計者の意向を反映できるとともに、そのあいまいさを取り扱うことが可能となっている。

## 6.4 むすび

本章では、数種類の故障モードを伴うとともにファジィ目標とファジィ制約をもつシステム信頼性最適設計問題を、GAを導入することによって解く解法を提案した。さらに、提案解法は次に示すような特徴を持つことを示した。

- 数種類の故障モードを考慮することによって、直列／並列のユニットが複雑に接続されて構成される現実のシステム信頼性の設計問題を系統的に解析することができるため、よりきめ細かい現実のシステム信頼性の最適設計問題に対応することが可能となった。
- また、ファジィ目標およびファジィ制約を導入することによって、システム設計者の主観的判断のあいまいさやシステムのおかれた環境の変化に対応した制

約条件の右辺定数の変動を考慮することができるため、柔軟で現実的な設計が可能となった。

- また、システム設計段階において、一般に相反する目標であると考えられるシステム設計者の要求するシステムの信頼度と、システムの重量、容積、コストなどに関する制約のトレードオフ関係を同時に考慮できるとともに、それらのあいまいさを考慮できるため、システム設計者にとってより柔軟で満足度の高いシステム設計が可能となった。
- さらに、このシステムの信頼性最適設計問題を、GAを導入することによって非線形のまま扱うことができるため、取扱いが容易なうえ、コンピュータメモリの点でも有利であるとともに、SGAを導入したことによって最良解を比較的簡単なアルゴリズムによって得ることが可能となった。



## 第7章

# ハイブリッド型遺伝的アルゴリズムによるシステム信頼性の最適設計

### 7.1 まえがき

一般に、システム信頼性の最適設計を行うにあたり、予算、重量、容積等資源の制約を満足する必要がある。こうした複数の制約を個々に満たす代わりに、すべての資源を金額に換算することにより一元化し、限られた予算の中で資源の最適配分を行う、De Novo 計画問題が Zeleny によって提案されている。この De Novo 計画問題は、従来の数理計画法が限られた資源の制約の下での最適化を目的としているのに対し、与えられた予算額の中で資源の最適配分を含む計画問題の最適な設計を行うことができるという点で優れている。すなわち、De Novo 計画問題は、与えられたシステムの最適化というよりも、より柔軟なシステムの最適設計を可能としている。この De Novo 計画問題は、離散的な変数を伴う、いわゆる整数計画問題については、ナップサック問題に変換することができる。また、システム信頼性やシステムアベイラビリティの最適配分に関する問題は、一般に非線形整数計画問題として定式化され、これは 0-1 計画問題に変換することができる。このとき、GUB (Generalized Upper Bounding) あるいは多重選択として知られる特有な構造をもつ制約が付加される。こうした問題は一般に大規模な問題となり、従来の数理計画法では効率上実用的ではないという問題があったが、Gen らによって GUB 構造特有の性質を利用した効率的な解法が提案されている[13, 14, 16, 17, 94, 100]。

他方、GA は、組合せ最適化問題などに対する有効な手法として、近年、数多くの研究成果が報告されている。GUB 構造を伴うナップサック問題を GA で解くための解法としては、Hadjalouane *et al.* の研究など[27, 29, 131]があるが、GUB 構造性を生かした GA ではない。

GUB構造を伴うナップサック問題は、簡単な構造にもかかわらず、予算編成問題、献立問題や製品価格問題など多方面に応用され、組合せ最適化問題の中で重要な位置を占めており、解法を研究することは意義がある。また、この問題はNP-困難であることが示されていて、分枝限定法や動的計画法などの最適解法よりも近似解法の研究が進められており、ラグランジュ緩和問題を適用した解法や連続緩和問題へ変換した解法などが提案されている[3, 65, 111, 134]。さらに、GAによってナップサック問題を解く解法は上述の Hadjaolune *et al.*のほか Olsen, Hinterding, 坂和らによって提案されているが、GUB構造を考慮した解法は提案されていない。

本章では、GUB構造を伴うナップサック問題として定式化される大規模なシステム信頼性の最適設計問題を解くために、GUB構造の特徴を生かした新しいGAによるハイブリッド解法を提案する。提案する解法は、次に示す特徴をもつ。

- GUB構造を有効に表現する新しい遺伝子表現を導入することにより、各種の遺伝的操作に対してGUB構造性が保持されるため、取扱いが容易である。さらに、0-1変数によるGUB構造の表現を行った場合に比べ、必要となる遺伝子数のはるかに少なくて済むため、計算効率および必要メモリ量の点で有利である。
- 各GUB制約の決定変数を効率指標に基づいてランク付けし、それに基づいて解の改善をはかる過程をGAに組み込む、すなわちハイブリッド化により、GUB構造性を利用した効率的な解の探索が可能になっている。

提案する解法の有効性を示すために、数値例として大規模なシステム信頼性の最適設計問題を取り上げる。

## 7.2 システム信頼性設計問題

### 7.2.1 GUB構造を伴う混合0-1線形計画問題

システム信頼性やシステムアベイラビリティの最適化に関する問題は、一般に非線形整数計画問題として定式化され、これは0-1線形計画問題に変換することができる。このとき、GUBと呼ばれる特殊な構造もつ制約条件が付加される。本章では、De Novo計画問題の概念に基づいてこのGUB制約を伴う0-1線形計画問題の各利用可能資源量（右辺定数）を単価で同定し、それらの総和が当初の予算額（本予算）と新たに追加された補正予算額によって制約される制約条件を伴う問題を取り扱う。一般に、システムの最適設計を行うにあたり、設計段階でシステム資源の制約には、多少なりとも調整の余地がある場合が多いと考えられる。ところが、従来の数理計画法では、資源の制約を厳密に与える必要があることから、

ある資源の制約のためにシステム全体の最適解が低く抑えられるという問題があった。提案解法では、必ず必要とされるシステム資源のほかに、補正予算として資源を適切に追加配分することにより、より最適なシステム設計を可能としている。この問題は、次のような  $m$  個のシステム制約と、 $n$  個の GUB 制約をもち 0-1 変数  $x_{ij}$  と非負実数変数  $\Delta b_r$  を決定変数とする混合 0-1 線形計画問題として定式化される。

0-1LP/GUB:

$$\max/\min f(x) = \sum_{i=1}^n \sum_{t=1}^{n_i} c_{ij} x_{ij} \quad (7.1)$$

$$\text{s.t.} \quad \sum_{i=1}^n \sum_{j=1}^{n_i} a_{rij} x_{ij} \leq b_r + \Delta b_r, r = 1, \dots, T \quad (7.2)$$

$$\sum_{r=1}^T \gamma_r \Delta b_r \leq \Delta B \quad (7.3)$$

$$\sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, \dots, n \quad (7.4)$$

$$x_{ij} = 0 \text{ or } 1, \quad \forall i, j \quad (7.5)$$

$$\Delta b_r \geq 0, \quad r = 1, \dots, T \quad (7.6)$$

ここで、式 (7.2) をシステム制約、式 (7.4) を GUB 制約と呼び、各係数と変数は次のとおりである。

$c_{ij}$ :  $i$  番目の GUB 制約の  $j$  番目の決定変数に対応する目的関数の係数

$a_{rij}$ : 制約条件  $r$  の  $i$  番目の GUB 制約における  $j$  番目の決定変数に対応するシステム制約の係数

$b_r$ :  $r$  番目のシステム制約の当初の資源量 (定数)

$\Delta b_r$ :  $r$  番目のシステム制約の補正資源量 (変数)

$\gamma_r$ : 資源  $r$  の単位量あたりの単価

$\Delta B$ : 補正予算額 (定数)

$x_{ij}$ :  $i$  番目の GUB 制約の  $j$  番目の 0-1 決定変数

$n_i$ :  $i$  番目の GUB 制約に含まれる決定変数の数

また、本予算額  $B$  (定数) は、次式で与えられる。

$$B = \sum_{r=1}^T \gamma_r b_r \quad (7.7)$$

## 7.2.2 GUB構造を伴うナップサック問題への変換

前節のGUB構造を伴う混合0-1線形計画問題において、式(7.2)と式(7.3)の定数 $b_r$ および変数 $\Delta b_r$ , ( $r = 1, 2, \dots, T$ )を次のように消去し、GUB構造を伴う(多重選択)ナップサック問題に変換する。

まず、式(7.2)の両辺を $\gamma_r (> 0)$ 倍し、 $r = 1, 2, \dots, T$ まで加えて

$$\sum_{r=1}^T \gamma_r \left( \sum_{i=1}^n \sum_{j=1}^{n_i} a_{rij} x_{ij} \right) \leq \sum_{r=1}^T \gamma_r (b_r + \Delta b_r) \quad (7.8)$$

とし、式(7.3), (7.7)より

$$\sum_{i=1}^n \sum_{j=1}^{n_i} \left( \sum_{r=1}^T \gamma_r a_{rij} \right) x_{ij} \leq B + \Delta B \quad (7.9)$$

を得る。ここで、

$$A_{ij} = \sum_{r=1}^T \gamma_r a_{rij}, \quad \forall i, j \quad (7.10)$$

とおくと、次のようなGUB構造を伴う(多重選択)ナップサック問題に変換される。  
0-1KP/GUB:

$$\max/\min \quad f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} c_{ij} x_{ij} \quad (7.11)$$

$$\text{s. t.} \quad G(\mathbf{x}) \equiv \sum_{i=1}^n \sum_{j=1}^{n_i} A_{ij} x_{ij} \leq B + \Delta B \quad (7.12)$$

$$g_i(\mathbf{x}) \equiv \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (7.13)$$

$$x_{ij} = 0 \text{ or } 1, \quad \forall i, j \quad (7.14)$$

この問題は、次のように解釈できる。すなわち、各システム制約の右辺の資源量を予算化し、それらの総和を本予算 $B$ で制限する。さらに、補正予算 $\Delta B$ が組まれた場合に、各システム制約に対する追加資源量 $\Delta b_r$ を最適に配分する。各GUB制約から排他的に一個の決定変数を1とする制約の下で、目的関数 $f(\mathbf{x})$ を最大化または最小化する問題である。また、各システム制約は、式(7.6)の非負条件から、当初の各資源量 $b_r$ は必ず割り当てられていることが保証されているという点でDe Novo計画問題の拡張となっている。従って、あるシステム制約が極端に小さな値で制約されるという問題は解消される。

ここで、GUB構造を伴う混合0-1線形計画問題(0-1LP/GUB)を、変換されたGUB構造を伴うナップサック問題(0-1KP/GUB)に変換する際、式(7.3)と非負条件式(7.6)が隠蔽されているために、0-1KP/GUBから得られた実行可能解から、次式

$$\Delta b_r = \max\left\{\sum_{i=1}^n \sum_{j=1}^{n_i} a_{rij} x_{ij} - b_r, 0\right\}, \quad (7.15)$$

$$r = 1, 2, \dots, T$$

により各 $\Delta b_r$ の値を求める<sup>[66]</sup>。また、設計案で実際に使用する補正予算額 $\Delta B^*$ は、式(7.15)から最適な補正資源量 $\Delta b_r^*$ , ( $r = 1, 2, \dots, T$ )が決定した後、次式により計算される。

$$\Delta B^* = \sum_{r=1}^T \gamma_r \Delta b_r^* \quad (7.16)$$

ここで、 $\Delta b_r^*$ , ( $r = 1, 2, \dots, T$ )は0-1LP/GUBの制約式(7.2)~(7.6)を満たす範囲で決定されることから、 $\Delta B^* \leq \Delta B$ , すなわち使用する補正予算額 $\Delta B^*$ は、与えられた補正予算額 $\Delta B$ の範囲内で決定される。

## 7.3 ハイブリッド型遺伝的アルゴリズム

### 7.3.1 遺伝子表現

本章では、GUB構造型を生かした、新しい遺伝子表現を導入する。導入する染色体は、一つのGUB制約に関する情報を一つの遺伝子として表現する。これにより、各GUB構造において選ばれる決定変数の位置および一つのGUB制約に含まれる決定変数の数を一つの遺伝子として表すことにより、簡単な表現にも関わらず、GUB制約の構造型を失うことなく遺伝的操作が可能である点に大きな特徴がある。従って、従来から提案されている各種の遺伝的操作の中で、対象とする問題に適した操作を選んで適用することが可能である。

$s_j$ は $j$ 番目のGUB制約において、値1をとる決定変数の位置を表し、 $1 \leq s_j \leq q_j$ ,  $\forall j$ の範囲で定まる。

この遺伝子表現から、 $\mathbf{V}_p$ を同一集団内における $p$ 番目の染色体を表すものとする、次のように表される。

$$\mathbf{V}_p = [s_{p1} \ s_{p2} \ \dots \ s_{pn}]$$

また、提案する遺伝子表現の代わりに、取り扱う問題の0-1変数を使用して遺伝子表現を行うことも可能であるが、その場合一つの染色体を表現するために必要な遺伝子が少なくとも $\sum_{j=1}^n q_j$ 個を必要とすることになる。その点、提案する遺伝子表現では $n$ 個で済むため、遺伝的操作を行う場合に有利であるうえ、特に大規模

な問題に対しては計算時間および必要メモリ量の面でも有利である。

```

procedure HGA(0-1KP/GUB)

begin
  sort decision variables by ranking based on
  efficiency index;
  t ← 0;
  initialize P(t);
  evaluate P(t);
  while (not terminal condition) do
  begin
    recombine P(t) to generate C(t);
    evaluate C(t);
    select P(t+1) from P(t) and C(t);
    exchange decision variables based on ranking for
    finding more efficient solution;
    t ← t+1;
  end
end

```

図7.1 提案する解法のアルゴリズム

ここで、

$P(t)$  : 第  $t$  世代の親染色体

$C(t)$  : 第  $t$  世代の子染色体

を表す。

### 7.3.2 評価関数

ナップサック制約の充足度をペナルティとして表現した、次のような評価関数を導入する。この関数は、予算の不足達成に対応したペナルティを与えている。

$$eval(\mathbf{V}_p) = h(\mathbf{V}_p) * f(\mathbf{V}_p) \quad (7.17)$$

ここで、ペナルティ関数  $h(\mathbf{V}_p)$  は次式で示される。

$$h(\mathbf{V}_p) = \begin{cases} G(\mathbf{V}_p)/(B + \Delta B); & G(\mathbf{V}_p) - (B + \Delta B) < 0 \\ & \text{かつ式 (7.3) が成立するとき} \\ 0 & ; \text{そうでないとき} \end{cases} \quad (7.18)$$

$p = 1, \dots, \text{pop-size}$

### 7.3.3 遺伝的操作

本章で適用する遺伝的操作は次のとおりである。既に述べたとおり、提案する解法は GUB 構造的性を有効に表現する遺伝子表現が可能であるため、問題に適した遺伝的操作を適用することが可能である。本章では、提案する遺伝子表現および計算法の有効性を明確にするために、遺伝的操作として従来から知られる一般的な操作を適用するものとする。

**交叉:** 交叉は、二つの親染色体間で一つの交叉位置でそれぞれの部分解を交換する一点交叉を用いる。

**突然変異:** ランダムに選ばれた遺伝子座において、 $1 \leq s_j \leq q_j$  を満たす遺伝子に置き換える。

**選択:** 親染色体および交叉や突然変異によって生成される子染色体からルーレット選択により次世代の個体を選択する。

### 7.3.4 決定変数のランクに基づく並換え

GUB 制約の構造的性を利用し、限られた予算の下での最大化問題を解くための効率的な解の探索を行うために、GA を適用する前に、各 GUB 制約の決定変数に対して次のようなランク付けに基づく並換えを行う。

**ステップ 1:** 同一集団内において、各 GUB 制約の決定変数をランク付けをするために、次式に基づいて指標を計算する。

$$e_{ij} = c_{ij} / A_{ij}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n_i \quad (7.19)$$

**ステップ 2:** 次式により、各 GUB 制約において指標の高い順に決定変数をランク付けする。

$$\bar{j}_i = [\bar{j}_{i1} \bar{j}_{i2} \dots \bar{j}_{in_i}] = \text{indsort}_i \{e_{ij} | j = 1, 2, \dots, n_i\}, \quad \forall i \quad (7.20)$$

ここで、indsort は index sort の省略形で、降順の並べ換えを表す。

ステップ3: 各GUB制約において, ランクの高い順に決定変数を並換える.

式(7.19)で示される指標は, 決定変数 $x_{ij}$ が選ばれた場合の予算制約の係数(投入コスト: $A_{ij}$ )に対する目的関数の係数(利得: $c_{ij}$ )の比を表し, 効率指標と呼ぶ. 効率指標は, 変数 $x_{ij}$ のコストパフォーマンスと解釈することができる.

## 7.4 計算法

本章で提案するGUB構造化を利用した効率的なGAに基づく計算法は次のとおりである. 本計算法は, ステップ2およびステップ6において, GUB構造化の特徴を利用して, 解空間においてより効率指標の高い解を探索するよう方向付けするヒューリスティクスを組み込んだハイブリッド化を行うことにより, 効率的に最良解を得ることを可能としている. すなわち, 一般に広域探索を得意とするGAに, 効率的に局所探索を行う手続きを組み込んでいる.

ステップ1: パラメータの設定

$max\_gen, pop\_size, p_c, p_m$ をそれぞれ設定する. 世代数を $t=0$ とおく.

ステップ2: ランクに基づく並換え

各GUB制約の決定変数を効率指標に基づいてランクの高い順に並べ換える.

ステップ3: 初期染色体集団の生成

解の多様性を維持しながら, より効率性の高い解を探索するため,  $n$ 要素からなる初期染色体 $V_p, (p=1, 2, \dots, pop\_size)$ の集団をランク-1の近傍でランダムに生成する

ステップ4: 評価関数の計算

評価関数値 $eval(V_p), (p=1, 2, \dots, pop\_size)$ を, 式(7.17)に基づいて計算する.

ステップ5: 遺伝的操作

以下の三つの遺伝的操作を行う.

(5.1) 交叉

乱数 $r_p, (p=1, 2, \dots, pop\_size)$ を発生させ,  $r_p < p_c$ となる染色体を, 前節の交叉法によって交叉させる. 交叉によって生成される染色体の数を $ccnt$ とする.

(5.2) 突然変異

乱数 $r_p, (p=1, 2, \dots, pop\_size + ccnt)$ を発生させ,  $r_p < p_m$ となる染色体を, 前節の突然変異法によって突然変異を行う. 突然変異によって生成される染色体の数を $mcnt$ とする.

## (5.3) 選択

親染色体と生成された子染色体の評価関数値を計算し,  $pop\_size + cnt + mcnt$  個の染色体の中から  $pop\_size$  個の染色体を前節のルーレット選択法によって選択する.

## ステップ6: 各染色体に対する解の改善

各染色体について, 実行可能性を調べる. 実行可能な染色体の場合, 各 GUB 制約の変数を一ランク高い決定変数, 一ランク低い決定変数にそれぞれ入れ換えた染色体  $V_p^{(1)}, V_p^{(2)}, \dots, V_p^{(2n)}$ , ( $p=1, 2, \dots, pop\_size$ ) および元の染色体の中から最大の目的関数値をもつ染色体を探し, 元の染色体と置き換える. 実行不可能な染色体の場合, 各 GUB 制約の変数を一ランク高い決定変数, 一ランク低い決定変数にそれぞれ入れ換えた染色体および元の染色体の中から最大のコストを持つ染色体を探し, 元の染色体と置き換える.

## ステップ7: 終了条件の判定

$t < max\_gen$  であれば,  $t = t + 1$  としてステップ5に戻る.  $t = max\_gen$  ならば, そのときの最大の評価関数値をもつ染色体で表現される解を最良解として終了する. 式(7.15)により補正される各システム資源量  $\Delta b_r^*$ , ( $r = 1, 2, \dots, T$ ) を計算するとともに, 設計案で使用する補正予算額  $\Delta B^*$  を式(7.16)から得る.

## 7.5 数値例

数値例として, Fyffeらの取り扱ったシステム信頼性の最適なユニット選択と冗長配分の問題を取り上げる. この問題は, 表7.1に示すような, 2つの制約条件の下で14のサブシステムからなるシステムの信頼度を最大にする設計案の選択とユニットの配分を求める次のような最適設計問題である. また, 2つの資源の単価をそれぞれ  $p_1=4, p_2=6$ , 補正予算額  $\Delta B$  を75とする.

$$\max \quad R(\mathbf{m}, \alpha) = \prod_{i=1}^{14} \{1 - (1 - R_{i\alpha_i})^{m_i}\} \quad (7.21)$$

$$\begin{aligned} \text{s. t.} \quad & \sum_{i=1}^{14} v_{1i\alpha_i} m_i \leq 130 + \Delta b_1 \\ & \sum_{i=1}^{14} v_{2i\alpha_j} m_i \leq 170 + \Delta b_2 \\ & 4\Delta b_1 + 6\Delta b_2 \leq 75 \\ & 1 \leq m_i \leq u_i, \quad i = 1, 2, \dots, 14 \end{aligned} \quad (7.22)$$

$$1 \leq \alpha_i \leq \beta_i$$

$$\Delta b_1, \Delta b_2 \geq 0$$

$$m_i, \alpha_i \geq 0: \text{integer}, \quad i = 1, 2, \dots, 14$$

ここで、 $R_{j\alpha_i}$ はサブシステム*i*において設計案 $\alpha_i$ を選択した場合の信頼度、 $v_{i\alpha_i}$ はサブシステム*i*で設計案 $\alpha_i$ を選択したときの $\alpha_i$ 番目のシステム制約の係数である。また、 $m_i$ は、サブシステム*i*に対するユニットの冗長配分数、 $\beta_i$ はサブシステム*i*に対する最大設計案数、さらに $u_i$ はサブシステム*i*に対する最大冗長配分数を表す。ここでは、すべてのサブシステムに対して $u_i = 5$ , ( $i = 1, 2, \dots, 14$ )とする。

この問題に、各サブシステムそれぞれに対して選択と冗長配分を決定する場合に1、そうでない場合には0をとる0-1変数を導入する。

$$x_{ij'} = \begin{cases} 1; & \text{サブシステム } i \text{ で, } j \text{ 番目の設計案} \\ & \text{及びユニットを選択するとき} \\ 0; & \text{そうでないとき} \end{cases} \quad (7.23)$$

ここで、添え字  $j'$  は、式(7.24)に示すように  $i$  番目のサブシステムの設計案  $\alpha_i$  と、そのときの冗長配分数  $m_i$  を同時に表現するもので、それぞれ設計案ごとに冗長配分数を決定することを表す。

$$j' = u_i(\alpha_i - 1) + m_i, \quad 1 \leq j' \leq n_i \quad (7.24)$$

例えば、サブシステム4において2番目の設計案を選択し、冗長配分数を3とすると、 $j' = 5 \cdot (2-1) + 3 = 8$  となり、 $x_{48} = 1$ 、それ以外の  $x_{4j'} = 0$ ,  $j' = 1, \dots, 15$  ( $j' \neq 8$ ) となる。また、 $n_i$  は、次式により決定される。

$$n_i = \beta_i \cdot u_i, \quad i = 1, 2, \dots, 14 \quad (7.25)$$

この0-1変数を導入するとともに、柔軟なシステム信頼性の最適設計を行うために、2つのシステム制約を総予算（本予算および補正予算）で総合的に制約するDe Novo計画問題を発展させた問題を適用すると、次に示すような14個のGUB制約を伴うナップサック問題に変換される。ここで、本予算  $B$  は式(7.7)より1540である。

$$\begin{aligned} \max \quad f(\mathbf{x}) &= \sum_{i=1}^{14} \sum_{j=1}^{n_i} c_{ij} x_{ij} \\ \text{s. t.} \quad G(\mathbf{x}) &\equiv \sum_{i=1}^{14} \sum_{j=1}^{n_i} A_{ij} x_{ij} \leq 1540 + \Delta B \end{aligned} \quad (7.26)$$

$$g_i(\mathbf{x}) \equiv \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, 14 \quad (7.27)$$

$$x_{ij} = 0 \text{ or } 1, \quad \forall i, j$$

ここで、目的関数の係数  $c_{ij}$ ,  $\forall i, j$  は、式 (7.21) に示す問題の目的関数の係数  $R_{i\alpha_i}$ ,  $\forall i, \alpha_i$  のそれぞれ自然対数をとることにより線形変換している。

表 7.1 システム信頼性の最適ユニット選択と冗長配分の数値例

サブシステム ( $j$ )	設計案 ( $\alpha_j$ )											
	1			2			3			4		
	$R_{ij\alpha_i}$	$v_{1i\alpha_i}$	$v_{2i\alpha_i}$	$R_{i\alpha_i}$	$v_{1i\alpha_i}$	$v_{2i\alpha_i}$	$R_{i\alpha_i}$	$v_{1i\alpha_i}$	$v_{2i\alpha_i}$	$R_{i\alpha_i}$	$v_{1i\alpha_i}$	$v_{2i\alpha_i}$
1	0.90	1	3	0.93	1	4	0.91	2	2	0.95	2	5
2	0.95	2	8	0.94	1	1	0.93	1	9	-	-	-
3	0.85	2	7	0.90	3	5	0.87	1	6	0.92	4	4
4	0.83	3	5	0.87	4	6	0.85	5	4	-	-	-
5	0.94	2	4	0.93	2	3	0.95	3	5	-	-	-
6	0.99	3	5	0.98	3	4	0.97	2	5	0.96	2	4
7	0.91	4	7	0.92	4	8	0.94	5	9	-	-	-
8	0.81	3	4	0.90	5	7	0.91	6	6	-	-	-
9	0.97	2	8	0.99	3	9	0.96	4	7	0.91	3	8
10	0.83	4	6	0.85	4	5	0.90	5	6	-	-	-
11	0.94	3	5	0.95	4	6	0.96	5	6	-	-	-
12	0.79	2	4	0.82	3	5	0.85	4	6	0.90	5	7
13	0.98	2	5	0.99	3	5	0.97	2	6	-	-	-
14	0.90	4	6	0.92	4	7	0.95	5	6	0.99	6	9

-:設計案なし

この問題を解くために、本章で提案する遺伝子表現を伴う GA を適用する。この問題の特徴である GUB 構造的性を考慮した遺伝子表現として、GUB 構造をもつ 14 個のサブシステムそれぞれに対して一個の遺伝子  $s_i$ , ( $i = 1, 2, \dots, 14$ ) を割り当てる。

ステップ 1: 最大世代数  $max\_gen = 5000$ , 集団サイズ  $pop\_size = 100$ , 交叉確率  $p_c = 0.4$ , および突然変異確率を  $p_m = 0.6$  と設定する。さらに、世代数を  $t = 0$  とおく。

ステップ 2 各 GUB 制約の変数を効率指標に基づいてランク付けする。表 7.2 に、一番目の GUB 制約の変数をランク付けした例を示す。

表7.2 一番目のGUB制約の効率指標に基づくランク付け

$\bar{j}_1$	目的関数 $c_{1\bar{j}_1}$	制約条件 $A_{1\bar{j}_1}$	効率指標 $e_{1\bar{j}_1}$	$k$
1	0.89464	17	0.0526259	1
2	0.90569	18	0.0503161	11
3	0.92743	21	0.0441633	6
4	0.94871	30	0.0316237	16
5	0.98995	34	0.0291162	2
6	0.99187	36	0.0275519	12
7	0.99509	42	0.0236926	7
8	0.99900	51	0.0195882	3
9	0.99927	54	0.0185050	13
10	0.99750	60	0.0166250	17
11	0.99966	63	0.0158676	8
12	0.99999	68	0.0147044	4
13	0.99993	72	0.0138879	14
14	0.99998	84	0.0119045	9
15	0.99999	85	0.0117646	5
16	1.00000	90	0.0111111	15
17	0.99987	90	0.0111097	18
18	1.00000	105	0.0095238	10
19	0.99999	120	0.0083333	19
20	1.00000	150	0.0066667	20

ステップ3: 生成された初期染色体は次のとおりである.

$$V_1 = [6 \ 3 \ 1 \ 8 \ 13 \ 1 \ 3 \ 5 \ 12 \ 12 \ 7 \ 1 \ 2 \ 8]$$

$$V_2 = [12 \ 12 \ 1 \ 2 \ 9 \ 8 \ 5 \ 3 \ 7 \ 13 \ 1 \ 20 \ 6 \ 9]$$

...

$$V_{100} = [1 \ 1 \ 1 \ 1 \ 5 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

ステップ4: 初期染色体に対する評価関数値は次のとおりである.

$$eval(V_1) = 0.4112$$

$$eval(V_2) = 0.5326$$

...

$$eval(V_{100}) = 0.8901$$

ステップ5: (5.1) 交叉

次の60個の染色体が新たに生成される.

$$\mathbf{V}'_1 = [19 \ 12 \ 7 \ 5 \ 7 \ 1 \ 10 \ 3 \ 11 \ 13 \ 1 \ 20 \ 4 \ 9]$$

$$\mathbf{V}'_2 = [12 \ 4 \ 1 \ 2 \ 9 \ 8 \ 5 \ 4 \ 7 \ 2 \ 2 \ 4 \ 6 \ 2]$$

$$\mathbf{V}'_{60} = [7 \ 4 \ 5 \ 8 \ 13 \ 7 \ 7 \ 10 \ 12 \ 12 \ 8 \ 1 \ 6 \ 8]$$

(5.2) 突然変異

次の40個の染色体が生成される.

$$\mathbf{V}'_{61} = [7 \ 4 \ 5 \ 8 \ 9 \ 7 \ 7 \ 10 \ 19 \ 14 \ 8 \ 2 \ 6 \ 7]$$

$$\mathbf{V}'_{62} = [11 \ 1 \ 12 \ 4 \ 12 \ 1 \ 9 \ 14 \ 5 \ 8 \ 4 \ 4 \ 10 \ 12]$$

...

$$\mathbf{V}'_{100} = [1 \ 1 \ 1 \ 1 \ 1 \ 4 \ 1 \ 1 \ 1 \ 11 \ 1 \ 1 \ 1 \ 1]$$

(5.3) 選択

新しく生成された染色体の評価関数値を求める.

$$\text{eval}(\mathbf{V}'_1) = 0.7697$$

$$\text{eval}(\mathbf{V}'_2) = 0.3142$$

...

$$\text{eval}(\mathbf{V}'_{100}) = 0.8930$$

親の染色体と新しく生成した染色体の中からルーレット選択することにより、次のような染色体が選択される.

$$\mathbf{V}_1 = [2 \ 1 \ 8 \ 1 \ 14 \ 16 \ 2 \ 2 \ 19 \ 14 \ 14 \ 1 \ 5 \ 10]$$

$$\mathbf{V}_2 = [6 \ 3 \ 1 \ 8 \ 13 \ 1 \ 3 \ 5 \ 12 \ 12 \ 7 \ 1 \ 2 \ 8]$$

...

$$\mathbf{V}_{100} = [1 \ 1 \ 1 \ 1 \ 1 \ 4 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

ステップ6: 選択された各染色体について、解の改善のために各GUB制約に対する実行可能性を調べる. 最初に、 $\mathbf{V}_1$ の必要コストは1463.0であり、実行可能であるから、各GUB制約の変数を一ランク上げた染色体と一ランク下げた染色体を生成する.

$$\mathbf{V}_1 = [2 \ 1 \ 8 \ 1 \ 14 \ 16 \ 2 \ 2 \ 19 \ 14 \ 14 \ 1 \ 5 \ 10]$$

$$\mathbf{V}_1^1 = [1 \ 1 \ 8 \ 1 \ 14 \ 16 \ 2 \ 2 \ 19 \ 14 \ 14 \ 1 \ 5 \ 10]$$

$$\mathbf{V}_1^2 = [2 \ 1 \ 8 \ 1 \ 14 \ 16 \ 2 \ 2 \ 19 \ 14 \ 14 \ 1 \ 5 \ 10]$$

$$\mathbf{V}_1^3 = [2 \ 1 \ 7 \ 1 \ 14 \ 16 \ 2 \ 2 \ 19 \ 14 \ 14 \ 1 \ 5 \ 10]$$

...

$$\mathbf{V}_1^{28} = [2 \ 1 \ 8 \ 1 \ 14 \ 16 \ 2 \ 2 \ 19 \ 14 \ 14 \ 1 \ 5 \ 11]$$

これらの染色体の中で，最大のパフォーマンスを持つ  $V_1^1$  を選ぶ．残りの染色体  $V_p$ , ( $p=1,2,\dots, pop\_size$ ) についても同様の置き換えを行う．

ステップ7:  $t < max\_gen$  であるから， $t=1$  とおいてステップ5に戻る．

5000世代まで繰り返した後，得られた最良の染色体  $V^*$  は次のとおりである．

$$V^* = [13 \ 3 \ 18 \ 5 \ 8 \ 2 \ 8 \ 5 \ 2 \ 9 \ 3 \ 5 \ 7 \ 17]$$

また，このときの補正資源量は， $\Delta b_1^* = 0.0$  および  $\Delta b_2^* = 12.33$  である．さらに，この設計案で使用する補正予算額は， $\Delta B^* = 74.0 (\leq 75.0)$  である．

## 7.6 定量的評価

設計案の選択とユニット配分を伴う大規模なシステム信頼性の最適設計問題に対して，本章で提案した手法を適用した結果得られた最良の染色体は，上述のとおりである．このとき，得られた最良解は，信頼度が0.995110，使用予算額が1614.0であった．採用する設計案と配分数は表7.3に示すとおりである．

表7.3 採用する設計案と配分数

サブシステム $i$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
設計案 $\alpha_i$	3	2	2	3	1	3	2	1	3	3	3	4	1	2
配分数 $m_i$	4	1	5	2	3	1	3	3	1	3	1	1	4	5

提案解法における遺伝子表現の有効性を確かめるために，0-1変数を導入して遺伝子を表現した場合と，計算時間及び変数の数と必要メモリ量について比較する．

表7.4 変数の数と必要メモリ量の比較

手法	0-1変数による遺伝子表現の場合 HGA(0-1KP/GUB)	
計算時間	35.6(sec)	2.75(sec)
変数の数	$\sum_{i=1}^n q_i$ (240)	n (14)
必要メモリ量 ( <i>pop-size</i> = 100)	$\sum_{i=1}^n q_i * pop\_size$ (24000)	$n * pop\_size$ (1400)

注：計算時間は10回の平均値 (Gateway G6-200で実行). ()中の数字は数値例の場合.  
また、必要メモリ量は、一世代あたりの必要量を示す.

数値例の問題は、14個のサブシステムに対し3または4個の設計案と1～5個の冗長ユニット配分数を決定することになるため、240個の0-1決定変数を伴い、解の組合せ数が $20^6 \times 15^8$ 通りにのぼる、大規模な組合せ最適化問題である。この問題に対して、0-1変数による遺伝子表現を行った場合の計算時間を提案解法と比較すると、表7.4に示すとおり約12.97倍の処理時間を要した。また、遺伝子表現に必要な遺伝子の数は、表7.4に示すように、14個である。それに対し、0-1変数をそのまま使用する遺伝子表現を適用する場合、必要な遺伝子数は、一般に $\sum_{j=1}^n q_j$ 個であり、本数値例の場合、240個となる。これは一つの染色体での相違であり、本数値例のようにこの染色体を集団サイズを100、最大世代数を5000としてGAを適用した場合、0-1変数による遺伝子表現は、一世代あたり $240/14 * 100$ 倍すなわち約1714倍のメモリを必要とすることになり、提案解法の優位性が明らかである。さらに、遺伝的操作を5000回繰り返すことになるため、計算時間の点でも提案解法は有利である。この差は、問題の規模が大きくなるほど顕在化することから、提案解法は、一般に大規模となる現実の問題に対して有効である。また、0-1変数による遺伝子表現では、交叉および突然変異の遺伝的操作を行うと、GUB構造性が崩れてしまうため、そのまま遺伝的操作が適用できない。それに対し、提案解法における遺伝子表現では、遺伝的操作を行っても、常にGUB構造性が保持されるため、この点でも提案解法は有利である。

また、提案する解法(HGA(0-1KP/GUB))の有効性を確かめるために、各GUB制約に対する解のランク付け(前節の計算法:ステップ2)および解の改善(ステップ6)の過程を行わない、すなわちハイブリッド化を行わないGA(GA(0-1KP/GUB))を適用した場合との比較実験を行った結果を図7.2に示す(試行10回の平均)。

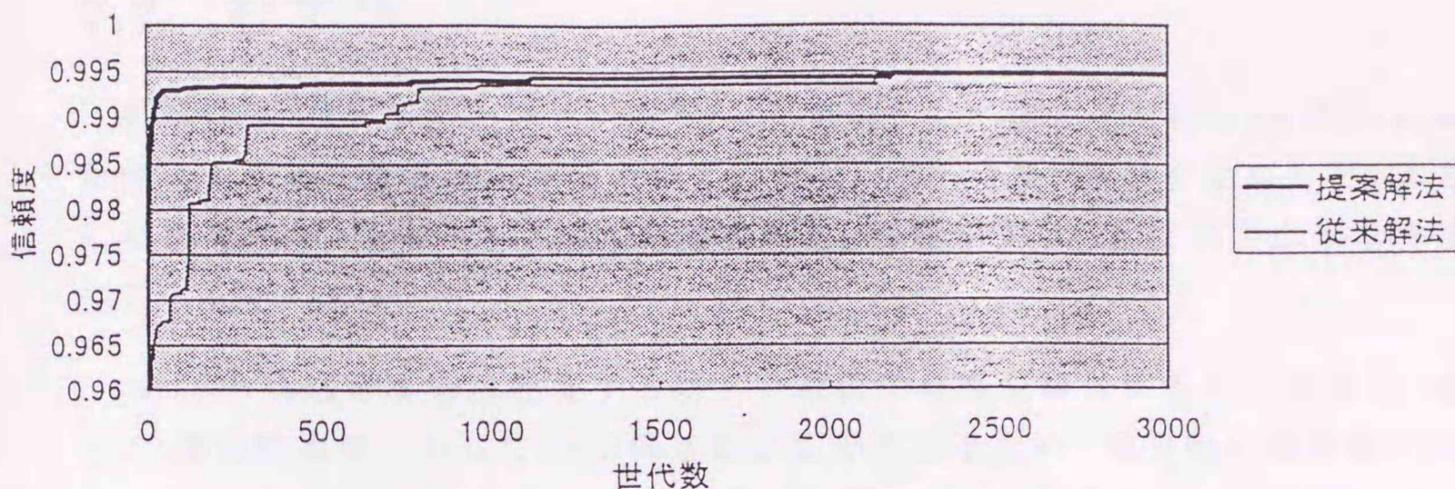


図 7.2 ハイブリッド化の効果

図 7.2 から分かるように，提案する解法は，従来解法に比べ最良解への収束が速いことが分かる．特に，初期の世代における改善効果が著しい．これは，提案する解法が，GUB 構造化を利用して効率指標の高い解の近傍を局所的に探索し，より解を改善する方向に探索を進めていくメカニズムを GA に組み込んだことにより，各世代において効果的に解の改善が行われていることを示すものである．

他方，本数値例で得られた最適設計案では，補正資源量を  $\Delta b_1 = 0.0$ ,  $\Delta b_2 = 12.33$  となった．これは，当初の補正予算額 (75.0) の範囲内 (74.0) で 2 番目のシステム資源に追加投資することにより，厳しかったこのシステム資源の制約を緩めていることを示していると考えられる．そこで，当初の問題の制約条件を予算制約化し，補正予算を配分することによる効果を確認するために，補正予算額 74.0 を 0-1LP/GUB 問題において二つのシステム資源に均等に割り振った場合，すなわち， $\Delta b_1 = 9.25$ ,  $\Delta b_2 = 6.17$  をそれぞれ与えて解を求めると，

$$[13 \ 2 \ 18 \ 13 \ 8 \ 7 \ 12 \ 4 \ 12 \ 8 \ 12 \ 4 \ 7 \ 12]$$

信頼度: 0.9764

$$g_1(x) = 125.0 < 130 + 9.25$$

$$g_2(x) = 176.0 < 170 + 6.17$$

を得る．システム制約  $g_2(x)$  のスラック量は 0.17 であり，この制約のために他のシステム制約に余裕があるにもかかわらず，システム全体が制約を受けていることが分かる．これは，提案解法が，各システム制約の利用可能資源量を単価で同定し，総予算によりシステム全体を制約しているため，最も制約の厳しいシステム制約に対し，より多くの資源を配分することが可能であるためである．

## 7.7 むすび

本章では, GUB 構造を伴うナップサック問題として定式化される De Novo 計画問題を伴う大規模な最適設計問題を解くために, GUB 構造の特徴を生かした GA を適用した効率的なハイブリッド解法を提案した. 提案解法は, 次に示すような特徴を持つことを示した.

- GUB 構造を有効に表現する新しい遺伝子表現を導入することにより, 各種の遺伝的操作に対して GUB 構造性が保持されるため, 取り扱いが容易になった. さらに, 0-1 変数による GUB 構造の表現を行った場合に比べ, 必要となる遺伝子数をはるかに少なく済むため, 計算効率および必要メモリ量の点で有利である.
- 各 GUB 制約の決定変数を効率指標に基づいてランク付けし, それに基づいて解の改善をはかる過程を GA に組み込む, すなわちハイブリッド化により, GUB 構造性を利用した効率的な解の探索が可能になった.

## 第8章

# ハイブリッド型遺伝的アルゴリズムによる二重線形最適化問題の最適設計

### 8.1 問題の概要

本論文では、二重線形最適化問題をハイブリッド型遺伝的アルゴリズムによって最適化する。二重線形最適化問題は、線形関数の最大化または最小化を、線形制約条件の下で行う。この問題は、多くの実用問題で発生し、効率的な最適化アルゴリズムの開発が求められている。本論文では、遺伝的アルゴリズムの探索能力と局所最適化アルゴリズムの精度を組み合わせることで、この問題を効果的に解決する。まず、遺伝的アルゴリズムを用いて大域的な最適解を探索し、その後、局所最適化アルゴリズムを用いてその解を高精度に最適化する。このハイブリッド型アプローチは、従来の最適化手法よりも優れた性能を示すことが期待される。

## 第 8 章

# ハイブリッド型遺伝的アルゴリズムによる二目的のシステム信頼性の最適設計

### 8.1 まえがき

システム信頼性の最適設計問題は、一般に非線形整数計画問題として定式化されるが、この問題の解法の一つに、0-1線形計画問題に変換する方法が知られている。このとき、GUB構造と呼ばれる特殊な構造が付加される。そのため、特に大規模な現実問題を解くにあたっては、このGUB構造を考慮することが重要となる。

前章で取り上げたように、システム信頼性の最適設計問題に対し、De Novo計画問題を導入することにより、より柔軟な設計が可能となる。このDe Novo計画問題は、ナップサック問題に変換される。このナップサック問題は、単純な構造にも関わらず予算編成問題、献立計画問題や製品価格など多方面に応用され、組合せ最適化問題の中で重要な位置を占めており、解法を研究することは非常に有用である。さらに、この問題はNP-困難であることが示されており、分枝限定法や動的計画法などの最適解法よりも近似解法の研究が進められており、ラグランジュ緩和問題を適用した解法や連続緩和問題へ変換した解法などが提案されている。また、システム信頼性の最適設計問題においては、目的関数として、信頼度の最大化と同時に、コストの最小化を考慮する場合が多い。そのため、本章では、多目的計画問題の一つのケースとして、二目的のナップサック問題を取り上げる。

さらに、GAは近年、組合せ最適化問題に対する効果的な解法として、各種の経営工学／オペレーションズ・リサーチ関連の分野でも数多くの成果が報告されている[21, 22, 24, 25, 86]。最近では、GAが特に有効とされる広域探索に加え、比較的弱い

とされる局所探索を行うためのヒューリスティックアルゴリズムを組合せることによって、GAの探索効率を向上させるハイブリッドアルゴリズムが数多く報告されている。

本章では、GUB構造を伴う二目的ナップサック問題を解くために新しいハイブリッド型GAを提案する。提案する解法は、次に示す特徴を持っている。

- GUB構造性を生かした新しい遺伝子表現を適用する。この遺伝子表現により、各種の遺伝的操作に対してもGUB構造性が保持されるため、バイナリ表現に基づく遺伝子表現に比べ取り扱いが非常に簡単となる。
- 必要メモリ量のはるかに少なくすむため、処理時間の点で有利なほか、特に大規模な問題への適用の点でも有利である。
- GUB構造性を利用して、GAの各世代において、効率指標に基づく解の改善過程を導入することにより、効率的に解の探索を行うアルゴリズムを導入（ハイブリッド化）する。

提案する解法の有効性を示すために、数値例として大規模なシステム信頼性の最適設計問題を取り上げる。

## 8.2 二目的のシステム信頼性設計問題

### 8.2.1 GUB構造を伴う二目的ナップサック問題

GUB制約を伴う二目的ナップサック問題(BKP/GUB)は、次のように定式化されるBKP/GUB:

$$\max f_1(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} c_{ij}^1 x_{ij} \quad (8.1)$$

$$\max f_2(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} c_{ij}^2 x_{ij} \quad (8.2)$$

$$\text{s. t. } g(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^{n_i} a_{ij} x_{ij} \leq b \quad (8.3)$$

$$\sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (8.4)$$

$$x_{ij} = 0 \text{ or } 1, \quad \forall i, j \quad (8.5)$$

ここで、式(8.3)はナップサック制約、また式(8.4)はGUB制約と呼ばれる。 $c_{ij}^1$ 及び $c_{ij}^2$ はGUB制約 $i$ の $j$ 番目の目的関数の係数である。また、 $a_{ij}$ はGUB制約 $i$ の $j$ 番目

のナップサック制約の係数,  $x_{ij}$ は GUB制約  $i$ の  $j$ 番目の決定変数である. さらに,  $n_i$ は GUB制約  $i$ の決定変数の数を表す.

二目的最適化問題を含む多目的最適化問題においては, 一般に非優越解あるいはパレート最適解の概念が適用される.

二目的最適化問題を含む多目的最適化問題では, 一般に何らかのスカラー化法により一目的の最適化問題に変換されるが, その代表的なスカラー化法の一つに重み係数法がある. 重み係数法は, 複数個の目的関数の重みつき総和を単一の目的関数として最適化する方法であるが, 一般に次元(単位系)の異なる複数の目的関数を取り扱うため, 単純に重みをつけて足し合わせた場合, 問題が生じる. 本章では, こうした問題を解決するために, GAの各世代において得られる個体数に相当する個数の目的関数について, 次のように各目的関数を正規化する.

$$\hat{f}_k(\mathbf{x}) = (f_k(\mathbf{x}) - \bar{f}_k) / \sigma_k, \quad k=1,2 \quad (8.6)$$

ここで,  $\hat{f}_k(\mathbf{x})$ は正規化された  $k$ 番目の目的関数,  $\bar{f}_k$ は  $k$ 番目の目的関数の平均値, また  $\sigma_k$ は  $k$ 番目の目的関数の標準偏差である.

各目的関数に相対的な重み  $\lambda_k$ ,  $k=1,2$ , を掛け合わせることににより, 次のような一目的の目的関数に変換される.

$$\begin{aligned} F(\mathbf{x}) &= \sum_{k=1}^2 \lambda_k \hat{f}_k(\mathbf{x}) \\ \lambda_1 + \lambda_2 &= 1 \\ \lambda_1, \lambda_2 &\geq 0 \end{aligned} \quad (8.7)$$

ここで導入するスカラー化法においては, 重み  $\lambda_k$ , ( $k=1,2$ )は意思決定者の各目的関数に対する重要度を反映すると考えられることから, 意思決定者の選好構造を表すという観点から, 重要な役割をもつ. しかしここでは重み  $\lambda_1$  及び  $\lambda_2$ はそれぞれ意思決定者によって与えられるものとする.

## 8.3 ハイブリッド型遺伝的アルゴリズム

### 8.3.1 遺伝子表現

本章では, GUB構造性を生かした, 新しい遺伝子表現を導入する. 導入する染色体は, 一つの GUB制約に関する情報を一つの遺伝子として表現する. これにより,

各 GUB 構造において選ばれる決定変数の位置および一つの GUB 制約に含まれる決定変数の数を一つの遺伝子として表すことにより, 簡単な表現にも関わらず, GUB 制約の構造性を失うことなく遺伝的操作が可能である点に大きな特徴がある. 従って, 従来から提案されている各種の遺伝的操作の中で, 対象とする問題に適した操作を選んで適用することが可能である.

$s_i$  は GUB 制約  $i$  において値,  $1 \leq s_i \leq q_i$  をとる決定変数の位置を表す. 従って, 染色体の各遺伝子  $s_i$ , ( $i = 1, 2, \dots, n$ ) は次のように示される:

$$s_i = j \quad \text{if } x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (8.8)$$

$$1 \leq j \leq n_i$$

この遺伝子表現により, 各世代において  $p$  番目の染色体  $\mathbf{V}_p$  は次のように表される:

$$\mathbf{V}_p = [s_{1p} \quad s_{2p} \quad \dots \quad s_{np}] \quad .$$

### 8.3.2 評価関数

GA において, 次の手順によりパレート解集合を保存する.

- (1) 各染色体に対応する目的関数値を計算する.
- (2) 評価関数値に基づいてパレート解を保存する.

評価関数値計算の手順を次に示す.

#### 評価関数値の計算

ステップ 1: BKP/GUB 問題において, 各目的関数値  $f_1(\mathbf{V}_p)$  および  $f_2(\mathbf{V}_p)$  をそれぞれ計算する.

ステップ 2: 各目的関数の正規化された値  $\hat{f}_k(\mathbf{V}_p)$  を式 (8.8) から計算する.

ステップ 3: 式 (8.13) より, 各染色体の評価関数値を計算する.

$$eval(\mathbf{V}_p) = F(\mathbf{V}_p) \quad (8.9)$$

$$p = 1, 2, \dots, pop\_size$$

### 8.3.3 遺伝的操作

本研究で適用する遺伝的操作は次のとおりである。既に述べたとおり、提案する解法はGUB構造的性を有効に表現する遺伝子表現が可能であるため、問題に適した遺伝的操作を適用することが可能である。本章では、提案する遺伝子表現および計算法の有効性を明確にするために、遺伝的操作として従来から知られる一般的な操作を適用するものとする。

**交叉:** 交叉は、二つの親染色体間で一つの交叉位置でそれぞれの部分解を交換する一点交叉を用いる。

**突然変異:** ランダムに選ばれた遺伝子座において、 $1 \leq s_i \leq nq_i$ を満たす遺伝子に置き換える。

**選択:** 親染色体および交叉や突然変異によって生成される子染色体からルーレット選択により次世代の個体を選択する。

### 8.3.4 決定変数のランクに基づく並換え

GUB制約の構造的性を利用し、限られた予算の下での最大化問題を解くための効率的な解の探索を行うために、GAを適用する前に、各GUB制約の決定変数に対して次のようなランク付けに基づく並換えを行う。

**ステップ1:** 同一集団内において、各GUB制約の決定変数をランク付けをするために、次式に基づいて指標を計算する。

$$e_{ij} = c'_{ij} / a_{ij}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n_i \quad (8.10)$$

ここで、 $c'_{ij}$  は目的関数  $F(\mathbf{x})$  の  $t$  番目の係数で、次式により示される。

$$c'_{ij} = \frac{\lambda_1}{\sigma_1} c_{ij}^1 + \frac{\lambda_2}{\sigma_2} c_{ij}^2, \quad \forall i, j \quad (8.11)$$

この指標は、各決定変数のコストパフォーマンスを表すと考えられることから、効率指標と呼ぶ。

**ステップ2:** 次式により、各GUB制約において指標の高い順に決定変数をランク付けする。

$$\bar{j}_i = [\bar{j}_{i1} \bar{j}_{i2} \dots \bar{j}_{in_i}] = \underset{j}{\text{indsort}}\{e_{ij} | j = 1, 2, \dots, n_i\}, \quad \forall i \quad (8.12)$$

ここで、indsort は降順の並替えを表す。

**Step 3:** 各GUB制約において、効率指標に基づいて各決定変数を降順に並べ換える。

## 8.4 計算法

本論文で提案する GUB 構造的な GA に基づく計算法は次のとおりである。本計算法は、ステップ2およびステップ6において、GUB 構造的な特徴を利用して、解空間においてより効率指標の高い解を探索するよう方向付けするヒューリスティクスを組み込んだハイブリッド化を行うことにより、効率的に最良解を得ることを可能としている。すなわち、一般に広域探索を得意とする GA に、効率的に局所探索を行う手続きを組み込んでいる。

### ステップ1: パラメータの設定

$max\_gen$ ,  $pop\_size$ ,  $p_c$ ,  $p_m$  をそれぞれ設定する。世代数を  $t=0$  とおく。

### ステップ2: ランクに基づく並換え

各 GUB 制約の決定変数を効率指標に基づいてランクの高い順に並べ換える。

### ステップ3: 初期染色体集団の生成

解の多様性を維持しながら、より効率性の高い解を探索するため、 $n$  要素からなる初期染色体  $V_p$ , ( $p=1, 2, \dots, pop\_size$ ) の集団をランク-1 の近傍でランダムに生成する。

### ステップ4: 評価関数の計算

評価関数値  $eval(V_p)$ , ( $p=1, 2, \dots, pop\_size$ ) を計算する。

### ステップ5: 遺伝的操作

以下の三つの遺伝的操作を行う。

#### (5.1) 交叉

乱数  $r_p$ ,  $p=1, 2, \dots, pop\_size$  を発生させ、 $r_p < p_c$  となる染色体を、前節の交叉法によって交叉させる。交叉によって生成される染色体の数を  $ccnt$  とする。

#### (5.2) 突然変異

乱数  $r_p$ , ( $p=1, 2, \dots, pop\_size + ccnt$ ) を発生させ、 $r_p < p_m$  となる染色体を、前節の突然変異法によって突然変異を行う。突然変異によって生成される染色体の数を  $mcnt$  とする。

#### (5.3) 選択

親染色体と生成された子染色体の評価関数値を計算し、 $pop\_size + ccnt + mcnt$  個の染色体の中から  $pop\_size$  個の染色体を前節のルーレット選択法によって選択する。

### ステップ6: 各染色体に対する解の改善

各染色体について、実行可能性を調べる。実行可能な染色体の場合、各 GUB

制約の変数を一ランク高い決定変数，一ランク低い決定変数にそれぞれ入れ換えた染色体  $V_p^{(1)}, V_p^{(2)}, \dots, V_p^{(2n)}$ , ( $p=1, 2, \dots, pop\_size$ ) および元の染色体の中から最大の目的関数値をもつ染色体を探し，元の染色体と置き換える．実行不可能な染色体の場合，各 GUB 制約の変数を一ランク高い決定変数，一ランク低い決定変数にそれぞれ入れ換えた染色体および元の染色体の中から最大のコストを持つ染色体を探し，元の染色体と置き換える．

ステップ7: 終了条件の判定

$t < max\_gen$  であれば， $t = t + 1$  としてステップ5に戻る． $t = max\_gen$  ならば，そのときの最大の評価関数値をもつ染色体で表現される解を最良解として終了する．

## 8.5 数値例

数値例として，システム信頼性の最適なユニット選択と冗長配分の問題を取り上げる．この問題は，表8.1に示すような，2つの制約条件の下で14のサブシステムからなるシステムの信頼度を最大にする設計案の選択とユニットの配分を求める最適設計問題である．

R-opt:

$$\begin{aligned} \max \quad & R(\mathbf{m}, \alpha) = \prod_{i=1}^{14} \{1 - (1 - R_{i\alpha_i})^{m_i}\} \\ \min \quad & c(\alpha, \mathbf{m}) = \sum_{i=1}^{14} c_{i\alpha_i} m_i \\ \text{s. t.} \quad & \sum_{i=1}^{14} a_{i\alpha_i} m_i \leq 200 \\ & 1 \leq m_i \leq u_i, \quad i = 1, 2, \dots, 14 \\ & 1 \leq \alpha_i \leq \beta_i, \quad i = 1, 2, \dots, 14 \\ & m_i, \alpha_i \geq 0: \text{integer}, \quad i = 1, 2, \dots, 14 \end{aligned}$$

ここで， $m_i$  は GUB 制約  $i$  における冗長ユニット数を，また  $R(\alpha, \mathbf{m})$  は GUB 制約  $i$  において設計案  $\alpha_i$  を選択した場合のシステムの信頼度を表す．また， $c_{i\alpha_i}$  および  $a_{i\alpha_i}$  はそれぞれ GUB 制約  $i$  のコストの関数  $c(\alpha, \mathbf{m})$  およびナップサック制約の係数を表す． $\beta_i$  および  $u_i$  はそれぞれ GUB 制約  $i$  における最大設計案数および最大冗長配分数を表す．さらに， $b$  は資源量を表す．

表 8.1 システム信頼性の最適ユニット選択および冗長配分の数値例

ブロック (i)	Selection( $\alpha_i$ )											
	1			2			3			4		
	$R_{i\alpha_i}$	$c_{i\alpha_i}$	$a_{i\alpha_i}$	$R_{i\alpha_i}$	$c_{i\alpha_i}$	$a_{i\alpha_i}$	$R_{i\alpha_i}$	$c_{i\alpha_i}$	$a_{i\alpha_i}$	$R_{i\alpha_i}$	$c_{i\alpha_i}$	$a_{i\alpha_i}$
1	0.90	1	3	0.93	1	4	0.91	2	2	0.95	2	5
2	0.95	2	8	0.94	1	1	0.93	1	9	-	-	-
3	0.85	2	7	0.90	3	5	0.87	1	6	0.92	4	4
4	0.83	3	5	0.87	4	6	0.85	5	4	-	-	-
5	0.94	2	4	0.93	2	3	0.95	3	5	-	-	-
6	0.99	3	5	0.98	3	4	0.97	2	5	0.96	2	4
7	0.91	4	7	0.92	4	8	0.94	5	9	-	-	-
8	0.81	3	4	0.90	5	7	0.91	6	6	-	-	-
9	0.97	2	8	0.99	3	9	0.96	4	7	0.91	3	8
10	0.83	4	6	0.85	4	5	0.90	5	6	-	-	-
11	0.94	3	5	0.95	4	6	0.96	5	6	-	-	-
12	0.79	2	4	0.82	3	5	0.85	4	6	0.90	5	7
13	0.98	2	5	0.99	3	5	0.97	2	6	-	-	-
14	0.90	4	6	0.92	4	7	0.95	5	6	0.99	6	9

-:設計案なし

この問題は、次に示す0-1変数を導入することにより、GUB制約をもつ二目的ナップサック問題に変換される。

$$x_{ij'} = \begin{cases} 1; & \text{GUB制約 } i \text{ の } j \text{ 番目のユニット} \\ & \text{が選択され、配分されるとき} \\ 0; & \text{それ以外するとき} \end{cases} \quad (8.13)$$

ここで、 $j'$ はGUB制約  $i$  における設計案  $\alpha_i (1 \leq \alpha_i \leq \beta_i)$  および冗長配分数  $m_i (1 \leq m_i \leq u_i)$  を同時に表す。

$$j' = u_i(\alpha_j - 1) + m_i, \quad (8.14)$$

$$1 \leq t \leq n_i, \quad i = 1, 2, \dots, 14$$

ただし  $n_i$  は次式によって決定される:

$$n_i = \beta_i \cdot u_i, \quad i = 1, 2, \dots, 14$$

本数値例においては,  $u_i = 5, i = 1, 2, \dots, 14$  とおく.

信頼性最適設計問題は, 次に示す GUB 制約をもつ二目的ナップサック問題として定式化される.

$$\begin{aligned} \max \quad & f_1(\mathbf{x}) = \sum_{i=1}^{14} \sum_{j=1}^{n_i} c_{ij}^1 x_{ij} \\ \min \quad & f_2(\mathbf{x}) = \sum_{i=1}^{14} \sum_{j=1}^{n_i} c_{ij}^2 x_{ij} \\ \text{s. t.} \quad & g(\mathbf{x}) = \sum_{i=1}^{14} \sum_{j=1}^{n_i} a_{ij} x_{ij} \leq 200 \\ & \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \dots, 14 \\ & x_{ij} = 0 \text{ or } 1, \quad \forall i, j \end{aligned}$$

ここで, 本数値例では,  $\lambda_1 = 0.7, \lambda_2 = 0.3, p_c = 0.5, p_m = 0.6, \text{pop-size} = 100$  とおく.

この問題を 5000 世代まで実行した結果, 第 3442 世代目に最良な染色体  $\mathbf{V}^*$  が得られた:

$$\mathbf{V}^* = [2 \ 4 \ 16 \ 13 \ 7 \ 18 \ 12 \ 3 \ 2 \ 9 \ 2 \ 2 \ 6 \ 7]$$

このときの信頼度は 0.99, またコストは 99.0, および制約式は 175.0 である. 提案解法による設計案の選択およびユニットの配分数は表 8.2 に示すとおりである.

表 8.2 提案解法による設計案の選択およびユニットの配分数

GUB 制約 ( $i$ )	1	2	3	4	5	6	7
設計案 ( $\alpha_i$ )	1	1	4	2	1	3	1
配分数 ( $m_i$ )	2	4	1	3	2	3	2
GUB 制約 ( $i$ )	8	9	10	11	12	13	14
設計案 ( $\alpha_i$ )	1	1	2	1	1	2	2
配分数 ( $m_i$ )	3	2	4	2	2	1	2

提案する解法 (HGA(0-1KP/GUB)) の有効性を確かめるために, 各 GUB 制約に対する解のランク付け (前節の計算法:ステップ 2) および解の改善 (ステップ 6) の過程を行わない, すなわちハイブリッド化を行わない GA (GA(0-1KP/GUB)) を適用した場合との比較実験を行った結果を図 8.1 および図 8.2 に示す (施行 10 回の平均)

HGA(KP/GUB):KP/GUB のためのハイブリッド型 GA (提案解法)

GA(KP/GUB):KP/GUB のための GA

0-1HGA(KP/GUB):KP/GUB のためのバイナリ遺伝子表現によるハイブリッド型 GA

0-1HGA(KP/GUB):KP/GUB のためのバイナリ遺伝子表現による GA

表 8.3 処理時間・変数の数および必要メモリ量の比較

手法	GA(KP/GUB)	0-1GA(KP/GUB)
処理時間	9.9	17.8
決定変数の数	$n$	$\sum_{i=1}^n n_i$
メモリ量 ( $pop\_size = 100$ )	$n * pop\_size$ 1400	$\sum_{i=1}^n n_i * pop\_size$ 24000

数値は本数値例の場合. メモリ量は1世代あたり

表 8.4 処理時間・変数の数および必要メモリ量の比較 (ハイブリッド化の場合)

手法	HGA(KP/GUB)	0-1HGA(KP/GUB)
処理時間	63.3	134.9
決定変数の数	$n$	$\sum_{i=1}^n n_i$
メモリ量 ( $pop\_size = 100$ )	$n * pop\_size$ 1400	$\sum_{i=1}^n n_i * pop\_size$ 24000

数値は本数値例の場合. メモリ量は1世代あたり

図8.1より, HGA(KP/GUB)とGA(KP/GUB)の比較から, 提案する解法HGA(KP/GUB)はGA(KP/GUB)より明らかに速く収束している. また, 図8.2により0-1HGA(KP/GUB)と0-1GA(KP/GUB)との比較についても成立している. これは, 提案解法が, 効率的に解の探索を行っていることを示している. これは, 提案解法が, GUB構造を利用することによって解の局所探索を行う過程をGAに組み込む(ハイブリッド化)ことにより, 効果的に解の改善がされていることを示している.

表8.3および表8.4から, HGA(KP/GUB)と0-1HGA(KP/GUB)の比較およびGA(KP/GUB)と0-1GA(KP/GUB)の比較の結果, 提案する遺伝子表現はバイナリ表現による遺伝

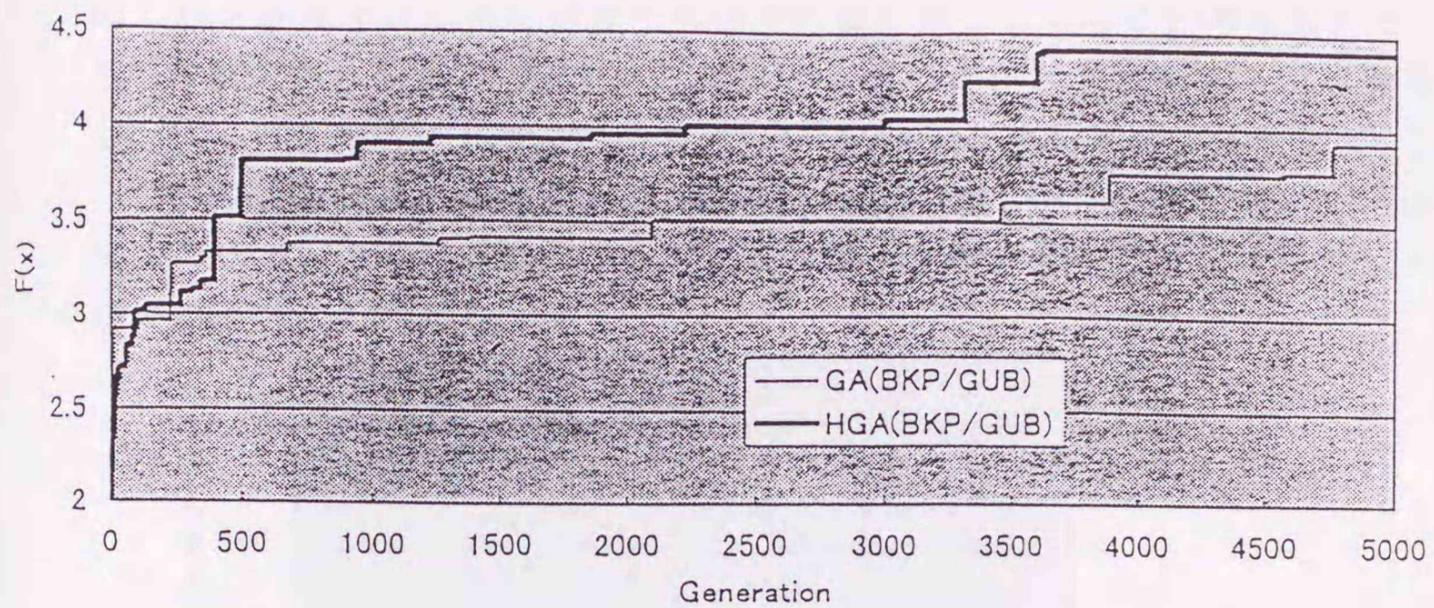


図 8.1 ハイブリッド化の効果

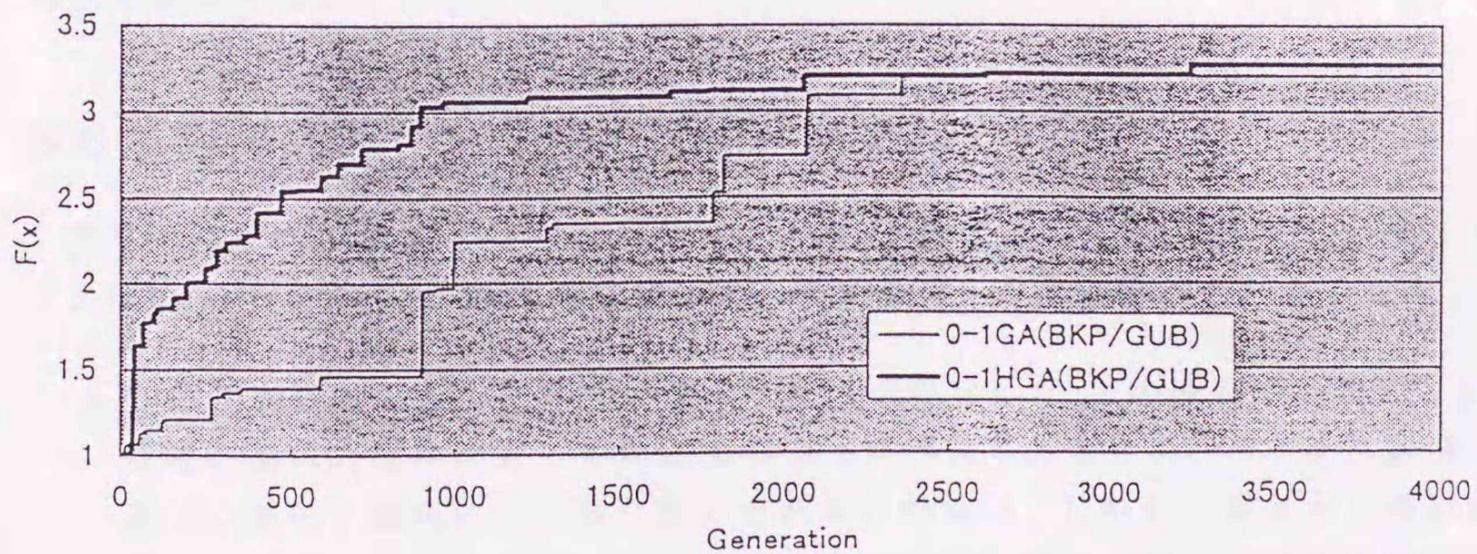


図 8.2 ハイブリッド化の効果 (バイナリ表現による遺伝子表現)

子表現に比べ、それぞれ4.26倍および5倍処理時間が速いことが分かる。これは、同じ解を表現するために必要な遺伝子の数の差によるものと考えられる。すなわち、本数値例において、提案する遺伝子表現の場合14個の遺伝子が必要なのに対し、バイナリ表現による遺伝子表現の場合240個の遺伝子を必要とする。この差は一つの染色体あたりの差であり、個体数が100の場合、その差は $240/14 * 100 (=1714)$ 倍にのぼる。このことから、提案する遺伝子表現は特に大規模な問題に対して有効であることが分かる。

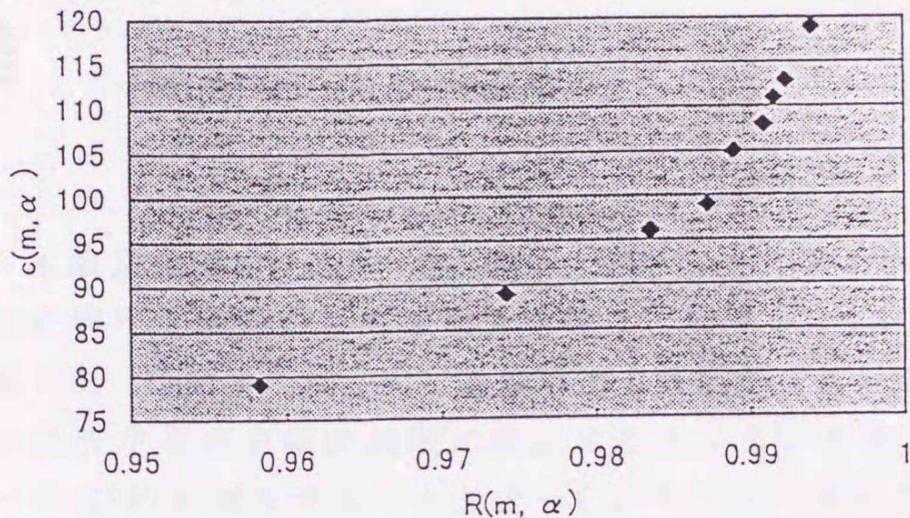


図8.3 パレート最適解集合

パレート最適解の集合を図8.3に示す。最良な妥協解は、パレート最適解の中から意思決定者によって決定される。

## 8.6 むすび

本章では、GUB構造を伴う二目的ナップサック問題を解くための新しいハイブリッド型GAを提案した。提案解法は、次に示すような特徴を持つことを示した。

- GUB構造性を生かした新しい遺伝子表現を提案した。この遺伝子表現により、各種の遺伝的操作に対してもGUB構造性が保持されるため、バイナリ表現に基づく遺伝子表現に比べ取り扱いが非常に簡単で、しかも必要メモリ量がはるかに少なくすむため、処理時間の点で有利なほか、特に大規模な問題への適用の点でも有利であることを示した。
- さらに、GUB構造性を利用して、GAの各世代において、効率指標に基づく解の改善過程を導入することにより、効率的に解の探索を行うアルゴリズムを導入（ハイブリッド化）した。

## 第 9 章

### 結 言

本論文では、システム信頼性の最適設計問題について、ファジィ目標とファジィ制約を伴う会話型多目的意思決定問題の解法と、ハイブリッド型GAによる解法を提案した。提案する解法は、従来解法と比較してファジィ目標を導入することによって意思決定者の主観的判断に起因するあいまいさを考慮することができるほか、ファジィ制約を導入することによってシステムの使用環境や経済情勢などの変化に対応できる、より現実的で柔軟な意思決定やシステム設計が可能となっている。目標計画問題の会話型解法では、意思決定者の選好構造をより反映させることを目的とした実用的な解法について、その定式化および計算法を提案している。また、GUB構造を考慮した遺伝子表現と探索過程をGAに導入したハイブリッド型のGAにより、システム信頼性の最適設計問題を効率的に解くことが可能となった。また、これらの解法の応用として、より現実的なシステム信頼性の最適設計問題である故障検出切換え装置の故障を考慮に入れたシステムに対する解法、さらにDe Novo計画問題やナップサック問題として定式化される意思決定問題の解法を提案した。

提案する解法は、数値実験により有効性が検証され、現実の問題に適用できる実用性を備えていることを示唆している。

本論文で提案した解法の成果を要約すると、次のとおりである。

1. 意思決定者の主観的判断に基づくあいまいさを、目標値に対する満足度をメンバーシップ関数を規定することによって定量化し、また環境の変化等によって変化するシステム制約条件の右辺定数にゆとりをもたせた会話型の目標計画法に基づく手法を提案した。また、本手法では、各目標に対するDMの重要度を反映させるためにAHPを導入するとともに、多くの非優越解の中から最良の妥協解を決定するための補助手段としてTOPSIS法を採り入れることによって、意思決定者の判断を助けることができる。さらに、具体的な多評価意

思決定として献立に関する最適栄養評価問題を取り上げ、提案する解法の有効性を示した。

2. 信頼性最適化問題においてより現実的な問題に即した解を得ることができる、新しいファジィ多目的0-1計画問題の一解法を提案した。提案する手法では、GUB構造を考慮した効率的な解法をより一般的な多目的0-1計画問題に拡張したほか、ファジィ理論の導入により各目的関数に対し、システム設計者の主観的判断に起因するあいまいさを考慮して”およそ設定した目標値くらい”というかたちで目標値を与えることができる。さらに、各目的関数や制約条件に関して、多評価意思決定のための支援手法として評価されているAHPに基づいた重要度に対応する重みを与えることが可能であるほか、システムの使用環境等により変化するシステム制約条件の右辺定数に上限と下限を設定することによってゆとりをもたせることができるため、システム設計者が満足するような最適設計を行うことができる等、多くの優れた特徴を備えていることを明らかにした。さらに、システム資源が二種類でかつ線形な制約条件の下で、線形なコストおよび作業人員関数をそれぞれ最小に、非線形な信頼度関数を最大にする最適ユニット選択と冗長配分の設計問題を取り上げ、本手法の有効性を確認した。
3. 切換え装置の故障を考慮した不完全なFDSをもつシステムの冗長ユニット配分問題において、目的関数に対するファジィ目標を設定できるため、目標値に対するシステム設計者のあいまいさを考慮に入れることが可能であるとともに、各目的関数および制約条件に対する目標値の達成度を最大にするような決定変数を、各GUB制約の中から効率的に選ぶことができるように改良した解法を提案した。また、システムの使用環境の変化等を表すことができるため、より現実的な最適信頼性設計が可能である。さらに、提案した解法の有効性を示すために、三種類の非線形なシステム資源の制約の下で、非線形な信頼度関数を最大に、また線形なコスト関数を最小にする冗長ユニット配分の最適設計問題を取り上げた。
4. 数種類の故障モードを伴うとともにファジィ目標とファジィ制約をもつシステム信頼性最適設計問題を、GAを導入することによって解く解法を提案した。この解法は、数種類の故障モードを考慮することによって、直列/並列のユニットが複雑に接続されて構成される現実のシステム信頼性の設計問題を系統的に解析することができるため、よりきめ細かい現実のシステム信頼性の最適設計問題に対応することが可能となった。また、ファジィ目標およびファジィ制約を導入することによって、システム設計者の主観的判断のあいまいさ

やシステムのおかれた環境の変化に対応した制約条件の右辺定数の変動を考慮することができるため、柔軟で現実的な設計が可能となった。また、システム設計段階において、一般に相反する目標であると考えられるシステム設計者の要求するシステムの信頼度と、システムの重量、容積、コストなどに関する制約のトレードオフ関係を同時に考慮できるとともに、それらのあいまいさを考慮できるため、システム設計者にとってより柔軟で満足度の高いシステム設計が可能となった。さらに、このシステムの信頼性最適設計問題を、GAを導入することによって非線形のまま扱うことができるため、取扱いが容易なうえ、コンピュータメモリの点でも有利であるとともに、単純GA導入したことによって最良解を比較的簡単なアルゴリズムによって得ることが可能となっている。

5. 与えられた予算額の中で資源の最適配分を含む計画問題の最適な設計を行うことができる De Novo 計画問題として定式化されるシステム信頼性の最適設計問題を解くためのハイブリッド型GAによる解法を提案した。提案する解法は、GUB構造を有効に表現する新しい遺伝子表現を導入することにより、各種の遺伝的操作に対してGUB構造性が保持されるため、取り扱いが容易である。また、0-1変数によるGUB構造の表現を行った場合に比べ、必要となる遺伝子数をはるかに少なく済むため、計算効率および必要メモリ量の点で有利であることを示した。さらに、各GUB制約の決定変数を効率指標に基づいてランク付けし、それに基づいて解の改善をはかる過程をGAに組み込む、すなわちハイブリッド化により、GUB構造性を利用した効率的な解の探索が可能になっている。さらに、二目的のシステム信頼性の最適設計問題に対する応用のための解法を提案した。

## 参考文献

- [1] 村松隆一, 大塚正一, 大塚正一, 大塚正一, 大塚正一 (1987)
- [2] 大塚正一, 大塚正一, 大塚正一, 大塚正一, 大塚正一 (1988)
- [3] Anderson, M. N. and David B. Fowler. "The Knapsack Problem with Multiple Classes." *Oper. Res. Soc. Japan*, 28(1), pp. 103-111 (1987)
- [4] Araki, Y. S. and S. Ozawa. "Frankly-convex, alternative multiprocess optimization algorithms." In *Proceedings of the 1987 International Conference on Intelligent Systems for Design Support*, pp. 1-10. Springer-Verlag, 1987.
- [5] Bevilacqua, F. *Design for Manufacture*. John Wiley & Sons, New York, 1976.
- [6] Bhatia, R. K. and K. S. Chiu. "The Design of a Hierarchical Decomposition and Large Systems." *IEEE Trans. Syst., Man, Cybernet.*, 13(1), pp. 1-10 (1983)
- [7] Chang, R. M. and M. S. Tsai. "The Design of a Hierarchical Decomposition Algorithm." *Intern. J. of Comput. & Inform. Engg.*, Vol. 10, No. 1-4, pp. 151-155 (1978)
- [8] Chao, P. S. *Mathematical Computer Systems Design*. Wiley, 1987.
- [9] Chen, W., K. H. Lu, and E. Zhang. "A Hierarchical Approach with the Large Multiplicity and Non-integer-Valued Variables." In *Proceedings of the 1987 International Conference on Intelligent Systems for Design Support*, pp. 1-10. Springer-Verlag, 1987.
- [10] Chen, W. "Efficiently Decomposing by BIL Programming of a System with Convex Objective." *IEEE Trans. on Syst., Man, Cybernet.*, Vol. 17, No. 1, pp. 217-224 (1987)
- [11] 大塚正一, 大塚正一, 大塚正一, 大塚正一, 大塚正一 (1987)
- [12] Gen, M. and A. S. I. "Hierarchical Multiple Objective Linear Programming Systems Implemented on a Personal Computer." *Intern. J. of Comput. & Inform. Engg.*, Vol. 12, No. 1-4, pp. 217-224 (1987)

## 参考文献

- [1] 阿部俊一：“システム信頼性解析法”，日科技連(1987).
- [2] 安居院猛，長尾智晴：『ジェネティックアルゴリズム』，昭晃堂，1993.
- [3] Aggarwal, V., N. Deo and D. Sarkar : "The Knapsack Problem with Disjoint Multiple-Choice Constraints", *Naval Research Logistics*, Vol.39, pp. 213-227(1992).
- [4] Arbel, A. S. & S. S. Oren : "Priority-based Interactive Multicriteria Optimization Algorithm", Y. Sawaragi, K. Inoue & H. Nakagawa eds. *Toward Interactive and Intelligent Decision Support System*, Vol.1, Springer Verlag, pp.163-171(1987).
- [5] Birolini, A. : Design for Reliability, in A. Kusiak ed. "Concurrent Engineering", pp.307-348, Wiley(1993).
- [6] Bulfin, R. L. & C. Y. Liu : "Optimal Allocation of Redundant Components for Large Systems", *IEEE Trans. Rel.*, R-34, 3, pp.241-247(1985).
- [7] Cheng, R., M. Gen & M. Sasaki : "Film-copy Deliverer Problem Using Genetic Algorithms", *Inter. J. of Comp. & Indust. Engg.*, Vol.29, Nos.1-4, pp.549-553 (1995)
- [8] Dhillon, B. S. : "Reliability in Computer System Design", Ablex Pub.(1987).
- [9] Kuo, W., H. H. Lin & Z. Zhang : "Reliability Optimization with the Large-Multiplier and Branch-and-Branch Technique", in S. Rai & D. P. Agrawal eds. *Distributed Computing Network Reliability*, IEEE Comp. Soc. Press, pp.275-281(1990).
- [10] Gen, M. : "Reliability Optimization by 0-1 Programming for a System with Several Failure Modes", *IEEE Trans. on Rel.*, Vol.R-24, pp.206-210(1975), also in [69] pp.252-256.
- [11] 玄 光男：“数種類の故障モードを伴うシステム信頼性の0-1計画法による最適化”，電子通信学会論文誌，Vol.7, No.7, pp.373-380(1975).
- [12] Gen, M. & K. Ida : "Interactive Multiple Objective Linear Programming System Implemented on a Microcomputer", *Int. J. of Comp. & Indust. Engg.*, Vol.11, Nos.1-4, pp.220-224(1986).

- [13] 玄 光男, 井田憲一, 李 在旭: "GUB構造を伴う0-1線形計画法によるシステムアベイラビリティの最適選択と配分", 電子情報通信学会論文誌(D), **J71-D**, No.10, pp.2140-2147(1988).
- [14] Gen, M., K. Ida, M. Sasaki & J. U. Lee: "Algorithm for Solving Large-Scale 0-1 Goal Programming and Its Application to Reliability Optimization Problems", *Inter. J. of Comp. & Indust. Engg.*, **17**, Nos.1-4, pp.525-530(1989).
- [15] 玄 光男, 井田憲一, 佐々木正仁: "会話型逐次目標計画問題による多目的意思決定問題の一解法", 電子情報通信学会論文誌(A), **J73-A**, No.3, pp.645-648(1990).
- [16] 玄 光男, 井田憲一, 李在旭: "GUB構造を伴う0-1目標計画問題の一解法とそのシステム信頼性の最適化問題への応用", 電子情報通信学会論文誌(A), **73-A**, No.3, pp.557-563(1990).
- [17] Gen, M., K. Ida & M. Sasaki: "A Computational Algorithm for Solving 0-1 Goal Programming with GUB Structure and Its Application for Optimization Problem of a System Reliability", *Elect. & Comm. in Japan Part 3*, Vol.**73**, No.12, pp.88-96(1990).
- [18] 玄 光男, 井田憲一: "ファジィ多目的0-1線形計画法とその応用", オペレーションズ・リサーチ, Vol.**36**, No.9, pp.450-454(1991).
- [19] 玄 光男, 井田憲一, 張 曉軍: "反復的積形式法による目標計画問題の一解法", 電子情報通信学会論文誌(A), **J75-A**, No.10, pp.1596-1599(1992).
- [20] Gen, M., K. Ida, Y. Tsujimura & C.E. Kim: "Large-scale 0-1 Fuzzy Goal Programming and Its Application to Reliability Optimization Problem", in special issue "Industrial Engineering Technology in Japan", *Inter. J. of Comp. & Indust. Engg.* Vol.**24**, No.4(1993).
- [21] 玄 光男, 横田孝雄: "遺伝的アルゴリズムの国際会議とパッケージ", *bit*, Vol.**25/26**, No.12/1 pp.52-59/pp.105-113(1993/1994).
- [22] 玄光男, 横田孝雄, 田口雄章: "パソコンGeneIntパッケージ1-3: 遺伝的アルゴリズムによる非線形整数計画問題パッケージと使用法", *パソコンリテラシ*, Vol.**19**, Nos.2-4, pp.89-103, pp.29-40, pp.49-61(1994).
- [23] Gen, M. and B. Liu: Evolution Program for Production Plan Problem, *Engg. Design & Auto.*, Vol.**1**, No.3, pp.199-204(1995).
- [24] Gen, M. and R. Cheng: *Genetic Algorithms and Engineering Design*, John & Wiley Sons, New York(1997).

- [25] Gen, M. & J.R.Kim: "GA-based Optimisation Reliability Design", in P. Bentley ed: *Evolutionary Design by Computers*, John Wiley & Sons, Inc.(1998).
- [26] Goldberg, D. E. : *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley(1989).
- [27] Hadj-Alouane, A.B. and J. C. Bean : "A Genetic Algorithm for the Multiple-Choice Integer Program", *Operations Research*, Vol.45, No.1, pp. 92-101(1997), also in [69] pp.252-256.
- [28] Hannan, E. L. : "On Fuzzy Goal Programming", *Decision Science*, Vol.12, pp.522-531(1981).
- [29] Hinterding, R. : "Mapping ,Order-independent Genes & the Knapsack Problem", *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE Press, Orlando,FL, pp. 13-17(1994).
- [30] Hwang, C. L. & A. S. M. Masud : "*Multiple Objective Decision Making*", Springer-Verlag(1979).
- [31] Hwang, C. L., H. B. Lee, F. A. Tillman & C. H. Lie : "Nonlinear Integer Goal Programming Applied to Optimal System Reliability" , in S. Rai & D. P. Agrawal eds. *Distributed Computing Network Reliability*, IEEE Comp. Soc. Press, pp.257-264(1990).
- [32] Hwang, C. L., Y. J. Lai & M. D. Ko : "ISGP-II for Multiobjective Optimization with Imprecise Objective Coefficients", *Computers & Ops. Res.*, Vol.20, No.5, pp.503-514(1993).
- [33] 井田憲一, 玄 光男 : "多目的線形計画のマイコン・パッケージ開発と数値実験", 日本経営工学会誌, Vol.39, No.4, pp.247-253(1988).
- [34] 井田憲一, 玄 光男 : "TOPSIS法による多目的線形計画の一解法と数値実験", 電子情報通信学会論文誌(A), J72-A, No.4, pp.731-733(1989).
- [35] 井田憲一, 玄 光男 : "0-1線形計画法とAHPによるLife-Supportシステム信頼性の最適化", 電子情報通信学会論文誌(A), J72-A, No.2, pp.367-374(1989).
- [36] Ida, K, M. Gen & M. Sasaki : "System Reliability Optimization with Incomplete FDS by Large Scale 0-1 Linear Programming", *Proc. of 16th InterRam*, Vol.16, pp.447-453(1989).
- [37] 井田憲一, 佐々木正仁, 玄 光男 : "GUB制約をもつ二目的0-1線形計画法によるシステム信頼性の最適化", 電子情報通信学会論文誌(A), J72-A, No.7, pp.1117-1124(1989).

- [38] 井田憲一：“多目的線形計画技法の開発とその大規模な信頼性最適化問題への応用に関する研究”，博士学位論文，工学院大学(1989).
- [39] Ida, K., M. Sasaki & M. Gen：“Optimization of System Reliability Using Bicriteria 0-1 Linear Programming with GUB Structure”，*Elect. & Comm. in Japan*, Vol.73, No.6, pp.97-104(1990).
- [40] 井田憲一, 玄 光男：“GUB制約をもつ0-1線形計画の解法パッケージ開発と数値実験”，電子情報通信学会論文誌(A), J75-A, No.7, pp.1222-1225(1992).
- [41] Ida, K., M. Gen & T. Yokota：“System Reliability Optimization with Several Failure Modes by Genetic Algorithm”，*Proc. of 16th Inter. Conf. on Comp. & Indust. Engg.*, Ashikaga, pp.349-352(1994).
- [42] Inuiguchi, H. Ichihashi, & H. Tanaka：“Fuzzy Programming：A Survey of Recent Developments”，in R. Slowinsky and J. Teghem eds., *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, Kluwer Academic Pub., pp.45-68(1990).
- [43] 北野宏明編：『遺伝的アルゴリズム』，産業図書(1993).
- [44] 小林重信他：遺伝的アルゴリズムの基礎と応用I-IV，オペレーションズ・リサーチ，Vol. 38, No. 5-8(1993).
- [45] Lai, Y. J. & C. L. Hwang：“Interactive Fuzzy Linear Programming”，*Fuzzy Sets and Systems*, Vol.45, pp.169-183(1992).
- [46] Lai, Y. J. & C. L. Hwang：“A New Approach to Some Possibilistic Linear Programming Problems”，*Fuzzy Sets and Systems*, Vol.49, pp.121-133(1992).
- [47] Lai, Y. J. & C. L. Hwang：“*Fuzzy Mathematical Programming*”，Springer-Verlag(1992).
- [48] Lai, Y. J. & C. L. Hwang：“IFLP-II: A Decision Support System”，*Fuzzy Sets and Systems*, Vol.54, pp.47-56(1993).
- [49] Lai, Y. J. & C. L. Hwang：“*Fuzzy Multiple Objective Linear Programming*”，Springer-Verlag(1994).
- [50] Lee, E. S. & R. J. Li：“Fuzzy Multiple Objective Programming and Compromise Programming with Pareto Optimum”，*Fuzzy Sets and Systems*, Vol.53, pp.275-288(1993).

- [51] Masud, A. S. & C. L. Hwang : "Interactive Sequential Goal Programming", *J. Opl. Res. Soc.*, Vol.32, No.5, pp.391-400(1981).
- [52] Michalewicz, Z. : "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag(1992).
- [53] 向殿政男編: コンピュータシステムの高信頼化技術入門, 日本規格協会(1988).
- [54] 中原陽三, 佐々木正仁, 井田憲一, 玄 光男 : "トランスピュータによる整数計画問題の並列分散アルゴリズムとその数値実験", 電子情報通信学会論文誌 (A), J74-A,, No.7, pp.1109-1130(1991).
- [55] 中原陽三, 佐々木正仁, 井田憲一, 玄 光男 : "区間係数を伴う 0 - 1 線形計画問題の効率的な一解法", 日本経営工学会誌, Vol.42, No.5, pp.345-351(1991).
- [56] Nakahara, Y., M. Sasaki & M. Gen : "On the Linear Programming Problems with Interval Coefficients", *Int. J. of Comp. & Indust. Engg.*, Vol.23, Nbs.1-4, pp.301-304(1992).
- [57] 中山弘隆 : "対話型多目的計画法とその応用", オペレーションズ・リサーチ, Vol.36, No.9, pp.435-439(1991).
- [58] 長沢啓行, 西山徳行 : "対話型多重多目的計画法", 日本経営工学会誌, Vol.39, No.2, pp.123-129(1988).
- [59] Narasimhan, R. : "Goal Programming in a Fuzzy Environment", *Decision Science*, Vol.11, pp.325-336(1980).
- [60] Narasimhan, R. : "A Geometric Averaging Procedure for Constructing Supertransitive Approximation to Binary Comparison Matrices", *Fuzzy Sets and Systems*, Vol.8, pp.53-61(1982).
- [61] 日本ファジィ学会編 : "講座ファジィ6, ファジィOR", 日刊工業新聞社(1993).
- [62] 西川禮一, 玉置 久 : "遺伝的アルゴリズムと最適化", オペレーションズ・リサーチ, Vol.38, No.7, pp.347-351(1993).
- [63] 野中保雄, 島岡 淳 : "冗長系", 日科技連(1990).
- [64] 太田英一, 山口俊和, 高野康浩 : "係数間の関係を考慮したファジィ多目標計画法", 日本ファジィ学会誌, Vol.6, No.1, pp.166-176(1994).

- [65] 岡田真幸: "ファジィ数の不等式関係に基づくファジィ数理計画問題の定式化とその解法", 博士学位論文, 大阪府立大学(1994).
- [66] 岡田真幸, 玄光男: "ファジィ多次元0-1ナップサック問題の解法", 日本ファジィ学会誌, Vol.6, No.6, pp. 1171-1181(1994).
- [67] Painton, L. & J. Campbell: "Genetic Algorithms in Operation of System Reliability", *IEEE Trans. On Reliab.*, Vol.44, No.2, pp.172-178(1995).
- [68] Pham, H. & W. Kuo: "Reliability Design for Manufacturing ULSI Circuits", in A. Kusiak ed. *Concurrent Engineering*, pp.349-362, Wiley(1993).
- [69] Rai, S. & D. P. Agrawal eds.: "*Distributed Computing Network Reliability*", IEEE Comp. Soc. Press(1990).
- [70] Ramesh, R., M. K. Karwan & S. Zionts: "Interactive Multicriteria Linear Programming: An Extension of the Method of Zionts and Wallenius", *Naval Research Logistics*, Vol.36, pp.321-335(1989).
- [71] Rao, J. R., R. N. Tiwari, & B. K. Mohanty: "A Preference Structure on Aspiration Levels in a Goal Programming Problem - A Fuzzy Approach", *Fuzzy Sets and Systems*, Vol.25, pp.175-182(1988).
- [72] Reeves, G. R. & S. R. Hedin: "A Generalized Interactive Goal Programming Procedure", *Computers Ops Res.*, Vol.20, No.7, pp.747-753(1993).
- [73] Sakawa, M.: "Multiobjective Optimization by the Surrogate Worth Trade-off Method", *IEEE Trans. Rel.*, Vol.R-27, No.5, pp.311-314(1978).
- [74] 坂和正敏, 荒田賢司: "直列-並列システムにおける大規模多目的最適化手法による信頼性設計", 電子通信学会論文誌(A), J63-D, No.3, pp.256-262(1980).
- [75] Sakawa, M.: "Interactive Multiobjective Optimization by the Sequential Proxy Optimization Technique(SPOT)", *IEEE Trans. Rel.*, R-31, No.5, pp.461-464(1982).
- [76] 坂和正敏: "多目的線形計画問題に対する対話型ファジィ意思決定手法とその応用", 電子通信学会論文誌(A), J65-A, No.11, pp.1182-1189(1983).
- [77] 坂和正敏: "多目的非線形計画問題に対する対話型ファジィ目標計画法", 電子通信学会論文誌(A), J67-A, No.12, pp.1138-1145(1984).

- [78] 坂和正敏：“多目的非線形計画問題に対する対話型ファジィ満足化手法”，システムと制御，Vol.28，No.10，pp.575-582(1984).
- [79] 坂和正敏，南後祐二：“多目的非線形計画問題に対するファジィ満足化領域を用いた対話型意思決定”，電子情報通信学会論文誌(A)，J67-A，No.12，pp.1146-1153(1984).
- [80] 坂和正敏，矢野 均，高橋淳哉：“ファジーパラメータを含む多目的線形分数計画問題に対するパレート最適性”，電子情報通信学会論文誌(A)，J72-A，No.6，pp.931-937(1989).
- [81] 坂和正敏：“ファジィ理論の基礎と応用”，森北出版(1989).
- [82] 坂和正敏，矢野 均：“ファジィ目標を考慮したファジィ係数を含む多目的非線形計画問題に対する新しい解の概念”，日本ファジィ学会誌，Vol.3，No.2，pp.318-328(1991).
- [83] 坂和正敏：“経営数理システムの基礎”，森北出版(1991).
- [84] Sakawa, M: “*Fuzzy Sets and Interactive Multiobjective Optimization*”，Plenum Pub.(1993).
- [85] 坂和正敏，乾口雅弘，澤田一哉：“角型構造の大規模多目的線形計画問題に対するファジー計画法”，電子情報通信学会論文誌(A)，J77-A，No.3，pp.430-38(1994).
- [86] 坂和正敏，乾口雅弘，砂田英昭，澤田一哉：“改良型遺伝的アルゴリズムによるファジィ多目的組合せ最適化”，日本ファジィ学会誌，Vol.6，No.6，pp.177-186(1994).
- [87] 佐々木正仁，玄 光男，井田憲一：“対話型ファジィ目標計画法による多目的意思決定問題の一解法”，電気関係学会四国支部連合大会講演論文集，pp.351-352(1989).
- [88] Sasaki, M., M.Gen & K.Ida: “Interactive Sequential fuzzy Goal Programming”，*Int. J. of Comp. & Indust. Engg.*, Vol.19, Nos.1-4, pp.567-571(1990).
- [89] 佐々木正仁，玄 光男，井田憲一：“ファジィ目標計画法による多目的意思決定問題の一解法”，日本経営工学会春季大会予稿集，pp.121-122(1990).
- [90] 佐々木正仁，玄 光男，井田憲一：“ファジィ多目的0-1計画法による信頼性最適化問題の一解法”，電気関係学会北海道支部連合大会，pp.289-290(1990).
- [91] 佐々木正仁，玄 光男，井田憲一：“ファジィ多目的0-1計画問題とその応用”，日本経営工学会秋季大会予稿集，pp.205-206(1990).
- [92] Sasaki, M., Y. Nakahara, M. Gen & K. Ida: “An Efficient Algorithm for Solving Fuzzy Multiple Objective 0-1 Linear Programming”，*Int. J. of Comp. & Indust. Engg.*, Vol.21, 1-4, pp.647-651(1991).

- [93] 佐々木正仁, 中原陽三, 玄 光男, 井田憲一: "区間係数を伴う 0-1 整数計画法によるシステム信頼性の最適化", 日本経営工学会春季大会予稿集, pp.240-241(1991).
- [94] 佐々木正仁, 玄 光男, 井田憲一: "ファジー多目的0-1計画法による信頼性最適化問題の一解法", 電子情報通信学会論文誌 (A), J74-A, No.6, pp.898-906(1991).
- [95] 佐々木正仁, 井田憲一, 玄 光男: "会話型ファジー多目的線形計画問題の一解法", 日本経営工学会秋季大会予稿集, pp.205-206(1991).
- [96] Sasaki, M., M. Gen & K. Ida: "A Method for Solving Reliability Optimization Problem by Fuzzy Multiobjective Linear Programming", *Elect. & Comm. in Japan*, Vol.74, No.12, pp.106-116(1991).
- [97] 佐々木正仁, 玄 光男: "ファジー係数を伴う対話型多評価意思決定問題の一解法", 日本経営工学会春季大会予稿集, pp.104-105(1992).
- [98] 佐々木正仁, 玄 光男: "会話型目標計画法によるファジー多目的意思決定問題の一解法", 電子情報通信学会論文誌 (A), J75-A, No.10, pp.1590-1595(1992).
- [99] Sasaki, M. & M. Gen: "An Extension of Interactive method for Solving Multiple Objective Linear Programming with Fuzzy Parameters", *Int. J. of Comp. & Indust. Engg.*, Vol.25, Nos.1-4, pp.9-12 (1993).
- [100] 佐々木正仁, 玄 光男: "ファジー多目的線形計画法による不完全な F D S を伴うシステム信頼性の最適化", 電子情報通信学会論文誌 (A), J76-A, No.3, pp.456-465(1993).
- [101] Sasaki, M. & M. Gen: "An Extended Interactive Method for Solving Multiple Objective Linear Programming with Fuzzy Parameters", *Proc. of Inter. Conf. on Sys. Sci. & Sys. Engg.*, Beijing, pp.339-344 (1993).
- [102] Sasaki, M. & M. Gen: "A Method for Solving Reliability Optimization Problem with Incomplete FDS by Fuzzy Multiobjective 0-1 Linear Programming", *Elect. & Comm. in Japan*, Vol.76, No.10, pp.32-43(1994).
- [103] 佐々木正仁, 玄 光男: "遺伝的アルゴリズムによるファジー多目的信頼性最適化問題の一解法", 日本経営工学会春季大会, pp.202-203(1994).
- [104] Sasaki, M., M. Gen & M. Yamashiro: "A Method for Solving Fuzzy De Novo Programming Problem by Genetic Algorithms", *Inter. J. of Comp. & Indust. Engg.*, Vol.29, Nos.1-4, pp.507-511 (1995)

- [105] 佐々木正仁, 横田孝雄, 玄 光男: "遺伝的アルゴリズムによるファジィ信頼性の最適設計問題の一解法", 日本ファジィ学会誌, Vol. 7, No.5, pp.1062-1072(1995).
- [106] Sasaki, M. & M. Gen: "A Method for Solving Bicriteria Knapsack Problem with GUB Structure by Genetic Algorithm", *Beijing Mathematics*, Vol.4, No.2, pp.63-74(1998).
- [107] Sasaki, M. & M. Gen: "Bicriteria Knapsack Problem with GUB Structure by Hybrid Genetic Algorithm", *Proc. of 1st Korea-Japan Joint Conf. on Indust. Engg. & Manag.*, pp.131-138(1998).
- [108] 佐々木正文: "情報システム工学", 共立出版(1989).
- [109] 澤田一哉, 坂和正敏: "角型構造の大規模多目的線形計画問題に対する対話型ファジィ満足化手法", 日本ファジィ学会誌, Vol.6, No.4, pp.669-678(1994).
- [110] Shi, Y.: "Studies on Optimum-Path Ratios in Multicriteria De Novo Programming Problem", *Computers Math. Applic.*, Vol.29, No.5, pp.43-50 (1995)
- [111] Sinha, P. and A. A. Zoltners: "The Multiple-Choice Knapsack Problem", *Operations Research*, Vol.27, No.3, pp. 503-515(1979).
- [112] Steuer, R. E.: "*Multiple Criteria Optimization: Theory, Computation and Application*", Wiley(1986).
- [113] 高崎 茂, 玄 光男, 奥野治雄: "0-1計画法による切換えが不完全なシステム信頼性の最適化", 電子通信学会論文誌(D), J60-D, No.7, pp.515-522(1977).
- [114] 玉置 久, 森 正勝, 荒木光彦: "遺伝アルゴリズムを用いたパレート最適解集合の生成法", 計測自動制御学会論文誌, Vol.31, No.8, pp.1185-1192(1995).
- [115] 田中英夫, 奥田徹示, 浅井喜代治: "Fuzzy 数理計画法", 計測自動制御学会論文誌, Vol.9, pp.607-613(1973).
- [116] Tanaka, H., T. Okuda & K. Asai: "On Fuzzy Mathematical Programming", *Journal of Cybernetics*, Vol.13, pp.37-46(1974).
- [117] Tapia, C. G. & B. A. Murtagh: "Interactive Fuzzy Programming with Preference Criteria in Multiobjective Decision-Making", *Computers & Ops. Res.*, Vol.18, No.3, pp.307-316(1991).
- [118] 高野康, 山口俊和: "ファジィ解を考慮した対話型多目標計画法", オペレーションズ・リサーチ, Vol.37, No.12, pp.603-610(1992).

- [119] Tillman, F. A. : "Optimization by Integer Programming of Constrained Reliability Problems with Several Modes of Failure", *IEEE Trans. on Rel.*, Vol.**R-18**, pp.47-53(1969).
- [120] Tillman, F., C. L. Hwang & W. Kuo : "Optimization of Systems Reliability", Dekker(1980).
- [121] Tiwari, R. N., S. Dharmar & J. R. Rao : "Fuzzy Goal Programming - An Additive Model", *Fuzzy Sets and Systems*, Vol.**24**, pp.27-34(1987).
- [122] Tiwari, R. N., S. Dharmar & J. R. Rao : "Priority Structure in Fuzzy Goal Programming", *Fuzzy Sets and Systems*, Vol.**19**, pp.251-259(1986).
- [123] Wiedey, G. and H.-J. Zimmermann : "Media Selection & Fuzzy Linear Programming", *J. Opl. Res. Soc.*, Vol.**29**, No.11, pp.1071-1084(1978).
- [124] Xu, Z., W. Kuo & H. Lin : "Optimization Limits in Improving System Reliability", *IEEE Trans. Rel.*, Vol.**39**, No.1, pp.51-60(1990).
- [125] Yokota, T. & M. Gen : "Solving for Nonlinear Integer Programming Problem using Genetic Algorithm and its Application", *Proc. of 1994 IEEE Inter. Conf. on Systems, Man, & Cybernetics*, San Antonio(1994).
- [126] Yokota, T., M. Gen & M. Sasaki : "Optimal Design of System Reliability with Fuzzy Data by Genetic Algorithm", *Proc. of 2nd European Congress on Intelligent Tech. and Soft Computing*, Archen (1994.9).
- [127] Yokota, T. & M. Gen : "Solving for Nonlinear Integer Programming Problem using Genetic Algorithm and its Application", *Proc. of 1994 IEEE International Conf. on Systems, Man, & Cybernetics*, San Antonio, pp.1602-1609(1994).
- [128] 横田孝雄, 玄光男, 井田憲一:"故障モードを伴うシステム信頼性最適化問題の遺伝的アルゴリズムによる一解法", *日本ファジィ学会誌*, Vol.**7**, No.1, pp.177-185(1995).
- [129] 横田孝雄, 井田憲一, 玄光男:"遺伝的アルゴリズムによる切換えが不完全なシステム信頼性の最適化", *日本信頼性学会誌*, Vol.**17**, No.2, pp.3-11(1995).
- [130] 横田孝雄, 玄光男, 井田憲一, 田口雄章:"改良遺伝的アルゴリズムによるシステム信頼性設計の最適化", *電子情報通信学会論文誌(A)*, **J78-A**, No.6, pp.702-709(1995).
- [131] Yu, G. : "On The Max-Min 0-1 Knapsack Problem with Robust Optimization Applications", *Operations Research*, Vol.**44**, No.2, pp. 407-415(1996).

- [132] Zeleny, M. : "Optimal System Design with Multiple Criteria: De Novo Programming Approach", *Engineering Costs & Troduction Economics*, Vol.10, pp. 89-94(1986).
- [133] Zeleny, M. : "Optimizing Given Systems *vs.* Designing Optimal Systems: The De Novo Programming Approach", *Int. J. General Systems*, Vol.17, pp.295-307(1990).
- [134] Zemel, E. : "The Linear Multiple Choice Knapsack Problem", *Operations Research*, Vol.28, No.6, pp.1412-1423, 1980.
- [135] Zimmermann, H. J. : "Description and Optimization of Fuzzy Systems", *Int. Journal of General Systems*, Vol.2, pp.209-215(1976).
- [136] Zimmermann, H. J. : "*Fuzzy Sets Decision Making and Expert Systems*", Kluwer Academic Pub.(1987).
- [137] Zimmermann, H. J. : "*Fuzzy Set Theory and its Applications*", 2nd ed. Kluwer Academic Pub.(1991).
- [138] Zionts, S. & J. Wallenius : "An Interactive Programming Method for Solving the Multiple Criteria Problem", *The Institute of Management Science*, Vol.22, No.6, pp.652-663(1976).
- [139] Zionts, S. & J. Wallenius : "An Interactive Multiple Objective Linear Programming Method for a Class of Underlying Nonlinear Utility Functions", *The Institute of Management Science*, Vol.29, No.5, pp.519-529(1983).



## 謝 辞

本論文を作成するにあたり、貴重な時間を割いて多大なご指導を賜りました北海道大学大学院工学研究科システム情報工学専攻数理情報工学講座教授佐藤義治博士に深甚な謝意を表します。さらに、本論文の質の向上のために、貴重なご指導、ご助言をいただきました審査委員の北海道大学大学院同教授新保勝博士、同伊達惇博士、ならびに同大内東博士に深謝いたします。また、著者の実質的な指導者であり、公私ともども日頃から貴重なご教示と有益なご助言、そして暖かいご配慮をいただいている足利工業大学工学部経営情報工学科教授玄光男博士に深く感謝いたします。さらに、日頃から多くのご助言をいただいている前橋工科大学情報工学科教授井田憲一博士をはじめとして、玄光男教授を中心とする足利工業大学における研究グループの研究者である、足利工業大学経営情報工学科助教授辻村泰寛博士、同講師村田忠彦博士に感謝いたします。また、著者の勤務先である足利工業大学工学部経営情報工学科教授山城光雄博士、同横田孝雄博士ならびに教職員の方々に深く感謝いたします。最後に、青山学院大学理工学部経営工学科在学中、たいへんお世話になった阿部俊一博士、そして筑波大学大学院経営・政策科学研究科在学中にご指導を賜った社会工学系教授江藤 肇博士ならびに同門田安弘博士に心から感謝申し上げます。

ステップ9: 各GUB制約において,  $\sum_{k=1}^q w_k e_k$  の値を最小にするような実行可能変数を探す. ここで,  $e_k, (k=1, 2, \dots, q)$  は各目的関数に対する目標値と達成値との差を表す.

$$e_k = |f_k(\mathbf{x}^1) - h_k| \quad (5.24)$$

存在する場合には, その変数の値を1にし, 同じGUB制約で1をもっていた変数の値を0にする.

ステップ10: 各目的関数値の達成度(目標の満足度)  $\mu_k, (k=1, 2, \dots, q)$  がシステム設計者の希望する値より大きければ, 得られた解を最良な妥協解として終了する. そうでなければ, 各目的関数の重要度  $w_k (k=1, 2, \dots, q)$  を再設定してステップ7に戻る.

ただし, 最大化問題の場合は, ステップ3において目的関数の型をminからmaxに換えるほか, ステップ4の条件を次のように変更する.

$$z_k^- < h_k \leq z_k^*, \quad k=1, 2, \dots, q \quad (5.18)'$$

さらに, ステップ5におけるランク付けを降順とする. また,  $g_r(\mathbf{x}^1) \lesssim b_r$  の場合は, ステップ3において制約条件(5.15)が, 次式と置き換えられる.

$$g_r(\mathbf{x}^1) \geq b_r - \delta_r/2, \quad r=1, 2, \dots, T \quad (5.15)'$$

### 5.2.3 最適設計問題への応用

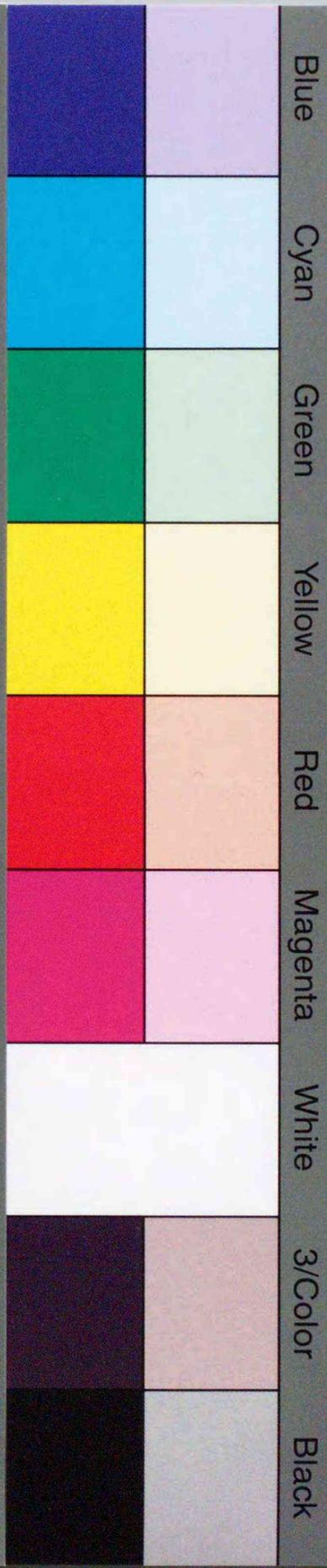
システム信頼性最適設計問題の例として, 表5.2に示すようなシステム資源が2種類でかつ線形な制約条件の下で, 線形なコストおよび作業人員関数をそれぞれ最小に, 非線形の信頼度関数を最大にする最適ユニット選択と冗長配分の設計問題を取り上げる.



inches 1 2 3 4 5 6 7 8  
cm 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

# Kodak Color Control Patches

© Kodak, 2007 TM: Kodak



Blue Cyan Green Yellow Red Magenta White 3/Color Black

# Kodak Gray Scale



© Kodak, 2007 TM: Kodak

A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19

